BPC-iMX6Se-01 Industrial Computer User Guide

Version: V1.0（2023-05）

Complied by: Polyhex Technology Company Limited（http://www.polyhex.net/）

BPC-iMX6Se-01 Industrial Computer is designed with security in mind. Based on the NXP i.MX 6 series processor, which is composed of a EMB-iMX6Se-01 (main board), and a steel and aluminum enclosure. It combines various types of harsh environment resistance features, including ruggedness, dustproof, anti-vibration, shock resistance, wide temperature, portability and other indicators; and also provides multi-core solutions based on 6DualLite, 6Quad, 6Dual and 6Solo, widely used in commercial and industrial.

Figure 1 BPC-iMX6Se-01

REVISION HISTORY		
Rev.	**Date**	**Description**
1.0	2023.05.26	First edition

Index

Chapter 1 Security

1.1. Safety Precaution

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

Table 1 Terms and conventions

Symbol	Meaning
Warning! 	Always disconnect the power cord from the chassis whenever there is no workload required on it. Do not connect the power cable while the power is on. A sudden rush of power can damage sensitive electronic components. Only experienced electricians should open the chassis.
Caution! 	Always ground yourself to remove any static electric charge before touching *BPC-iMX6Se-01*. 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.

1.2. Safety Instruction

To avoid malfunction or damage to this product please observe the following:

1．Disconnect the device from the DC power supply before cleaning. Use a damp cloth. Do not use liquid detergents or spray-on detergents.

2．Keep the device away from moisture.

3．During installation, set the device down on a reliable surface. Drops and bumps will lead to damage.

4．Before connecting the power supply, ensure that the voltage is in the required range, and the way of wiring is correct.

5．Carefully put the power cable in place to avoid stepping on it.

6．If the device is not used for a long time, power it off to avoid damage caused by sudden

overvoltage.

7．Do not spill liquid into the venting holes of the enclosure, as this could cause fire or electric shock.

8．For safety reasons, the device can only be disassembled by professional personnel.

9．If one of the following situations occur, 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.

10．Do not place the device in a place where the ambient temperature is below -20°C (-4°F) or above 70°C (158°F). This will damage the machine. It needs to be kept in an environment at controlled temperature.

11．Due to the sensitive nature of the equipment, it must be stored in a restricted access location, only accessible by qualified engineer.

DISCLAIMER: Polyhex disclaims all responsibility for the accuracy of any statement of this instructional document.

1.3. Declaration of Compliance

CE: This equipment has passed CE certified.

FCC: This equipment has passed FCC certified.

RoHS: This equipment is manufactured in compliance with RoHS regulations.

REACH: This equipment is manufactured in compliance with REACH regulations.

1.4. Technical Support

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

Email: *info@polyhex.net*

Chapter 2 BPC-iMX6Se-01 Industrial Computer Introduction

BPC-iMX6Se-01 Industrial Computer is a small embedded PC based on the NXP i.MX 6 series processor. With compact appearance and rich I/Os, it is mostly used in digital signage, kiosk, network security, IoT gateway and other fields.

Main features:

- Cortex-A9-based solution
- i.MX 6Quad family
  - a quad-core, up to 1.2 GHz with 1 MB of L2 cache
  - Support hardware accelerated graphics, 64-bit DDR3 or 2-channel, 32-bit LPDDR2
  - Integrated FlexCAN and MLB busses, PCI Express® and SATA2 provide excellent connectivity
  - Integration of dual lane MIPI display ports, MIPI camera port and HDMI v1.4
- i.MX 6Dual family
  - a dual-core, up to 1.2 GHz with 1 MB of L2 cache
  - Support hardware accelerated graphics, 64-bit DDR3 or 2-channel, 32-bit LPDDR2
  - Leveraging the same integration of the i.MX 6Quad family
- i.MX 6DualLite family
  - a dual-core, up to 1.0 GHz with 512 KB of L2 cache
  - Support 64-bit DDR3 or 2-channel, 32-bit LPDDR2
  - Integrated FlexCAN and MLB busses, PCI Express, LVDS
  - Support MIPI cameras and displays as well as HDMI v1.4
- i.MX 6Solo family
  - a single core, up to 1.0 GHz with 512 KB of L2 cache
  - Support 32-bit DDR3/LPDDR2
  - Integrated LVDS, MIPI display, MIPI camera port, HDMI v1.4, FlexCAN and MLB
- Support Yocto 2.5.2, Android 9.0, Ubuntu 16.04

2.1. Overview

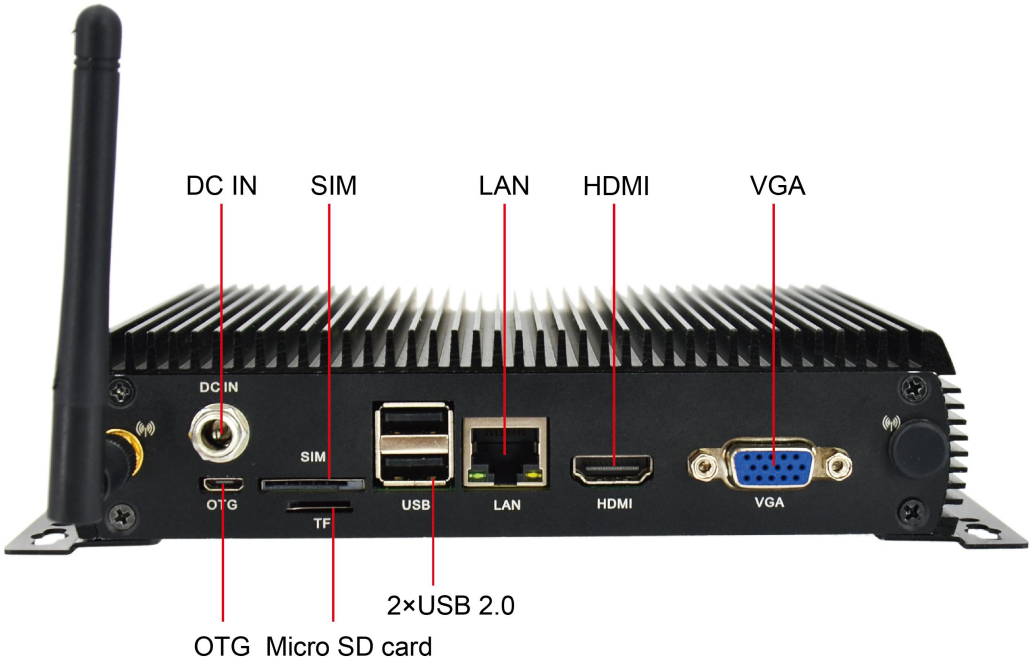


Figure 2 IO of BPC-iMX6Se-01

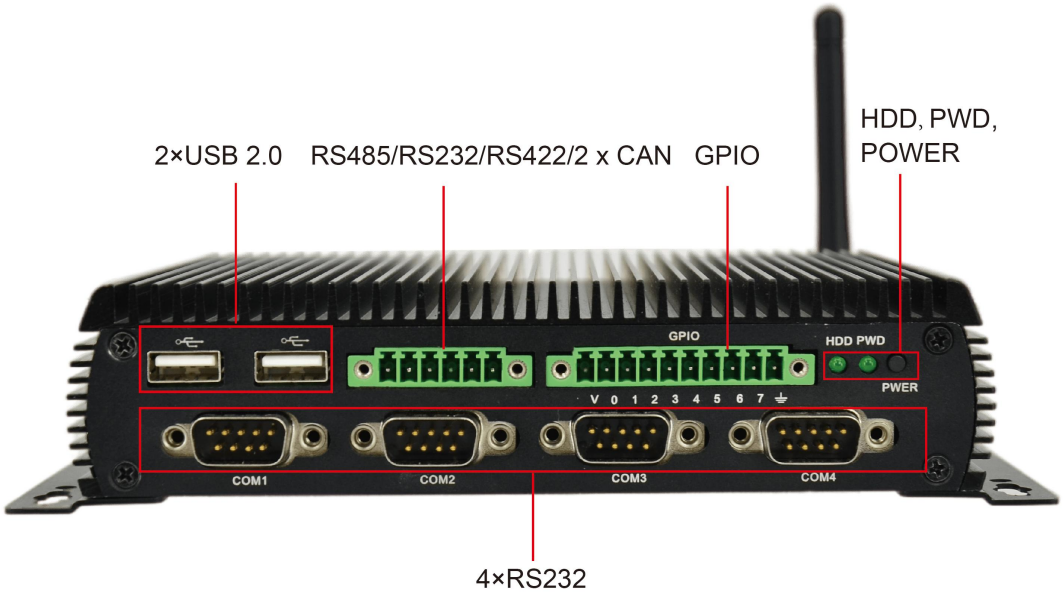


Figure 3 IO of BPC-iMX6Se-01

BPC-iMX6Se-01 Industrial Computer uses EMB-iMX6Se-01 Board as the main board, which supports Gigabit Ethernet, wifi and other functions, with dustproof, shock and vibration resistance, etc.. The data specifications are as follows.

Table 2 BPC-iMX6Se-01 specification

System	
Motherboard	EMB-iMX6Se-01
Model	BPC-iMX6Se-01
CPU	(1) NXP i.MX 6DualLite 1.0GHz (Commercial) (2) NXP i.MX 6Quad 1.0GHz (Extended Commercial optional) (3) NXP i.MX 6Solo (Industrial grade optional) (4) NXP i.MX 6Dual (Industrial grade optional)
Memory	Onboard 1GB DDR3 (2GB optional)
Storage	Onboard 8GB eMMC (16GB/32GB/64GB/128GB/256GB optional)
OS	Yocto 2.5.2, Android 9.0, Ubuntu 16.04
Communication	
Gigabit Network	1 x 10Mbps/100Mbps/1000Mbps RJ45 network port
Wi-Fi & Bluetooth	External Wi-Fi SMA antenna interface, (1) 1 x 2.4GHz WiFi, support IEEE 802.11 b/g (2) Option: 1 x 2.4GHz WiFi and BT V2.1/ V3.0/ V4.0, support IEEE 802.11 b/g/n
Video	
Display	(1) 1 x VGA output, connector is DB-15 (2) 1 x HDMI v1.4 output, the connector is Type A HDMI female
External I/O Interface	
DC Block	1 x DC socket, supports 5.5mm x 2.1mm plug
USB	(1) 1 x Micro USB OTG (2) 2 x USB 2.0 Host, the connector is double layer Type-A interface (3) 2 x USB 2.0 Host, the connector is Type-A interface
UART	(1) 4 x RS232 (2) 1 x RS485/RS232/RS422/2 x CAN(default RS485)
GPIO	1 x GPIO

Slot	(1) 1 x SIM slot (2) 1 x Micro SD slot
LED & Key	(1) 1 x System status indicator (2) 1 x Power status indicator (3) 1 x ON/OFF key
Power Supply	
Power Input	DC 12V/2A
Mechanical & Environmental	
Enclosure Material	Steel and aluminum alloy
Dimension (W x D x H)	169.42mm x 124mm x 38.9mm
Weight	780g
Heat Dissipation	No fan, heat dissipation through the enclosure
Operating Temp.	-20°C~70°C

2.2. Composition

BPC-iMX6Se-01 Industrial Computer assembly consists of these main components: EMB-iMX6Se-01 board, enclosure and antenna.

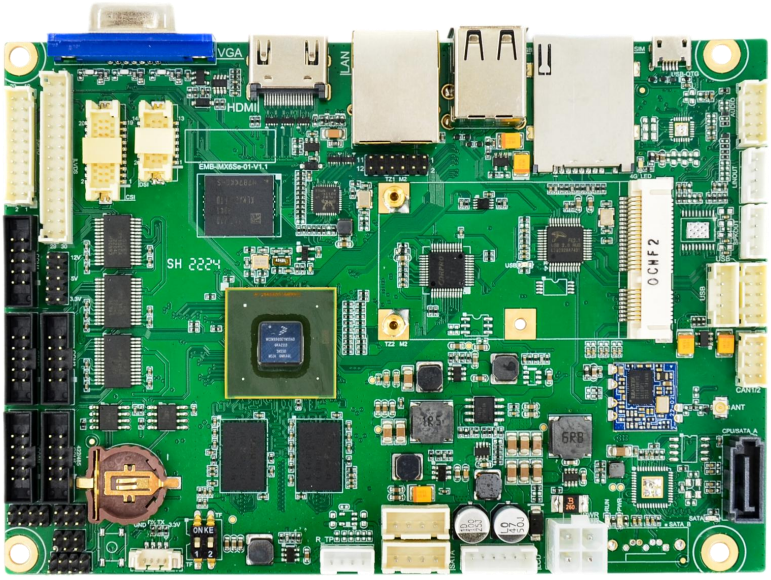


Figure 4 EMB-iMX6Se-01

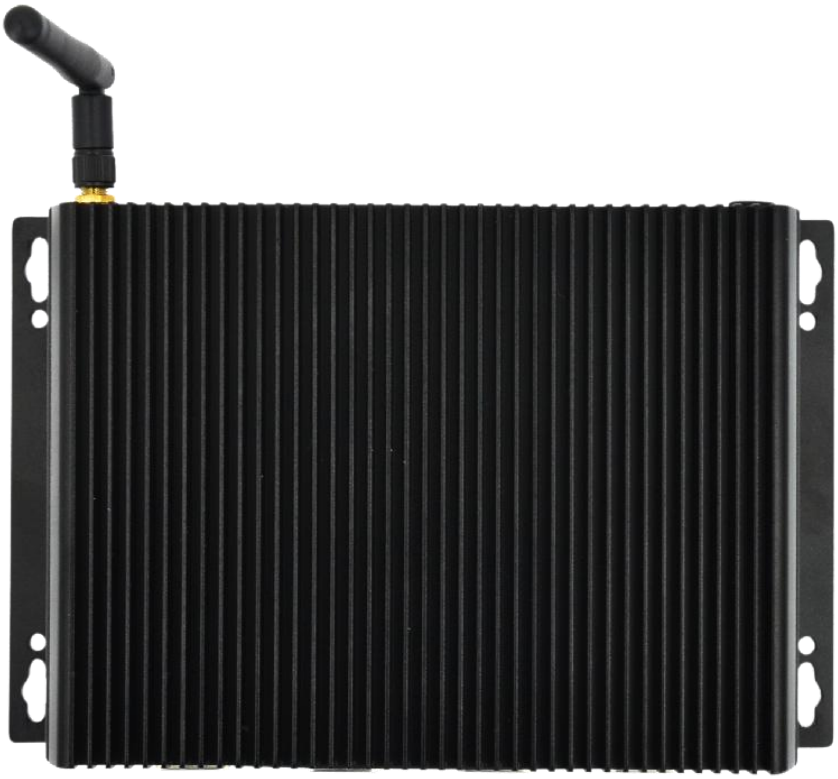


Figure 5 Enclosure and antenna

2.3. External Interface

2.3.1. Power Interface

BPC-iMX6Se-01 Industrial Computer provides 1 power connector (DC socket), with default DC 12V/2A power input. As shown in the figure below.

Figure 6 DC-IN Interface

2.3.2. USB Interface

BPC-iMX6Se-01 Industrial Computer has two USB controllers and PHY, supports USB 2.0. Two USB 2.0 interfaces with dual-layer Type-A connector and a Micro USB OTG interface, and another two USB 2.0 interfaces with Type-A connector. As shown in the figures below.

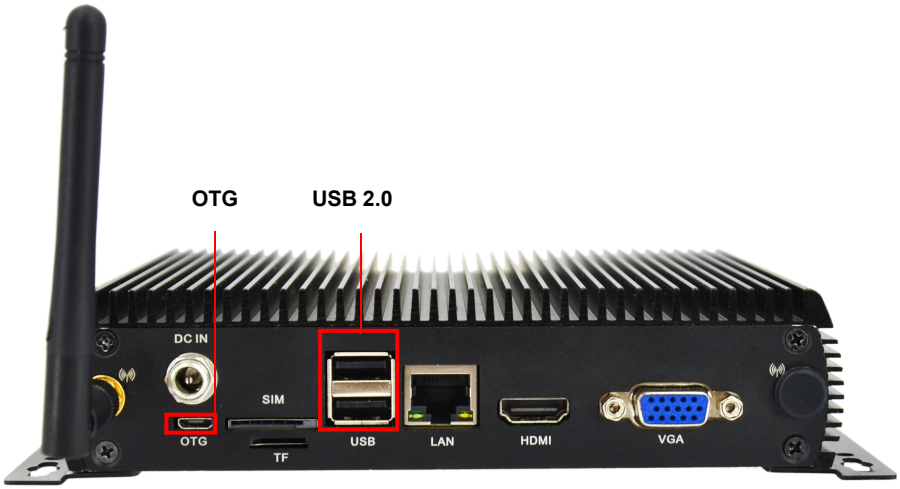


Figure 7 Micro USB OTG and USB 2.0

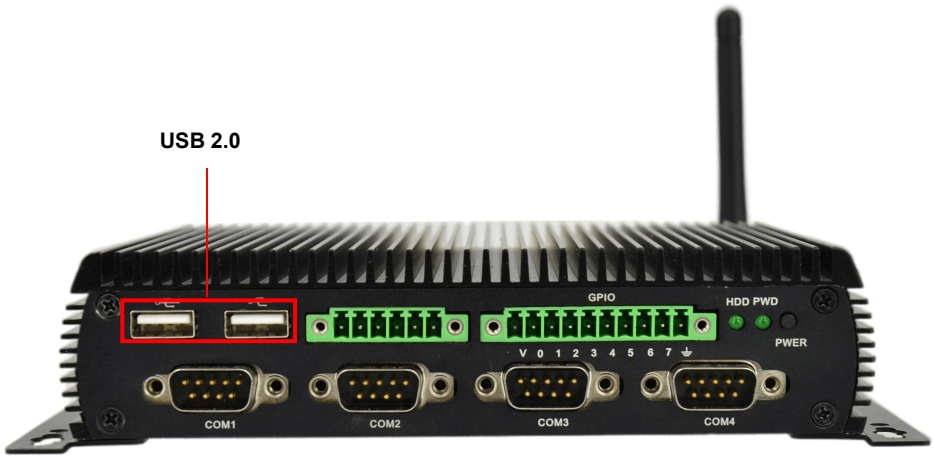


Figure 8 USB 2.0

2.3.3. Ethernet Interface

BPC-iMX6Se-01 Industrial Computer provides an independent MAC RJ45 Ethernet port (Network port: LAN), connect Industrial Computer to network through the network cable of RJ45 connector. A set of status indicators below the interface displays the status signal, the green one is a Link connection indicator, and the other yellow is an Active signal transmission indicator.

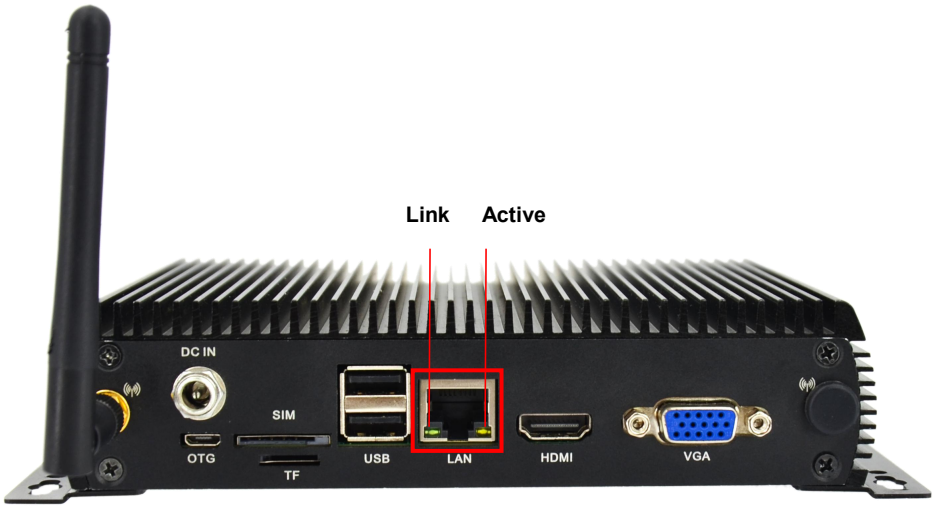


Figure 9 Ethernet Interface

Table 3 Ethernet port status

LED	Color	Description
Link	Green	Light, the network cable is plugged in, network connection status is good

Active	Yellow	Blinking, network data is being transmitted

2.3.4. Display Interface

- One is an HDMI interface, and the connector is an A-type HDMI female socket, which is used to connect a monitor, TV or projector. HDMI resolution is supported up to 1366x768. Audio supports 32 channel audio, output supports 1 S/PDIF audio and eARC input support.
- One is a VGA interface with DB-15 connector for connecting a monitor, KVM, digital signage, or projector. Supports VGA resolution up to 1920x1080.

BPC-iMX6Se-01 Industrial Computer supports VGA and HDMI dual screen display by default. The device is connected to the corresponding display with an HDMI cable and a VGA cable, and the system interface can be seen after power on.

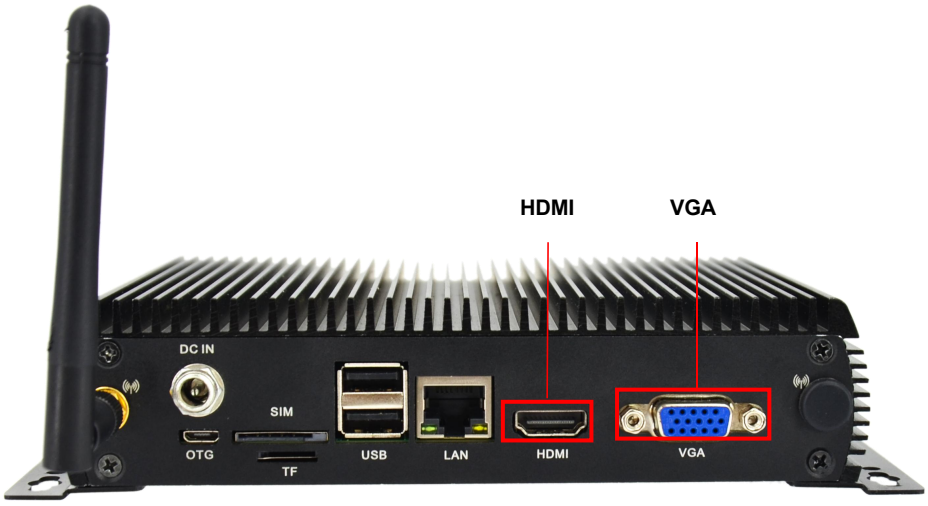


Figure 10 Display Interface

The HDMI pin sequence is as shown in the figure:

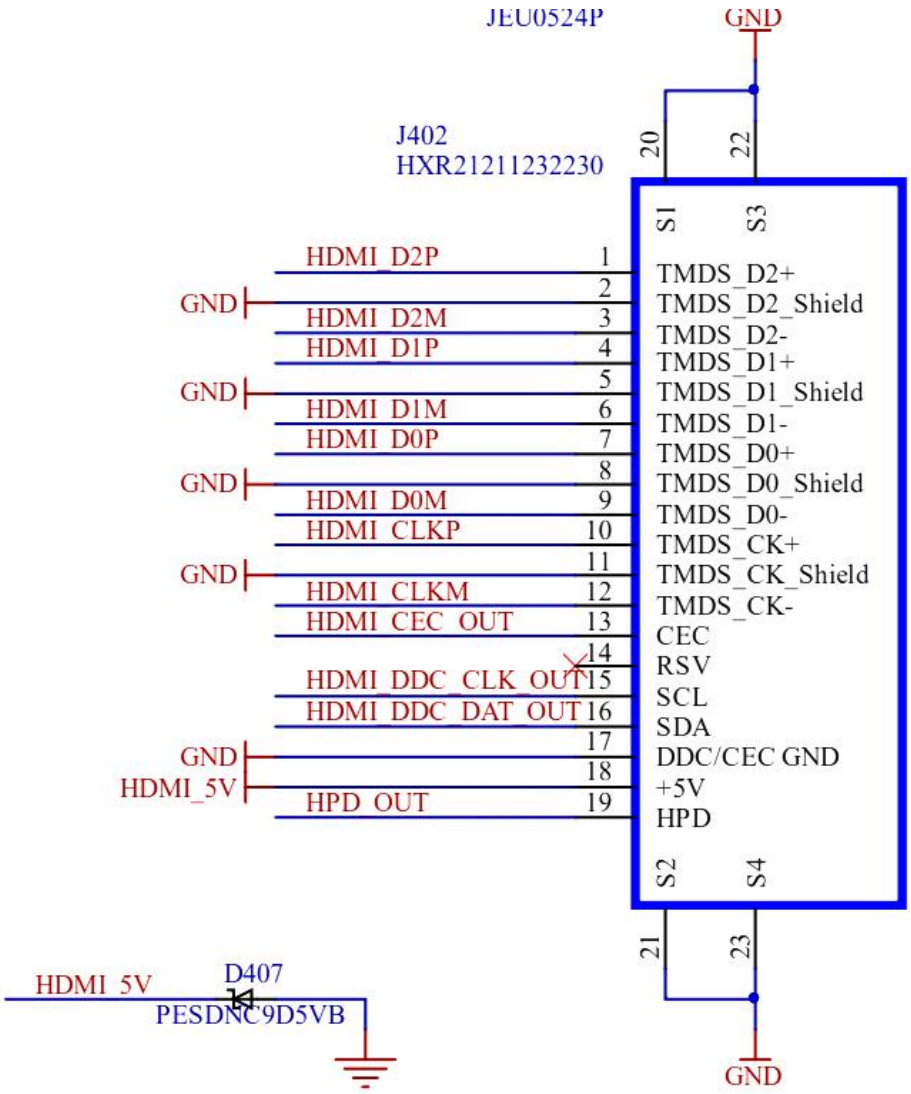


Figure 11

The HDMI interface is defined as follows:

Table 4 Pin definition of HDMI interface

Pin	Definition	Pin	Definition
1	HDMI_D2P	2	GND
3	HDMI_D2M	4	HDMI_D1P
5	GND	6	HDMI_D1M
7	HDMI_D0P	8	GND
9	HDMI_D0M	10	HDMI_CLKP
11	GND	12	HDMI_CLKM

13	HDMI_CEC_OUT	14	NC
15	HDMI_DDC_CLK_OUT	16	HDMI_DDC_DAT_OUT
17	GND	18	HDMI_5V
19	HPD_OUT	20	GND
21	GND	22	GND
23	GND		

The VGA pin sequence is as shown in the figure:

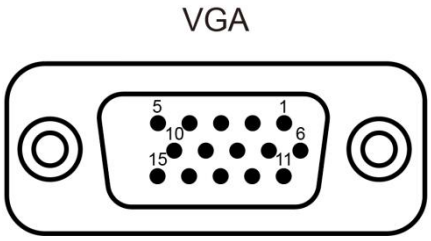


Figure 12

The VGA interface is defined as follows:

Table 5 Pin definition of VGA interface

Pin	Definition	Description
1	VGA_RED	Red component signal
2	VGA_GRN	Green component signal
3	VGA_BLU	Blue component signal
4	NC	Not used
5	GND	GND
6	GND	GND
7	GND	GND
8	GND	GND
9	VGA_5V	5V input
10	GND	GND
11	NC	Not used
12	VGA_I2C_SDA	Serial data signal

13	VGA_HSYNC	Horizontal synchronization (line synchronization)
14	VGA_VSYNC	Vertical synchronization (field synchronization)
15	VGA_I2C_SCL	Serial clock signal

2.3.5. RS485/RS232/RS422/CAN Interface

NOTE

The default configuration is an RS485 interface. RS485, RS232, RS422 and CAN on the same socket, only one can be used at the same time.

BPC-iMX6Se-01 Industrial Computer has 4 x RS232 interfaces (Port: COM1~COM4), and 1 x RS485/RS232/RS422/2 x CAN interface. As shown in the figure below:

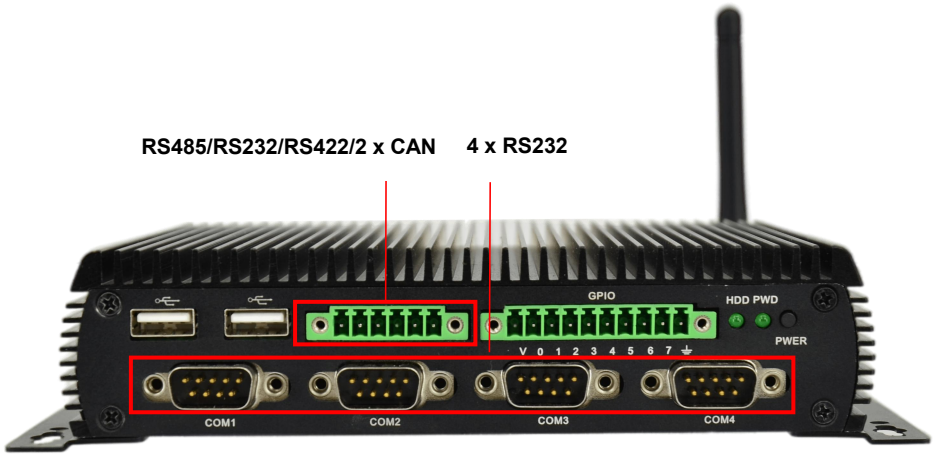


Figure 13 URAT Interface

The RS485 /RS232/ RS422/ 2 x CAN pin sequence is as shown in the figure:

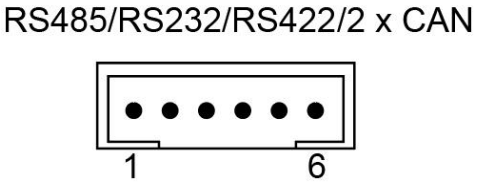


Figure 14

Table 6 Pin definition of RS485/ RS232/ RS422/ 2 x CAN (RS485 by default)

Pin	RS485	RS232	RS422	CAN	Device node
1	GND	GND	GND	GND	/dev/ttymxc4

2	RS485_DATA-	NC	RS422_TX-	CAN1_H	
3	RS485_DATA+	RS232_RXD	RS422_TX+	CAN1_L	
4	GND	GND	GND	GND	
5	NC	RS232_TXD	RS422_RX+	CAN2_H	
6	NC	NC	RS422_RX-	CAN2_L	

The 4 x RS232 pin sequence is as shown in the figure:

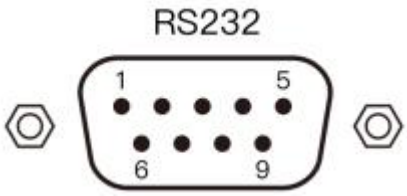


Figure 15

Table 7 Device nodes of COM1-4(RS232)

Function Name	IO Name	Description	Device node
COM1	RS232_RXD1	RS232 receiver	/dev/ttymxc0
	RS232_TXD1	RS232 sender	
COM2	RS232_RXD2	RS232 receiver	/dev/ttymxc1
	RS232_TXD2	RS232 sender	
COM3	RS232_RXD3	RS232 receiver	/dev/ttymxc2
	RS232_TXD3	RS232 sender	
COM4	RS232_RXD4	RS232 receiver	/dev/ttymxc3
	RS232_TXD4	RS232 sender	

The 4 x RS232 interface is defined as follows:

Table 8 Pin definition of COM1-4(RS232)

Pin	Definition	Pin	Definition
1	NC	2	RS232_RXD
3	RS232_TXD	4	NC

5	GND	6	NC
7	NC	8	NC
9	NC		

2.3.6. GPIO

BPC-iMX6Se-01 Industrial Computer provides a GPIO interface, as shown below:

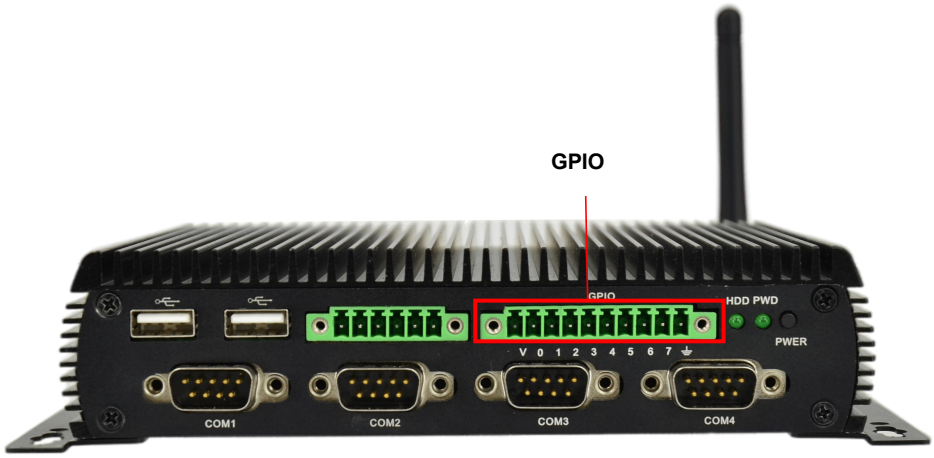


Figure 16 GPIO Interface

The GPIO interface is defined as follows:

Table 9 Pin definition of GPIO

Pin	Definition	Pin	Definition
V	VDD 3.3V	0	SIO_GP70
1	SIO_GP71	2	SIO_GP72
3	SIO_GP73	4	SIO_GP74
5	SIO_GP75	6	SIO_GP76
7	SIO_GP77	G	SIO_GND

2.3.7. LED & Key

BPC-iMX6Se-01 Industrial Computer provides two LED and a POWER key, as shown in the figure below.

Figure 17 LED and Key

Table 10 LED & Key description

LED & Key	Status	Description
HDD	Lighting	Device works normally
	off	Device works abnormally
PWD	Lighting	Power is on
	off	Power is off
POWER key	Short press	Screen display shutdown menu, with options for shutdown, hibernation and other functions
	Long press for 8 seconds	Force device shutdown
	Long press for 15 seconds	Force device restart

2.3.8. Slot

BPC-iMX6Se-01 Industrial Computer provides two card slots, one is standard SIM card slot and one is Micro SD card slot, as shown below:

Figure 18 SIM and Micro SD card slot

When inserting the SIM / Micro SD card into the corresponding slot, you need to pay attention to the insertion and removal direction (the direction position has been marked on the device).

2.4. Packing List

- ✓ 1 x WiFi external antenna
- ✓ 4 x M3-5 Black screw
- ✓ 2 x Wall bracket
- ✓ 1 x BPC-iMX6Se-01 box

Chapter 3 Getting started

3.1. Installation

After receiving the product, install the accessories as follows.

1. If you need to hang the device on the wall, use 4 black screws to fix 2 wall brackets on the enclosure, and then fix the device on the wall.
2. Install the WiFi antenna to the WiFi antenna connector as shown in the following figure.

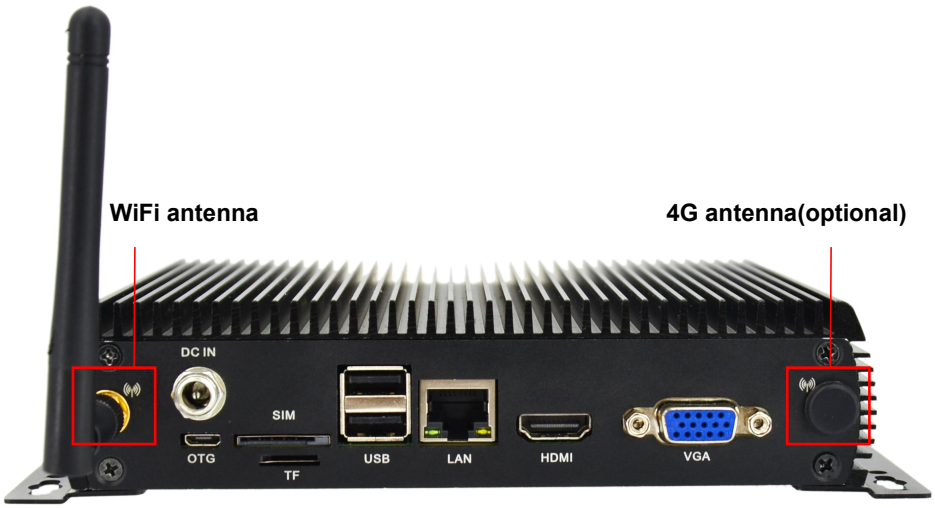


Figure 19 WiFi antenna

3. Connect the power adapter to the DC-IN connector of enclosure. When the HDD and PWD LED are on, it proves that the Industrial Computer is powered on.

Figure 20 Power adapter

3.2. Power on

Note: BPC-iMX6Se-01 Industrial Computer factory default for eMCC boot and Ubuntu 16.04. If you need to change to other boot modes or OS, please contact our engineer for modification before leaving the factory, and do not disassemble the machine by yourself.

Chapter 4 Software Application Examples

4.1. Usage of Ethernet

Network port (LAN), port number: eth0

1. Open a Terminal and type the command to query the network port.

```
ip a
```

```
root@polyhex:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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 gr
oup default qlen 1000
    link/ether 10:07:23:6e:02:eb brd ff:ff:ff:ff:ff:ff
    inet6 240e:36d:d82:3900:5c06:d6b7:d784:1575/64 scope global temporary dynami
c
       valid_lft 216745sec preferred_lft 85468sec
    inet6 240e:36d:d82:3900:80a7:b8ba:6b91:2fb6/64 scope global mngtmpaddr nopre
fixroute dynamic
       valid_lft 216745sec preferred_lft 130345sec
    inet6 240e:36d:d82:3900:1207:23ff:fe6e:2eb/64 scope global mngtmpaddr dynami
c
       valid_lft 216745sec preferred_lft 130345sec
    inet6 fe80::1207:23ff:fe6e:2eb/64 scope link
       valid_lft forever preferred_lft forever
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
6: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group
 default qlen 1000
    link/ether 0c:c6:55:75:5d:39 brd ff:ff:ff:ff:ff:ff
root@polyhex:~#
```

2. Apply ping command.

```
ping 192.168.1.1
```

```
root@polyhex:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.664 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.610 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.585 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.562 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.436 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=0.582 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=64 time=0.573 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=64 time=0.551 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=64 time=0.424 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=64 time=0.527 ms
64 bytes from 192.168.1.1: icmp_seq=11 ttl=64 time=0.441 ms
64 bytes from 192.168.1.1: icmp_seq=12 ttl=64 time=0.392 ms
64 bytes from 192.168.1.1: icmp_seq=13 ttl=64 time=0.534 ms
64 bytes from 192.168.1.1: icmp_seq=14 ttl=64 time=0.560 ms
64 bytes from 192.168.1.1: icmp_seq=15 ttl=64 time=0.593 ms
64 bytes from 192.168.1.1: icmp_seq=16 ttl=64 time=0.455 ms
64 bytes from 192.168.1.1: icmp_seq=17 ttl=64 time=0.426 ms
64 bytes from 192.168.1.1: icmp_seq=18 ttl=64 time=0.644 ms
64 bytes from 192.168.1.1: icmp_seq=19 ttl=64 time=0.544 ms
64 bytes from 192.168.1.1: icmp_seq=20 ttl=64 time=0.593 ms
64 bytes from 192.168.1.1: icmp_seq=21 ttl=64 time=0.424 ms
```

3. Query the speed of the network port.

```
ethtool eth0
```

```
root@polyhex:~# sudo ethtool eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: Symmetric
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: Symmetric
        Advertised auto-negotiation: Yes
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full
                                             1000baseT/Full
        Link partner advertised pause frame use: Symmetric
        Link partner advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 1
        Transceiver: external
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: d
        Link detected: yes
root@polyhex:~#
```

4.2. Usage of WiFi

- unplug the network cable, and connect the device to WiFi (polyhex_mi) via the command;

or connect to available WiFi by clicking the network icon in the lower left corner of the display.

```
nmcli r wifi on       #Enable WiFi
nmcli dev wifi        #Find available WiFi
nmcli dev wifi connect “SSID” password “PASSWORD” ifname wlan0    #Connect to the specified WiFi
```

```
root@polyhex[16:22:28]:~
$ nmcli r wifi on
root@polyhex[16:22:46]:~
$ nmcli dev wifi
*  SSID                        MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
*  polyhex_mi1                 Infra  3     54 Mbit/s   75      ***   WPA2
   tsc_wh                      Infra  6     54 Mbit/s   69      ***   WPA1 WPA2
   --                          Infra  11    54 Mbit/s   65      ***   WPA2
   tsc                         Infra  11    54 Mbit/s   59      ***   WPA2
   --                          Infra  11    54 Mbit/s   55      **    WPA2
   DIRECT-0b-HP M132 LaserJet  Infra  3     54 Mbit/s   42      **    WPA2
   ChinaNet-polyhex            Infra  11    54 Mbit/s   35      **    WPA1 WPA2
   polyhex-3                   Infra  11    54 Mbit/s   25      *     WPA1 WPA2
   HUISHI-503                  Infra  1     54 Mbit/s   19      *     WPA1 WPA2
   ????                    Infra  13    54 Mbit/s  19      *     WPA1 WPA2
   ChinaNet-ASqW               Infra  6     54 Mbit/s   15      *     WPA1 WPA2
   ZY_WH_7A31_2                Infra  11    54 Mbit/s   15      *     WPA1 WPA2
   Ns                          Infra  11    54 Mbit/s   15      *     WPA1 WPA2
   htsc-wuhan                  Infra  1     54 Mbit/s   12      *     WPA1 WPA2
root@polyhex[16:23:04]:~
$ nmcli dev wifi connect "polyhex_mi1" password "bohai2021" ifname wlan0
Device 'wlan0' successfully activated with '90a3c624-5322-49b8-ac84-f191a3ca0653'.
root@polyhex[16:24:31]:~
$
```

- Apply ping command to check the network connection status.

```
ping 192.168.1.1
```

```
root@polyhex[16:24:31]:~
$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=16.8 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=3.78 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=4.83 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=63 time=18.2 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=63 time=3.71 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=63 time=6.38 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=63 time=3.43 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=63 time=4.61 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=63 time=3.84 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=63 time=20.9 ms
64 bytes from 192.168.1.1: icmp_seq=11 ttl=63 time=5.82 ms
64 bytes from 192.168.1.1: icmp_seq=12 ttl=63 time=4.23 ms
64 bytes from 192.168.1.1: icmp_seq=13 ttl=63 time=2.98 ms
64 bytes from 192.168.1.1: icmp_seq=14 ttl=63 time=6.25 ms
64 bytes from 192.168.1.1: icmp_seq=15 ttl=63 time=3.26 ms
64 bytes from 192.168.1.1: icmp_seq=16 ttl=63 time=3.61 ms
```

4.3. Usage of USB

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

```
df -h
```

```
root@polyhex[03:00:04]:~
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        14G  2.7G   11G  21% /
devtmpfs        499M     0  499M   0% /dev
tmpfs           500M  484K  499M   1% /dev/shm
tmpfs           500M   14M  486M   3% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           500M     0  500M   0% /sys/fs/cgroup
/dev/mmcblk2p1  490M  7.8M  482M   2% /boot
tmpfs           100M   44K  100M   1% /run/user/0
/dev/sda1       500M   32M  469M   7% /media/root/68BA-C562
/dev/sda2        29G  3.4G   25G  13% /media/root/79de8ff0-265b-451f-be52-87356c
5f68c0
root@polyhex[03:00:13]:~
```

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

```
fdisk -l
```

```
root@polyhex[02:54:25]:~
$ fdisk -l
Disk /dev/ram0: 64 MiB, 67108864 bytes, 131072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/ram1: 64 MiB, 67108864 bytes, 131072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/ram2: 64 MiB, 67108864 bytes, 131072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes


Disk /dev/ram3: 64 MiB, 67108864 bytes, 131072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
```

```
Disk /dev/mmcblk2boot1: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mmcblk2boot0: 4 MiB, 4194304 bytes, 8192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sda: 29.7 GiB, 31914983424 bytes, 62333952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x000dba0b

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1          20480  1044479  1024000  500M  c W95 FAT32 (LBA)
/dev/sda2       1228800 62333951 61105152 29.1G 83 Linux
root@polyhex[03:00:04]:~
```

- Mounting the U disk:

```
mount /dev/sda1 /mnt
```

```
root@polyhex[03:00:13]:~
$ mount /dev/sda1 /mnt
root@polyhex[03:01:36]:~
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        14G  2.7G   11G  21% /
devtmpfs        499M     0  499M   0% /dev
tmpfs           500M  484K  499M   1% /dev/shm
tmpfs           500M   14M  486M   3% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           500M     0  500M   0% /sys/fs/cgroup
/dev/mmcblk2p1  490M  7.8M  482M   2% /boot
tmpfs           100M   44K  100M   1% /run/user/0
/dev/sda1       500M   32M  469M   7% /mnt
/dev/sda2        29G  3.4G   25G  13% /media/root/79de8ff0-265b-451f-be52-87356c
5f68c0
root@polyhex[03:01:44]:~
```

2. Enter the U disk directory:

```
cd /mnt
```

```
root@polyhex[03:01:44]:~
$ cd /mnt
root@polyhex[03:02:15]:/mnt
$ ls
Image
System Volume Information
imx8mp-debix-4g-board.dtb
imx8mp-debix-core-HC050IG40029-D58V.C.dtb
```

4.4. Usage of 4G Module

Connect the 4G module, insert the SIM card, and connect the 4G antenna.

The 4G module is identified as /dev/ttyUSB2 under the system, and the following verification

is done in a scenario where the other network is disconnected.

- Query 4G module command

```
cd /etc/ppp/peers

./quectel-pppd.sh

ifconfig
```

```
root@polyhex:/etc/ppp/peers# ifconfig
eth0      Link encap:Ethernet  HWaddr 72:67:d7:f3:8d:7f
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:96 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7664 (7.6 KB)  TX bytes:7664 (7.6 KB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.212.83.249  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:240 (240.0 B)  TX bytes:182 (182.0 B)
```

- Apply ping command to check network

```
ping www.baidu.com
```

```
root@polyhex:/etc/ppp/peers# ping baidu.com
PING baidu.com (220.181.38.251) 56(84) bytes of data.
64 bytes from 220.181.38.251: icmp_seq=1 ttl=49 time=91.8 ms
64 bytes from 220.181.38.251: icmp_seq=2 ttl=49 time=79.6 ms
64 bytes from 220.181.38.251: icmp_seq=3 ttl=49 time=78.5 ms
64 bytes from 220.181.38.251: icmp_seq=4 ttl=49 time=88.1 ms
64 bytes from 220.181.38.251: icmp_seq=5 ttl=49 time=78.6 ms
64 bytes from 220.181.38.251: icmp_seq=6 ttl=49 time=78.8 ms
64 bytes from 220.181.38.251: icmp_seq=7 ttl=49 time=80.7 ms
^[^A64 bytes from 220.181.38.251: icmp_seq=8 ttl=49 time=78.7 ms
a64 bytes from 220.181.38.251: icmp_seq=9 ttl=49 time=80.8 ms
64 bytes from 220.181.38.251: icmp_seq=10 ttl=49 time=79.9 ms
64 bytes from 220.181.38.251: icmp_seq=11 ttl=49 time=76.3 ms
64 bytes from 220.181.38.251: icmp_seq=12 ttl=49 time=94.5 ms
```

4.5. Verification of RS485/RS232/RS422/CAN

NOTE

The default configuration is an RS485 interface. RS485, RS232, RS422 and CAN on the same socket, only one can be used at the same time.

Open the CuteCom tool, the serial port parameters are set as follows.

Table 11 Parameter set for cutecom tool

Parameter	Value
Baud rate	115200
Data bits	8
Stop bits	1
Parity	None
Flow Control	None

4.5.1. 4 x RS232

Connect Pin3 of COM1 to the sending end Pin5 of COM3, Pin5 to the receiving end Pin3 of COM3, and Pin9 to the ground terminal Pin9 of COM3. The wiring is shown in the following figure.

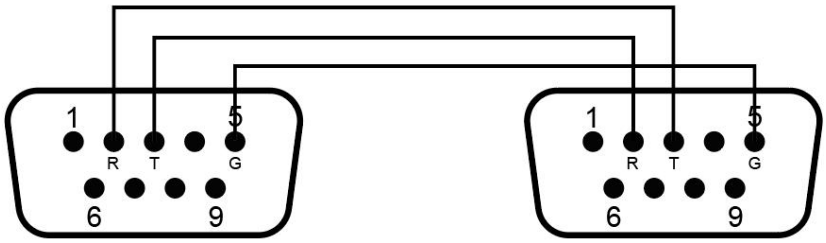


Figure 21

1. Run the CuteCom tool, set the Device to /dev/ttymxc0, set other parameters as shown in the table, click Open device.
2. Open another CuteCom window, set the Device to /dev/ttymxc2 and click Open device.
3. Send and receive data via CuteCom. Enter the test string in the CuteCom input box, press Enter to send, you can see that another CuteCom receiving box receives the same

message, indicating that the communication is successful, and the result is as follows:

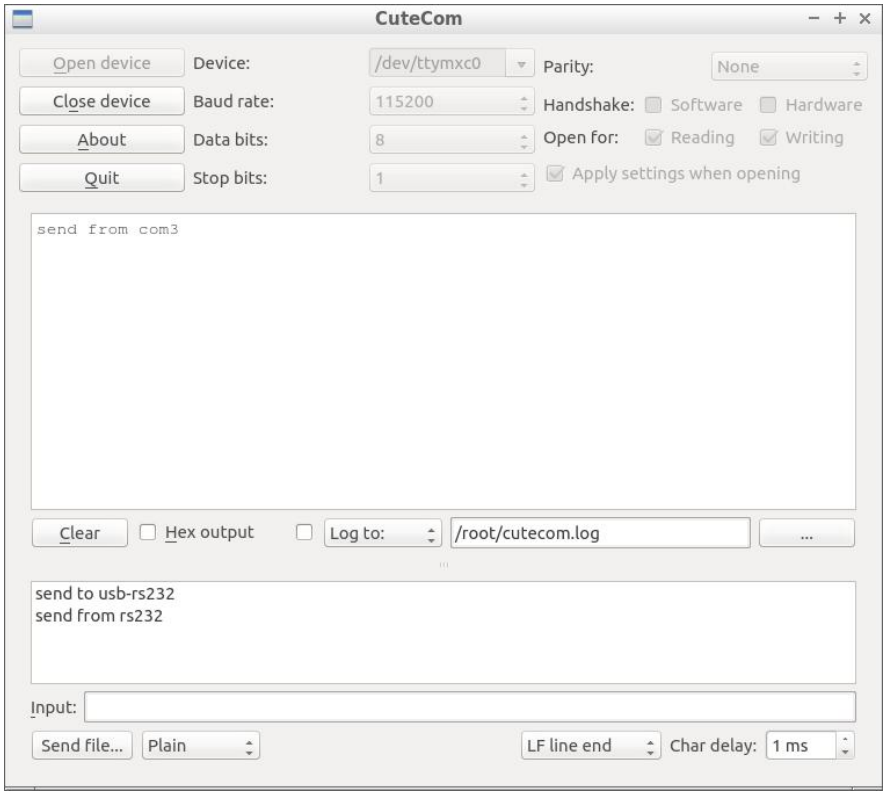


Figure 22

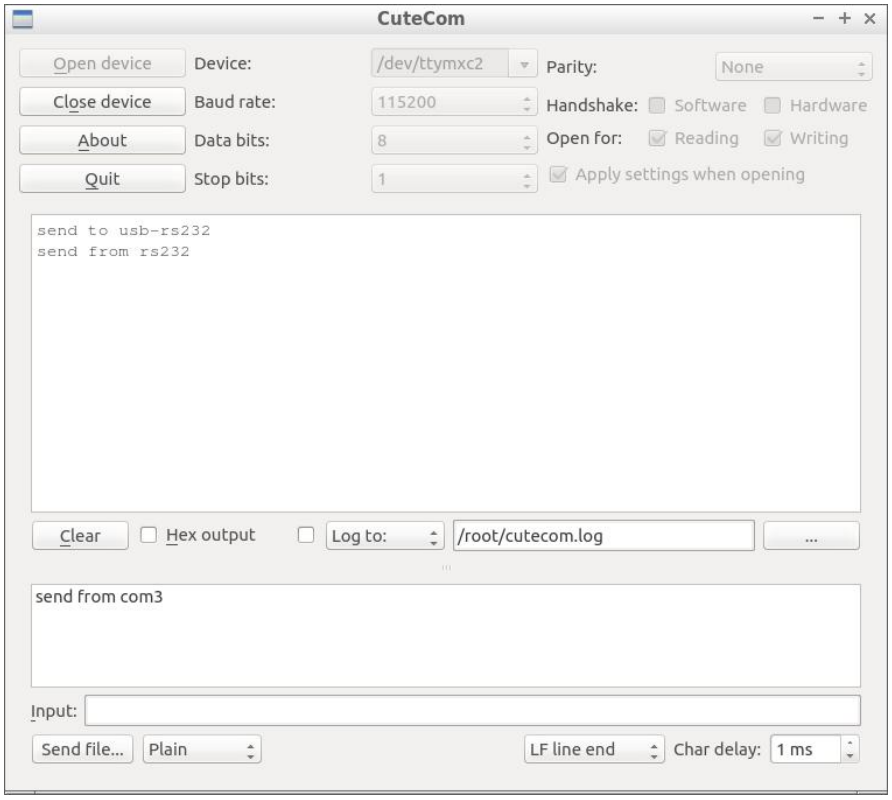


Figure 23

4.5.2. RS485/RS232/RS422/2 x CAN (RS485 by default)

➢ **RS485**

Connect Pin2 of RS485 to B of USB-RS485, Pin3 to A of USB-R485, USB-R485 is connected to USB interface on the PC. The wiring is shown in the following figure.

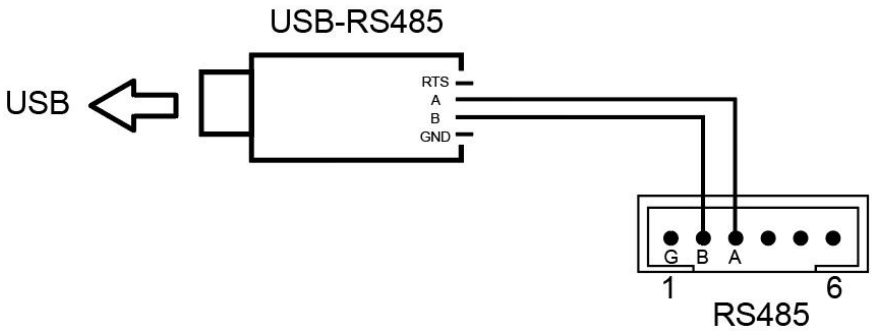


Figure 24

1. Run the CuteCom tool, set the Device to /dev/ttymxc4, click Open device.
2. Open MobaXterm tool on PC, click Sessions > New Session, click Serial in the Session settings window, set Serial port and other parameters in the Advanced Serial settings window, click OK. as follows:

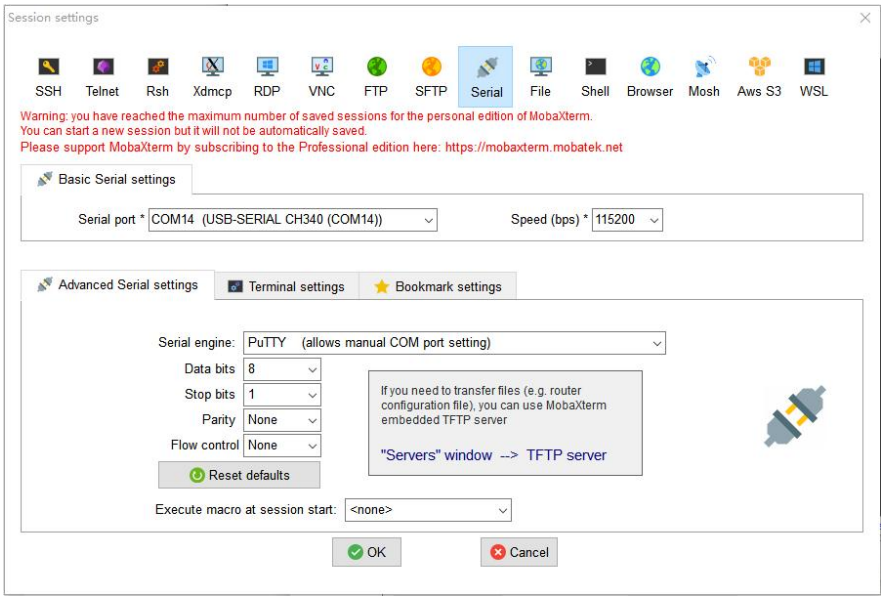


Figure 25

3. Enter the test string in the MobaXterm tool on PC, press Enter to send, you can receive the same message in the receive box of the CuteCom tool; enter the test string in the input box of the CuteCom tool, press Enter to send, you can receive the same message in the MobaXterm tool on the PC, indicating that the communication is successful, and the result is as follows:

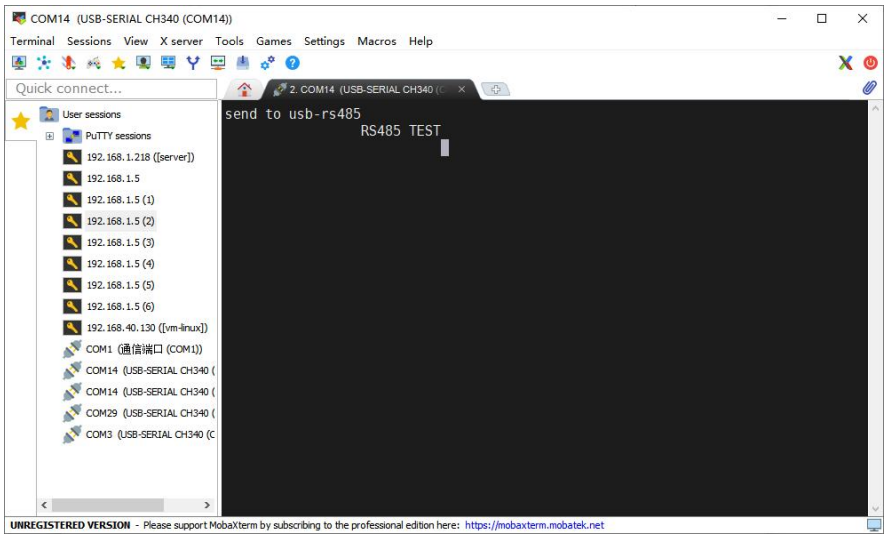


Figure 26

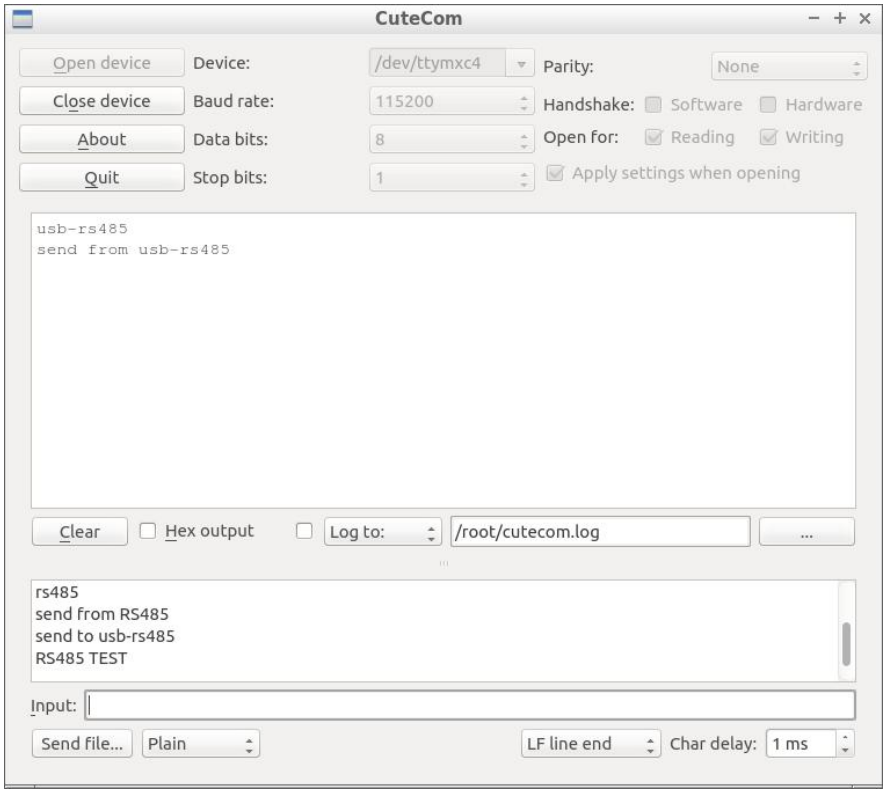


Figure 27

➢ **RS232**

Connect Pin3 of RS232 to the sending end of USB-RS232, Pin5 to the receiving end of USB-R232, and Pin1 to the ground terminal of USB-RS232, USB-R232 is connected to USB interface on the PC. The wiring is shown in the following figure.

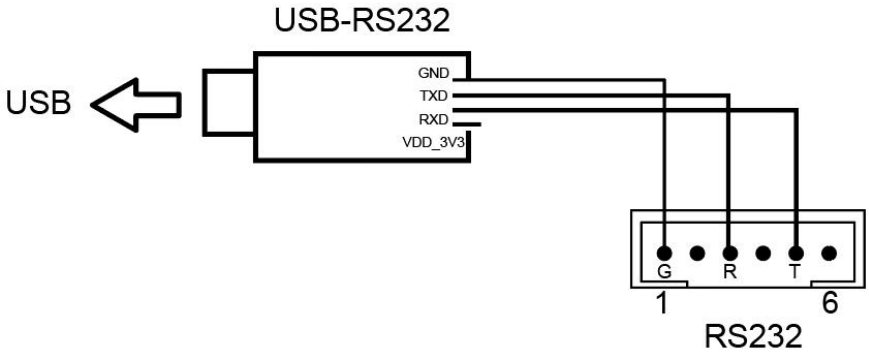


Figure 28

Repeat steps 1~3 of RS485 verification, the same information is sent and received, it indicates the communication is successful, and the result is as follows:

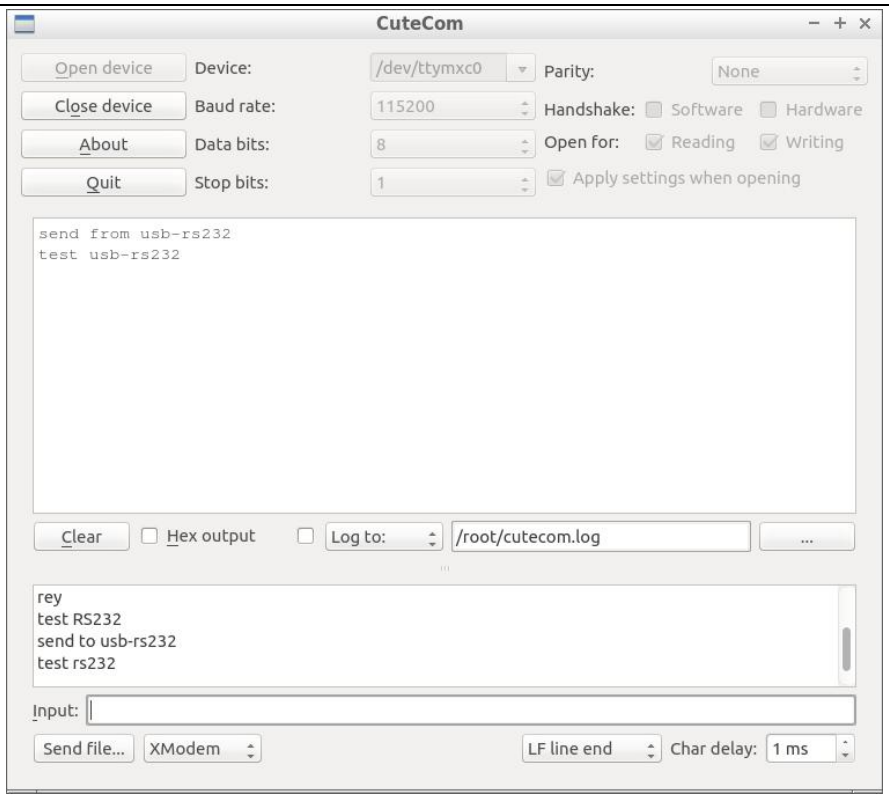


Figure 29

➢ **RS422**

Connect Pin7 of RS422 to the sending end T/R- of the adapter, Pin5 to the sending end T/R+ of the adapter, Pin1 to the receiving end R- of the adapter, Pin3 to the receiving end R+ of the adapter, Pin9 to the ground terminal of the adapter, and the adapter to USB-R232 connected to USB interface of the PC. The wiring is shown in the following figure.

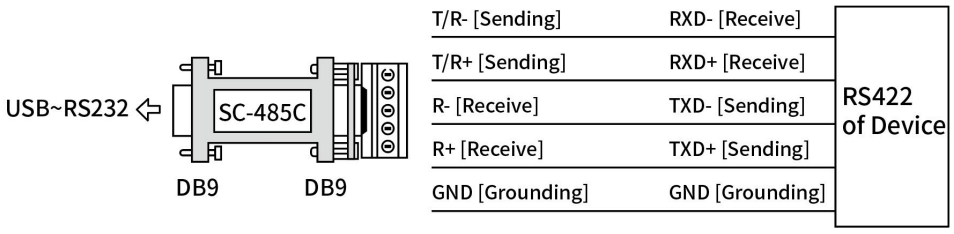


Figure 30

Repeat steps 1~3 of RS485 verification, the same information is sent and received, it indicates the communication is successful.

For example, if you type "send to iMX6Se from PC" in the MobaXterm tool, you can see the

same message in the receive box of the CuteCom tool; if you type "send to PC from iMX6Se" in the input box of the CuteCom tool, and the MobaXterm tool shows the same content, which proves that the communication is successful, and the result is as follows:

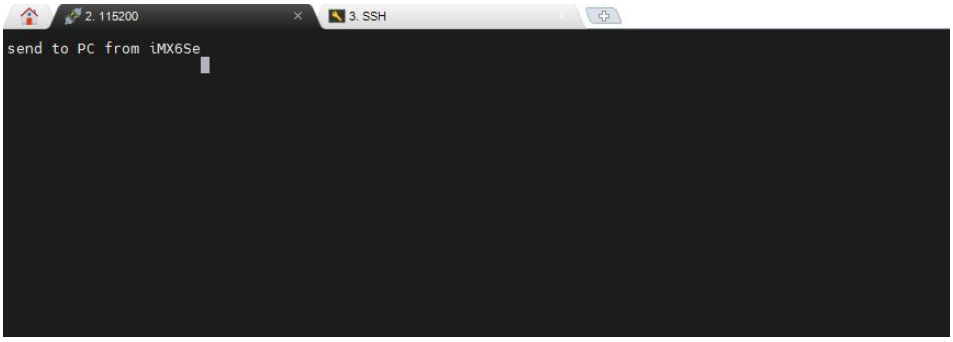


Figure 31

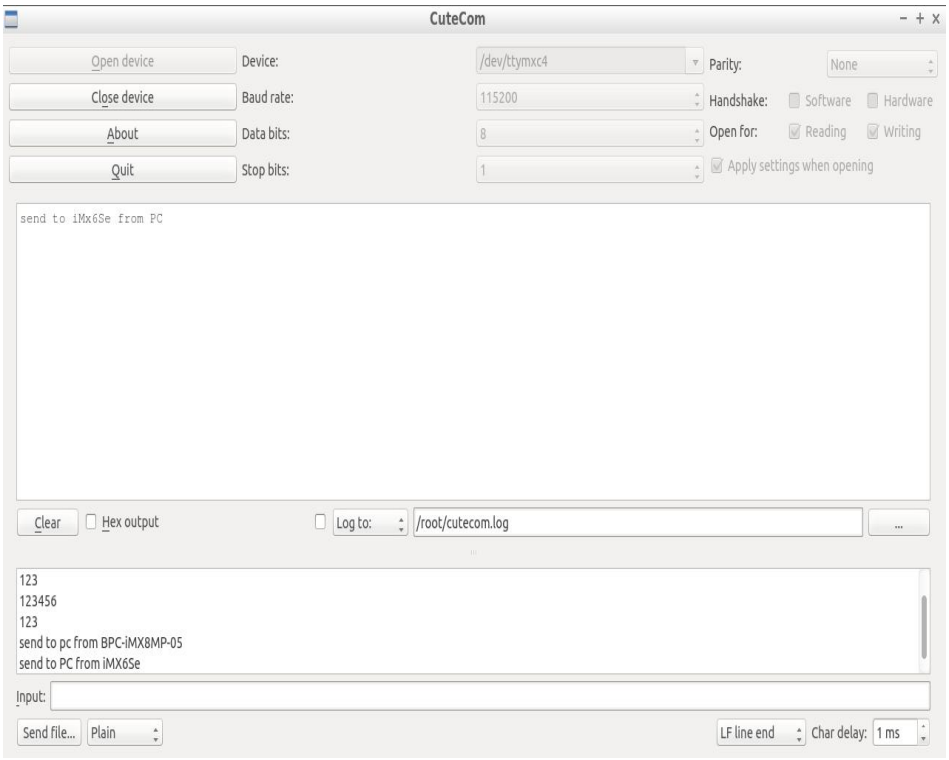


Figure 32

➢ CAN

Connect Pin2 to Pin5, and Pin3 to Pin6 of the CAN bus interface (that is, H to H, and L to L of two CANs). The wiring is shown in the following figure.

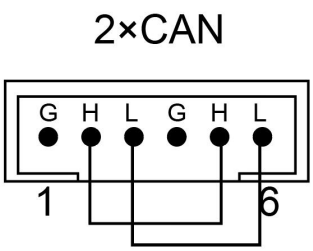


Figure 33

- CAN1 sends data, CAN0 receives data

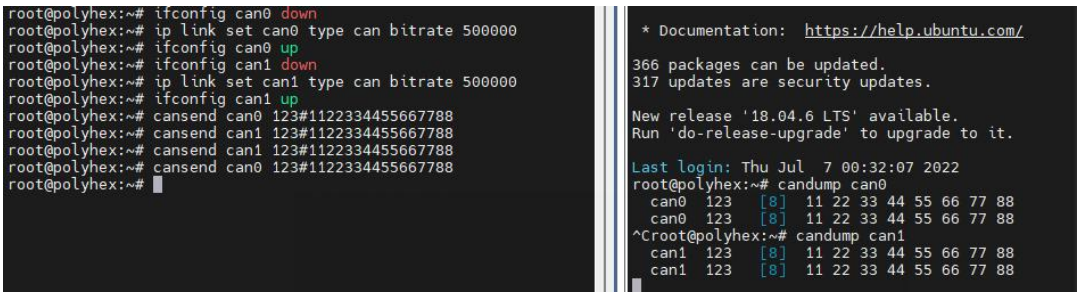
Open a Terminal, configure CAN0 to receive, and CAN1 to send.

```
ifconfig can0 down
ip link set can0 type can bitrate 500000
ifconfig can0 up
ifconfig can1 down
ip link set can1 type can bitrate 500000
ifconfig can1 up
candump can0                          #(can0 backend receive data)
cansend can1 123#1122334455667788     #(can1 send data)
```

- CAN1 receives data, CAN0 sends data

In the Terminal, switch can1 to receive and can0 to send.

```
candump can1                          #(can1 backend receive data)
cansend can0 123#1122334455667788     #(can0 send data)
```



4.6. Verification of GPIO

GPIO output:

1. Enter the gpio-1 control directory

```
cd /sys/class/leds/gpio-1
```

2. GPIO output low

```
echo 0 > brightness
```

3. GPIO output high

```
echo 1 > brightness
```

Other GPIOs are verified in the same way, and the GPIO ports are as follows:

```
root@polyhex:/# cd /sys/class/leds/
root@polyhex:/sys/class/leds# ls
gpio-1   gpio-3  gpio-D17  gpio-D30       gpio-spi-ss0      input2::scrolllock
gpio-16  gpio-4  gpio-D18  gpio-spi-miso  gpio-spi-ss1      mmc0::
gpio-19  gpio-5  gpio-D21  gpio-spi-mosi  input2::capslock  mmc2::
gpio-2   gpio-6  gpio-D22  gpio-spi-sck   input2::numlock   mmc3::
```

4.7. Verification of RTC

1. Check the current system time:

```
date
```

```
root@polyhex[07:17:03]:~
$ date
Fri May  5 07:17:07 UTC 2023
```

2. Check the current RTC time:

```
hwclock
```

```
root@polyhex[07:17:07]:~
$ sudo hwclock
Fri May  5 07:17:24 2023  .363071 seconds
```

3. Modify the current system time:

```
date -s “2023-5-5 15:18:00”
```

```
root@polyhex[07:17:24]:~
$  sudo date -s "2023-5-5 15:18:00"
Fri May  5 15:18:00 UTC 2023
```

4. Write system time to RTC:

```
hwclock -w

hwclock    #Check the RTC time
```

```
root@polyhex[15:18:04]:~
$ sudo hwclock -w
root@polyhex[15:18:24]:~
$ hwclock
Fri May  5 15:18:30 2023  .593077 seconds
```