BUCHER
automation

User Manual

60887527_01

JetViewMobile 205

ISOBUS terminal

Bucher Automation AG has created this document with the requisite care and based on the current state of technology. Changes and further technical developments to our products are not automatically made available in a revised document. Bucher Automation AG shall accept no liability or responsibility for errors of content or form, missing updates or any damage or disadvantages arising therefrom.

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

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

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

www.bucherautomation.com

Translation of the german original User Manual

Revision	2.00
Date of issue	2/13/2024

Table of contents

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.

Purpose of the Document

This document describes rules for the design of the system menu in ISOBUS operating devices of Bucher Automation AG.

It does not describe all the functions and operation of the virtual ISOBUS terminal.

Target Groups

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

Availability of Information

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

Start | Bucher Automation - We automate your success.

For further information refer to the following information products:

- Application-oriented manuals
  Product-independent documentation
- Version updates
  Information about changes to the software products and operating system of your device

1.2 Typographical conventions

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

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

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

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

ⓘ Info

Further information and practical tips

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

1.3 Legal notices

The firmware and tools used by the device use software products or components of the third-party providers named below.

Licenses of third-party providers

Product	Licensor	Link to the license document	License type
FSF lib	Free Software Foundation	https://www.fsf.org/licensing	GPLv3
CMSYS	ARM Limited	ARM contract reference LEC-PRE-00489n-V3.0	Open Source
ST lib	ST Microelectronics	http://www.st.com/software_license_agreement_liberty_v2	Free for controllers belonging to ST
Python	Python Software Foundation (PSF)	https://docs.python.org/3/license.html	GPL-compatible
srec_cat	Peter Miller	http://srecord.sourceforge.net	GNU GPLv3

Tab. 1: Licenses of third-party providers

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.

E1 type approval

The device has an E1 approval according to ECE R10 Rev. 5.

RoHS 2

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

2.2 Purpose

2.2.1 Intended use

The JVM-205 device is intended for operation of ISOBUS applications for mobile work machines.

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

2.2.2 Usage other than intended

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

Machinery Directive

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

2.3 Warnings used in this document

⚠ DANGER

High risk

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

⚠ WARNING

Medium risk

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

⚠ CAUTION

Low risk

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

NOTICE

Material damage

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

3 Product Description

The JVM-205 control unit is an ISOBUS Universal Terminal that features intuitive operation. Due to the ISOBUS-UT functionality, it works together with ISOBUS controllers. Communication is via the ISOBUS protocol.

Supported firmware

This document refers to firmware version 2.0.0.84 and higher.

3.1 Design

Front side

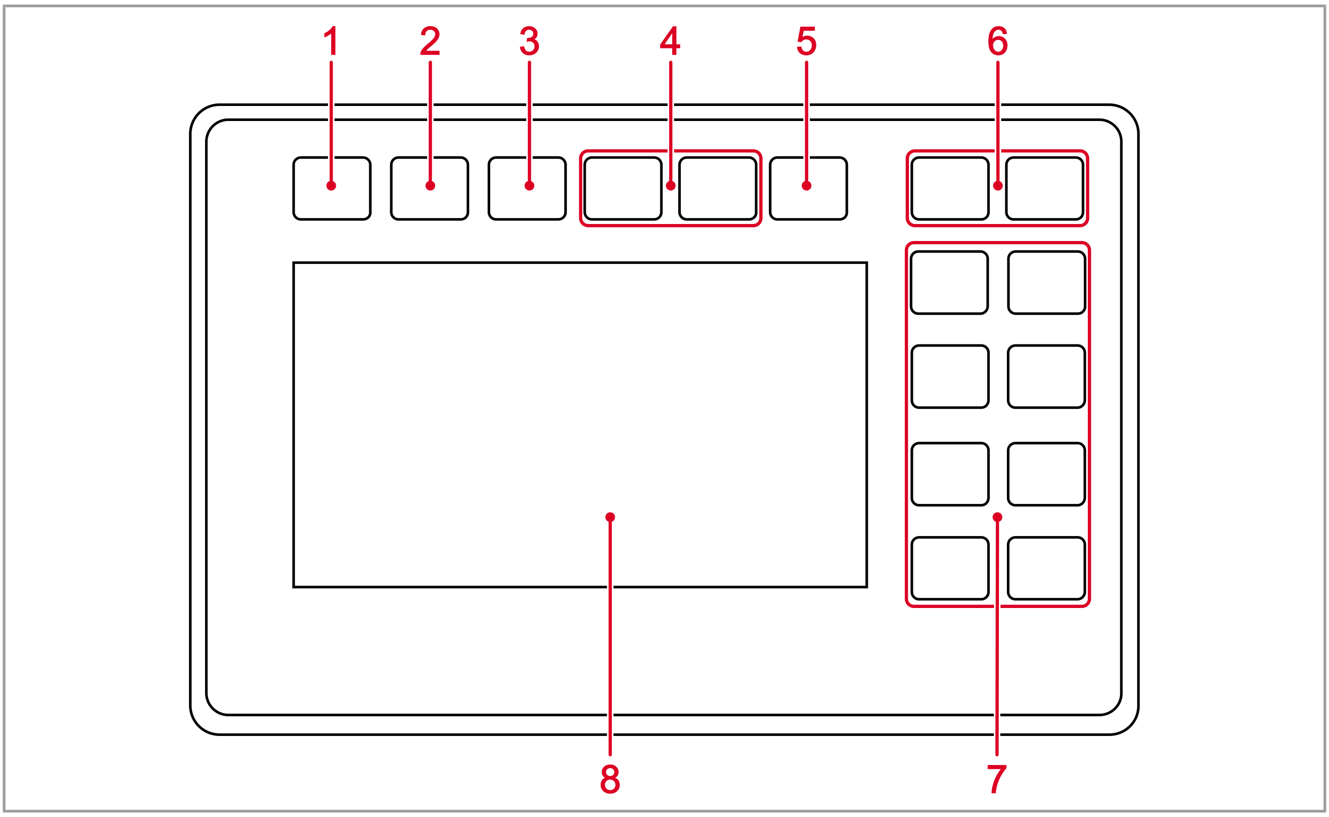


Fig. 1: Front side of the control unit

1	On/Off key
2	ISB key
3	Home key
4	Navigation keys
5	OK key
6	Save key
7	Function keys
8	TFT touchscreen

Rear

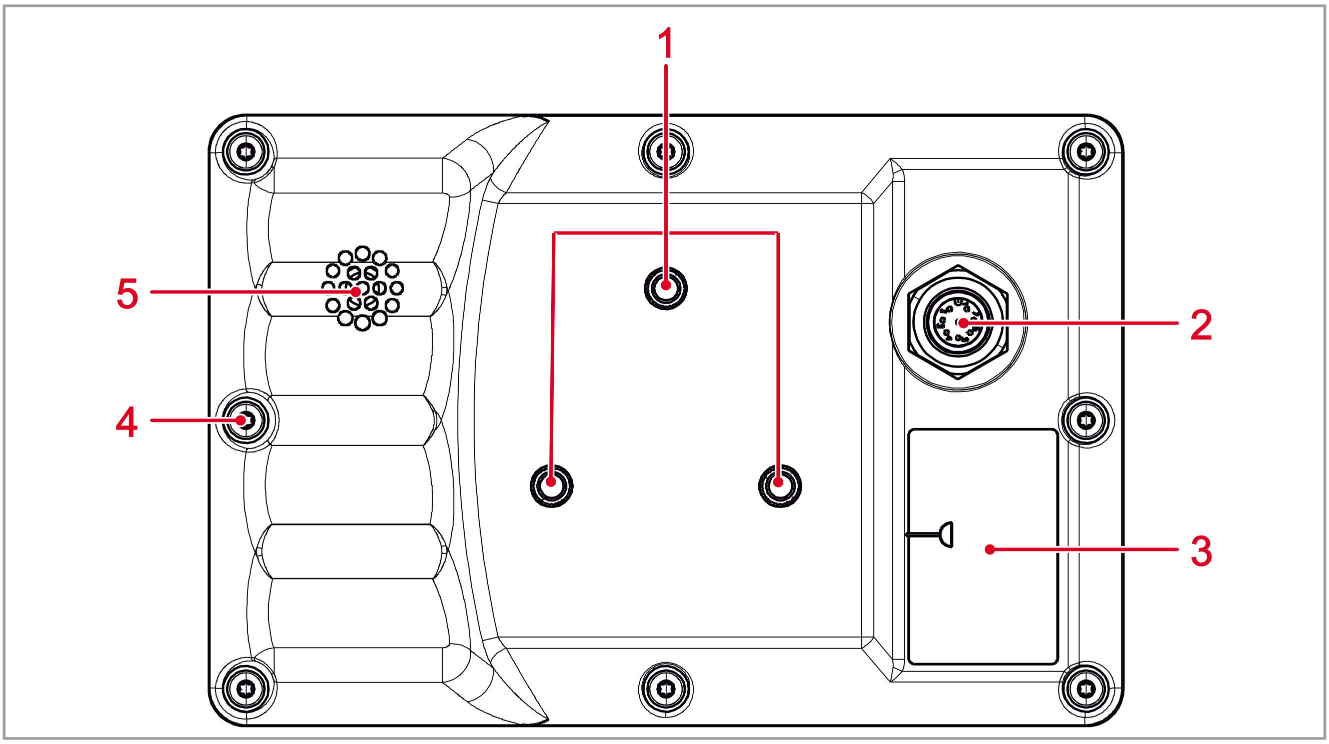


Fig. 2: Rear of the control unit

1	Female thread for RAM mount holder
2	M12 plug, 8-pin
3	Nameplate
4	Housing screw connection
5	Loudspeaker

3.2 Product Features

- Display: 4.3" touchscreen, high-resolution
- Input: 11 user-programmable function keys (soft keys F1 … F8, ISB; M1, M2);
  6 permanently assigned hard keys (On/Off, Home, ESC, Navigation, OK)
- 1 digital input for the ignition signal
- Real-time clock (RTC) with battery backup (optional)
- ISOBUS-UT function
- 16-MB flash memory
- Update of the operating system and system menu via USB
- Adaptable system menu

3.3 Nameplate

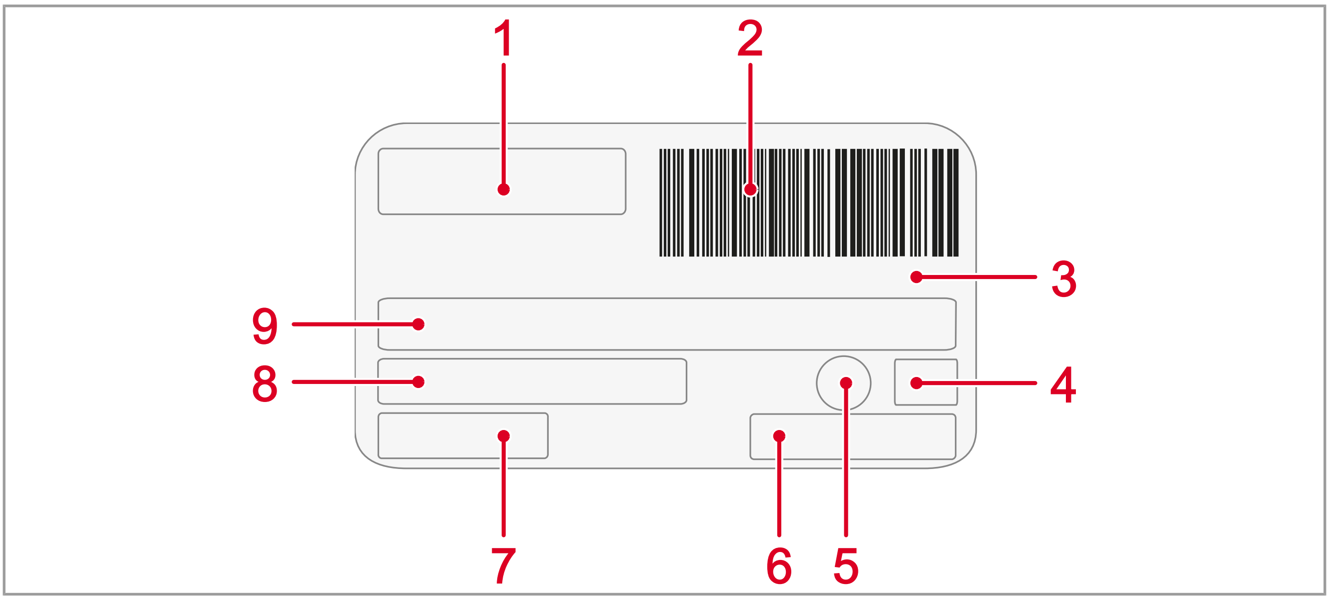


Fig. 3: Nameplate

1	Manufacturer logo
2	Barcode
3	Serial number
4	CE certification mark
5	E1 approval
6	E1 approval number
7	Hardware revision
8	Part number
9	Product type

3.4 Scope of Delivery

Scope of delivery	Item number	Quantity
JVM-205-K00-O31	10002136	1

4 Technical Specifications

This chapter contains information on both electrical and mechanical data as well as on operating data of the JVM-205 device.

4.1 Dimensions

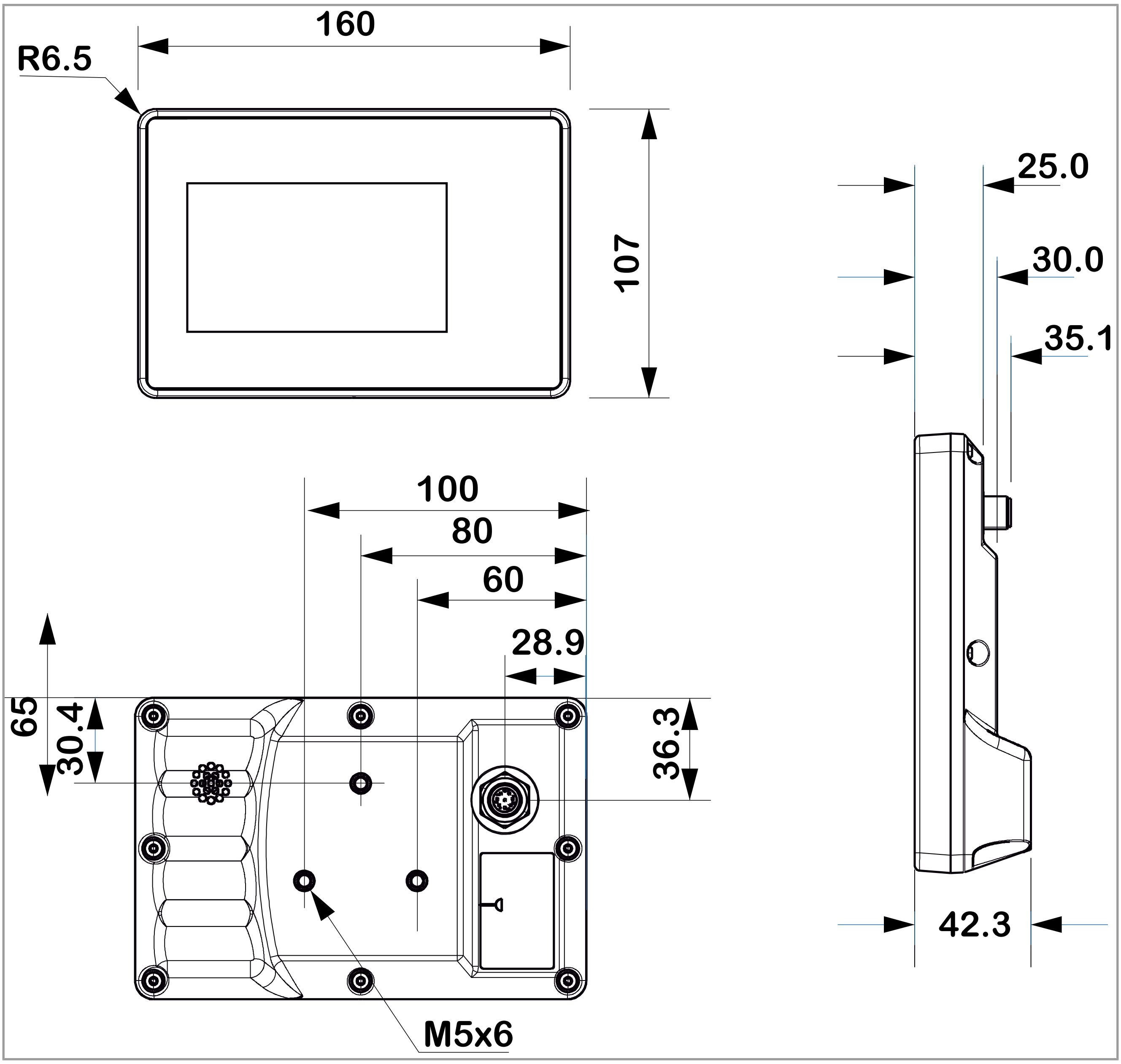


Fig. 4: Dimension in mm

4.2 Display

Parameter	Description
Type	TFT screen with touch function
Resolution	480 x 272 pixels
Screen diagonal	4.3"
Background lighting	LED, typ. 600 cd/m^2 , dimmable

Tab. 2: Technical data – display

4.3 Pushbuttons

Parameter	Description
Number of membrane keys	16 keys
Background lighting	Dimmable
	Adjustable between 0% … 100%
Typical service life	1,000,000 activations

Tab. 3: Technical data – keys

4.4 Processor Kernel

Parameter	Description
CPU	STM32H7
FLASH	16 MB

Tab. 4: Technical data – computer core

4.5 Mechanical specifications

Parameter	Description	Standards
Weight	800 g	
Enclosure specifications		
Material	Plastic	
Vibration strength	10 Hz … 150 Hz, 6 h	ISO 16750-3
Shock resistance		
Type of shock	Half-sine wave	ISO 16750-3
Intensity and duration	50 g for 18 ms	
Number and direction	10 shocks in all 3 directions of the spatial axes	

Tab. 5: Technical data – mechanical properties

4.6 Environmental conditions

Parameter	Description	Standards
Operating temperature	-20 °C … +70 °C	ISO 16750-4
Climatic conditions	Humid heat	
Storage temperature	-30 °C … +70 °C	ISO 16750-4 DIN EN 60068-2-1 DIN EN 60068-2-2
Relative humidity	5% … 95%	
Degree of protection	With mating plug: IP65; Without mating plug: IP20	

Tab. 6: Technical data – environmental conditions

4.7 Electrical properties

Power supply T_ECU

Parameter	Description
Nominal voltage	DC 8.5 V … 18 V
Permissible voltage range VBAT_ECU	DC 8 V ... 18 V
Maximum current consumption	2 A
Typical logic current consumption (VBAT_ECU)	320 mA at DC 12 V
Power consumption	Approx 3 W
Integrated protective functions	Reverse polarity protection, overvoltage, short voltage pulses

Tab. 7: Technical data – power supply VBAT_ECU

4.8 EMC values

The JVM-205 has CE approval in accordance with EN ISO 14982 for 12 V on-board systems.

Pulses in accordance with ISO 7637-2

Test pulse	Values	Function class
1	-450 V	C
2a	+37 V	A
2b	+20 V	C
3a	-150 V	A
3b	+150 V	A
4	Ua1: -12 V / 50 ms Ua2: -5 V / 500 ms	B (24-V systems)

Tab. 8: Pulses in accordance with ISO 7637-2

Pulses in accordance with ISO 16750-2 for 12 V on-board systems

Test pulse	Values	Function class
5b	Load Dump 70 V / 2 Ω / 350 ms	C

Tab. 9: Pulses in accordance with ISO 16750-2 for 12 V on-board systems

Irradiation in accordance with ISO 11452

Irradiation	Function class
20 MHz ... 2 GHz 30 V/m	A
20 MHz ... 2 GHz 100 V/m	B

Tab. 10: Irradiation in accordance with ISO 11452

Emitted radiation acc. to CISPR 25

Emitted radiation	Limit values
Narrowband emission 30 MHz ... 1,000 MHz	30 MHz … 75 MHz = 52 … 42 dBµV/m (logarithmically decreasing) 75 MHz … 400 MHz = 42 … 53 dBµV/m (logarithmically increasing) 400 MHz … 1000 MHz = 53 dBµV/m (constant)
Wideband emission 30 MHz ... 1,000 MHz	30 MHz … 75 MHz = 62 … 52 dBµV/m (logarithmically decreasing) 75 MHz … 400 MHz = 52 … 63 dBµV/m (logarithmically increasing) 400 MHz … 1000 MHz = 63d BµV/m (constant)

Tab. 11: Emitted radiation acc. to CISPR 25

ESD acc. to EN 61000-4-2

Electrostatic discharge (ESD)	Function class
Contact ±4 kV	A
Air ±8 kV	A

Tab. 12: Electrostatic discharge (ESD) acc. to EN 61000-4-2

4.9 Ports and interfaces

4.9.1 ISOBUS interface

Parameter	Description
Baud rate	250 kBaud
Protocol	ISOBUS
Terminating resistance	Integrated
Cable specification	CAN-compliant, twisted wire pairs, unshielded

Tab. 13: Technical data – ISOBUS interface

4.9.2 USB interface

Parameter	Description
Supply unit	5 V, max. 500 mA
Data rate	USB 1.1
Protocol	USB-Host-Interface

Tab. 14: Technical data – USB interface

4.10 Inputs/outputs according to ISO11786:1995

Within the operating voltage range, all inputs are voltage-proof and overcurrent protected.

Input/output for ignition

Parameter	Description
Ignition input/output	
Abbreviation	IGN_KEY
Quantity	1
Input	
Power supply	12 V
Output	
Max. output current	0.5 A

Tab. 15: Input/output for ignition

4.11 Acoustic signal generator

Category	Description	
Type	Piezo	
Volume	> 70 dB	With resonance frequency and without mechanical damping.
Volume control	Turned off, 10% … 100%	Controlled via PWM.
Frequency range	200 Hz … 4 kHz	

Tab. 16: Technical data – acoustic signal generator

4.12 Real-time clock

Optional equipment.

Parameter	Description	
Battery type	CR2032	
Service life	Approx. 10 years from production.	The service life of the battery depends on the ambient conditions and may therefore differ.

Tab. 17: Technical data – real-time clock

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.

NOTICE

Compliance with degree of protection

The protection class of the device is only ensured if the M12 cable gland is tightened securely.

5.1 Allowed mounting orientations

The device has a RAM mount connection and can be installed in the driver's cab with a suitable holder.

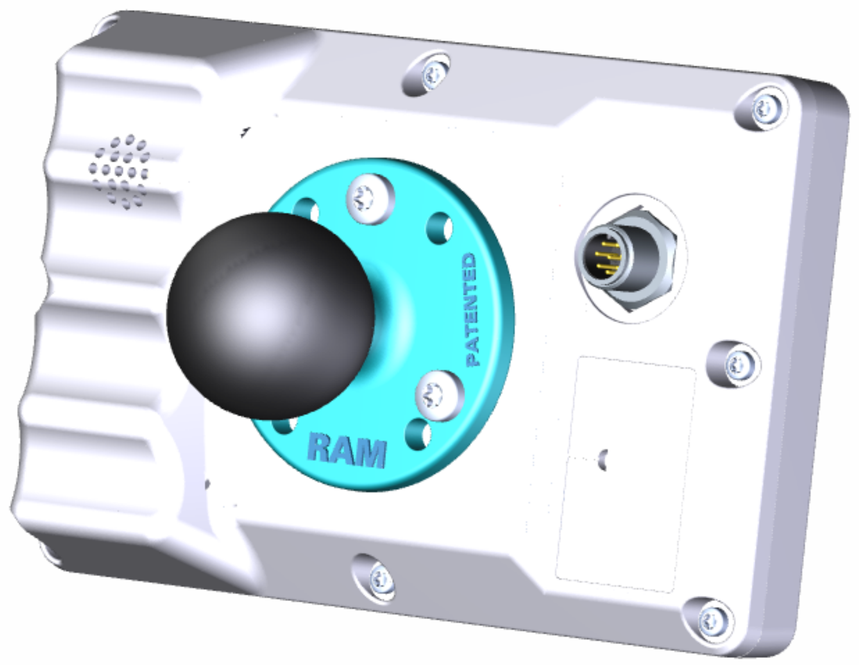


Fig. 5: Permitted installation position with RAM mount holder

6 Electrical connection

⚠ WARNING

Signal interference due to faulty CAN wiring

Unshielded or untwisted CAN lines can result in communication disruptions. In extreme cases a malfunction of the device may result in personal injuries.

- ► Connect 120 Ω terminating resistors to both ends of the CAN bus.

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 8-pin M12 plug

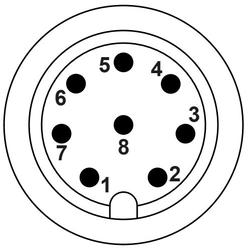


Fig. 6: M12 male connector

1	Voltage supply term.30 (+12 V)
2	USB data transfer +
3	Ignition input/output (max. 0.5 A)
4	USB data transfer -
5	CAN-L
6	Voltage supply term. 31 (GND)
7	CAN-H
8	USB +5 V (USB_VBUS)

Tab. 18: Pin assignment of the M12 plug

ⓘ **Info** To use the USB signal, the use of a cable T-piece (Accessories [▶ 41]) is recommended.

7 Firmware and system menu

This section describes how to load the firmware and system menu into the control unit. The following options are available:

- Via USB
- Via CAN bus

Both require special Accessories [▶ 41].

7.1 Creating an EDC file

The EDC file contains all the information needed to update the firmware, including the system menu.

The EDC file must be generated. Follow these steps:

1. Unpack the corresponding firmware package. This package is provided for you by Bucher-Automation support (support@bucherautomation.com).
2. Run script *edc_make.bat*.

⇨ 3 files are generated, one for each of the available interfaces:

- *41434445.EDC*: Updating the firmware and system menu via USB
- *41434445_EDC_x_x_x_x_.hex*: Updating the firmware and system menu via CAN bus
- *41434445 _Emergency.hex*: Update firmware and system menu via RS232 (not supported).

7.2 Description of the IOP file of the system menu

The system menu is written by an IOP file that is usually created using ISO-Designer software from Bucher-Automation.

Running script *edc_make.bat*, which is included in the firmware package, links the IOP file with the files of the firmware in an EDC file and transfers them together to the control unit (Creating an EDC file).

Structure of EDC file – example

```
4544-4334;1.6.0.48;41434445; output file (EUID is hardware-dependent)
4669-726d;1.6.0.48;small_HMI.hex; firmware
426f-6f74;1.0.0.2;bootloader_v2.hex; Bootloader
496f-7072;1.0.0.0;MyProject1.iop; customer-specific system menu
4c5f-3f3f;0.0.0.8;system_menu_de.iop; system menu (additional language)
5069-6374;1.0.0.3;start_up.png; illustration on start screen
```

Where:

- Column 1 (`496f-7072`) is the EUID that identifies the content of the system menu.
  It must not be changed!
- Column 2 is the version number of the IOP file of the system menu; available at runtime.
- Column 3 (`MyProject1.iop`) is the name of the IOP file.
  Adapt the content of this column to the name of your IOP file if necessary.
- Column 4 is the comment field. It is not compiled.

Multiple menu languages

To integrate additional menu languages see Language selection in the system menu [▶ 32].

7.3 Updating the firmware and system menu via USB

The firmware and system menu of the JVM-205 can be updated via USB.

An update cable is required for the update (Accessories [▸ 41]).

ⓘ Info

Do not update while using

The firmware and system menu must not be updated while the motor is running or the vehicle is being used.

ⓘ Info

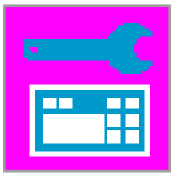
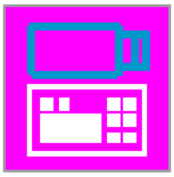
This procedure requires an operating system that is already installed.

Update preparation

✓ The required firmware package *JVM-205-K00-O31_rev_x_x_x_x.zip* has been provided by support (support@bucherautomation.com).

1. Make certain the control unit is turned off.
2. Connect the 12 V voltage supply.
3. Unpack the firmware package *JVM-205-K00-O31_rev_x_x_x_x.zip*.
   The sequence *x_x_x_x* represents the firmware version.
4. Create the EDC file: Creating an EDC file.

Perform update

1. Copy file *41434445 .EDC* onto a USB stick.
2. Insert the USB stick in the USB A socket of the device.
3. Turn on the device.
4. Open the system settings:
   
5. Select option Update via USB:
   

⇨ The firmware and system menu are updated.

7.4 Updating the firmware and system menu via CAN bus

The firmware and system menu of the JVM-205 can be updated via CAN bus.

For a detailed description of the Python script *smallhmi_updater.py* please contact support at support@bucherautomtation.com.

ⓘ Info

Do not update while using

The firmware and system menu must not be updated while the motor is running or the vehicle is being used.

Update preparation

✓ The required file package *JVM-205_CAN_updater_rev_1_0_0_2.zip* and the system requirements have been provided by support (support@bucherautomation.com).

✓ The required PEAK-P-CAN USB/CAN dongle is on hand.

✓ A PC with operating system Windows 10 Pro 64 is available.

1. Connect the control unit and PC by means of the dongle and a CAN cable.
2. Make certain the control unit is turned off.
3. Connect the 12 V voltage supply.

Perform update

1. Insert the PEAK-P-CAN USB/CAN dongle in the USB interface of a Windows computer.
2. Unpack file package *JVM-205_CAN_updater_rev_1_0_0_2.zip* in a directory of your choice.
3. Create the EDC file: Creating an EDC file.
4. Optional: Replace the standard file *41434445_EDC_x_x_x_x.hex* with a customized file.
5. Connect the JVM-205 with the dongle.
6. Turn on the JVM-205.
   NOTICE! Make certain it is turned on throughout the entire process.
7. Run file *JVM-205 _updater.bat*.

   ⇨ The device-specific firmware file (*41434445_EDC_x_x_x_x.hex*) is loaded in the control unit via CAN.
8. Restart the control unit. To do this, either turn on the ignition or press the On/Off key on the control unit.

⇨ The firmware and system menu are updated.

8 Programming

This section is intended exclusively for qualified personnel familiar with ISOBUS visualization.

The device already has a system menu for end users when it leaves the factory that can be adapted to relevant needs.

8.1 Properties of the system menu

The system menu contains the general functions of the control unit. It is written by a customer-specific IOP file that is transferred to the control unit together with the firmware. The script for linking the two files is provided by Bucher Automation together with the file package.

No ECU visualization is required to use these functions. The objects of the system menu are used separately from the ECU objects, which means that no check for any object collisions is necessary.

Displaying the system menu

To open the system menu while ECU visualization is active, press the HOME key.

ⓘ Info

Version-dependent availability of device functions

The description of the object IDs contains all the available system menu functions for the product. However, some functions are not accessible for all device versions. Because of this, check the device-specific description before adapting the system menu:

- The RTC values are synchronized with the real-time clock, provided the hardware is present.
- The hardware-specific objects are only active if the corresponding hardware is present:
  - Input or output for ignition

8.1.1 Reserved Object IDs

Data mask/alarm mask objects

Alarm masks are defined without a signal tone.

Object ID	Description
1000	Home data mask.
1001 … 1029	Data masks for settings.
1030	Download the IOP file.
1031	Prompt for update via USB.
1032	Update via USB.
1036	InputNumber editor.
1037	InputString editor.
1038	InputList editor.
1040	Messaging: The IOP cache has been deleted!
1041	Messaging: New settings become active after a restart. Appears for example after the language of the system menu is changed.

Object ID	Description
1042	Alarm mask: Shows the "Shutdown" messaging. Appears while the system is being shut down.
1050	Alarm mask: Error during download (outdated).
1051	Alarm mask: Error: Connection lost.
1052	Alarm mask: Error during update via USB.
1053	Alarm mask: Error: USB stick not found.
1054	Alarm mask: Error: Information about parsing error.
1100	Stop key pressed. Appears in the system menu only when the Stop key is pressed.

Tab. 19: Data mask/alarm mask objects

FillAttributes objects

These objects can be used to define different background colors for incorrect values in input fields. The numeric editor checks for min/max values. As soon as a value is outside of the permitted range, object 101 is used, otherwise object 100.

Object ID	Description
100	Used when the entry in the editor field is correct. Set this attribute on rectangle object 200.
101	Used when the entry in the editor field is faulty. Set this attribute on rectangle object 200.

Tab. 20: FillAttribute objects

Rectangular objects

Object ID	Description
200	Editor field

Tab. 21: Rectangular objects

String variables

Object ID	Description
65000	Content of the editor field.
65001	Original value of the edited object.
65002	Serial number
65003	Manufacturing date
65004	Software version
65005	Hardware revision
65006	Device type
65007	ActiveObjectPool version

Object ID	Description
65008	Min/max value in the editor as text string. Structure is automatically arranged as follows: [%min_value-%max_value]
65009	Information about the EDC file (version of the loaded file collection).
65010	Version of the saved EDC file.
65042	Key name (access from KEY-TEST data mask only).
65044	Bootloader version

Tab. 22: String variables

Numeric variables

The variables identified with an asterisk (rw*) are saved in non-volatile memory (EEPROM).

Object ID	Access	Description
65101	rw	Edited Boolean value.
65102	ro	Download status of the IOP file as %.
65103	ro	Operating hours
65104	ro	Memory usage as %.
65105	rw	Real-time clock: Year
65106	rw	Real-time clock: Month
65107	rw	Real-time clock: Day
65108	rw	Real-time clock: Weekday
65109	rw	Real-time clock: Hour
65110	rw	Real-time clock: Minute
65111	rw	Real-time clock: Second
65112	rw*	Language code **NOTICE! Language according to index number (**Language selection in the system menu [▶ 32]).
65113	rw*	System units of measure (0 = metric, 1 = imperial)
65114	rw*	Decimal separator (0 = comma, 1 = dot)
65121	ro	Device temperature in °C (CPU chip temperature).
65122	ro	Operating voltage in mV.
65123	rw*	Background brightness of the screen as %; adjustment range 10 % … 100 %.
65124	rw*	Background brightness of the keyboard as %; adjustment range 0 % … 100 %.
65126	ro	Battery voltage of the real-time clock in V.
65127	ro	Upper part of the scrollbar of the ListObject editor.

Object ID	Access	Description
65128	ro	Lower part of the scrollbar of the ListObject editor.
65137	rw*	Max. volume as %; Adjustment range: 0% … 100%.
65138	rw*	Real-time clock format: 0 = 12 h, 1 = 24 h
65139	rw*	Real-time clock time change: 0 = standard time, 1 = daylight savings time
65140	rw*	am/pm setting for real-time clock: 0 = am, 1 = pm, 2 = empty (for 24-h format)
65146	rw*	Operating principle of the ignition: 0 = input, 1 = output
65164	rw*	Activate key tone.
65165	rw*	UT instance number: 0 … 31
65168	rw*	Background brightness control: 1 = automatic, 0 = fixed (no brightness reduction by the light sensor), 2 … 100 = fixed backlight sensitivities (optional)
65180	rw	Date format selection: Adjustment range: 0 … 5; affects pointer 65520.
65181	ro	Number of active download sessions; 0 = no pool download from the ECU.
65182	ro	Error code for parsing errors; 0 = no errors.
65183	ro	ID of the incorrect object. Used for parsing errors.

Tab. 23: Numeric variables

Object pointer

Object ID	Points to
65200	Edited list object elements
65201	Edited list object elements +1
65202	Edited list object elements +2
65203	Edited list object elements +3
65204	Edited list object elements +4
65205	Edited list object elements +5
65206	Edited list object elements -1
65207	Edited list object elements -2
65208	Edited list object elements -3
65209	Edited list object elements -4
65210	Edited list object elements -5

Object ID	Points to
65219	Edited list object elements, previously selected element
65300	Object pointer to containers of keys. Applied by StringEditor.
65500 … 65509	Active ECU object pool (up to 10 ECUs). If ECU is connected: Points to working set designator. If ECU is not connected: Pointer is NULL.
65520	Pointer to container holding the selected date format (OID 3030 … 3035).

Tab. 24: Object pointer

Softkey objects

Object ID	Description
5001	No longer in use; reserved for backward compatibility.

Tab. 25: Softkey objects

Key objects

Object ID	Description
256 … 511	Button objects for alphanumeric keyboard; Offset: 8-bit character code (IEC8859-1) Example: Code for letter "A" (ASCII 65_{dec}) = 256 + 65 = 321

Tab. 26: Key objects

Container objects

Object ID	Description
3100 … 3199	Container for keyboard assignments.
3000	Listbox
3030 … 3035	Container for different date formats.

Tab. 27: Container objects

8.1.2 Physical keys

Key name	Description / behavior
PWR_ON	Power supply on/off; Turn on: press briefly once. Turn off: press and hold for approx. 2 s.
ISB	Switches the inputs off when activated. Press and hold for 1 s.
HOME	Switch between start screen of the system menu and visualization of active ECU objects.

Key name	Description / behavior
TOP	Moves the input focus up. Press once: increase/ upward navigation by lines or steps. Press and hold: Progressive increase/upward navigation.
BOTTOM	Moves the input focus down. Press 1 time: reduction/ downward navigation by lines or steps. Press and hold: Progressive reduction/downward navigation.
ENTER	Starts or exits editing mode; changes are saved (OK key).
SK_1 … SK_n	Softkeys (F1 … F8, M1, M2); function depends on the selected mask.

Tab. 28: Function and behavior of physical keys

8.1.3 Key Codes

Key code	Description
1 … 29	Opens data or alarm masks 1000 …. 1029; Example: 1 = opens screen 1001. 29 = opens Home screen (1000).
30	Jumps to the pool of active objects, if present (first ECU).
33	Like UP key.
34	Like DOWN key.
35	Like ENTER key.
36	Like ESC key.
37	Like ISB key.
38	CLR: Deletes the content of the editor field (numeric: set to zero)
39	DEL: Deletes the marked character in the editor field.
40	Deletes all saved IOPs.
41	Jumps to the end of the list.
42	Jumps to the beginning of the list.
43	Increases the numeric value.
44	Reduces the numeric value.
46	Opens the status mask for the USB update and updates the variable with the USB information.
47	Starts the update via USB and opens the USB update mask.
48	Opens touch calibration (system-internal mask).
50	Selects edited list object: current element +1 (following element).
51	Selects edited list object: current element +2.
52	Selects edited list object: current element +3.
53	Selects edited list object: current element +4.
54	Selects edited list object: current element +5.

Key code	Description
55	Selects edited list object: current element -1 (previous element).
56	Selects edited list object: current element -2.
57	Selects edited list object: current element -3.
58	Selects edited list object: current element -4.
59	Selects edited list object: current element -5.
68	Opens the KEY TEST data mask.
69	Opens the FLASH TOOL data mask.
90 … 99	Scrolls through active object pool (90: ECU 1, 99: ECU 10)
102	Scrolls on softkey page; Becomes relevant, when more softkeys have been configured than there are hardware keys available.
105	Cancels editing. Changes to string or numeric variables are discarded.
106	Closes editing with OK. Manual changes to the variable are saved.
200	Transfers the ID (- 256) of the key that was pressed to the editor as an ASCII key code.
202	Forwards the ID of the key that was pressed (ID % 100) + 3100 to the value of object pointer 65300.
203	Shows the download mask if a download is active.

Tab. 29: Key codes

8.2 Character set types of the system menu

The character set types listed in the table below are supported.

Character formats bold, italic and underlined are not supported.

Type	Character set
0	ISO8859-1 (ISO Latin-1, West European)
1	ISO8859-15 (ISO Latin-9, West European)
2	ISO8859-2 (ISO Latin-2, Central European)
4	ISO8859-4 (ISO Latin-4, North European)
5	ISO8859-5 (Cyrillic)
7	ISO8859-7 (Greek)

Tab. 30: Supported character set types

8.3 Language selection in the system menu

The EDC file contains a separate IOP file for each language. Once while the device is booting up, the firmware loads the IOP file of the system menu and automatically searches for the corresponding IOP file with the contents in the selected system language.

The individual language files in the file collection are written in file *edc.cfg*. EUID *496f-7072* contains the language file of the system menu currently being used. Separate IOP files must be added for each additional language that is needed. This is also done in file *edc.cfg* using EUID *4c5f-3f3f*. This IOP file contains only those objects that differ from the default system language. Generally this does not relate to the system menu as a whole.

EDC file – sample configuration

Example of the language-related section of the edc.cfg file:

- `496f-7072;1.0.0.0;language_EN.iop; system menu`

  English is the default language of the system menu and is always parsed.
- `4c5f-3f3f;1.0.0.0;language_DE.iop; first language-specific IOP file`

  German was added as the first additional system language and is parsed as soon as German is selected as the menu language in the settings.
- `4c5f-3f3f;1.0.0.0;language_FR.iop; second language-specific IOP file`

  French was added as the second additional system language and is parsed as soon as French is selected as the menu language in the settings.

Compiler *edc.py* reads the two-place language abbreviation of the IOP file name and stores the binary information in the EDC file together with the corresponding hexadecimal representation (UDS and Emergency).

Language abbreviation

The table below shows the language abbreviations permitted in ISOBUS for the languages supported by the device as well as their assignment to an index. The language abbreviation is forwarded via the ISOBUS. The selected index is stored in variable ID 65112.

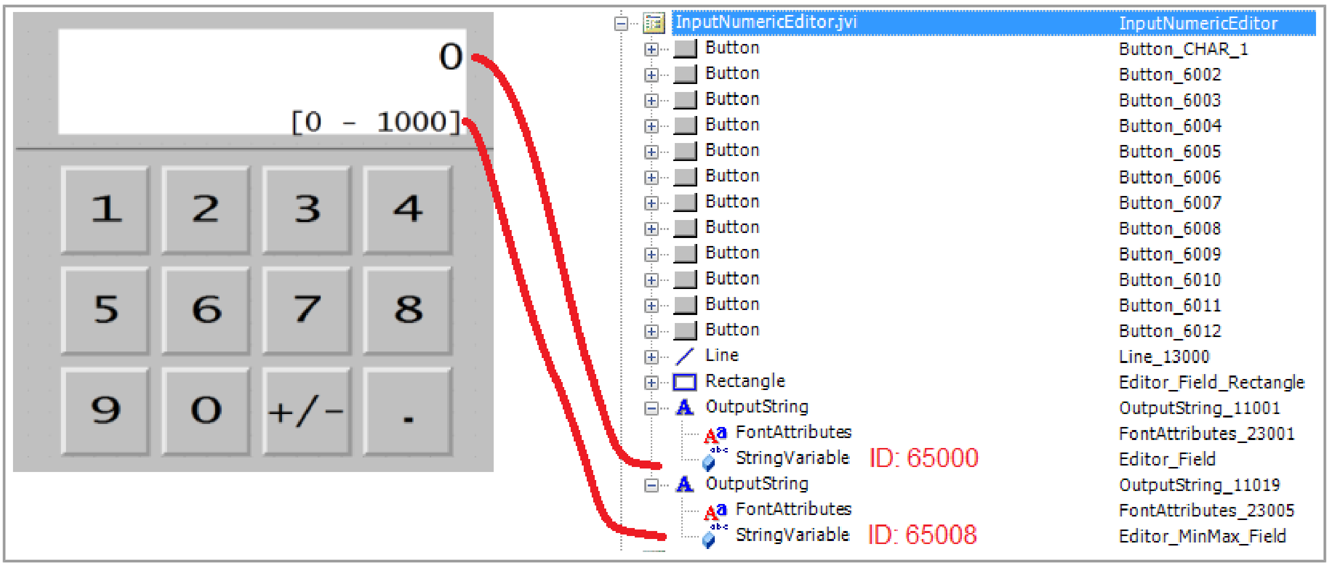
Index	Language abbreviation	Language
0	en	English
1	nl	Dutch
2	fr	French
3	de	German
4	it	Italian
5	es	Spanish
6	da	Danish
7	sv	Swedish
26	cs	Czech
45	hr	Croatian
46	hu	Hungarian
83	no	Norwegian
88	pl	Polish
94	ro	Romanian
122	tr	Turkish

Tab. 31: Assignment of index and language abbreviation

8.4 Programming examples

8.4.1 Programming example for numeric editor

Buttons with numbers, characters and a decimal point are available in the numeric editor. The editor value and min/max range are represented as a string.



```
InputNumericEditor.jvi          InputNumericEditor
  Button                        Button_CHAR_1
  Button                        Button_6002
  Button                        Button_6003
  Button                        Button_6004
  Button                        Button_6005
  Button                        Button_6006
  Button                        Button_6007
  Button                        Button_6008
  Button                        Button_6009
  Button                        Button_6010
  Button                        Button_6011
  Button                        Button_6012
  Line                          Line_13000
  Rectangle                     Editor_Field_Rectangle
  OutputString                  OutputString_11001
    FontAttributes              FontAttributes_23001
    StringVariable  ID: 65000   Editor_Field
  OutputString                  OutputString_11019
    FontAttributes              FontAttributes_23005
    StringVariable  ID: 65008   Editor_MinMax_Field
```

Fig. 7: Programming example for numeric editor

8.4.2 Programming example for list editor

This example contains a list with 5 navigable lines: the active line ±2 relative positions. Navigation by ±5 lines is possible.

The scrollbar consists of two linear bar graphs with one superimposed on the other.

Keys X (key code 36) and OK (key code 35) are defined as buttons.

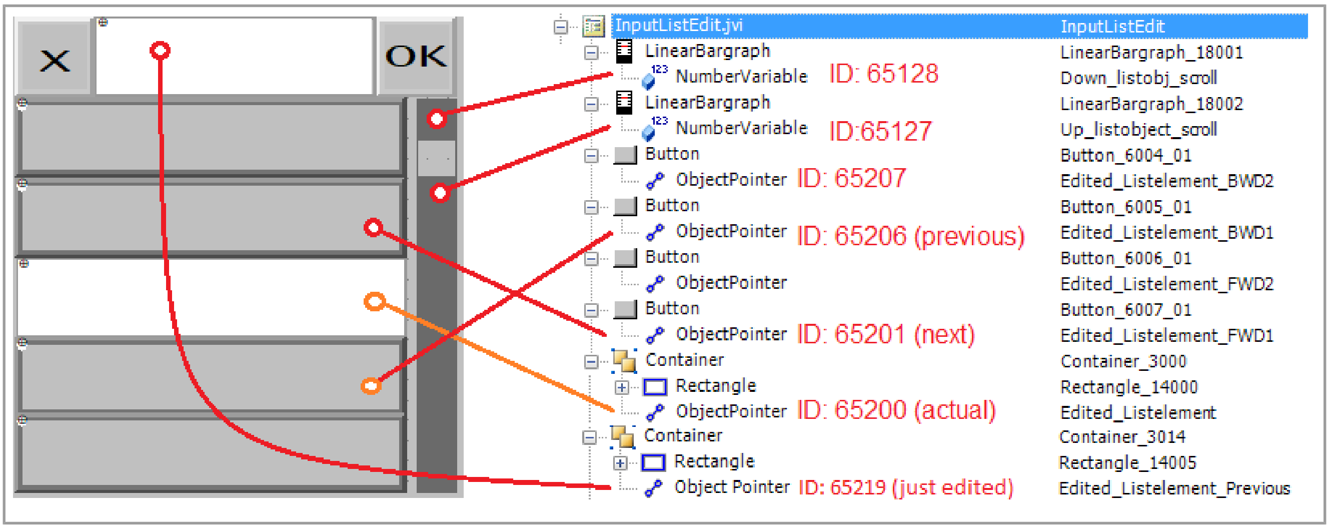


Fig. 8: Programming example for list editor

8.4.3 Programming example for string editor

The example shows a keyboard assignment. A letter is assigned to each key based on the following formula:

ObjectID = 256 + ASCII code of the respective letter.

Example:

The letter A has ASCII code 65. Therefore:

256 + 65 = 321

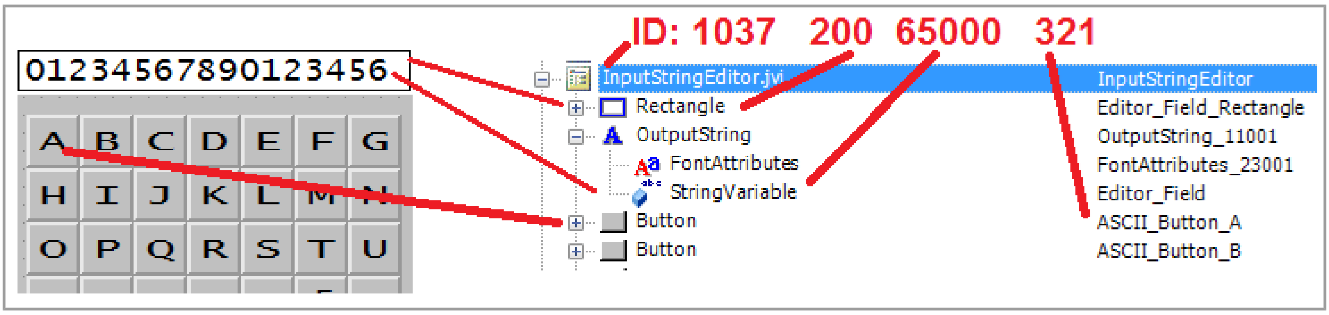


Fig. 9: Programming example for string editor

The object ID of the string editor is 1037.

The object ID of the editor field (rectangle object) is 200. The ID of the editor field content is stored in the value of object ID 65000, a string variable.

8.4.4 Programming example for multiple keyboards

Object IDs 3100 … 3199 are reserved for keyboard containers. This means that 100 keyboard sets can be programmed. The keyboard with object ID 3100 is the assignment that is shown when the string editor is opened.

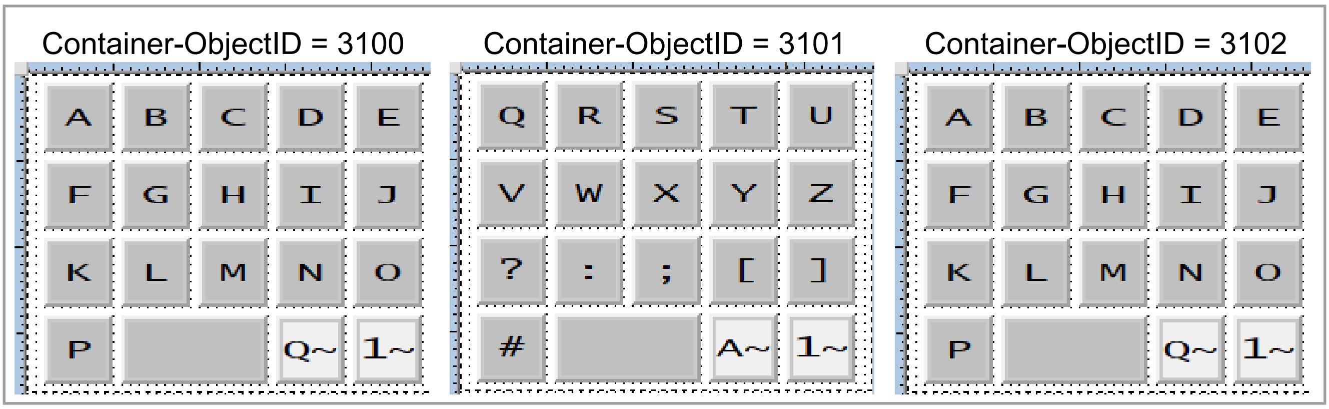


Fig. 10: Programming example for multiple keyboard sets

Follow the steps below to create multiple keyboard sets.

1. Create a container for each keyboard.
2. Assign the appropriate buttons. There are 2 types available:
   - **Keys with standard letters:**

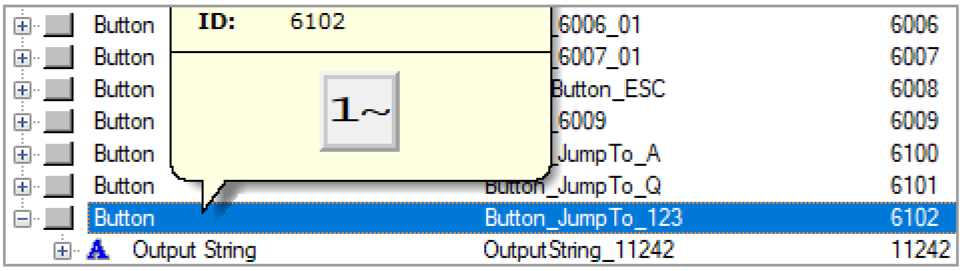
     Key code: 200

     Object ID: ASCII-Code + 256
   - **Keys that link to the next keyboard set:**

     Key code: 202

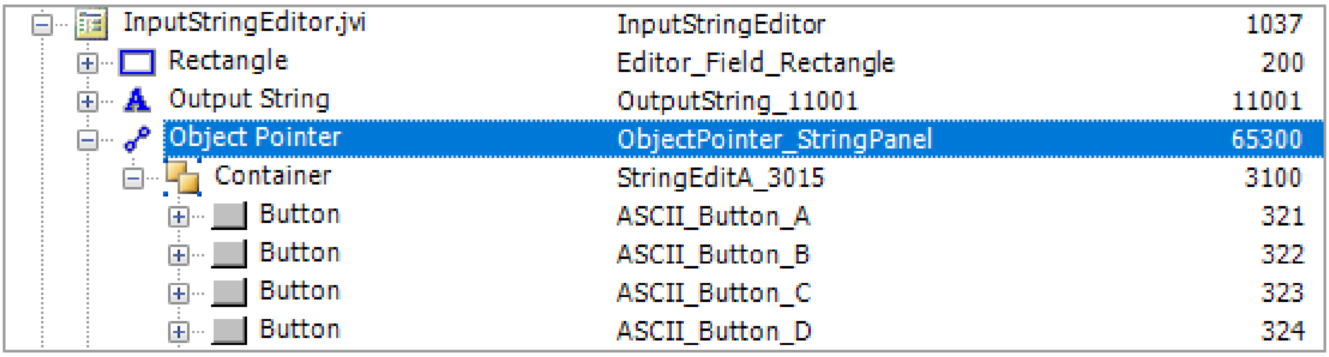
     Object ID: 61 xx

     Where
     xx stands for the last two places of the object ID of the targeted container.
     In the next example, ID 6102 replaces this container with 3102.



Button	6006_01	6006
Button	6007_01	6007
Button	utton_ESC	6008
Button	6009	6009
Button	JumpTo_A	6100
Button	Button_JumpTo_Q	6101
Button	Button_JumpTo_123	6102
Output String	OutputString_11242	11242

3. Assign an object pointer to the data mask of the string editor (object ID 1037) instead of an individual key. The object pointer refers to another keyboard container. The object ID of the pointer (65300) behaves differently: Its value changes as soon as the keyboard is switched.

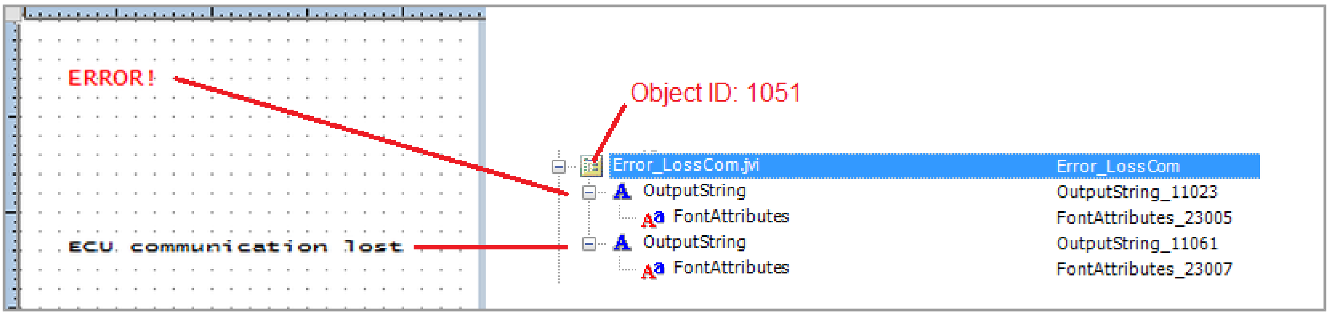


InputStringEditor.jvi	InputStringEditor	1037
Rectangle	Editor_Field_Rectangle	200
Output String	OutputString_11001	11001
Object Pointer	ObjectPointer_StringPanel	65300
Container	StringEditA_3015	3100
Button	ASCII_Button_A	321
Button	ASCII_Button_B	322
Button	ASCII_Button_C	323
Button	ASCII_Button_D	324

Fig. 11: Object pointer for changing the keyboard set

8.4.5 Programming example for error messages

This example shows an error message that appears when the connection to the ECU is interrupted. Additional error messages can be created in a similar manner using the relevant object ID.



ERROR!

ECU communication lost

Error_LossCom.jvi	Error_LossCom
OutputString	OutputString_11023
FontAttributes	FontAttributes_23005
OutputString	OutputString_11061
FontAttributes	FontAttributes_23007

Fig. 12: Programming example for error messages

8.5 Defining the date format

The date format applied on the control unit is in line with ISO 11783-7:2009, section A.23.3). The numeric variable (object ID 65180) is used to select the desired date format (values 0 … 5).

Value	Date format	Container
0	ddmmyyyy	3030
1	ddyyymm	3031
2	mmyyyydd	3032
3	mmddyyyy	3033
4	yyyymmdd	3034
5	yyyyddmm	3035

Tab. 32: Available date formats

Pointer 65520 points to the selected container. Containers 3030 … 3035 contain the data related to the year, month and day in the relevant order. For example, container 3034 organizes the date components in the order year – month – day.

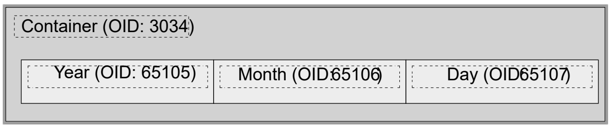
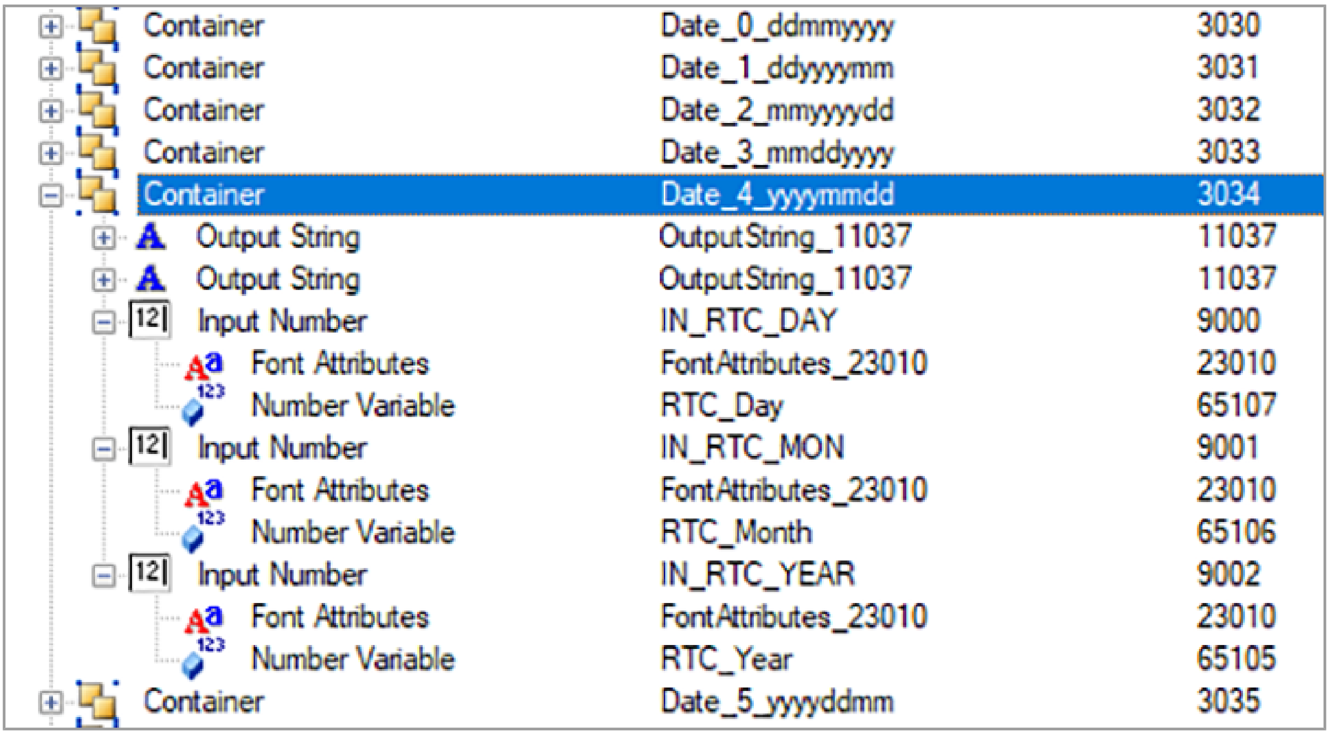


Fig. 13: Container IDs of the date components



Container	Date_0_ddmmyyyy	3030
Container	Date_1_ddyyyymm	3031
Container	Date_2_mmyyyydd	3032
Container	Date_3_mmddyyyy	3033
Container	Date_4_yyyymmdd	3034
Output String	OutputString_11037	11037
Output String	OutputString_11037	11037
Input Number	IN_RTC_DAY	9000
Font Attributes	FontAttributes_23010	23010
Number Variable	RTC_Day	65107
Input Number	IN_RTC_MON	9001
Font Attributes	FontAttributes_23010	23010
Number Variable	RTC_Month	65106
Input Number	IN_RTC_YEAR	9002
Font Attributes	FontAttributes_23010	23010
Number Variable	RTC_Year	65105
Container	Date_5_yyyyddmm	3035

Fig. 14: Container for date format

The format is defined with variable 65180. The output strings named here contain the date format; for example, the value of the first string is DD-MM-YYYY.

The object IDs of the strings are random. Only the positions in the input list are permanent (1 … 6).

9 Maintenance

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

9.1 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.

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

9.3 Return and Disposal

Disposal options

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

Meaning of WEEE icon

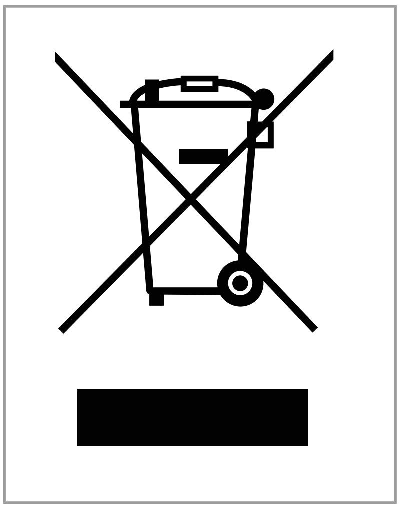


Fig. 15: WEEE icon – crossed out trash can

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

Batteries

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

Personal data

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

10 Service

10.1 Technical support

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

Technical Support | Bucher Automation - We automate your success.

Or send an email to:

support@bucherautomation.com

Please supply the following information when contacting Technical Support:

- Hardware revision and serial number
  The hardware revision and serial number is printed on the nameplate of the product.

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 Bucher Automation AG.

11.1 Accessories

Component	Item number
Programming and service cable: – USB socket & RS-232 to SubD; – 3-way switch with the following positions: Boot: Device starts in boot mode. OFF: Device start by means of its own power button. IGN: Device starts via external ignition.	60884809_00
INCAB T-CONNECT T-cable for connecting the control unit to the InCab socket.	60884890_00
USB-Update-Kit Adapter with USB interface for firmware update via USB.	60887276_00

List of figures

List of tables

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

www.bucherautomation.com