

PCAN-RS-232

Programmable Converter
CAN to RS-232

User Manual v1.1.0



Products taken into account

Product Name	Model	Part number
PCAN-RS-232	Plastic casing, screw terminal block	IPEH-002100

Product names mentioned in this document may be the trademarks or registered trademarks of their respective companies. They are not explicitly marked by “™” and “®”.

© 2013 PEAK-System Technik GmbH

PEAK-System Technik GmbH
Otto-Roehm-Strasse 69
64293 Darmstadt
Germany

Phone: +49 (0)6151 8173-20
Fax: +49 (0)6151 8173-29

www.peak-system.com
info@peak-system.com

Document version 1.1.0 (2013-03-21)

Contents

1	Introduction	5
1.1	Properties at a Glance	5
1.2	Scope of Supply	6
1.3	Prerequisites for Operation	6
2	Connectors and Coding Solder Jumpers	7
2.1	Screw Terminal Block	8
2.2	J5 Connector Panel: JTAG Ports	9
2.3	Coding Solder Jumpers	11
3	Operation	13
4	Software	14
4.1	Installing the GNU ARM Toolchain	14
4.2	Library	15
4.3	Firmware Examples	15
4.3.1	Compiling a Firmware Example	16
5	Firmware Upload	17
5.1	Uploading Firmware via CAN	17
5.1.1	System Requirements	17
5.1.2	Preparing Hardware and Software	18
5.1.3	Uploading the Firmware	19
5.2	Uploading Firmware via the Serial Connections	22
6	Technical specifications	24
Appendix A	CE Certificate	25
Appendix B	Dimension Drawings	26

Appendix C Port Assignment of the Microcontroller 27

1 Introduction

The PCAN-RS-232 converter has one CAN channel and one RS-232 port. The data traffic between CAN and RS-232 is controlled by a freely programmable microcontroller. Thus, devices without CAN connection (e.g. PLC systems) can be connected to a CAN bus via almost any RS-232 protocol.

The provided library and Yagarto toolchain can be used to produce your own firmware and then transfer it to the converter via CAN. This gives a whole range of options for manipulation, evaluation, filtering, and routing of the data traffic.

On delivery the PCAN-RS-232 converter is supplied with an example firmware that routes from CAN to RS-232 and vice versa. It allows to configure the data transfer as well as the hardware with serial control commands. The corresponding source code is included.

1.1 Properties at a Glance

- Microcontroller of the NXP LPC21 series (16/32-bit ARM CPU)
- External 32-kByte EEPROM
- High-speed CAN channel (ISO 11898-2) with 40 kbit/s to 1 Mbit/s (lower bit rates on request)
- Data transfer between CAN and RS-232 with a maximum bit rate of 115,200 bit/s
- Additional digital input (Low-active) and output (Low-side switch)
- Duo LED for status indication
- 10-pole screw terminal block (Phoenix), exposed connector contacts on request

- └ Supply voltage from 8 to 30 V
- └ Extended operating temperature range from -40 to +85 °C (-40 to +185 °F)
- └ Import of a new firmware via CAN
- └ 4-bit coding of the hardware by solder jumpers
- └ Plastic casing

1.2 Scope of supply

- └ PCAN-RS-232 in a plastic casing
- └ 10-pin screw terminal block
- └ Windows development software (Yagarto GNU ARM toolchain, flash program)

1.3 Prerequisites for operation

- └ Power supply in the range of 8 to 30 V DC
- └ For uploading a new firmware via CAN:
 - CAN interface of the PCAN series for the computer (e.g. PCAN-USB)
 - Operating system Windows 8/7/Vista/XP (32/64-bit)

2 Connectors and Coding Solder Jumpers

The PCAN-RS-232 converter has a 10-pole screw terminal block for connecting the following components:

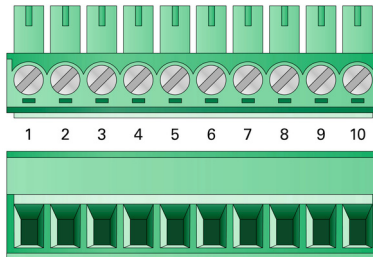
- └ Supply voltage
- └ CAN
- └ RS-232
- └ Digital input and digital output
- └ CAN bootloader activation

For direct access to the debugging ports (JTAG) of the micro-controller, an additional – yet not equipped – connector panel is available on the circuit board of the converter.

Furthermore the board has four coding solder jumpers in order to assign a fixed status to the corresponding input bits of the micro-controller. A concrete application is to identify a PCAN-RS-232 converter on the CAN bus for a firmware upload, especially if there are several converters connected and in operation.

The following subsections describe each connector assignment.

2.1 Screw Terminal Block

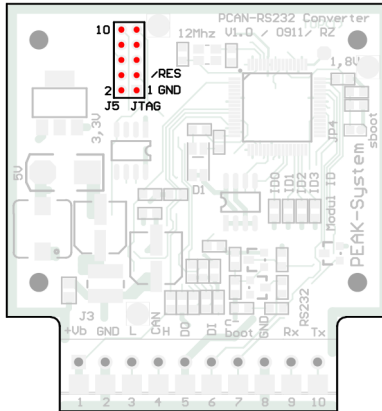


Terminal	Identifier	Function
1	+Vb	Power supply 8 - 30 V DC
2	GND	Ground
3	CAN_L	Differential CAN signal
4	CAN_H	
5	DOut	Digital output, Low-side switch
6	DIn	Digital input, Low-active
7	Boot CAN	CAN bootloader activation, High-active
8	GND	Ground
9	RS-232 RxD	RS-232 interface
10	RS-232 TxD	

For further connection details that are not needed for programming of the PCAN-RS-232 converter because of implementation in a library, see also Appendix C *Port Assignment of the Microcontroller* on page 27.

2.2 J5 Connector Panel: JTAG Ports

The unpopulated connector panel J5 on the PCAN-RS-232 converter's circuit board provides an access option to the JTAG ports of the LPC2194/01 microcontroller (μC) for hardware debugging.

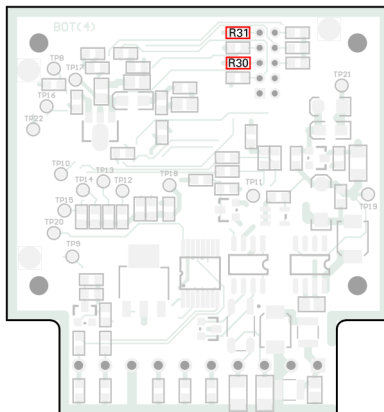


JTAG panel on the circuit board (not equipped)

Pin	Signal	Port μC	Internal wiring
1, 2	GND		
3	/Reset	/Reset	Pull-up
4	3.3 V		
5	TCK	P1.29	Pull-down (R30)
6	TMS	P1.30	Pull-up
7	TDO	P1.27	Pull-up
8	TDI	P1.28	Pull-up
9	RTCK	P1.26	Pull-down (R31)
10	TRST	P1.31	Pull-up

If constant internal pull-down wiring of the TCK or RTCK signals is not suitable for your purposes, you can remove the respective pull-

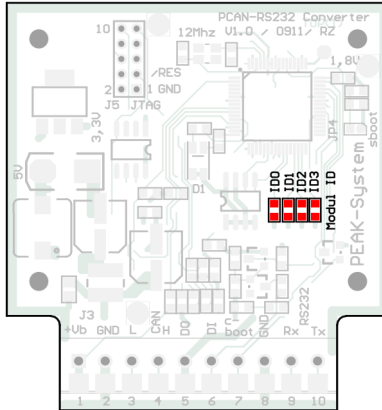
down resistor on the circuit board of the PCAN-RS-232 converter by soldering it out.



Pull-down resistors on the circuit board bottom:
R30 for pin 5 TCK, R31 for pin 9 RTCK

2.3 Coding Solder Jumpers

The four positions for coding solder jumpers (ID0 - ID3) are each assigned to one port of the LPC2194/01 microcontroller (μC).



Coding solder jumpers on the circuit board

Position	0	1	2	3
Port μC	P0.4	P0.5	P0.6	P0.7

Position is ...	Status at the port
bridged	Low
open	High

The status of the ports is relevant in the following cases:

- └ The loaded firmware is programmed so that it reads the status at the corresponding ports of the microcontroller. For example, the activation of certain functions of the firmware or the coding of an ID is conceivable here.
- └ For a firmware upload via CAN the PCAN-RS-232 converter is identified by a 4-bit ID which is determined by solder jumpers. A

bit is set (1) when the corresponding solder jumper position is open (default setting: ID 15, all positions open).

Position	0	1	2	3
Binary digit	0001	0010	0100	1000
Decimal equivalent	1	2	4	8

See also section 5.1 *Uploading Firmware via CAN* on page 17.

3 operation

The PCAN-RS-232 converter is activated by applying the supply voltage to the respective input pins (see chapter 2 *Connectors and Coding Solder Jumpers* on page 7). The firmware in the flash memory is subsequently run.

The status indication of the LED depends on the used firmware.

On delivery the PCAN-RS-232 is supplied with an example firmware that routes from CAN to RS-232 and vice versa. It allows to configure the data transfer as well as the hardware with serial control commands.

The LED is green when the converter is switched on, and blinks during an established connection to a serial host (e.g. the shareware program CANHacker).

You can find the documentation about the example firmware on the supplied DVD in the directory branch

```
/Develop/Microcontroller hardware/PCAN-RS-232/Example/
```

and there as HTML pages under

```
6_CAN_TO_SER_BY_COMMAND/help/index.html
```

4 Software

This chapter covers the installation of the Yagarto GNU ARM toolchain and gives notes about the software library and the firmware examples.

Software, source code, and additional information are included on the supplied DVD in the following directory branch:

```
/Develop/Microcontroller hardware/PCAN-RS-232/
```

4.1 Installing the GNU ARM Toolchain

To compile the code examples and the custom firmware code under Windows, install Yagarto on your computer. Yagarto is a collection of tools to develop applications for ARM processors and microcontrollers on Windows platforms. The collection includes the GNU GCC compiler for C and C++, Make, and further tools. Further information about Yagarto: www.yagarto.de

System requirement: Windows 8/7/Vista/XP (32/64-bit)

▶ Do the following to install Yagarto:

1. From the directory branch on the provided DVD mentioned above, change to the `Compiler` subdirectory.

The directory contains the two installation programs `yagarto-*.exe` and `yagarto-tools-*.exe`.

2. Execute the first installation program and follow its instructions.

If you don't want to use the default destination folder, make sure that your customized path doesn't contain any spaces. Otherwise compile operations will not work later.

3. Afterwards, execute the second installation program and follow its instructions.

In the system environment, the installation programs create search paths for the executable files. These new search paths are effective only for programs and command prompts that are started afterwards.

4.2 Library

The development of applications for the PCAN-RS-232 is supported by the library `libPCAN-RS-232-GNU*s.a` (* stands for version number), a binary file. You can access all resources of the converter by means of this library. The library is documented in the header files (*.h). The files are located in each example directory.


4.3 Firmware Examples

On the DVD, the `Example` subdirectory contains source code for several firmware examples that you can use and test directly and that you can reuse for custom firmware.

On delivery the PCAN-RS-232 converter is supplied with the example firmware `6_CAN_TO_SER_BY_COMMAND` that routes from CAN to RS-232 and vice versa. It allows to configure the data transfer as well as the hardware with serial control commands.

4.3.1 Compiling a Firmware Example

▶ Do the following to compile a firmware example under Windows:

1. From the provided DVD, copy the subdirectory of the desired example from the `Example` directory to the local hard disk.
2. Open a **command prompt** by using the Windows Start menu. Alternatively you can press the key combination  + **R** and enter `cmd.exe` as program to be executed.
3. At the command prompt change to the previously copied directory.
4. Execute the following command in order to clean-up the target directories (e.g. `.out`) from files that have been generated earlier:

```
make clean
```

5. Execute the following command to compile the example firmware:

```
make all
```

If the compiler has finished without errors (“Errors: none”), you can find the firmware file with the extension `.bin` in the subdirectory `.out`. This file is then used for firmware upload to the PCAN-RS-232 converter.

5 Firmware Upload

The microcontroller in the PCAN-RS-232 converter can be equipped with new firmware in two different ways:

- └ Via CAN. The scope of supply includes the Windows program PCAN-Flash to transfer the firmware from the computer to the PCAN-RS-232 converter. This is the recommended method for a firmware upload.
- └ Via the serial ports of the microcontroller (only in special cases). For this the access to the circuit board of the PCAN-RS-232 converter is needed.

5.1 Uploading Firmware via CAN

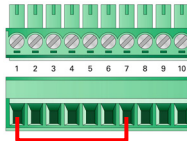
5.1.1 System Requirements

The following prerequisites must be given, so that the PCAN-RS-232 converter can be updated with new firmware:

- └ CAN interface of the PCAN series for the computer (e.g. PCAN-USB)
- └ CAN cabling between the CAN interface and the PCAN-RS-232 converter with proper termination (120 Ω on each end of the CAN bus)
- └ Operating system Windows 8/7/Vista/XP (32/64-bit)
- └ If you want to update several PCAN-RS-232 converters connected to the same CAN bus, you must assign a unique ID to each converter. See section 2.3 *Coding Solder Jumpers* on page 11.

5.1.2 Preparing Hardware and Software

- ▶ Perform the following steps for preparation of the hardware:
 1. Switch the PCAN-RS-232 converter off by disconnecting it from the power supply.
 2. Establish a connection between “Boot CAN” and “U_b” at the connectors of the converter.



Connection at the screw terminal block
between terminals 1 and 7

This measure later applies the “Boot CAN” connection with a high level.

3. Connect the CAN bus of the converter with a CAN interface connected to the computer. Pay attention to the proper termination of the CAN cabling (2 x 120 Ω).
- ▶ Perform the following steps for preparation of the software:
 1. On the supplied DVD, change to the following directory:
`/Develop/Microcontroller hardware/PCAN-RS-232/`
 2. Copy the subdirectory `PcanFlash` to the local hard disk.

The contained Windows software that copies the Firmware via CAN (`PcanFlash.exe`) can only be started from a data carrier that is writable.

5.1.3 Uploading the Firmware

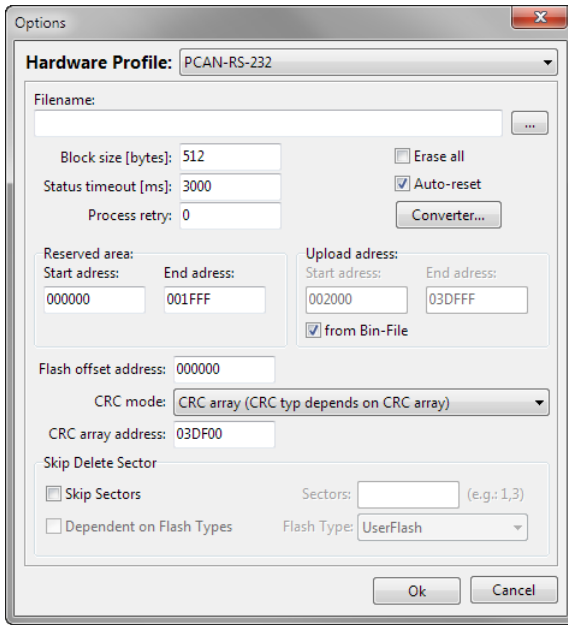
▶ The process of transferring a new firmware to the PCAN-RS-232 converter is as follows:

1. Ensure that a connection is established between the “Boot CAN” and “U_b” connections of the converter (details: see above).

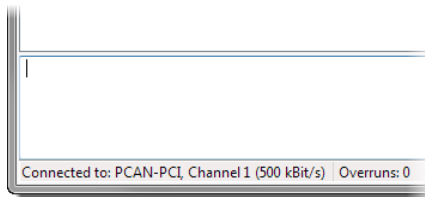
2. Switch on the converter by applying a voltage supply.

Due to the High level at the “Boot CAN” connection, the converter starts the CAN bootloader. This is indicated by the LED blinking orange.

3. Run the program `PcanFlash.exe` under Windows from the local hard drive.
4. Click on the (Options) button in order to call up the dialog box.
5. From the **Hardware Profile** dropdown list, select the **PCAN-RS-232** entry.

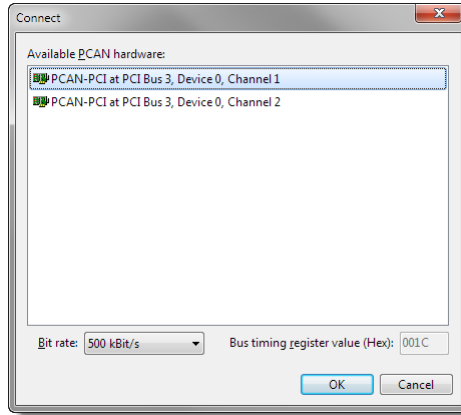


6. Click on the ... button next to the **File name** field in order to select the desired firmware file (*.bin) to be uploaded.
7. Click on the **OK** button.
8. Make sure that the PCAN-Flash program is connected with 500 kbit/s to the available CAN interface at the computer.



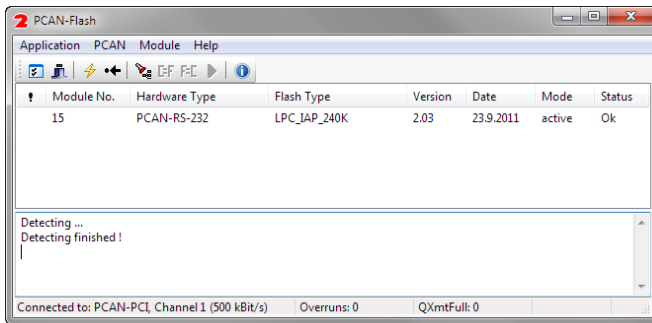
PCAN-Flash: Display of a connection in the status bar on the bottom.

If not, click the ⚡ (Connect) button in order to change the selection in the according dialog box.



9. Click the 🔍 (Detect) button in order to detect the PCAN-RS-232 converter connected to the CAN bus.

An entry for the converter appears in the main window.



10. Select the entry for the PCAN-RS-232 converter.
11. Click the ▶ (Program) button in order to start uploading the new firmware to the converter.


Observe the status indication at the bottom of the window. The process was successful if the last message to appear is “Flashing of module(s) finished!”.


12. Disconnect the power supply from the converter.
13. At the converter, disconnect “Boot CAN” from “U_b”.

You can now use the PCAN-RS-232 converter with the new firmware.

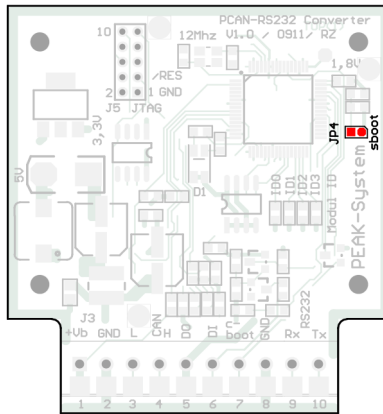
5.2 Uploading Firmware via the Serial Connections

This section shows how to activate the microcontroller’s boot-loader. The actual upload process depends on the upload software used which is supplied by a third party and is not described here.

 **Important note:** When uploading a firmware via the RS-232 interface, the CAN bootloader may be overwritten. Afterwards a firmware upload via CAN is not possible anymore.

 Do the following to activate the microcontroller's bootloader:

1. Switch the PCAN-RS-232 converter off by disconnecting it from the power supply.
2. Open the casing of the converter by removing the screws in order to gain access to the circuit board.
3. Shorten the jumper JP4 „sboot“, e.g. with a temporary solder connection.



4. Establish a serial connection to the computer or the programming adapter via the RS-232 connectors RxD (9) and TxD (10).
5. Switch on the PCAN-RS-232 converter by applying a voltage supply.

Due to the Low level on port P0.14 of the microcontroller, the converter starts the bootloader for serial transfer. The LED stays off.

6 Technical specifications

Functionality	
Microcontroller	NXP LPC2194/01 clocked with 60 MHz
Add-on memory	32 kByte, EEPROM Atmel AT24C32B (via I ² C)
CAN	High-speed CAN ISO 11898-2 Transceiver NXP TJA1040T Bit rates 40 kbit/s - 1 Mbit/s (lower bit rates on request) No termination
RS-232	RxD and TxD serial connections with RS-232 levels
Digital input (Din)	Low-active, max. level U _b
Digital output (Dout)	Low-side switch, max. 60 V/0.7 A
Status indication	Duo LED
Connectors	Screw terminal block, 10-pin, pitch 3.5 mm (Phoenix Contact MC 1,5/10-ST-3,5 - 1840447)

Power supply	
Supply voltage (U _b)	8 - 30 V DC
Current consumption	max. 70 mA at 12 V

Measures	
Size	Casing: 68 x 57 x 21 mm (W x D x H) Circuit board: 51 x 54 mm (W x D) See also dimension drawings in Appendix A on page 25
Weight	36 g

Environment	
Operating temperature	-40 - +85 °C (-40 - +185 °F)
Temperature for storage and transport	-40 - +100 °C (-40 - +212 °F)
Relative humidity	15 - 90 %, not condensing
EMC	EN 61326-1:2006-10 EC directive 2004/108/EG
Ingress protection (IEC 60529)	IP20

Appendix A CE Certificate

PCAN-RS-232 IPEH-002100 – EC Declaration of Conformity
PEAK-System Technik GmbH



Notes on the CE Symbol

The following applies to the PCAN-RS-232 product
IPEH-002100

EC Directive

This product fulfills the requirements of EC directive
2004/108/EG on "Electromagnetic Compatibility" and is
designed for the following fields of application as per the
CE marking:

Electromagnetic Immunity/Emission

DIN EN 61326-1; publication date: 2008-06
Electrical equipment for measurement, control and laboratory use – EMC requirements –
Part 1: General requirements (IEC 61326-1:2005);
German version EN 61326-1:2006-10

Declarations of Conformity

In accordance with the above mentioned EU directives,
the EC declarations of conformity and the associated
documentation are held at the disposal of the competent
authorities at the address below:

PEAK-System Technik GmbH

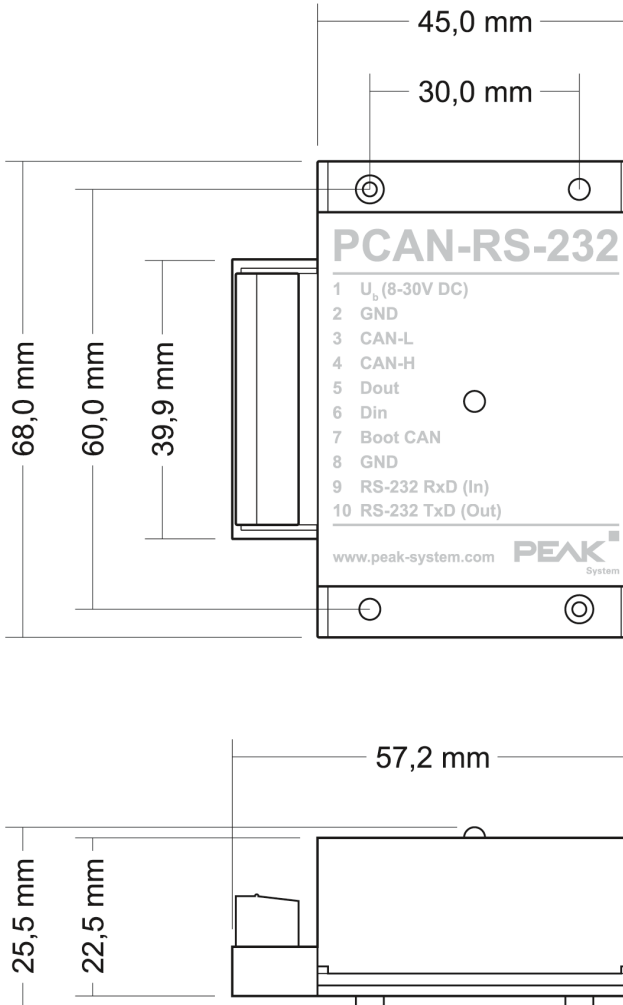
Mr. Wilhelm
Otto-Roehm-Strasse 69
64293 Darmstadt
Germany

Phone: +49 (0)6151 8173-20
Fax: +49 (0)6151 8173-29
info@peak-system.com

A handwritten signature in black ink, appearing to read "U. Wilhelm".

Signed this 15th day of February 2012

Appendix B Dimension Drawings



The figures do not show the actual size of the product.

Appendix C Port Assignment of the Microcontroller

The following table lists the used inputs and outputs (ports) of the LPC2194/01 microcontroller (μ C) and their function in the PCAN-RS-232 converter. It is meant as supplemental information. The converter's functionality is implemented by the supplied library.

Note: Get additional information about the LPC2194/01 microcontroller on the homepage of NXP (www.nxp.com).

Port	I/O	μ C function	Signal	Active (μ C)	Function/connection ¹
P0.0	O	TxD UART0	TxD0		Serial communication, Transmit, STB:10 (RS-232 levels)
P0.1	I	RxD UART0	RxD0		Serial communication, Receive, STB:9 (RS-232 levels)
P0.2	I, O	SCL	SCL		I ² C bus to the Atmel AT24C256B EEPROM
P0.3	I, O	SDA	SDA		
P0.4	I	Port pin	ID0	High	Coding solder jumpers on board (ID 0 - 3), bridged = Low
P0.5	I	Port pin	ID1	High	
P0.6	I	Port pin	ID2	High	
P0.7	I	Port pin	ID3	High	
P0.12	O	Port pin			Reserved
P0.13	I, O	Port pin			
P0.14	I	Port pin	/Boot_ser	Low	Activate flashing via serial interface, JP4

¹ STB:n Terminal n on the screw terminal block
 J4/5:n Pin n of the respective connector panel on the circuit board
 JPx Jumper position on the circuit board (set = active)

Port	I/O	µC function	Signal	Active (µC)	Function/connection ¹
P0.15	I	Port pin	/Boot_CAN	Low	Activate flashing via CAN with 500 kbit/s, STB:7 (due to wiring High-active)
P0.17	O	Port pin	V24_en	High	Deactivate the RS-232 component by Low level (activated by default); possibility for energy saving
P0.19	I	Port pin	Switch	High	Digital input Din, STB:6 (due to wiring Low-active)
P0.22	O	Port pin	CAN_en_1	Low	Activate the CAN transceiver ²
P0.25	I	RD1	CAN_RxD		CAN receive
TD1	O	TD1	CAN_TxD		CAN transmit
P0.28	I	Analog input	V-Power1		Measure voltage U_b , maximum value (0x03FF) corresponds to 33.1 V
P0.29	I	Analog input			Lies on GND
P0.30	I	Analog input			Lies on 1.8 V (microcontroller supply)
P1.16	O ³	Port pin		Low	LED red
P1.17	O ³	Port pin		Low	LED green
P1.21	O	Port pin		Low	Digital output Dout, STB:5 (Low-side switch)
P1.26		JTAG interface	RTCK		Debugging, J5:9
P1.27		JTAG interface	TDO		Debugging, J5:7
P1.28		JTAG interface	TDI		Debugging, J5:8
P1.29		JTAG interface	TCK		Debugging, J5:5
P1.30		JTAG interface	TMS		Debugging, J5:6
P1.31		JTAG interface	TRST		Debugging, J5:10

² After resetting the microcontroller, the CAN transceiver is deactivated and must be reactivated to use it.

³ It may occur that the LED glows slightly when the output is inactive. If you would like to prevent this, your firmware must change the port type to input (I). Before switching on the LED again, the respective port type must be set to output (O).