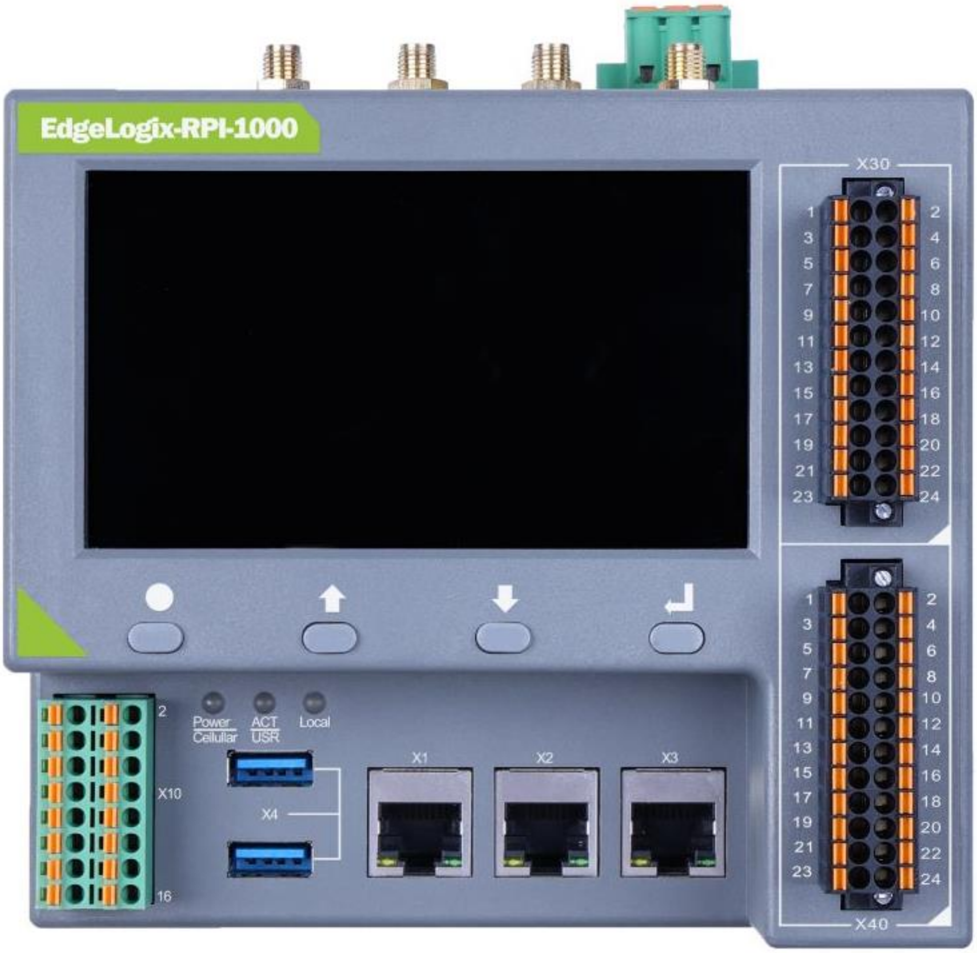
——

EdgeLogix-RPI-1000 User Manual **V1.2**

Industrial Edge controller for IIoT



Revision History

Revision	Date	Changes
1.0	24-08-2022	Initial
1.1	09-03-2023	Update pinout maps, LoraWAN module example
1.2	30-03-2023	

Explanation of symbols used

The following symbols are used in these instructions:

NOTE

NOTE indicates tips, recommendations and useful information on specific actions and facts.

NOTICE

NOTICE indicates a situation which may lead to property damage if not avoided.

CAUTION

CAUTION indicates a dangerous situation of risk

Contents

1. Introduction

EdgeLogix-RPI-1000, modular, open-architecture edge controllers manage complex interfaces across assets and devices or into the cloud directly, with legacy and next generation industry control system. EdgeLogix-RPI-1000 provides performance and scalability for a wide range of industrial applications, including motion control, networking, IO and IIoT in a compact model, as while as the advantages of traditional IEC-61131-3 programming with the flexibility of Linux.

The EdgeLogix-RPI-1000 Series covers all the functions required of logic computing, including easy operation and compatibility with a variety of control and measurement applications. This flexible device for ultra-reliable measurement and control of industry users the customized logic and control needed to meet application requirements. Industry-leading configurability and programmability fulfill standard to complex requirements while secure, built-in flow measurement calculations make it easier than ever to prove compliance. In addition, cloud-enabled functionality for licensing simplifies day-to-day operations. Digitally transform your operation by streamlining complex processes with the agile, intuitive EdgeLogix-RPI-1000. This new controller is also part of Seeed Studio’s next-generation EdgeLogix-RPI-1000 Series measurement and control platform that offers a common set of configuration tools, to expedite setup and facilitate commonly performed tasks.

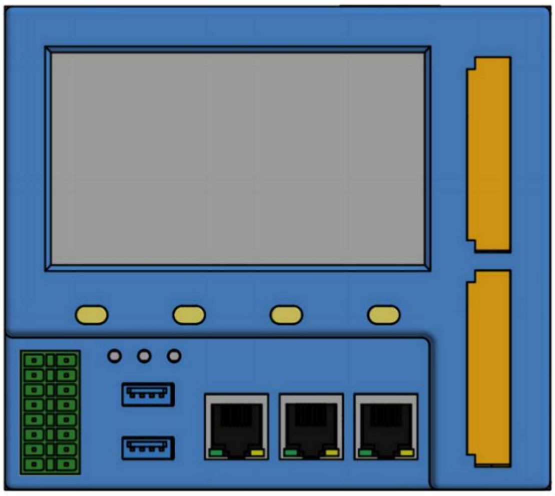
1.1 Features

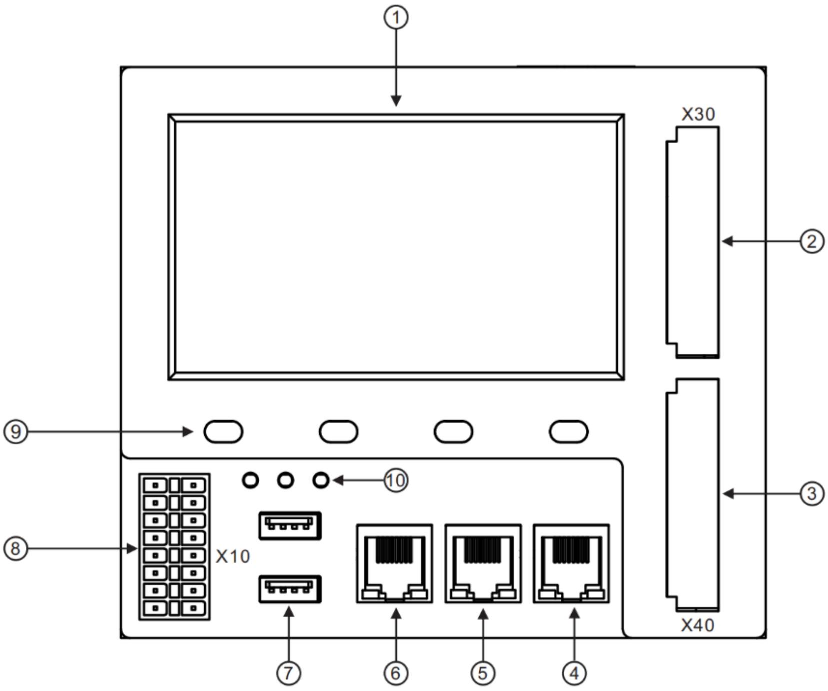
- Rugged, reduced-maintenance hardware
- High isolation, surge, and short circuit protection
- Open architecture support custom programming
- Ethernet, I/O ,4G/LTE, CANopen and Modbus bridging
- Natively Supports Modbus & CANopen Protocols
- Cloud Connectivity to IIoT Cloud Platforms
- Display for commissioning and diagnostics
- IEC 61131-3 compliant programs support (under developing).
- Flexible of local BUS extension
- Neural Processing Unit (NPU) enables Artificial Intelligence for automation
- Wide power supply from 10.8 to 36V DC

These features make the EdgeLogix-RPI-1000 designed as a cost-effective controller that provides the functions required for a variety of field automation applications. The EdgeLogix-RPI-1000 monitors, measures, and controls equipment in a remote environment. It is ideal for applications requiring flow computation; Proportional, Integral, and Derivative (PID) control loops; logic sequencing control; and a gateway with flexible wireless and field sensors expansion.

1.2 Interfaces summary

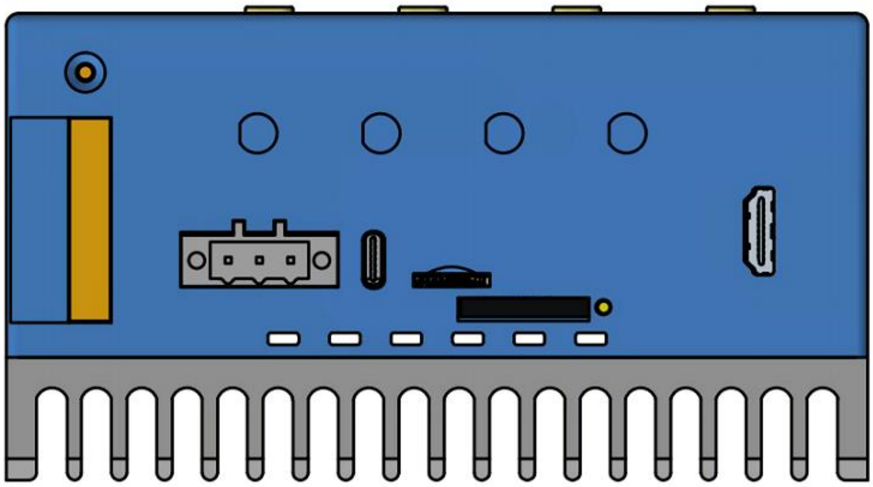
1.2.1 Front view

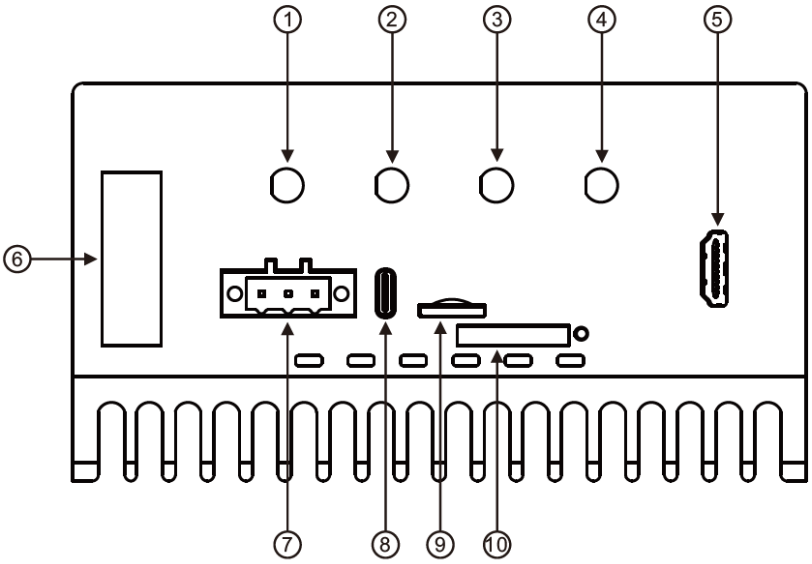




1. 4.3-inch LCD panel
2. X30, up connector of Signal Board
3. X40, down connector of Signal Board
4. X3, Ethernet port 3
5. X2, Ethernet port 2
6. X1, Ethernet port 1
7. Dual USB 2.0 port
8. X10, Left connector, Multi-Func phoenix connector
9. Key pad
10. 3x dual color LED

1.2.2 Top view

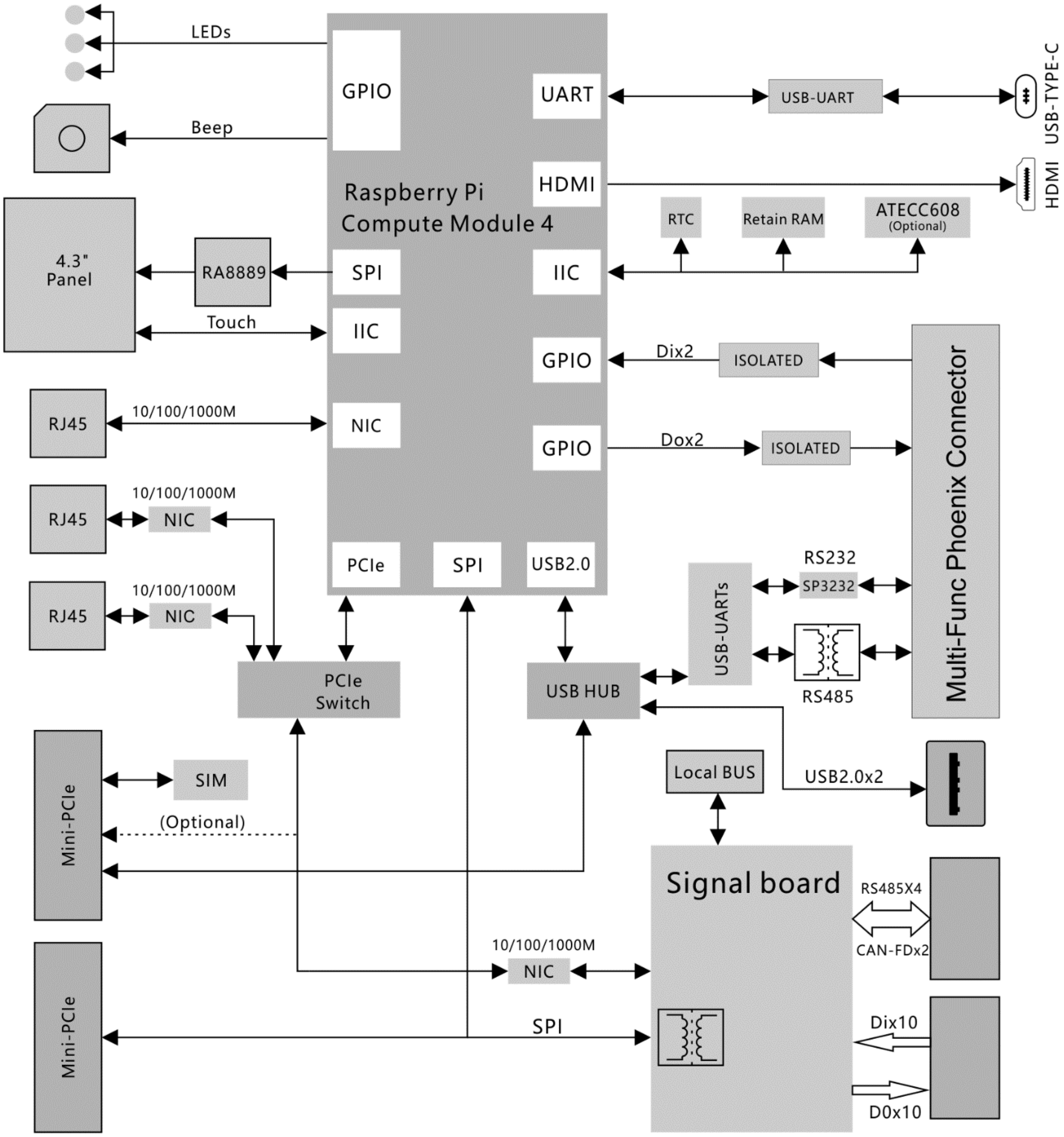




1. Antenna A4
2. Antenna A3
3. Antenna A2
4. Antenna A1
5. HDMI port to monitor
6. LocalBUS port
7. Main power supply
8. Console of Type-C
9. TF card slot
10. SIM card slot

1.3 Block Diagram

The processing core of the EdgeLogix-RPI-1000 is a Raspberry CM4 board. A customized carrier board implements the specific features. Refer to next figure for the block diagram.



2. Installation and Wiring

2.1 Mounting

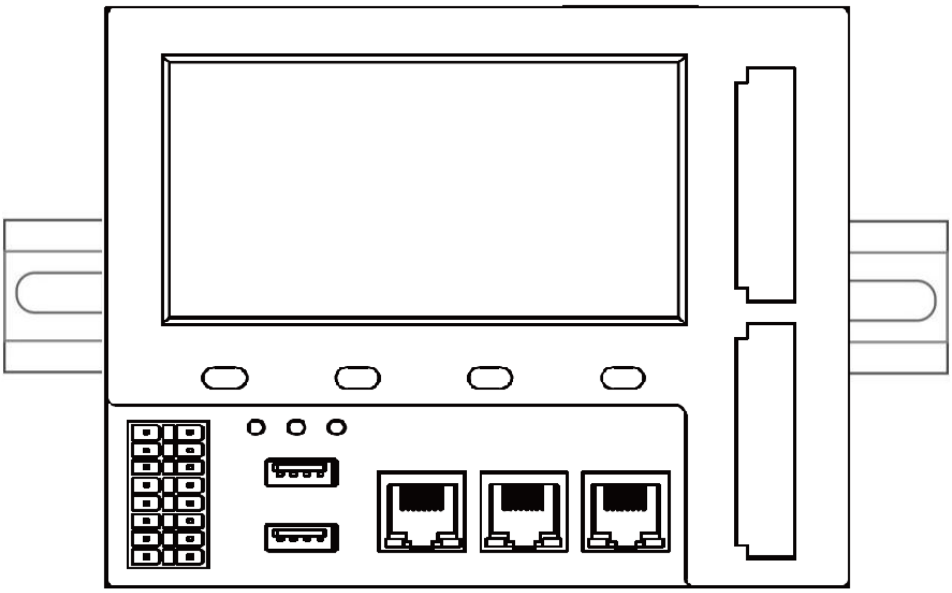
The 35mm DIN-rail mount is the primary method, as while as the wall mount.

⚠

• The entire power supply must be disconnected and electrostatic discharge must take place on the housing or ground connection before removing any covers or components from the device and installing or removing any accessories, hardware or cables.

• Remove the power cable from the device.

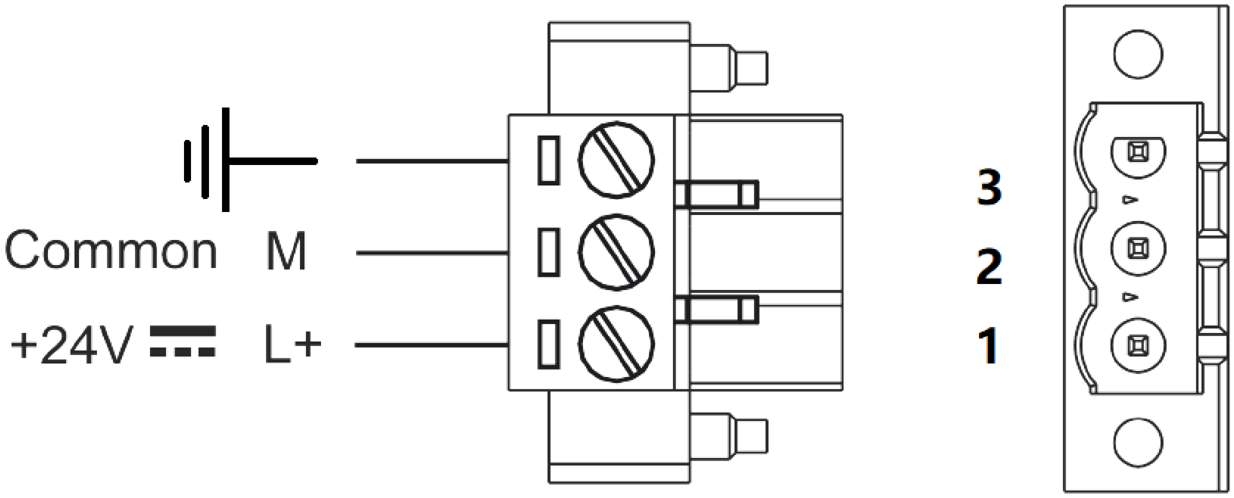
• All covers and components, accessories, hardware and cables must be installed.



2.2 Connectors and Interfaces

2.2.1 Power supply

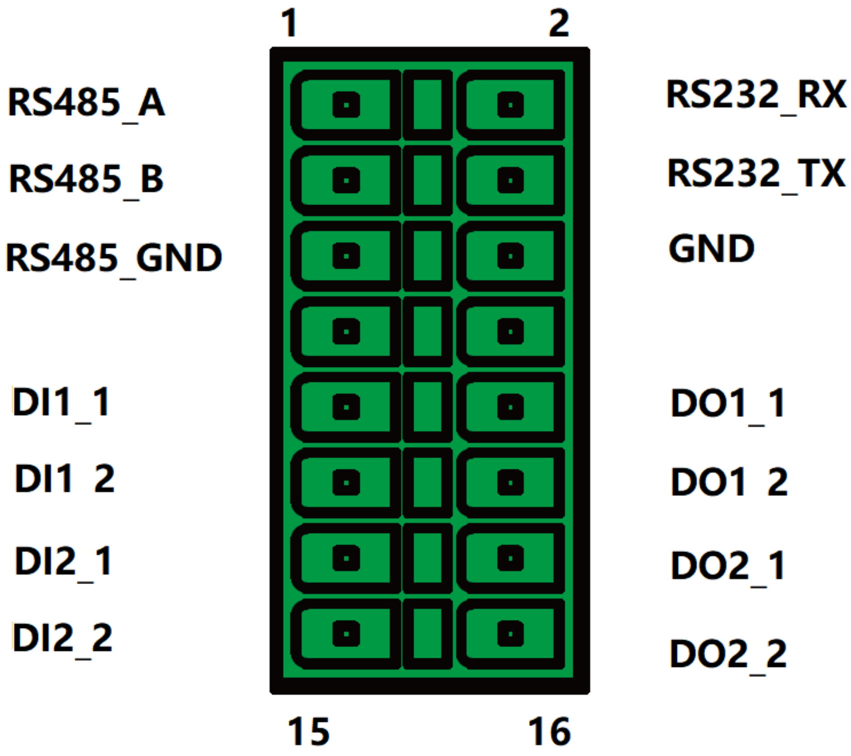
Connect the device to the voltage supply according to the following figure.



1. 24V, main power supply+
2. GND, main power supply-
3. EARTH, connect to earth

2.2.2 Left connector

Left connector is a Multi-Func phoenix connector.



NOTE 1: 24awg to 16awg cable are suggested.

NOTE 2: All RS485 signals are isolated with other signals.

NOTE 3: All DO and DI signals are isolated.

Pin#	Signal of terminal	PIN Level of active	PIN of GPIO from BCM2711	NOTE
1	RS485_A			
3	RS485_B			
5	RS485_GND			
2	RS232_RX			
4	RS232_TX			
6	RS232_GND			
09	DO1_1	HIGH	GPIO24	
11	DO1_2			
13	DO2_1	HIGH	GPIO25	
15	DO2_2			
10	DI1_1	HIGH	GPIO17	
12	DI1_2			
14	DI2_1	HIGH	GPIO27	
16	DI2_2			

◆ Characteristics of Isolated RS485 Interface

- Can used as Modbus/RTU Master or Modbus/RTU Extension .
- Supported Function Codes: #01, #02, #03, #04, #05, #06, #07, #0F, #10 .
- Maximum 32 devices on bus (1 master and 31 extensions) .
- Built-in asymmetrical protection against transient voltages resulting from electro-static discharge (ESD), electrical fast transients (EFT), and lighting.
- Terminal resistor of 120 OHM has been installed default.

◆ Characteristics of RS232 Interface

- The RS-232 serial interface communication standard has been in use for many
- years. It is one of the most widely used connections for serial data transmitting
- because it is simple and reliable.
- The RS232 serial interface standard still retains its popularity and remains in widespread
- use. It is still found on some computers and many interfaces, often being
- used for applications ranging from data acquisition to supply a serial data communication
- facility in general computing environments.
- The interfaces intended to operate over distances of up to 15 meters.

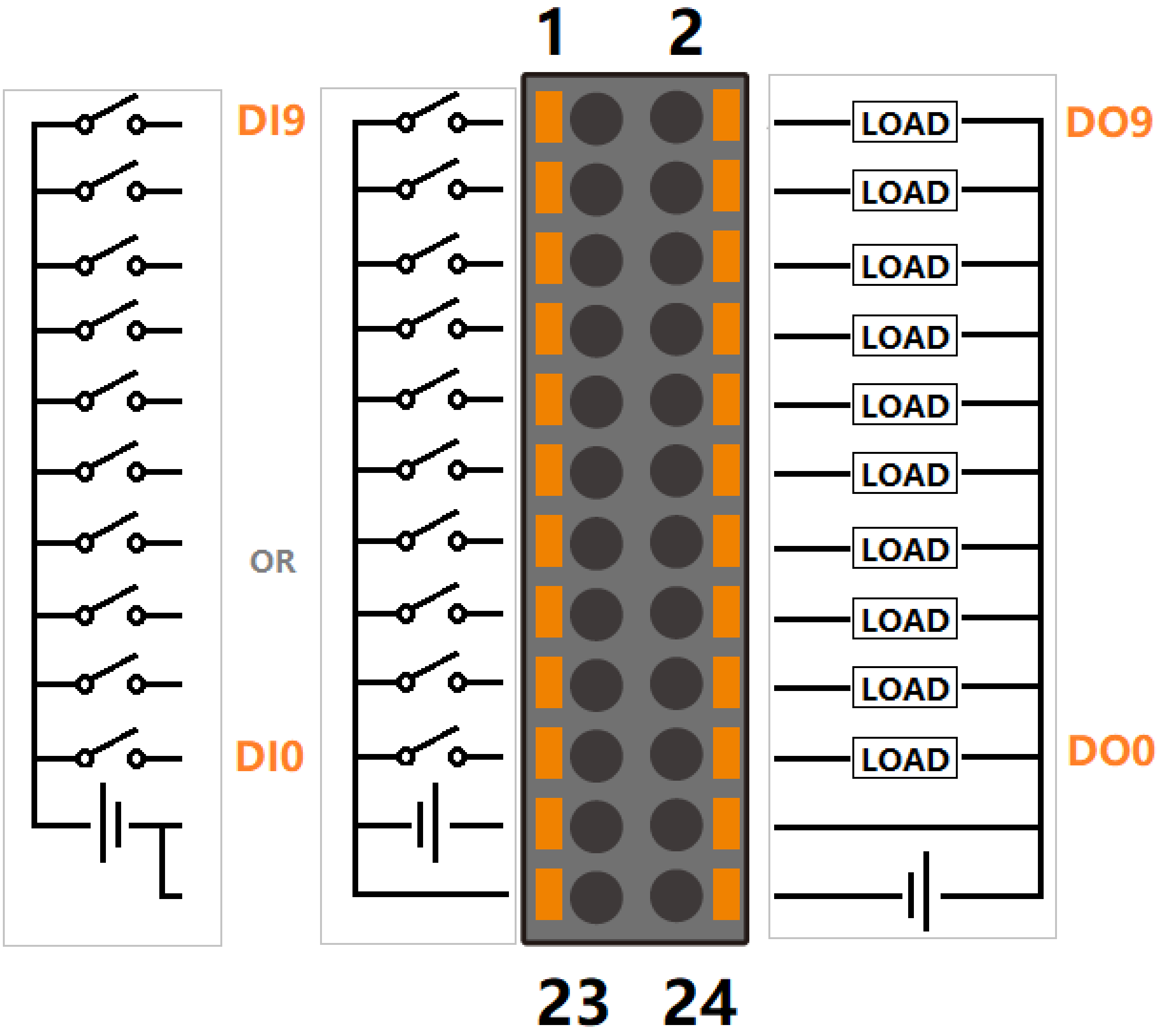
◆ Characteristics of I/O Interface

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

2.2.3 Connectors of Signal Board

The signal board is fully isolated with main PCB board. It contains X30, X40 and X6 of connectors. X30 is used as DIO signals. The following figure shows details of wiring.

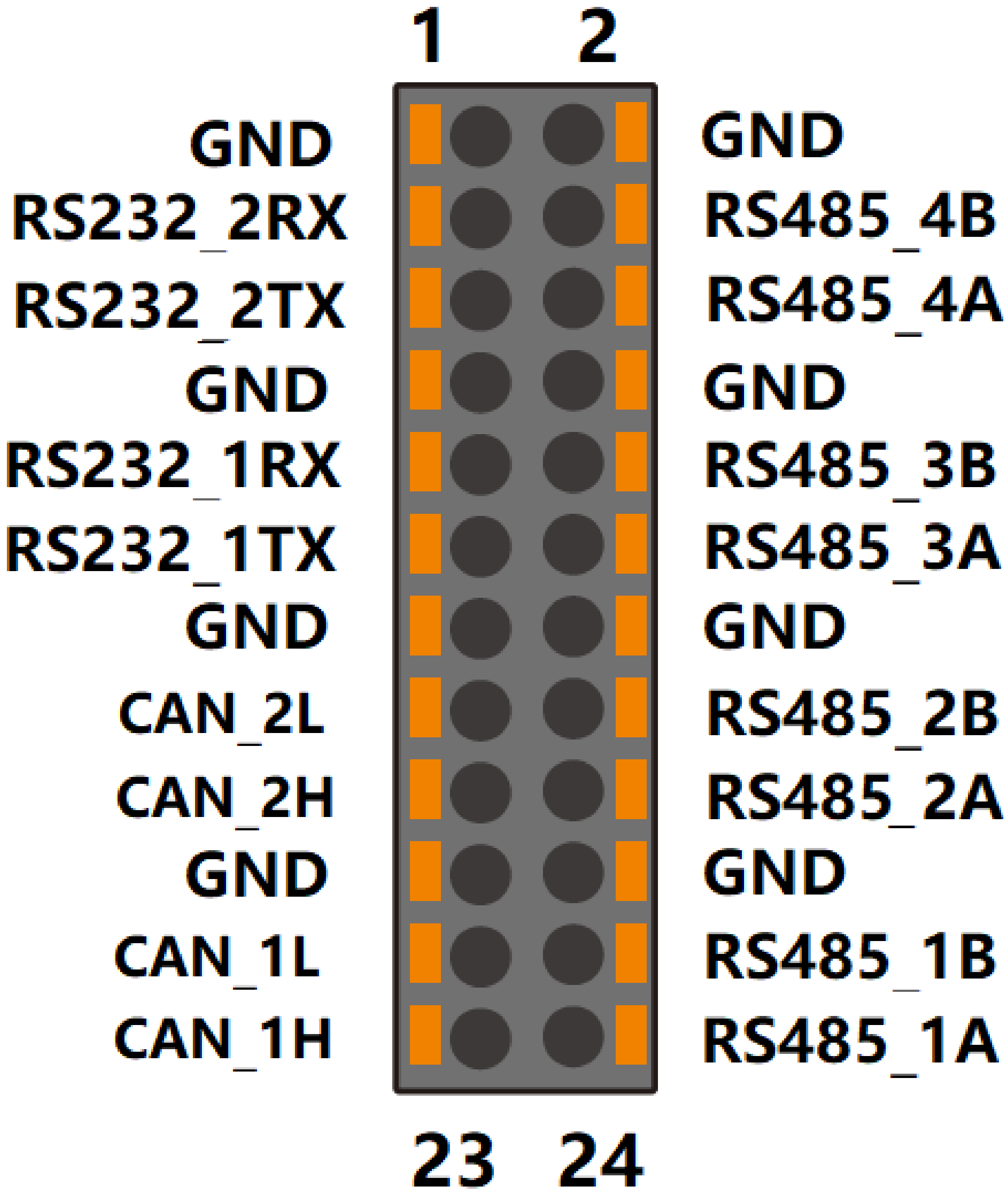
X30 wiring



◆ Characteristics of DI and DO Interface

- The power supply of DI should be 5-36V DC,24V default.
- The power supply of DO should be 10.8-60V DC,24V default, and the current of each channel is 1A.

X40 wiring



NOTE: All “GND” signals are connected together and isolated with main power island.

X6 connector

X6 connector is used for Local Bus extensions, such as DO, DI, AO, AO or RTD module can be connected in this bus.

2.2.4 HDMI

Directly connected to the Raspberry PI CM4 board with TVS array. The default display in EdgeLogix-RPI-1000 conforms to the HDMI standard.

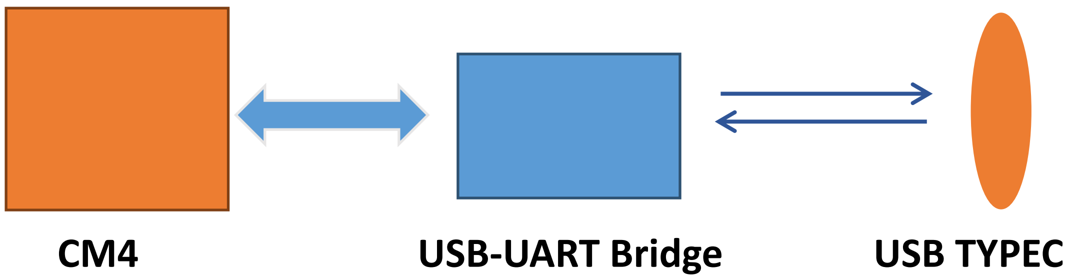
2.2.5 Ethernet

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.

2.2.6 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 1600ma.

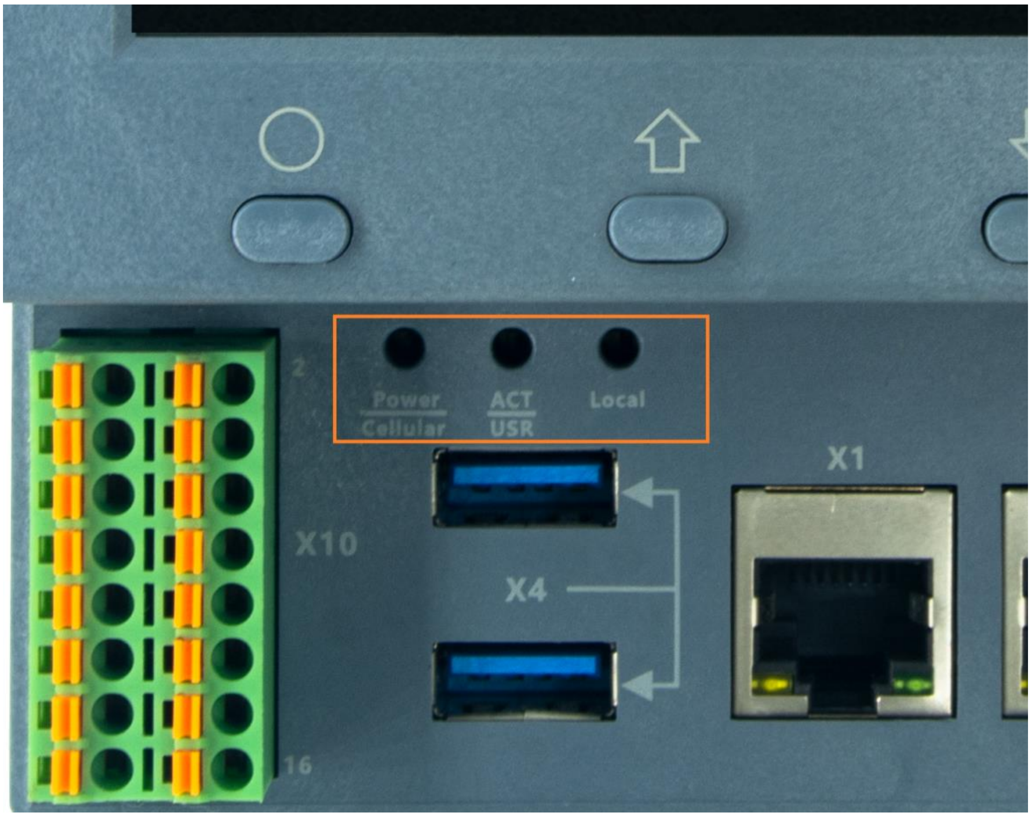
2.2.7 Console (USB TYPEC)



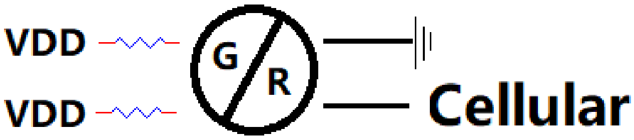
The design of console used a USB-UART converter, most OS of the computer have the driver, if not, the link below may be useful:
http://www.wch-ic.com/products/CH9102.html
This port is used as a Linux console default. You can log into the OS use the settings of 115200,8n1(Bits: 8, Parity: None, Stop Bits: 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

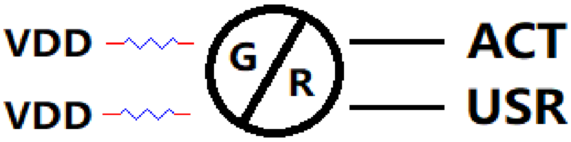
EdgeLogix-RPI-1000 use three green/red dual color LED as outside indicators.



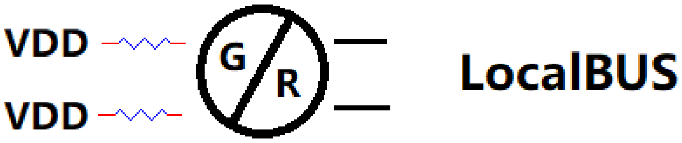
LED1: green as power indicator and red when 4G/LTE active.



LED2: green as signal indicator and red as user programmable led connected to GPIO13, high active and programmable.

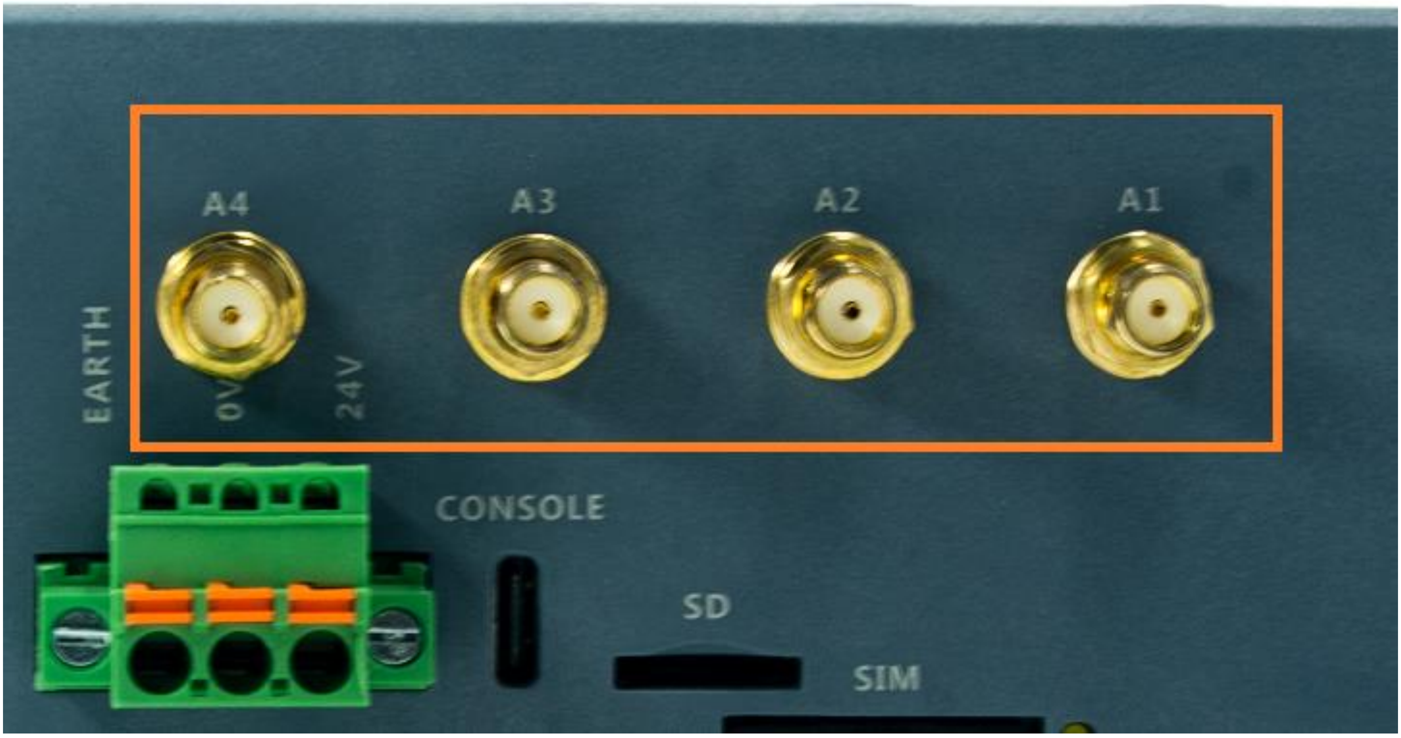


LED3: LED3 is used for Local BUS, it is to be defined in future. It indicates the communication between main and extensions.



2.2.9 SMA Connector

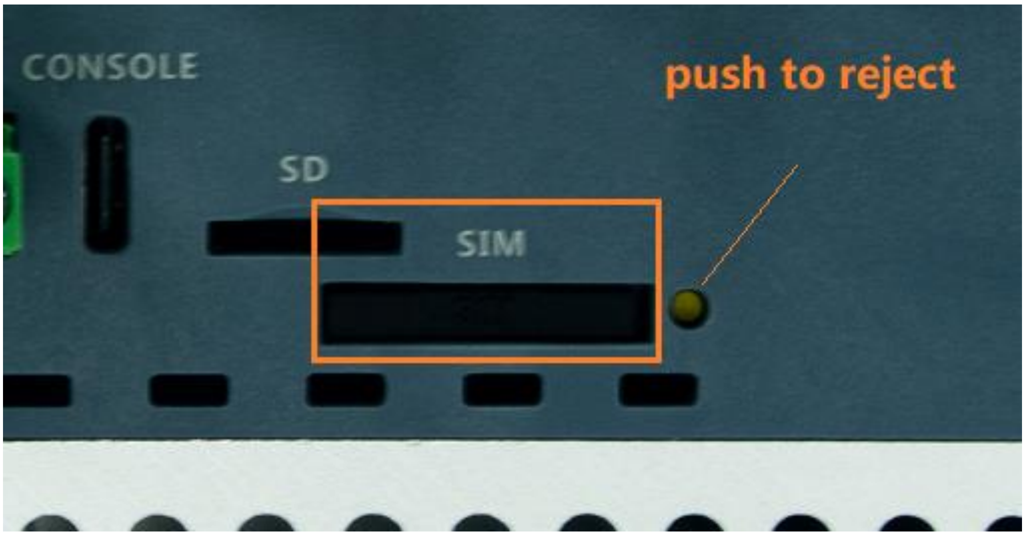
There are four SMA Connector holes for antennas. The antenna types are very depended on what modules fitted into the Mini-PCIe socket. The A1 is recommended for WI-FI signal from CM4 module and A2 for cellular.



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

2.2.10 SIM card slot

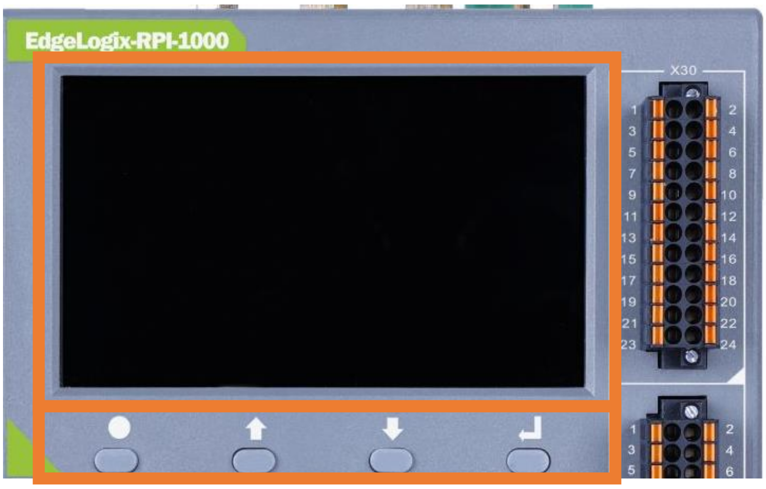
The sim card is only needed in cellular (4G/LTE or others based on cellular technology) mode.



NOTES: Only Standard Sim card is accepted, pay attention to the card size.

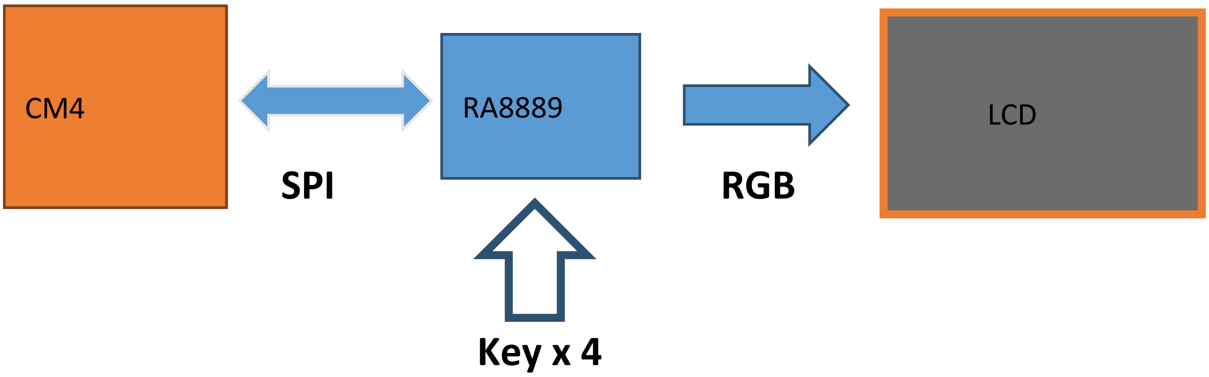
2.2.11 LCD panel and keyboard

The LCD display allows you to view meter data and perform basic configuration, or used for system diagnosis.



◆ LCD panel

The LCD display has the resolution of 800x480 pixel. It has an individual display controller connector to main CPU (Raspberry PI CM4) via SPI interface. The program model is full open to customers.



The backlight dims after a defined period of inactivity. When the meter detects an unacknowledged active high priority alarm, the display flashes until the alarm is acknowledged.

◆ ○ Home button

Pressing the home button takes you to the associated menu screen. If you are in a data screen, pressing the home button takes you to the display menu, and pressing home twice takes you to the summary display screen. If you are in a setup screen, pressing home takes you to the setup menu, and pressing home again takes you to the display menu.
Function keys

◆⇧⇩ Arrow button

The two arrow keys can be used for navigation and selection.

◆ ⮥ Enter button

Normally used as confirm or Enter.

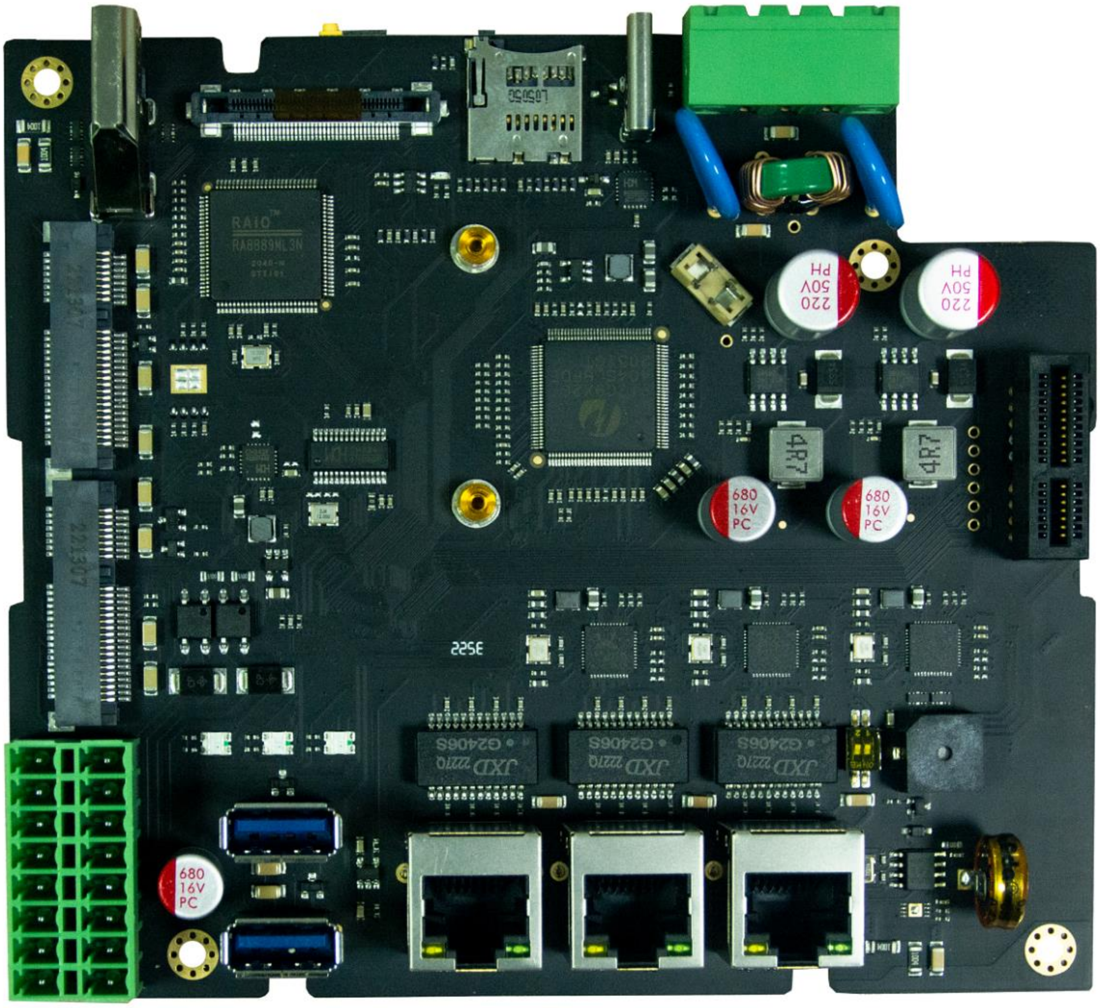
2.3 GPIO Multiplex

Overview of the GPIO usage from CM4, most of the GPIO have the fixed function as list.

Name	IO of BCM2711	Type	Function
	GPIO 0		
	GPIO 1		
I2C_SDA	GPIO 2		
I2C_SCL	GPIO 3		
WDT_O	GPIO 4	Output	Active and feed watchdog
mPCIE_RST1	GPIO 5	Output, high active	Mini PCIE 1 reset
mPCIE_RST2#	GPIO 6	Output, low active	Mini PCIE 2 resets
SPI1_SS2#	GPIO 7		SPI1
SPI1_SS1#	GPIO 8		
SPI1_MISO	GPIO 9		
SPI1_MOSI	GPIO 10		
SPI1_SCK	GPIO 11		
Key_INT#	GPIO 12		
LED_USR	GPIO 13	High active	
Uart0_tx	GPIO 14		Console
Uart0_rx	GPIO 15		
SPI2_INT#	GPIO 16	Input	SPI2
DI1	GPIO 17	Input	
SPI2_SS1#	GPIO 18		SPI2
SPI2_MISO	GPIO 19		
SPI2_MOSI	GPIO 20		
SPI2_SCK	GPIO 21		
mPCIE_PWR1	GPIO 22	High active	Open power supply of mini PCIe slot 1
Buzzer	GPIO 23	High active	
DO1	GPIO 24	High active	
DO2	GPIO 25	High active	
SPI2_CMD#	GPIO 26	Output	SPI2
DI2	GPIO 27	Input	

2.4 Mainboard

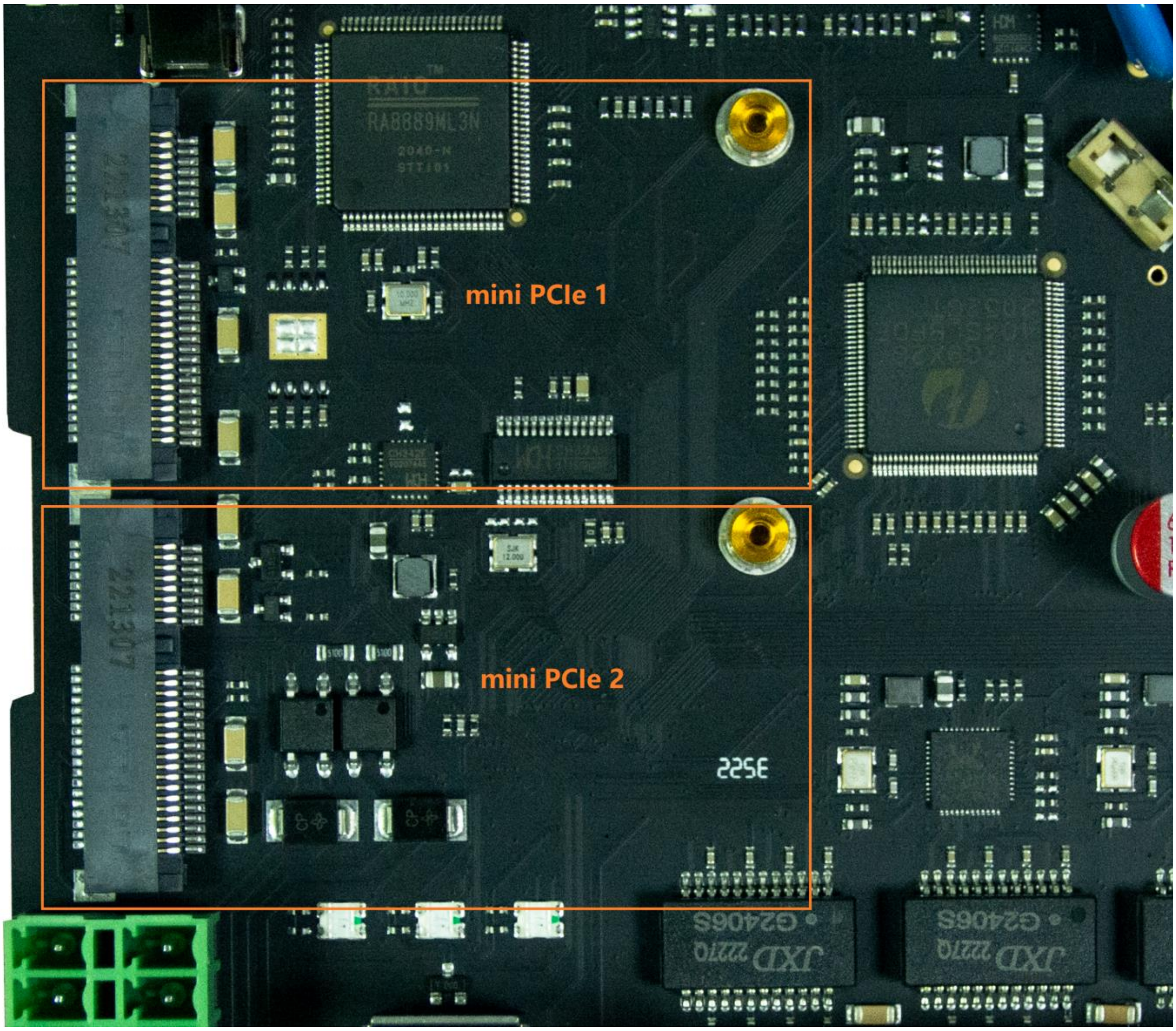
The Mainboard spans the inner width of the device and has outward-facing ports on frond and up sides. It is mounted to the heat sink with four M3x6 pan head screws.



2.4.1 Mini PCIe

EdgeLogix-RPI-1000 itself has two mini-PCIe card slots, one for 4G/LTE with SIM cad support and the other has SPI signals.

The orange area is the rough 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.

2.4.1.1 Mini-PCIe 1

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
	45	46	
	47	48	
	49	50	GND
	51	52	4G_PWR

NOTE 1: All blank signals are NC (not connect).

NOTE 2: 4G_PWR is the individual power supply for Mini-PCIe card. It can be shut down or turn on by the GPIO22 of CM4, the control signal is high active.

NOTE 3: 4G_LED signal is connected to LED1 internally

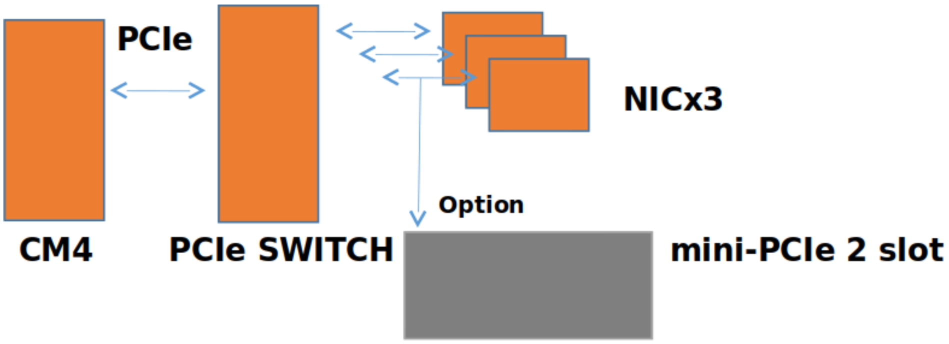
2.4.1.2 Mini-PCIe 2

Signal	PIN#	PIN#	Signal
	1	2	PWR
	3	4	GND
	5	6	
	7	8	
GND	9	10	
PCIe_clkn(optional)	11	12	
PCIe_clkp(optional)	13	14	
GND	15	16	
	17	18	GND
	19	20	
GND	21	22	PERST#
PCIe_txn(optional)	23	24	PWR
PCIe_rxp(optional)	25	26	GND
GND	27	28	
GND	29	30	
PCIe_txn(optional)	31	32	
PCIe_txp(optional)	33	34	GND
GND	35	36	
GND	37	38	
PWR	39	40	GND
PWR	41	42	
GND	43	44	
SPI1_SCK	45	46	
SPI1_MISO	47	48	
SPI1_MOSI	49	50	GND
SPI1_SS	51	52	PWR

NOTE 1: SPI1 signals are used only for LoraWAN card, such as SX1301, SX1302.

NOTE 2: All PCIe signals are optional.

2.4.2 PCIe subsystem



The CM4 itself has only on channel of PCIe 1x. The design of EdgeLogix-RPI-1000 use a switch to extern 3 NIC cards and one optional channel to mini PCIe 2 slots.

3. Drivers and Programming

3.1 LED

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

```
$ sudo -i      #enable root account privileges
$ cd /sys/class/gpio
$ echo 13 > export #GPIO21 which is user LED of LED2
$ cd gpio13
$ echo out > direction
$ echo 1 > value # turn on the user LED, HIGH active
OR
$ echo 0 > value # turn off the user LED
```

3.2 LCD Panel Demo:

Since the LCD Panel is controlled by RA8889, here we have prepared the Demo code for testing, to download the latest example code, please visit:
https://files.seeedstudio.com/wiki/Edge_Logix/LCD_test.zip

3.3 Left Connector 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')
  10
```

3.4 Left Connector DI&DO

The Edgelogix-RPI-1000 has 2 isolated DO ports in the Left connector.

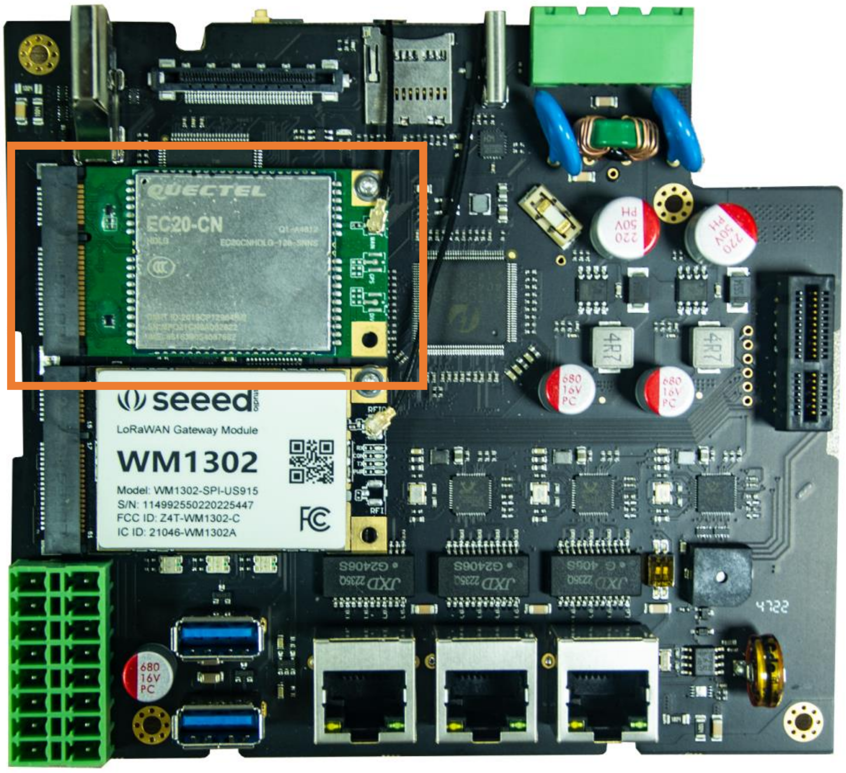
```
  $ cd /sys/class/gpio/
  $ echo 24 > export
  $ echo 25 > export
  $ cd gpio24
  $ echo out > direction
  $ echo 1 >value
  $ cd ../gpio25
  $ echo out > direction
  $ echo 1 >value
```

And 2 isolated DI ports.

```
  $ cd /sys/class/gpio/
  $ echo 17 > export
  $ echo 27 > export
  $ cd gpio17
  $ echo in > direction
  $ cat value
  $ cd ../gpio27
  $ echo in > direction
  $ cat value
```

3.5 Cellular over Mini-PCIe

- **Use Quectel EC20 as a example and follow the steps:**



1. **Insert the EC20 into Mini-PCIe socket and 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 22 > export    #GPIO22 which is POW_ON signal
$ echo 5 > export    #GPIO5 which is reset signal

$ cd gpio22
$ 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 cellular is start to flash.

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.

```
$ 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
```

5. How to use AT command

```
$ miniterm --- Available ports:
---    1: /dev/ttyAMA0              'ttyAMA0'
---    2: /dev/ttyttyACM0              'CP2105 Dual USB to UART Bridge Controller'
---    3: /dev/ttyttyACM1              'CP2105 Dual USB to UART Bridge Controller'
---    4: /dev/ttyUSB0              'Android'
---    5: /dev/ttyUSB1              'Android'
---    6: /dev/ttyUSB2              'Android'
---    7: /dev/ttyUSB3             'Android'
--- Enter port index or full name:
$ miniterm /dev/ttyUSB3 115200
```

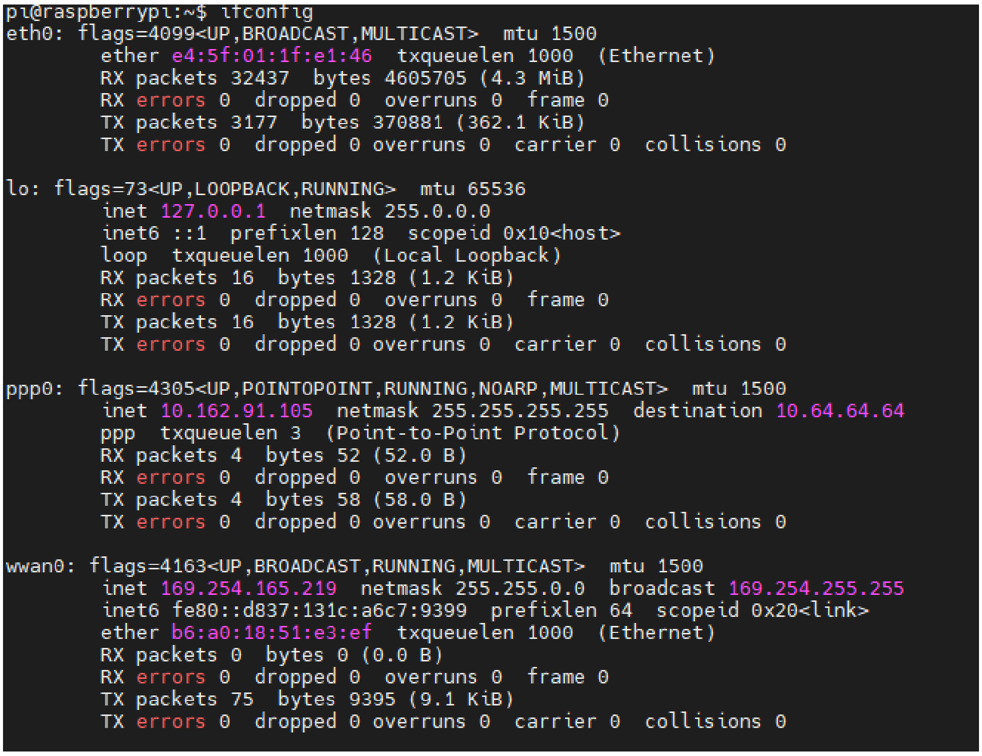
Some useful AT command :

- 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

6. 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:



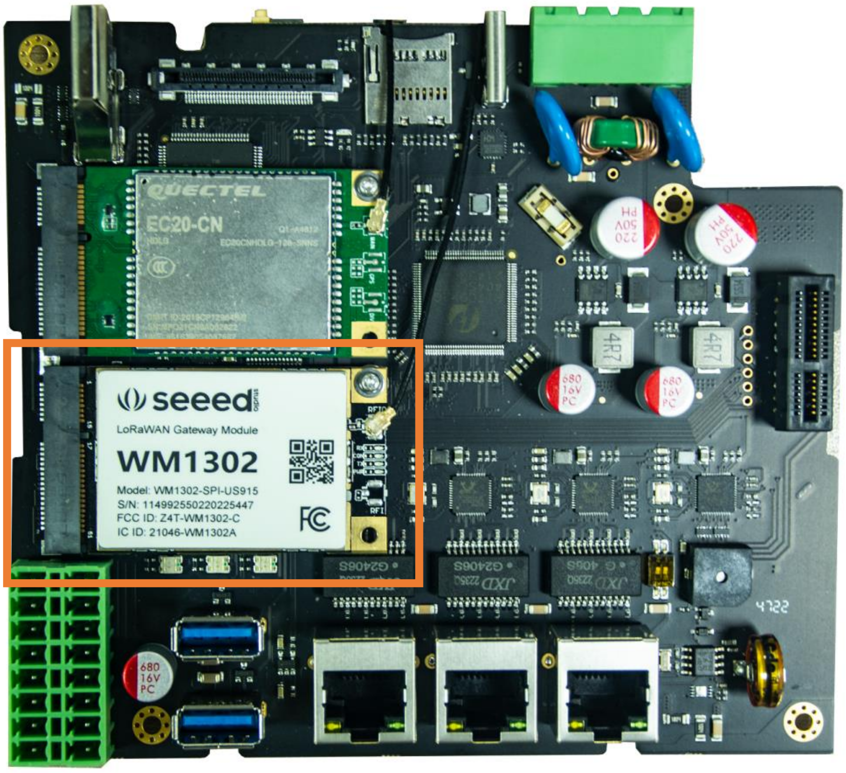
```
pi@raspberrypi:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether e4:5f:01:1f:e1:46  txqueuelen 1000  (Ethernet)
        RX packets 32437  bytes 4605705 (4.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3177  bytes 370881 (362.1 KiB)
     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

7. **Add the router path**

$ route add default gw 10.64.64.64 or your gateway XX.XX.XX.XX
Then test with ping command:
$ ping google.com

3.6 LoraWAN® Module over Mini-PCIe

- **Here is an example of using Seeed Studio's WM1302 LoraWAN® Module**



Step 1:

Insert the WM1302 card into Mini-PCIe 2 slot ,connect the antenna.

Step 2:

Turn on the power of Mini-PCIe 2

```
$ sudo -i      #enable root account privileges
$ cd /sys/class/gpio
$ echo 22 > export    #GPIO22 which is POW_ON signal

$ cd gpio22
$ echo out > direction
$ echo 1 > value    # turn on the power of Mini PCIe
```

Step 3:

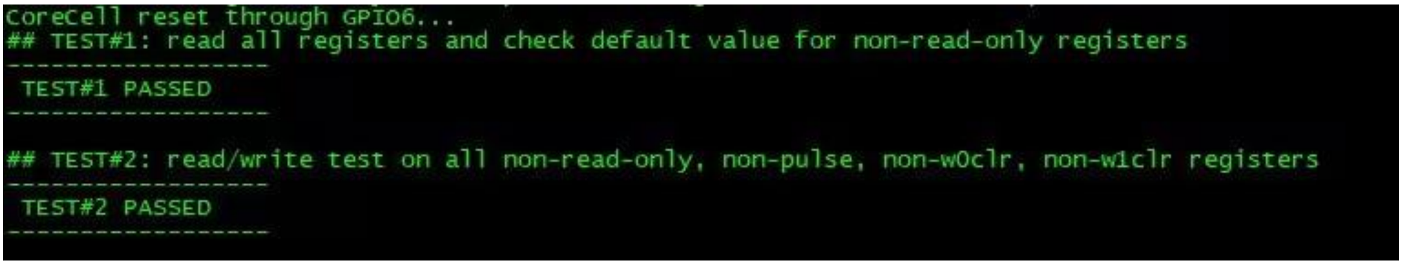
Download the test software @

$ git clone https://github.com/Lora-net/sx1302_hal

Modify the SX1302_RESET_PIN=6 (We use GPIO6 as RESET signal)   reset_lgw.sh

Step 4:

```
$ ./test_loragw_reg -d /dev/spidev0.1
```



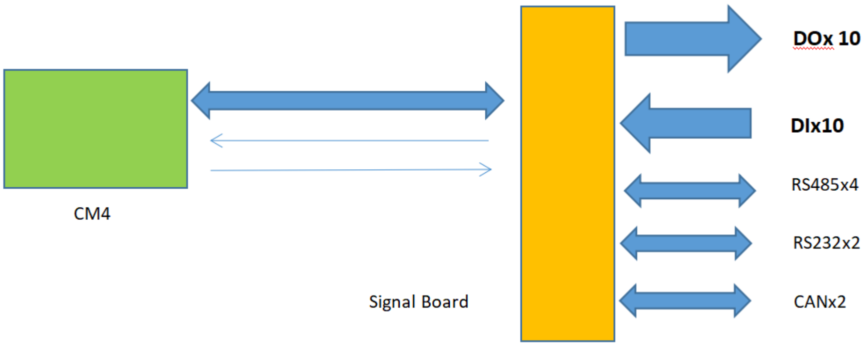
```
CoreCell reset through GPIO6...
## TEST#1: read all registers and check default value for non-read-only registers
------------------
 TEST#1 PASSED
------------------

## TEST#2: read/write test on all non-read-only, non-pulse, non-w0clr, non-w1clr registers
------------------
 TEST#2 PASSED
------------------
```

```
$ ./test_loragw_hal_tx -d /dev/spidev0.1 -r 1250 -f 868 -m LORA -b 125 -s 12 -z 20
$./test_loragw_hal_rx -d /dev/spidev0.1 -r 1250 -a 475.5 -b 476.5
```

3.7 Signal board

3.7.1 Logic architecture

The Logic architecture between CM4 and Signal Board.



There are 6 signals between CM4 and Signal Board.

Signals	Direction			
SPI_MOSI	Out			
SPI_MISO	In			
SPI_SCK	Out			
SPI_CS#	Out			
CMD#	Out			
INT#	In			

NOT: All the 6 signals are isolated from main board and signal board.

3.7.2 Update the firmware of signal board

The signal board has a independent MCU itself and a firmware. it also has a built-in update system.
Put the firmware.bin in the update folder, the run:

```
sudo ./update firmware.bin
```

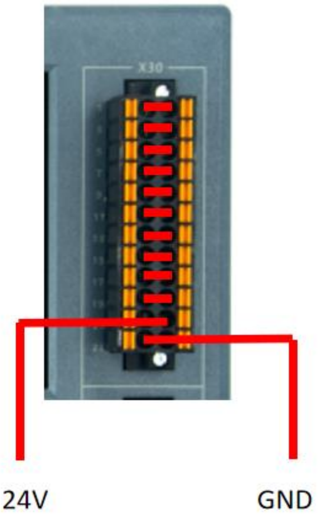
NOTE 1: To download the latest firmware, Please visit:

https://files.seeedstudio.com/wiki /Edge_Logix/update.zip

NOTE 2: The source code example for programming signal board, Related files :spidev.c, spidev.h, api.c, api.h and test.c, refer to the source of the source code.

3.7.3 Example Demo of the Digital I/O

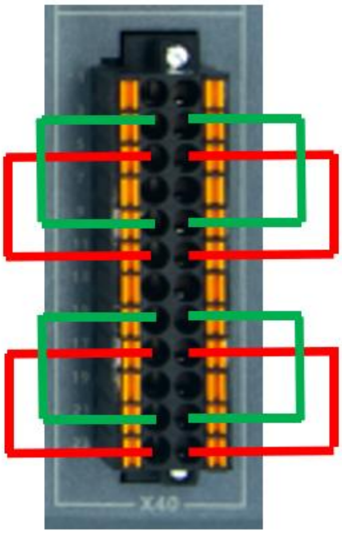
Connector X30 is the DI&DO port of signal expansion board. Please connect the DI and DO ports as shown below to perform the SELF TEST Demo, to download the latest example code, please visit: https://files.seeedstudio.com/wiki/Edge_Logix/DOandDI_test.zip



Then run following command:
$ sudo ./test

3.9.4 Example Demo of the Communication Ports

Connector X40 is the communication port of signal expansion board. Please connect the communication ports as shown below to perform the SELF TEST Demo, to download the latest example code, please visit: https://files.seeedstudio.com/wiki/Edge_Logix/DOandDI_test.zip

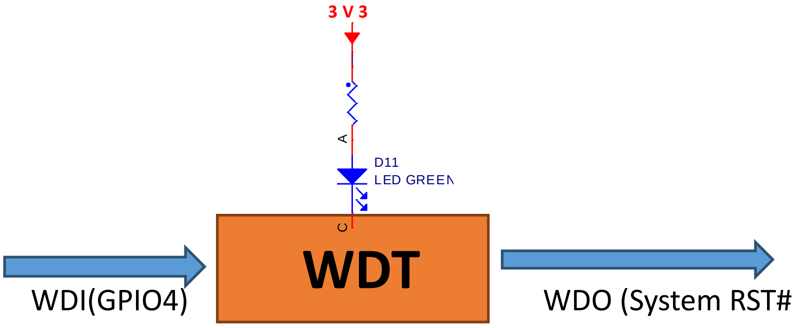


Then run following command:
$ sudo ./test

3.8 WDT

3.8.1 Block Diagram of WDT

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



The WDT is not utilized by default which has no influence on the system. To enable it please download the script here: https://files.seeedstudio.com/wiki /Edge_Logix/WDT.py
Reference:

```
$ python
$ import time
$ import RPi.GPIO as GPIO
$ GPIO.setmode(GPIO.BCM)
$ GPIO.setup(4,GPIO.OUT) # WDT FEEED

# open the WDT
i=10
while(i>0)：
GPIO.output(4,1)
time.sleep(0.001)
GPIO.output(4,0)
# then the WDT LED is on

# feed the WDT in typical 1 second OR the systom will reset
while(1)：
GPIO.output(4,1)
time.sleep(1)
GPIO.output(4,0)
```

3.8.2 How it works

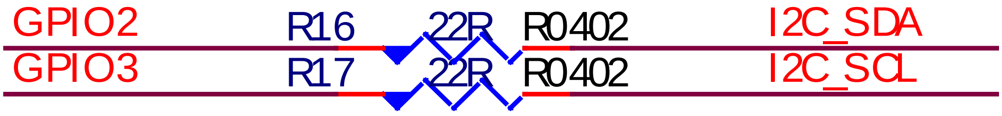
1. System POWER ON.
2. Delay 200ms.
3. Send WDO a negative pulse with 200ms low level to reset the system.
4. Pull up WDO.

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.

3.9 RTC

3.9.1 RTC Chip information

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.

3.9.2 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

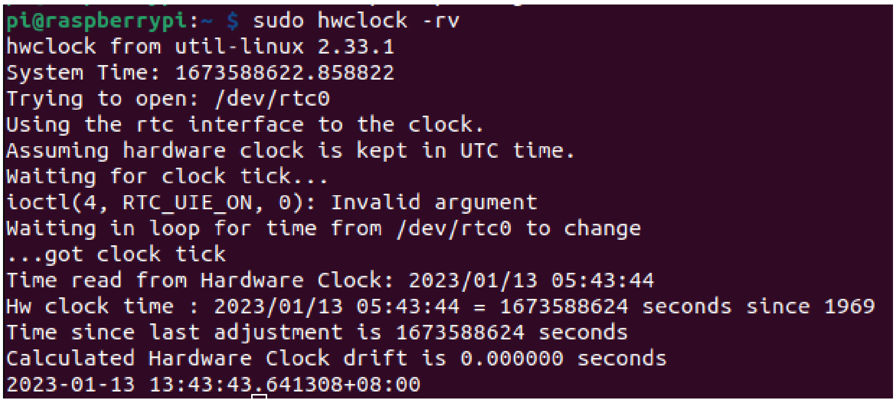
Then reboot the system

$sudo reboot

Then use 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.

NOTE 2: the estimated backup time of the RTC is 10 days.

3.10 Buzzer

Here is an example of toggle the GPIO to activate the buzzer.

```
$ cd /sys/class/gpio/
$ echo 23 > export
$ cd gpio23
$ echo out > direction
$ echo 1 > value # on
$ echo 0 > value # off
```

3.9 Signal Boar

4. Electrical specifications

4.1 Power consumption

The power consumption of the EdgeLogix-RPI-1000 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 EdgeLogix-RPI-1000:

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

Mode of operation	Current(ma)	Remark
Idle	320	LCD ON
Stress test	360	stress -c 4 -t 10m -v &