PCAN-RS-232

Programmable Converter
CAN to RS-232

User Manual

Document version 1.4.0 (2020-11-04)

Relevant Products

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

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 1.4.0 (2020-11-04)

Contents

1 Introduction

PCAN-RS-232 is a programmable module for the communication between RS-232 and CAN. The conversion of data traffic is done via a NXP LPC21 series microcontroller.

The behavior of the PCAN-RS-232 can be programmed freely for specific applications. The firmware is created using the included development package with GNU compiler for C and C++ and is then transferred to the module via CAN. Various programming examples facilitate the implementation of own solutions.

On delivery the PCAN-RS-232 is provided with a demo 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 as an example in the scope of supply.

1.1 Properties at a Glance

- NXP LPC21 series microcontroller (16/32-bit ARM CPU)
- 32 kbyte EEPROM
- High-speed CAN channel (ISO 11898-2) with bit rates from 40 kbit/s up to 1 Mbit/s (lower bit rates on request)
- Complies with CAN specifications 2.0 A/B
- Data transfer between CAN and RS-232 with a maximum bit rate of 115,200 bit/s
- One digital input and one digital output (low-active)

- 2-color LED for status signaling
- Connection via a 10-pole terminal strip[1]
- Voltage supply from 8 to 30 V
- Extended operating temperature range from -40 to 85 °C (-40 to 185 °F)
- New firmware can be loaded via CAN interface

1.2 Scope of Supply

- PCAN-RS-232 in plastic casing including mating connector
- Windows development package with GCC ARM Embedded, flash program, and programming examples
- Library with programming examples
- Manual in PDF format

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 10 or 8.1 (32/64-bit)

[1] 10-pin, 3.5 mm pitch, mating connector Phoenix Contact MC 1.5/10-ST-3.5 - 1840447

2 Connectors and Coding Solder Jumpers

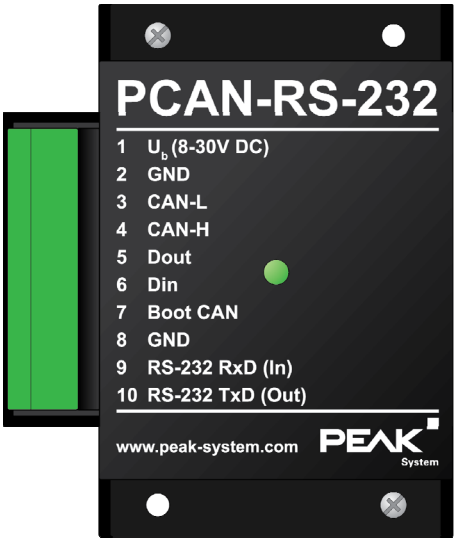


Figure 1: Top view of the housing

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

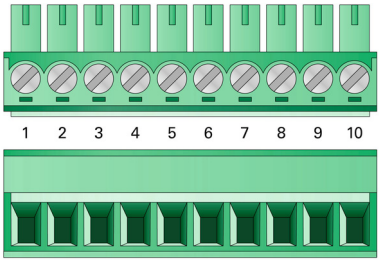


Figure 2: Screw terminal block;
mating connector Phoenix Contact MC 1.5/10-ST-3.5 - 1840447

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 29.

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.

- Do the following to get access to the unpopulated connector panel J5:

Danger of short circuit! Take great care when soldering to avoid short circuits.

Attention! Electrostatic discharge (ESD) can damage or destroy components on the card. Take precautions to avoid ESD.

1. Disconnect the device from the power supply.
2. Unscrew the two screws.
3. Remove the housing cover.
4. Make your desired settings.

   Figure 3 shows the positions of the JTAG panel (not equipped) on the top of board. The table below contains information of the microcontroller and internal wiring.

   Figure 4 shows the positions for soldering out the pull-down resistors on the bottom of the board. You can do that if the constant internal pull-down wiring of TCK or RTCK signals is not suitable for your purposes.

5. Put the housing cover back in place.
6. Screw the two screws back into their original positions.

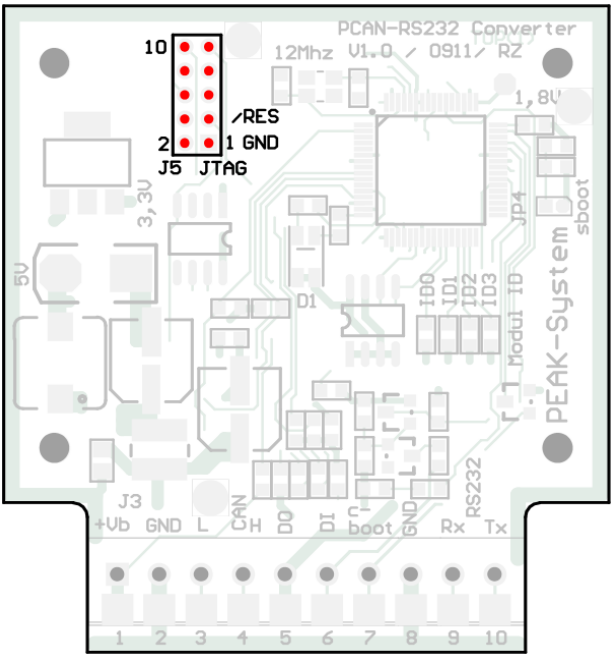


Figure 3: JTAG panel on the circuit board (not equipped)

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

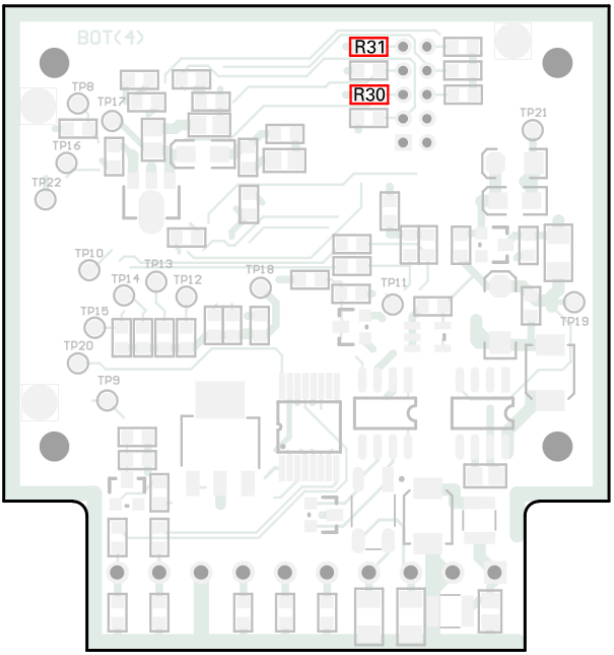


Figure 4: Pull-down resistors on the circuit board bottom:
R30 for pin 5 TCK, R31 for pin 9 RTCK

2.3 Coding Solder Jumpers

The board has four coding solder jumpers to assign a permanent state to the corresponding input bits of the microcontroller. The four positions for coding solder bridges (ID 0 - 3) are each assigned to one port of the microcontroller LPC2194/01 (µC). A bit is set (1) if the corresponding solder field is open.

A concrete application is the identification of a PCAN-RS-232 on the CAN bus during a firmware upload, especially if several devices are connected and in operation.

➧ Do the following to activate the coding solder jumpers:

Danger of short circuit! Take great care when soldering to avoid short circuits.

Attention! Electrostatic discharge (ESD) can damage or destroy components on the card. Take precautions to avoid ESD.

1. Disconnect the device from the power supply.
2. Unscrew the two screws.
3. Remove the housing cover.
4. Solder the solder bridge(s) on the board according to the desired setting.

   Figure 5 shows the positions of the solder fields on the board. The table below contains the possible settings.
5. Put the housing cover back in place.
6. Screw the two screws back into their original positions.

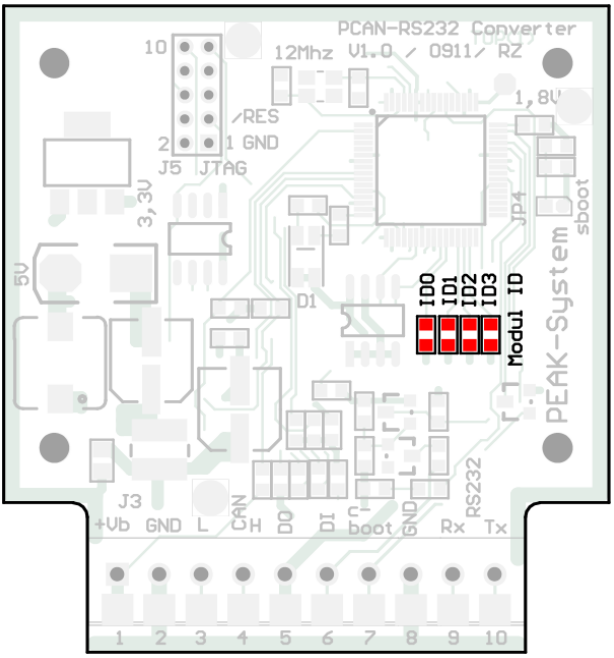


Figure 5: Coding solder jumpers on the circuit board

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

Solder Field Status	Port Status
	Low
	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 18.

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 6). 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.

The documentation for the example firmware can be found in the development package in the following directory branch:

`Hardware\PCAN-RS-232\Examples\06_CAN_TO_SER_BY_COMMAND\help`

The development package can be downloaded via this link:

`www.peak-system.com/quick/DLP-DevPack`

4 Creating Own Firmware

With the help of the development package, you can program your own application-specific firmware for PEAK-System programmable hardware products.

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.

Download of the development package:

URL: www.peak-system.com/quick/DLP-DevPack

System Requirements:

- PC with Windows® 10 (32-/64-bit)
- CAN interface of the PCAN series to upload the firmware to your hardware via CAN

Content of the package:

- `Build Tools\`
  Tools for automating the build process
- `Compiler\`
  Compilers for the supported programmable products
- `Hardware\`
  Contains sub directories of the supported hardware which include several firmware examples. Use the examples for starting your own firmware development.
- `PEAK-Flash\`
  Windows tool for uploading the firmware to your hardware via CAN. Copy the directory to your PC and start the software without further installation.

- `LiesMich.txt` and `ReadMe.txt`
- `SetPath_for_VSCode.vbs`
  VBScript to modify the example directories for the Visual Studio Code IDE.

Do the following to create your own firmware:

1. Create a folder on your local PC. We recommend using a local drive.
2. Copy the complete unzipped `PEAK-DevPack` directories into your folder, incl. all subs.

   No installation is required at all.
3. Run the script `SetPath_for_VSCode.vbs`. This script will modify the example directories for the Visual Studio Code IDE (https://code.visualstudio.com/).

   After that every example directory has a folder called `.vscode` containing the needed files with your local path information.
4. Now you can start Visual Studio Code which is available for free from Microsoft.
5. Select the folder of your project and open it.

   For example: `d:\PEAK-DevPack\Hardware\PCAN-RS-232\Examples\01_CAN_ECHO`
6. You can edit the C code and call `make clean`, `make all`, or compile single file via the menu **Terminal > Run Task**.
7. Create your firmware with `Make All`.

The firmware is the `*.bin` in the sub directory `out` of your project folder.

4.1 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. Die library is documented in the header files (`*.h`) which are located in the `inc` subdirectory of each example directory.

5 Firmware Upload

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

- **Via CAN (recommended):**
  Via a CAN channel and the Windows software PEAK-Flash, the firmware can be transferred from the computer to the PCAN-RS-232. More information can be found in the following subchapters.
- Via RS-232 interface or via serial connections of the microcontroller. For more information, see section 5.2 *Uploading Firmware via the Serial Connections* on page 23.

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 10 and 8.1 (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

For an upload of new firmware via CAN, the CAN bootloader must be activated in the PCAN-RS-232.

Do the following to prepare the hardware:

Danger of short circuit! Take great care when soldering to avoid short circuits.

Attention! Electrostatic discharge (ESD) can damage or destroy components on the card. Take precautions to avoid ESD.

1. Disconnect the device from the power supply.
2. Establish a connection between the terminals **Boot** and the power supply $\mathbf{U}_b$.

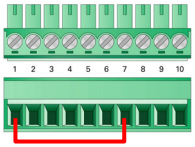


Figure 6: Connection at the screw terminal block between terminals 1 and 7

   This measure later applies the **Boot** 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).
4. Switch on the PCAN-RS-232 by applying a supply voltage.

   Due to the High level at the **Boot** connection, the PCAN-RS-232 starts the CAN bootloader. This can be determined by orange quickly blinking of the status LED.

5.1.3 Transfer Firmware

The microcontroller in the PCAN-RS-232 is equipped with new firmware via CAN. The firmware is uploaded via a CAN bus using the supplied Windows program PEAK-Flash.

Do the following to transfer a new firmware with PEAK-Flash:

1. The software PEAK-Flash is included in the development package, which can be downloaded via the following link: `www.peak-system.com/quick/DLP-DevPack`
2. Open the zip file and extract it to your local storage medium.
3. Run the `PEAK-Flash.exe`

   The program opens.

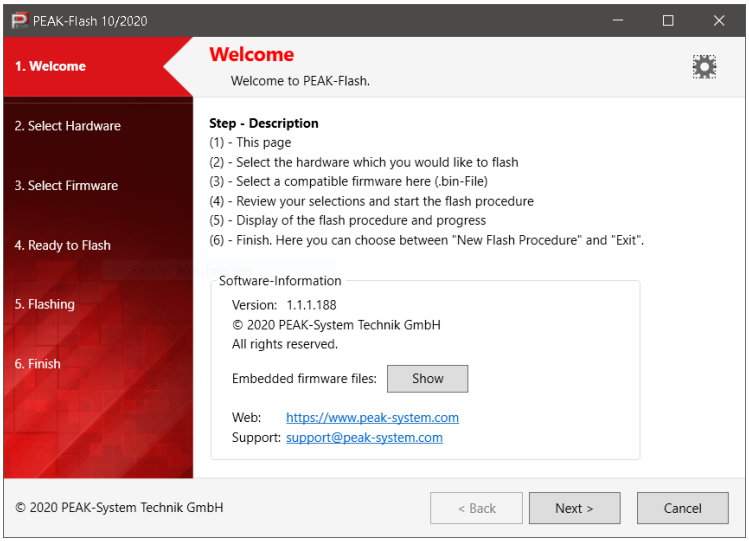


Figure 7: Main window of PEAK-Flash

4. Click the **Next** button.

5. Click on the **Modules connected to the CAN bus** radio button.
6. In the **Channels of connected CAN hardware** drop-down menu, select a CAN interface connected to the computer (e.g. PCAN-USB).
7. In the **Bit rate** drop-down menu, select the nominal bit rate available on the CAN bus.

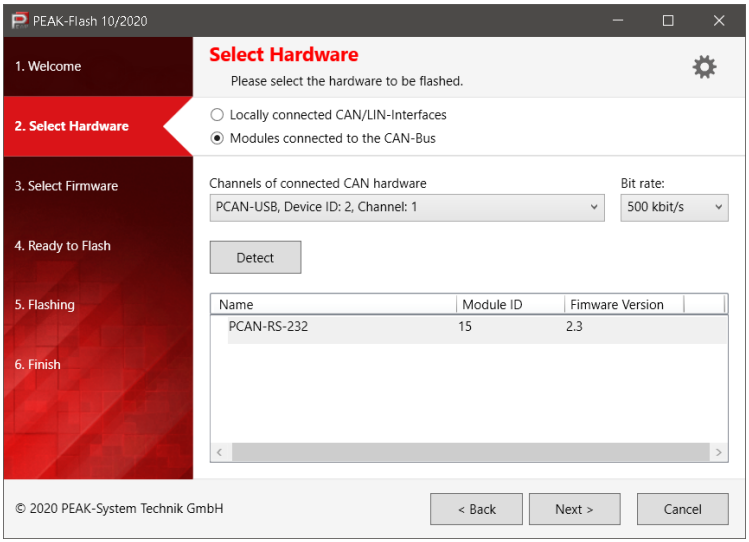


Figure 8: Hardware selection

8. Click on **Detect**.

   In the list, the **PCAN-RS-232** appears together with the **Module ID** and **Firmware version**. If not, check whether a proper connection to the CAN bus with the appropriate nominal bit rate exists.
9. Click **Next**.
10. Select the **Firmware File** radio button and click **Select**.

11. Select the corresponding file (`*.bin`).

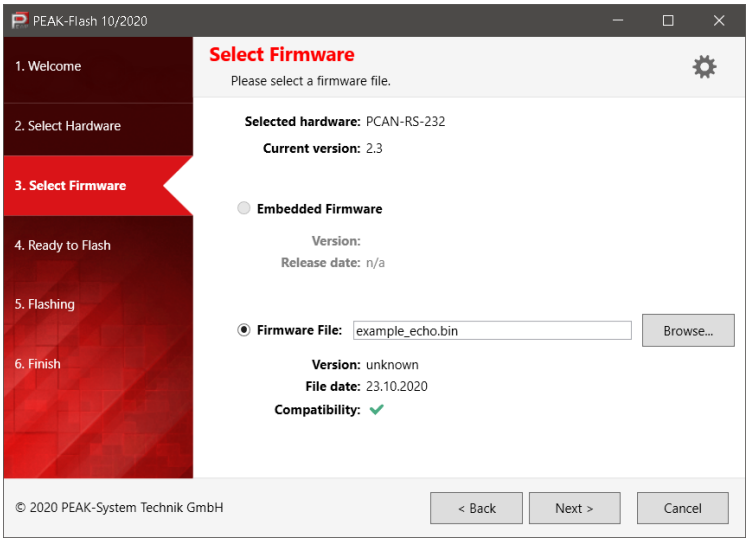


Figure 9: Selection of the firmware file (`*.bin`)

12. Click **Next**.

    The **Ready to Flash** dialog appears.

13. Click **Start** to transfer the new firmware to the PCAN-RS-232.

    The **Flashing** dialog appears.

14. After the process is complete, click **Next**.
15. You can exit the program.
16. Disconnect the device from the power supply.
17. Remove the connection between **Boot** and the power supply $\mathbf{U_b}$.
18. Connect the device to the power supply.

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

5.2 Uploading Firmware via the Serial Connections

This section shows how to activate the microcontroller's bootloader. 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:

Danger of short circuit! Take great care when soldering to avoid short circuits.

Attention! Electrostatic discharge (ESD) can damage or destroy components on the card. Take precautions to avoid ESD.

1. Disconnect the device from the power supply.
2. Unscrew the two screws.
3. Remove the housing cover.
4. Solder the solder bridge **JP4** on the board according to the desired setting.

   Figure 10 shows the position of the solder field on the board. The table below contains the possible settings.
5. Establish a serial connection to the computer or the programming adapter via the RS-232 terminals **RxD (9)** and **TxD (10)**.
6. Reconnect the power supply.

   Due to the low level at **port P0.14** of the microcontroller the converter starts the bootloader for the serial transmission. The LED remains off.

7. Put the housing cover back in place.
8. Screw the two screws back into their original positions.

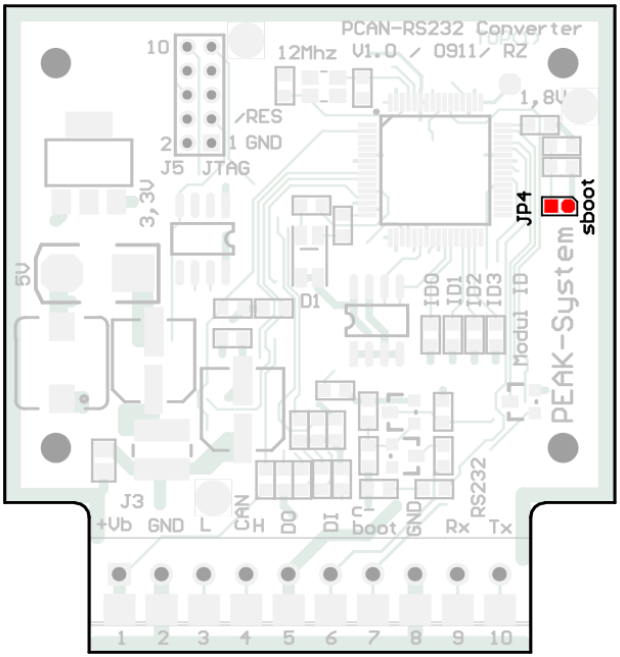


Figure 10: Activation of the bootloader by jumper JP4 on the board

Solder Field Status	Port Status
	Default: Bootloader is not activated.
	Bootloader is activated.

6 Technical Specifications

Functionality	
Microcontroller	NXP LPC2194/01 clocked with 60 MHz
Add-on memory	32 kByte, EEPROM Atmel AT24C32B (via I^2C)
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 27
Weight	36 g

Environment	
Operating temperature	-40 - +85 °C (-40 - +185 °F)
Temperature for storage and transport	-40 - +85 °C (-40 - +185 °F)
Relative humidity	15 - 90 %, not condensing
Ingress protection (IEC 60529)	IP20

Conformity	
EMV	EU Directive 2014/30/EU DIN EN 61326-1:2013-07; VDE 0843-20-1:2013-07
RoHS 2	EU Directive 2011/65/EU (RoHS 2) EU Directive 2015/863/EU (amended list of restricted substances) DIN EN IEC 63000:2019-05;VDE 0042-12:2019-05

Appendix A CE Certificate

EU Declaration of Conformity

This declaration applies to the following product:

Product name: **PCAN-RS-232**

Item number(s): **IPEH-002100**

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

CE We declare under our sole responsibility that the mentioned product is in conformity with the following directives and the affiliated harmonized standards:

EU Directive 2011/65/EU (RoHS 2) + 2015/863/EU (amended list of restricted substances)

DIN EN IEC 63000:2019-05;VDE 0042-12:2019-05
Technical documentation for the assessment of electrical and electronic products with respect to the restriction of hazardous substances (IEC 63000:2016); German version EN IEC 63000:2018

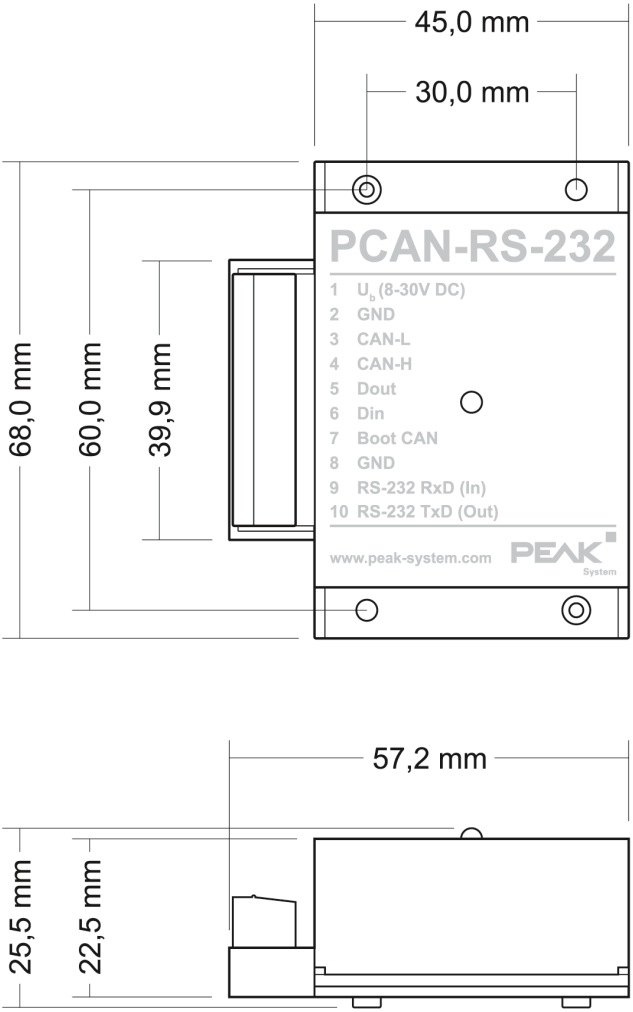
EU Directive 2014/30/EU (Electromagnetic Compatibility)

DIN EN 61326-1:2013-07;VDE 0843-20-1:2013-07
Electrical equipment for measurement, control and laboratory use - EMC requirements - Part 1: General requirements (IEC 61326-1:2012); German version EN 61326-1:2013

Darmstadt, 22 October 2020

Uwe Wilhelm, Managing Director

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 (µC) 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.

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

Port	I/O	µC Function	Signal	Active (µC)	Function/Connection[2]
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		I²C 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

[2] 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[2]
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[3]
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[4]	Port pin		Low	LED red
P1.17	O[4]	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

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

[4] 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).