



EdgeBox-RPI User Manual V1.0

Raspberry CM3+ Based Edge computer



Revision History

Revision	Date	Changes
1.0	10/16-2019	Created

Contents

- 1. Introduction
 - 1.1 Features
 - 1.2 Interfaces
 - 1.3. Block diagram
- 2. Installation
 - 2.1 Mounting
 - 2.2 External Interfaces
 - 2.2.1 Power supply
 - 2.2.2 Serial Port (RS232 and RS485)
 - 2.2.3 CAN-FD
 - 2.2.4 HDMI
 - 2.2.5 Ethernet
 - 2.2.6 USB host
 - 2.2.7 Console(USB typeC)
 - 2.2.8 LED
 - 2.2.9 SMA Connector
 - 2.2.10 Micro SIM card slot
 - 2.2.11 Mini-PCle
- 3. Drivers and Programming Interfaces
 - 3.1 LED
 - 3.2 Serial Port (RS232 and RS485)
 - 3.3 Cellular over Mini-PCle
 - 3.4 WDT
 - 3.5 RTC
 - 3.6 Encryption
 - 3.7 CAN-FD
- 4. Electrical specifications
- 5. Mechanical Drawings

1. Introduction

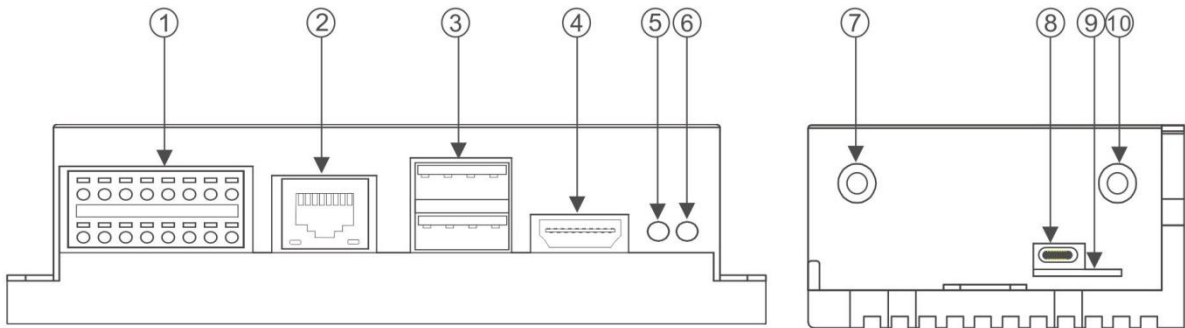
EdgeBox-RPI is a rugged fan less edge computing system fitted with Raspberry Pi Computer Module 3+(CM3+), proven, out-of-the-box controller for industry environment. It can be used to connect the field networks with cloud or IoT applications.

1.1 Features

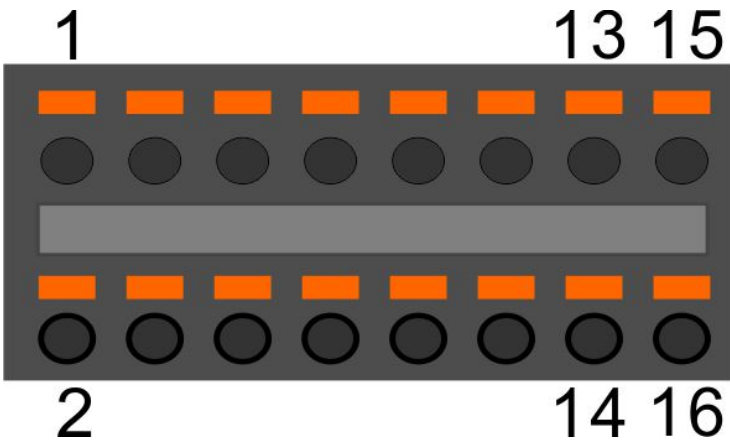
- State-of-the-art Aluminium chassis for Harsh environment
- Integrated passive heat sink
- Built-in mini PCIe socket for RF module, such as 4G, WI-FI, lora or zigbee
- SMA antenna holes x2
- Modbus over RS485 or Ethernet support
- CAN-FD interface up to 8 Mbit/s for distributed I/O or motion control
- Encryption chip ATECC608A
- Hardware Watchdog
- RTC with Super Capacitor
- 35mm DIN Rail support
- Wide power supply from 9 to 36V DC

These features make the EdgeBox-RPI 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 A53 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.

1.2 Interfaces



① Multi-Func phoenix connector

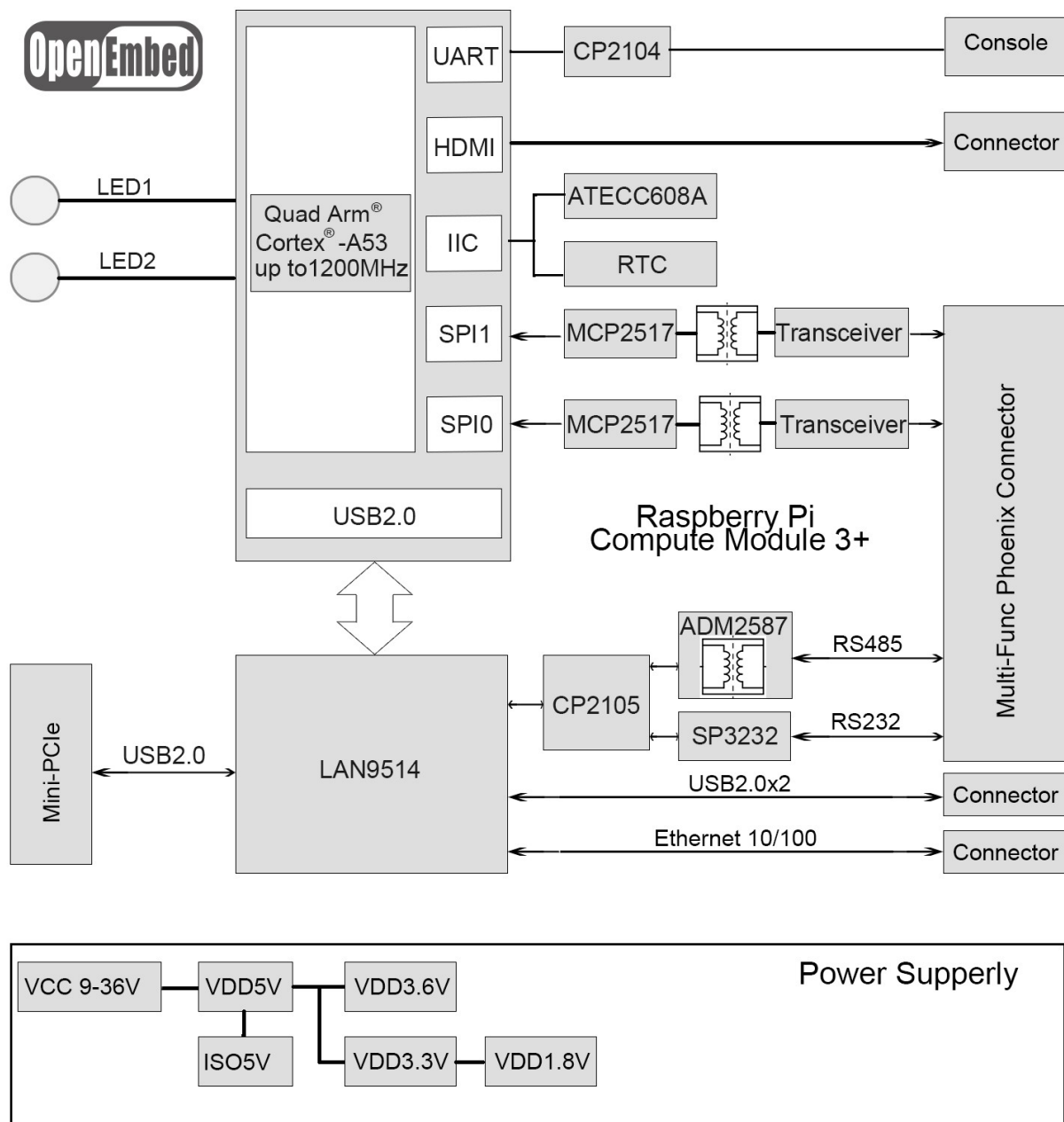


Note	Func name	PIN #	PIN#	Func name	Note
	POWER_IN	1	2	PE	
	GND	3	4	GND	
	RS485_GND	5	6	RS232_RX	
	RS485_A	7	8	RS232_TX	
	RS485_B	9	10	GND	
	CAN_GND	11	12	CAN_GND	
	CAN1_L	13	14	CAN2_L	
	CAN1_H	15	16	CAN2_H	

- 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

1.3 Block Diagram

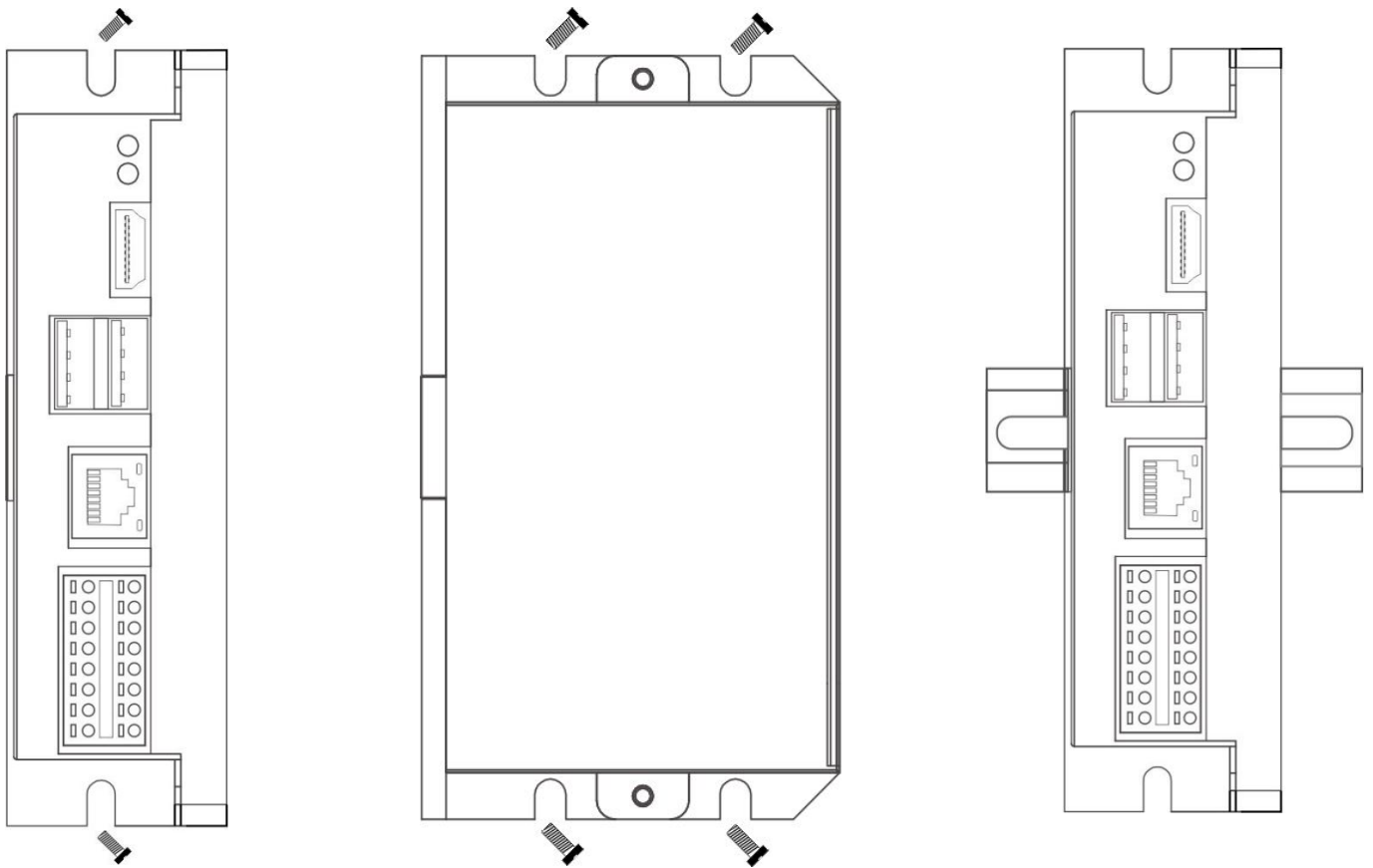
The processing core of the EdgeBox-RPI is a Raspberry CM3+ board. A OpenEmed specific base board implements the specific features. Refer to next figure for the block diagram.



2. Installation

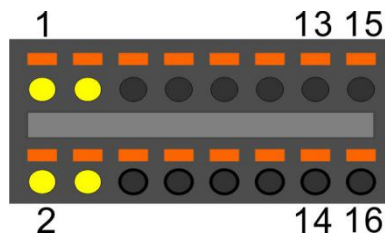
2.1 Mounting

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



2.2 Connectors and Interfaces

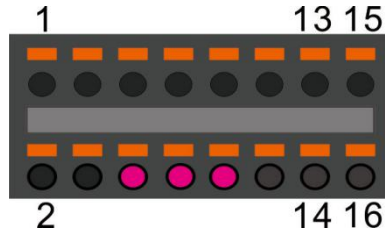
2.2.1 Power supply



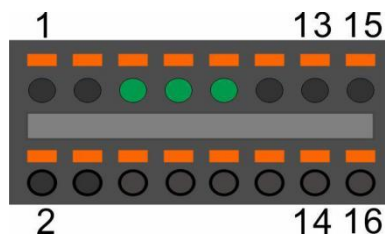
Pin#	Signal	Description
1	POWER_IN	DC 9-36V
2	PE	Functional earth
3	GND	Ground (Reference potential)
4	GND	Ground (Reference potential)

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

2.2.2 Serial Port (RS232 and RS485)



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



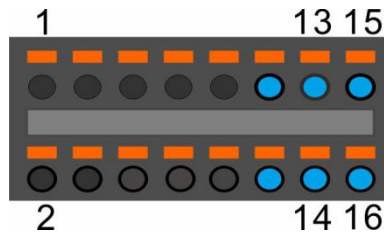
Pin#	Signal	Description
5	RS485_GND	RS485 Ground (isolated from GND)
7	RS485_A	RS485 difference line high
9	RS485_B	RS485 difference line low

The RS485_GND signal is isolated with “GND” signal. If a shielded twisted pair wire is used ,the RS485

_GND is connected to the shield.

NOTE:The 120 Ohm termination resistor for RS485 is enabled inside.

2.2.3 CAN-FD



Pin#	Signal	Description
11	CAN_GND	CAN Ground (isolated from GND)
13	CAN1_L	CAN1 difference line low
15	CAN1_H	CAN1 difference line high
12	CAN_GND	CAN Ground (isolated from GND)
14	CAN2_L	CAN2 difference line low
16	CAN2_H	CAN2 difference line high

The CAN_GND signal is isolated with “GND” signal. If a shielded twisted pair cable is used ,the CAN_GND should be connected to the shield. The signals of CAN1 and CAN2 are independent, but the CAN_GND of CAN1 and CAN2 is connected internally.

NOTE: The 120 Ohm termination resistor for CAN bus is enabled inside.

2.2.4 HDMI

Directly connected to the Raspberry PI CM3+ board with TVS array.

2.2.5 Ethernet

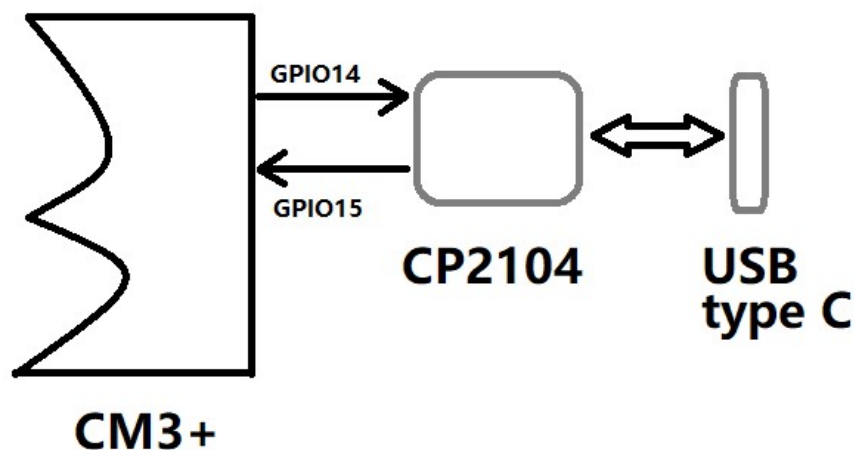
Ethernet interface is same as Raspberry PI,10/100-BaseT supported, available through the shielded modular jack. Twisted pair cable or shielded twisted pair cable can be used to connect to this port.

2.2.6 USB HOST

There are two USB interfaces at the connector panel. The two ports share the same electronic fuse. The GPIO26 is used to turn USB power on, on board 10k ohm resistance pull up and high active.

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

2.2.7 Console(USB typeC)



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:

<https://www.silabs.com/products/interface/usb-bridges/classic-usb-bridges/device.cp2104>

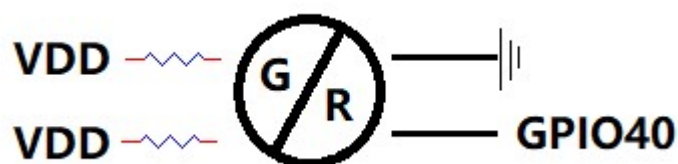
This port is used as a Linux console default. You can log into the OS use the settings of 115200, 8n1 (Bits: 8, Parity: None, Stop Bits: 1, Flow Control: None). A terminal program such as putty is needed, too.

The default user name is pi and password is raspberry.

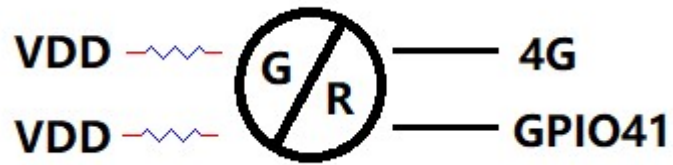
2.2.8 LED

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

LED1: green as power indicator and red as user led connected to GPIO40, low active, programmable.



LED2: green as 4G indicator and red as user led connected to GPIO41, low active, programmable.



2.2.9 SMA Connector

There are two SMA Connector holes for antennas. The antenna types are very depend on what modules fitted into the Mini-PCle socket.

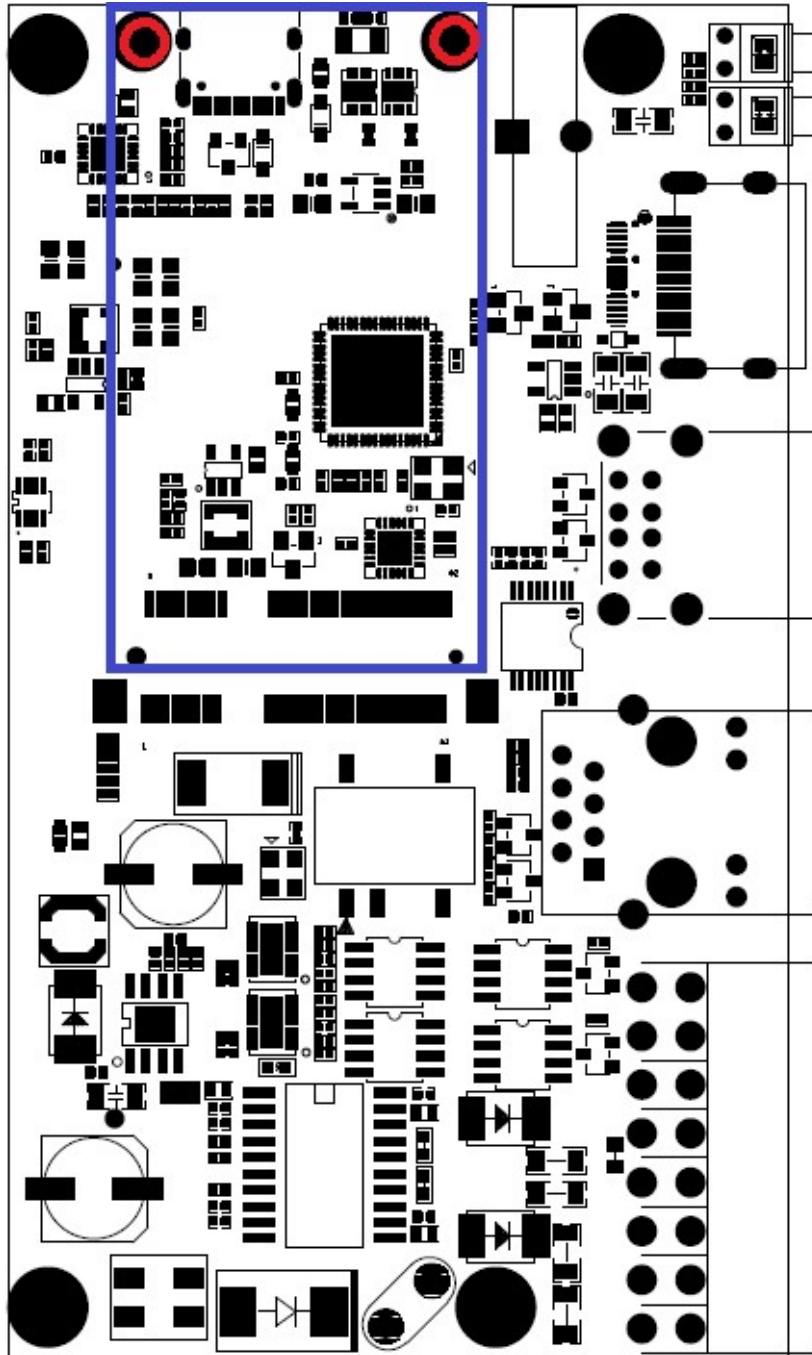
2.2.10 Micro SIM card slot

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

NOTE: only micro sim card is accepted,pay attention to the card size.

2.2.11 Mini-PCle

The blue area is the rough Mini-PCle add-on card position,and the red for two m2x5 screws.



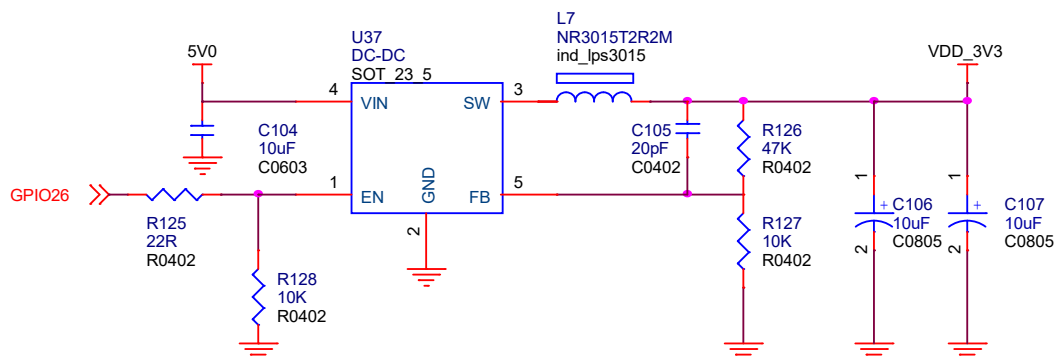
The table below show all the signals. Full size Mini-PCIe card are supported.

Signal	PIN#	PIN#	Signal
	1	2	VDD_3V3
	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	VDD_3V3
	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
VDD_3V3	39	40	GND
VDD_3V3	41	42	4G_LED
GND	43	44	USIM_DET
	45	46	
	47	48	
	49	50	GND
	51	52	VDD_3V3

NOTE 1: All blank signals are NC(no connect).

NOTE 2: VDD_3V3 is a individual power supply for Mini-PCle card . It can be shout down or turn on by the GPIO26 of CM3+,and the control signal is high active.



NOTE 3:4G_LED signal is connected to LED2 internally, refer to section of 2.2.8.

NOTE 4:PIN30,PIN32 share the same channel UART signals with the RS232 connected to Multi-Func phoenix connector, they can not be used at same time and the phoenix connector is default.

3. Drivers and Programming Interfaces

The original Raspbian operating system from official Raspberry Pi is burned into the eMMC flash of CM3+ before delivery, as while as some additional drivers and software for interfaces of the EdgeBox-RPI ,eg. RTC, CAN , RS232 and RS485.

3.1 LED

There are two LEDs used as indicators, refer to 2.2.8 .
Use LED1 as an example to test the function.

```
$ python
$ import RPi.GPIO as GPIO
$ GPIO.setmode(GPIO.BCM)
$ GPIO.setup(40,GPIO.OUT)
$ GPIO.output(40,0)
```

This step can turn on the red led of LED1.

3.2 Serial Port (RS232 and RS485)

There are two individual serial ports in the system. The /dev/ttyUSB1 as RS232 port and /dev/ttyUSB0 as RS485 port. Use RS232 as an example.

```
$ python
$ import serial
$ ser=serial.Serial('/dev/ttyUSB1',115200,timeout=1)
$ ser.isOpen()
true
$ ser.isOpen()
$ ser.write('1234567890')
10
```

3.3 Cellular over Mini-PCle

Use Quectel EC20 as an example and follow the steps :

1. Insert the EC20 into Mini-PCle socket and micro sim card in related slot.
2. Log in the system via console use pi/raspberry.
3. Turn on the power of Mini-PCle socket.

```
$ python
$ import RPi.GPIO as GPIO
$ GPIO.setmode(GPIO.BCM)
$ GPIO.setup(26,GPIO.OUT)
$ GPIO.output(26,1)
```

4. 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
```

xx:xx:xx:xx:xx:xx is the MAC address.

Now, you can config the port and dial.

```
$ 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  carrier 0  collisions 0
```


4.Electrical specifications

5.Mechanical Drawings