PROPRIETARY NOTICE

All rights reserved by Janz Tec AG.
No parts of this technical manual may be modified, copied, or reproduced in any form or by any means for commercial use without the prior written permission of Janz Tec AG, Germany.
All instructions, information and specification contained in this manual are for reference only and remain subject to change without announcement.

emPC-A/RPI4

embedded PC

(Hardware and Software Manual)

Version 1.0

refers to product revision no.
1.1

Title:	emPC-A/RPI4 embedded PC
File:	j:\as\entwicklung\projekte\hw\empc\empc-arpi4\doc\manual_empc-arpi4.doc
Pattern:	\\ntserv5\janzgrp\as\entwicklung\formulare\entwicklung (ew)\ew_manual.dot
Created:	Stefan Althöfer, 06.03.2015
Last Update:	Stefan Althöfer, 04.12.2024

© Janz Tec AG 2024

Im Dörener Feld 8
D-33100 Paderborn, Germany

Tel.:	+49-5251-1550-0
FAX:	+49-5251-1550-190
email:	support@janztec.com
Internet:	www.janztec.com

Contents

List of Figures

List of Tables

© **Janz Tec AG**

About this Manual

This is the hardware manual for the emPC-A/RPI4 embedded PC.

Conventions

If numbers are specified in this manual, they will be either decimal or hexadecimal. We use C-notation to identify hexadecimal numbers (the 0x prefix).

If we refer to low active signal names, they will be suffixed by a "#" character.

Some parts of the manual contain notices you have to observe to ensure your personal safety, or to prevent damage to property. These are visually marked with the following alert symbols:

DANGER

Indicates that death or severe personal injury *will* result if proper precautions are not taken.

WARNING

Indicates that death or severe personal injury *may* result if proper precautions are not taken.

CAUTION

Indicates that *minor* personal injury can result if proper precautions are not taken.

NOTICE

Indicates that damage to equipment can result if proper precautions are not taken.

Indicates information that we think you should have read to save your time by avoiding common problems. Important suggestions that should be followed will also be marked with this sign.

Acronyms and Abbreviations

EMC	Electromagnetic capability.
ESD	Electrostatic discharge.
GND	System ground potential. Inside the product this is connected to the metal housing, which might be connected to protective earth by the installation. There exist some isolated reference grounds for communication interfaces or IO. These reference signals are referred to as GND-x, where x indicates function.
SELV	Safety extra low voltage.

1 Introduction

1.1 Features

- Raspberry Compute Module 4
  - BCM2711 quad core Cortex-A72 processor @ 1.5 GHz
  - Up to 8 GB SDRAM with ECC
  - Optional WLAN/Bluetooth with external antenna
- 2 x USB 2.0
- HDMI graphics output
- Storage options (exclusive-or)
  - Micro SDcard socket
  - eMMC (up to 32 GB)
- 10/100/1000 Mbit/s Ethernet, supporting IEEEE 1588
- CAN-FD, ISO/DIN 11898-2 (isolated from logic), jumper selectable 120 Ohm termination resistor
- Console, RS232 (2wire) on 9 pin DSUB
- Serial port, jumper selectable RS232 (4 wire) or RS485, jumper selectable 120 Ohm termination resistor for RS485
- RTC with battery backup
- Integrated TPM chip
- Power management MCU
  - supply supervision (e.g. adjustable undervoltage lockout, VIN and VS readout)
  - system control (on/off control by external input, with selectable delays)
  - watchdog function (restart by power supply toggle)
- 4 x digital OUT (0.5 A max.), 10 .. 34 V externally powered (isolated from logic)
- 4 x digital IN (isolated from logic)
- Internal (proprietary) expansion connector with PCIe, USB, GPIOs
- Optionally: Cellular Modem with external antenna(s)

1.2 Ordering Information

Oder Number	Features
SY-EPC-RPI420	• 2 GB RAM • uSDcard socket (no EMMC) • WLAN/BT • WLAN/BT antenna included in delivery • No LTE

SY-EPC-RPI42L	• 2 GB RAM
	• uSDcard socket (no EMMC)
	• WLAN/BT
	  • WLAN/BT antenna included in delivery
	• LTE CAT1 Quectel EG915NEUAG-N06-SGNSA
	  • LTE antenna included in delivery
	  • EGSM900, DCS 1800
	  • LTE Band 1,3,7,8,20
	  • GNSS (GPS/GLONASS/Galileo/BDS/QZSS/SBAS)
	  • Certifications: EU only

SY-EPC-RPI44L	Same as SY-EPC-RPI42L, but with 4 GB RAM

1.3 Functional Overview

The processing core of the emPC-A/RPI4 is the Raspberry Pi Compute Module 4. A Janz Tec specific baseboard implements the specific features. Refer to figure 1 for the block diagram.

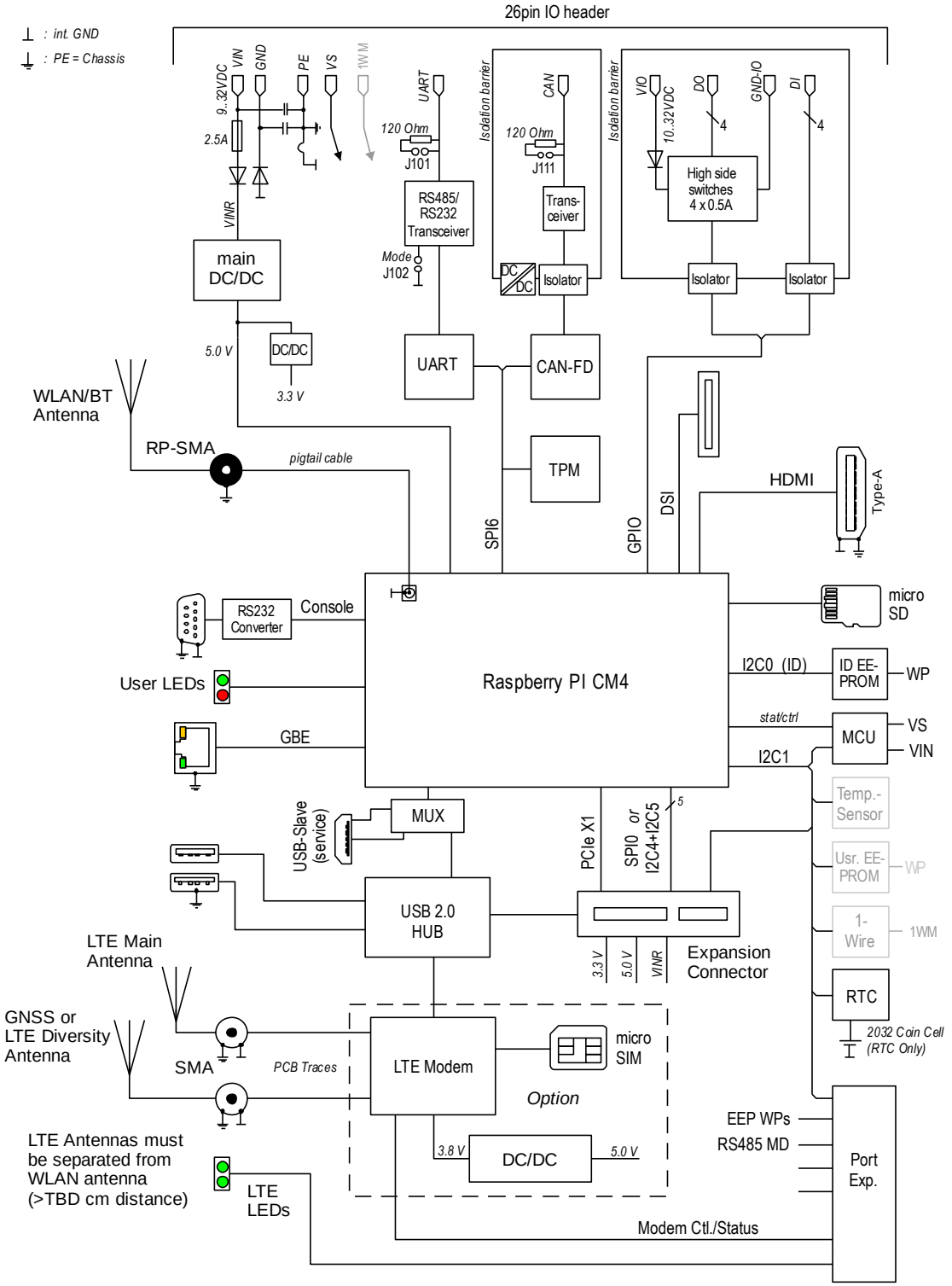


figure 1: emPC-A/RPI4 block diagram

2 Safety Instructions

Refer to page iv for explanation of the warning notice system.

The product described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products.

2.1 Installation and Maintenance

DANGER: Electrical Shock

Danger to life.
This product operates with 9..34 V DC SELV power supply. Do not connect this product to an improper power supply (No AC power, no more than 34 V DC/DC or no non-SELV circuit)!

DANGER: Electrical Shock

Danger to life.
The IO interfaces (connectors) of the product are only suited to be connected to SELV circuits. Use interfaces (connectors) for their intended use only.

CAUTION: Explosive Risk

The installed computer board is equipped with a Lithium battery.
Danger of explosion if battery is incorrectly replaced. Replace only with battery of the same or equivalent type.

CAUTION: Fire Risk

The digital IO must be powered by a SELV power supply, that complies with the requirements of a limited energy source (LPS) (By using an appropriate supply or an external fuse).

WARNING: Burns Hazard

The product generates considerable amount of heat. The housing transports this heat to the environment and therefore gets hot. **Caution when touching the housing, burns hazard**!

WARNING: Health Impact

The antennas of this product must not be installed closer than 20 cm (0,2 m) to human bodies.

Only connect approved antennas (such as delivered with the product) to the product.

When using WLAN, make sure you have configured your regional settings so that you comply with local regulations. Refer to the software section for more information.

2.2 Ambient and Environmental Conditions

CAUTION: Damage

Do not operate the product beyond the specified ambient conditions

DANGER: Explosive Risk

Do not operate the product in potentially explosive atmosphere.

NOTICE: EMI

This product is a class A device. This product may cause radio interference. In this case the user must take adequate measures.

3 Intended Use

The emPC-A/RPI4 is designed for computing purposes in industrial environments. It is destined to be used indoor only.

3.1 Disposal and Recycling

Janz Tec products are manufactured to satisfy environmental protection requirements where possible. Some of the components used are capable of being recycled. Final disposal of this product after its service life must be accomplished in accordance with applicable country-, state-, or local-laws or -regulations.

This product contains a lithium battery. This should be removed before disposal and be returned separately

Don’t forget to delete all sensitive data on the product before disposal. Especially delete all person related data.

3.1.1 EU

Janz Tec electronic products are labelled with the crossed-out trash can. This means that the products **must not** be disposed into the domestic garbage.

Used Janz Tec products have to be recycled properly. If in doubt, you can return them to Janz Tec at your own expense. Refer to https://www.janztec.com/recycling or contact Janz Tec under recycling@janztec.com for more instructions how and where to return the products.

3.1.2 Germany

All Janz Tec products are registered as B2B custom at the german EAR. Hence Janz Tec products **must not** be disposed to public collection points for used electronic equipment. Refer to §14 of General Terms & Conditions of Janz Tec AG for the details regarding the mutual obligations as to the disposal of Janz Tec products.

The Lithium battery included in the product is registered at “GRS Batterien” and can be disposed to public collecting points for used batteries.

As stated above, used Janz Tec products can also be returned to Janz Tec at you own expense for free-of-charge recycling.

© **Janz Tec AG**

4 Installation

WARNING: Burns Hazard

The product generates considerable amount of heat. The housing transports this heat to the environment and therefore gets hot. **Caution when touching the housing, burns hazard**!

The product can be operated with DC power supply from 9 to 34 V.**32**

4.1 Mounting

The emPC-A/RPI4 is intended for 35mm DIN-rail mount. It can as well be used on the desktop. Refer to figure 2 for the recommended mounting orientation.

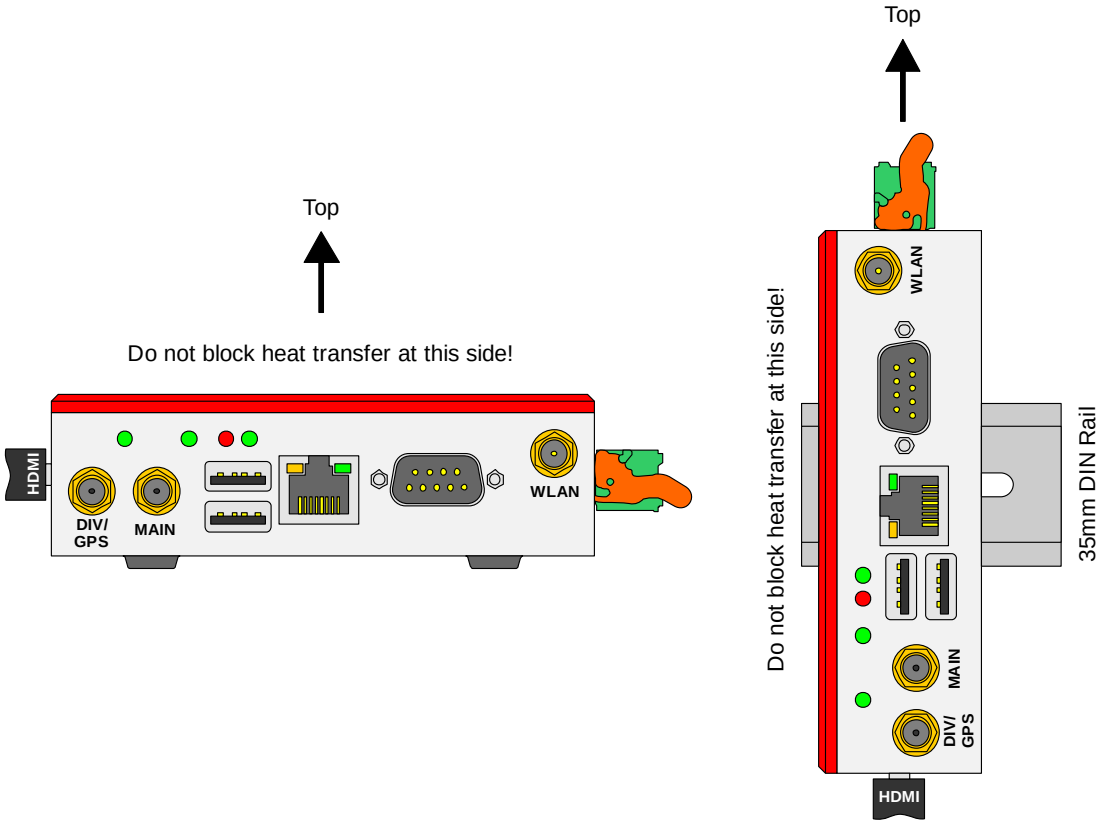


figure 2: emPC-A/RPI4 mounting options: Desk-top and DIN Rail

For desk top use, you should apply the rubber feat to avoid scratching your desk top surface.

4.2 Connectors and Operators

The 24 pin multi connector provides interface to many of the emPC-A/RPI4 interfaces that are described later.

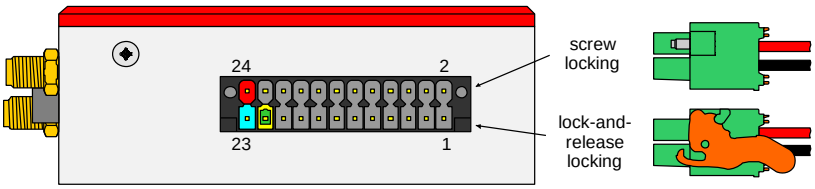


figure 3: Multi connector for IO (Power Supply Pins highlighted)

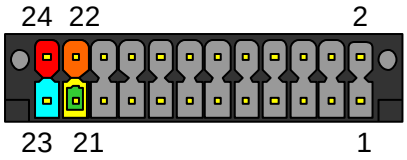
A suitable mating connector with lock-and-release levers is Phoenix Contact order number 1790580. A Model witch screw locking is available with order number 1790399.

The plug uses spring-cage connection and accepts solid or flexible wires from 0.2 mm² to 1.5 mm². Push in connection is possible, in case of flexible wires you must use ferrules to allow push in. If you use ferrules with plastic sleeve, the maximum wire size reduces to 0.75 mm².

Stripping length for the wires is 10 mm.
If you use ferrules, obey that standard ferrules for 0.75 mm² have only 7 mm length.

4.2.1 POWER IN

The system power supply is located on the multi connector.



21	PE	22	VS
23	GND	24	VIN (9..34 VDC)

table 1: Power Supply connector pins

DANGER

The product may only be operated with power supplies which can be considered SELV circuits.

Pins 21 should be connected to EMI protective earth (PE). These contacts are internally connected to the DC power supply EMI filter and to the housing. In locations where no EMI noise is present, the PE connection is not required. The PE has no safety funcion. When the DIN Rail is grounded , the PE contact at Pin 21 is redundant.

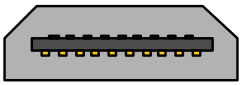
Pin 22 has changed regarding to the emPC-A/RPI3!
ARPI4: When you connect PE to pin 22 only, it will not provide the expected EMI countermeasure. When you connect PE to pin 21 and 22 no issue will arise.
ARPI3: When you use the same cable as for ARPI4, you might damage the VS source.

VS is an analog input signal to the MCU, intended for monitoring and state control. Input voltage range is the same as for VIN.

© **Janz Tec AG**

4.2.2 Graphics connector (HDMI)

A single HDMI type A (Standard Size) connector is available to attach monitors.

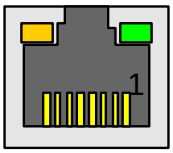


1	Data2+	11	GND
2	GND	12	Clock-
3	Data2-	13	CEC
4	Data1+	14	N.C.
5	GND	15	
6	Data1-	16	SCL
7	Data0+	17	SDA
8	GND	18	+5V
9	Data0-	19	HOTPLUG
10	Clock+		

table 2: HDMI Connector

4.2.3 Ethernet Interface

Ethernet interfaces of the Raspberry PI. The Ethernet physics is 10/100/1000BaseT, available through the shielded modular jack. Twisted pair cable can be used to connect to this port.

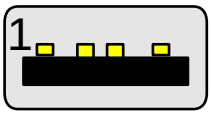


1	MDI0+	5	MDI2+
2	MDI0-	6	MDI2-
3	MDI1+	7	MDI3+
4	MDI1-	8	MDI3-

table 3: Ethernet connector

4.2.4 USB host interfaces (USB1-USB2)

Two USB type A host interfaces are available.



1	+5V, I_{MAX} = 500 mA, 750 mA for all ports
2	USB-
3	USB+
4	GND

table 4: USB host connector

NOTICE

Although each port can deliver supply current of 500 mA, the overall load on USB1 to USB4 interfaces must be limited to 750 mA to prevent power supply from overheating.

If you draw excessive power from USB ports or short circuit USB power supply outputs, then the Raspberry PI board will temporarily shut off the USB power supply and retry after some time.

4.2.5 Micro USB slave interface (service only)

A micro USB port is available for service purpose. It allows to configure the CM4 module when a blank eMMC or SDcard is installed, or to perform updates.

1	+5V, I_{MAX} = 500 mA, 750 mA for all ports
2	USB-
3	USB+
4	ID
5	GND

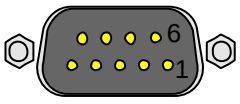
table 5: Micro USB slave connector

When the Micro USB connector is plugged in, the following happens:

- Micro USB data signals are routed to CM4 USB pins
  - All regular USB functions of the emPC-ARPI become immediately unavailable.
- CM4 module is strapped for USB boot (nRPIBOOT pulled low), which takes effect at the next power cycle (TBD: reset?)

4.2.6 Console (RS232)

The Console (RS232) serial interface is provided by Raspberry PI board directly. The connector is a standard 9 pin D-SUB plug, but the interface is connected with two wires only, table 6 shows the pin out.



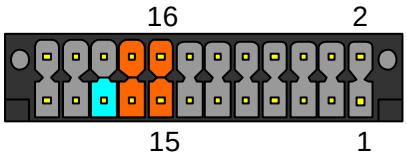
9pin male D-Sub

1	n.c.	6	n.c.
2	RxD	7	n.c.
3	TxD	8	n.c.
4	n.c.	9	n.c.
5	GND		

table 6: Console (RS232) connector

4.2.7 Serial Port (RS232/RS485)

The Serial Port is implemented by SC16IS740 UART. A multiprotocol transceiver is utilized to support both RS232 and RS485. The interface signals are located on the multi connector, refer to table 7.



15	TxD , D-	16	RTS , D+
17	RxD	18	CTS
19	GND	20	

table 7: Serial Port connector pins

RS232 mode is selected by shorting jumper J301, else RS485 mode is selected. The case has to be opened to access this jumper.

In RS232 mode, the signals TxD (out), RxD (in), RTS (out) and CTS (in) are provided.

In the RS485 mode the bidirectional differential signals D+ and D- are provided. When the transmitter is enabled and no data is send (1=MARK), then D+ > D- (refer to figure 4).

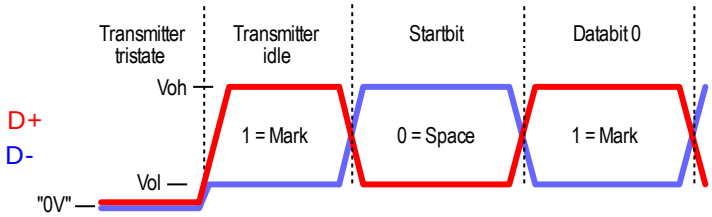


figure 4: RS485 signaling

Note that D+ corresponds to pin “A” of the transceiver chip. However this is signal B according to the EIA-485 or Profibus standards.

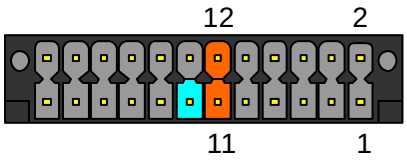
The receiver has the fail safe input feature, which ensures that no data is received when D+ and D- are floating or are shorted (e.g. by the termination resistor).

Refer to section 7 (Programming Information) for more information about RS485 control.

An internal 120 Ohm termination resistor can be enabled between D+ and D- by shorting jumper J300. The case has to be opened to access this jumper.

4.2.8 CAN

The CAN interface is implemented with a MCP2518FD controller chip. The interfaces is located on the multi connector, refer to table 8. The CAN interface signals support ISO 11898-2 (high speed) and are isolated from system logic, as result it is always required to connect GND-CAN (pin 13) to your CAN network reference ground.



11	CAN-H	12	CAN-L
13	GND-CAN	14	n.c.

table 8: CAN connector pins

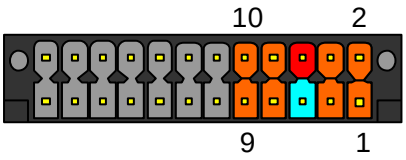
An internal 120 Ohm CAN termination resistor is enabled by shorted jumper J400. To disable the termination, the case has to be opened to remove this jumper.

CAUTION

Factory setting: Jumper J400 is **not** set by default, and therefore the 120 Ohm termination resistor is disabled!

4.2.9 Digital IO

Digital input (DI1-DI4) and output (DO1-DO4) ports are provided on the multi IO connector (refer to table 9).



1	DO4	2	DO3
3	DO2	4	DO1
5	GND-IO	6	VIO (10..34 VDC)
7	DI4	8	DI3
9	DI2	10	DI1

table 9: Digital IO connector pins

CAUTION

The digital IO must be powered by a SELV power supply, that complies with the requirements of a limited energy source (LPS) (By using an appropriate supply or an external fuse).

The digital inputs and outputs are isolated from system power supply. Refer to figure 4 for detailed schematic.

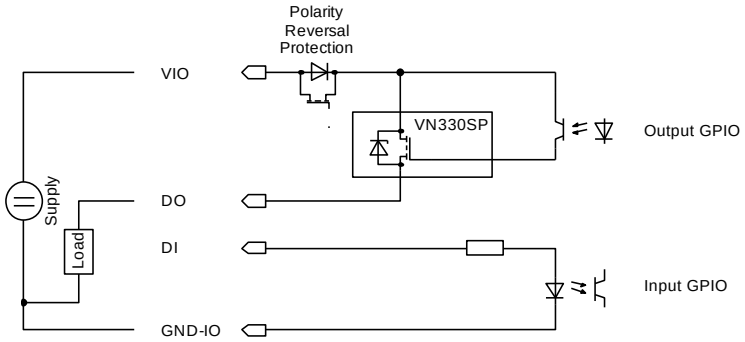


figure 5: digital IO detailed schematic

The digital input signals (DI1-DI4) refer to GND-IO and have an input impedance of about 5 kOhm. Switching level for *low* is ≤4V and *high* ≥9V. The digital inputs do not require VIO to be present.

The output driver supports resistive and inductive loads.

NOTICE

When using inductive loads, the output driver has internal clamping to demagnetize the load. This clamps the output pin to about 55 V below VIN. When using this feature, very carefully evaluate that the demagnetization energy does not overheat the output driver. Parameters to check are the switching frequency, the load current, the load inductance and VIO voltage. Refer to the VN330SP datasheet for more information. If in doubt, add external clamping circuits.

© **Janz Tec AG**

4.2.10 WLAN Antenna

If option is installed.

RP-SMA

1	Antenna Input/Output
2	GND (Return)

WARNING: Health Impact

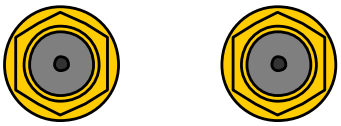
The antenna of this product must not be installed closer than 20 cm (0,2 m) to human bodies.

Only connect approved antennas (such as delivered with the product) to the RP-SMA connector.

When using WLAN, make sure you have configured your regional settings so that you comply with local regulations. Refer to the software section for more information.

4.2.11 LTE Antenna(s)

If option is installed.



SMA
Diversity/GPS

SMA
Main

1	Antenna Input/Output
2	GND (Return)

WARNING: Health Impact

The antenna of this product must not be installed closer than 20 cm (0,2 m) to human bodies.

Only connect approved antennas (such as delivered with the product) to the SMA connector.

5 Maintenance

NOTICE
Always follow common ESD practice when you service the product!

Task	TODO	Screws to remove
Replace µSD-card or SIM	Remove DIN rail clip	Blue marked
Change jumper settings	Open the case	Orange marked
Replace Battery	Open the case	Orange marked

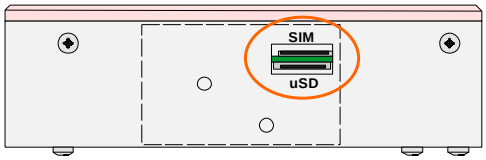


figure 6: emPC-A/RPI4 micro SDcard/SIM location

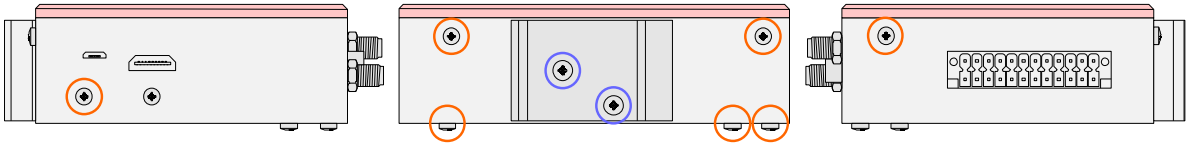


figure 7: emPC-A/RPI4 screws for case opening

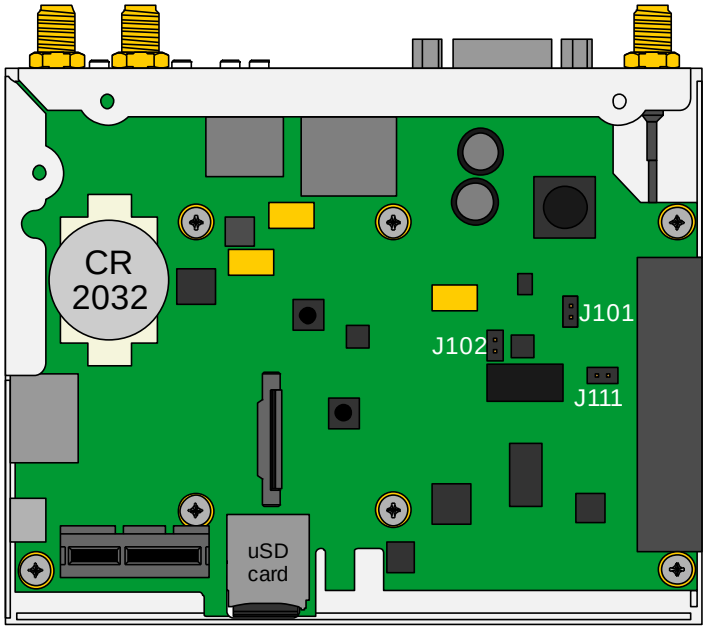


figure 8: emPC-A/RPI4 jumper and battery location on PCB
after case opening. Heatsink face down.

© **Janz Tec AG**

5.1 Micro SDcard Replacement

NOTICE
Power must be turned off before removing or inserting the micro SDcard

Refer to figure 6 to identify the location of the Micro SDcard. Remove the DIN rail clip if installed.

5.2 SIM card Replacement

Refer to figure 6 to identify the location of the Micro SDcard. Remove the DIN rail clip if installed.

5.3 Option Jumpers

Here is a list of user settable jumpers. Refer to figure 8 for the jumper location

Jumper	Default	Description
J101	Removed	Enable 120 Ohm termination on RS485 differential pair when set. Do not set this jumper when RS232 is selected.
J102	Set	Set: RS232 mode selected Remove: RS485 mode selected Software can read the jumper position an configures the UART accordingly.
J111	Removed	Enable 120 Ohm termination on CAN interface when set.

5.4 Battery Replacement

CAUTION

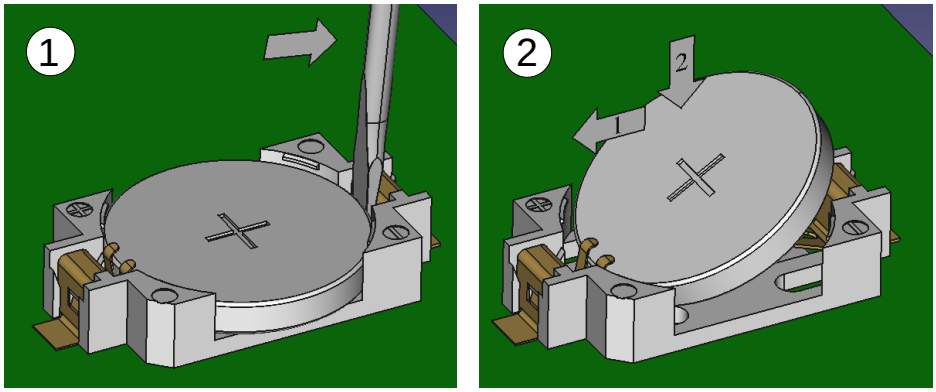
The installed computer board is equipped with a Lithium battery.
Danger of explosion if battery is incorrectly replaced. Replace only with battery of the same or equivalent type (3-volt lithium coin cell battery).

- Do not attempt to recharge the battery.
- Do not disassemble, crush, puncture, short external contacts, or dispose of in fire or water.

Compatible battery type: CR2032 (3 Volt lithium coin cell battery)

The battery is used for backing up the system time when the power supply is removed.

1. Turn off the computer properly through the operating system, then turn off any external devices.
2. Disconnect the power supply from the power inlet and disconnect any external devices.
3. Disassemble the housing (refer chapter 5) and locate the battery on the Janz Tec board.
4. Remove the battery from the holder (See figure 9)
5. Insert the new battery (See figure 9)
6. Re-assemble the housing
7. See chapter 6.2.2.4 for important software information



1. Removal: Insert screwdriver at right side and bend so that the battery pops outs. Use only gentle force, otherwise the battery holder might be damaged.
   The use of a plastic tool is preferred to avoid shorting the battery.
2. Insertion: Align new battery to the left side of the holder (below the hooks) and gently press down on the right side of the battery until the battery snaps into the holder.

figure 9: Removing and replacing coin cell battery

© **Janz Tec AG**

6 Operating System and Drivers

You can order the emPC-A/RPI4 with pre-installed software (Raspberry PI OS). In this case our default Janz Tec AG Linux image is pre-installed on the µSD-card and the emPC-A/RPI4 driver package is also pre-installed and fully configured. For more information, refer to 6.2.

Of course, you can order the emPC-A/RPI4 without software, so you can choose a Raspberry PI 4 compatible operating system of your choice and install it manually on a µSD card. (Raspberry PI OS recommended)
Refer to 6.1.1 on how to install an image from the official Raspberry Pi website.

6.1 Standard Raspbian Operating System

If the additional interfaces of the emPC-A/RPI4 (RTC, CAN and RS232/RS485) are not required, then the latest official Raspberry PI OS (32-bit) is recommend for emPC-A/RPI4 systems. (Digital IO and LEDs are controllable by standard GPIO commands and console (RS232) is supported by Raspbian by default). To use all of the emPC-A/RPI4 features, refer to chapter 6.2 and 6.3.

6.1.1 Installation

For more information on how to prepare a µSD-card to run the Raspberry PI OS visit:

https://www.raspberrypi.com/documentation/computers/getting-started.html#install-an-operating-system

Download:

https://www.raspberrypi.com/software/operating-systems/

Default log in credentials:

Username: **pi**
Password: **raspberry**

(depending on your keyboard layout language, the last “y” might be “z” on your keyboard)

For more information visit the official Raspberry quick start guide:

https://www.raspberrypi.com/documentation/computers/getting-started.html#set-up-your-raspberry-pi

6.2 emPC-A/RPI4 Driver Installation Script

NOTICE

Always create a backup of the µSD card before installing this script!

NOTICE

This script is only compatible to Raspbian based installations!

To automatically install all configuration entries and drivers to enable the additional features of the emPC-A/RPI4 (RTC, CAN, RS232, RS485, TPM and LTE), we provide an installation script and installation instructions available at:

https://github.com/janztec/empc-arpi4-linux-drivers

6.2.1 Installation Script

Prerequisites:
The installation script requires that the emPC-A/RPI4 has booted successfully from the Raspberry PI OS (32-bit) image on the µSD-card prepared in chapter 6.1.1, an active Internet Connection and at least 1GB free disk space. It is also recommended, if you have sensible data on the µSD-card, that you creat a backup copy of your µSD-card.

Installation Instructions:
Follow the instructions on https://github.com/janztec/empc-arpi4-linux-drivers to use our installation script to install all required drivers and settings automatically.

6.2.2 Driver Details

The following drivers are automatically installed by our driver installation script:

6.2.2.1 SPI driver

Default spi driver with three chip selects. CAN, RS232/RS485 controllers and TPM are connected by SPI to the Raspberry PI board.

6.2.2.2 sc16is7xx : Serial RS232/RS485 driver

By default, the emPC-A/RPI4 is configured to RS232 mode (Jumper J102 is set).

If you want to use this serial port in RS485 mode, open the case as described in chapter 5, and remove the jumper J102.

The jumper state is checked every system start up (/etc/rc.local) and if no jumper is detected, tool “tty-auto-rs485” switches the Linux device /dev/ttySC0 to RS485 mode. As a result, your application can use Linux device */dev/ttySC0* identically, regardless of configured RS232 or RS485 mode.

6.2.2.3 mcp251xfd : CAN driver

This driver installation includes a SocketCAN compatible driver. In the configuration file /etc/network/interfaces.d/can0.interface the CAN device with the Linux device name **can0** is enabled by default (bus-on) and is configured to 1Mbit (bitrate 1000000). To change the default bitrate, edit file /etc/network/interfaces.d/can0.interface, change the bitrate parameter and restart your system.

Here are some can-utils examples:

Print incoming can frames:

```
pi@raspberry ~$ candump can0
```

Send random can frames:

```
pi@raspberry ~$ cangen can0
```

Show can bus load information:

```
pi@raspberry ~$ canbusload can0@1000000 -r -t -b -c
```

6.2.2.4 rv8803: RTC

The emPC-A/RPI4 driver installation script installs the RTC service, so that the RTC automatically sets the system time on every boot during the initramfs execution.

Reading the RTC time:

```
pi@raspberry ~$ sudo hwclock -r
```

Writing the system time to the RTC

```
pi@raspberry ~$ sudo hwclock --systohc -D --noadjfile --utc
```

6.3 Janz Tec AG Linux Image for emPC-A/RPI4

Our pre-installed Janz Tec AG Image is optionally available and contains the combination of the official Raspberry PI OS as well as our emPC-A/RPI4 drivers. File /root/imageversion.txt contains the image name and version of the installed Janz Tec image.

You can start the *raspi-config* Raspberry PI configuration tool with “sudo raspi-config”, which offers important configuration settings in a convenient menu.

https://www.raspberrypi.com/documentation/computers/configuration.html

6.4 General Information

- To reduce the risk of losing information stored in files on the µSD-card, especially when writing to the card during a power failure event, consider the use of an external UPS (uninterruptable power supply). Alternatively, mount the µSD card as read-only, in combination with a RAM-file system overlay.

6.5 Optional Software

6.5.1 CODESYS Control for Raspberry Pi SL Information

NOTICE

Without further agreements, Janz Tec AG does not provide support for configuring and running CODESYS Control for Raspberry PI SL on the emPC-A/RPI4!

CODESYS control runtime system is a soft PLC created by the company CODESYS GmbH and is available for Raspberry Pi based devices. Using this IEC 61131-3 programming system in combination with our emPC-A/RPI4 system is ideal for testing and educational applications, because the control can make use of the additional interfaces the emPC-A/RPI4 offers (digital IO, CAN, RTC and RS232/RS485).

To download a time limited evaluation version, or buy a full license visit: **http://store.codesys.com/**

NOTICE

Always consult the official “CODESYS Control for Raspberry Pi SL” documentation, product description, features and licensing terms.

The following summary of features is only for your information and might not be accurate.

Even with the standard Raspberry PI OS without any modifications, the CODESYS Control for Raspberry Pi SL can control all digital IOs on the emPC-A/RPI4 by using the GPIO device already included in the CODESYS device repository. (Red and green LED also controllable by this GPIO device). The serial console (RS232) is also support by default. *2

When the optionally available emPC-A/RPI4 driver package is installed, the CODESYS control can also access the CAN port for CAN/CANopen as well as the serial console (RS232) and the RS232/RS485 ports for serial and Modbus RTU communications. The RTC support is especially useful when you do not have a network time server connected, but you want to use date and time in e.g. log files.

CODESYS Features	**Standard Raspberry PI OS**	**Raspberry PI OS +** emPC-A/RPI4 **driver package**
CAN/CANopen *1	not supported	supported (using SocketCAN)
Modbus RTU Master *2	console (RS232, 2wire)	console (RS232, 2wire) serial port RS232 (4wire) serial port RS485
GPIO device	digital IO (4in, 4out) red and green LEDs	digital IO (4in, 4out) red and green LEDs
Date / Time	requires a time server in the network	• RTC / system time • (network time server also possible)

*1: In the current version: 2.1.1.0 of the CODESYS control package, CAN/CANopen is included in the evaluation and paid version, but not mentioned in the product description
*2: May require manual configuration in the CODESYS configuration files

6.5.1.1 Information

If you are interested in CODESYS runtime systems with real-time support, more Ethernet ports, or more fieldbus ports, our embedded PC series emPC and panel PC series emVIEW offer the whole range of industrial PCs in many performance classes.

https://www.janztec.com/en/industrial-pcs-embedded-pcs/

6.5.1.2 License Information

NOTICE
We strongly recommend, that you create a backup copy of an activated license!

In case that the contents of the card will become corrupt, the license will be lost! We strongly recommend, that your first step is to create a backup copy of the activated license.

To create a backup copy of the license, copy all *.WibuCmRaU files from the Raspberry Pi (accessible in CODESYS via double-click on the device under the tab “Files”) to a safe external storage.

7 Programming Information

This chapter gives technical information on how to program, or adapt existing Linux drivers to the emPC-A/RPI4 hardware and is intended for users familiar with creating own Linux kernels and drivers.

For standard Linux users, optionally our emPC-A/RPI4 driver package (chapter 6.2) is available and this already includes all required drivers, therefore no driver programming is needed to use the emPC-A/RPI4 hardware and it's interfaces.

7.1 User LEDs

The front panel LEDs are controllable by the user or might be used to signal states as defined by the compute module.

LED	Signal
GREEN (RPI)	PI_nLED_Activity (CM4, Pin 21)
RED (RPI)	PI_LED_nPWR (CM4, Pin 95)
GREEN1 (LTE)	LED_LTE1 (I2C-Port Expander PA3)
GREEN2 (LTE)	LED_LTE2 (I2C-Port Expander PA4)

NOTICE

When the emPC-A/RPI4 driver package (chapter 6.2) has been installed, the green LED is configured to show the µSD-card activity.

7.2 I2C Bus

The emPC-A/RPI4 utilizes the Raspberry I2C buses to connect to several peripheral devices.

7.2.1 I2C-0 (ID)

An HAT-ID 24C32 EEROM is present to detect the features of the emPC-A/RPI4 baseboard. The address is strapped to 0x50.

7.2.2 I2C-1

Type	Device	Address	Notes
Port Expander	MCP23018	0x20	
RTC	RV-8803-C7	0x32	
Thermal Sensor	DS75	0x48	Optional
1-Wire	DS282-100	0x18	Optional
SO-8 EEPROM	TBD	0x50	Optional
Expansion-ID	AT24C32E	0x57	Added with Expansion
PMON MCU	Attiny	0x60	Pulse stretching!

table 10: Internal I²C devices

All devices support 400 kHz.

The MCU requires working pulse stretching. As the pulse stretching capability of the BCM2711 is broken, you must select bit-bang-I2C implementation.

7.3 SPI Bus

The emPC-A/RPI4 utilizes the CM4 SPI-6 bus to connect CAN, UART and TPM.

Type	Device	Chip Select	Interrupt	SPI CLK (max)
CAN-FD	MCP2518FD	CE0# (GPIO18)	GPIO25	17 MHz
UART	SC16IS760	CE1# (GPIO27)	GPIO26	15 MHz
TPM	SLM 9670AQ2.0	GPIO24	-	43 MHz

Limit the SPI clock rate to below 15 MHz for all Devices to work properly unless you can switch the clock rate dynamically. Speeds > 15 MHz were not tested.

7.4 CAN

The CAN controller is clocked by a 40 MHz crystal. CAN signals is isolated form digital logic. The transceiver supports bitrates up to 4 MHz.

7.5 Serial Port (RS232/RS485)

The UART is clocked by a 1.8432 MHz oscillator.

When in RS485 mode, the transmitter is controlled by the UART's RTS# signal. A low enables the transmitter.

The receiver is disabled whenever the transmitter is enabled, hence transmitted characters are not received. It is possible to modify the hardware such that the receiver is always enabled, to allow verification of transmitted characters. Refer to to figure 10 for the mode selection and receiver control details.

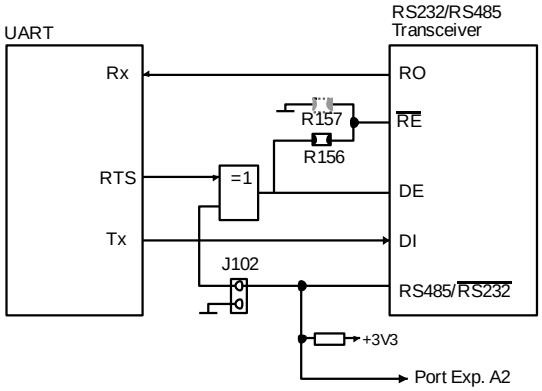


figure 10: RS232/RS485 mode switch and transceiver control

When the jumper is not set, it is even possible to control the RS232/RS485 mode selection by software. The drawback is however that the mode is possibly wrong from power on, until the time when the software is running. We recommend using the A2-Port as input to control the UART RS485 mode to follow the jumper setting.

7.6 Digital IO

The digital inputs and outputs are controlled by Raspberry PI GPIO signals.

IO	GPIO
DI1	GPIO4
DI2	GPIO5
DI3	GPIO6
DI4	GPIO7
DO1	GPIO12 (PWM capable)
DO2	GPIO13 (PWM capable)
DO3	GPIO22
DO4	GPIO23

To turn digital output on (drive VIO to the output), set the GPIO to high state. All other states (low, high-Z or weak pullup) will turn the digital output off.

Digital input GPIOs must be configured as input. A high level will be detected if sufficient voltage is detected on the digital input pin.

7.7 Console RS232

On Raspbian based installations, by default this port is configured as a serial Linux console and you can log into the Raspberry using a standard serial (null modem) cable and a terminal program like “Putty”. Use the following settings to log in: Speed (baud rate): 115200, Bits: 8, Parity: None, Stop Bits: 1, Flow Control: None

© **Janz Tec AG**

7.8 Power Supply Monitor

The power supply monitor unit (PMON) can be used to monitor the input voltage, and to program thresholds, at which the power supply is turned off (under voltage lockout). Refer to figure 11 for a block diagram of the PMON.

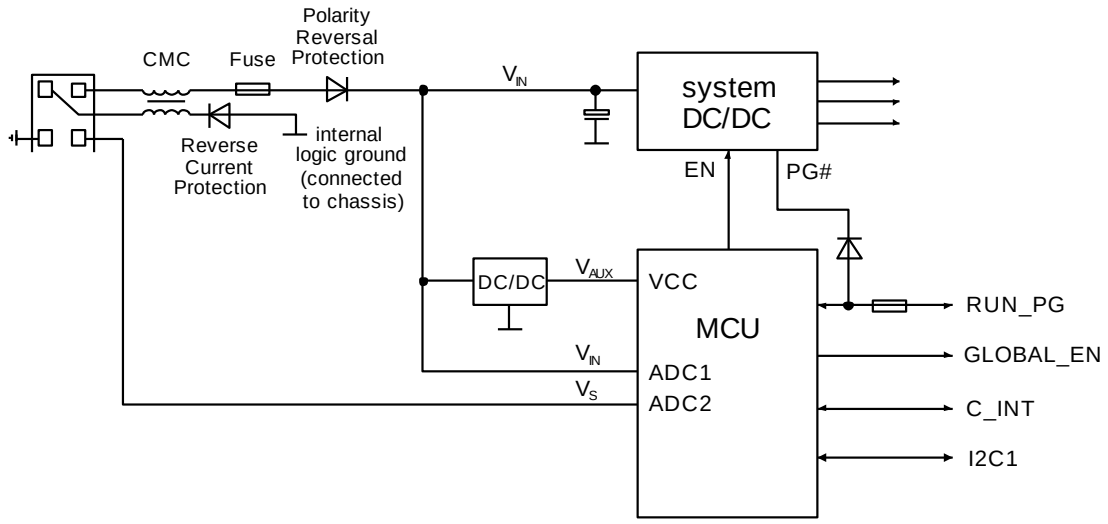


figure 11: Power Supply Monitor Unit

The power supply monitor unit is implemented by an independent microcontroller, which measures the input voltage (V_{IN}) and the sense voltage (V_S). The MCU can shut down the main power supply, reset the system or alert the main CPU. This can be utilized for a couple of applications:

- To protect from operation at low input voltages:
  - Because this causes higher input current which might destroy or overload external power supply unit or might even blow internal fuses
  - Because this might cause damage when the external power supply is a battery
- To turn off the system by an external control signal. A timeout can be configured to give the system a chance to shut down before the power is finally forced off.

The MCU is powered by a separate voltage regulator. Therefore, only very little current (~ 2mA) is drawn when the system power supply is shut down. The MCU operates when V_{IN} > 4V.

Communication between main CPU and PMOM is possible via an I2C bus (see the overall block diagram of the power supply and communication module). This can be used for configuration and supervisory (e.g. checking of input voltage level). Refer to section **Fehler! Verweisquelle konnte nicht gefunden werden.** for programming of I2C transfer.

The PMON MCU can generate an interrupt to the compute module (C_INT).

The Vs input can be utilized as level sensitive control signal (with programmable threshold), or can be used in a power button mode where a short voltage pulse requests system power state. Vs has an input impedance of about 100 kΩ.

The PMON MCU also implements a watchdog. When this watchdog times out, power supply is turned off for a short period to allow a clean restart. The watchdog can be configured to be enabled automatically to also repower if very early boot stages fail.

7.8.1 Timing Diagrams

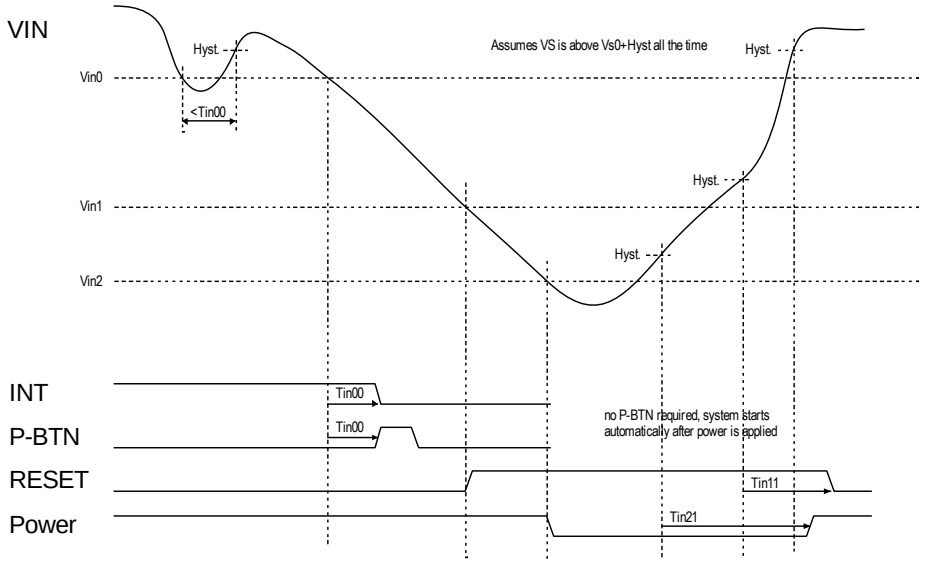


figure 12: System control by power supply voltage (VIN)

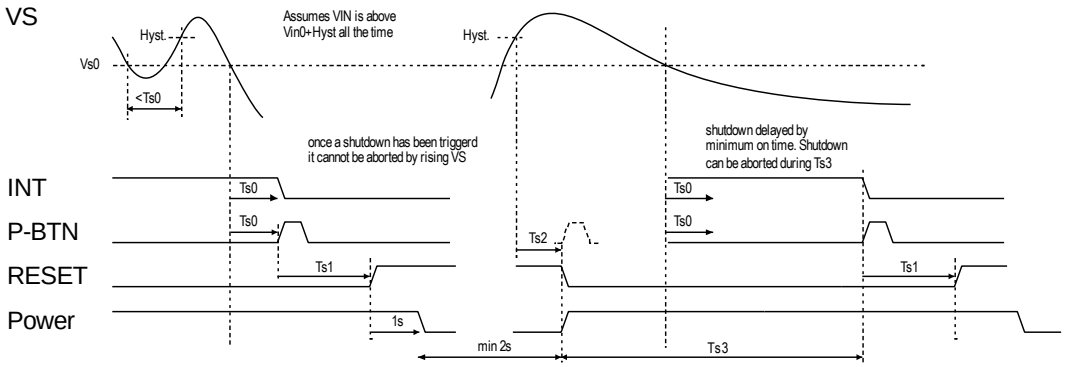


figure 13: System control by remote control input (VS)

- INT: This is the interrupt from the PMON CPU to the compute module (active low)
- P-BTN: not applicable to the emPC-A/RPI4.
- RESET: The reset to compute module (RUN_PG). Drawn active high, although the compute modue RUN_PG signal is actually active low.
- POWER: Main power supply state. Drawn high for power-on and low for power-off.

Refer to the register descriptions in the following sections for more information.

The MCU utilizes its ADC to monitor input voltages. It takes about 600µs to sample all input voltages. This is roughly the fastest response time to power supply events on falling edges.

Note: Although shown in figure 12 tin00 is not currently implemented in the firmware.

7.8.2 State Machine

In figure 14 you find the basic state machine of the power management CPU. It is shown here to allow a better understanding of operation.

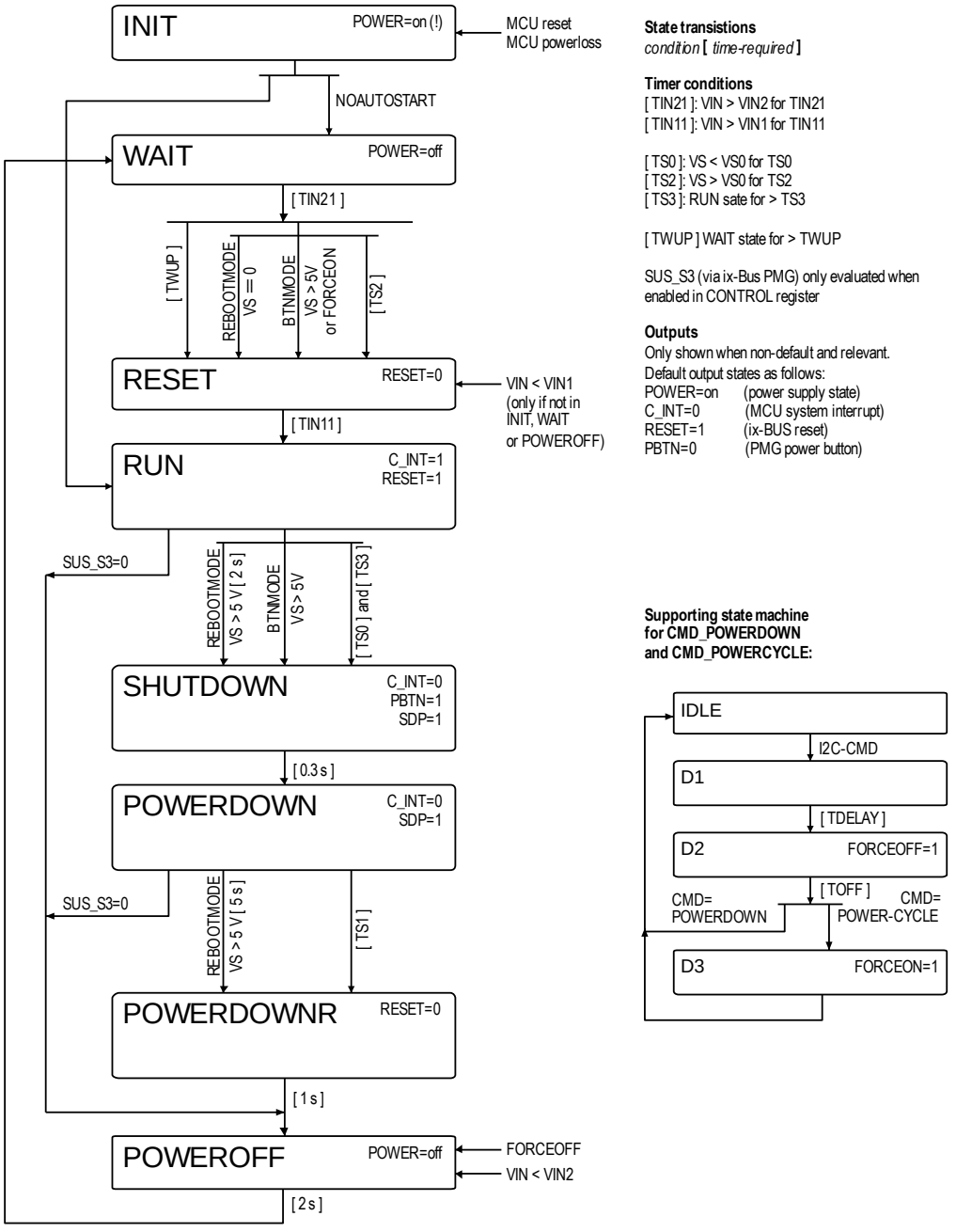


figure 14: PMON state machine

7.8.3 I2C Communication

The I2C interface is used to transfer data between main CPU and PMON. Note that PMON always acts as slave in I2C communication. Refer to section **Fehler! Verweisquelle konnte nicht gefunden werden.** for the corresponding I2C master description.

PMON provides a couple of 16bit registers which can be read/written by the I2C master. In figure 15 you find the protocols for reading and writing the registers. Refer to the I2C specification for details about the protocol.

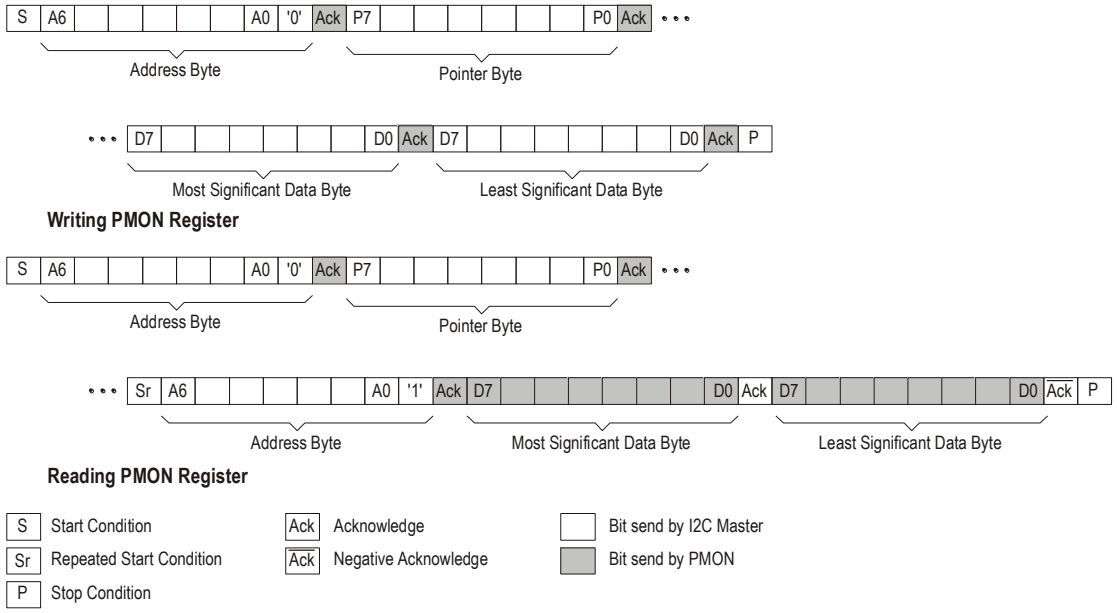


figure 15: PMON I2C Protocols

7.8.4 I2C Address

The I2C address used by PMON is 0x60.

7.8.5 I2C Registers

All registers are 16bit wide. They are accessed indirectly by a pointer byte, which has to be written at the beginning of each transaction. Reserved registers should not be written.

Pointer	Register	EEPROM save
0	ID	
1	CONTROL	Yes
2	STATUS	
3	COMMAND	
4	DATA	
5	Reserved	
6	VIN_ACTUAL	
7	VS_ACTUAL	
8	VIN0	Yes
9	VIN1	Yes
10	VIN2	Yes
11	VS0	Yes
12	VHYST	Yes
13..15	Reserved	
16	TIN00	Yes
17	TIN01	Yes

18	TIN11	Yes
19	TIN21	Yes
20	TS0	Yes
21	TS1	Yes
22	TS2	Yes
23	TS3	Yes
24	TS4	Yes
25	TWUP	Yes
26..29	Reserved	
30	WDOG	
31	WDOGINI	Yes
32..255	Reserved	

Indicated register settings can be internally saved to EEPROM. These values will then be restored upon power on of PMON. The EEPROM has a limited endurance of 50000 save operations.

Registers that are not documented are reserved for future feature enhancement of the MCU firmware. They should not be written to avoid compatibility problems with future versions.

Following you find the description of the defined registers. All voltage related registers are stored in units of 1/100 V. All time related registers are stored in units of 10 ms (for a maximum parameter of 650s), unless otherwise noted (TWUP, WDOG and POWERCYC/POWERDOWN commands).

Various parameters are visualized in figure 12 and figure 13

ID **Pointer 0 (16bit, ro)**

15	0
PMONID	

PMONID This provides possibility to identify the PMON functionality and will be used to indicate revisions in the future.
Check for the value 0x0104 to match with this document. Find a brief info about changes in the functionality below:

0x0102	• First release
0x0103	• Added firmware loader function
0x0104	• added VS pin support • added CONTROL_VSBTNMODE • added IX_PMON support (CONTROL_IXPMG_SUSS3) • added TWUP • added COMMAND_PWRCYCLE and COMMAND_PWRDOWN • added CONTROL_NOAUTOSTART • power reduction in power-off state
0x0105	• added STATUS_SDP bit
0x0106	• added CONTROL_VSREBOOTMODE
0x0107	• added system watchdog • fixed behaviour of power-on when power is first applied • in VSREBOOTMODE: system will not start when VS is high

Note that shipped hardware might not have the latest version installed, you should check at runtime.
If bit 7 of the ID register is set, this indicates an engineering version of the firmware.

CONTROL **Pointer 1 (16bit, ro)**

15															0
												VSR	IXP	NOA	VSB
RESET:															
Restored from EEPROM															

VSBTNMODE	If this bit is set, the VS input (mid contact on power connector) act as a power button input. If VS is > 5 V (typ.), the system is powered on. If this function is activated, you should program the VS0 threshold to zero.
NOAUTOSTART	If this bit is set, then rising power supply does not power on the system until started by some other event (VS in button mode or wakeup timer).
IXPMG_SUSS3	If this bit is set, then the IXPMG power management signal is evaluated when the system is powered on. Once a low is detected on IXPMG, the power is turned off. This is the SUS_S3 function used with x86 systems (emPC-X). This bit exists, as there is hardware (namely ARM boards) that do not implement the IXPMG signal. Do not set this bit on those boards to prevent the floating input signal cause unwanted power downs.
VS_REBOOTMODE	If this bit is set, the VS input (mid contact on power connector) act as a request-reboot button input. If VS is > 5 V (typ.) for > 2 sec, a reboot request is sent to the operating system via MCU_INT or SUS_S3. The request may also be detected by polling SDP bit in status register. If VS is > 5 V for > 5 sec, a reset is applied, and the system is restarted by a power cycle. If VS is > 5 V when the power supply voltage is turned on, then the system will not boot (you might use this as a boot inhibit). If this function is activated, you should program the VS0 threshold to zero. TS1 timer is active if this option is enabled, so you need to set TS1 timer to a reasonable value. TS3 timer is not active if this option is enabled.
Reserved	Reserved bits should be written as zero.

STATUS **Pointer 2 (16bit, ro)**

15	14	13	12	11	10	9	8	7		5	4				0
WDOG				WDR	BOR	EXR	POR	SDP			FSTATE				
RESET:															
undefined															

FSTATE	The state of the PMON internal state machine.
SDP	Shutdown pending if read as one.
POR	Set to one if last PMON MCU reset was a power-on reset.
EXR	Set to one if last PMON MCU reset was an external reset.
BOR	Set to one if last PMON MCU reset was a brown-out reset.
WDR	Set to one if last PMON MCU reset was a watchdog reset.
WDOG	Set to one if the system watchdog has triggered. This bit is sticky and resets when either the PMON MCU resets, or the RESET_ERR command is executed.

COMMAND **Pointer 3 (16bit, wo/ro)**

15															0

© **Janz Tec AG**

CMDCODE

RESET:

0

CMDCODE To trigger functions in the PMON firmware, write the corresponding command code into this register.
Once the command has completed, the register is cleared to zero by the PMON firmware.

DATA **Pointer 4 (16bit, rw)**

15	0
DATA	

DATA Data to be supplied with the command to be executed is supplied in this register. Write to DATA before writing to COMMAND.
For commands that return data, the data is available in this register after the command has been executed (COMMAND register flips back to IDLE).

VIN_ACTUAL **Pointer 6 (16bit, rw)**

15	0
VIN_ACTUAL	

VIN_ACTUAL Voltage reading of the power supply voltage (+ contact on power connector). Reading is in units of 1/100 V, so that register value of 0x800=2048 corresponds to 20.48 V.

VS_ACTUAL **Pointer 7 (16bit, rw)**

15	0
VS_ACTUAL	

VS_ACTUAL Voltage reading of the sense voltage input (mid contact on power connector). Reading is in units of 1/100 V, so that register value of 0x800=2048 corresponds to 20.48 V.

VIN0 **Pointer 8 (16bit, rw)**

15	0
VIN0	

RESET:

Restored from EEPROM

VIN0

VIN1 **Pointer 9 (16bit, rw)**

15	0
VIN1	

RESET:

Restored from EEPROM

VIN1 If the power supply voltage level is below VIN1, then the system is held in reset. This state can be left if input voltage level is above VIN1+VHYST for longer than TIN11.

VIN2 **Pointer 10 (16bit, rw)**

15 0

VIN2

RESET:

Restored from EEPROM

VIN2 If the power supply voltage level is below VIN2, the power supply is immediately turned off. This state can be left if input voltage level is above VIN2+VHYST for longer than TIN21.
This parameter has a factory set low limit. You cannot program values below this limit. This limit might depend on the system configuration. Generally, it will be higher for a system with high power consumption to prevent operation with low voltage and high current.

VS0 **Pointer 11 (16bit, rw)**

15		0
VS0		

RESET:

Restored from EEPROM

VS0 If the sense voltage level is below VS0 for a certain time, a system shutdown is triggered. When the voltage level is above VIN2+VHYST for a certain time, then a system is powered on. Refer to the various timing parameters for more details.
Program this register to zero to disable the sense voltage function. The sense voltage level can be read from VS_ACTUAL regardless of the VS0 setting.

VHYST **Pointer 12 (16bit, rw)**

15		0
VHYST		

RESET:

Restored from EEPROM

VHYST This register sets the hysteresis for the power supply and sense input signals.

TIN11 **Pointer 18 (16bit, rw)**

15		0
TIN11		

RESET:

Restored from EEPROM

TIN11 Reset delay (see VIN1).

TIN21 **Pointer 19 (16bit, rw)**

15		0
TIN21		

RESET:

Restored from EEPROM

TIN21 Power good delay (see VIN2).

TS0 **Pointer 20 (16bit, rw)**

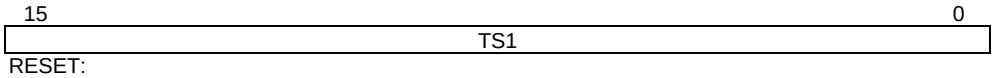
15		0
TS0		

RESET:

Restored from EEPROM

TS0 When VS is below the VS0 threshold for longer than TS0, then a system shutdown is triggered. If VS passed VS0+VHYST before TS0 expires, then no action is taken.
However, once a system shutdown has been triggered it cannot be aborted by rising VS.

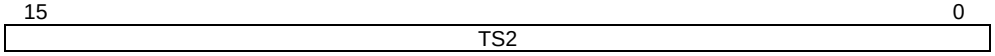
TS1 **Pointer 21 (16bit, rw)**



RESET:
Restored from EEPROM

TS1 TS1 is the shutdown timeout for the system (the system CPU module). Once a shutdown has been triggered, the PMON firmware waits for the system to shut down (monitored by COM Express SUS_S3 signal). If a shutdown has not been detected after TS1 expires, then a system reset is applied and power is turned off one second later.

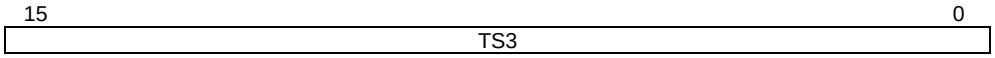
TS2 **Pointer 22 (16bit, rw)**



RESET:
Restored from EEPROM

TS2 The system is off (and a minimum off time of 2 seconds has elapsed) and VS is > VS0+VHYST for longer than TS2, then the system is turned on. The power supply is turned on, and if the system CPU module does not start automatically, a single power button event is generated to force turn on.

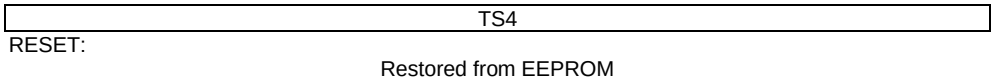
TS3 **Pointer 23 (16bit, rw)**



RESET:
Restored from EEPROM

TS3 A system shutdown is inhibited until the system is running for more than the time programmed by TS3. This can be used to prevent a shutdown during operating system bootup phase, when the time set by TS1 might not be enough to perform a proper shutdown.

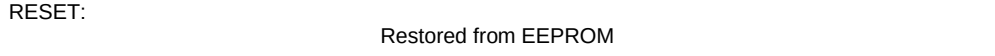
TS4 **Pointer 24 (16bit, rw)**



RESET:
Restored from EEPROM

TS4 Do not write to this register.

TWUP **Pointer 25 (16bit, rw)**



RESET:
Restored from EEPROM

TWUP Wake up timer. If this value is nonzero, then power is turned on when

PMON is in the power off state and is waiting for a start event for longer than TWUP seconds.

WDOG **Pointer 30 (16bit, wo)**

15	14	13	12	11	10	9	8	7	5	4	0
reserved					D	L	S	WDDATA			

System-POWERON:

loaded from WDOGINI

WDDATA This field has several functions:

- When S, D and L are written zero and WDDATA=0x5a then the watchdog is retriggered.
- When S, D and L are written zero and WDDATA=0x3c, then the next write to WDOG might change the watchdog parameters (set, lock or disable).
- When S is written as 1, then WDDATA sets the watchdog timeout period.
  The timeout period is passed as a mini floating-point number in the following format:

7	6	5	0
EXP		MANT	

The timeout is MANT if EXP==0, and MANT•$2^{(EXP+2)}$ otherwise.
Example Codes: 0x00→0, 0x01→1, 0x02→2, 0x021→4, 0x022→8, 0xff→15872.

S SET and enable watchdog if this bit is written 1. When watchdog timeout time has been changed, it must be retriggered for the new timeout to be active.

L LOCK watchdog if this bit is written 1. When the watchdog is locked, no further changes in the watchdog configuration (set/disable or timeout time) are possible. This bit can be set in the same write operation that also starts the watchdog.

D DISABLE watchdog if this bit is written 1.
The watchdog is implicitly disabled when the system is powered off (PMON state machine enters POWEROFF).

Reserved Reserved bits should be written 0

WDOGINI **Pointer 31 (16bit, wo)**

15	14	13	12	11	10	9	8	7	5	4	0
WDOGINI											

RESET:

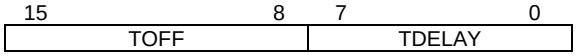
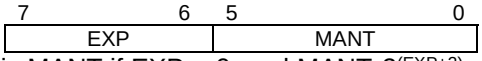
Restored from EEPROM

WDOGINI This value is written to WDOG register whenever the system is powered on (PMON state machine changes to RUN state).

7.8.6 Command Codes

This section lists the command codes that are accepted by the PMON firmware. Undocumented command codes should not be issued.

CMD code	Command	Description
0	IDLE	When read, this command code indicates idle state.
1	SAVE_EEP	Save all parameters that are indicated as “EEPROM saved” into the PMON MCUs internal EEPROM. This makes changes permanent. Before writing, some simple sanity checks are made. E.g. you cannot save a voltage threshold that is higher than the

		supported operating value. No checks are made on the timing parameters.
2	RESET_ERR	Reset the PMON MCU reset and watchdog information in the STATUS register.
7	POWERCYC	Trigger a power cycle (forcing power down and up again). Timing parameters are passed by the DATA register: DATA **POWERCYC/POWERDOWN**  TDELAY (in seconds) is the delay until the power is turned off, and TOFF (in seconds) is the time for which the system is held in power off mode. FIXME: There is no way of waking up the system during TOFF (except restarting the MUC by bringing power down to near zero) FIXME: It is not possible to abort a POWERCYC one it is started. Both parameters are passed as mini floating-point numbers in the following format:  The value is MANT if EXP==0, and MANT•$2^{(EXP+2)}$ otherwise. Example Codes: 0x00→0, 0x01→1, 0x02→2, 0x021→4, 0x022→8, 0xff→15872.
8	POWERDOWN	This is very much like POWERCYC, however the system does not start again after TOFF has expired.
10	LOADER_JMP	Transfer execution to the firmware loader to start update process. Must be unlocked by LOADER EN in order to transfer execution. For more information about the update process refer to the documentation of the Janztec I2C boot loader.
13	LOADER_EN	This command needs to be executed just before the LOADER_JMP command with no other commands in between.
14	WDOG_RDCNTR	Read the current value of the watchdog timeout timer in seconds. The returned value is the time until the watchdog expires. Load DATA register with 0 before executing this command.

7.8.7 I2C Timing

PMON can operate with 400 kHz clock speed. However, it requests wait states by holding down the SCL line when required.

7.8.8 Update

The PMON firmware can be updated via I2C. A user space utility for updates is available (ilavr). However, you must prevent all clients from accessing PMON registers during updates, or the update will fail.

8 Appendices

8.1 Technical Data

Only CM4 features usable on emPC-A/RPI4 are listed.

Processing Core	
CPU	BCM2711, up to 1500MHz quad-core 64bit ARMv8 Cortex A72
Memory	
Main Memory	Up to 8 GB SDRAM
Storage (exclusive or)	
Micro SDcard	1 x internal socket (standard)
eMMC	Up to 32 GB (option)
External Interfaces	
Video	1 x HDMI (up to 4kp60 supported)
Ethernet	1 x 10/100/1000 Mbit/s Ethernet, IEEE 1588 support
WLAN	As defined by CM4 Module, external antenna required
Bluetooth	As defined by CM4 Module, external antenna required
WAN/LTE	Various LTE options possible (e.g. LTE-CAT1/4, LTE-NB, with or without GPS). Up to two antenna connectors. Refer to ordering information.
USB	2 x USB2.0
CAN	MCP2518FD controller • ISO/DIS 11898-2 (isolated from logic), • jumper switchable 120 Ohm termination resistor
Serial Port	Console (RPI defined) • RS232 (2 wire) SC16IS760 controller • RS232 (4 wire) • RS485
Digital IO	• 4 x digital OUT (0.5 A max.), 10 .. 34 V • 4 x digital IN • Isolated from system ground
Indicators and Switches	
User LEDs	2 programmable LEDs (red, green controlled by CM4) 2 programmable LEDS (green)
System	
Housing	Galvanized sheet metal and painted aluminium
RTC	RV-8803-C7, temperature compensated. Accuracy ±5 ppm over temperature, aging ±5 ppm (first year @ 25°C)
Battery	CR 2032, for real time clock
TPM	SLM 9670AQ2.0
Power Requirements	
Power Supply	DC power, 9 .. 34 V
Inv.-pol. protection	Yes
Fuse	Internal 2.5 A melting fuse in DC in, GND is unfused
potential separation	No, GND is connected to connector shells and PE contact

Inrush Current (max)	TBD
Power Dissipation	TBD

External Load Capabilities

+5V (USB)	Max. 500 mA per USB port, max 750 mA for all USB ports

Environmental Specifications

Ambient Temperature operating	-20 .. +TBD °C (vertical orientation / wall and DIN rail mounting) -20 .. +TBD °C (horizontal orientation / desk-top mounting)

TBD maximum temperatures to be finalized.

Preliminary data:
- Max 45 °C with full CPU load in still air (no external airflow)
- Max 60 °C with full CPU load and external airflow

Power Consumption in this condition ~ 6W

at sea level, derated by 1 °C per 300 m above sea level to a maximum of 2000 m.

Temperature storage	-20 .. +75 °C [2)]
Humidity	5% .. 95% r.H., non condensing
Protection Class	IP20

EMC

CE (RED)	EN 301 489 - 1 V2.2.3 (2019-11) EN 301 489 - 17 V3.2.4 (2020-09) EN 301 489 - 19 V2.2.1 (2022-09) EN 301 489 - 52 V1.2.1 (2021-11) Emissions: Class A
Industrial Immunity	EN 61000-4-3: 10 V/m (w.o. LTE/GNSS), 3 V/m (w. LTE/GNSS) EN 61000-4-4: 2 kV (power), 1 kV (IO signals, except USB/Ant.) EN 61000-4-5: 0,5 kV (power) EN 61000-4-6: 10 V (except USB/Ant.)

Physical Dimensions

Size (WxHxD)	117 x 35.0 x 93.5 mm (excluding connector overhang and screws)
Weight	520 g

8.2 References

These references direct you to manuals and specifications that you might need to know when you attempt to program the product. Most of the documents can be downloaded from the Internet. Look for the WWW servers of the component/chip manufacturers.

[1] https://www.raspberrypi.org/downloads/
[2] https://www.raspberrypi.org/documentation/

WWW-References

Janz Tec AG	www.janztec.com

© **Janz Tec AG**

8.3 Dimensions

Refer to figure 16 for the housing dimensions.

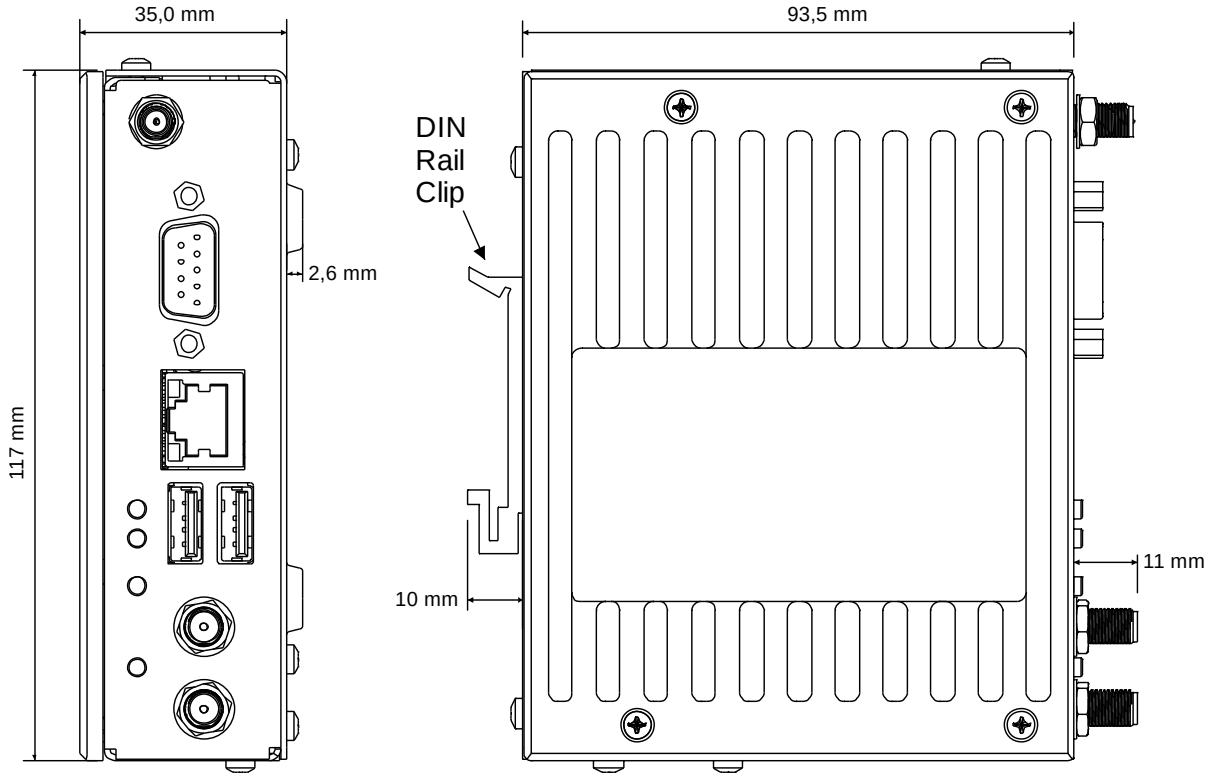


figure 16: housing dimensions

8.4 CM4 GPIO Usage

GPIO	Signal	Remarks
0	ID-SD	
1	ID-SC	
2	SDA1	Internal I2C devices
3	SCL1	Internal I2C devices
4	DI1	Digital Input
5	DI2	Digital Input
6	DI3	Digital Input
7	DI4	Digital Input
8	SPI0_CE0	Expansion Connector, use as SPI0/I2C4/UART4
9	SPI0	Expansion Connector, use as SPI0/I2C4/UART4
10	SPI0	Expansion Connector, use as SPI0/I2C5/UART4
11	SPI0	Expansion Connector, use as SPI0/I2C5/UART4
12	DO0	Digital Output (PWM capable)
13	DO1	Digital Output (PWM capable)
14	TXD0	Console
15	RXD0	Console
16	SPI0_GPIO	Expansion Connector, use as you want (e.g. INT or 2nd SPI-CE)
17	C_INT	Signal from ATtiny MCU (interrupt)
18	SPI6_CE_CAN	Internal SPI
19	SPI6_MISO	Internal SPI
20	SPI6_MOSI	Internal SPI
21	SPI6_SCLK	Internal SPI
22	DO2	Digital Output
23	DO3	Digital Output
24	SPI6_CE_TPM	Internal SPI (GPIO based CE)
25	SPI6_INT_CAN	Internal SPI
26	SPI6_INT_UART	Internal SPI
27	SPI6_CE_UART	Internal SPI

© **Janz Tec AG**

8.5 Port Expander Usage

The I2C port expander MCP23018 has the functions as described below.

After reset all port expander outputs are in “1” state (=high zero on output).

Port	Function	DIR	Remarks
A0	EEP_WP	Out	Set 0 to enable writing the optional I2C EEPROM
A1	EEP_WP_PEX	Out	Set 0 to set CM4 EEPROM_nWP signal (pin 20) to low
A2	RS485_SEL	In	Read 0 when RS485 selection jumper is set
A3	LTE1 LED	Out	Set 0 to turn on LED
A4	LTE2 LED	Out	Set 0 to turn on LED
A5	LTE_VBUS_EN	Out	Set 0 to enable power to the LTE modem VBUS power rail
A6	EEP_WP_CB	Out	Set 0 to enable writing the ID EEPROM
A7	N/A		Unused
B0	LTE_PWRKEY	Out	Set 0 to pull LTE the modem PWRKEY pin low
B1	LTE_RESET_N	Out	Set 0 to reset the LTE modem
B2	LTE_STATUS	In	Read as 0 when the LTE modem STATUS pin is high
B3	LTE_USB_BOOT	Out	Set 0 to apply power to the LTE modem USB_BOOT pin to force the modem to stay in bootloader after a power cycle.
B4	LTE_PIN_18	Out	Set 0 to force the LTE modem pin 18 low. The function is modem depending. No external pullup is available
B5	LTE_PWR_EN	Out	Set 0 to turn off the LTE modem power supply to save power
B6	LTE_PIN_19	Out	Set 0 to force the LTE modem pin 19 low. The function is modem depending. No external pullup is available
B7	N/A		unused

8.6 Expansion Connector

To allow adding external expansions without opening the case, an expansion connector was added.

NOTICE
The connector uses a proprietary pinout. Damage to equipment is likely when you insert a standard PCIe card into this slot.

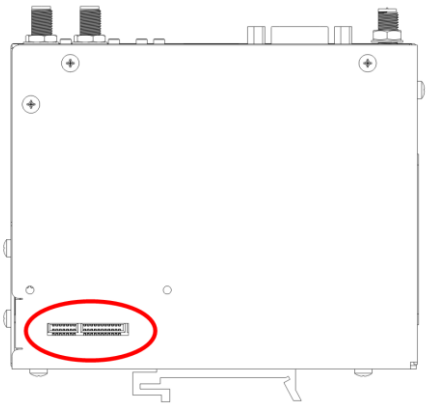


figure 17: expansion connector location

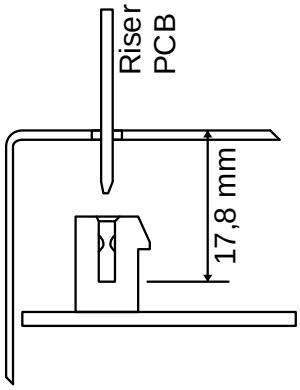


figure 18: riser PCB: nominal insertion depth

The following signals are available:

Pin	SideB	SideA	Comment
1	GND	NC	
2	USB_DP	VIN (9..34 V)	
3	USB_DN	VIN (9..34 V)	
4	GND	GND	
5	I2C1_CLK	GPIO11	
6	I2C1_DAT	GND	
7	GND	GPIO10	
8	+3.3 V	GPIO9	
9	GPIO8	+3.3 V	
10	+5 V	+3.3 V	
11	GPIO16	PCIE_RST#	
Key notch			
12	CLKREQ#	GND	
13	GND	REFCLK+	
14	HSOp(0)	REFCLK−	
15	HSOn(0)	GND	
16	GND	HSIp(0)	
17	GND	HSIn(0)	
18	GND	GND	

The GPIOs can be utilized for various purposes:

GPIO	SPI Function	I2C Function	UART Function
8	SPI0_CE0	SDA4	TXD4
9	SPI0_MISO	SCL4	RXD4
10	SPI0_MOSI	SDA5	CTS4
11	SPI0_SCLK	SCL5	RTS4

I2C1 was only added for detection purposes (to read a HAT like EEPROM). If you want to add I2C expansions, you should prefer I2C4 or I2C5. Remember that I2C1 must be forced to bit-bang mode and thus has increased CPU overhead.

8.7 EMC Cable Installation

In figure 19 we show the EMC shield (PE) connection that we utilized during EMC compatibility tests to pass industrial level immunity:

- DIN rail is connected to PE
- Clamp cable shields (for signals running to the unshielded multi connector) to the EMC shield (DIN rail)
- Connecting PE to the multi connectors PE contact is not necessary when the DIN rail is utilized as above

The terminals/clamps utilized are just examples; other models might work equally well.

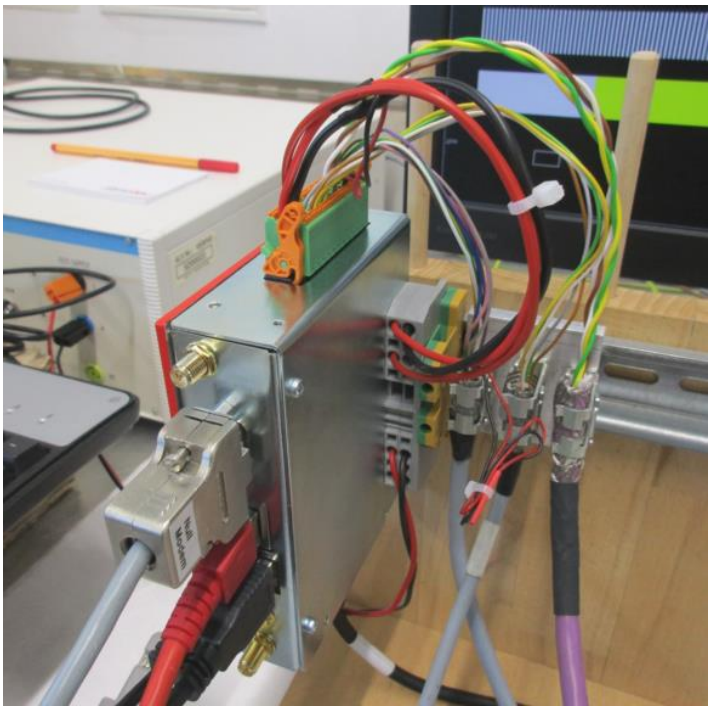


figure 19: EMC Shielding

© **Janz Tec AG**

8.8 emPC-ARPI2/3 vs emPC-ARPI4

The emPC-ARPI4 tries to follow the emPC-ARPI3 in some aspects (basic form-factor, interfaces). However, some internals have been changed to make optimum use of the new possibilities of the CM4's BCM2711 SoC IO features. Features that were not available in those early days when the PCB board of the emPC-ARPI2/3 was designed for the Model 2 B. Here are some of the non-obvious changes.

8.8.1 General Incompatibilities

- Pin 22 of the IO connector is no longer PE, but can be used as control input

8.8.2 Different Usage of GPIO signals

ARPI2/3	GPIO	ARPI4	Remarks
-	0	ID-SD	
-	1	ID-SC	
I2C-SDA	2	SDA1	Equal use
I2C-SCL	3	SCL1	Equal use
EEP_WP	4	DI1	
LED green	5	DI2	
DI1	6	DI3	
SPI_CE_UART	7	DI4	
SPI_CE_CAN	8	SPI0_CE0	SPI0 is now for the expansion connector
SPI_MISO	9	SPI0_MISO	
SPI_MOSI	10	SPI0_MOSI	
SPI_CLK	11	SPI0_CLK	
LED red	12	DO0	
DI2	13	DO1	
Console TxD	14	TXD0	Equal use
Console RxD	15	RXD0	Equal use
DI3	16	SPI0_GPIO	
INT_UART#	17	C_INT	
DO4	18	SPI6_CE_CAN	SPI6 is now for the internal devices
DI4	19	SPI6_MISO	
-	20	SPI6_MOSI	
-	21	SPI6_SCLK	
DO2	22	DO2	
DO1	23	DO3	
RS485_M.DT.	24	SPI6_CE_TPM	
INT_CAN#	25	SPI6_INT_CAN	
-	26	SPI6_INT_UART	
DO3	27	SPI6_CE_UART	
• The Janztec devicetree overlays will solve all issue regarding the SPI peripherals • You have to change code if use access the LEDs directly via GPIO • You have to change code if you use the digital inputs and outputs			

8.9 Product History

Version	Release Date	Name	Changes
PRE	-	-	• Early engineering samples
1.0	-	-	• First prototypes • Compared to the PRE samples, the LTE_VBUS_EN has changed polarity: Must be set to 0 to enable modem USB.
1.1	-	-	• For demo and compliance testing • No functional changes (PCB layout fixes)
			•

8.10 Manual History

Version	Release Date	Name	Changes
1.0	2024-11-15	as	• Updated VIO to 34 V max (the IO recommended supply voltage of the VN330SP is 10 .. 36 V) • Some reading improvements • Added figure “RS232/RS485 mode switch and transceiver control” • Cosmetic Update to some figure • Added chapter “emPC-ARPI2/3 vs emPC-ARPI4” • Added image riser card insertion depth
			•
			•
			•

© **Janz Tec AG**