BPC-iMX6ULL-02 User Manual

Version: V1.0 (2022-05)
Editor: Polyhex Technology Company Limited (http://www.polyhex.net/)

intel IoT Solutions Alliance

Declaration of conformity

CE

This product has passed the CE test for environment specifications.

FCC

This equipment has been tested and found to comply with the FCC rules.

RoHS

This product has passed the RoHS test

CCC

This product has passed the CCC test

Technical support and assistance

1. Visit polyhex website http://www.polyhex.net/ where you can find the latest information about the product.
2. Contact your distributor, sales representative or polyhex's customer service center for technical support if you need additional assistance.Please have the following info ready before you call:
   - Product name
   - Description of your peripheral attachments
   - Description of your software(operating system, version, application software, etc.)
   - A complete description of the problem
   - The exact wording of any error messages

Safety instructions

1. Read these safety instructions carefully.
2. Keep this User Manual for later reference.
3. Disconnect this equipment from any DC outlet before cleaning. Use a damp cloth. Do not use liquid or spray detergents for cleaning.
4. Keep this equipment away from humidity.
5. Put this equipment on a reliable surface during installation. Dropping it or letting it fall may cause damage.
6. Make sure the voltage of the power source is correct before connecting the equipment to the power outlet.
7. The openings on the enclosure are for air convection. Protect the equipment from overheating. DO NOT COVER THE OPENINGS.
8. Position the power cord so that people cannot step on it. Do not place anything over the power cord.
9. If the equipment is not used for a long time, disconnect it from the power source to avoid damage by transient overvoltage.
10. Never pour any liquid into an opening. This may cause fire or electrical shock.
11. Never open the equipment. For safety reasons, the equipment should be opened only by qualified service personnel.
12. If one of the following situations arises, get the equipment checked by service personnel:
    The power cord or plug is damaged.
    Liquid has penetrated into the equipment.
    The equipment has been exposed to moisture.
    The equipment does not work well, or you cannot get it to work according to the user's manual.
    The equipment has been dropped and damaged.
    The equipment has obvious signs of breakage.
13. DO NOT LEAVE THIS EQUIPMENT IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY GO BELOW 0° C (0° F) OR ABOVE 70° C (158° F). THIS COULD DAMAGE THE EQUIPMENT. THE EQUIPMENT SHOULD BE IN A CONTROLLED ENVIRONMENT.
14. Due to the sensitive nature of the equipment it must be stored in a restricted access location, only accessible by qualified engineers.

DISCLAIMER: Polyhex disclaims all responsibility for the accuracy of any statement of these instructions.

intel IoT Solutions Alliance

Contents

Chapter 1 Introduction 5
1.1 Product introduction 5
1.2 Hardware specifications 5
1.3 Safety Precaution 6
Chapter 2 Hardware Function 8
2.1 Overview 8
2.2 LED Status Indicator 8
2.3 Reset button 10
2.4 SIM slot 10
Chapter 3 wiring and installation 11
3.1 Wiring 11
3.2 Installation 15
Chapter 4 Software Application Examples 17
4.1 Remote login SSH 17
4.2 Use of Ethernet 18
4.3 Use of Wifi 19
4.4 Use of Bluetooth 20
4.5 Use of USB 21
4.5 Verification of RS232 22
4.6 Verification of RS485 23
4.7 CAN functional verification 24
4.8 Verification of DI 25
4.9 Verification of DO 26
4.10 Use of 4G Module 26
4.11 Use of Lora Module 28
4.12 Verification of RTC 29

Chapter 1 Introduction

1.1 Product introduction

BPC-iMX6ULL-02 is a RISC architecture platform with high performance, wide temperature and flexible design. It serves as a gateway connecting inverters and remote monitoring center in power and energy application, which plays an important role.

1.2 Hardware specifications

1.2.1 General

- **Certification**: CE, FCC, RoHS, CCC
- **Dimensions(W x D x H)**:107mm x 107mm x 35mm
- **Power Requirements**:DC12V1A
- **Weight**:0.375kg
- **OS Support**: buildroot-2021.11-rc3, Yocto 3.2.1

1.2.2 System Hardware

- **CPU**: NXP i.MX 6ULL(default), support NXP i.MX 6UltraLite
- **Memory**: 512MB DDR3
- **Indicators**: LED for Power, LAN(LINK, ACT),STAT LED,SYS LED
- **Storage**: 8GB eMMC(default)
- **SIM Slot**: 1 x Micro-SIM slot

1.2.3 System Software

- **OS Support**: buildroot-2021.11-rc3, Yocto 3.2.1
- **Protocol support**: Modbus, IEC-60870-101(master)/104(slave)
- **Programming**:C

1.2.4 I/O Interface

- **Serial ports**: 2 x Physically-isolated RS-232, 2 x Physically-isolated RS-485
- **Serial port Speed**:115200bps
- **LAN**: 2 x 10M/100Mbps Base-T RJ-45 ports
- **USB Ports**: 1 x USB, Rev. 2.0 Host(default)
- **SIM**: 1 x Micro SIM
- **CAN**: 2 x Physically-isolated CAN
- **LED**: 1 x Power LED, 2 x GPIO LED(functions can be customized)
- **Reset**: 1 x Reset
- **SMA RF**: 2 x SMA RF ANT(WIFI ANT for default, choose one from 4G/LoRa/UWB(optional))
- **LoRa**: 1 x LoRa(optional)
- **UWB**: 1 x UWB(optional)

1.2.5 Environment

- **Humidity**: 5 ~ 95%(non-condensing)
- **Operating Temperature**: 0~70℃
- **Storage Temperature**: -40~85℃
- **Safety Cert. Temperature**: -20~50℃（-4~122°F）
- **Operating Humidity**: 20~95%(non-condensing)

1.3 Safety Precaution

The following messages informs how to make each connection. In most cases, you will simply need to connect a standard cable.

Warning!

Always disconnect the power cord from your chassis whenever you are not working on it. Do not connect while the power is on. A sudden rush of power can damage sensitive electronic components. Only experienced electronics personnel should open the chassis.

Caution!

Always ground yourself to remove any static electric charge before touching BPC-iMX6ULL-02. Modern electronic devices are very sensitive to electric charges.Use a grounding wrist strap at all times.Place all electronic components on a static-dissipative surface or in a static-shielded bag.

Chapter 2 Hardware Function

2.1 Overview

The following figures show the indicators and connectors on BPC-iMX6ULL-02.

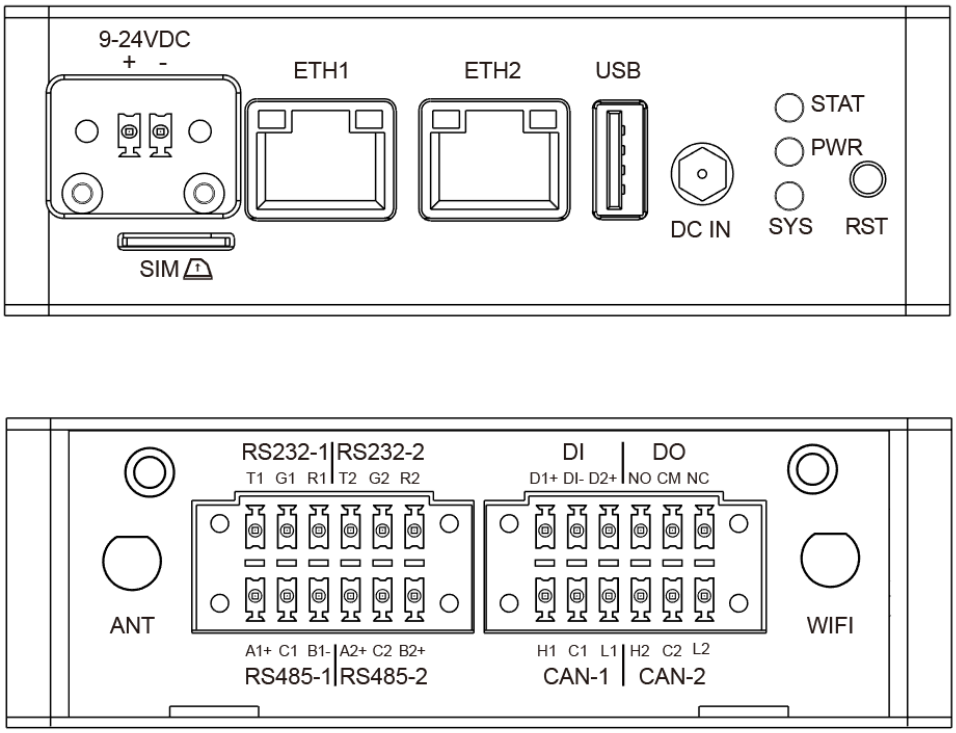


Figure1 BPC-iMX6ULL-02 overview

2.2 LED Status Indicator

2.2.1 System Status Indicator

intel IoT Solutions Alliance

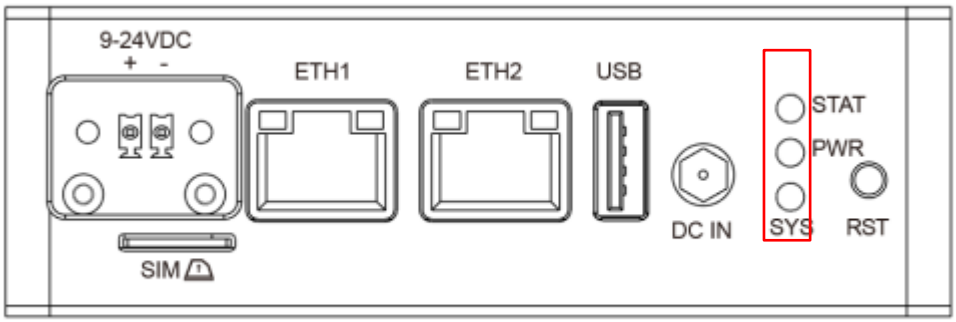


Figure2 system status indicator

LED	Status	Description
PWR (not in control)	lighting	Power is on
	off	Power if off
STAT (in control)	blinking	Power is on
	off	Power is off
SYS (not in control)	Lighting	Power is on
	off	Power is off

2.2.2 Ethernet Status Indicator

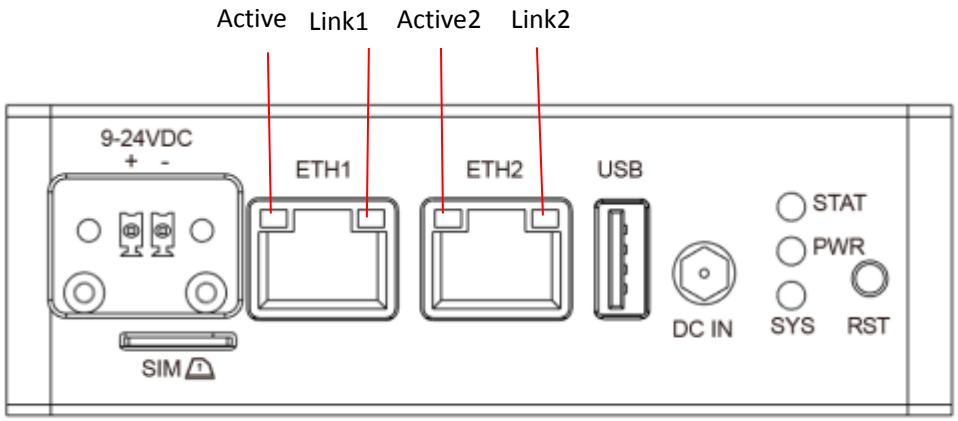


Figure3 Ethernet Status Indicator

LED	Color	Description
Active1	Green	Lighting, the Ethernet cable is plugged in
Link1	Orange	Blinking, Ethernet data being transmitted
Active2	Green	Lighting, the Ethernet cable is plugged in
Link2	orange	Blinking, Ethernet data being transmitted

intel IoT Solutions Alliance

2.3 Reset button

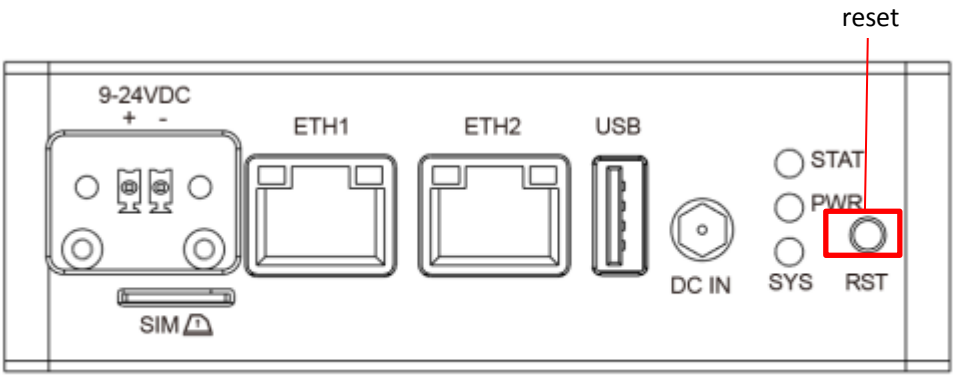


Figure4 reset button

Press reset button continuously for 6 seconds to reset the system

2.4 SIM slot

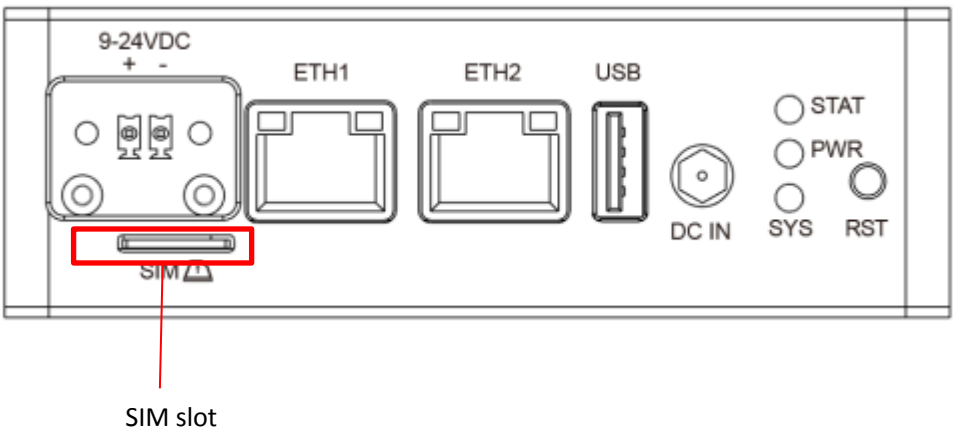


Figure5 SIM slot

When insert and unplug the SIM from the SIM slot, please take care of the direction which has been indicated on the box.

intel IoT Solutions Alliance

Chapter 3 wiring and installation

3.1 Wiring

3.1.1 Power Supply Wiring

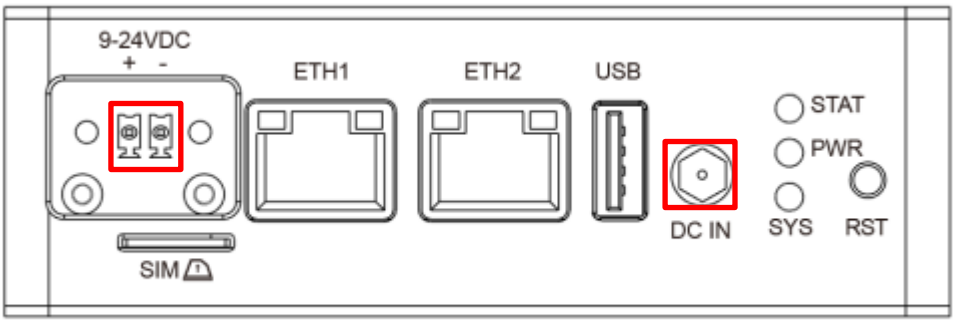


Figure6 Power supply wiring

BPC-iMX6ULL-02 supports power input ranging from 9 VDC to 24 VDC.

DC Power Input Connector Pin Definition		
Function	Pin	description
Power input	+	DC power input positive pin
	-	DC Power input negative pin
DC IN	DC IN	DC Power input pin

3.1.2 Communication Ports

3.1.2.1 RS-232 Serial Ports

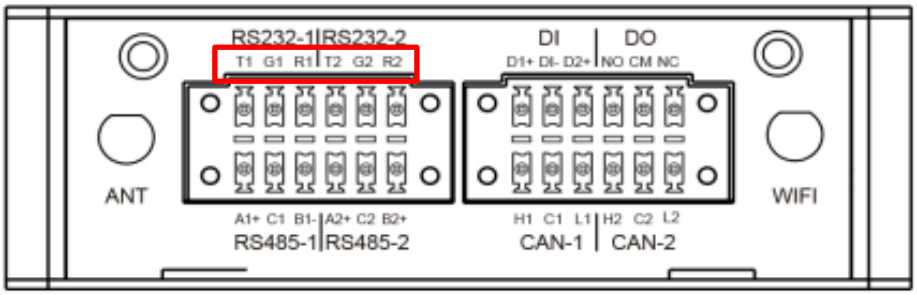


Figure7 RS-232 Serial Ports

intel IoT Solutions Alliance

RS-232 Serial Ports(Pin Assignments)		
Function	pin	description
RS232-1	T1	Transfer data1
	G1	GND1
	R1	Receive data1
RS232-2	T2	Transfer data2
	G2	GND2
	R2	Receive data2

3.1.2.2 RS-485 Serial Ports

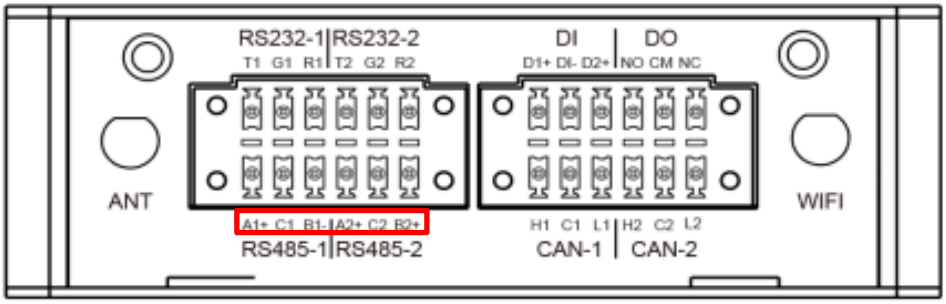
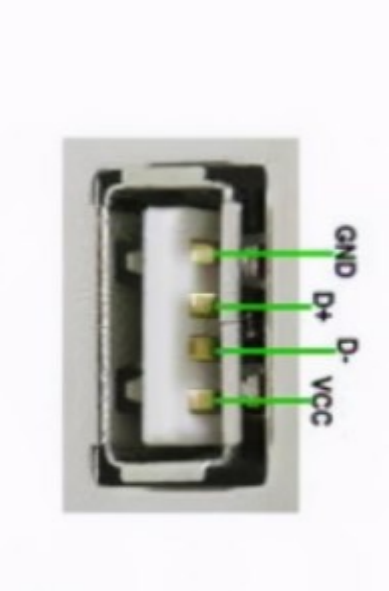


Figure8 RS-485 Serial Ports

RS-485 Serial Ports(Pin Assignments)		
Function	pin	description
RS485-1	A1+	Noninverting receiver input1 and Noninverting driver output1
	C1	Ground1
	B1-	Inverting receiver input1 and inverting driver output1
RS485-2	A2+	Noninverting receiver input2 and Noninverting driver output2
	C2	Ground2
	B2+	Inverting receiver input2 and inverting driver output2

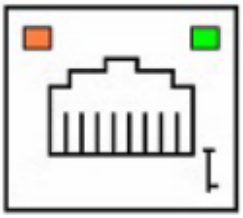
3.1.2.3 USB Connector



USB Connector Pin Assignment	
Pin	Signal
1	VCC
2	DATA-
3	DATA+
4	GND

Default: UART-MODE

3.1.2.4 LAN Connectors(LAN1~LAN2)



LAN Connector Pin Assignments		
Pin	Assignment	Description
1	TD+	Transmit+
2	TD-	Transmit-
3	RD+	Receive+
4	N/C	Not used
5	N/C	Not used
6	RD-	Receive -
7	N/C	Not used
8	N/C	Not used

3.1.2.5 Can Ports(CAN 1~2)

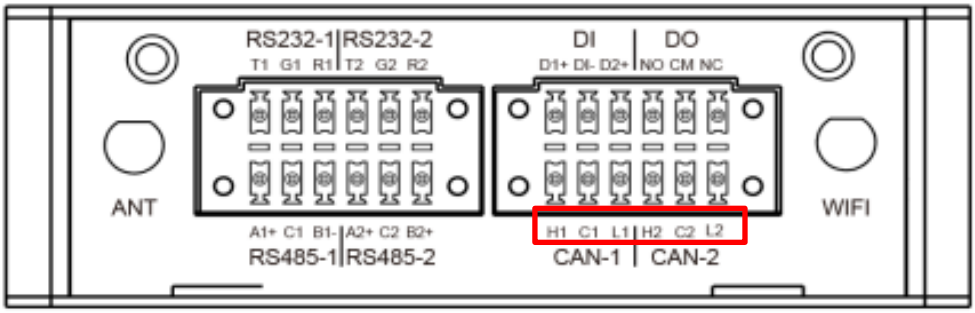


Figure9 CAN ports

CAN Ports(Pin Assignments)		
function	pin	Description
CAN1	H1	High-level CAN bus line1
	C1	Ground1
	L1	Low-level CAN bus line1
CAN2	H2	High-level CAN bus line2
	C2	Ground2
	L2	Low-level CAN bus line2

3.1.2.6 DI/DO Ports

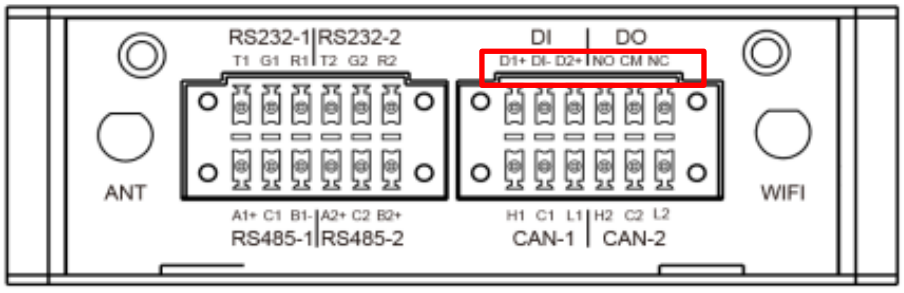


Figure10 DI/DO ports

DI/DO Ports(Pin Assignments)		
function	pins	description

intel IoT Solutions Alliance

DI	D1+	Digital input1 positive
	DI-	Digital input negative
	D2+	Digital input2 positive
DO	NO	Normal open
	CM	Common
	NC	Normal connected

3.2 Installation

- Check the BPC-iMX6ULL-02 box pc, Wi-Fi antenna and power adapter after you receive the product.

Figure11 antenna

intel IoT Solutions Alliance

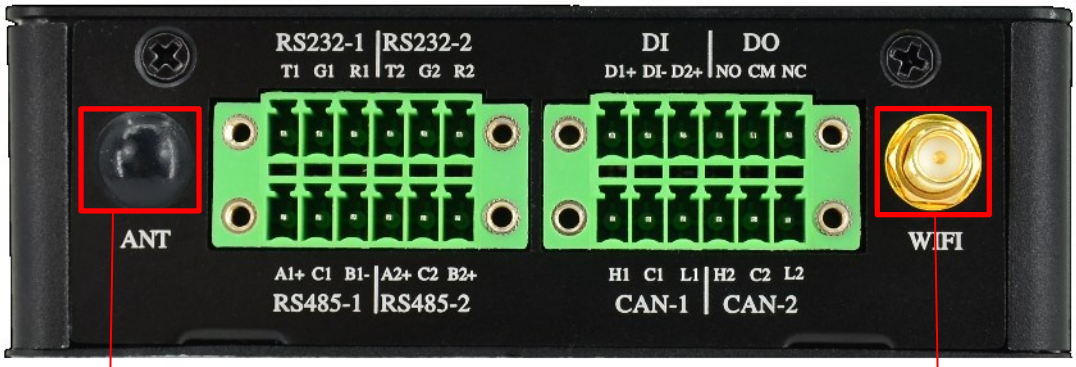


Figure12 BPC-iMX6ULL-02

- Connect the WIFI antenna to the WIFI antenna interface

Figure13 power adapter

intel IoT Solutions Alliance

Figure 14 BPC-iMX6ULL-02 with antenna

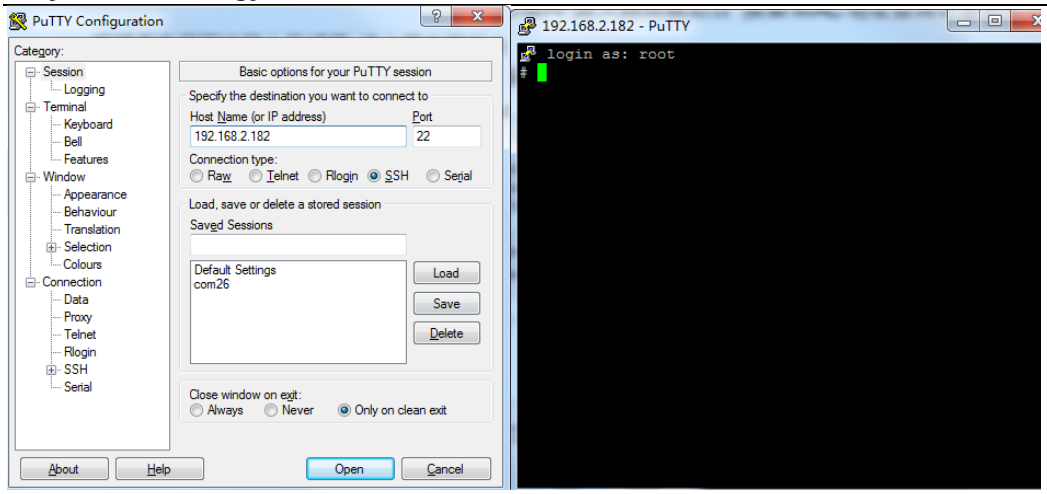
- Connect the power adapter to the DC interface of BPC-iMX6ULL-02

When you find that the SYS LED and PWR LED are lighting, and the STAT LED is blinking, it means the gateway has been powered on.

Chapter 4 Software Application Examples

4.1 Remote login SSH

Connect the device to the LAN, enter the router background, query the IP address obtained by the device according to the MAC address, ssh to the device background through "putty" or other tools, access the account: root, and the password is blank by default; as shown below:



Change root password command:
#passwd root

Enter the new password twice in a row:

```
# passwd root
Changing password for root
New password:
Bad password: too short
Retype password:
passwd: password for root changed by root
#
```

4.2 Use of Ethernet

query ip command:
#ip a

```
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
3: can1: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 10:07:23:6d:c6:12 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.182/24 brd 192.168.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::1207:23ff:fe6d:c612/64 scope link
       valid_lft forever preferred_lft forever
5: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 10:07:23:6d:c6:13 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.108/24 brd 192.168.2.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::1207:23ff:fe6d:c613/64 scope link
       valid_lft forever preferred_lft forever
6: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
7: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default q
len 1000
    link/ether ac:6a:a3:15:23:3f brd ff:ff:ff:ff:ff:ff
#
```

As shown above: eth0 network card corresponds to the network port of the device silkscreen

"eth1" (left side)

The eth1 network card corresponds to the network port of the device silkscreen "eth2" (right side)

Application command:
#ping -i eth0 192.168.2.254

```
# ping -i eth0 192.168.2.254
ping: option argument contains garbage: eth0
ping: this will become fatal error in the future
PING 192.168.2.254 (192.168.2.254) 56(84) bytes of data.
64 bytes from 192.168.2.254: icmp_seq=1 ttl=254 time=1.27 ms
64 bytes from 192.168.2.254: icmp_seq=2 ttl=254 time=0.733 ms
64 bytes from 192.168.2.254: icmp_seq=3 ttl=254 time=1.34 ms
64 bytes from 192.168.2.254: icmp_seq=4 ttl=254 time=1.19 ms
64 bytes from 192.168.2.254: icmp_seq=5 ttl=254 time=1.19 ms
64 bytes from 192.168.2.254: icmp_seq=6 ttl=254 time=1.17 ms
64 bytes from 192.168.2.254: icmp_seq=7 ttl=254 time=1.20 ms
64 bytes from 192.168.2.254: icmp_seq=8 ttl=254 time=1.17 ms
64 bytes from 192.168.2.254: icmp_seq=9 ttl=254 time=1.16 ms
64 bytes from 192.168.2.254: icmp_seq=10 ttl=254 time=1.19 ms
64 bytes from 192.168.2.254: icmp_seq=11 ttl=254 time=1.21 ms
64 bytes from 192.168.2.254: icmp_seq=12 ttl=254 time=1.23 ms
64 bytes from 192.168.2.254: icmp_seq=13 ttl=254 time=1.20 ms
64 bytes from 192.168.2.254: icmp_seq=14 ttl=254 time=1.23 ms
64 bytes from 192.168.2.254: icmp_seq=15 ttl=254 time=1.22 ms
64 bytes from 192.168.2.254: icmp_seq=16 ttl=254 time=1.23 ms
64 bytes from 192.168.2.254: icmp_seq=17 ttl=254 time=1.20 ms
64 bytes from 192.168.2.254: icmp_seq=18 ttl=254 time=1.23 ms
64 bytes from 192.168.2.254: icmp_seq=19 ttl=254 time=1.19 ms
```

4.3 Use of Wifi

Edit the configuration file and set the "SSID" and connection password of the connected router:

#vi /etc/wpa_supplicant.conf

```
ctrl_interface=/var/run/wpa_supplicant
ap_scan=1

network={
  #key_mgmt=NONE
  ssid="BH123"
   psk="1234567890"
}
```

SSID

passwd

#wpa_supplicant -Dnl80211 -iwlan0 -c/etc/wpa_supplicant.conf &

intel IoT Solutions Alliance

```
# wpa_supplicant -Dnl80211 -iwlan0 -c/etc/wpa_supplicant.conf &
# Successfully initialized wpa_supplicant
rfkill: Cannot open RFKILL control device
wlan0: Trying to associate with SSID 'BH123'
wlan0: Associated with 0c:d8:6c:a9:cc:08
wlan0: CTRL-EVENT-CONNECTED - Connection to 0c:d8:6c:a9:cc:08 completed [id=0 id_str=]
wlan0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0

#
```

udhcpc -i wlan0 –n

The ip address assigned to the obtained router is obtained as follows

```
#  udhcpc -i wlan0 -n
udhcpc: started, v1.34.1
udhcpc: broadcasting discover
udhcpc: broadcasting select for 192.168.10.8, server 192.168.10.254
udhcpc: lease of 192.168.10.8 obtained from 192.168.10.254, lease time 86400
deleting routers
adding dns 202.96.134.133
adding dns 114.114.114.114
#
```

4.4 Use of Bluetooth

Start bluetooth, and match the bluetooth command as follows:

#hciconfig hci0 up
#bluetoothctl
power on
agent on
default-agent
scan on
pair yourDeviceMAC

```
# bluetoothctl
Agent registered
[CHG] Controller AC:6A:A3:15:23:40 Pairable: yes
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# agent on
Agent is already registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller AC:6A:A3:15:23:40 Discovering: yes
[NEW] Device 6F:77:E4:55:30:6B 6F-77-E4-55-30-6B
[NEW] Device 58:1F:3E:7C:17:CE 58-1F-3E-7C-17-CE
[NEW] Device 61:8D:F0:19:75:3E 61-8D-F0-19-75-3E
[NEW] Device 68:7A:15:E7:AD:CA 68-7A-15-E7-AD-CA
[NEW] Device 78:21:08:79:5C:85 78-21-08-79-5C-85
[NEW] Device 6F:66:07:AC:13:D7 6F-66-07-AC-13-D7
[NEW] Device 68:E4:6A:8E:99:74 68-E4-6A-8E-99-74
[NEW] Device 54:AF:B7:03:4D:69 54-AF-B7-03-4D-69
[NEW] Device 74:5F:D2:47:FC:43 74-5F-D2-47-FC-43
```

```
[bluetooth]# pair 4C:02:20:3C:2A:6C
Attempting to pair with 4C:02:20:3C:2A:6C
[CHG] Device 4C:02:20:3C:2A:6C Connected: yes
Request confirmation
[agent] Confirm passkey 381184 (yes/no): yes
[CHG] Device 4C:02:20:3C:2A:6C Modalias: bluetooth:v038Fp1200d1436
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 0000fdaa-0000-1000-8000-00805f9b34fb
[CHG] Device 4C:02:20:3C:2A:6C UUIDs: 98b97136-36a2-11ea-8467-484d7e99a198
[CHG] Device 4C:02:20:3C:2A:6C ServicesResolved: yes
[CHG] Device 4C:02:20:3C:2A:6C Paired: yes
Pairing successful
```

4.5 Use of USB

Access the U disk in FAT32 format, the system will automatically mount it to the /mnt path

#df -h

```
# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                 4.8G     60.7M      4.4G   1% /
devtmpfs                163.9M         0    163.9M   0% /dev
tmpfs                   244.4M         0    244.4M   0% /dev/shm
tmpfs                   244.4M     68.0K    244.4M   0% /tmp
tmpfs                   244.4M     40.0K    244.4M   0% /run
/dev/mmcblk1p3            1.7G     60.0K      1.6G   0% /recovery
/dev/sda1                28.7G    544.0K     28.7G   0% /mnt
#
```

If the U disk is not mounted, you can mount the U disk with the following command:

#fdisk -l

Query the U disk letter:

```
# fdisk -l
Disk /dev/mmcblk1: 7456 MB, 7818182656 bytes, 15269888 sectors
238592 cylinders, 4 heads, 16 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device       Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/mmcblk1p1    320,0,1     959,3,16         20480    1044479    1024000  500M  c Win95 FAT32 (L
BA)
/dev/mmcblk1p2    768,0,1     639,3,16       1228800   11509759   10280960 5020M 83 Linux
/dev/mmcblk1p3    640,0,1     1023,3,16     11509760   15269887    3760128 1836M 83 Linux
Disk /dev/sda: 29 GB, 30784094208 bytes, 60125184 sectors
3742 cylinders, 255 heads, 63 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device  Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size Id Type
/dev/sda1    0,0,33      1023,254,63         32   60125183   60125152 28.6G  c Win95 FAT32 (LBA)
#
```

Mounting

#mount /dev/sda1 /mnt

```
# mount /dev/sda1 /mnt
# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                 4.8G     60.8M      4.4G   1% /
devtmpfs                163.9M         0    163.9M   0% /dev
tmpfs                   244.4M         0    244.4M   0% /dev/shm
tmpfs                   244.4M     52.0K    244.4M   0% /tmp
tmpfs                   244.4M     36.0K    244.4M   0% /run
/dev/mmcblk1p3            1.7G     60.0K      1.6G   0% /recovery
/dev/sda1                28.7G    544.0K     28.7G   0% /mnt
#
```

Access the U disk and copy files:
#cd /mnt

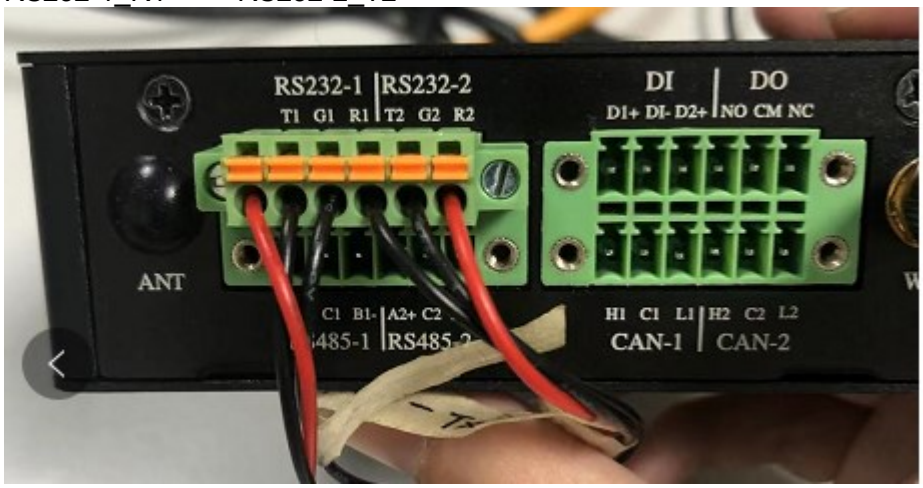
```
# cd /mnt/
# ls
BMB07-factorytest            BMB07-factorytest-V1.0.4.rar  System Volume Information
# cp BMB07-factorytest-V1.0.4.rar /root/
```

4.5 Verification of RS232

Two sets of RS232 are recognized under the system as: /dev/ttymxc2 and /dev/ttymxc3

```
# ls /dev/ttymxc* -l
crw--w----    1 root     root      207,  16 Nov 10 00:58 /dev/ttymxc0
crw-------    1 root     root      207,  18 Jan  1  1970 /dev/ttymxc2
crw-------    1 root     root      207,  19 Jan  1  1970 /dev/ttymxc3
crw-------    1 root     root      207,  20 Jan  1  1970 /dev/ttymxc4
crw-------    1 root     root      207,  21 Jan  1  1970 /dev/ttymxc5
crw-------    1 root     root      207,  22 Jan  1  1970 /dev/ttymxc6
crw-------    1 root     root      207,  23 Jan  1  1970 /dev/ttymxc7
```

ttymxc2 corresponds to the serial port of the device silkscreen "RS232-1" (left side)
ttymxc3 corresponds to the serial port of the device silkscreen "RS232-2" (right side)
The example adopts two sets of RS232 to send and receive each other for verification, and the wiring method is as follows
RS232-1_T1 ------ RS232-2_R2
RS232-1_G1 ------ RS232-2_G2
RS232-1_R1 ------ RS232-2_T2



Command:
#uart_read /dev/ttymxc2 115200 &
#uart_write /dev/ttymxc3 115200 123
#uart_write /dev/ttymxc3 115200 123456
#killall uart_read
#uart_read /dev/ttymxc3 115200 &
#uart_write /dev/ttymxc2 115200 456
#uart_write /dev/ttymxc3 115200 456789

#killall uart_read

```
# uart_read /dev/ttymxc3 115200 &
# open /dev/ttymxc3 speed 115200 8n1

# uart_write /dev/ttymxc2 115200 123
open /dev/ttymxc2 speed 115200 8n1
read [3] [123]
# uart_write /dev/ttymxc2 115200 123456
open /dev/ttymxc2 speed 115200 8n1
read [6] [123456]
# killall uart_read
[3]+  Terminated                 uart_read /dev/ttymxc3 115200
# uart_read /dev/ttymxc2 115200 &
# open /dev/ttymxc2 speed 115200 8n1

# uart_write /dev/ttymxc3 115200 456
open /dev/ttymxc3 speed 115200 8n1
read [3] [456]
# uart_write /dev/ttymxc3 115200 456789
open /dev/ttymxc3 speed 115200 8n1
read [6] [456789]
# killall uart_read
[3]+  Terminated                 uart_read /dev/ttymxc2 115200
#
```

4.6 Verification of RS485

Two sets of RS485 are identified as /dev/ttymxc4 and /dev/ttymxc5 under the system

```
# ls /dev/ttymxc* -l
crw--w----    1 root     root      207,  16 Nov 10 00:58 /dev/ttymxc0
crw-------    1 root     root      207,  18 Nov 10 02:13 /dev/ttymxc2
crw-------    1 root     root      207,  19 Nov 10 02:15 /dev/ttymxc3
crw-------    1 root     root      207,  20 Jan  1  1970 /dev/ttymxc4
crw-------    1 root     root      207,  21 Jan  1  1970 /dev/ttymxc5
crw-------    1 root     root      207,  22 Jan  1  1970 /dev/ttymxc6
crw-------    1 root     root      207,  23 Jan  1  1970 /dev/ttymxc7
#
```

ttymxc4 corresponds to the serial port of the device silkscreen "RS485-1" (left side)
ttymxc5 corresponds to the serial port of the device silkscreen "RS485-2" (right side)
The example adopts two groups of RS485 to send and receive each other for verification, and the wiring method is as follows
RS485-1_A1+ ------ RS485-2_A2+
RS485-1_C1 ------ RS485-2_C2
RS485-1_B1- ------ RS485-2_B2+

Command:
#uart_read /dev/ttymxc4 115200 &
#uart_write /dev/ttymxc5 115200 123
#uart_write /dev/ttymxc5 115200 123456

intel IoT Solutions Alliance

#killall uart_read
#uart_read /dev/ttymxc5 115200 &
#uart_write /dev/ttymxc4 115200 456
#uart_write /dev/ttymxc4 115200 456789
#killall uart_read

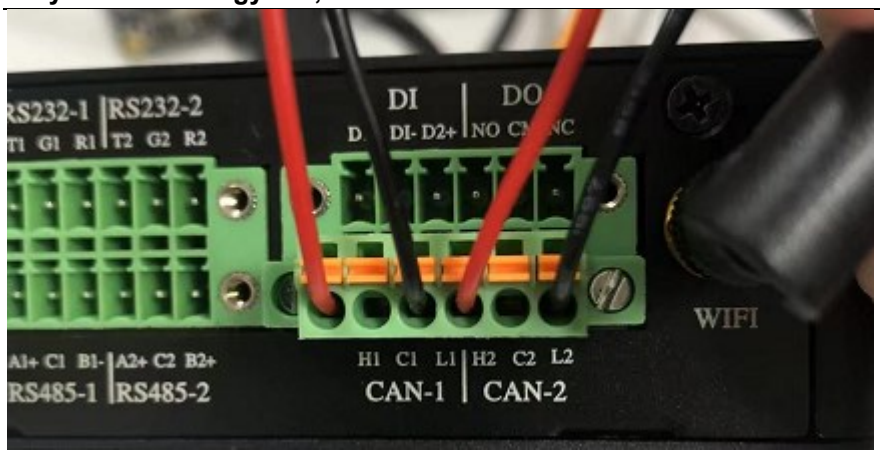
```
# uart_read /dev/ttymxc4 115200 &
# open /dev/ttymxc4 speed 115200 8n1
# uart_write /dev/ttymxc5 115200 123
open /dev/ttymxc5 speed 115200 8n1
read [3] [123]
# uart_write /dev/ttymxc5 115200 123456
open /dev/ttymxc5 speed 115200 8n1
read [5] [12345]
read [1] [6]
# killall uart_read
[3]+  Terminated                 uart_read /dev/ttymxc4 115200
# uart_read /dev/ttymxc5 115200 &
# open /dev/ttymxc5 speed 115200 8n1
# uart_write /dev/ttymxc4 115200 456
open /dev/ttymxc4 speed 115200 8n1
read [3] [456]
# uart_write /dev/ttymxc4 115200 456789
open /dev/ttymxc4 speed 115200 8n1
read [5] [45678]
read [1] [9]
#
```

4.7 CAN functional verification

Two sets of CAN are identified as can0 and can1 under the system

```
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
3: can1: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 10:07:23:6d:c6:12 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.182/24 brd 192.168.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::1207:23ff:fe6d:c612/64 scope link
       valid_lft forever preferred_lft forever
5: eth1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default ql
en 1000
    link/ether 10:07:23:6d:c6:13 brd ff:ff:ff:ff:ff:ff
6: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
7: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether ac:6a:a3:15:23:3f brd ff:ff:ff:ff:ff:ff
#
```

can0 corresponds to the can interface of the device silkscreen "can1" (left side)
can1 corresponds to the can interface of the device silkscreen "can2" (right side)
The example adopts two groups of CAN to send and receive each other for verification, and the wiring method is as follows:
CAN_H1 ------ CAN_H2
CAN_L1 ------ CAN_L2



Command:
#ifconfig can0 down
#ifconfig can1 up
#ip link set can0 type can bitrate 20000
#ip link set can1 type can bitrate 20000
#ifconfig can0 up
#ifconfig can1 up
#candump can0 & (can0 receives data in the background)
#cansend can1 123#1122334455667788 (can1 sends data)
#candump can1 & (can1 receives data in the background)
#cansend can0 123#1122334455667788 (can0 sends data)
As image below:

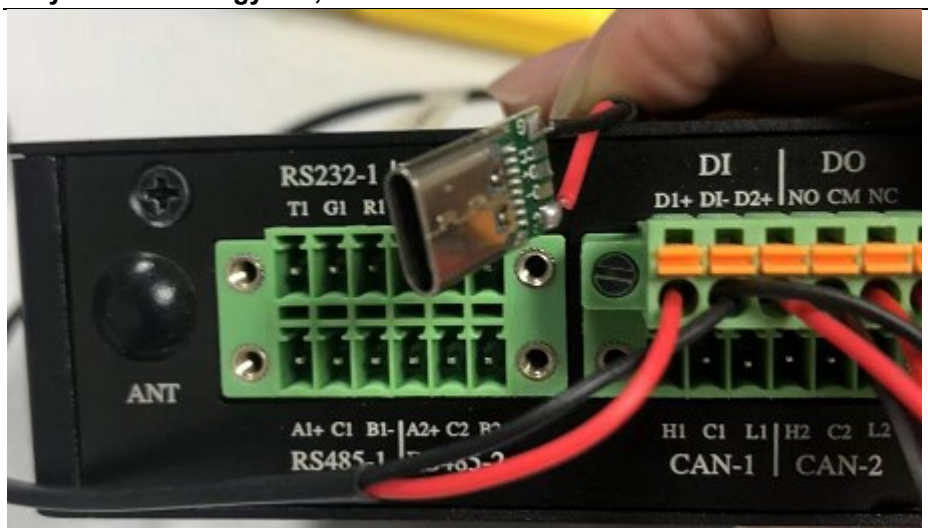
```
# ifconfig can0 down
# ifconfig can1 down
# ip link set can0 type can bitrate 20000
# ip link set can1 type can bitrate 20000
# ifconfig can0 up
# ifconfig can1 up
# candump can0 &
# canse
cansend      cansequence
# cansend can1 123#1122334455667788
#   can0  123   [8]  11 22 33 44 55 66 77 88

# candump can1 &
# cansend can0 123#1122334455667788
#   can0  123   [8]  11 22 33 44 55 66 77 88
  can1  123   [8]  11 22 33 44 55 66 77 88
```

4.8 Verification of DI

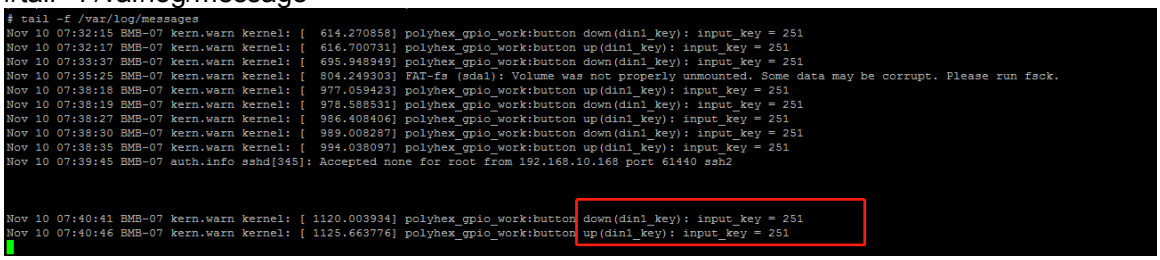
Two sets of DIs are recognized as /dev/input/event1 and /dev/input/event2 under the system

```
# ls /dev/input/event* -l
crw-------    1 root     root       13,  64 Jan  1  1970 /dev/input/event0
crw-------    1 root     root       13,  65 Nov 10 07:22 /dev/input/event1
crw-------    1 root     root       13,  66 Nov 10 07:22 /dev/input/event2
#
```

DIN1 corresponds to the interface of the device silkscreen "D1+ & DI-" (left side)
DIN2 corresponds to the interface of the device silk screen "D2+ & DI-" (right side)
Verification method:
Take DIN1 as an example, draw a type-C socket as shown in the figure, input/disconnect a 5V voltage, query the message event, and you will get the following events;



#tail –f /var/log/message



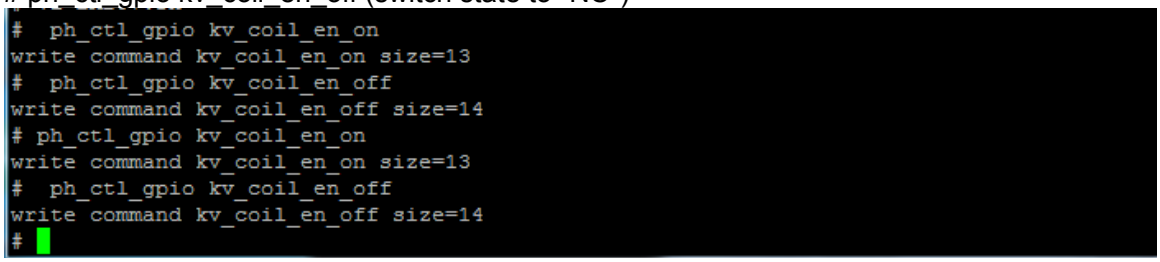
4.9 Verification of DO

The default state of DO is "NC", and the connectivity between CM and NC can be measured by a multimeter;

The state can be switched by the following command

ph_ctl_gpio kv_coil_en_on (switch status is "NO")

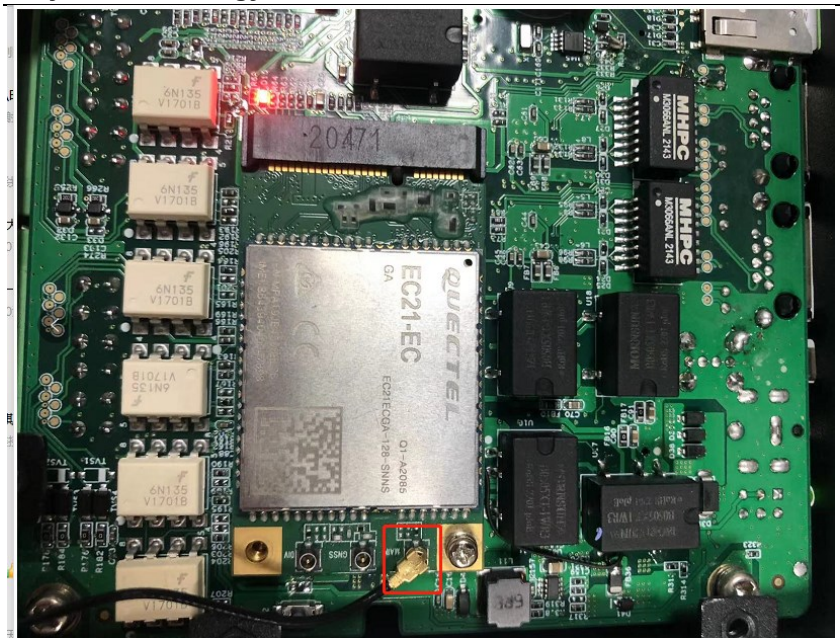
ph_ctl_gpio kv_coil_en_off (switch state to "NC")



4.10 Use of 4G Module

Insert the SIM card, connect to the 4G module (take EC21ECGA-128-SSNS as an example), connect the antenna adapter cable, and support the external antenna.

The module is recognized as /dev/ttyUSB2 under the system, which can be verified by the serial debugging tool microcom



Command:
#ifdown ppp0
#ifup ppp0
#ip a

```
# ifdown ppp0
# ifup ppp0
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: can0: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
3: can1: <NOARP,ECHO> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 10:07:23:6d:c6:12 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.182/24 brd 192.168.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::1207:23ff:fe6d:c612/64 scope link
       valid_lft forever preferred_lft forever
5: eth1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 10:07:23:6d:c6:13 brd ff:ff:ff:ff:ff:ff
6: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
7: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether ac:6a:a3:15:23:3f brd ff:ff:ff:ff:ff:ff
8: wwan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 26:43:fb:f3:15:66 brd ff:ff:ff:ff:ff:ff
10: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 3
    link/ppp
    inet 10.214.138.254 peer 10.64.64.64/32 scope global ppp0
       valid_lft forever preferred_lft forever
#
```

Example:
#ping -I ppp0 www.baidu.com

```
# ping -I ppp0 www.baidu.com
PING www.a.shifen.com (112.80.248.76) from 10.214.138.254 ppp0: 56(84) bytes of data.
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=1 ttl=55 time=1206 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=2 ttl=55 time=201 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=3 ttl=55 time=110 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=4 ttl=55 time=107 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=5 ttl=55 time=105 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=6 ttl=55 time=104 ms
64 bytes from 112.80.248.76 (112.80.248.76): icmp_seq=7 ttl=55 time=106 ms
```

#microcom /dev/ttyUSB2
AT+CPIN? #SIM card verification

AT+CIMI #Query SIM card number CIMI
AT+CGSN #Query module IMEI
AT+CSQ #Query signal strength

```
# microcom /dev/ttyUSB2

+CPIN: READY

OK

460065021200496

OK

864394040047898

OK

+CSQ: 23,99

OK
```

4.11 Use of Lora Module

Connect the Lora module (take HLM5934-H01 as an example), connect the antenna adapter cable, and support the external antenna
The module is recognized as /dev/spidev1.0 under the system

```
# ls /dev/spidev* -l
crw-------    1 root     root      153,   0 Jan  1  1970 /dev/spidev0.0
crw-------    1 root     root      153,   1 Jan  1  1970 /dev/spidev1.0
#
```

Our company can provide compiled executable scripts to start the lora module

```
# cd lora/
# ls
global_conf.json  lora_pkt_fwd      reset_lgw.sh
```

#./lora_pakt_fwd

intel IoT Solutions Alliance

```
# ./lora_pkt_fwd &
# *** Packet Forwarder ***
Version: 2.1.0
*** SX1302 HAL library version info ***
Version: 2.1.0;
***
INFO: Little endian host
INFO: found configuration file global_conf.json, parsing it
INFO: global_conf.json does contain a JSON object named SX130x_conf, parsing SX1302 parameters
INFO: com_type SPI, com_path /dev/spidev1.0, lorawan_public 1, clksrc 0, full_duplex 0
INFO: antenna_gain 0 dBi
INFO: Configuring legacy timestamp
INFO: no configuration for SX1261
INFO: Configuring Tx Gain LUT for rf_chain 0 with 16 indexes for sx1250
INFO: radio 0 enabled (type SX1250), center frequency 470600000, RSSI offset -207.000000, tx enabled 1, single input mode 1
INFO: radio 1 enabled (type SX1250), center frequency 471400000, RSSI offset -207.000000, tx enabled 0, single input mode 1
INFO: Lora multi-SF channel 0>  radio 0, IF -300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 1>  radio 0, IF -100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 2>  radio 0, IF 100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 3>  radio 0, IF 300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 4>  radio 1, IF -300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 5>  radio 1, IF -100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 6>  radio 1, IF 100000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 7>  radio 1, IF 300000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora std channel> radio 1, IF -200000 Hz, 250000 Hz bw, SF 7, Explicit header
INFO: FSK channel> radio 1, IF 300000 Hz, 125000 Hz bw, 50000 bps datarate
INFO: global_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
INFO: gateway MAC address is configured to FFFEDCA6320E9516
INFO: server hostname or IP address is configured to "localhost"
INFO: upstream port is configured to "1700"
INFO: downstream port is configured to "1700"
INFO: downstream keep-alive interval is configured to 10 seconds
INFO: statistics display interval is configured to 30 seconds
INFO: upstream PUSH_DATA time-out is configured to 100 ms
INFO: packets received with a valid CRC will be forwarded
INFO: packets received with a CRC error will NOT be forwarded
INFO: packets received with no CRC will NOT be forwarded
INFO: Beaconing period is configured to 0 seconds
INFO: Beaconing signal will be emitted at 869525000 Hz
INFO: Beaconing datarate is set to SF9
INFO: Beaconing modulation bandwidth is set to 125000Hz
INFO: Beaconing TX power is set to 14dBm
INFO: Beaconing information descriptor is set to 0
INFO: global_conf.json does contain a JSON object named debug_conf, parsing debug parameters
INFO: got 2 debug reference payload
INFO: reference payload ID 0 is 0xCAFE1234
INFO: reference payload ID 1 is 0xCAFE2345
INFO: setting debug log file name to loragw_hal.log
write command lora_en_off size=11
write command lora_pwd_off size=12
write command lora_pwd_on size=11
write command lora_en_on size=10
write command lora_rst_on size=11
CoreCell reset through /dev/lora_reset...
CoreCell power enable through /dev/lora_en...
write command lora_en_on size=10
write command lora_rst_off size=12
write command lora_rst_on size=11
Opening SPI communication interface
Note: chip version is 0x10 (v1.0)

# INFO: Configuring SX1250_0 in single input mode
INFO: Configuring SX1250_1 in single input mode
INFO: using legacy timestamp
INFO: LoRa Service modem: configuring preamble size to 8 symbols
ARB: dual demodulation disabled for all SF
INFO: [main] concentrator started, packet can now be received
INFO: concentrator EUI: 0x0016c001f10a62ef
```

4.12 Verification of RTC

Chip model: HYM8563S
Confirm that the HYM8653S driver module is loaded successfully
#dmesg | grep rtc-hym8653

```
# dmesg |grep rtc-hym8563
[    2.699420] rtc-hym8563 1-0051: registered as rtc0
[    2.705764] rtc-hym8563 1-0051: setting system clock to 2030-08-04T19:31:40
TC (1912102300)
#
```

Set and read RTC time
#hwclock --systohc
#hwclock --show

intel IoT Solutions Alliance

```
# hwclock --systohc
# hwclock --show
Wed Nov  9 12:12:16 2022  0.000000 seconds
#
```