User Manual

JVM-104-O08
This document has been compiled by Jetter AG with due diligence, and based on the known state of the art. Revisions and further development of our products are not automatically mentioned in a reviewed document. Jetter AG shall not be liable for errors in form or content, or for missing updates, as well as for damages or disadvantages resulting from such failure.
Table of contents

1 Introduction .................................................................................................................................................. 6
  1.1 Information on this document .................................................................................................................. 6
  1.2 Typographical conventions ........................................................................................................................ 6

2 Safety ....................................................................................................................................................... 7
  2.1 General information ................................................................................................................................. 7
  2.2 Purpose .................................................................................................................................................... 7
    2.2.1 Intended use ...................................................................................................................................... 7
    2.2.2 Usage other than intended .............................................................................................................. 7
  2.3 Warnings used in this document ............................................................................................................. 7

3 Product description .................................................................................................................................... 8
  3.1 Design ..................................................................................................................................................... 8
  3.2 Functions ................................................................................................................................................ 8
  3.3 Nameplate .............................................................................................................................................. 9
  3.4 Scope of delivery .................................................................................................................................... 9

4 Technical specifications ............................................................................................................................ 10
  4.1 Dimensions .......................................................................................................................................... 10
  4.2 Electrical properties ............................................................................................................................... 10
  4.3 Mechanical specifications ....................................................................................................................... 10
  4.4 Environmental conditions ...................................................................................................................... 11
  4.5 Display .................................................................................................................................................. 11
  4.6 Acoustic signal generator ....................................................................................................................... 11
  4.7 Ports and interfaces ............................................................................................................................... 11
    4.7.1 CAN port ....................................................................................................................................... 11
  4.8 EMI values ............................................................................................................................................ 12

5 Mechanical installation ............................................................................................................................ 14
  5.1 Requirements for the installation location .............................................................................................. 14
  5.2 Preparing for installation ....................................................................................................................... 15
  5.3 Installing the HMI ................................................................................................................................. 16

6 Electrical connection .................................................................................................................................. 18
  6.1 Pin assignment of M12 connector ........................................................................................................... 19

7 Programming .............................................................................................................................................. 20
  7.1 Abbreviations, module register properties and formats ......................................................................... 20
  7.2 CANopen® STX API ............................................................................................................................. 21
    7.2.1 STX Functions ............................................................................................................................... 21
    7.2.2 CANopen® Object dictionary ....................................................................................................... 22
  7.3 File system ............................................................................................................................................ 22
Table of contents

7.3.1 Directories ................................................................. 23
7.4 Storage options - Overview ................................................ 23
  7.4.1 Types of program and data memory ........................................... 23
  7.4.2 Operating system memory .................................................. 23
  7.4.3 File system memory ......................................................... 23
  7.4.4 Application program memory .............................................. 23
  7.4.5 Flash disk ........................................................................ 24
  7.4.6 Storing registers and variables ............................................. 24
  7.4.7 Special registers ............................................................... 24
  7.4.8 Flag ................................................................................ 25
7.5 Controls and ignition ....................................................... 25
  7.5.1 Input keys ........................................................................ 25
  7.5.2 Digipot ............................................................................ 26
  7.5.3 Ignition and OFF delay ...................................................... 27
7.6 Operating system update .................................................. 28
  7.6.1 Updating the operating system via programming tool .................... 29
  7.6.2 Performing an OS update via JetEasyDownload ......................... 29
7.7 Application program ....................................................... 30
8 Registers - Overview .......................................................... 31
  8.1 Default address on the CANopen® bus ........................................ 31
  8.2 General overview - Registers .................................................. 31
  8.3 I/Os - General overview ....................................................... 31
  8.4 Flags - General overview ...................................................... 31
  8.5 Electronic nameplate ............................................................ 31
  8.6 Electronic name plate (device as a whole) .................................... 32
  8.7 CAN .................................................................................. 32
  8.8 Flash memory ....................................................................... 32
  8.9 System information ............................................................... 33
  8.10 General system registers ....................................................... 33
  8.11 Application program ........................................................... 34
  8.12 File system/data file function ............................................... 35
  8.13 Application registers ............................................................ 35
  8.14 Display .............................................................................. 35
  8.15 Flag .................................................................................. 36
  8.16 System Functions .................................................................. 37
9 Maintenance and repairs .................................................... 39
  9.1 Maintenance, repairs and disposal ....................................... 39
  9.2 Storage and shipment .......................................................... 39
10 Service ............................................................................. 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1 Customer service</td>
<td>40</td>
</tr>
<tr>
<td><strong>11 Spare parts and accessories</strong></td>
<td>41</td>
</tr>
<tr>
<td>11.1 Accessories</td>
<td>41</td>
</tr>
</tbody>
</table>
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:

- Version updates
  Information about new versions of software products or of the operating system of your controller.
- Online help of the JetSym software
  Detailed description of software functions with application examples
- Application-oriented manuals
  Information on file systems and communication interfaces

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.

More information on this subject is available on our website.

Start | Jetter - We automate your success.

INFO

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

2.1 General information
At the time of placing on the market, this product corresponds to the current state of the art and meets the recognized safety rules.
Besides this user manual, laws and regulations in the operator's country are relevant to the operation of the product. The operator is responsible for complying with the directives mentioned below:
- Applicable legislation, rules, and regulations
- Relevant accident prevention regulations
- Accepted safety rules
- EU directives and other country-specific regulations

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

<table>
<thead>
<tr>
<th>Warning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DANGER</strong></td>
<td>High risk</td>
</tr>
<tr>
<td><strong>WARNING</strong></td>
<td>Medium risk</td>
</tr>
<tr>
<td><strong>CAUTION</strong></td>
<td>Low risk</td>
</tr>
<tr>
<td><strong>NOTICE</strong></td>
<td>Material damage</td>
</tr>
</tbody>
</table>
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

![Diagram of the JVM-104-O08 device configuration]

Fig. 1: Device configuration

<table>
<thead>
<tr>
<th>1</th>
<th>TFT display</th>
<th>2</th>
<th>F3 key</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>F4 key</td>
<td>4</td>
<td>Home key</td>
</tr>
<tr>
<td>5</td>
<td>ESC key</td>
<td>6</td>
<td>Digipot</td>
</tr>
<tr>
<td>7</td>
<td>Scroll key</td>
<td>8</td>
<td>Power key</td>
</tr>
<tr>
<td>9</td>
<td>F1 key</td>
<td>10</td>
<td>F2 key</td>
</tr>
<tr>
<td>11</td>
<td>Screw hole</td>
<td>12</td>
<td>M12 male connector</td>
</tr>
<tr>
<td>13</td>
<td>Nameplate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 Functions

- Display: 3.5” TFT, 320x240 px
- CPU: ARM11, 500 MHz, 128 MB RAM, 512 MB Flash
- Surface mount model with connector M12x1
- Input device: 8 keys, 1 DigiPot
- Ports and interfaces: 1x CAN
- Front/rear degree of protection: IP65/IP65:
3.3 Nameplate

Fig. 2: Nameplate

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Logo</td>
</tr>
<tr>
<td>2</td>
<td>Serial number</td>
</tr>
<tr>
<td>3</td>
<td>Barcode</td>
</tr>
<tr>
<td>4</td>
<td>Article number and hardware revision</td>
</tr>
<tr>
<td>5</td>
<td>Model code number</td>
</tr>
</tbody>
</table>

3.4 Scope of delivery

<table>
<thead>
<tr>
<th>Scope of delivery</th>
<th>Item number</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>JVM-104-O08</td>
<td>10001558</td>
<td>1</td>
</tr>
</tbody>
</table>
4 Technical specifications

4.1 Dimensions

![Dimensions diagram](image)

Fig. 3: Dimensions (in mm)

4.2 Electrical properties

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power supply VBAT_ECU</td>
<td></td>
</tr>
<tr>
<td><strong>Rated voltage</strong></td>
<td>DC 12 V or 24 V</td>
</tr>
<tr>
<td><strong>Permissible voltage range</strong></td>
<td>VBAT_ECU: DC 8 V ... DC 32 V</td>
</tr>
<tr>
<td><strong>Permissible voltage range</strong></td>
<td>at terminal 31 (ignition): DC 8 V ... DC 32 V</td>
</tr>
<tr>
<td><strong>Maximum current</strong></td>
<td>2 A</td>
</tr>
<tr>
<td><strong>Typical current consumption</strong></td>
<td>logic circuit (VBAT_ECU): 170 mA at DC 12 V 90 mA at DC 24 V</td>
</tr>
<tr>
<td><strong>Power consumption</strong></td>
<td>Approx. 2 W</td>
</tr>
<tr>
<td><strong>Integrated protective functions</strong></td>
<td>Reverse polarity protection, overvoltage, voltage surges</td>
</tr>
</tbody>
</table>

Tab. 1: Power supply VBAT_ECU

4.3 Mechanical specifications

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vibration</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Broadband noise</td>
<td>10 Hz / 0.005 (m/s²)²/Hz, 200 Hz / 0.02 (m/s²)²/Hz, 300 Hz / 0.01 (m/s²)²/Hz, 350 Hz / 0.002 (m/s²)²/Hz</td>
<td>DIN EN 60068-2-64</td>
</tr>
<tr>
<td>Period</td>
<td>3x 24 h</td>
<td></td>
</tr>
<tr>
<td><strong>Shock resistance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type of shock</td>
<td>Half-sine wave</td>
<td>DIN EN 60068-2-27</td>
</tr>
<tr>
<td>Intensity and duration</td>
<td>30 g (300 m/s²) for 18 ms</td>
<td></td>
</tr>
<tr>
<td>Number and direction</td>
<td>18 shocks in all 6 directions</td>
<td></td>
</tr>
</tbody>
</table>
4.4 **Environmental conditions**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating temperature</td>
<td>-20 … +65 °C</td>
<td>ISO 16750-4</td>
</tr>
<tr>
<td>Climatic conditions</td>
<td>Humid heat</td>
<td></td>
</tr>
<tr>
<td>Storage temperature</td>
<td>-20 … +70 °C</td>
<td>ISO 16750-4</td>
</tr>
<tr>
<td>Air humidity</td>
<td>10 … 95 %</td>
<td>DIN EN 61131-2</td>
</tr>
<tr>
<td>Pollution degree</td>
<td>2</td>
<td>DIN EN 61131-2</td>
</tr>
</tbody>
</table>

**Tab. 3: Environmental conditions**

4.5 **Display**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>TFT LCD flat screen monitor</td>
</tr>
<tr>
<td>Resolution</td>
<td>320 x 240 pixels</td>
</tr>
<tr>
<td>Size</td>
<td>3.5”</td>
</tr>
<tr>
<td>Background lighting</td>
<td>LED, typically 350 cd/m², dimmable</td>
</tr>
<tr>
<td>Horizontal viewing angle</td>
<td>70° to each side</td>
</tr>
<tr>
<td>Vertical viewing angle</td>
<td>50° from above, 70° from below</td>
</tr>
</tbody>
</table>

**Tab. 4: Technical data - Display**

4.6 **Acoustic signal generator**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Speakers</td>
</tr>
<tr>
<td>Volume</td>
<td>83 dB</td>
</tr>
</tbody>
</table>

**Tab. 5: Acoustic signal generator**

4.7 **Ports and interfaces**

4.7.1 **CAN port**

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>125, 250, 500 kBaud</td>
</tr>
<tr>
<td>Protocol</td>
<td>CANopen®</td>
</tr>
<tr>
<td>Default node ID on the CANopen® bus</td>
<td>127 (0x7F)</td>
</tr>
<tr>
<td>Terminating resistor</td>
<td>Does not exist. Must be connected externally.</td>
</tr>
<tr>
<td>Cable specification</td>
<td>Twisted pair conductors, unshielded</td>
</tr>
</tbody>
</table>

**Tab. 6: CAN interface specification**
### Specification - CAN bus cable

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wire cross-section</td>
<td>500 kBaud: 0.34 … 0.50 mm²</td>
</tr>
<tr>
<td></td>
<td>250 kBaud: 0.34 … 0.60 mm²</td>
</tr>
<tr>
<td>Cable capacitance</td>
<td>60 pF/m max.</td>
</tr>
<tr>
<td>Resistivity</td>
<td>500 kBaud: Max. 60 Ω/km.</td>
</tr>
<tr>
<td></td>
<td>250 kBaud: Max. 60 Ω/km.</td>
</tr>
<tr>
<td>Number of cores</td>
<td>2</td>
</tr>
<tr>
<td>Twisting</td>
<td>CAN_L and CAN_H cables are twisted pairwise</td>
</tr>
</tbody>
</table>

**Tab. 7: Specification - CAN bus cable**

### CAN interface specification

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>125, 250, 500 kBaud</td>
</tr>
<tr>
<td>Protocol</td>
<td>CANopen®</td>
</tr>
<tr>
<td>Default node ID on the CANopen® bus</td>
<td>127 (0x7F)</td>
</tr>
<tr>
<td>Terminating resistor</td>
<td>Does not exist. Must be connected externally.</td>
</tr>
<tr>
<td>Cable specification</td>
<td>Twisted pair conductors, unshielded</td>
</tr>
</tbody>
</table>

**Tab. 8: CAN interface specification**

#### Line lengths

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

<table>
<thead>
<tr>
<th>Baud rate</th>
<th>Max. line length</th>
<th>Max. stub line length</th>
<th>Total line length</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 kBaud</td>
<td>100 m</td>
<td>5 m</td>
<td>30 m</td>
</tr>
<tr>
<td>250 kBaud</td>
<td>250 m</td>
<td>10 m</td>
<td>60 m</td>
</tr>
</tbody>
</table>

**Tab. 9: Line lengths**

### 4.8 EMI values

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

#### Pulses to ISO 7637-2

<table>
<thead>
<tr>
<th>Test pulse</th>
<th>Values</th>
<th>Functional class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-450 V</td>
<td>C</td>
</tr>
<tr>
<td>2a</td>
<td>+37 V</td>
<td>A</td>
</tr>
<tr>
<td>2b</td>
<td>+20 V</td>
<td>C</td>
</tr>
<tr>
<td>3a</td>
<td>-150 V</td>
<td>A</td>
</tr>
<tr>
<td>3b</td>
<td>+150 V</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>Ua1: -12 V / 50 ms</td>
<td>A (24 V systems)</td>
</tr>
<tr>
<td></td>
<td>Ua2: -5 V / 500 ms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ua1: -6 V / 15 ms</td>
<td>C (12 V systems)</td>
</tr>
<tr>
<td></td>
<td>Ua2: -2.5 V / 1000 ms</td>
<td></td>
</tr>
<tr>
<td>5b</td>
<td>Load dump, capped 70 V / 2 Ω</td>
<td>A</td>
</tr>
</tbody>
</table>

**Tab. 10: Pulses to ISO 7637-2**

#### Irradiation to ISO 11452

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Functional class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protection against RF noise</td>
<td>20 MHz ... 2 GHz 30 V/m</td>
<td>A</td>
</tr>
</tbody>
</table>

**Tab. 11: Irradiation to ISO 11452**
<table>
<thead>
<tr>
<th>ESD EN 61000-4-2</th>
<th>Values</th>
<th>Functional class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact discharge</td>
<td>±4 kV (to conductive surfaces)</td>
<td>A</td>
</tr>
<tr>
<td>Discharge through air</td>
<td>±8 kV (to insulating surfaces)</td>
<td>A</td>
</tr>
</tbody>
</table>

**Tab. 12: ESD EN 61000-4-2**
5 Mechanical installation

NOTICE

**Damages to material or functional impairment**
Welding on the chassis may cause damages to material of the device, or impair its 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.

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.

![Diagram of space requirements for installation work (in mm)](image)

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

### Avoiding unsuitable installation locations

The following installation locations are unsuitable for mounting the device:

<table>
<thead>
<tr>
<th>Unsuitable installation location</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outdoor installation</td>
<td>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.</td>
</tr>
<tr>
<td>Installation location close to heat-sensitive materials</td>
<td>The materials could become warped or misshapen as a result of heat produced by the device.</td>
</tr>
</tbody>
</table>

**Tab. 13:** Unsuitable installation locations
5.2 Preparing for installation

Use the following accessories [41] for installation:

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

Tab. 14: Mounting accessories

Fig. 5: All dimensions of screw holes are in millimeters.
5.3 Installing the HMI

The illustration below shows how to install the device:

![Installation drawing](image)

**Fig. 6:** Installation drawing

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 x self-locking nuts</td>
<td>2</td>
<td>4 x screw for fixing to the JVM-104-O08</td>
</tr>
<tr>
<td>3</td>
<td>RAM Mount ball</td>
<td>4</td>
<td>Mounting plate with opening for connector</td>
</tr>
<tr>
<td>5</td>
<td>2 x countersunk screws for mounting a RAM Mount ball</td>
<td>6</td>
<td>Alternate position of the RAM Mount ball</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>NOTICE</th>
<th>Damages to material or functional impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Improper implementation of the wiring harness may cause mechanical stress.</td>
</tr>
<tr>
<td></td>
<td>► Protect the cables from bending, twisting or chafing.</td>
</tr>
<tr>
<td></td>
<td>► Install strain reliefs for the connecting cables.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NOTICE</th>
<th>Surges resulting from missing protection or fusing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Surges may cause malfunctions or damage to the product.</td>
</tr>
<tr>
<td></td>
<td>► Protect the voltage inputs from surges according to the requirements.</td>
</tr>
<tr>
<td></td>
<td>► Ensure that the device is handled in accordance with ESD regulations.</td>
</tr>
</tbody>
</table>
6.1 Pin assignment of M12 connector

Features

The M12 connector has the following function:

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

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.

INFO Current consumption

When the JVM-104-O08 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: M12 connector, 8 pins

Mating part

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>e.g. BELDEN</td>
</tr>
<tr>
<td>Manufacturer's item no.</td>
<td>RKCN 8/9</td>
</tr>
<tr>
<td>Wire size range</td>
<td>0.5 mm² (AWG 20)</td>
</tr>
</tbody>
</table>

Tab. 15: Compatible M12 connector (female)
7 Programming

7.1 Abbreviations, module register properties and formats

Abbreviations

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

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R 100</td>
<td>Register 100</td>
</tr>
<tr>
<td>MR 150</td>
<td>Module register 150</td>
</tr>
</tbody>
</table>

Tab. 16: Abbreviations

Module register properties

Each module register is characterized by certain properties. For many module registers most properties are identical. For example, their value after reset is 0. In the following description, module register properties are mentioned only if a property deviates from the default properties listed below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Standard design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of access</td>
<td>Read/write</td>
</tr>
<tr>
<td>Value after reset</td>
<td>0 or undefined (e.g. revision/version number)</td>
</tr>
<tr>
<td>Takes effect</td>
<td>Immediately</td>
</tr>
<tr>
<td>Write access</td>
<td>Always</td>
</tr>
<tr>
<td>Data type</td>
<td>Integer</td>
</tr>
</tbody>
</table>

Tab. 17: Module register properties

Numerical formats

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Format of numerical values</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Decimal</td>
</tr>
<tr>
<td>0x100</td>
<td>Hexadecimal</td>
</tr>
<tr>
<td>0b100</td>
<td>Binary</td>
</tr>
</tbody>
</table>

Tab. 18: Numerical formats

JetSym sample programs

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Format of numerical values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Var, When, Task</td>
<td>Keyword</td>
</tr>
<tr>
<td>BitClear();</td>
<td>Commands</td>
</tr>
<tr>
<td>100 0x100 0b100</td>
<td>Constant numerical values</td>
</tr>
<tr>
<td>// This is a comment</td>
<td>Comment</td>
</tr>
<tr>
<td>// ...</td>
<td>Further program processing</td>
</tr>
</tbody>
</table>

Tab. 19: JetSym sample programs
7.2 CANopen® STX API

CANopen® is an open standard for networking and communication, e.g. in the automotive sector. The CANopen® protocol has been further developed by the CiA e.V. (CAN in Automation) and works on the physical layer with CAN Highspeed in accordance with ISO 11898.

Specifications

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

The key specification documents are:

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

INFO

Further information on the subject CANopen® STX-API can be found in the application-oriented manual „CANopen® STX-API“ which can be downloaded from our Homepage.

7.2.1 STX Functions

Application

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

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

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanOpenInit()</td>
<td>The function CanOpenInit() lets you initialize one of the CAN busses.</td>
</tr>
<tr>
<td>CanOpenSetCommand()</td>
<td>The function CanOpenSetCommand() lets you change the heartbeat status of the device itself and of all other devices (NMT slaves) on the CAN bus.</td>
</tr>
<tr>
<td>CanOpenUploadSDO()</td>
<td>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.</td>
</tr>
<tr>
<td>CanOpenDownloadSDO()</td>
<td>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.</td>
</tr>
<tr>
<td>CanOpenAddPDORx()</td>
<td>By calling up the CanOpenAddPDORx() function, process data, sent by other CANopen® devices, can be entered for reception.</td>
</tr>
<tr>
<td>CanOpenAddPDOTx()</td>
<td>By calling up the CanOpenAddPDOTx() function, process data can be placed on the bus.</td>
</tr>
</tbody>
</table>

INFO

Further information about STX functions can be found in the application-oriented manual "CANopen® STX-API" on our homepage and in the JetSym online help.
### 7.2.2 CANopen® Object dictionary

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

<table>
<thead>
<tr>
<th>Index (hex)</th>
<th>Object (code)</th>
<th>Object name</th>
<th>Data type</th>
<th>Type of access</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>VAR</td>
<td>Device type</td>
<td>Unsigned32</td>
<td>RO (read only)</td>
</tr>
<tr>
<td>1001</td>
<td>VAR</td>
<td>Error Register</td>
<td>Unsigned8</td>
<td>RO</td>
</tr>
<tr>
<td>1002</td>
<td>VAR</td>
<td>Manufacturer status</td>
<td>Unsigned32</td>
<td>RO</td>
</tr>
<tr>
<td>1003</td>
<td>ARRAY</td>
<td>Pre-defined error field</td>
<td>Unsigned32</td>
<td>RO</td>
</tr>
<tr>
<td>1008</td>
<td>VAR</td>
<td>Manufacturer Device Name</td>
<td>String const</td>
<td>1008</td>
</tr>
<tr>
<td>1009</td>
<td>VAR</td>
<td>Manufacturer Hardware Version</td>
<td>String const</td>
<td>1009</td>
</tr>
<tr>
<td>100A</td>
<td>VAR</td>
<td>Manufacturer Software Version</td>
<td>String const</td>
<td>100A</td>
</tr>
<tr>
<td>100B</td>
<td>VAR</td>
<td>Node ID</td>
<td>Unsigned32</td>
<td>RO</td>
</tr>
<tr>
<td>1017</td>
<td>VAR</td>
<td>Producer Heartbeat Time</td>
<td>Unsigned16</td>
<td>R/W (read &amp; write)</td>
</tr>
<tr>
<td>1018</td>
<td>RECORD</td>
<td>Identity</td>
<td>Identity</td>
<td>RO</td>
</tr>
<tr>
<td>1200</td>
<td>RECORD</td>
<td>Server 1 - SDO parameter</td>
<td>SDO parameter</td>
<td>RO</td>
</tr>
<tr>
<td>1201</td>
<td>RECORD</td>
<td>Server 2 - SDO parameter</td>
<td>SDO parameter</td>
<td>R/W</td>
</tr>
<tr>
<td>1203</td>
<td>RECORD</td>
<td>Server 3 - SDO parameter</td>
<td>SDO parameter</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Further information about this subject can be found in the topic "CANopen® Object Directory" in the application-oriented manual "CANopen® STX-API" on our homepage.

### 7.3 File system

The file system lets you access files located on the internal flash disk. When problems occur, a good understanding of the file system is very helpful.

**NOTICE**

Malfunctions caused by missing or damaged system files.

Careless handling of system files may cause malfunctions.

Some files may be protected against read/write access or deletion. Some of these files are virtual files, such as firmware images, or protected files, such as EDS files.

**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

For more information refer to the application-oriented manual "File System" which can be downloaded from our homepage.
7.3.1 Directories

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

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\System</td>
<td>System configuration</td>
</tr>
<tr>
<td></td>
<td>System information</td>
</tr>
<tr>
<td></td>
<td>Splash screen (boot image)</td>
</tr>
<tr>
<td></td>
<td>Screenshot</td>
</tr>
<tr>
<td>\App</td>
<td>Directory for applications</td>
</tr>
<tr>
<td>\Data</td>
<td>Folder for storing data.</td>
</tr>
<tr>
<td>\Windows</td>
<td>Windows CE system directory</td>
</tr>
<tr>
<td>\</td>
<td>RAM disk drive</td>
</tr>
</tbody>
</table>

7.4 Storage options - Overview

7.4.1 Types of program and data memory

The controller features several types of program and data memory. There is, for example, 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.

7.4.2 Operating system memory

The OS is stored to a non-volatile flash memory in the CPU. Therefore, the OS can be executed immediately after the device is powered up.

Properties
- Internal flash memory for storing OS data
- Internal volatile RAM for storing OS data

Type of access
- The user is not allowed to directly access the OS memory.
- Changes to the OS can be made by means of an OS update.

7.4.3 File system memory

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

Properties
- Non-volatile
- Size of internal flash disk: 368 MByte

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

7.4.4 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 \app\program name

Type of access
- By operating system
- By JetSym
- By means of file commands from within the application program
7.4.5 Flash disk

Data can be stored directly on the flash disk. The following rules apply when using a flash disk:

- Do not open more than 8 files at a time.
- Separate directory names by a slash "/".
- The properties of all created files hold their creation date and time as provided by the controller.
- Date, time, and file size are not available for all system files.

Flash disk Capacity

The following disk space is available to the user:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash disk capacity, total</td>
<td>512 MB</td>
</tr>
<tr>
<td>Size of folder <strong>App</strong> (of the a.m. capacity)</td>
<td>64 MByte</td>
</tr>
<tr>
<td>Size of folder <strong>Data</strong> (of the a.m. capacity)</td>
<td>368 MByte</td>
</tr>
</tbody>
</table>

Tab. 20: Flash disk capacity

Properties

The properties of the internal flash disk drive are listed below:

- 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.

7.4.6 Storing registers and variables

Storing to a non-volatile (remenant) 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-remenant) 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.

7.4.7 Special registers

Properties

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

- Global variables with dedicated addresses (%VL)
- When the operating system is launched, special registers are initialized using their default values.
- Register numbers: 100000 ... 999999
Type of access
- By JetSym
- By browser (via HTTP server)
- From HMIs
- From the application program
- From other controllers

7.4.8 Flag
Flags are one-bit operands. This means they can either have the value TRUE or FALSE.

Properties of user flags
- Global variables which are assigned to permanent addresses (%MX)
- Non-volatile
- Quantity: 256
- Flag numbers: 0 ... 255

Properties of overlaid user flags
- Global variables which are assigned to permanent addresses (%MX)
- Non-volatile
- Overlaid by registers 1000000 ... 1000055
- Quantity: 1,792
- Flag numbers: 256 ... 2047

Properties of special flags
- Global variables which are assigned to permanent addresses (%MX)
- When the operating system is launched, special flags are initialized using their default values.
- Quantity: 256
- Flag numbers: 2048 ... 2303

Type of access
- By JetSym
- From HMIs
- From the application program

7.5 Controls and ignition
This chapter describes how to program the controls and ignition of the JVM-104-O08.

7.5.1 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:

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>361000</td>
<td>Bit-coded map of the input keys</td>
</tr>
<tr>
<td>361000.0</td>
<td>[F1] key</td>
</tr>
<tr>
<td></td>
<td>Bit 0 = 1: [F1] key is pressed.</td>
</tr>
<tr>
<td>361000.1</td>
<td>[F2] key</td>
</tr>
<tr>
<td></td>
<td>Bit 1 = 1: [F2] key is pressed.</td>
</tr>
</tbody>
</table>
## Register Description

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>361000.2</td>
<td>[F3] key Bit 2 = 1: [F3] key is pressed.</td>
</tr>
<tr>
<td>361000.3</td>
<td>[F4] key Bit 3 = 1: [F4] key is pressed.</td>
</tr>
<tr>
<td>361000.4</td>
<td>[POWER] key Bit 4 = 1: [POWER] key is pressed.</td>
</tr>
<tr>
<td>361000.5</td>
<td>[SCROLL] key Bit 5 = 1: [SCROLL] key is pressed.</td>
</tr>
<tr>
<td>361000.6</td>
<td>[ESC] key Bit 6 = 1: [ESC] key is pressed.</td>
</tr>
<tr>
<td>361000.7</td>
<td>[HOME] key Bit 7 = 1: [HOME] key is pressed.</td>
</tr>
</tbody>
</table>

**Tab. 21: Input key registers**

### Virtual key codes

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

<table>
<thead>
<tr>
<th>Input key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>[F1]</td>
<td>0x70  (Ground)</td>
</tr>
<tr>
<td>[F2]</td>
<td>0x71  (Ground)</td>
</tr>
<tr>
<td>[F3]</td>
<td>0x72  (Ground)</td>
</tr>
<tr>
<td>[F4]</td>
<td>0x73  (Ground)</td>
</tr>
<tr>
<td>[POWER]</td>
<td>0x74</td>
</tr>
<tr>
<td>[SCROLL]</td>
<td>0x75</td>
</tr>
<tr>
<td>[ESC]</td>
<td>0x1B</td>
</tr>
<tr>
<td>[HOME]</td>
<td>0x24</td>
</tr>
</tbody>
</table>

**Tab. 22: Virtual key codes**

### 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.

#### Digipot registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>363000</td>
<td>This register holds the current count value. If you turn the digipot, the count value increments or decrements. Where: ■ Turning the digipot clockwise increments the count value ■ Turning the digipot counter-clockwise decrements the count value</td>
</tr>
<tr>
<td>363001</td>
<td>Bit 0: 0 = Pushbutton not pressed Bit 0: 1 = Pushbutton pressed</td>
</tr>
<tr>
<td>363002</td>
<td>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.</td>
</tr>
<tr>
<td>363003</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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;

INFO
For more information on this subject, refer to JetSym online help.

7.5.3 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:

<table>
<thead>
<tr>
<th>If ...</th>
<th>... then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 0 = 0</td>
<td>Ignition is switched on and voltage is present at terminal 15 Ignition (+)</td>
</tr>
<tr>
<td>Bit 0 = 1</td>
<td>Ignition is off and no voltage is present at terminal 15 Ignition (+)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>If ...</th>
<th>... and ...</th>
<th>... then ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>... voltage is applied to the HMI, ...</td>
<td>... the ignition is off, ...</td>
<td>... the HMI does not boot up.</td>
</tr>
<tr>
<td>... voltage is applied to the HMI, ...</td>
<td>... the ignition is on, ...</td>
<td>... the HMI boots up.</td>
</tr>
<tr>
<td>... the HMI is powered on, ...</td>
<td>... the ignition is switched off (not the power supply), ...</td>
<td>... the HMI remains switched on.</td>
</tr>
</tbody>
</table>

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:Boolean) :Boolean;`

Function parameters
The `Shutdown()` function has the following parameters:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reboot</td>
<td>System restart</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>The system is shut down</td>
<td>False</td>
</tr>
</tbody>
</table>

Return value

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

<table>
<thead>
<tr>
<th>Return value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK</td>
</tr>
<tr>
<td>-1</td>
<td>Ignition is still switched on</td>
</tr>
</tbody>
</table>

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.

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

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

INFO

For more information on this subject, refer to JetSym online help.

### 7.6 Operating system update

Jetter AG are continuously striving to enhance the operating systems of their products. Enhancing means adding new features, upgrading existing functions and fixing bugs. Current OS files are available for download on our homepage in the `Downloads` area of the respective product.

More information on this subject is available on our website.

*Start | Jetter - We automate your success.*

Transfer options

There are several options to transfer the OS file to the device:

- With the programming tool JetSym
- Via JetEasyDownload
7.6.1 Updating the operating system via programming tool
The programming tool JetSym offers an easy way to transfer an OS file to the JVM-104-O08.

Performing 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 powered on during the operating system update.
1. Select in the JetSym menu Build menu item 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. To activate the newly installed OS, re-boot the device.

Performing the update via JetEasyDownload
To update the operating system of a JVM-104-O08 use a Peak CAN dongle and the command line tool JetEasyDownload (version 1.00.0.15 or higher) by Jetter. Alternatively, you can update the device from a controller by Jetter.

Performing 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.
7.7 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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Startup]</td>
<td>Section name</td>
</tr>
<tr>
<td>Project</td>
<td>Path to the application program file. This path is relative to \ app.</td>
</tr>
<tr>
<td>Program</td>
<td>Name of the application program file</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The operating system reads the file \App\start.ini from the internal flash disk.</td>
</tr>
<tr>
<td>2</td>
<td>The OS evaluates the Project entry. It contains the path leading to the application program file.</td>
</tr>
<tr>
<td>3</td>
<td>The OS evaluates the Program entry. This entry contains the program name.</td>
</tr>
<tr>
<td>4</td>
<td>The OS loads the application program from the file &lt;Project&gt;/&lt;Program&gt;.</td>
</tr>
</tbody>
</table>
8 Registers - Overview

This register overview describes the registers of the JVM-104-O08 in summarized form.

8.1 Default address on the CANopen® bus

Default address of the JVM-104-O08:

Node ID: 127 (0x7F)

8.2 General overview - Registers

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

8.3 I/Os - General overview

<table>
<thead>
<tr>
<th>Register areas</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>361000 ... 361007</td>
<td>Bit-coded map of input keys</td>
</tr>
</tbody>
</table>

8.4 Flags - General overview

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ... 255</td>
<td>Non-volatile application flags</td>
</tr>
<tr>
<td>256 ... 2047</td>
<td>Mapped to registers R 1000000 ... 1000055</td>
</tr>
<tr>
<td>2048 ... 2303</td>
<td>Special flags</td>
</tr>
</tbody>
</table>

8.5 Electronic nameplate

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

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100500 0</td>
<td>CPU</td>
</tr>
<tr>
<td>1</td>
<td>Baseboard</td>
</tr>
<tr>
<td>Identification:</td>
<td></td>
</tr>
<tr>
<td>100600</td>
<td>Internal version number</td>
</tr>
<tr>
<td>100601</td>
<td>Module ID</td>
</tr>
<tr>
<td>100602 ... 100612</td>
<td>Module name (register string)</td>
</tr>
<tr>
<td>100613</td>
<td>PCB revision</td>
</tr>
<tr>
<td>100614</td>
<td>PCB options</td>
</tr>
<tr>
<td>Production:</td>
<td></td>
</tr>
<tr>
<td>100700</td>
<td>Internal version number</td>
</tr>
<tr>
<td>100701 ... 100707</td>
<td>Serial number (register string)</td>
</tr>
<tr>
<td>100708</td>
<td>Day</td>
</tr>
<tr>
<td>100709</td>
<td>Month</td>
</tr>
<tr>
<td>100710</td>
<td>Year</td>
</tr>
<tr>
<td>Register ranges</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>100711</td>
<td>TestNum.</td>
</tr>
<tr>
<td>100712</td>
<td>TestRev.</td>
</tr>
</tbody>
</table>

**Features:**

- I/O module

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100800</td>
<td>Internal version number</td>
</tr>
<tr>
<td>100801</td>
<td>MAC address (Jetter)</td>
</tr>
<tr>
<td>100802</td>
<td>MAC address (device)</td>
</tr>
</tbody>
</table>

### 8.6 Electronic name plate (device as a whole)

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
</table>

#### Production:

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100900</td>
<td>Internal version number</td>
</tr>
<tr>
<td>100901 ... 100907</td>
<td>Serial number (register string)</td>
</tr>
<tr>
<td>100708</td>
<td>Day</td>
</tr>
<tr>
<td>100709</td>
<td>Month</td>
</tr>
<tr>
<td>100710</td>
<td>Year</td>
</tr>
</tbody>
</table>

#### Identification:

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100950</td>
<td>Internal version number</td>
</tr>
<tr>
<td>100951</td>
<td>Module ID</td>
</tr>
<tr>
<td>100952 ... 100962</td>
<td>Module name (register string)</td>
</tr>
<tr>
<td>100965</td>
<td>Config ID</td>
</tr>
<tr>
<td>100966</td>
<td>Vendor ID</td>
</tr>
<tr>
<td>100967</td>
<td>Variant ID</td>
</tr>
<tr>
<td>100968</td>
<td>Type ID</td>
</tr>
<tr>
<td>100992</td>
<td>Navision ID</td>
</tr>
<tr>
<td>100993</td>
<td>FBG version</td>
</tr>
</tbody>
</table>

### 8.7 CAN

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>106000</td>
<td>Baud rate CAN 1</td>
</tr>
<tr>
<td>106001</td>
<td>Node ID CAN 1</td>
</tr>
</tbody>
</table>

### 8.8 Flash memory

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>107510</td>
<td>Available sectors</td>
</tr>
<tr>
<td>107511</td>
<td>Used sectors</td>
</tr>
<tr>
<td>107512</td>
<td>Blocked sectors</td>
</tr>
<tr>
<td>107513</td>
<td>Free sectors</td>
</tr>
<tr>
<td>107520</td>
<td>Size of the flash disk in bytes</td>
</tr>
<tr>
<td>107521</td>
<td>Used memory in bytes</td>
</tr>
<tr>
<td>107522</td>
<td>Blocked memory in bytes</td>
</tr>
<tr>
<td>107523</td>
<td>Free memory in bytes</td>
</tr>
</tbody>
</table>
8.9 System information

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

8.10 General system registers

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200000</td>
<td>OS version (major * 100 + minor)</td>
</tr>
<tr>
<td>200001</td>
<td>Bit 0 = 1 Application program is running</td>
</tr>
<tr>
<td>200008</td>
<td>Error register identical with 210004</td>
</tr>
<tr>
<td>Bit 8</td>
<td>Illegal jump</td>
</tr>
<tr>
<td>Bit 9</td>
<td>Illegal call</td>
</tr>
<tr>
<td>Bit 10</td>
<td>Illegal index</td>
</tr>
<tr>
<td>Bit 11</td>
<td>Illegal opcode</td>
</tr>
<tr>
<td>Bit 12</td>
<td>Division by 0</td>
</tr>
<tr>
<td>Bit 13</td>
<td>Stack by 0</td>
</tr>
<tr>
<td>Bit 14</td>
<td>Stack underflow</td>
</tr>
<tr>
<td>Bit 15</td>
<td>Illegal stack</td>
</tr>
<tr>
<td>Bit 16</td>
<td>Failed to load application program</td>
</tr>
<tr>
<td>Bit 24</td>
<td>Timeout - Cycle time</td>
</tr>
<tr>
<td>Bit 25</td>
<td>Timeout - Task lock</td>
</tr>
<tr>
<td>Bit 31</td>
<td>Unknown error</td>
</tr>
<tr>
<td>200168</td>
<td>Bootloader version (IP format)</td>
</tr>
<tr>
<td>200169</td>
<td>OS version (IP format)</td>
</tr>
<tr>
<td>201000</td>
<td>Runtime registers in milliseconds</td>
</tr>
<tr>
<td>201001</td>
<td>Runtime register in seconds</td>
</tr>
<tr>
<td>201002</td>
<td>Runtime register in R 201003</td>
</tr>
<tr>
<td>201003</td>
<td>10 ms units for R 201002</td>
</tr>
<tr>
<td>201004</td>
<td>Runtime registers in milliseconds</td>
</tr>
<tr>
<td>201005</td>
<td>Runtime registers in microseconds</td>
</tr>
<tr>
<td>202960</td>
<td>Password for system command register:</td>
</tr>
<tr>
<td>202961</td>
<td>System command register</td>
</tr>
<tr>
<td>202980</td>
<td>Error history: Number of entries</td>
</tr>
<tr>
<td>Register ranges</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>202981</td>
<td>Error history: Index</td>
</tr>
<tr>
<td>202982</td>
<td>Error history: Entry</td>
</tr>
<tr>
<td>203100 ... 203107</td>
<td>32-bit overlaying Flags 0 ... 255</td>
</tr>
<tr>
<td>203108 ... 203123</td>
<td>16-bit overlaying Flags 0 ... 255</td>
</tr>
<tr>
<td>203124 ... 203131</td>
<td>32-bit overlaying Flags 2048 ... 2303</td>
</tr>
<tr>
<td>203132 ... 203147</td>
<td>16-bit overlaying Flags 2048 ... 2303</td>
</tr>
<tr>
<td>209700</td>
<td>System logger: Global enable</td>
</tr>
<tr>
<td>209701 ... 209739</td>
<td>Enabling system components</td>
</tr>
</tbody>
</table>

### 8.11 Application program

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>210000</td>
<td>State of application program</td>
</tr>
<tr>
<td>Bit 0 = 1</td>
<td>Application program is running</td>
</tr>
<tr>
<td>210001</td>
<td>JetVM version</td>
</tr>
<tr>
<td>210004</td>
<td>Error register</td>
</tr>
<tr>
<td>Bit 8</td>
<td>Illegal jump</td>
</tr>
<tr>
<td>Bit 9</td>
<td>Illegal call</td>
</tr>
<tr>
<td>Bit 10</td>
<td>Illegal index</td>
</tr>
<tr>
<td>Bit 11</td>
<td>Illegal opcode</td>
</tr>
<tr>
<td>Bit 12</td>
<td>Division by 0</td>
</tr>
<tr>
<td>Bit 13</td>
<td>Stack overflow</td>
</tr>
<tr>
<td>Bit 14</td>
<td>Stack underflow</td>
</tr>
<tr>
<td>Bit 15</td>
<td>Illegal stack</td>
</tr>
<tr>
<td>Bit 16</td>
<td>Error when loading the application program</td>
</tr>
<tr>
<td>Bit 24</td>
<td>Timeout - Cycle time</td>
</tr>
<tr>
<td>Bit 25</td>
<td>Timeout - Task lock</td>
</tr>
<tr>
<td>Bit 31</td>
<td>Unknown error</td>
</tr>
<tr>
<td>210006</td>
<td>Highest task number</td>
</tr>
<tr>
<td>210007</td>
<td>Minimum program cycle time</td>
</tr>
<tr>
<td>210008</td>
<td>Maximum program cycle time</td>
</tr>
<tr>
<td>210009</td>
<td>Current program cycle time</td>
</tr>
<tr>
<td>210011</td>
<td>Current task number</td>
</tr>
<tr>
<td>210050</td>
<td>Current program position within an execution unit</td>
</tr>
<tr>
<td>210051</td>
<td>ID of the execution unit being processed</td>
</tr>
<tr>
<td>210056</td>
<td>Desired total cycle time in µs</td>
</tr>
<tr>
<td>210057</td>
<td>Calculated total cycle time in µs</td>
</tr>
<tr>
<td>210058</td>
<td>Maximum time slice per task in µs</td>
</tr>
<tr>
<td>210060</td>
<td>Task ID (for R210061)</td>
</tr>
<tr>
<td>210061</td>
<td>Priority for task [R210060]</td>
</tr>
<tr>
<td>210063</td>
<td>Length of scheduler table</td>
</tr>
<tr>
<td>210064</td>
<td>Index in scheduler table</td>
</tr>
<tr>
<td>210065</td>
<td>Task ID in scheduler table</td>
</tr>
<tr>
<td>210070</td>
<td>Task ID (for R210071)</td>
</tr>
<tr>
<td>210071</td>
<td>Timer number (0 ... 31)</td>
</tr>
<tr>
<td>210072</td>
<td>Manual triggering of a timer event (bit-coded)</td>
</tr>
<tr>
<td>210073</td>
<td>End of cyclic task (task ID)</td>
</tr>
<tr>
<td>210074</td>
<td>Command for cyclic tasks</td>
</tr>
</tbody>
</table>
## Register ranges

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>210075</td>
<td>Number of timers</td>
</tr>
<tr>
<td>210076</td>
<td>Timer number (for R210077)</td>
</tr>
<tr>
<td>210077</td>
<td>Timer value in milliseconds</td>
</tr>
<tr>
<td>210100 ... 210199</td>
<td>Task state</td>
</tr>
<tr>
<td>210400 ... 210499</td>
<td>Task - Program address</td>
</tr>
<tr>
<td>210600</td>
<td>Task ID of a cyclical task (for R210601)</td>
</tr>
<tr>
<td>210601</td>
<td>Processing time of a cyclical task in per mil figure</td>
</tr>
<tr>
<td>210609</td>
<td>Tasklock timeout in ms</td>
</tr>
<tr>
<td>210610</td>
<td>Tasklock timeout in ms (bit-coded) (-1 Monitoring disabled)</td>
</tr>
<tr>
<td>Bit: 0</td>
<td>Timeout</td>
</tr>
<tr>
<td>-&gt; Timer 0</td>
<td></td>
</tr>
</tbody>
</table>

### 8.12 File system/data file function

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>312977</td>
<td>Status of file operation</td>
</tr>
<tr>
<td>312978</td>
<td>Task ID</td>
</tr>
</tbody>
</table>

### 8.13 Application registers

<table>
<thead>
<tr>
<th>Register ranges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000 .. 1005999</td>
<td>32-bit integer (remanent)</td>
</tr>
</tbody>
</table>

### 8.14 Display

#### Input keys

<table>
<thead>
<tr>
<th>Input keys</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>361000 ... 361007</td>
<td>Bit-coded map of the keys (e.g. bit 0 =1 → key 1 pressed)</td>
</tr>
<tr>
<td>361000.0</td>
<td>[F1]</td>
</tr>
<tr>
<td>361000.1</td>
<td>[F2]</td>
</tr>
<tr>
<td>361000.2</td>
<td>[F3]</td>
</tr>
<tr>
<td>361000.3</td>
<td>[F4]</td>
</tr>
<tr>
<td>361000.4</td>
<td>[POWER]</td>
</tr>
<tr>
<td>361000.5</td>
<td>[SCROLL]</td>
</tr>
<tr>
<td>361000.6</td>
<td>[ESC]</td>
</tr>
<tr>
<td>361000.7</td>
<td>[HOME]</td>
</tr>
</tbody>
</table>

#### Ignition (IGN)

<table>
<thead>
<tr>
<th>Ignition (IGN)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>361100</td>
<td>Bit 0 = 0 The ignition is ON.</td>
</tr>
<tr>
<td>Bit 0 = 1</td>
<td>Ignition is OFF</td>
</tr>
</tbody>
</table>

#### Digipot

<table>
<thead>
<tr>
<th>Digipot</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>363000</td>
<td>Present count value</td>
</tr>
<tr>
<td>363001</td>
<td>Digipot key</td>
</tr>
<tr>
<td>363002</td>
<td>Minimum count value</td>
</tr>
<tr>
<td>363003</td>
<td>Maximum count value</td>
</tr>
</tbody>
</table>

#### Illumination

<table>
<thead>
<tr>
<th>Illumination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>64000</td>
<td>Background lighting</td>
</tr>
<tr>
<td>364001</td>
<td>Night-lighting of keys</td>
</tr>
</tbody>
</table>
### Register areas

<table>
<thead>
<tr>
<th>Visualization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>365100</td>
<td>Language selection according to ID</td>
</tr>
</tbody>
</table>

## 8.15 Flag

### 32 combined flags

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>203100</td>
<td>0 ... 31</td>
</tr>
<tr>
<td>203101</td>
<td>32 ... 63</td>
</tr>
<tr>
<td>203102</td>
<td>64 ... 95</td>
</tr>
<tr>
<td>203103</td>
<td>96 ... 127</td>
</tr>
</tbody>
</table>

### 16 combined flags

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>203108</td>
<td>0 ... 15</td>
</tr>
<tr>
<td>203109</td>
<td>16 ... 31</td>
</tr>
<tr>
<td>203110</td>
<td>32 ... 47</td>
</tr>
<tr>
<td>203111</td>
<td>48 ... 63</td>
</tr>
<tr>
<td>203112</td>
<td>64 ... 79</td>
</tr>
<tr>
<td>203113</td>
<td>80 ... 95</td>
</tr>
<tr>
<td>203114</td>
<td>96 ... 111</td>
</tr>
<tr>
<td>203115</td>
<td>112 ... 127</td>
</tr>
</tbody>
</table>

### 16 combined special flags

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>203132</td>
<td>2048 ... 2063</td>
</tr>
<tr>
<td>203133</td>
<td>2064 ... 2079</td>
</tr>
<tr>
<td>203134</td>
<td>2080 ... 2095</td>
</tr>
<tr>
<td>203135</td>
<td>2096 ... 2111</td>
</tr>
<tr>
<td>203136</td>
<td>2112 ... 2127</td>
</tr>
<tr>
<td>203137</td>
<td>2128 ... 2143</td>
</tr>
<tr>
<td>203138</td>
<td>2144 ... 2159</td>
</tr>
<tr>
<td>203139</td>
<td>2160 ... 2175</td>
</tr>
</tbody>
</table>

### 32 combined special flags

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>203124</td>
<td>2048 ... 2079</td>
</tr>
<tr>
<td>203125</td>
<td>2080 ... 2111</td>
</tr>
<tr>
<td>203126</td>
<td>2112 ... 2143</td>
</tr>
<tr>
<td>203127</td>
<td>2144 ... 2175</td>
</tr>
</tbody>
</table>

### Application registers with overlaid flags

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000</td>
<td>256 ... 287</td>
</tr>
<tr>
<td>1000001</td>
<td>288 ... 319</td>
</tr>
<tr>
<td>1000002</td>
<td>320 ... 351</td>
</tr>
<tr>
<td>1000003</td>
<td>352 ... 383</td>
</tr>
<tr>
<td>1000004</td>
<td>384 ... 415</td>
</tr>
<tr>
<td>1000005</td>
<td>416 ... 447</td>
</tr>
<tr>
<td>1000006</td>
<td>448 ... 479</td>
</tr>
<tr>
<td>1000007</td>
<td>480 ... 511</td>
</tr>
</tbody>
</table>
### Application registers with overlaid flags

<table>
<thead>
<tr>
<th>Register</th>
<th>Application</th>
<th>Overlaid Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000008</td>
<td>512 ... 543</td>
<td>1000036 1408 ... 1439</td>
</tr>
<tr>
<td>1000009</td>
<td>544 ... 575</td>
<td>1000037 1440 ... 1471</td>
</tr>
<tr>
<td>1000010</td>
<td>576 ... 607</td>
<td>1000038 1472 ... 1503</td>
</tr>
<tr>
<td>1000011</td>
<td>608 ... 639</td>
<td>1000039 1504 ... 1535</td>
</tr>
<tr>
<td>1000012</td>
<td>640 ... 671</td>
<td>1000040 1536 ... 1567</td>
</tr>
<tr>
<td>1000013</td>
<td>672 ... 703</td>
<td>1000041 1568 ... 1599</td>
</tr>
<tr>
<td>1000014</td>
<td>704 ... 735</td>
<td>1000042 1600 ... 1631</td>
</tr>
<tr>
<td>1000015</td>
<td>736 ... 767</td>
<td>1000043 1632 ... 1663</td>
</tr>
<tr>
<td>1000016</td>
<td>768 ... 799</td>
<td>1000044 1664 ... 1695</td>
</tr>
<tr>
<td>1000017</td>
<td>800 ... 831</td>
<td>1000045 1696 ... 1727</td>
</tr>
<tr>
<td>1000018</td>
<td>832 ... 863</td>
<td>1000046 1728 ... 1759</td>
</tr>
<tr>
<td>1000019</td>
<td>864 ... 895</td>
<td>1000047 1760 ... 1791</td>
</tr>
<tr>
<td>1000020</td>
<td>896 ... 927</td>
<td>1000048 1792 ... 1823</td>
</tr>
<tr>
<td>1000021</td>
<td>928 ... 959</td>
<td>1000049 1824 ... 1855</td>
</tr>
<tr>
<td>1000022</td>
<td>960 ... 991</td>
<td>1000050 1856 ... 1887</td>
</tr>
<tr>
<td>1000023</td>
<td>992 ... 1023</td>
<td>1000051 1888 ... 1919</td>
</tr>
<tr>
<td>1000024</td>
<td>1024 ... 1055</td>
<td>1000052 1920 ... 1951</td>
</tr>
<tr>
<td>1000025</td>
<td>1056 ... 1087</td>
<td>1000053 1952 ... 1983</td>
</tr>
<tr>
<td>1000026</td>
<td>1088 ... 1119</td>
<td>1000054 1984 ... 2015</td>
</tr>
<tr>
<td>1000027</td>
<td>1120 ... 1151</td>
<td>1000055 2016 ... 2047</td>
</tr>
</tbody>
</table>

### 8.16 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 functions

<table>
<thead>
<tr>
<th>System functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Converting BCD to HEX</td>
</tr>
<tr>
<td>5</td>
<td>Converting HEX to BCD</td>
</tr>
<tr>
<td>20</td>
<td>Square root</td>
</tr>
<tr>
<td>21</td>
<td>Sine</td>
</tr>
<tr>
<td>22</td>
<td>Cosine</td>
</tr>
<tr>
<td>23</td>
<td>Tangent</td>
</tr>
<tr>
<td>24</td>
<td>Arc sine</td>
</tr>
<tr>
<td>25</td>
<td>Arc cosine</td>
</tr>
<tr>
<td>26</td>
<td>Arc tangent</td>
</tr>
<tr>
<td>27</td>
<td>Exponential function</td>
</tr>
<tr>
<td>28</td>
<td>Natural logarithm</td>
</tr>
<tr>
<td>29</td>
<td>Absolute value</td>
</tr>
<tr>
<td>30</td>
<td>Separation of digits before and after the decimal point</td>
</tr>
<tr>
<td>50</td>
<td>Sorting register values</td>
</tr>
<tr>
<td>90</td>
<td>Writing a data file</td>
</tr>
<tr>
<td>91</td>
<td>Appending a data file</td>
</tr>
<tr>
<td>92</td>
<td>Reading a data file</td>
</tr>
<tr>
<td>96</td>
<td>Deleting a data file</td>
</tr>
</tbody>
</table>

#### System functions

<table>
<thead>
<tr>
<th>System functions</th>
<th>Corresponding JetSym STX function</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Function Bcd2Hex(Bcd: int): Int;</td>
</tr>
<tr>
<td>5</td>
<td>Function Hex2Bcd(Hex: int): Int;</td>
</tr>
<tr>
<td>System functions</td>
<td>Corresponding JetSym STX function</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>90/91</td>
<td>Function FileDAWrite(Const Ref FileName: String, Const Ref Mode: String, VarType: DWRITE_TYPE, First: Int, Last: int): Int;</td>
</tr>
<tr>
<td>92</td>
<td>Function FileDARead(Const Ref FileName: String): Int;</td>
</tr>
</tbody>
</table>
9 Maintenance and repairs

9.1 Maintenance, repairs and disposal

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

Repairs
Defective components could cause dangerous malfunctions and could compromise safety. Only the manufacturer is allowed to repair the device. Do not open the device!

Disposal of obsolete equipment
The device must be disposed of in accordance with the Environmental Product Declaration EPD. Applicable local environmental directives and regulations must be complied with. This product must be disposed of as waste electronic equipment. Waste packaging material must be recycled or reused.

Modifications and alterations 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.

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

10.1 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 number, please refer to the nameplate.
- OS version
  - To determine the operating system version, use the functions of the development environment.
11 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.

11.1 Accessories

**INFO**

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

<table>
<thead>
<tr>
<th>Accessories</th>
<th>Item number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mounting plate for RAM Mount ball consisting of mounting plate and screws for housing with Deutsch or M12 connector, without RAM Mount attachments</td>
<td>10001621</td>
</tr>
<tr>
<td>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</td>
<td>10001551</td>
</tr>
<tr>
<td>Connecting cable Length 5 m, pre-assembled on one end with M12x1 connector, 8-pole, opposite end open</td>
<td>60882261</td>
</tr>
<tr>
<td>Interconnecting programming cable M12 on both ends, pre-assembled and tested, CAN1 brought out to Sub-D</td>
<td>60882086</td>
</tr>
</tbody>
</table>

Tab. 23: Accessories
List of figures

Fig. 1  Device configuration ........................................................................................................ 8
Fig. 2  Nameplate ...................................................................................................................... 9
Fig. 3  Dimensions (in mm) ....................................................................................................... 10
Fig. 4  Space requirements for installation work (in mm) ........................................................ 14
Fig. 5  All dimensions of screw holes are in millimeters. ......................................................... 15
Fig. 6  Installation drawing ....................................................................................................... 16
Fig. 7  Installing the strain relief .............................................................................................. 17
Fig. 8  M12 connector, 8 pins ................................................................................................ 19
**List of tables**

Tab. 1  Power supply VBAT_ECU .................................................................................. 10
Tab. 2  Mechanical specifications .............................................................................. 10
Tab. 3  Environmental conditions ............................................................................. 11
Tab. 4  Technical data - Display ................................................................................ 11
Tab. 5  Acoustic signal generator ............................................................................. 11
Tab. 6  CAN interface specification .......................................................................... 11
Tab. 7  Specification - CAN bus cable ....................................................................... 12
Tab. 8  CAN interface specification .......................................................................... 12
Tab. 9  Line lengths .................................................................................................... 12
Tab. 10  Pulses to ISO 7637-2 .................................................................................. 12
Tab. 11  Irradiation to ISO 11452 ........................................................................... 12
Tab. 12  ESD EN 61000-4-2 ..................................................................................... 13
Tab. 13  Unsuitable installation locations ................................................................ 14
Tab. 14  Mounting accessories .................................................................................. 15
Tab. 15  Compatible M12 connector (female) ............................................................... 19
Tab. 16  Abbreviations .............................................................................................. 20
Tab. 17  Module register properties .......................................................................... 20
Tab. 18  Numerical formats ....................................................................................... 20
Tab. 19  JetSym sample programs ............................................................................ 20
Tab. 20  Flash disk capacity ...................................................................................... 24
Tab. 21  Input key registers ....................................................................................... 25
Tab. 22  Virtual key codes ......................................................................................... 26
Tab. 23  Accessories .................................................................................................. 41