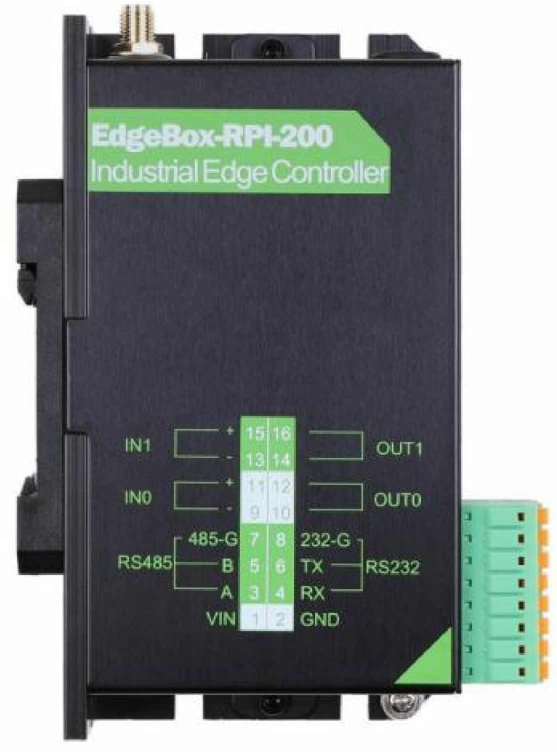
Seeedstudio EdgeBox-RPI-200 EC25 Raspberry PI CM4 Based Edge computer User Manual

Home » Seeedstudio » **Seeedstudio EdgeBox-RPI-200 EC25 Raspberry PI CM4 Based Edge computer User Manual**

Contents

seeed studio

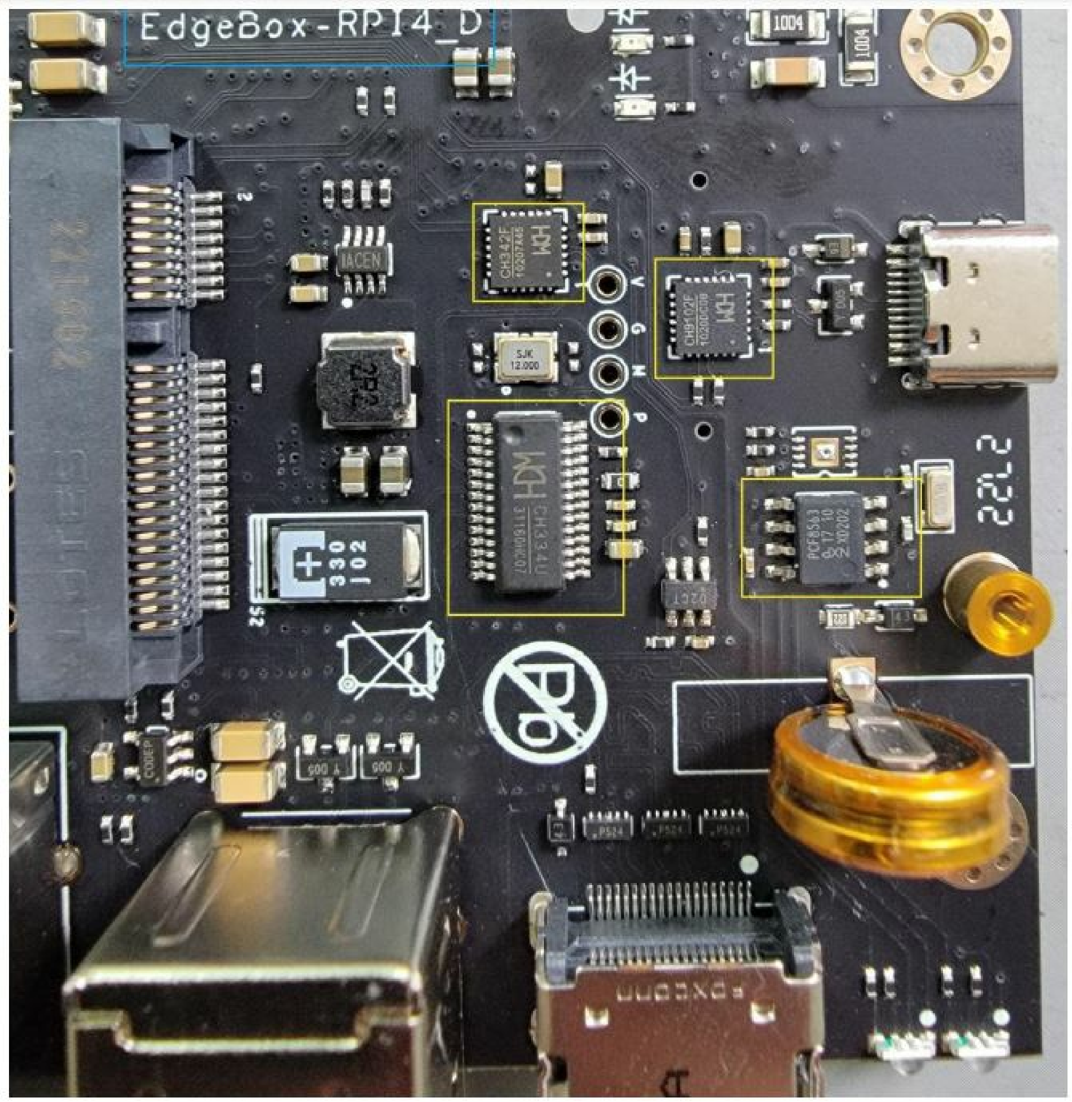
Seeedstudio EdgeBox-RPI-200 EC25 Raspberry PI CM4 Based Edge computer



Revision History

Revision	Date	Changes
1.0	17-08-2022	Created
2.1	13-01-2022	Product Change Notice

Product Change Notice:



As a part of our continual improvement process, we made the below changes in hardware version D.
There is impact on the software due to this change.

- CP2104->CH9102F
- USB2514B->CH334U
- CP2105->CH342F
- The description in the Linux have been changed:
  - ttyUSB0-> ttyACM0
  - ttyUSB1-> ttyACM1
  - MCP79410->PCF8563ARZ
  - The address of the new RTC is 0x51.

Introduction

EdgeBox-RPI-200 is a rugged fan less Edge Computing Controller with Raspberry Pi Computer Module 4(CM4) for harsh industry environment. It can be used to connect the field networks with cloud or IoT applications. It is designed from the ground up to meet the challenges of rugged applications at competitive prices, ideal for small business or small order with scale multi-level demands.

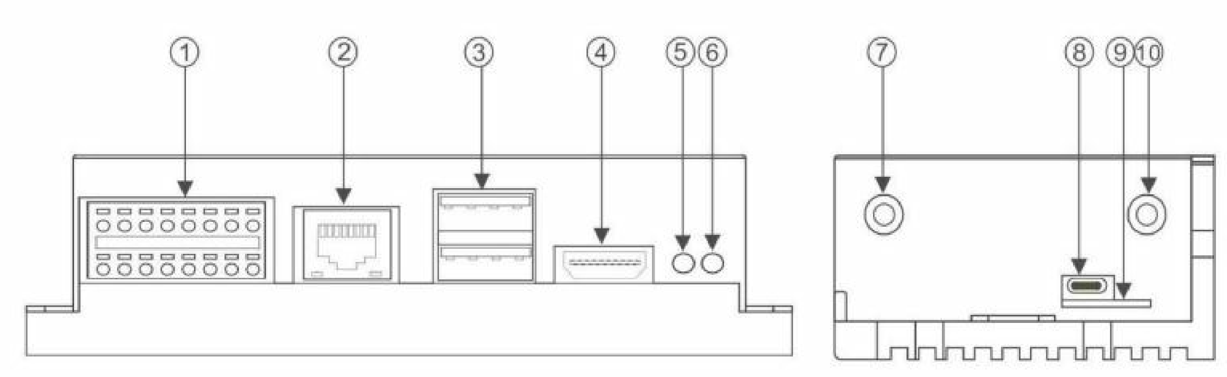
Features

- State-of-the-art Aluminum chassis for Harsh environment
- Integrated passive heat sink
- Built-in mini PCIe socket for RF module, such as 4G, WI-FI, Lora or Zigbee
- SMA antenna holes x2
- Encryption chip ATECC608A
- Hardware Watchdog
- RTC with Super Capacitor
- Isolated DI&DO terminal
- 35mm DIN Rail support
- Wide power supply from 9 to 36V DC
- Optional: UPS with SuperCap for safe shutdown*
- Raspberry Pi CM4 onboard WiFi 2.4 GHz, 5.0 GHz IEEE 802.11 b/g/n/ac equipped**
- Raspberry Pi CM4 onboard Bluetooth 5.0, BLE equipped**

These features make the EdgeBox-RPI-200 designed for easy setup and quick deployment for typical industrial applications, such as status monitoring, facility management, digital signage and remote control of public utilities. Furthermore, it is a user-friendly gateway solution with 4 cores ARM Cortex A72 and most industry protocols can save on total deployment costs including electrical power cabling cost and help reduce the product's deployment time. Its ultra-lightweight and compact design is the answer for applications in space-constricting environments ensures it can operate reliably in a variety of extreme environments including in-vehicle applications.

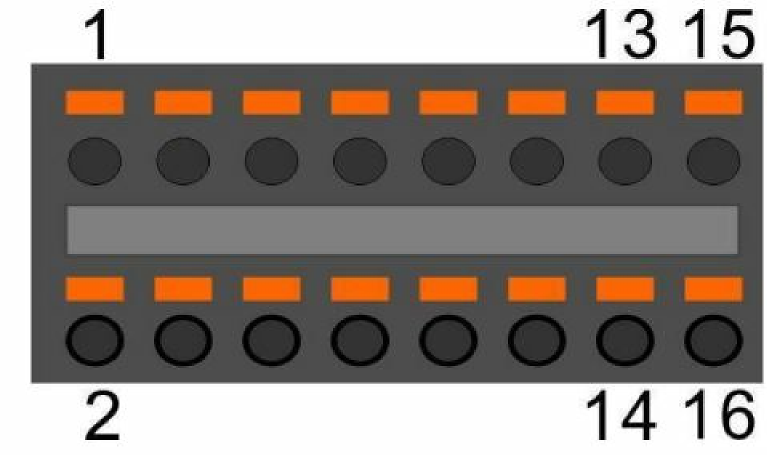
NOTE: For UPS function please contact us for more information. WiFi and BLE features can be found in 2GB and 4GB versions.

Interfaces



1. Multi-Func phoenix connector
2. Ethernet connector
3. USB 2.0 x 2
4. HDMI
5. LED2
6. LED1
7. SMA antenna 1
8. Console (USB type C)
9. SIM card slot
10. SMA antenna 2

Multi-Func phoenix connector

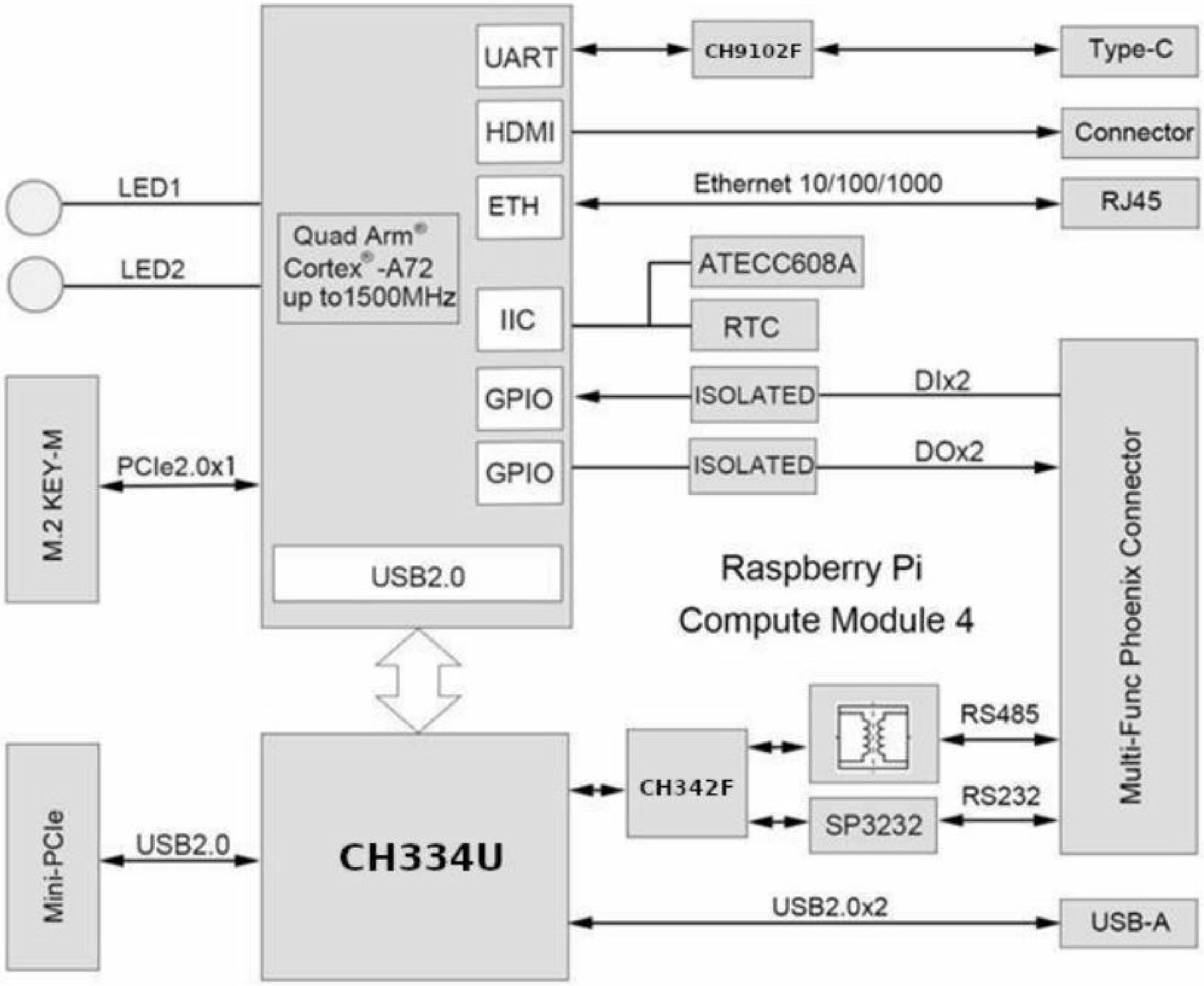


Note	Func name	PIN #	PIN#	Func name	Note
	POWER	1	2	GND	
	RS485_A	3	4	RS232_RX	
	RS485_B	5	6	RS232_TX	
	RS485_GND	7	8	RS232_GND	
	DI0-	9	10	DO0_0	
	DI0+	11	12	DO0_1	
	DI1-	13	14	DO1_0	
	DI1+	15	16	DO1_1	

NOTE: 24awg to 16awg cable are suggested

Block Diagram

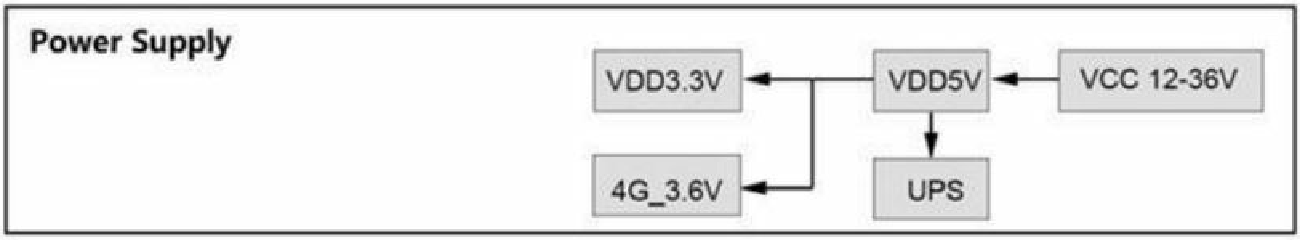
The processing core of the EdgeBox-RPI-200 is a Raspberry CM4 board. A specific base board implements the specific features. Refer to next figure for the block diagram.



Installation

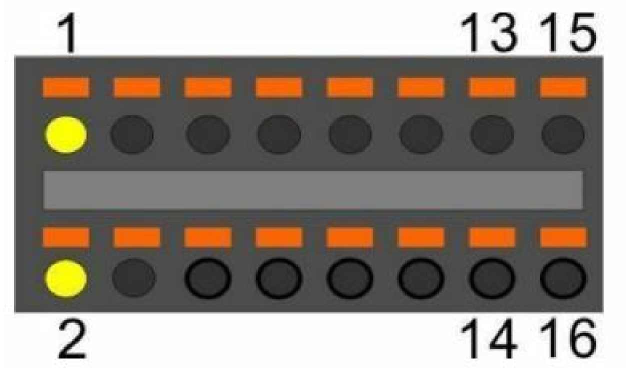
Mounting

The EdgeBox-RPI-200 is intended for two wall mounts, as well one with 35mm DIN-rail. Refer to next figure for the recommended mounting orientation.



Connectors and Interfaces

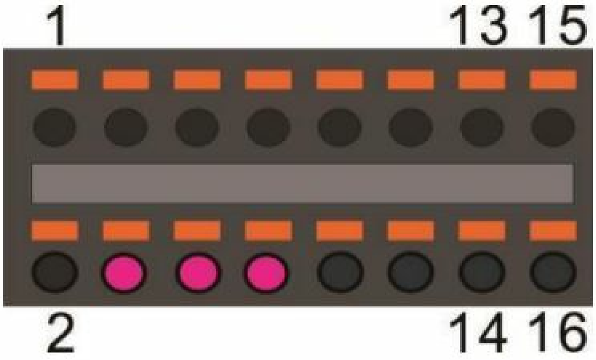
Power supply



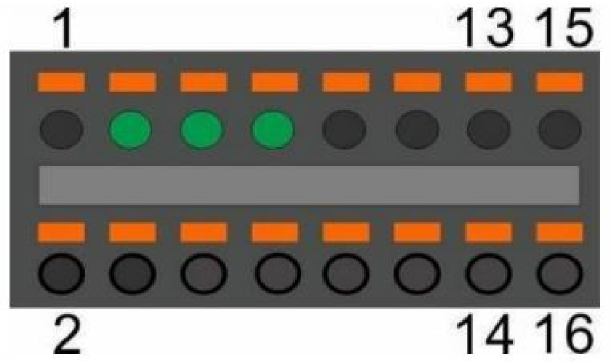
Pin#	Signal	Description
1	POWER_IN	DC 9-36V
2	GND	Ground (Reference potential)

The PE signal is optional. If there is no EMI present, the PE connection can leave open.

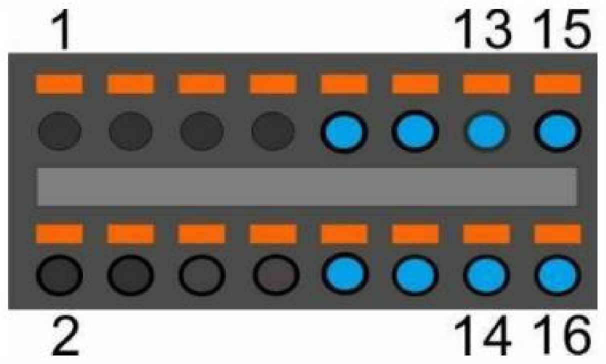
Serial Port (RS232 and RS485)



Pin#	Signal	Description
4	RS232_RX	RS232 receive line
6	RS232_TX	RS232 transmit line
8	GND	Ground (Reference potential)

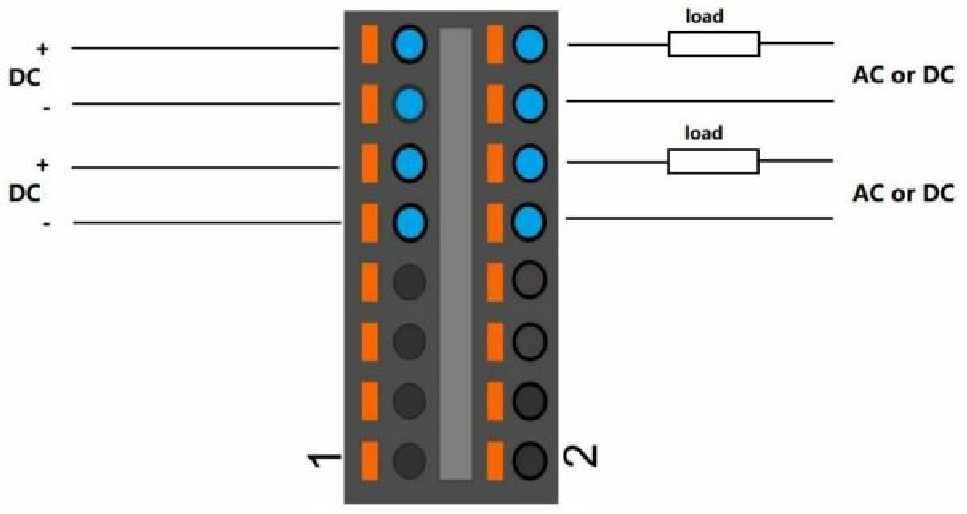


Pin#	Signal	Description
3	RS485_A	RS485 difference line high
5	RS485_B	RS485 difference line low
7	RS485 _GND	RS485 Ground (isolated from GND)



Pin#	Signal of terminal	PIN Level of active	PIN of GPIO from BCM2711	NOTE
09	DI0-	HIGH	GPIO17	
11	DI0+			
13	DI1-	HIGH	GPIO27	
15	DI1+			
10	DO0_0	HIGH	GPIO23	
12	DO0_1			
14	DO1_0	HIGH	GPIO24	
16	DO1_1			

NOTE:



NOTE:

1. DC voltage for input is 24V (+- 10%).
2. DC voltage for output should be under 60V, the current capacity is 500ma.
3. Channel 0 and channel 1 of input are isolated to each other
4. Channel 0 and channel 1 of output are isolated to each other

HDMI

Directly connected to the Raspberry PI CM4 board with TVS array.

Ethernet

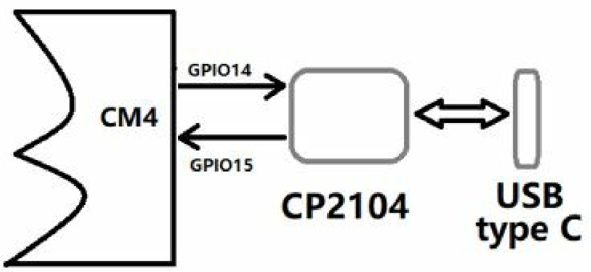
Ethernet interface is same as Raspberry PI CM4,10/100/1000-BaseT supported, available through the shielded modular jack. Twisted pair cable or shielded twisted pair cable can be used to connect to this port.

USB HOST

There are two USB interfaces at the connector panel. The two ports share the same electronic fuse.

NOTE: Max current for both ports is limited to 1000ma.

Console (USB type-C)

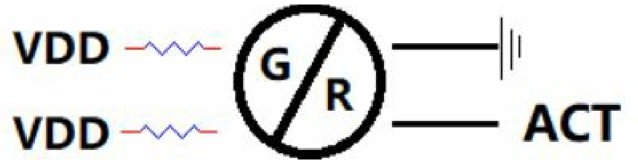


The design of console used a USB-UART converter, most OS of the computer have the driver, if not, the link below may be useful: This port is used as a Linux console default. You can log into the OS use the settings of 115200,8n1(Bits: 8, Parity: None, Stop Bits: 1, Flow Control: None). A terminal program such as putty is needed, too. The default user name is pi and password is raspberry.

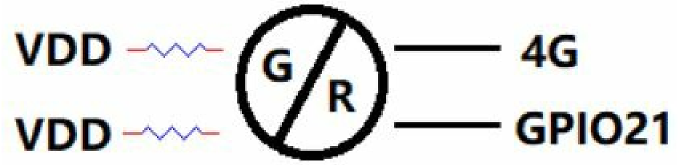
LED

EdgeBox-RPI-200 use two green/red dual colour LED as outside indicators.

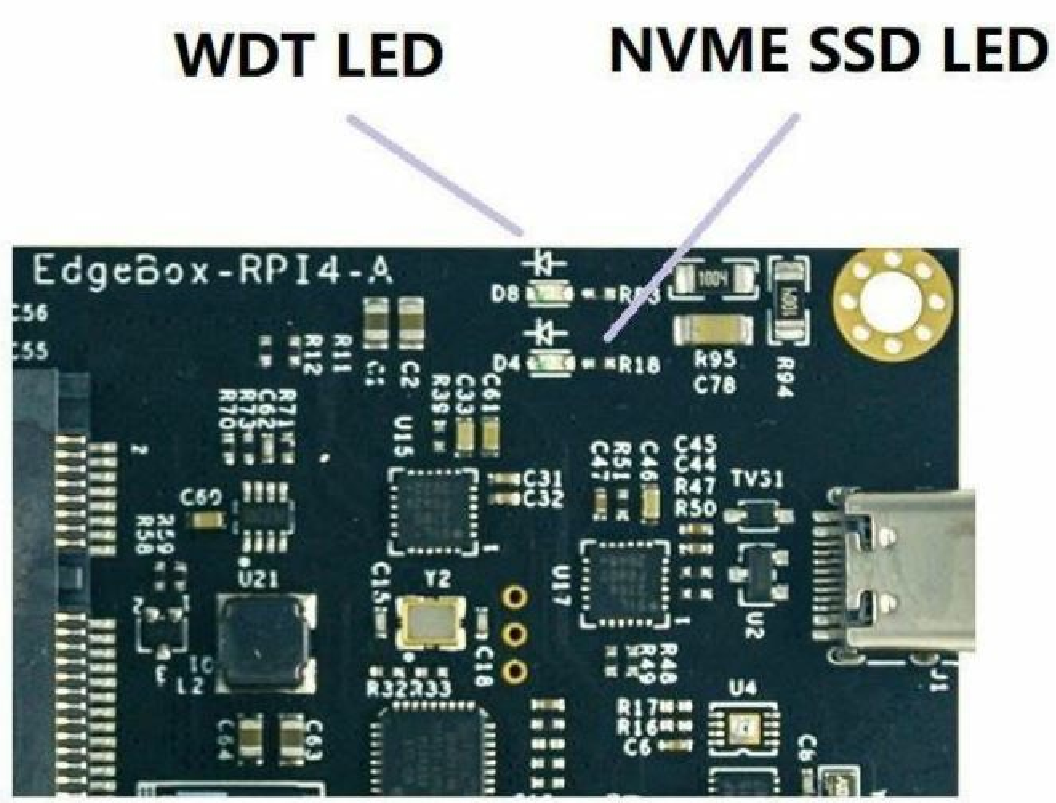
LED1: green as power indicator and red as eMMC active.



LED2: green as 4G indicator and red as user programmable led connected to GPIO21, low active, programmable.

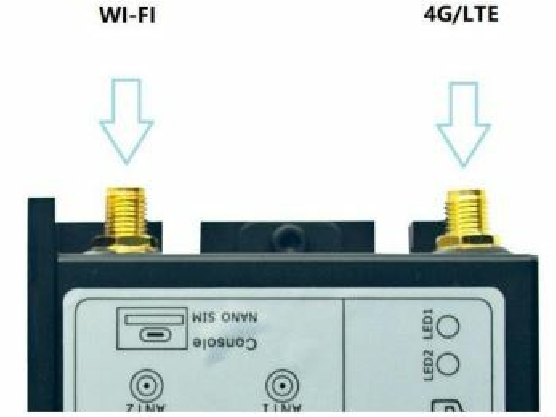


EdgeBox-RPI-200 also use two green colour LED for debug.



SMA Connector

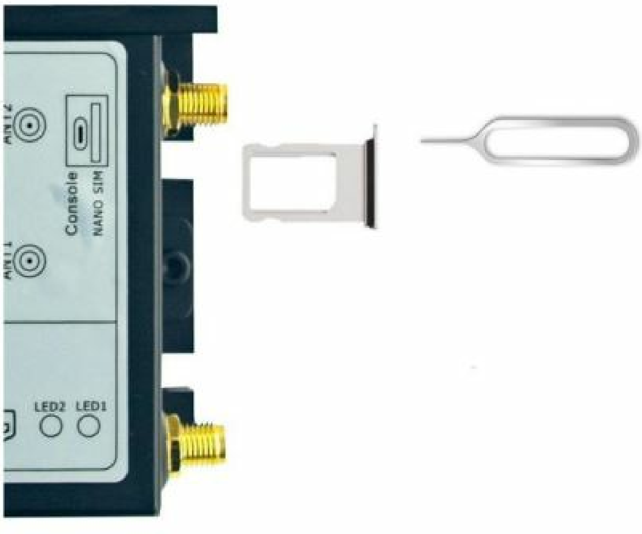
There are two SMA Connector holes for antennas. The antenna types are very depended on what modules fitted into the Mini-PCIe socket. The ANT1 is default used for Mini-PCIe socket and ANT2 is for Internal WI-FI signal from CM4 module.



NOTE: The functions of the antennas are not fixed, maybe adjusted to cover other usage.

NANO SIM card slot (Optional)

The sim card is only needed in cellular (4G, LTE or others based on cellular technology) mode.

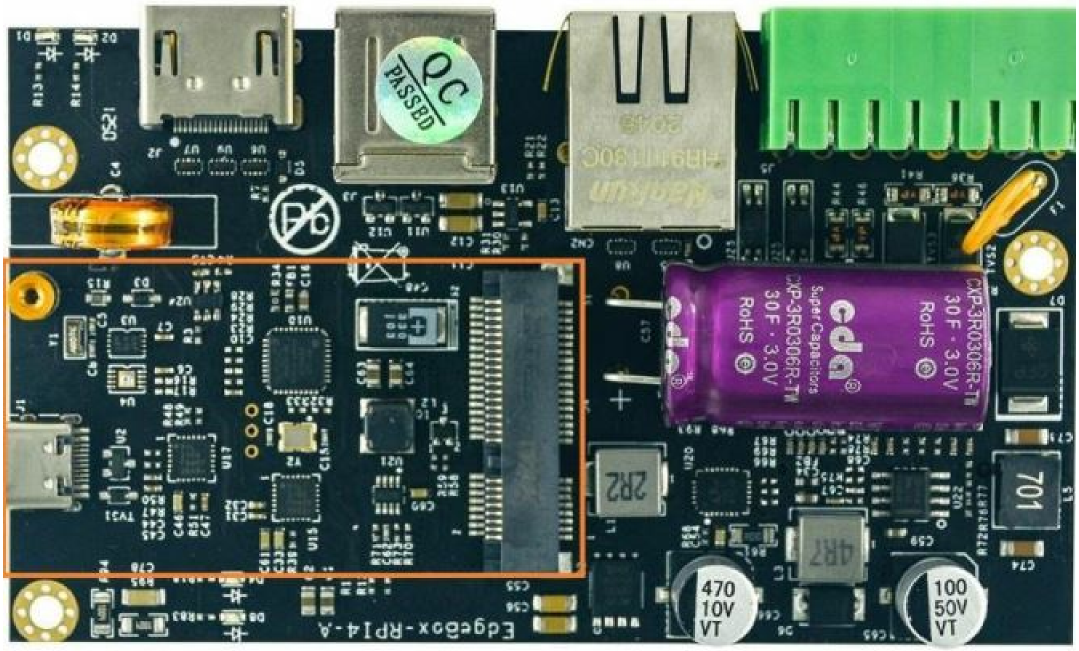


NOTE:

1. Only NANO Sim card is accepted, pay attention to the card size.
2. The NANO sim card is inserted with chip side top.

Mini-PCIe

The orange area is the rough Mini-PCIe add-on card position, only one m2x5 screw is needed.



The table below show all the signals. Full size Mini-PCIe card are supported.

Pinout:

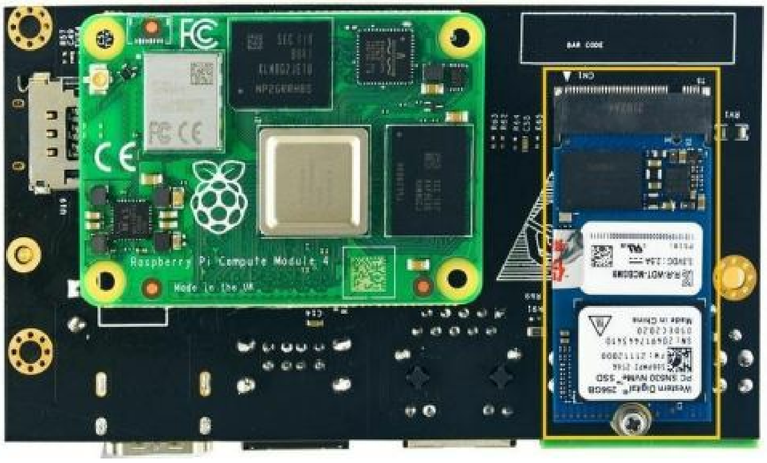
Signal	PIN#	PIN#	Signal
	1	2	4G_PWR
	3	4	GND
	5	6	USIM_PWR
	7	8	USIM_PWR
GND	9	10	USIM_DATA
	11	12	USIM_CLK
	13	14	USIM_RESET#
GND	15	16	
	17	18	GND
	19	20	
GND	21	22	PERST#
	23	24	4G_PWR
	25	26	GND
GND	27	28	
GND	29	30	UART_PCIE_TX
	31	32	UART_PCIE_RX
	33	34	GND
GND	35	36	USB_DM
GND	37	38	USB_DP
4G_PWR	39	40	GND
4G_PWR	41	42	4G_LED
GND	43	44	USIM_DET
SPI1_SCK	45	46	
SPI1_MISO	47	48	
SPI1_MOSI	49	50	GND
SPI1_SS	51	52	4G_PWR

NOTE:

1. All blank signals are NC (not connect).
2. 4G_PWR is the individual power supply for Mini-PCIe card. It can be shut down or turn on by the GPIO6 of CM4, the control signal is high active.
3. 4G_LED signal is connected to LED2 internally, refer to section of 2.2.8.
4. SPI1 signals are used only for LoraWAN card, such as WM1302.

M.2

EdgeBox-RPI-200 equipped a M.2 socket of M KEY type. ONLY 2242 size NVME SSD card is support, NOT mSATA.



Drivers and Programming Interfaces

LED

The is a LED used as user indicator, refer to 2.2.8. Use LED2 as an example to test the function.

- $ sudo -i #enable root account privileges
- $ cd /sys/class/gpio
- $ echo 21 > export #GPIO21 which is user LED of LED2
- $ cd gpio21
- $ echo out > direction
- $ echo 0 > value # turn on the user LED, LOW active

  OR
- $ echo 1 > value # turn off the user LED

Serial Port (RS232 and RS485)

There are two individual serial ports in the system. The /dev/ ttyACM1 as RS232 port and /dev/ ttyACM0 as RS485 port. Use RS232 as a example.

$ python

```
>>> import serial
>>> ser=serial.Serial('/dev/ttyACM1',115200,timeout=1) >>> ser.isOpen()
true
>>> ser.isOpen()
>>> ser.write('1234567890')
```

Cellular over Mini-PCIe (Optional)

Use Quectel EC20 as a example and follow the steps:

1. Insert the EC20 into Mini-PCIe socket and micro sim card in related slot, connect the antenna.
2. Log in the system via console use pi/raspberry.
3. Turn on the power of Mini-PCIe socket and release the reset signal.

- $ sudo -i #enable root account privileges
- $ cd /sys/class/gpio
- $ echo 6 > export #GPIO6 which is POW_ON signal
- $ echo 5 > export #GPIO5 which is reset signal
- $ cd gpio6
- $ echo out > direction
- $ echo 1 > value # turn on the power of Mini PCIe

  AND
- $ cd gpio5
- $ echo out > direction
- $ echo 1 > value # release the reset signal of Mini PCIe

NOTE: Then the LED of 4G is start to flash.

Check the device:

$ lsusb

```
Bus 001 Device 005: ID 2c7c:0125 Quectel Wireless Solutions Co., Ltd. EC25 LTE modem
```

$ dmesg

```
[ 185.421911] usb 1-1.3: new high-speed USB device number 5 using dwc_otg
[ 185.561937] usb 1-1.3: New USB device found, idVendor=2c7c, idProduct=0125, bcdDevice= 3.18
[ 185.561953] usb 1-1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 185.561963] usb 1-1.3: Product: Android
[ 185.561972] usb 1-1.3: Manufacturer: Android
[ 185.651402] usbcore: registered new interface driver cdc_wdm
[ 185.665545] usbcore: registered new interface driver option
[ 185.665593] usbserial: USB Serial support registered for GSM modem (1-port)
[ 185.665973] option 1-1.3:1.0: GSM modem (1-port) converter detected
[ 185.666283] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB2 [ 185.666499] option 1-1.3:1.1:
GSM modem (1-port) converter detected
[ 185.666701] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB3 [ 185.666880] option 1-1.3:1.2:
GSM modem (1-port) converter detected
[ 185.667048] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB4 [ 185.667220] option 1-1.3:1.3:
GSM modem (1-port) converter detected
[ 185.667384] usb 1-1.3: GSM modem (1-port) converter now attached to ttyUSB5 [ 185.667810] qmi_wwan 1-
1.3:1.4: cdc-wdm0: USB WDM device
[ 185.669160]qmi_wwan 1-1.3:1.4 wwan0: register ‘qmi_wwan’ at usb-3f980000.usb-1.3, WWAN/QMI
device,xx:xx:xx:xx:xx:xx
```

NOTE: xx:xx:xx:xx:xx: xx is the MAC address

$ ifconfig -a

```
…… wwan0: flags=4163<UP, BROADCAST,RUNNING, MULTICAST> mtu 1500
inet 169.254.69.13 netmask 255.255.0.0 broadcast 169.254.255.255 inet6 fe80::8bc:5a1a:204a:1a4b prefixlen 64
scopeid 0x20<link> ether 0a:e6:41:60:cf:42 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 165 bytes 11660 (11.3 KiB)
TX errors 0 dropped 0 overruns 0 carriers 0 collisions 0
```

How to use AT command

$ miniterm — Available ports:

- 1: /dev/ttyACM0 'USB Dual_Serial'
- 2: /dev/ttyACM1 'USB Dual_Serial'
- 3: /dev/ttyAMA0 'ttyAMA0'
- 4: /dev/ttyUSB0 'Android'
- 5: /dev/ttyUSB1 'Android'
- 6: /dev/ttyUSB2 'Android'
- 7: /dev/ttyUSB3 'Android'

Enter port index or full name:

$ miniterm /dev/ttyUSB5 115200

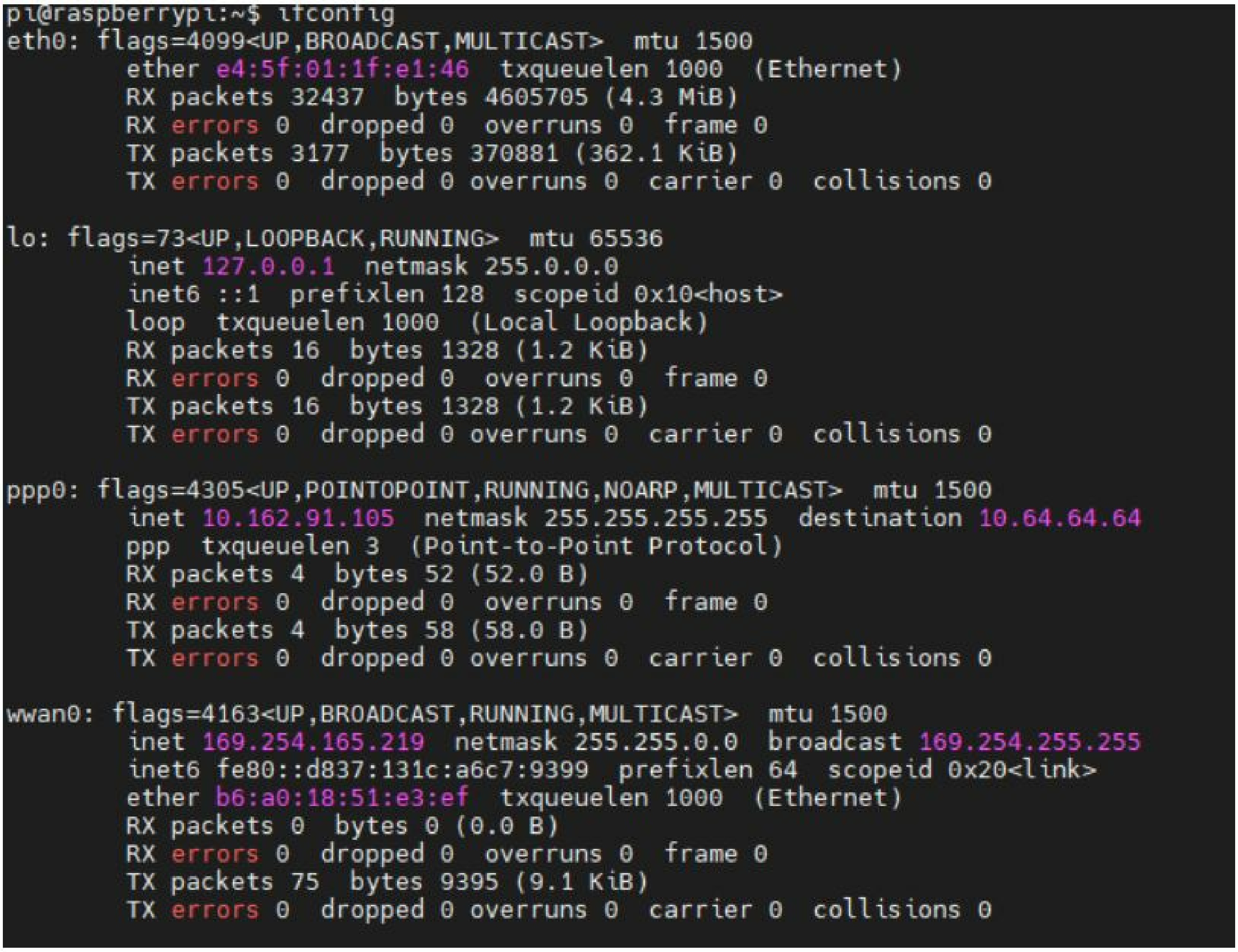
Some useful AT commands:

- AT //should return OK
- AT+QINISTAT //return the initialization status of (U)SIM card, the response should be 7
- AT+QCCID //returns the ICCID (Integrated Circuit Card Identifier) number of the (U)SIM card

How to dial

- $su root
- $ cd /usr/app/linux-ppp-scripts
- $./quectel-pppd.sh

Then the 4G led is flashing. If success, the return like this



```
pi@raspberrypi:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether e4:5f:01:1f:e1:46  txqueuelen 1000  (Ethernet)
        RX packets 32437  bytes 4605705 (4.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3177  bytes 370881 (362.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 1328 (1.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1328 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.162.91.105  netmask 255.255.255.255  destination 10.64.64.64
        ppp  txqueuelen 3  (Point-to-Point Protocol)
        RX packets 4  bytes 52 (52.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 58 (58.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wwan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 169.254.165.219  netmask 255.255.0.0  broadcast 169.254.255.255
        inet6 fe80::d837:131c:a6c7:9399  prefixlen 64  scopeid 0x20<link>
        ether b6:a0:18:51:e3:ef  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 75  bytes 9395 (9.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Add the router path

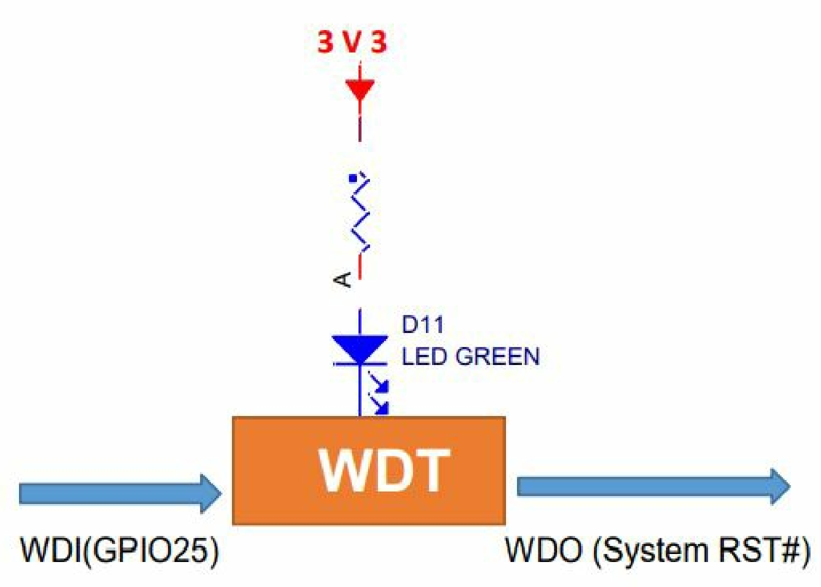
- $ route add default gw 10.64.64.64 or your gateway XX.XX.XX.XX

Then have a test with ping:

- $ ping google.com

WDT
Block Diagram of WDT

The WDT module have three terminals, input, output and LED indicator.



NOTE: The LED is optional and not available in earlier hardware version.

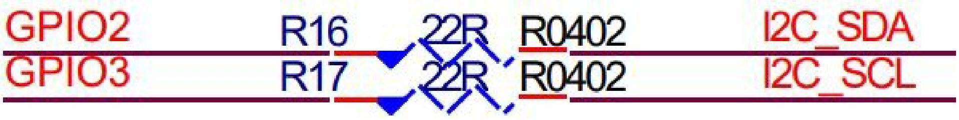
How it works

1. System POWER ON.
2. Delay 200ms.
3. Send WDO a negative pulse with 200ms low level to reset the system.
4. Pull up WDO.
5. Delay 120 seconds while the indicator flashing (typical 1hz).
6. Turn off the indicator.
7. Wait for 8 pulses at WDI to active WDT module and light the LED.
8. Get Into WDT-FEED mode, at least one pulse should be feed into WDI in at least every 2 seconds, if not, the WDT module should output a negative pulse to reset the system.
9. Goto 2.

RTC

RTC Chip information

New Revision: The chip of RTC is PCF8563 from NXP. It is mounted on the system I2C bus, the i2c address should be 0x51.

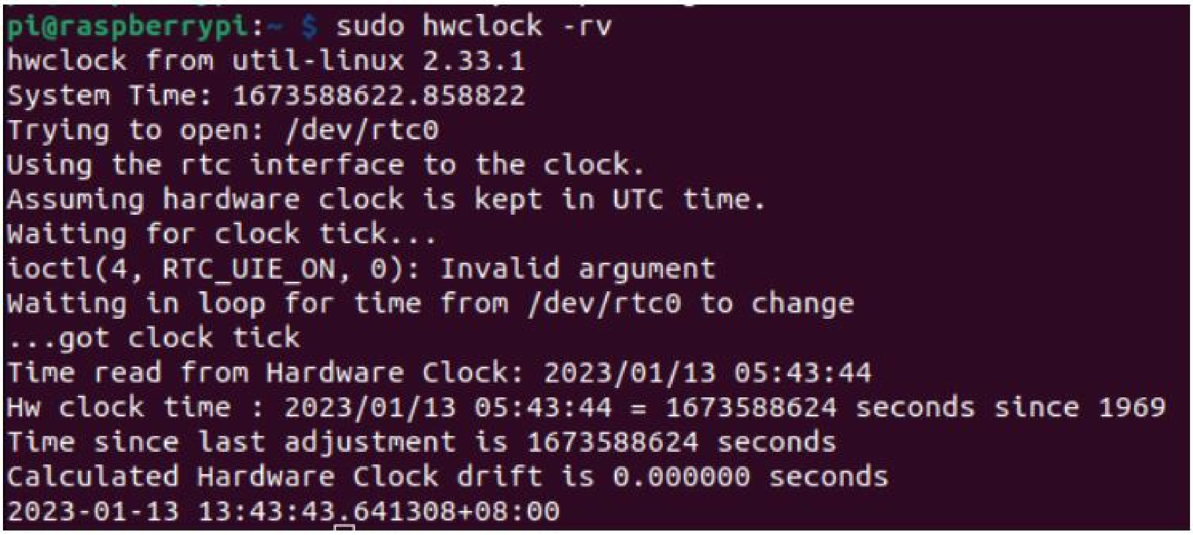


The OS itself has the driver inside, only we need are some configurations.

Enable RTC

- To Enable the RTC you need to:
  - $sudo nano /boot/config.txt
- Then add the following line at the bottom of the /boot/config.txt

  - dtoverlay=i2c-rtc,pcf8563
- Then reboot the system
  - $sudo reboot
- Then use the following command to check if the RTC is enabled:
  - $sudo hwclock -rv
- The Output should be:



```
pi@raspberrypi:~ $ sudo hwclock -rv
hwclock from util-linux 2.33.1
System Time: 1673588622.858822
Trying to open: /dev/rtc0
Using the rtc interface to the clock.
Assuming hardware clock is kept in UTC time.
Waiting for clock tick...
ioctl(4, RTC_UIE_ON, 0): Invalid argument
Waiting in loop for time from /dev/rtc0 to change
...got clock tick
Time read from Hardware Clock: 2023/01/13 05:43:44
Hw clock time : 2023/01/13 05:43:44 = 1673588624 seconds since 1969
Time since last adjustment is 1673588624 seconds
Calculated Hardware Clock drift is 0.000000 seconds
2023-01-13 13:43:43.641308+08:00
```

NOTE

1. make sure the i2c-1 driver point is open, and the point is closed default.
2. the estimated backup time of the RTC is 15 days.

Product Change NOTE

OLD Revision: The chip of RTC is MCP79410 from microchip. It is mounted on the system I2C bus. The i2c address of this chip should be 0x6f. To enable it you need to:

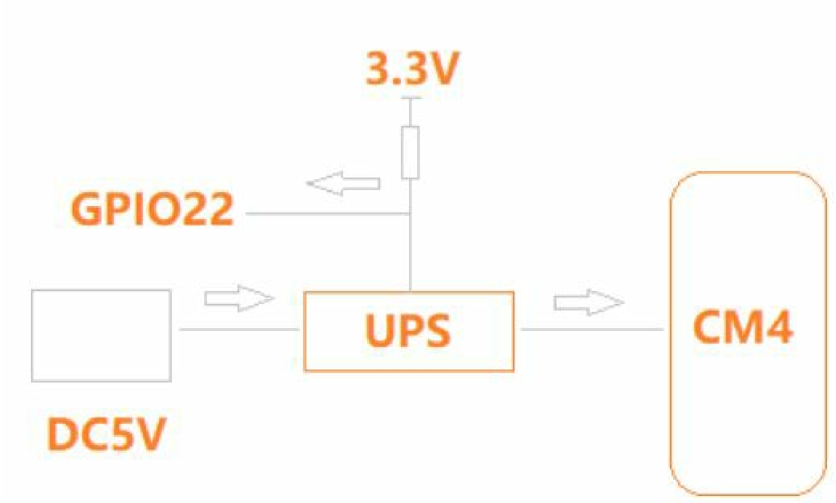
Open /etc/rc.local AND add 2 lines:

echo “mcp7941x 0x6f” > /sys/class/i2c-adapter/i2c-1/new_device hwclock -s

Then reset the system and the RTC is working

UPS for safe shut down (Optional)

The UPS module diagram is listed below.



The UPS module is inserted between the DC5V and CM4, a GPIO is used to alarm CPU when the 5V power supply is down. Then the CPU should do something urgent in a script before energy exhaustion of super capacitor and run a “$ shutdown” Another way to use this function is Initiate a shutdown when GPIO pin changes. The given GPIO pin is configured as an input key that generates KEY_POWER events. This event is handled by systemd-logind by initiating a shutdown. Systemd versions older than 225 need an udev rule enable listening to the input device: Use /boot/overlays/README as reference, then modify /boot/config.txt. dtoverlay=gpio-shutdown, gpio_pin=GPIO22,active_low=1

NOTE:

1. For UPS function please contact us for more information.
2. The alarm signal is active LOW.

Electrical specifications

Power consumption

The power consumption of the EdgeBox-RPI-200 strongly depends on the application, the mode of operation and the peripheral devices connected. The given values have to be seen as approximate values. The following table shows power consumption parameters of the EdgeBox-RPI-200:

Note: On condition of power supply 24V, no add-on card in sockets and no USB devices.

Mode of operation	Current(ma)	Power	Remark
Idle	**81**		
Stress test	**172**		**stress -c 4 -t 10m -v &**

UPS (Optional)

The backup time of UPS module is very depended on the system load of the system. Some typical conditions are listed below. The test module of CM4 is 4GB LPDDR4,32GB eMMC with Wi-Fi module.

Mode of operation	Time(second)	Remark
Idle	55	
Full load of CPU	18	stress -c 4 -t 10m -v &

Mechanical Drawings

- WWW.SEEEDSTUDIO.COM

Documents / Resources

 seeed studio EdgeBox-RPI-200 User Manual V2.1 Raspberry PI CM4 Based Edge computer Revision History Revision Date Changes 1.0 17-08-2022 Created 2.1 13-01-2022 Product Change Notice	Seeedstudio EdgeBox-RPI-200 EC25 Raspberry PI CM4 Based Edge computer [pdf] User Manual EdgeBox-RPI-200 EC25 Raspberry PI CM4 Based Edge computer, EdgeBox-RPI-200, EC25 Raspberry PI CM4 Based Edge computer, Raspberry PI CM4 Based Edge computer, CM4 Based Edge computer, Based Edge computer

References

- Seeed Studio Bazaar, The IoT Hardware enabler.
- Seeed Studio Bazaar, The IoT Hardware enabler.
- User Manual