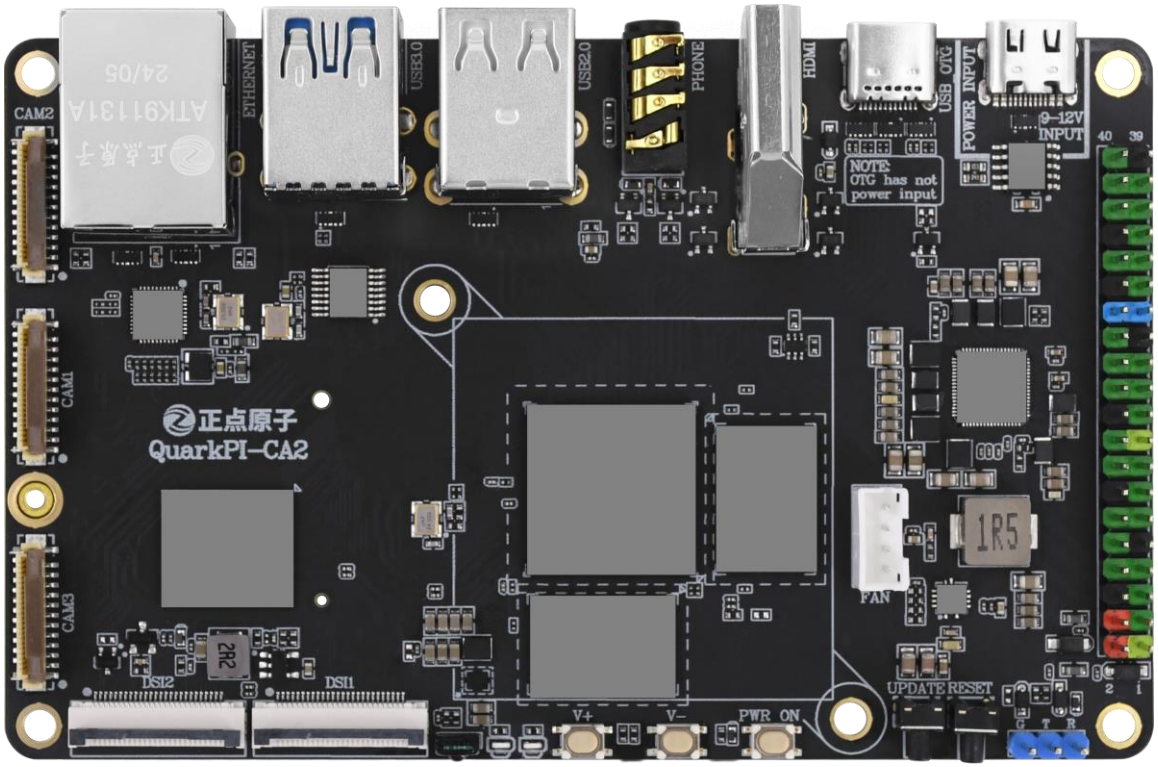
QuarkPi-CA2 Debian

Quick test manual

V1.0



ALIENTEK

1. Shopping:

TMALL: https://zhengdianyuanzi.tmall.com

TAOBAO: https://openedv.taobao.com

2. Download

Address: http://www.openedv.com/docs/index.html

3. FAE

Website : www.alientek.com

Forum : http://www.openedv.com/forum.php

Videos : www.yuanzige.com

Fax : +86 - 20 - 36773971

Phone : +86 - 20 - 38271790

Disclaimer

The product specifications and instructions mentioned in this document are for reference only and subject to update without prior notice; Unless otherwise agreed, this document is intended as a product guide only, and none of the representations made herein constitutes a warranty of any kind. The copyright of this document belongs to Guangzhou Xingyi Electronic Technology Co., LTD. Without the written permission of the company, any unit or individual shall not be used for profit-making purposes in any way of dissemination.

In order to get the latest version of product information, please regularly visit the download center or contact the customer service of Taobao ALIENTEK flagship store. Thank you for your tolerance and support.

Revision History:

Version	Version Update Notes	Responsible person	Proofreading	Date
V1.0	release officially	ALIENTEK	ALIENTEK	

Catalogue

Brief 1
Chapter 1. Preparations Before Using the Card Computer 2
1.1 Pre-requisites Before Powering On 3
1.2 Installation of Serial Port Software 3
1.2.1 Installation of CH340 Serial Port Driver 3
1.2.2 Installation of MobaXterm software 5
1.2.3 Use of serial ports in MobaXterm software 10
1.2.4 SSH Usage of MobaXterm Software 13
1.3 Installing SFTP Client on Windows 16
1.3.1 How to Use FileZilla 20
Chapter 2. Testing of Card Computer Peripherals 24
2.1 LED Testing 25
2.2 evtest Test 25
2.2.1 Key Test 25
2.2.2 Infrared Module Test 27
2.3 Screen test 28
2.3.1 HDMI display output 28
2.3.2 DP display output 28
2.3.3 MIPI Display Output 28
2.4 Audio Testing 30
2.4.1 MIPI camera 30
2.4.2 USB Camera 31
2.5 USB & TF Card & EMMC Testing 32
2.5.1 USB HOST Testing 32
2.5.2 USB 2.0 Test 32
2.5.3 USB3.0 Test 33
2.5.4 Slave Test 33
2.6 Network Testing 33
2.6.1 Serial Port Terminal Command Line Mode 34
2.7 M.2 PCI-E WIFI Test 36
2.8 Bluetooth Test 40
2.9 CPU & GPU Temperature 43
2.10 CPU Frequency 43

Brief

Note: This document will be tested using the factory-installed system of the ALIENTEK QuarkPi-CA2 card computer!!!

Chapter 1. Preparations Before Using the Card Computer

Here, we will introduce the matters that need to be noted when using the card computer. Please read carefully. Because if you don't pay attention to these matters, it may lead to the failure of your experiment or damage to the card computer, etc. Therefore, you must pay attention.

1.1 Pre-requisites Before Powering On

- Firstly, check if the power supply is properly plugged in. Once it is plugged in, verify that the power switch is turned on and the power supply is functioning normally.
- Even after prolonged use, it is necessary to observe whether any foreign objects have fallen onto the development board to prevent short circuits. Pay attention to waterproofing, moisture protection, and dust prevention.

1.2 Installation of Serial Port Software

1.2.1 Installation of CH340 Serial Port Driver

The ALIENTEK Card computer uses the domestic chip CH340 to achieve USB to serial port conversion, and can directly connect the debugging serial port of the board card to the computer via a USB cable without the need for a USB to serial port adapter cable. This is convenient for users. However, the CH340 driver needs to be installed under Windows in order to recognize the debugging serial port of the board card.

The board card data package has already provided the installation file of the CH340 driver for users. The path is: Development Board CD-ROM A Disk - Basic Materials -> 03_softwares -> CH343SER.EXE. Double-click the executable file CH343SER.EXE to install the CH340 driver (do not connect the debugging serial port of the board card before installation as shown in Figures 1.2.1 to 1.2.2):

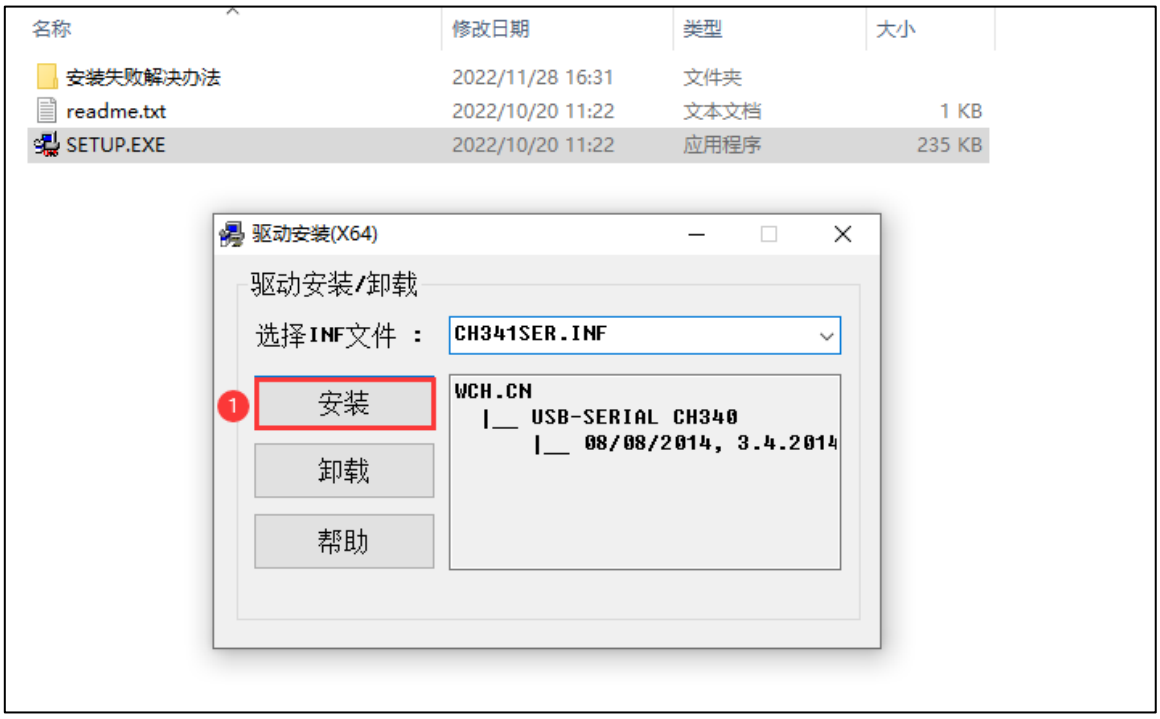


Figure 1.2-1 Install CH340 driver (1)

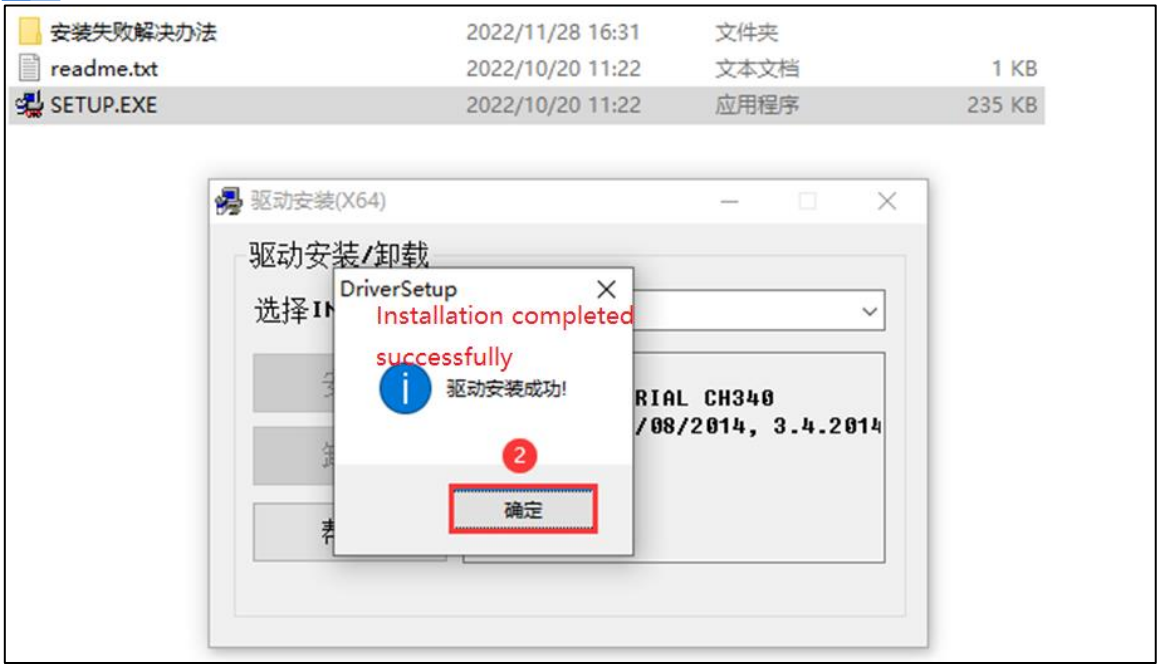


Figure 1.2-2 Install CH340 driver (2)

After the installation is completed, connect the debugging serial port of the board card to the computer via a USB cable. The connection method is shown in Figure 1.2.1.3:

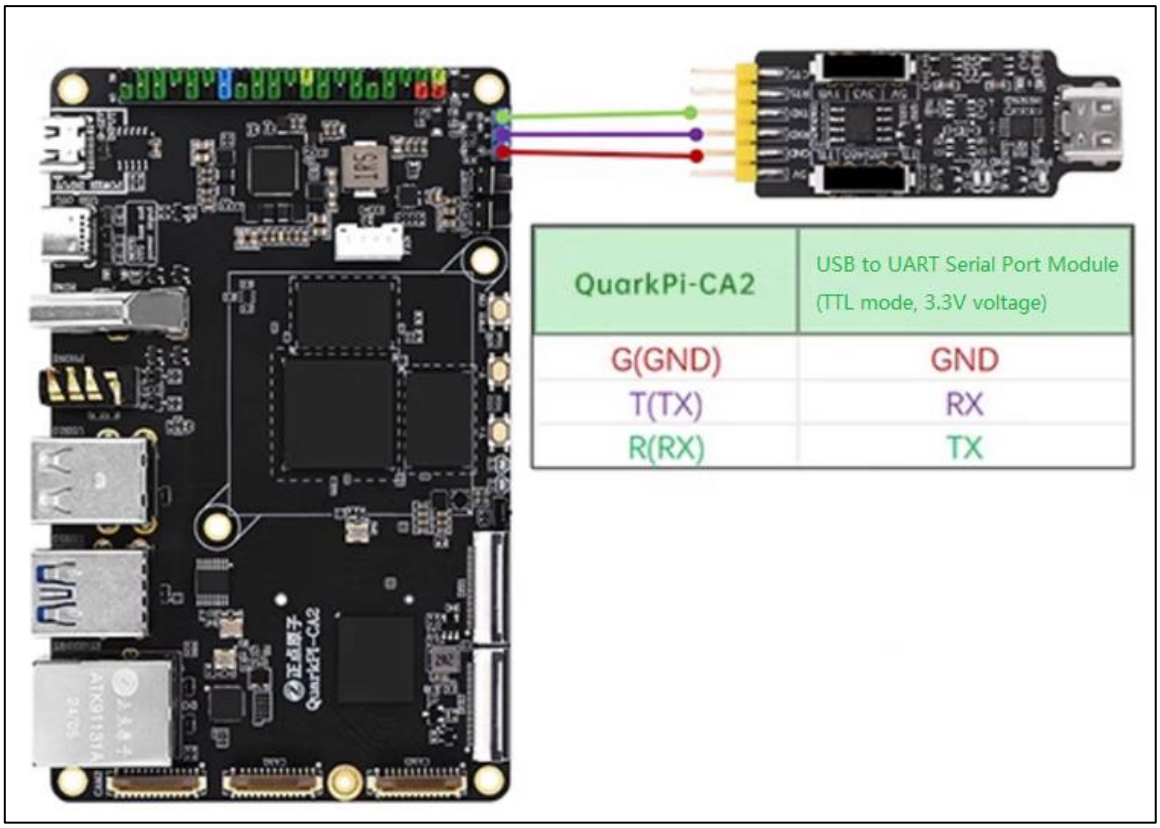


Figure 1.2-3 Board card debugging: Connect the serial port to the computer.

After the connection is successful, at this point, the computer will detect a USB device and open the Windows Device Manager:

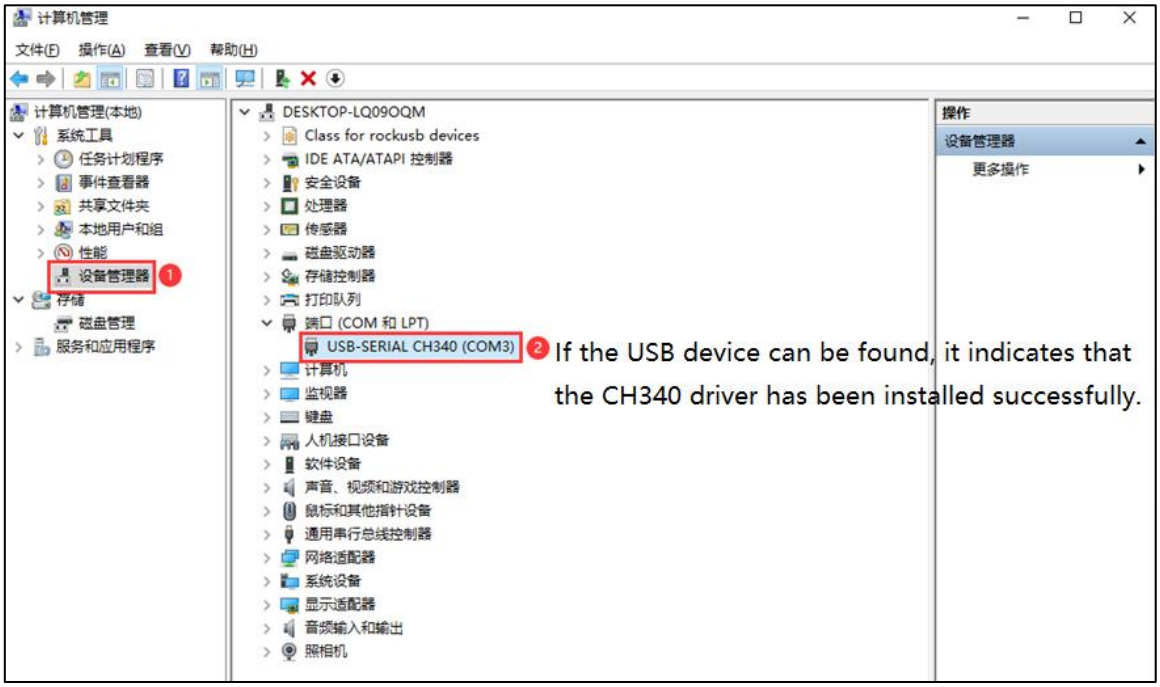


Figure 1.2-4 Windows Device Manager

If a device named "USB-SERIAL CH340" can be found under "Device Manager Ports (COM & LPT)", it indicates that the CH340 driver has been installed successfully! If the device named "USB-SERIAL CH340" cannot be found, please have the user check if the hardware connection is incorrect! You can unplug the USB cable, reconnect it, or try connecting to another USB port on the computer.

1.2.2 Installation of MobaXterm software

MobaXterm is a versatile remote terminal software with powerful functions and it is free (there is also a paid version). It supports numerous remote connection features such as SSH, Telnet, Rsh, Xdmc, RDP, VNC, FTP, SFTP, serial port (Serial COM), etc. MobaXterm offers a user-friendly interface and is extremely powerful, so we recommend users to use this terminal software.

- Download of MobaXterm software

The MobaXterm software installation package has been provided to users in the board card data package. The path is: Development board optical disc A disk - Basic materials -> 03_softwares -> MobaXterm_Installer_v24.1.zip; Users can also download it themselves through the link address: https://mobaxterm.mobatek.net/download.html

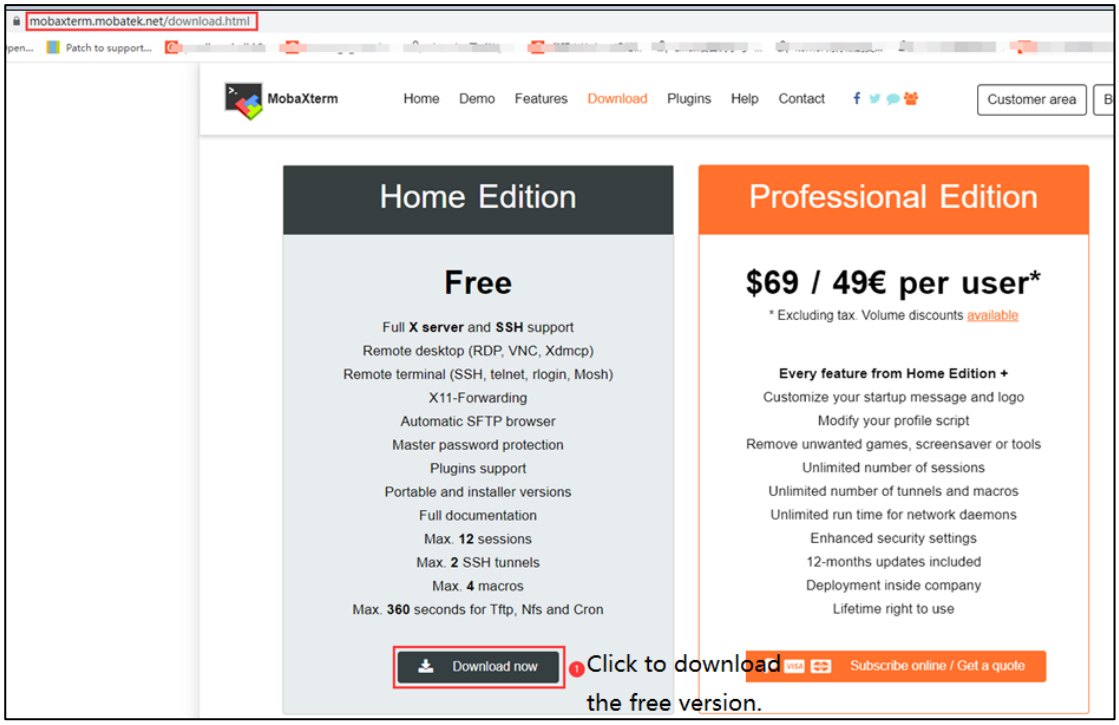


Figure 1.2-5 Download the MobaXterm software (1)

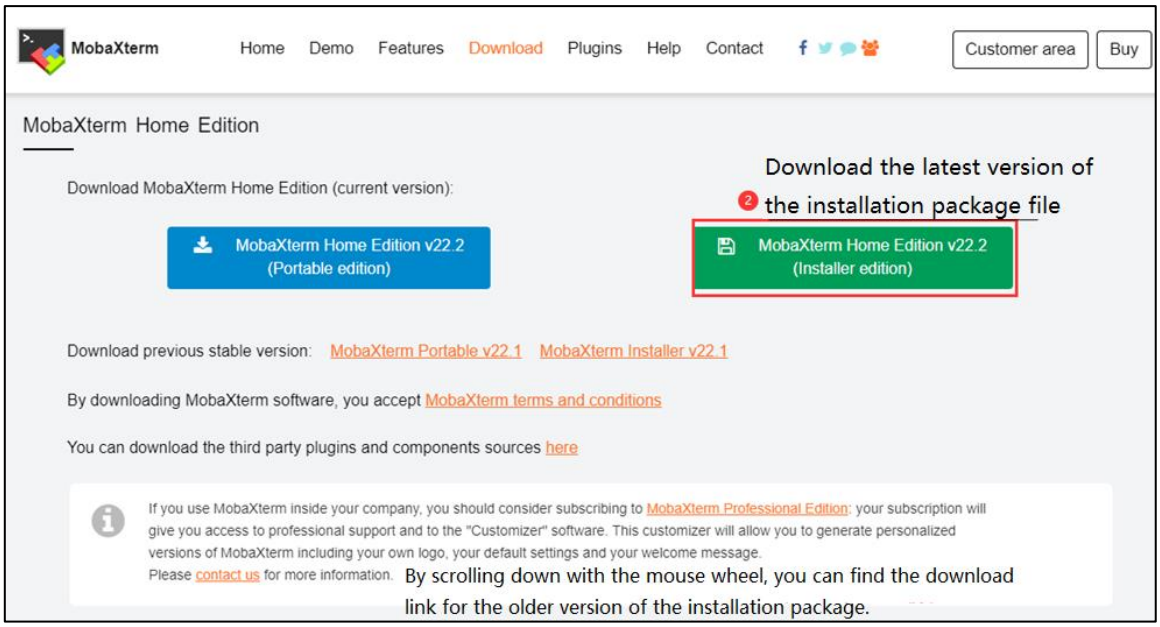


Figure 1.2-6 Download the MobaXterm software (2)

After the download is complete, a compressed file named "MobaXterm_Installer_vxxx.zip" (where xxx represents the version number) will be obtained. Currently, the latest version is 22.2. The version of the installation package provided for users in the data package is 12.3. You can use the new version or the old one, it doesn't matter. Here, we take the 12.3 version as an example.

- Installation of MobaXterm software

Extract the MobaXterm_Installer_v12.3.zip compressed file. After extraction, it will look like this:

Figure 1.2-7 The files after extracting MobaXterm_Installer_v12.3.zip

Then double-click to run the MobaXterm_Installer_v12.3.msi file and follow the steps shown in Figures 1.2.2.4 to 1.2.2.8 to install the MobaXterm software:

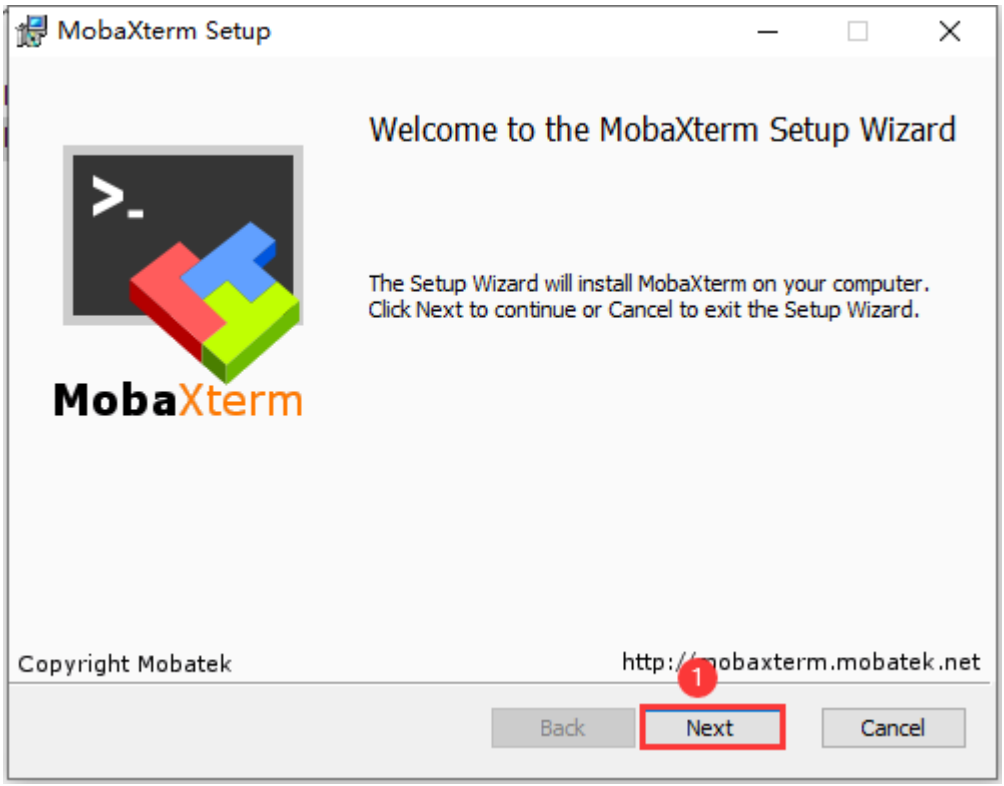


Figure 1.2-8 Install the MobaXterm software (1)

Figure 1.2-9 Install the MobaXterm software (2)

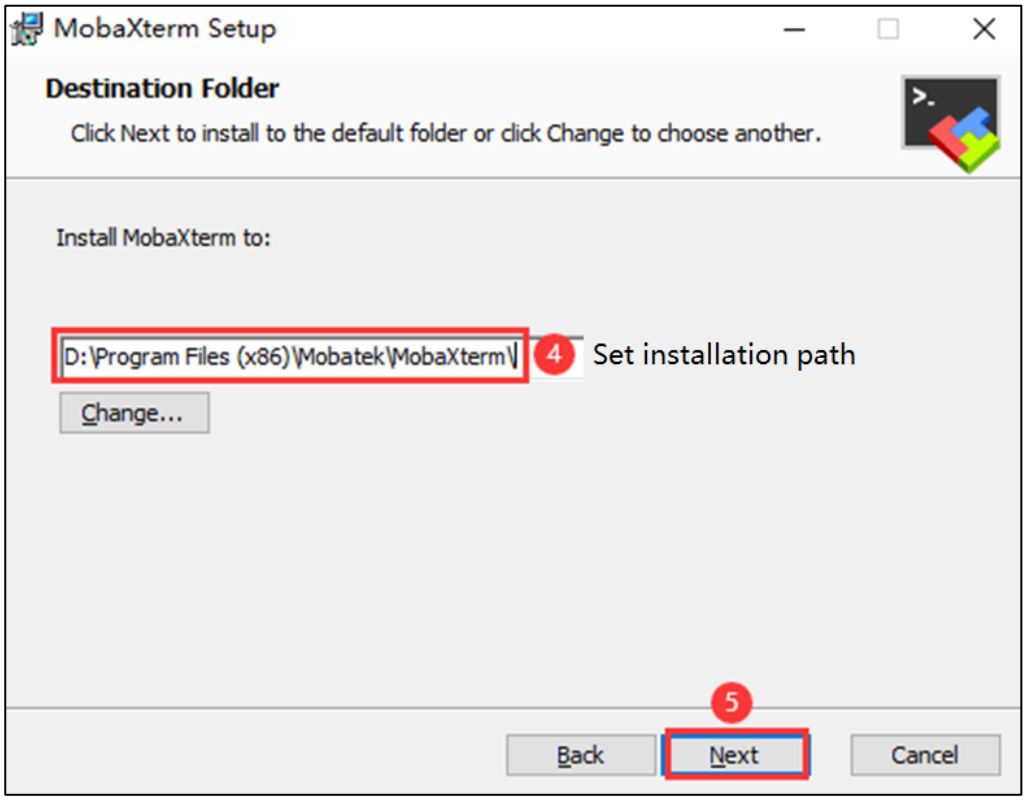


Figure 1.2-10 Install the MobaXterm software (3)

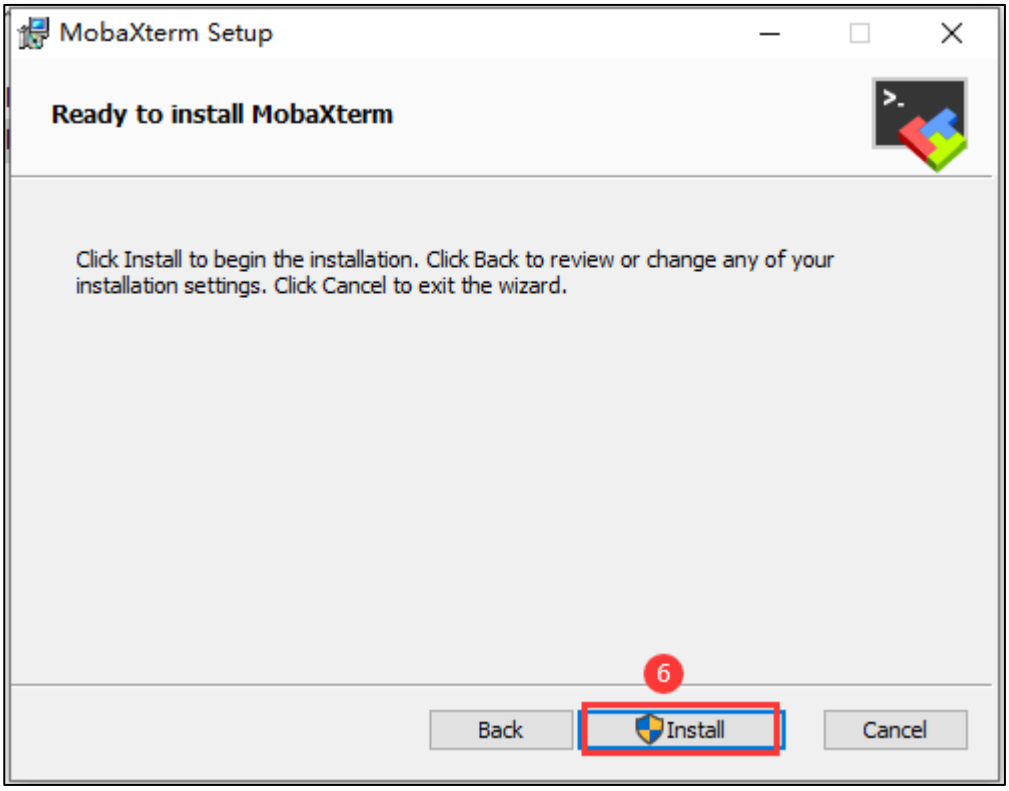


Figure 1.2-11 Install the MobaXterm software (4)

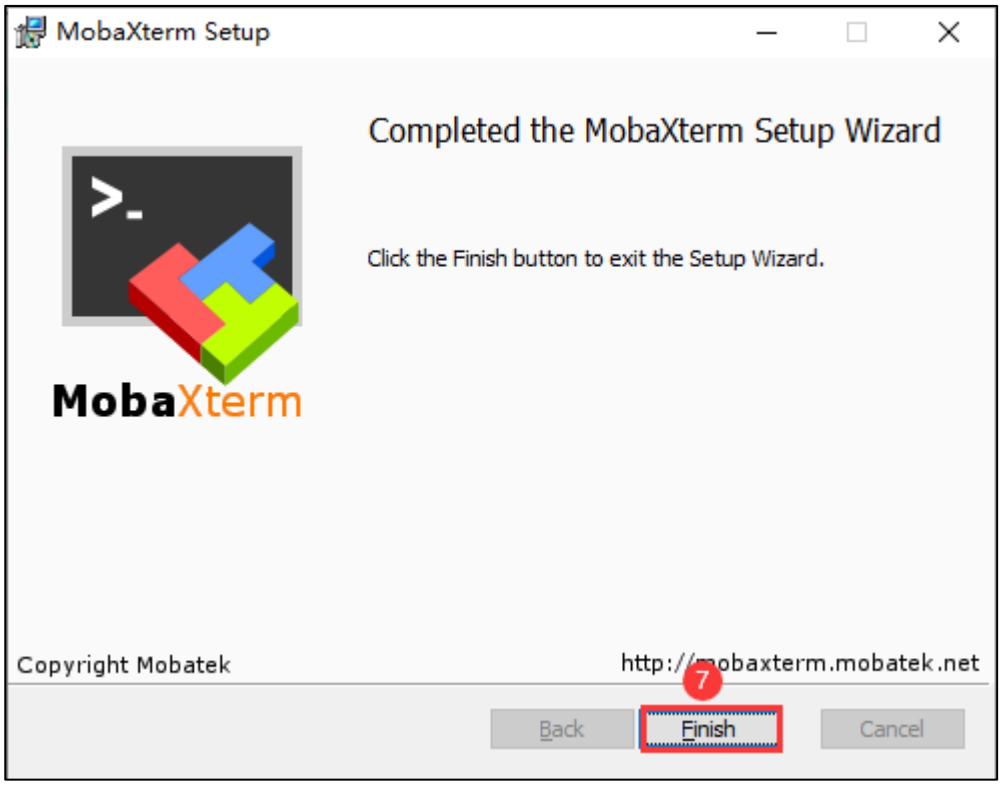


Figure 1.2-12 Install the MobaXterm software (5)

At this point, the software installation is complete. The desktop will automatically generate a shortcut icon for the MobaXterm software.

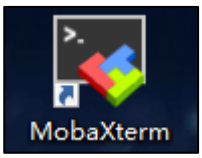


Figure 1.2-13 MobaXterm software desktop icon

1.2.3 Use of serial ports in MobaXterm software

Double-click the MobaXterm desktop icon to open the software, as shown in Figure 1.2.14:

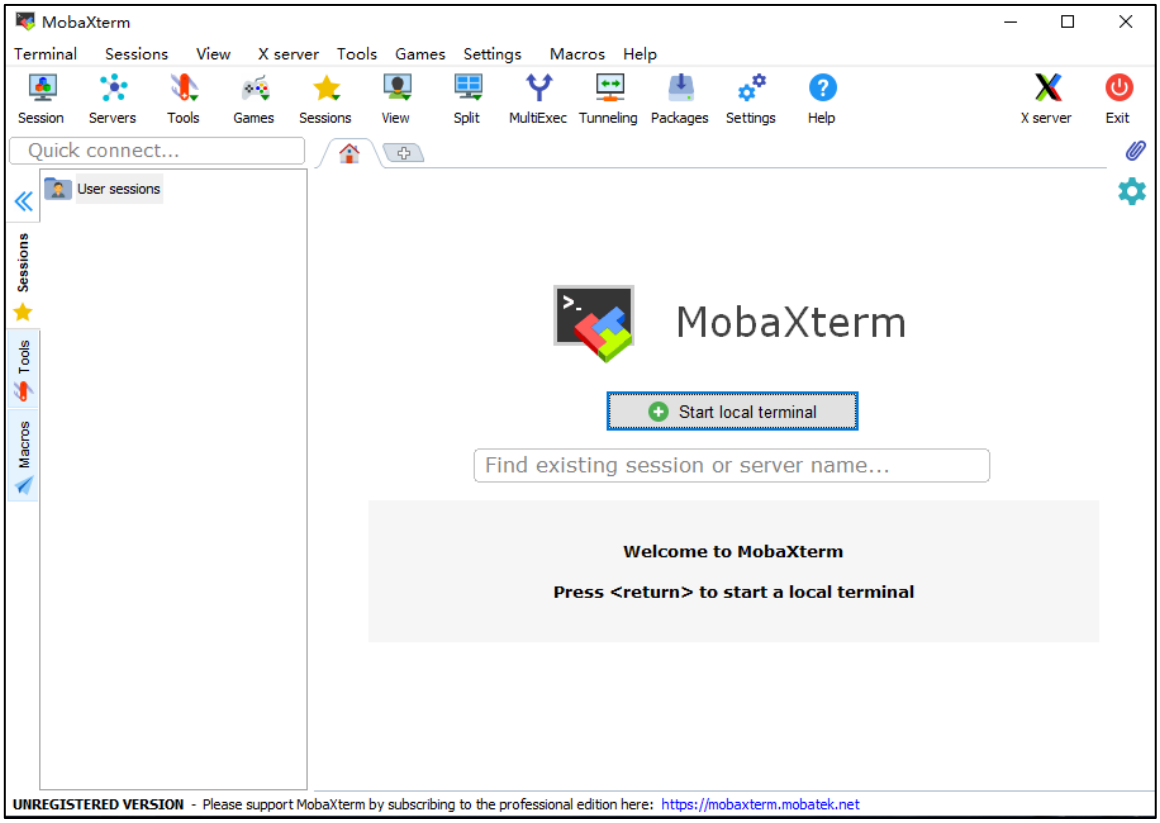


Figure 1.2-14 The main interface of MobaXterm software

Next, I will introduce how to establish Serial (serial port) connection and ssh remote connection.

1. Serial port connection

There is a debugging serial port on the card computer, which can be directly connected to the computer via a USB cable. The serial port is one of the most common communication interfaces for embedded devices. It not only enables data transmission between the computer and the embedded device, but also allows the computer to control the embedded device. In the embedded development process, it is usually used as a debugging interface (serial port debugging) to debug the embedded device.

During the development process, we interact with the card computer through information exchange via serial port. Of course, it can also be done via the network. The network needs to know the IP address of the card computer first. Therefore, the serial port is very important at this time.

1. Connect one end of the serial port module to the computer, and the other end to the pins G (GND), R (RX), and T (TX) of the card computer. The connection diagram is as follows:

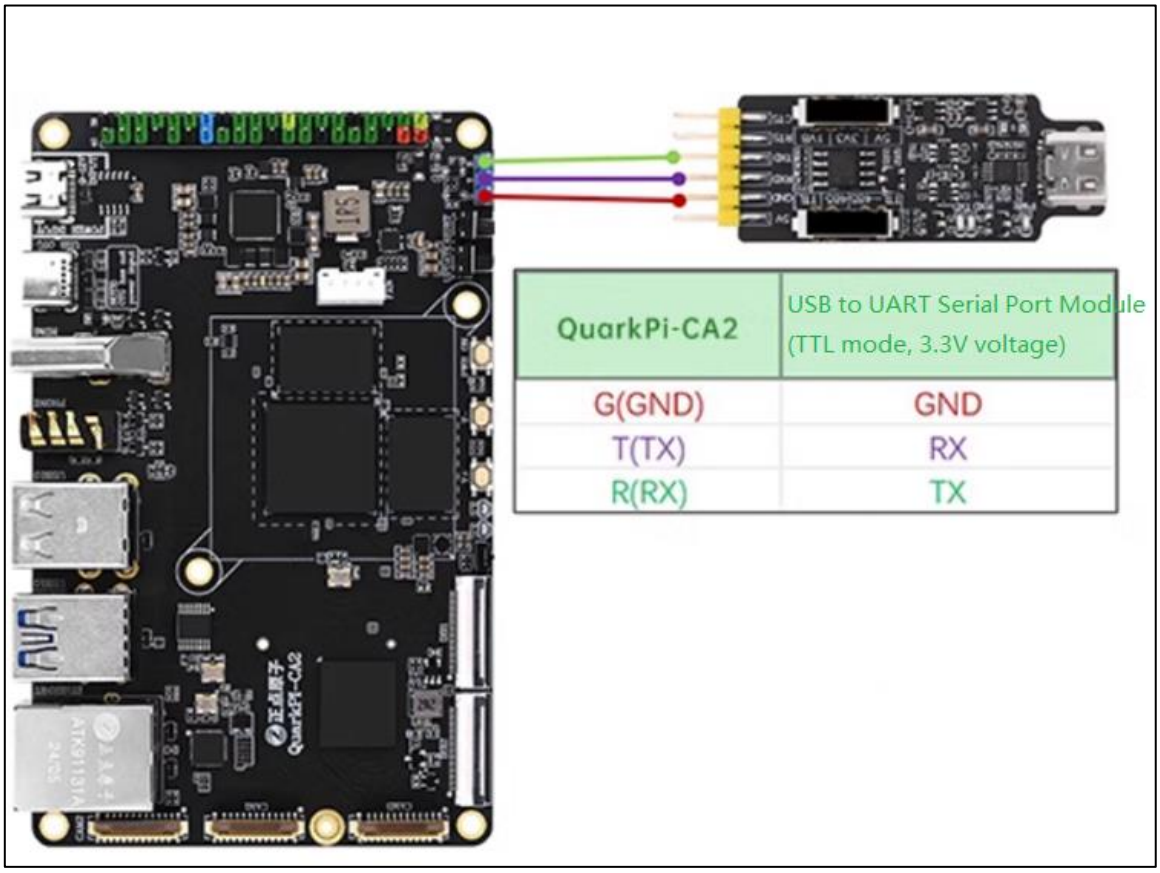


Figure 1.2-15 Connection diagram

Serial module mode selection: Select TTL mode, voltage selection: 3.3V mode.

Follow the operation steps shown in Figures 1.2.16 to 1.2.19 to establish a Serial connection (before establishing the connection, the debugging serial port of the card computer needs to be connected to the computer via a USB cable, and the CH340 driver has been installed):

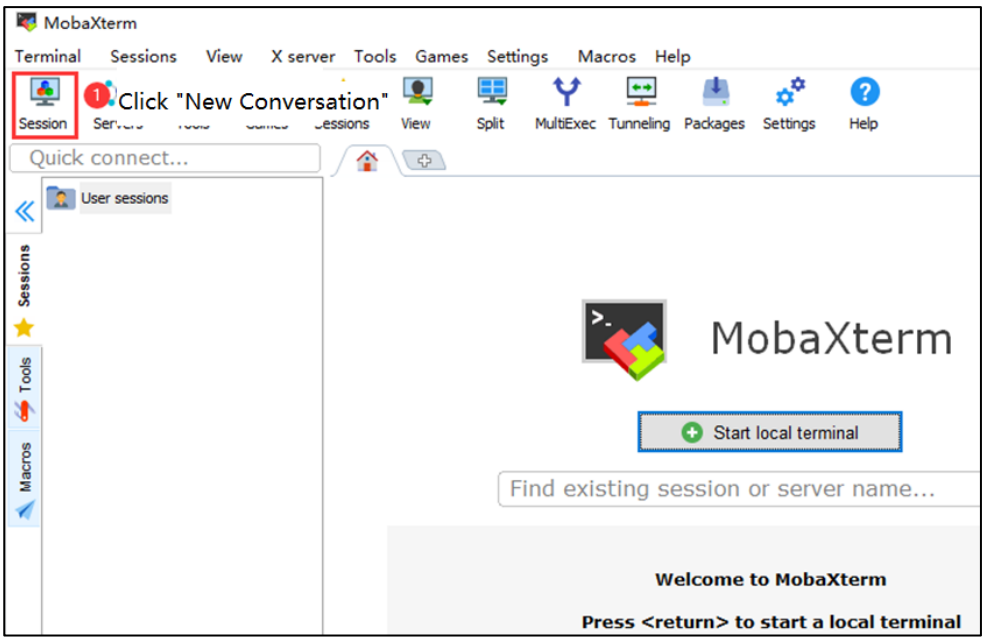


Figure 1.2-16 Establish serial port connection (1)

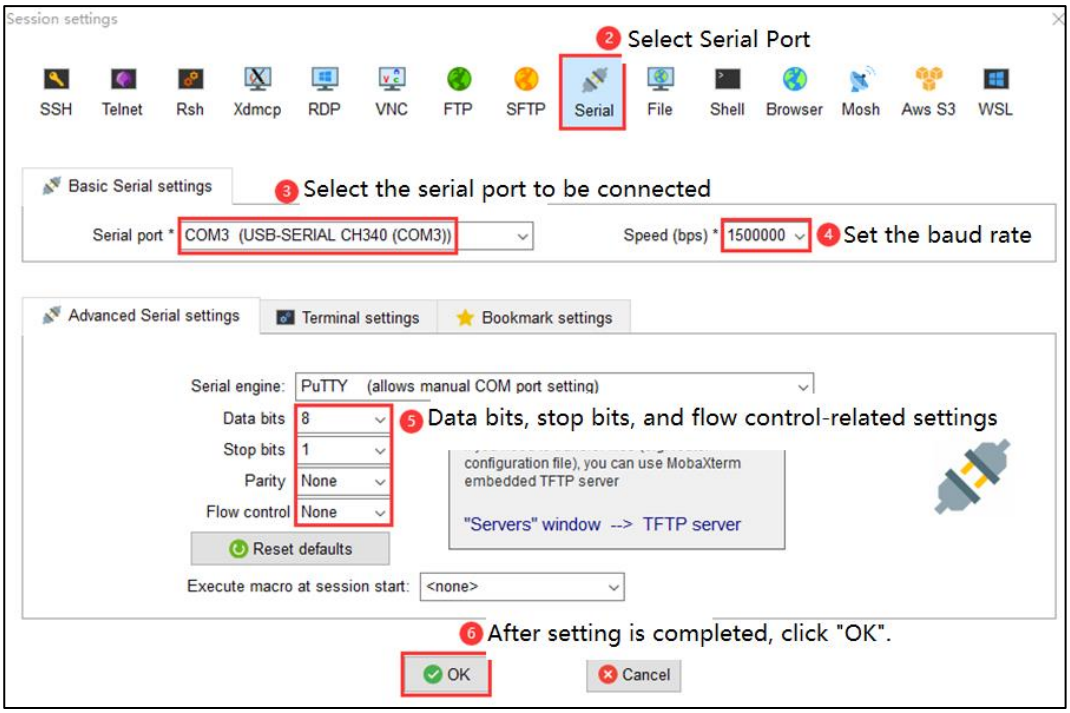


Figure 1.2-17 Establish serial port connection (2)

First, select the serial port that needs to be connected. Make sure that the debugging serial port of the card computer is already connected to the computer via a USB cable, and that the CH340 driver has been successfully installed ("Device Manager Ports (COM and LPT)"), under which you can find a device named "USB-SERIAL CH340". Then, the MobaXterm software can recognize the debugging serial port of the card computer. You can find the corresponding serial port of the card computer (USB-SERIAL CH340) in the "Serial port" drop-down list, and then select it!

Next, set the serial port communication baud rate. Set it according to the actual situation. For the debugging serial port of the QuarkPi-CA2 platform, its default baud rate is 1500000 (1.5M). Then, set the data bits, stop bits, and flow control, etc. After setting, click the "OK" button.

The serial connection has been successfully established, as shown in Figure 1.2.18:

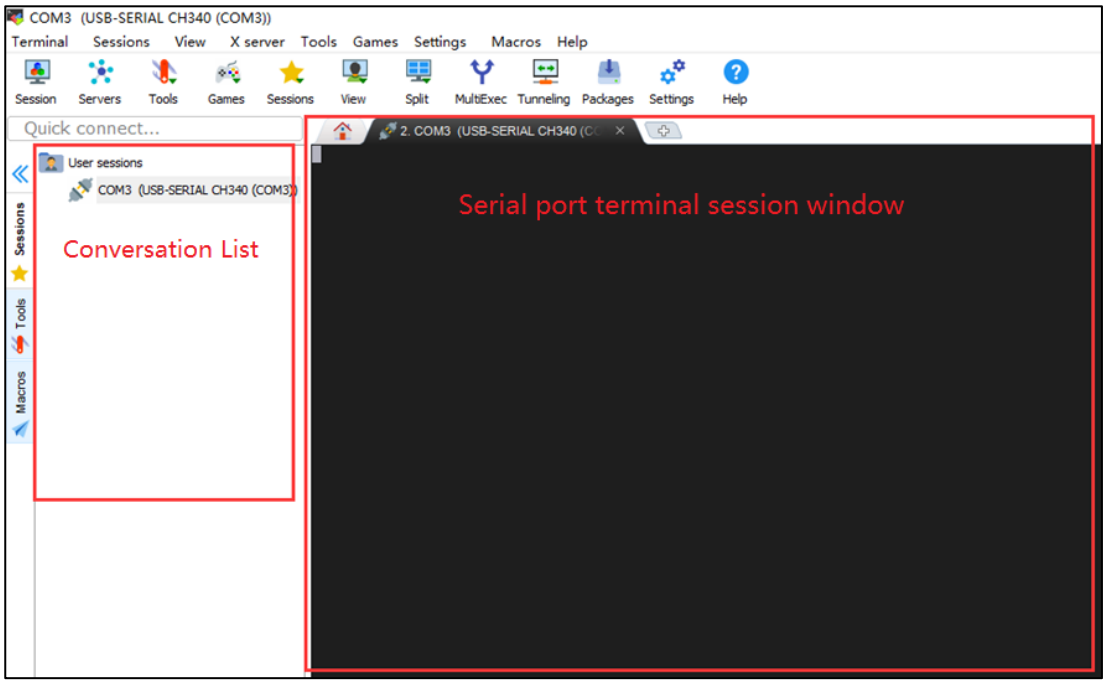


Figure 1.2-18 Establish serial port connection (3)

Turn on the power switch, power on, and the pre-installed system is the Debian system. The information output via the serial port is as follows.

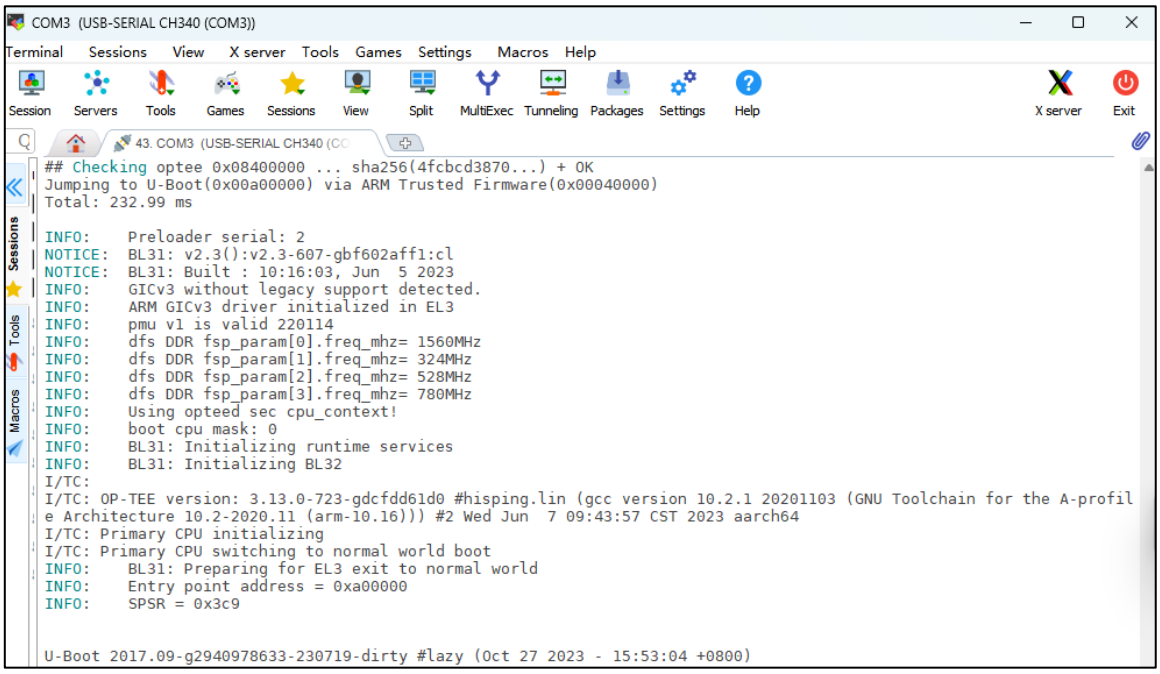


Figure 1.2-19 Serial port logging of logs

1.2.4 SSH Usage of MobaXterm Software

Next, we will introduce how to establish an SSH remote connection. Through SSH, it is possible to remotely log into the Debian system of the card computer. Before establishing the SSH remote connection, it is necessary to first input "ifconfig" through the debugging serial port of the card

computer to check the IP address of the card computer. By default, the card computer does not have a static address and needs to obtain the IP address automatically through the connection to the router. Follow the operation steps shown in Figures 1.2.20 to 1.2.24 to establish the SSH remote connection and achieve the remote login operation:

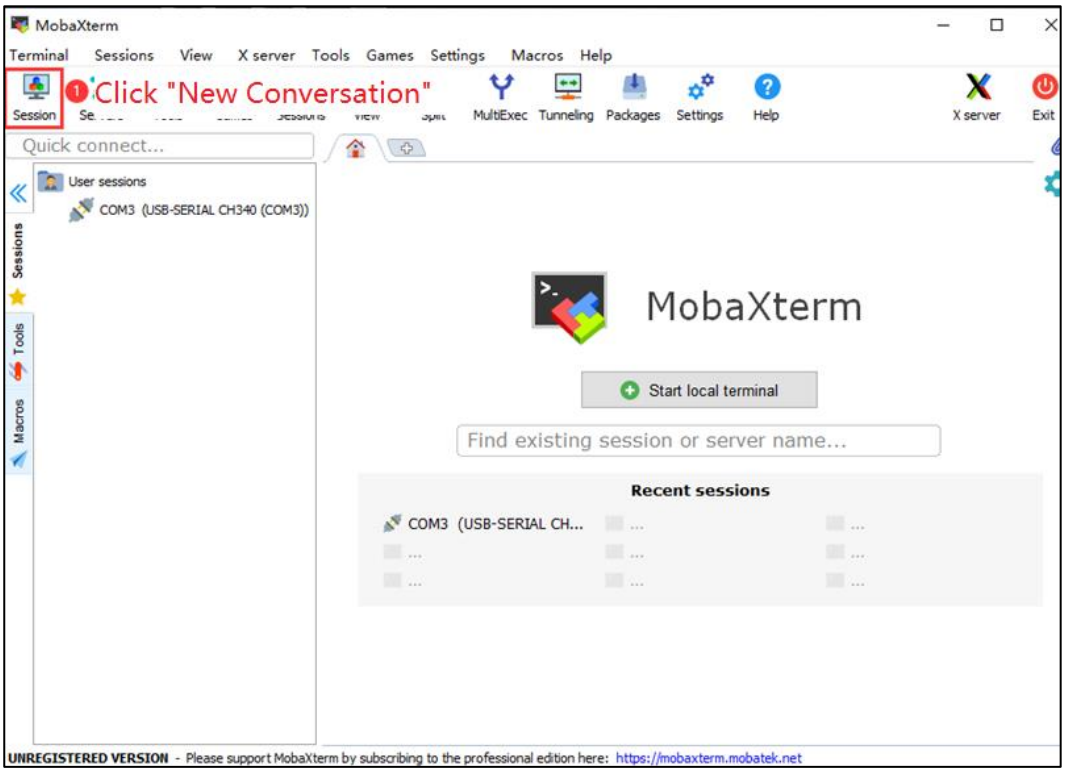


Figure 1.2-20 Establish SSH remote connection (1)

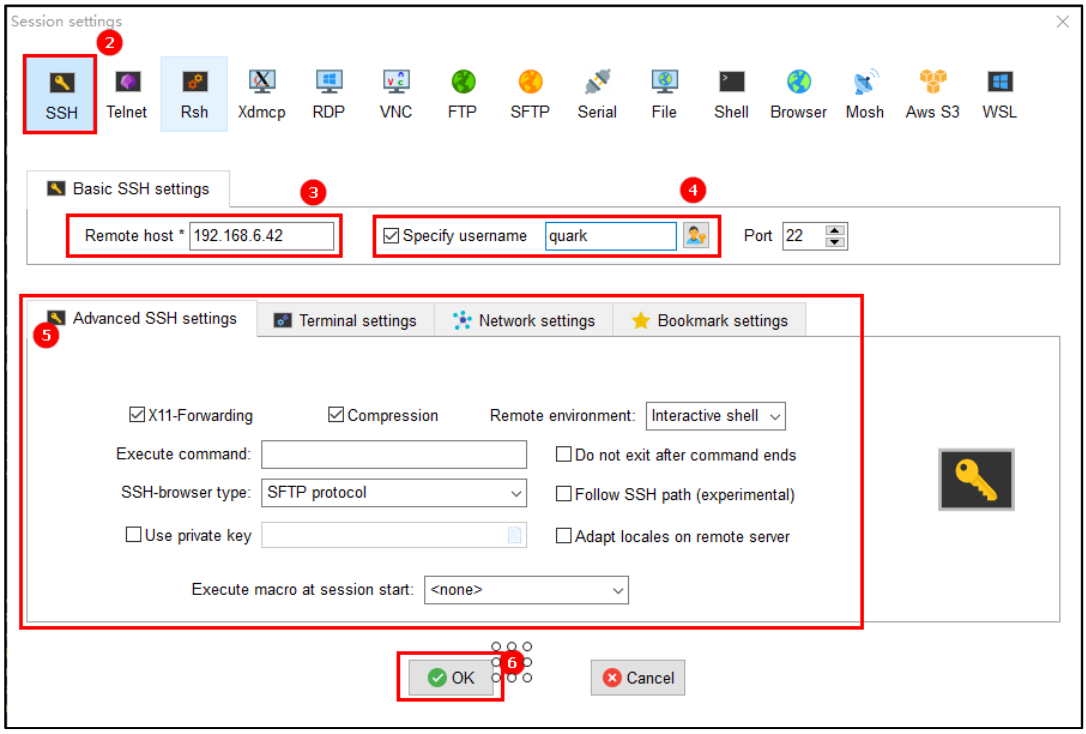


Figure 1.2-21 Establish SSH remote connection (2)

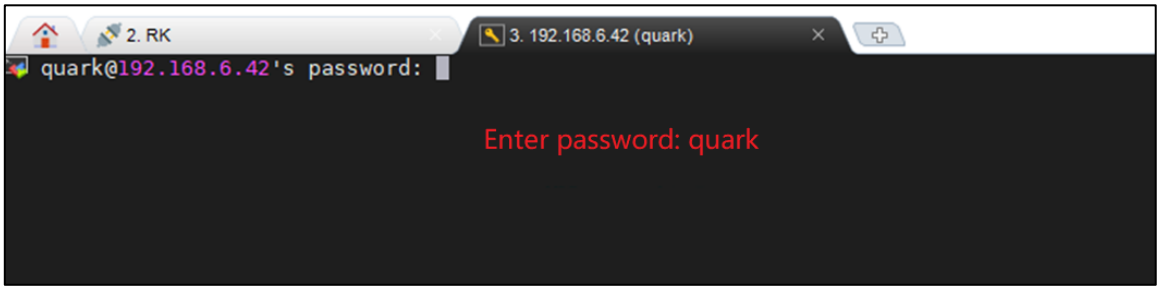


Figure 1.2-22 Establish SSH remote connection (3)

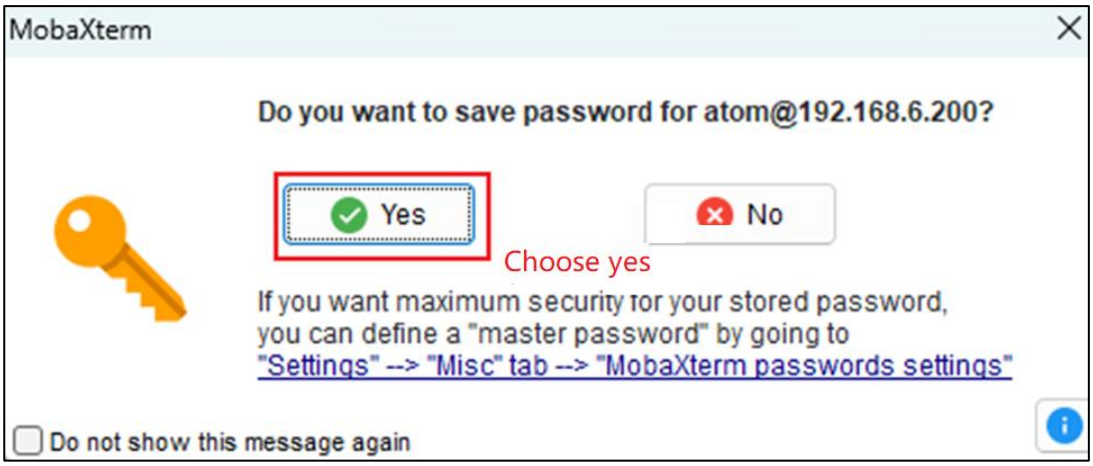


Figure 1.2-23 Establish SSH remote connection (4)

A pop-up window asks the user whether to save the password. We can choose "Yes" to save the password or "No" to not save it, depending on personal preference. At this point, the SSH remote connection has been successfully created, as shown in Figure 1.2.24:

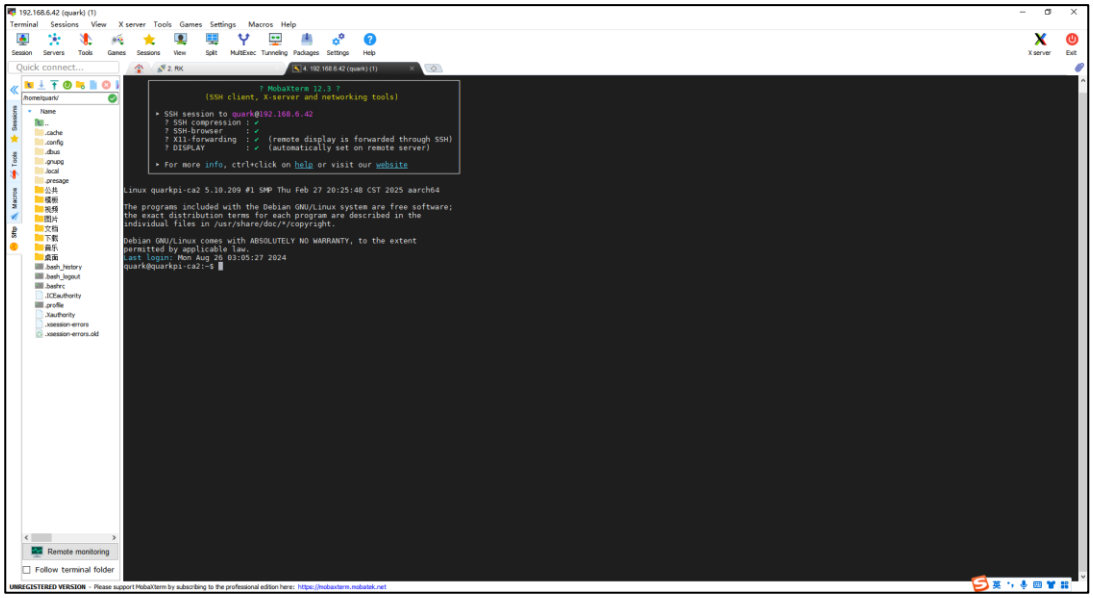


Figure 1.2-24 SSH remote login was successful.

Under the condition of SSH remote login, we can also use the MobaXterm software to achieve file transfer between Windows and Ubuntu (using the SFTP protocol for file transmission).

1.3 Installing SFTP Client on Windows

We need to install an FTP client software on the Windows system. Here, we choose FileZilla as the FTP client software. It is a free FTP client software.

The FileZilla software installation package has been provided to users in the card computer data package. The path is: Development board CD-ROM A drive - Basic materials -> 04, Software -> FileZilla_3.61.0_win64-setup.exe. Users can also download and install the package file through the link address: https://www.filezilla.cn/download/client.

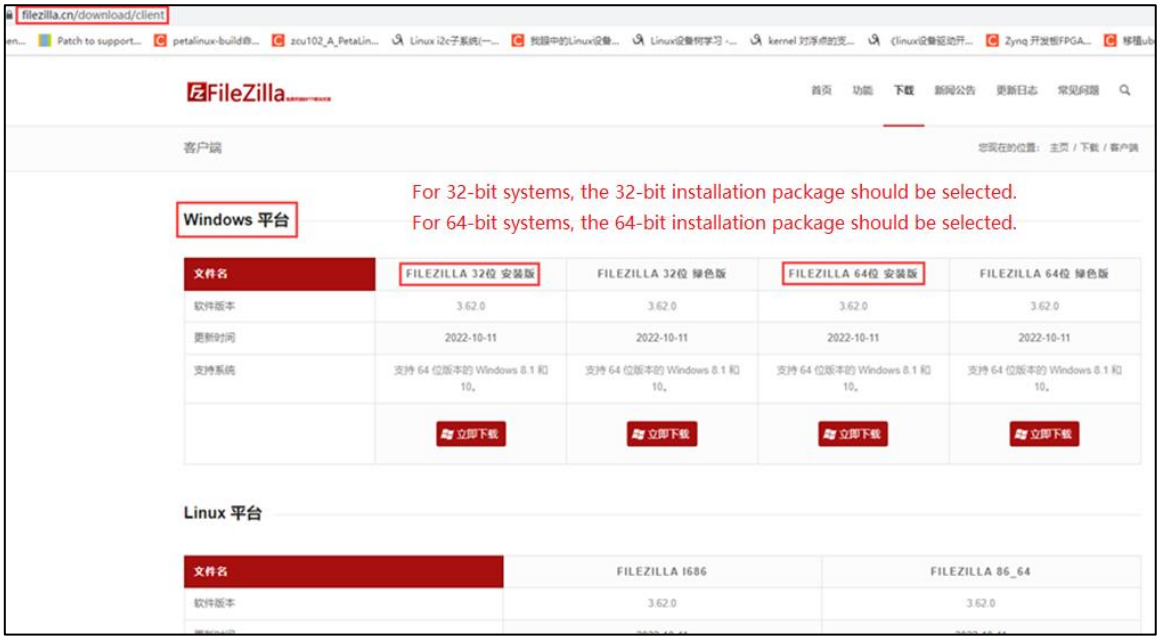


Figure 1.3-1 Download the installation package of FileZilla software

After the download is complete, it is as shown in Figure 1.3.2:

Figure 1.3-2 FileZilla software installation package

Next, double-click the installation package file FileZilla_3.61.0_win64-setup.exe and follow the steps shown in Figures 1.3.3 to 1.3.8 for installation:

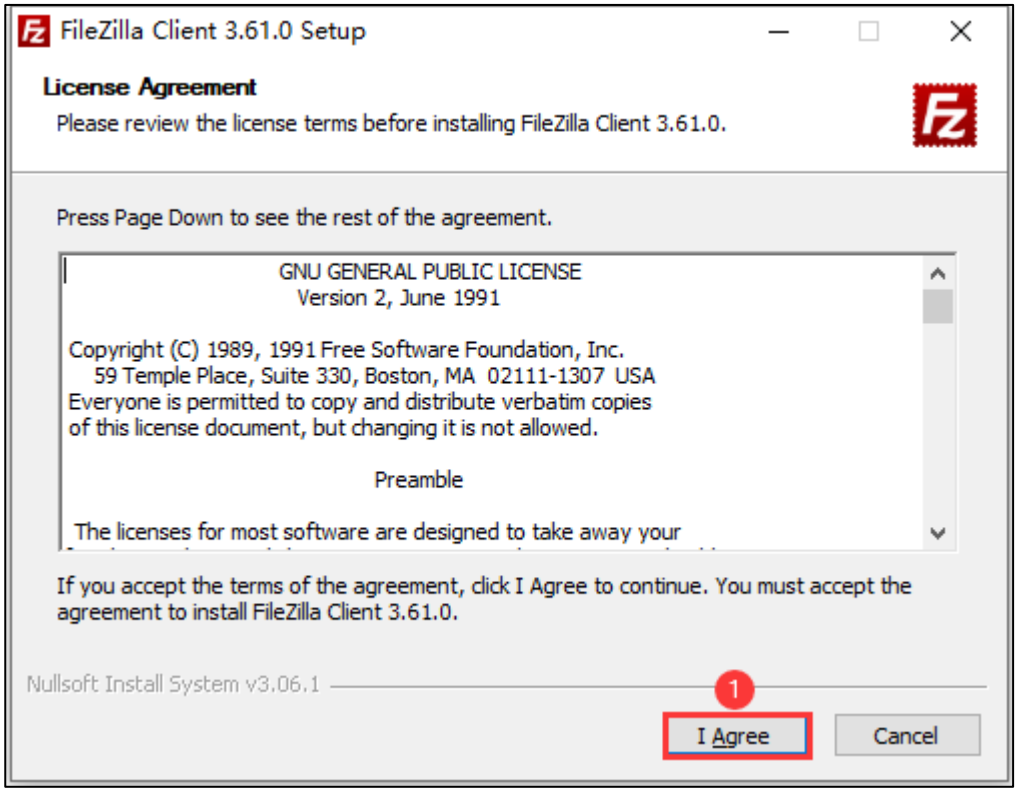


Figure 1.3-3 Install the FileZilla software (1)

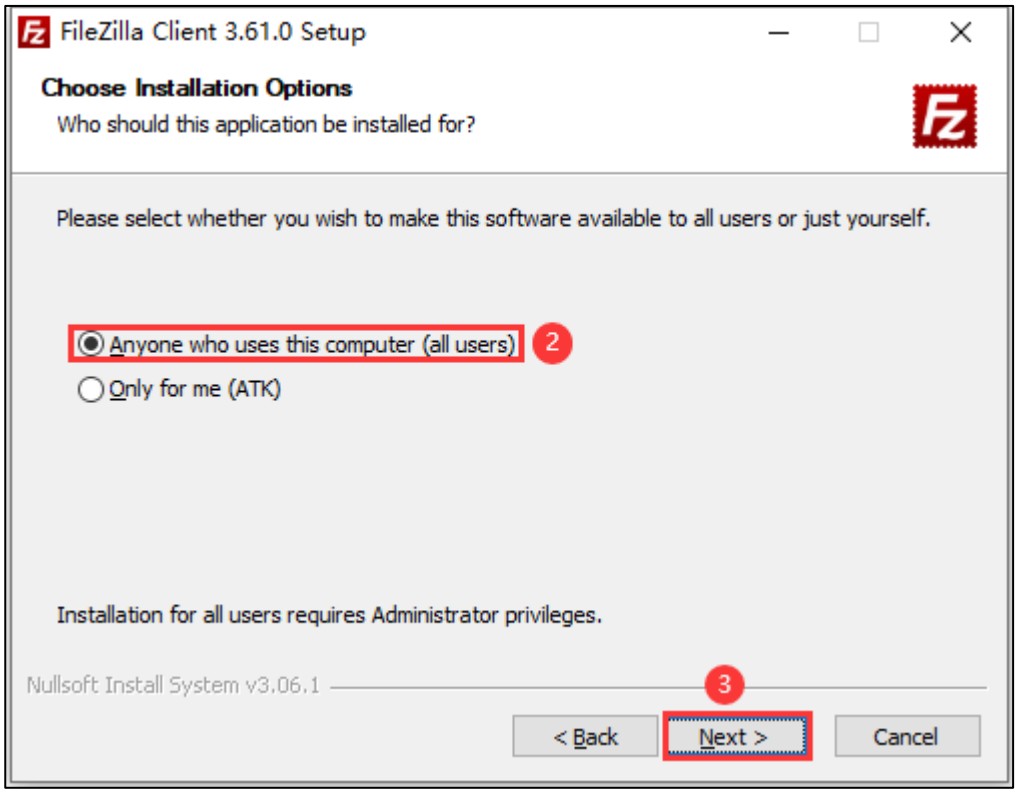


Figure 1.3-4 Install the FileZilla software (2)

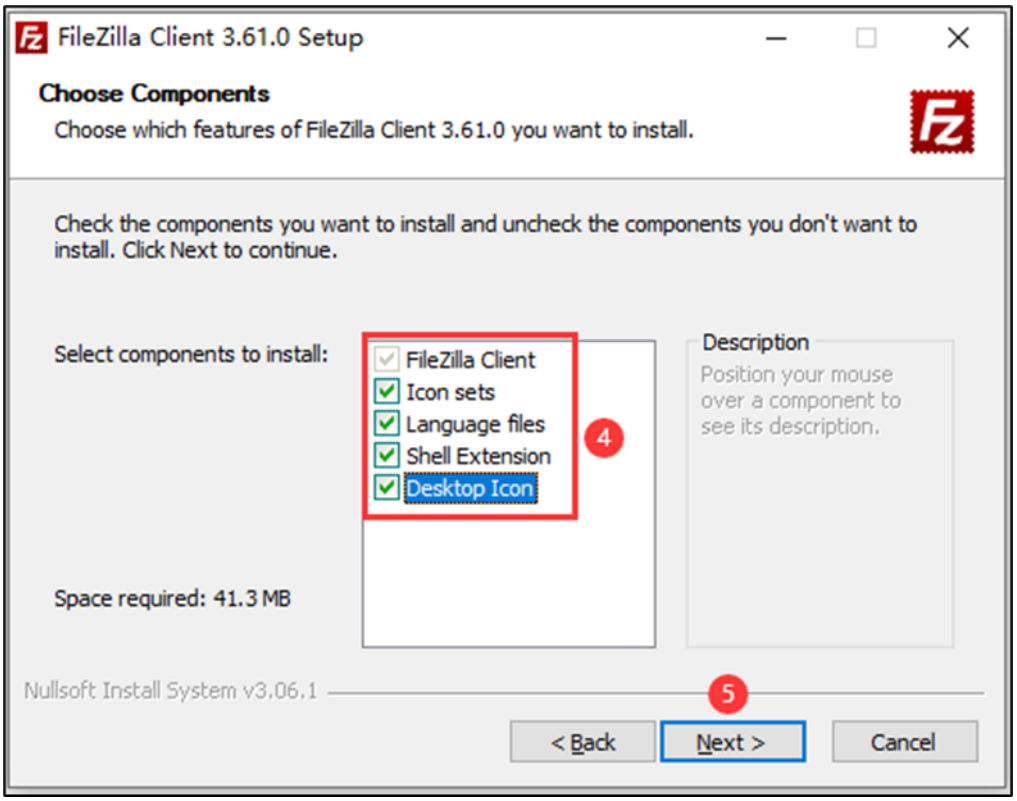


Figure 1.3-5 Install the FileZilla software (3)

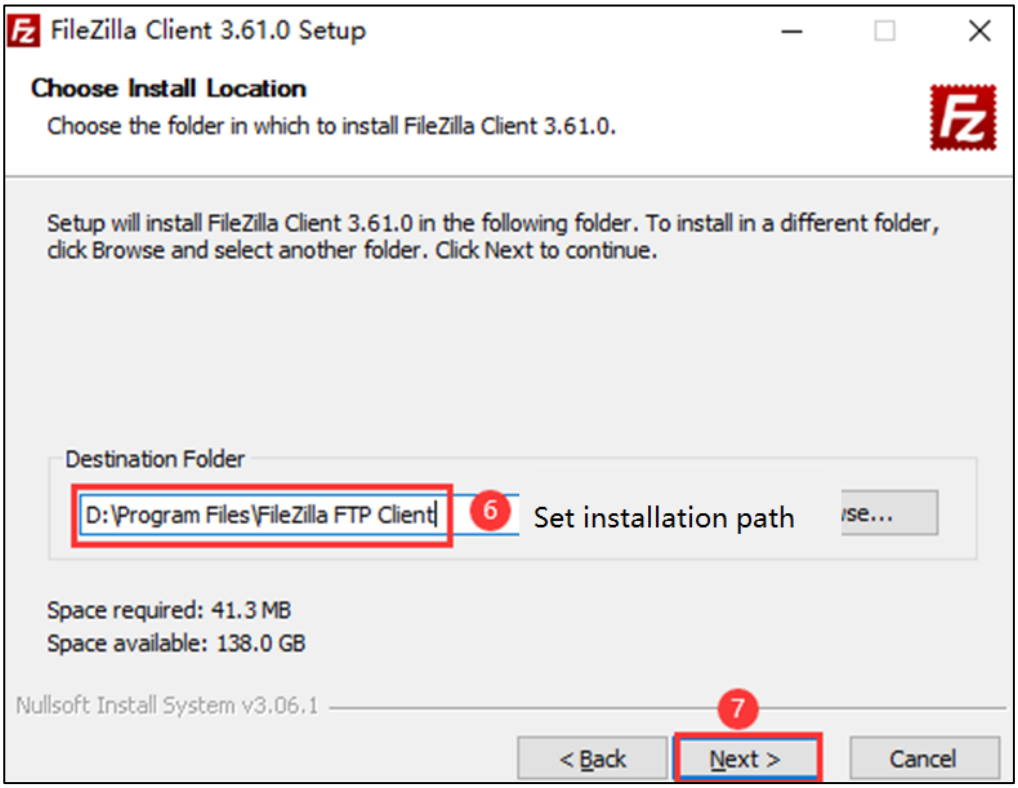


Figure 1.3-6 Install the FileZilla software (4)

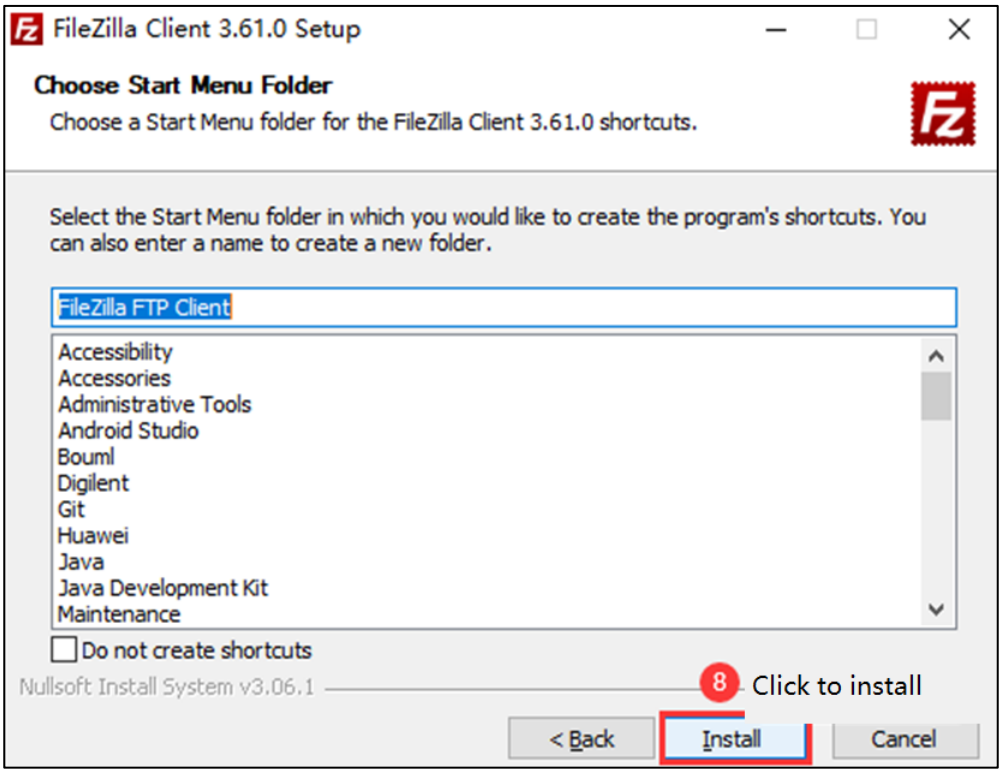


Figure 1.3-7 Install the FileZilla software (5)

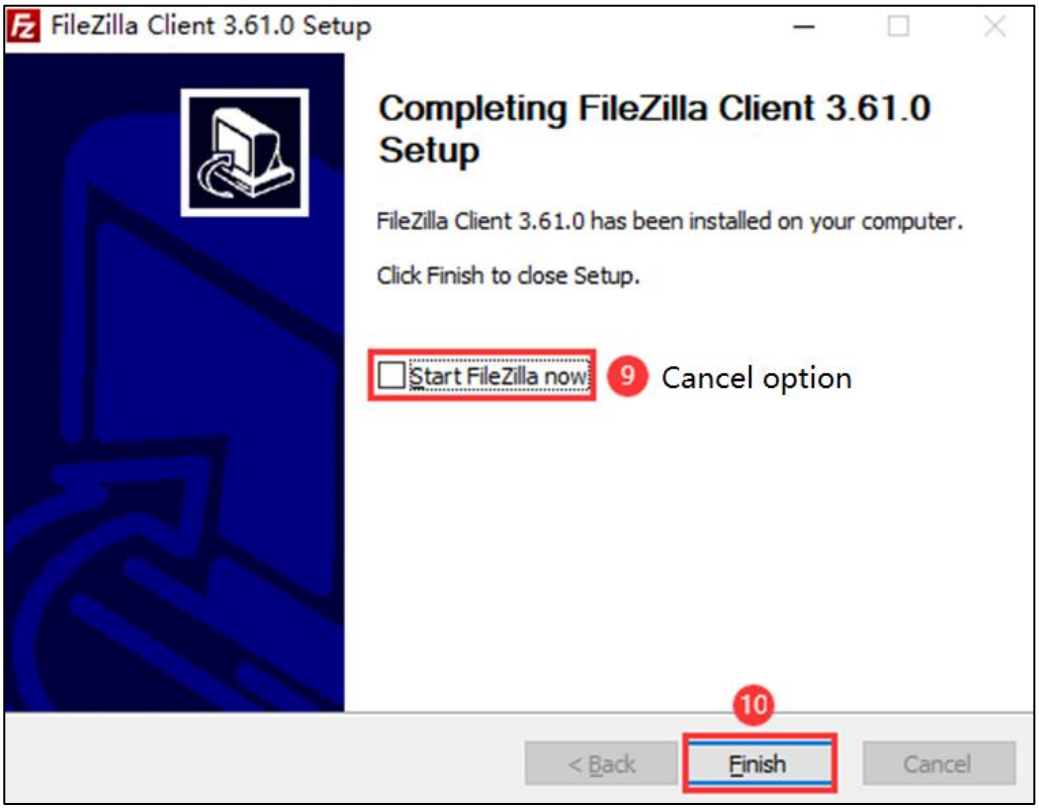


Figure 1.3-8 Install the FileZilla software (6)

At this point, the software installation is complete and a corresponding shortcut will be created on the desktop.

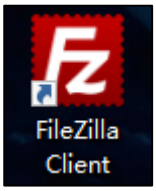


Figure 1.3-9 FileZilla desktop icon

1.3.1 How to Use FileZilla

Now, let's introduce the usage method of FileZilla. First, double-click the FileZilla desktop icon to open the software:

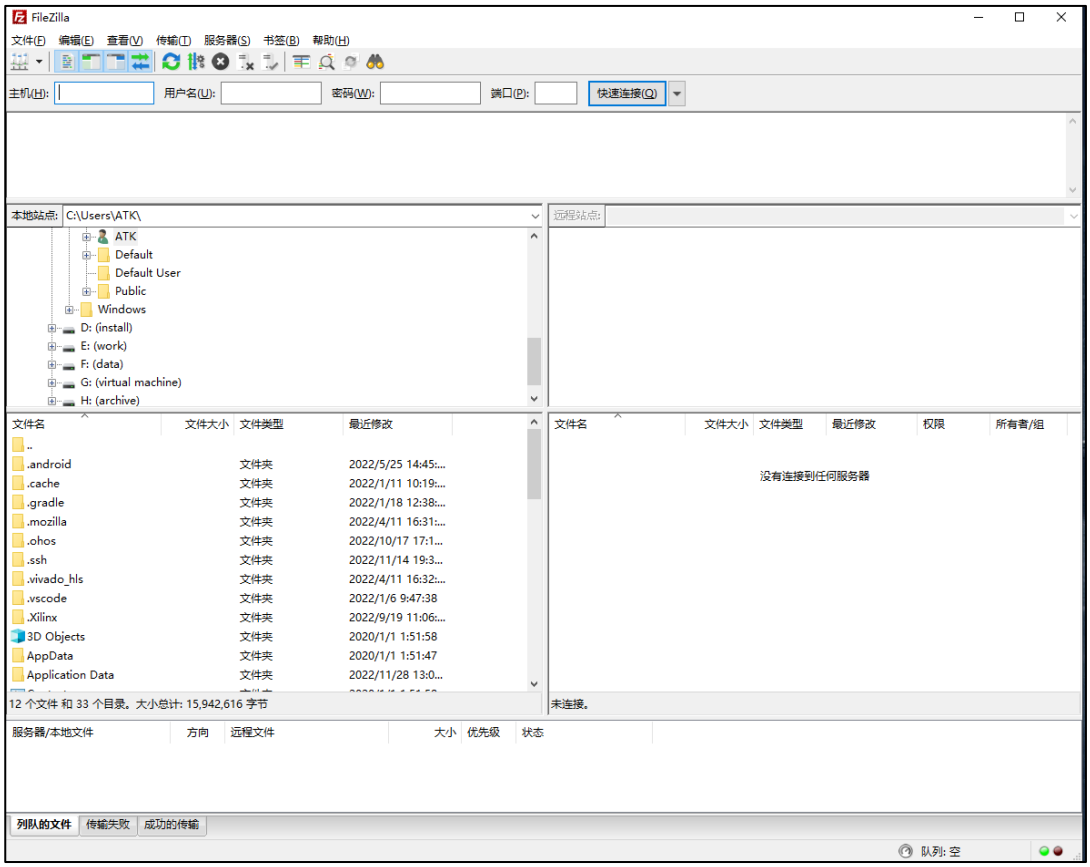


Figure 1.3-10 FileZilla software interface

When using a card computer as the FTP server and Windows as the FTP client, before conducting file transfer, the client needs to first connect to the server. Follow the steps shown in Figures 1.3.11 to 1.3.16 to connect to the FTP server:

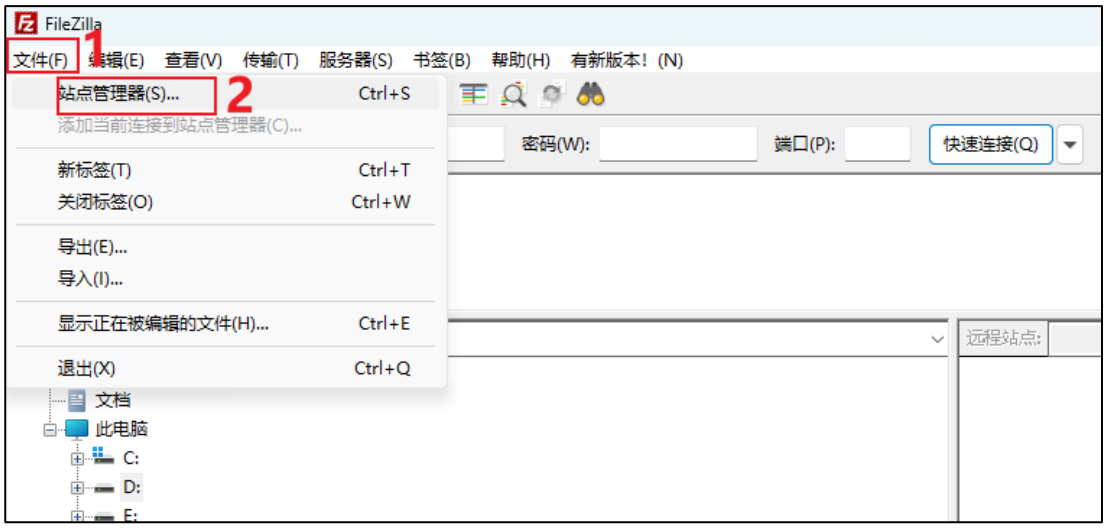


Figure 1.3-11 FTP client connects to the FTP server (1)

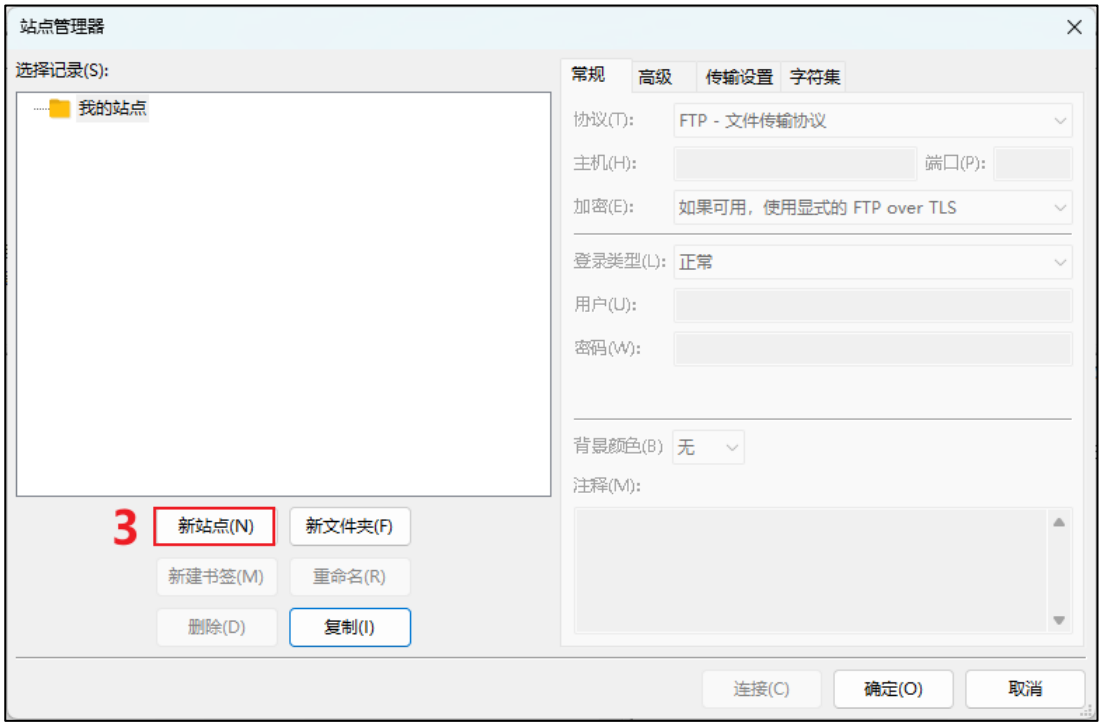


Figure 1.3-12 FTP client connects to the FTP server (2)

Figure 1.3-13 FTP client connects to the FTP server (3)

Figure 1.3-14 FTP client connects to the FTP server (4)

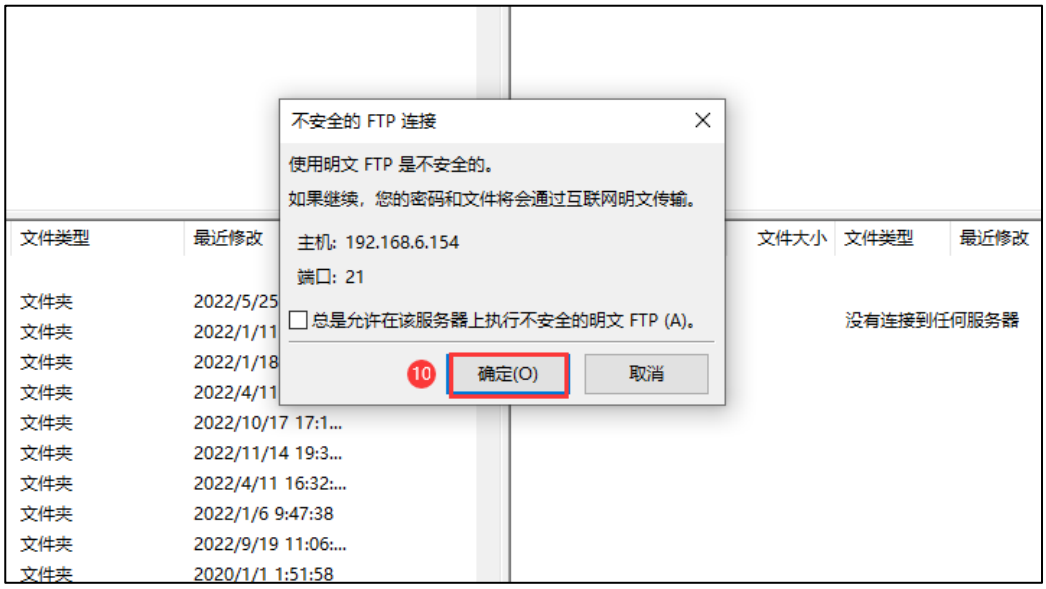


Figure 1.3-15 FTP client connects to the FTP server (5)

The connection was successful as shown in Figure 1.3.16:

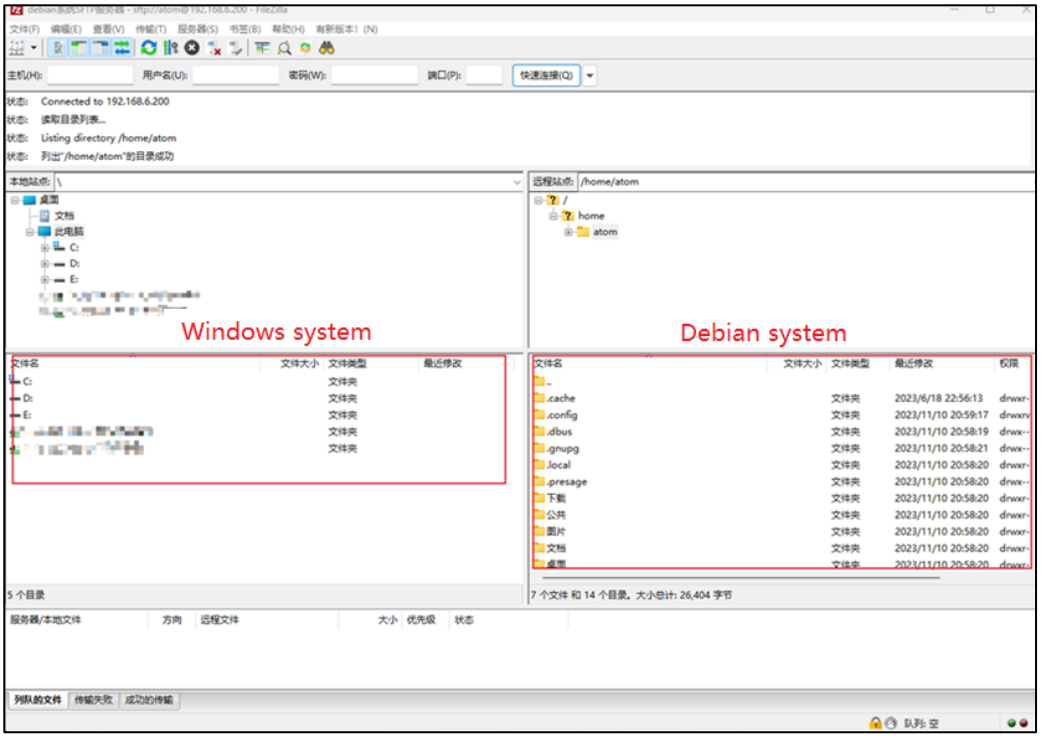


Figure 1.3-16 Connection successful

After the client successfully connects to the server, file transfer can be carried out. The transfer method is very simple. Select the file to be transferred, and directly use the left mouse button to drag it to the Windows directory area or the card computer system directory area! For example, to copy the test.txt file under the Windows system to the card computer system, first find this file test.txt in the left Windows directory area, then use the left mouse button to drag it to the right card computer directory area and release it; similarly, copying files from the card computer system to the Windows system is also the same!

Chapter 2. Testing of Card Computer Peripherals

This chapter will introduce how to test the card computer. It explains the usage and testing methods of the card computer. It is very helpful for development. Note that many tests require self-provided testing tools. Please complete the test steps according to your personal situation. Most of the hardware has already been tested when it was manufactured. Before testing, you need to switch to the root user and execute the su command to switch. The account password for root is root.

2.1 LED Testing

The ALIENTEK Card computer is equipped with a user LED. The relationship between them and the corresponding hardware pins on the baseboard is as shown in the table below. In the factory system, we use the LED as a heartbeat indicator, typically to indicate whether the system is operating normally.

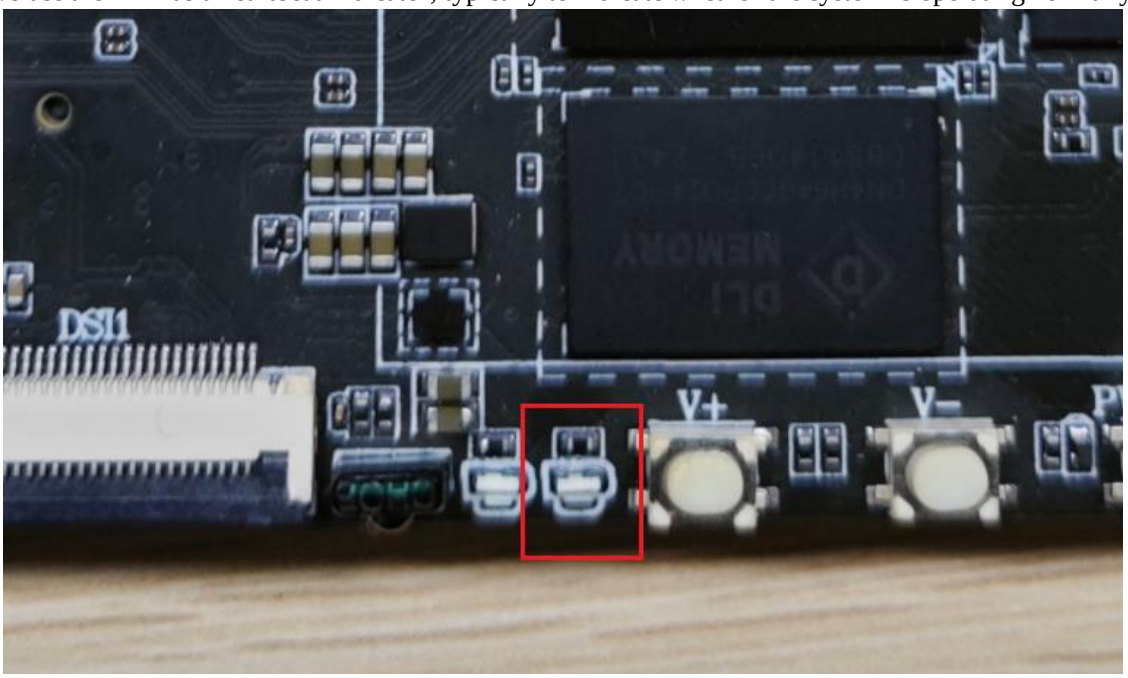


Figure 2.1-1 LED

Pin name	PI0
Card computer	GPIO1_D7

In the factory system, we can control the LED status by using the following instructions. After inputting them, check the LED status on the card computer.

cat /sys/class/leds/work/trigger # View the current triggering method and the supported triggering methods. The default is [heartbeat].

echo none > /sys/class/leds/work/trigger # Change the trigger mode of LED0

echo 1 > /sys/class/leds/work/brightness # Turn on LED0. At this time, it is in a constant-on state and the value ranges from 1 to 255.

echo 0 > /sys/class/leds/work/brightness # Turn off LED0. At this moment, it is in the off state, and the value is 0.

2.2 evtest Test

2.2.1 Key Test

The card computer is equipped with 4 buttons. Among them, 2 buttons are analogized using ADC0. When the buttons are pressed, corresponding numerical feedback will be provided. ADC0 has different voltage variations.

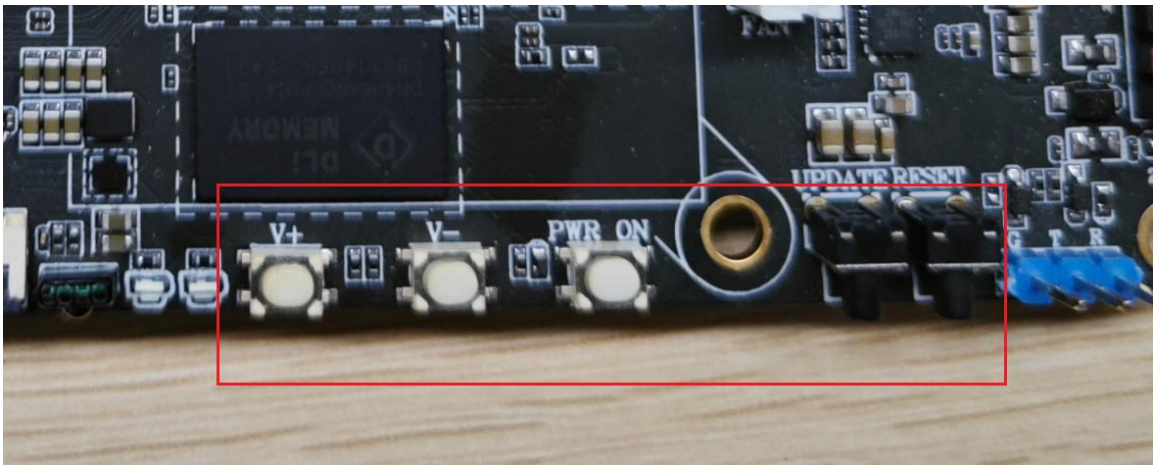


Figure 2.2-1 KEY

Name of the button	V+	V-
Code value	115	114

Run the following command in the card computer serial port terminal for testing, and then enter the corresponding number. Here, the number entered is "4" because the key event is event6. If you need to stop the test, press Ctrl + c. Please press the above two keys on the card computer. Do not press any other keys, such as the reset key REST.

```
evtest
```

The running result is as follows. As you can see, by pressing the corresponding key, the key type will be printed and the pressing time value will be 1, while the releasing time value will be 0.

```
root@quarkpi-ca2:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      fd8b0030.pwm
/dev/input/event1:      rk805 pwrkey
/dev/input/event2:      dsi1_ts_gt9xx
/dev/input/event3:      rockchip-dp0 rockchip-dp0
/dev/input/event4:      headset-keys
/dev/input/event5:      rockchip-es8388 Headset
/dev/input/event6:      adc-keys
/dev/input/event7:      rockchip-hdmi0 rockchip-hdmi0
Select the device event number [0-7]: 6
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "adc-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 114 (KEY_VOLUMEDOWN)
    Event code 115 (KEY_VOLUMEUP)
Properties:
Testing ... (interrupt to exit)
Event: time 1724612990.018564, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 1724612990.018564, -------------- SYN_REPORT ------------
Event: time 1724612990.121948, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 1724612990.121948, -------------- SYN_REPORT ------------
Event: time 1724612991.361892, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 1
Event: time 1724612991.361892, -------------- SYN_REPORT ------------
Event: time 1724612991.568556, type 1 (EV_KEY), code 115 (KEY_VOLUMEUP), value 0
Event: time 1724612991.568556, -------------- SYN_REPORT ------------
Event: time 1724612992.291896, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 1
Event: time 1724612992.291896, -------------- SYN_REPORT ------------
Event: time 1724612992.911603, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 0
Event: time 1724612992.911603, -------------- SYN_REPORT ------------
Event: time 1724612993.841887, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 1
Event: time 1724612993.841887, -------------- SYN_REPORT ------------
Event: time 1724612994.048349, type 1 (EV_KEY), code 114 (KEY_VOLUMEDOWN), value 0
Event: time 1724612994.048349, -------------- SYN_REPORT ------------
```

Figure 2.2-2 Evtest Key Test

2.2.2 Infrared Module Test

The silk-screened position of the onboard infrared module is as shown in the following figure.

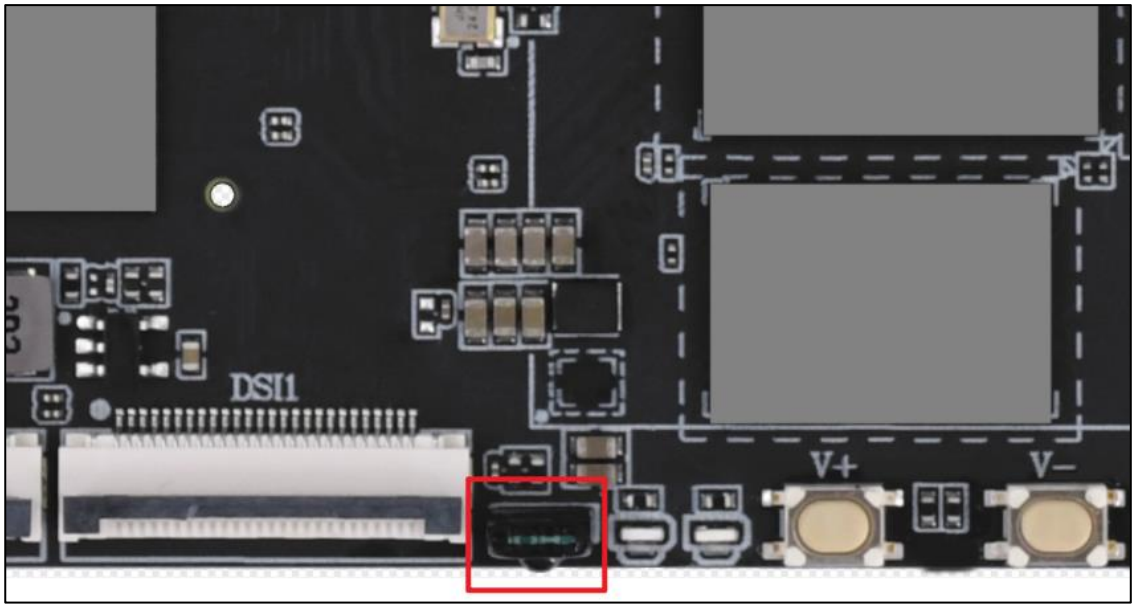


Figure 2.2-3 Built-in infrared module

We need to find an infrared remote control (the infrared remote control is a non-card computer accessory, please prepare it yourself), such as the infrared remote control module from the ALIENTEK store. If you have other infrared remote controls at hand, you can also test them. Note that the codes of different remote controls may be different. As shown in the following picture, this module is an infrared remote control from the ALIENTEK store, which is commonly used on the STM32 board. Our card computer has already decoded its infrared code into corresponding keys or function keys. For example, inputting the number "1" can be recognized as the number "1".

Figure 2.2-4 ALIENTEK Infrared Remote Controller

Now we will execute the following instructions to test this remote controller.

First, turn off the printing of information, and then carry out the test.

```
echo 0 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
evtest
```

The running result is as follows. As you can see, by pressing the corresponding key, the key type and the pressing time value (1 when pressed, 0 when released) will be printed. It can be observed that the corresponding keys have been decoded into corresponding numerical values.

```
root@quarkpi-ca2:~# echo 0 > /sys/module/rockchip_pwm_remotectl/parameters/code_print
root@quarkpi-ca2:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      fd8b0030.pwm
/dev/input/event1:      rk805 pwrkey
/dev/input/event2:      dsi1_ts_gt9xx
/dev/input/event3:      rockchip-dp0 rockchip-dp0
/dev/input/event4:      headset-keys
/dev/input/event5:      rockchip-es8388 Headset
/dev/input/event6:      adc-keys
/dev/input/event7:      rockchip-hdmi0 rockchip-hdmi0
Select the device event number [0-7]: 0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x524b product 0x6 version 0x100
Input device name: "fd8b0030.pwm"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 2 (KEY_1)
    Event code 3 (KEY_2)
    Event code 4 (KEY_3)
    Event code 5 (KEY_4)
    Event code 6 (KEY_5)
    Event code 7 (KEY_6)
    Event code 8 (KEY_7)
    Event code 9 (KEY_8)
    Event code 10 (KEY_9)
    Event code 11 (KEY_0)
    Event code 14 (KEY_BACKSPACE)
    Event code 102 (KEY_HOME)
    Event code 103 (KEY_UP)
    Event code 108 (KEY_DOWN)
    Event code 114 (KEY_VOLUMEDOWN)
    Event code 115 (KEY_VOLUMEUP)
    Event code 116 (KEY_POWER)
    Event code 143 (KEY_WAKEUP)
    Event code 164 (KEY_PLAYPAUSE)
    Event code 168 (KEY_REWIND)
    Event code 208 (KEY_FASTFORWARD)
Properties:
Testing ... (interrupt to exit)
Event: time 1724613342.893426, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1724613342.893426, -------------- SYN_REPORT ------------
Event: time 1724613343.184786, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1724613343.184786, -------------- SYN_REPORT ------------
```

Figure 2.2-5 Infrared receiver test

2.3 Screen test

2.3.1 HDMI display output

The default factory system supports HDMI. Please connect the display with an HDMI cable for testing the display.

2.3.2 DP display output

The default factory system also supports DP display output. Please use a full-function type-c cable or type-c to D adapter cable that supports DP output to connect the display for testing the display.

2.3.3 MIPI Display Output

The factory system does not default support MIPI display. Manual configuration is required. Customers who have purchased the MIPI screen from ALIENTEK should connect the wires as shown in the figure below. As shown in the figure:

Figure 2.3-1 Connect the board to the MIPI screen.

Then modify the /boot/mmc0_extlinux/extlinux.conf configuration file in the Debian system, add the configuration items in the red box as shown in the figure. Take a 5.5-inch 1080P MIPI screen as an example here, then save the configuration file and restart the card computer.

```
#timeout 10
#menu title select kernel

label QUARKPI-CA2
    kernel /Image
    fdt /rk3588s-quarkpi-ca2.dtb
    fdtoverlays /dtbs/rk3588s-cam1.dtbo /dtbs/rk3588s-dsi1-5p5-1080x1920-vp2.dtbo
    append earlycon=uart8250,mmio32,0xfeb50000 console=ttyFIQ0 earlyprintk root=/dev/mmcblk0p3 rootwait rw
~
~
~
~
~
~
~
~
```

Figure 2.3-2 Add 5.5-inch MIPI screen configuration

For other MIPI screen configurations, you can refer to the dtbo files in the /boot/dtbs directory. It should be noted that for the same VP, only one MIPI screen can be configured at a time.

```
root@quarkpi-ca2:~# ls /boot/dtbs/rk3588s-dsi* -l
-rwxr-xr-x 1 root root 6926  2月  6  2025 /boot/dtbs/rk3588s-dsi1-10p1-800x1280-vp2.dtbo
-rwxr-xr-x 1 root root 6926  2月  6  2025 /boot/dtbs/rk3588s-dsi1-10p1-800x1280-vp3.dtbo
-rwxr-xr-x 1 root root 6298  2月  6  2025 /boot/dtbs/rk3588s-dsi1-5p5-1080x1920-vp2.dtbo
-rwxr-xr-x 1 root root 6298  2月  6  2025 /boot/dtbs/rk3588s-dsi1-5p5-1080x1920-vp3.dtbo
-rwxr-xr-x 1 root root 6302  2月  6  2025 /boot/dtbs/rk3588s-dsi1-5p5-720x1280-vp2.dtbo
-rwxr-xr-x 1 root root 6302  2月  6  2025 /boot/dtbs/rk3588s-dsi1-5p5-720x1280-vp3.dtbo
-rwxr-xr-x 1 root root 6754  2月  6  2025 /boot/dtbs/rk3588s-dsi2-10p1-800x1280-vp2.dtbo
-rwxr-xr-x 1 root root 6754  2月  6  2025 /boot/dtbs/rk3588s-dsi2-10p1-800x1280-vp3.dtbo
-rwxr-xr-x 1 root root 6126  2月  6  2025 /boot/dtbs/rk3588s-dsi2-5p5-1080x1920-vp2.dtbo
-rwxr-xr-x 1 root root 6126  2月  6  2025 /boot/dtbs/rk3588s-dsi2-5p5-1080x1920-vp3.dtbo
-rwxr-xr-x 1 root root 6130  2月  6  2025 /boot/dtbs/rk3588s-dsi2-5p5-720x1280-vp2.dtbo
-rwxr-xr-x 1 root root 6130  2月  6  2025 /boot/dtbs/rk3588s-dsi2-5p5-720x1280-vp3.dtbo
root@quarkpi-ca2:~#
```

Figure 2.3-3 MIPI screen DTBO file

2.4 Audio Testing

Use the system command aplay or execute the following command to play the audio file. Users can also copy the audio file to the file system and use the command aplay/gst-play-1.0 + audio file to play it.

Note: Do not open the music or video player on the factory interface to play audio or video. Otherwise, it will occupy the audio device. The reason why the computer can play multiple audio simultaneously is due to software support and algorithm optimization. Please connect the headphones!

aplay /usr/share/sounds/alsa/Front_Center.wav # Use aplay to play audio files. It supports wav format but does not support mp3.

You can use the following command to record a WAV audio file, and then use the aplay command to play it.

arecord -D hw:2,0 -f cd -t wav test.wav

You can determine the sound card to be used by the following command. Since we are using the ES8388 for recording, in the arecord command, we should use "hw:2,0" for the recording.

aplay -l

```
root@quarkpi-ca2:/home/quark# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.1-0011-0 [dailink-multicodecs ES8323.1-0011-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
root@quarkpi-ca2:/home/quark#
```

Figure 2.4-1 Sound card in the system

Camera test

2.4.1 MIPI camera

Supports MIPI CSI camera. ALIENTEK has equipped this card computer with IMX415 (800 pixel). As shown in the figure below (taking IMX415 as an example).

Special note: IMX415 requires version V1.5 or above. Versions below are not compatible!!!

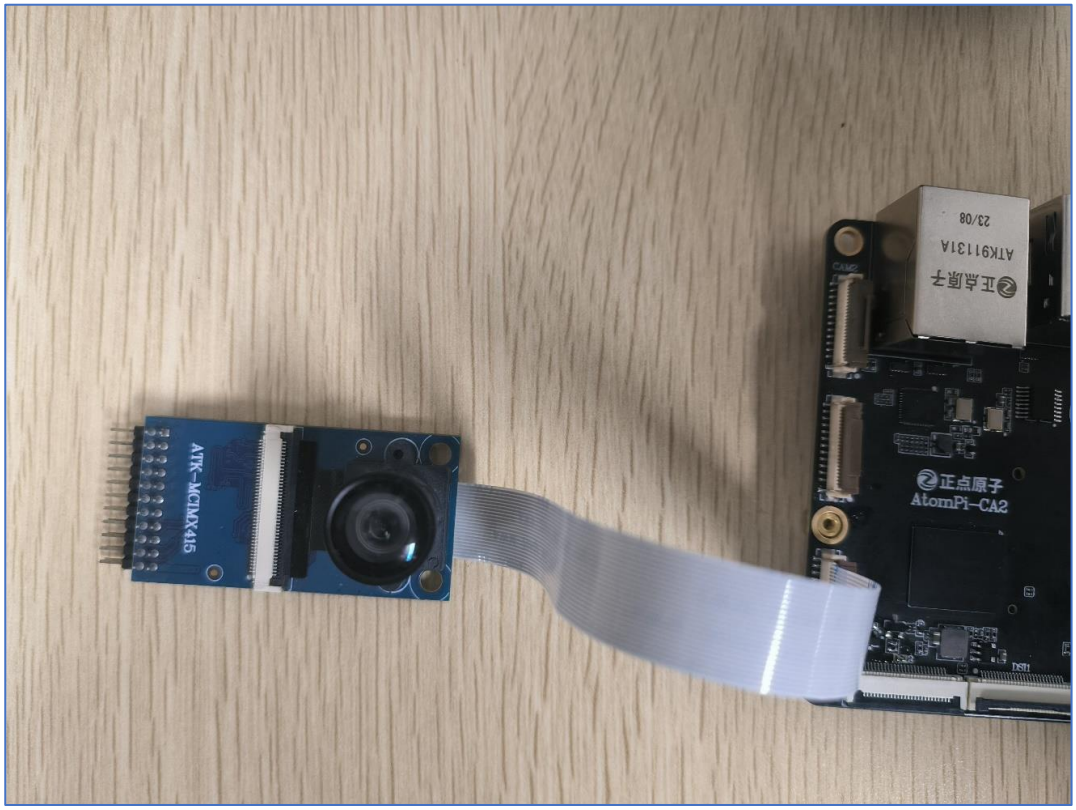


Figure 2.4-2 MIPI Camera Connection

Execute the following command to capture video and display it on the MIPI screen (you need to configure the dtbo file of the MIPI screen first). Note: The App in the Debian system cannot be used simultaneously with the command for video capture below.

```
gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=NV12,width=640,height=480,\
framerate=30/1 ! videoconvert ! ximagesink
```

The factory system has enabled all the camera interfaces by default. The corresponding relationship between the interface markings on the board and the /dev/video nodes of the system is as follows:

Board card silk-screen printing	/dev/videoX
CAM1	/dev/video33
CAM2	/dev/video42
CAM3	/dev/video51

2.4.2 USB Camera

Similarly, before the experiment, please prepare a USB camera that complies with the UVC (USB video device class) protocol. Any camera that conforms to this protocol is acceptable. UVC, full name: USB video class or USB video device class, is a protocol standard jointly developed by Microsoft and several other device manufacturers for USB video capture devices. Hardware devices that meet UVC specifications can be used normally in the host without the need to install additional drivers.

Insert the USB camera, which supports hot plugging. The serial port printout is as follows.

```
root@quarkpi-ca2:~# ls
root@quarkpi-ca2:~# [   32.156204] usb 2-1: new high-speed USB device number 2 using ehci-platform
[   32.350359] usb 2-1: New USB device found, idVendor=0c45, idProduct=636b, bcdDevice= 1.00
[   32.350452] usb 2-1: New USB device strings: Mfr=2, Product=1, SerialNumber=3
[   32.350475] usb 2-1: Product: RGB Camera
[   32.350496] usb 2-1: Manufacturer: Sonix Technology Co., Ltd.
[   32.350516] usb 2-1: SerialNumber: SN0001
[   32.353576] uvcvideo: Found UVC 1.00 device RGB Camera (0c45:636b)
[   32.355215] uvcvideo: Failed to query (GET_INFO) UVC control 5 on unit 1: 0 (exp. 1).
[   32.355843] uvcvideo: Failed to query (GET_INFO) UVC control 7 on unit 1: 0 (exp. 1).
[   32.356148] uvcvideo: Failed to query (GET_INFO) UVC control 9 on unit 1: 0 (exp. 1).
[   32.372854] input: RGB Camera: RGB Camera as /devices/platform/fc880000.usb/usb2/2-1/2-1:1.0/input/input8
[   32.412333] usb 2-1: 3:1: cannot get freq at ep 0x84
[   32.420656] usbcore: registered new interface driver snd-usb-audio
[   32.607743] usb 2-1: 3:1: cannot get freq at ep 0x84
[   32.629372] usb 2-1: 3:1: cannot get freq at ep 0x84

root@quarkpi-ca2:~#
root@quarkpi-ca2:~#
```

Figure 2.4-3 USB camera

Reinsert and remove the USB device and check the nodes under the /dev/video* directory to determine which node is the camera. Mine is video60.

Execute the following command to capture video and display it on the MIPI screen. Note: In the Debian system, the App and the command for capturing video cannot be used simultaneously.

```
gst-launch-1.0 v4l2src device=/dev/video60 ! video/x-raw,format=YUY2,width=640,height=480,\
framerate=30/1 ! videoconvert ! ximagesink
```

2.5 USB & TF Card & EMMC Testing

2.5.1 USB HOST Testing

This section will conduct the testing based on the USB test. The TF card and EMMC tests are similar to the USB test, as long as the operation partition is confirmed, the read and write speed tests can be carried out.

Insert a FAT32 formatted USB drive directly into the USB interface using a card reader (any of the 4 USB interfaces will do, with two of them being USB 2.0, the blue interface above is USB 3.0, and the blue interface below is USB 2.0, as shown in the following figure.

```
root@quarkpi-ca2:~# [  743.666097] usb 2-1: new high-speed USB device number 4 using ehci-platform
[  743.815181] usb 2-1: New USB device found, idVendor=14cd, idProduct=1212, bcdDevice= 1.00
[  743.815277] usb 2-1: New USB device strings: Mfr=1, Product=3, SerialNumber=2
[  743.815300] usb 2-1: Product: Mass Storage Device
[  743.815320] usb 2-1: Manufacturer: Generic
[  743.815340] usb 2-1: SerialNumber: 121220160204
[  743.817051] usb-storage 2-1:1.0: USB Mass Storage device detected
[  743.817933] scsi host0: usb-storage 2-1:1.0
[  744.841327] scsi 0:0:0:0: Direct-Access     Mass     Storage Device   1.00 PQ: 0 ANSI: 0 CCS
[  744.956616] sd 0:0:0:0: [sda] 61067264 512-byte logical blocks: (31.3 GB/29.1 GiB)   Capacity size
[  744.957578] sd 0:0:0:0: [sda] Write Protect is off
[  744.958384] sd 0:0:0:0: [sda] No Caching mode page found
[  744.958440] sd 0:0:0:0: [sda] Assuming drive cache: write through
[  744.987007]  sda: sda1   Zoning
[  744.991962] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

Figure 2.5-1 USB drive output information

2.5.2 USB 2.0 Test

Note: The test read/write speed is related to the maximum read/write speed of the personal card reader or USB/TF card.

Write speed test:

```
mount /dev/sda1 /mnt/
dd if=/dev/zero of=/mnt/write_file bs=1M count=1024 conv=fdatasync
```

```
root@quarkpi-ca2:~# mount /dev/sda1 /mnt/
[  884.538901] FAT-fs (sda1): utf8 is not a recommended IO charset for FAT filesystems, filesystem wdd if=/dev/zero of=/mnt/write_file bs=1M count=1024
 conv=fdatasync2:~# dd if=/dev/zero of=/mnt/write_file bs=1M count=1024 conv=fdatasync
记录了1024+0 的读入
记录了1024+0 的写出
1073741824字节（1.1 GB，1.0 GiB）已复制，129.898 s，8.3 MB/s
root@quarkpi-ca2:~#
root@quarkpi-ca2:~#
```

Figure 2.5-2 USB 2.0 Test Results

Use the dd command for reading. Due to the mechanism of the Linux system, generally there is no need to specifically release the cache that has been used. These cache contents can increase the reading and writing speed of the file.

Execute the following command to clear the cache.

```
echo 3 > /proc/sys/vm/drop_caches          # Clear cache
dd if=/mnt/write_file of=/dev/null bs=1M   # Read speed test
rm /mnt/write_file                         # Delete this file after reading it.
```

```
root@quarkpi-ca2:~#
root@quarkpi-ca2:~# echo 3 > /proc/sys/vm/drop_caches
[ 1111.605048] bash (1222): drop_caches: 3
root@quarkpi-ca2:~# dd if=/mnt/write_file of=/dev/null bs=1M
记录了1024+0 的读入
记录了1024+0 的写出
1073741824字节（1.1 GB，1.0 GiB）已复制，68.6132 s，15.6 MB/s
root@quarkpi-ca2:~#
root@quarkpi-ca2:~# rm /mnt/write_file
root@quarkpi-ca2:~#
```

Figure 2.5-3 Test results after clearing the cache

2.5.3 USB3.0 Test

As mentioned earlier, the Type-C interface and USB3.0 interface are compatible. When we want to use the USB3.0 interface as the host, the Type-C interface does not need to be connected with a cable. Similarly, the test method is the same as the test steps in 3.6.2. Note: The read/write speed is most closely related to the speed of the USB flash drive.

2.5.4 Slave Test

The OTG interface on the card computer can be used as ADB. We connect the OTG interface to the PC (computer) using a USB Type-C cable. Using the adb command, communication with the card computer can be achieved.

2.6 Network Testing

The card computer is equipped with a gigabit network chip, which is adaptive to 10/100/1000M. To test the network in the file system, use the following commands. Plug the network cable into one of the two network ports and ensure that the network cable can access the internet. Check the IP obtained from the network.

Note: The experimental environment requires that the board needs to be directly connected to the router!

Test whether it is a gigabit network. Pay attention to using a gigabit network cable, a gigabit router (most common home routers are of 100M), or a gigabit switch. The PC (computer) network card must be a gigabit network card. Otherwise, the speed tested or identified may be 100M! As shown in the figure, seeing "1Gbps/Full" indicates it is a gigabit network.

```
root@quarkpi-ca2:~# [ 1567.909297] rk_gmac-dwmac fe1c0000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[ 1567.909439] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
```

Figure 2.6-1 The board card is connected to the gigabit router via network cable.

2.6.1 Serial Port Terminal Command Line Mode

ifconfig

```
root@quarkpi-ca2:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.6.156  netmask 255.255.255.0  broadcast 192.168.6.255
        inet6 fe80::7826:6e16:7a85:87fd  prefixlen 64  scopeid 0x20<link>
        ether 0e:90:cb:b3:a0:d2  txqueuelen 1000  (Ethernet)
        RX packets 1990  bytes 131438 (128.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 129  bytes 11552 (11.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 100

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 1016  bytes 79032 (77.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1016  bytes 79032 (77.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@quarkpi-ca2:~#
```

Figure 2.6-2 Board card Gigabit network port IP address

Check if the network can access the Internet. If there is a data response, it indicates that the network function is normal. Press Ctrl+c to terminate the instruction.

ping www.baidu.com

```
root@quarkpi-ca2:~# ping www.baidu.com
PING www.a.shifen.com (183.2.172.42) 56(84) bytes of data.
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=1 ttl=52 time=4.28 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=2 ttl=52 time=4.16 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=3 ttl=52 time=3.99 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=4 ttl=52 time=4.44 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=5 ttl=52 time=4.58 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=6 ttl=52 time=4.39 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=7 ttl=52 time=4.19 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=8 ttl=52 time=4.19 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=9 ttl=52 time=4.42 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=10 ttl=52 time=4.12 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=11 ttl=52 time=4.30 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=12 ttl=52 time=3.93 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=13 ttl=52 time=4.36 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=14 ttl=52 time=4.32 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=15 ttl=52 time=4.18 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=16 ttl=52 time=4.17 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=17 ttl=52 time=4.41 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=18 ttl=52 time=4.29 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=19 ttl=52 time=4.64 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=20 ttl=52 time=3.99 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=21 ttl=52 time=4.13 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=22 ttl=52 time=4.33 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=23 ttl=52 time=4.38 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=24 ttl=52 time=4.37 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=25 ttl=52 time=4.20 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=26 ttl=52 time=4.24 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=27 ttl=52 time=4.33 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=28 ttl=52 time=4.05 ms
64 bytes from 183.2.172.42 (183.2.172.42): icmp_seq=29 ttl=52 time=4.34 ms
^C
--- www.a.shifen.com ping statistics ---
29 packets transmitted, 29 received, 0% packet loss, time 28045ms
rtt min/avg/max/mdev = 3.926/4.265/4.644/0.164 ms
root@quarkpi-ca2:~#
```

Figure 2.6-3 Output of the ping command

The iperf3 command can be used to test the connection speed of a gigabit network.

Set Ubuntu as the server. Ubuntu and the card computer must be in the same local area network. If Ubuntu does not have the iperf command, you need to enter "sudo apt install iperf3" to install it.

View Ubuntu IP

```
ifconfig
```

```
lixinkai@lixinkai-virtual-machine:~/atk_quarkpi-ca2_buildroot$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.6.83  netmask 255.255.255.0  broadcast 192.168.6.255
        inet6 fe80::ca29:f1ee:fb9c:bfda  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:28:53:b9  txqueuelen 1000  (以太网)
        RX packets 1408769  bytes 1114506428 (1.1 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 147934  bytes 10788583 (10.7 MB)
        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  (本地环回)
        RX packets 892  bytes 83825 (83.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 892  bytes 83825 (83.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lixinkai@lixinkai-virtual-machine:~/atk_quarkpi-ca2_buildroot$
```

Figure 2.6-4 Virtual machine IP address

Enter iperf3 -s to set Ubuntu as the server.

```
iperf3 -s
```

```
lixinkai@lixinkai-virtual-machine:~/atk_quarkpi-ca2_buildroot$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

Figure 2.6-5 iperf3 server

The QuarkPi-Ca2 board is the client. By using the iperf3 command, input the IP address of the server (Ubuntu) to proceed.

```
iperf3 -c 192.168.6.83 -i 1
```

Command parsing:

- -s: Indicates that the current device acts as a server.
- -c: Indicates that the current device acts as a client.
- 192.168.6.83: The IP address of the server.
- -i: Trigger period.

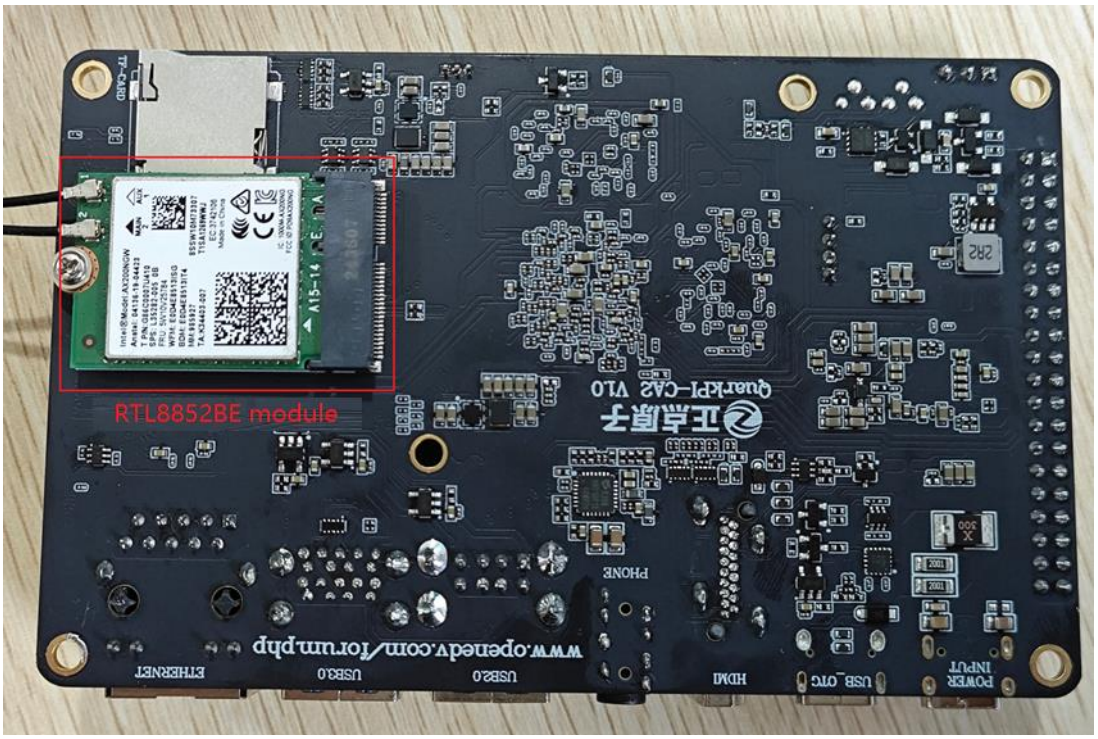
```
root@quarkpi-ca2:~# iperf3 -c 192.168.6.83 -i 1
Connecting to host 192.168.6.83, port 5201
[  5] local 192.168.6.156 port 36874 connected to 192.168.6.83 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   114 MBytes   953 Mbits/sec    0    515 KBytes
[  5]   1.00-2.00   sec   106 MBytes   891 Mbits/sec    0    594 KBytes
[  5]   2.00-3.00   sec   112 MBytes   940 Mbits/sec    0    594 KBytes
[  5]   3.00-4.00   sec   112 MBytes   938 Mbits/sec    0    624 KBytes
[  5]   4.00-5.00   sec   113 MBytes   946 Mbits/sec    0    666 KBytes
[  5]   5.00-6.00   sec   112 MBytes   937 Mbits/sec    0    666 KBytes
[  5]   6.00-7.00   sec   112 MBytes   943 Mbits/sec    0    666 KBytes
[  5]   7.00-8.00   sec   113 MBytes   946 Mbits/sec    0    666 KBytes
[  5]   8.00-9.00   sec   112 MBytes   940 Mbits/sec    0    666 KBytes
[  5]   9.00-10.00  sec   111 MBytes   928 Mbits/sec    0    747 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.09 GBytes   936 Mbits/sec    0             sender
[  5]   0.00-10.04  sec  1.09 GBytes   931 Mbits/sec                  receiver

iperf Done.
```

Figure 2.6-6 iperf3 test results

The Bitrate received and sent by the test are approximately 936 Mbits/sec, which is the speed of a gigabit network (please do not run other applications before the test to avoid affecting the test speed). If it is not so, please check whether a gigabit network cable is used, whether the network card of the PC is a gigabit card, whether the router is gigabit, etc. None of the devices can be of the hundred-megabit type; otherwise, the test result will be that of a hundred-megabit network.

2.7 M.2 PCI-E WIFI Test

The QuarkPi-CA2 board card is factory-adapted with the RTL 8852BE PCIE WIFI module. Since there are many versions of the RTL PCIE WIFI module, it is recommended to directly purchase the RTL 8852BE PCIE WIFI module from the ALIENTEK Taobao flagship store for testing. Follow the installation steps shown in the figure below. Before testing, it is necessary to install the module antenna to enhance the signal.



After connecting the PCIE WIFI module, power on the board. The factory system can automatically recognize the RTL8852BE PCIE WIFI module and automatically load the corresponding driver, as shown in the following figure:

```
root@quarkpi-ca2:~#
root@quarkpi-ca2:~# lsmod
Module                  Size  Used by
rtk_btusb              77824  0
8852be               4071424  0
btbcm                  24576  0
btintel                28672  0
root@quarkpi-ca2:~#
```

Figure 2.7-1 Related driver

Execute the "ifconfig" command to view the wlan node of the PCIE WIFI

```
ifconfig
```

```
root@quarkpi-ca2:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 0e:90:cb:b3:a0:d2  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 100

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 936  bytes 72744 (71.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 936  bytes 72744 (71.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlP4p65s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether ca:fe:0f:ee:be:17  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether c8:fe:0f:ee:be:17  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@quarkpi-ca2:~#
```

Figure 2.7-2 WLAN node

Next, use the nmcli command-line tool to connect to WIFI.

First, execute the following command to check the status of the network interface:

```
nmcli device status
```

Because no network connection is currently established, it is in a disconnected state.

```
root@quarkpi-ca2:~# nmcli device status
DEVICE              TYPE      STATE   CONNECTION
wlP4p65s0           wifi      已断开  --
wlan0               wifi      已断开  --
p2p-dev-wlP4p65s0   wifi-p2p  已断开  --
p2p-dev-wlan0       wifi-p2p  已断开  --
eth0                ethernet  不可用  --
dummy0              dummy     未托管  --
lo                  loopback  未托管  --
root@quarkpi-ca2:~#
```

Figure 2.7-3 Network interface status

Then, use the following nmcli command to obtain the WIFI scan results. Note that the nmcli network management will automatically scan WIFI. nmcli merely retrieves the scan results. Repeated acquisitions will obtain the results from the last scan. Wait for the next scan to start. Generally, it will automatically scan once within no more than 30 seconds.

```
nmcli device wifi
```

```
root@quarkpi-ca2:~# nmcli device wifi
IN-USE  BSSID              SSID                          MODE   CHAN  RATE
        02:D0:29:4F:50:C7  ALIENTEK-linux                Infra  40    270 Mbit/
        02:D0:29:4F:50:C3  ALIENTEK-linux                Infra  6     270 Mbit/
        B4:CF:E0:BB:3A:22  ChinaNet-ymyk                 Infra  2     130 Mbit/
        84:76:37:1D:8B:FD  --                            Infra  8     130 Mbit/
        84:76:37:1D:8B:79  --                            Infra  9     130 Mbit/
        3E:06:A7:5B:FB:F0  --                            Infra  10    270 Mbit/
        3C:06:A7:AB:FB:F0  RICH901                       Infra  10    270 Mbit/
        C4:36:55:4F:4F:1A  863egsw                       Infra  11    130 Mbit/
        84:76:37:1D:8B:FC  ALIENTEK-SYS                  Infra  8     130 Mbit/
        84:76:37:1D:8B:75  --                            Infra  9     130 Mbit/
        84:76:37:1D:8C:01  --                            Infra  8     130 Mbit/
        84:76:37:1D:8B:74  ALIENTEK-WX                   Infra  9     130 Mbit/
        2A:D0:EA:73:6E:14  DIRECT-HCDESKTOP-7A885MVmsJN  Infra  1     130 Mbit/
        4C:50:77:35:9C:88  ATK-HUAWEI                    Infra  6     130 Mbit/
        12:CE:51:48:F9:BC  DESKTOP-GASKALP 1177          Infra  36    270 Mbit/
        C8:EE:A6:3F:EF:88  wireless-3FEF88               Infra  10    130 Mbit/
        3E:06:A7:BB:FB:F0  --                            Infra  44    270 Mbit/
        84:76:37:FD:8B:78  --                            Infra  153   270 Mbit/
        84:76:37:1D:8B:7A  --                            Infra  153   270 Mbit/
        B4:CF:E0:BB:3A:21  ChinaNet-ymyk-5G              Infra  52    270 Mbit/
        84:76:37:1D:8B:78  ALIENTEK-WX_5G                Infra  153   270 Mbit/
        42:34:98:34:50:AA  ATK-HUAWEI                    Infra  6     270 Mbit/
        70:FD:45:BC:11:B6  301                           Infra  6     130 Mbit/
        3C:06:A7:AB:FB:F1  RICH901-5G                    Infra  44    270 Mbit/
        34:96:72:72:51:AC  C21                           Infra  1     405 Mbit/
        E2:FE:38:72:13:BB  ALIENTEK-7办公                Infra  11    270 Mbit/
        E6:06:4D:94:ED:D2  waterguy                      Infra  11    130 Mbit/
        18:9E:2C:FD:D7:BD  --                            Infra  11    130 Mbit/
        84:76:37:1D:8C:00  ALIENTEK-SYS_5G               Infra  149   270 Mbit/
        84:76:37:FD:8C:00  --                            Infra  149   270 Mbit/
        B8:08:D7:93:1A:84  606-2                         Infra  1     270 Mbit/
        84:76:37:1D:8C:02  --                            Infra  149   270 Mbit/
        18:9E:2C:FD:EF:1D  --                            Infra  6     130 Mbit/
        18:9E:2C:FD:EF:1C  ALIENTEK-XHYS                 Infra  6     130 Mbit/
        B4:CF:E0:BA:C4:2A  ChinaNet-9pdf                 Infra  9     130 Mbit/
        18:9E:2C:FD:D7:C1  --                            Infra  11    130 Mbit/
        0C:83:9A:72:C9:A0  广州利鸥                      Infra  11    130 Mbit/
        18:9E:2C:FD:D7:BC  ALIENTEK-7机房                Infra  11    130 Mbit/
        48:4C:29:61:A1:E4  Mark080                       Infra  5     130 Mbit/
        42:34:98:34:50:AF  --                            Infra  6     270 Mbit/
        70:37:8E:79:C3:41  神饮科技                      Infra  11    270 Mbit/
        90:86:9B:22:CD:F0  ChinaNet-psQW                 Infra  8     130 Mbit/
        18:9E:2C:FD:EF:22  --                            Infra  149   270 Mbit/
```

Figure 2.7-4 WIFI scan results

Then use the nmcli command to connect to WIFI. The command format is as follows, using double quotation marks to enclose the names of hotspots with spaces to prevent issues.

```
nmcli device wifi connect "WIFI name " password "WIFI password"
```

```
root@quarkpi-ca2:~# nmcli device wifi connect "ALIENTEK-linux" password "159[illegible]"
[ 2012.374664] IPv6: ADDRCONF(NETDEV_CHANGE): wlP4p65s0: link becomes ready
成功用 "wlP4p65s035549447-0881-4a04-8ac2-ef66c87ab48a" 激活了设备 ""。
root@quarkpi-ca2:~#
```

Figure 2.7-5 Connect to WIFI

Check the status of the network interface again, as shown in the following picture

```
nmcli device status
```

```
root@quarkpi-ca2:~# nmcli device status
DEVICE              TYPE      STATE   CONNECTION
wlP4p65s0           wifi      已连接  ALIENTEK-linux
wlan0               wifi      已断开  --
p2p-dev-wlP4p65s0   wifi-p2p  已断开  --
p2p-dev-wlan0       wifi-p2p  已断开  --
eth0                ethernet  不可用  --
dummy0              dummy     未托管  --
lo                  loopback  未托管  --
root@quarkpi-ca2:~#
```

Figure 2.7-6 Recheck the WLAN status

Use ifconfig to view the IP address of the WIFI network card. You can see that it can successfully obtain an IP address.

ifconfig

```
root@quarkpi-ca2:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 0e:90:cb:b3:a0:d2  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 100

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 1288  bytes 100392 (98.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1288  bytes 100392 (98.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlP4p65s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.3.89  netmask 255.255.255.0  broadcast 192.168.3.255
        inet6 fe80::7e81:62e3:beb1:a282  prefixlen 64  scopeid 0x20<link>
        ether ca:fe:0f:ee:be:17  txqueuelen 1000  (Ethernet)
        RX packets 177  bytes 23434 (22.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 206  bytes 15542 (15.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether c8:fe:0f:ee:be:17  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@quarkpi-ca2:~#
```

Figure 2.7-7 Check the IP address of the WLAN.

Using the ping command, it is possible to successfully access Baidu.

```
root@quarkpi-ca2:~# ping -I wlP4p65s0 www.baidu.com
PING www.a.shifen.com (103.235.46.96) from 192.168.3.89 wlP4p65s0: 56(84) bytes of data.
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=1 ttl=45 time=214 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=2 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=3 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=4 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=5 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=6 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=7 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=8 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=9 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=10 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=11 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=12 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=13 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=14 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=15 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=16 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=17 ttl=45 time=214 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=18 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=19 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=20 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=21 ttl=45 time=212 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=22 ttl=45 time=213 ms
64 bytes from 103.235.46.96 (103.235.46.96): icmp_seq=23 ttl=45 time=212 ms
```

Figure 2.7-8 The ping command test results for WLAN

If you want to disconnect the WIFI, reconnect to WIFI and delete the existing WIFI connection, you can use the following commands:

```
nmcli connection down "WIFI name"      # Disconnect from the WIFI connection
nmcli connection up " WIFI name"       # Reconnect to WIFI
nmcli connection delete " WIFI name"   # Delete WIFI connection
```

If testing is conducted on the desktop, simply click the network link icon on the desktop to connect to WIFI.

2.8 Bluetooth Test

By default, the factory system will automatically load the WIFI and Bluetooth drivers upon detecting the insertion of the RTL8852BE PCIE WIFI module. We can directly conduct the functional test.

Enter the following command to enter the bluetoothctl interactive mode.

```
sudo bluetoothctl
```

Execute the following instructions in sequence.

```
power on          # Turn on the power
agent on          # Turn on the power
discoverable on   # Enable detectability
scan on           # Enable scanning of devices
```

```
root@ATK-DLRK3568-Debian:~# bluetoothctl
Agent registered
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# agent on
Agent is already registered
[bluetooth]# discoverable on
Changing discoverable on succeeded
[bluetooth]# scan on
[bluetooth]# [  909.521190] Bluetooth: hu 00000000dc63c9a8 retransmitting 1 pkts
[  909.525808] rtk_btcoex: hci (periodic)inq start
Discovery started
[CHG] Controller 48:8F:4C:10:2E:BD Discovering: yes
[bluetooth]# [  909.588059] Bluetooth: Unknown advertising packet type: 0x100
[  909.588162] Bluetooth: hci0: advertising data len corrected
[NEW] Device 68:DB:B7:43:DF:D7 68-DB-B7-43-DF-D7
[bluetooth]# [  909.627639] Bluetooth: Unknown advertising packet type: 0x100
[  909.668153] Bluetooth: Unknown advertising packet type: 0x100
[NEW] Device 43:76:F4:B3:26:7E 43-76-F4-B3-26-7E
[NEW] Device 43:32:ED:80:11:D7 43-32-ED-80-11-D7
```

Once your own Bluetooth device is detected, input "scan off" to stop the scanning process.

```
scan off
```

As shown in the figure below, the author scanned the Bluetooth MAC address of the Android phone and, after confirmation, the pairing process could be initiated.

```
[bluetooth]# [  992.739140] rtk_btcoex: hci (periodic)inq cancel/exit
Discovery stopped
[CHG] Controller 48:8F:4C:10:2E:BD Discovering: no
[CHG] Device 55:EF:13:8A:AC:5C RSSI is nil
[CHG] Device D2:32:E0:5D:64:3F RSSI is nil
[CHG] Device 6C:B3:EC:CF:F8:1D TxPower is nil
[CHG] Device 6C:B3:EC:CF:F8:1D RSSI is nil
[CHG] Device 75:A5:9E:4A:4F:01 TxPower is nil
[CHG] Device 75:A5:9E:4A:4F:01 RSSI is nil
[CHG] Device 73:BE:5F:23:0A:E8 TxPower is nil
[CHG] Device 73:BE:5F:23:0A:E8 RSSI is nil
[CHG] Device E9:E6:5E:39:31:13 RSSI is nil
[CHG] Device 4C:D8:BC:5F:3E:4D TxPower is nil
[CHG] Device 4C:D8:BC:5F:3E:4D RSSI is nil
[CHG] Device 7E:AA:B1:1E:11:CE TxPower is nil
[CHG] Device 7E:AA:B1:1E:11:CE RSSI is nil
[CHG] Device E7:E8:E6:3F:18:81 RSSI is nil
[CHG] Device 0C:AE:B0:FB:5B:0E RSSI is nil
[CHG] Device 44:A9:D1:ED:86:82 RSSI is nil
[CHG] Device AC:D6:18:5C:C2:28 RSSI is nil
[CHG] Device 74:D2:85:88:BD:59 RSSI is nil
[CHG] Device 64:68:76:9D:A5:71 RSSI is nil
[CHG] Device 22:56:BD:A8:CB:63 RSSI is nil
[CHG] Device 5F:31:3F:31:78:9D RSSI is nil
[CHG] Device 53:DE:A9:F4:3A:10 TxPower is nil
[CHG] Device 53:DE:A9:F4:3A:10 RSSI is nil
[CHG] Device 56:D2:E0:53:93:43 TxPower is nil
[CHG] Device 56:D2:E0:53:93:43 RSSI is nil
[CHG] Device 5A:78:8A:55:04:7F TxPower is nil
[CHG] Device 5A:78:8A:55:04:7F RSSI is nil
[CHG] Device 7C:2A:DB:03:DF:77 RSSI is nil
[CHG] Device 00:13:EF:A0:04:F1 TxPower is nil
[CHG] Device 00:13:EF:A0:04:F1 RSSI is nil
[CHG] Device 6F:2D:97:F5:77:30 RSSI is nil
[CHG] Device 5C:D5:55:04:3A:B0 TxPower is nil
[CHG] Device 5C:D5:55:04:3A:B0 RSSI is nil
[CHG] Device 42:42:D1:55:64:5D TxPower is nil
[CHG] Device 42:42:D1:55:64:5D RSSI is nil
[CHG] Device 60:AB:67:65:93:E8 RSSI is nil
[CHG] Device 43:32:ED:80:11:D7 TxPower is nil
```

Figure 2.8-1 Scan result

Enter the following instructions to perform the pairing.

```
pair 60:AB:67:65:93:E8        # Please enter your own Bluetooth MAC address.
```

```
[bluetooth]# pair 60:AB:67:65:93:E8
Attempting to pair with 60:AB:67:65:93:E8
[bluetooth]# [ 1414.120919] Bluetooth: hu 00000000dc63c9a8 retransmitting 1 pkts
[ 1414.122829] rtk_btcoex: hci create connection, start paging
[ 1414.470207] rtk_btcoex: connected, handle 000b, status 0x00
[ 1414.470304] rtk_btcoex: Page success
[ 1414.500259] rtk_btcoex: io capability request
[CHG] Device 60:AB:67:65:93:E8 Connected: yes
Request confirmation
[agent] Confirm passkey 289260 (yes/no): yes
[蓝牙测试]# [ 1431.287034] rtk_btcoex: link key notify
[ 1431.327388] rtk_btcoex: l2cap op 2, len 16, out 1
[ 1431.327516] rtk_btcoex: TX l2cap conn req, hndl 0x000b, PSM 0x0001, scid 0x0040
[ 1431.327543] rtk_btcoex: PSM(0x0001) do not need parse
[ 1431.343600] rtk_btcoex: l2cap op 3, len 20, out 0
[ 1431.343816] rtk_btcoex: RX l2cap conn rsp, hndl 0x000b, dcid 0x0042, scid 0x0040, result 0x0000
[ 1431.352224] rtk_btcoex: l2cap op 2, len 16, out 0
[ 1431.352416] rtk_btcoex: RX l2cap conn req, hndl 0x000b, PSM 0x0001, scid 0x0043
[ 1431.352457] rtk_btcoex: PSM(0x0001) do not need parse
[ 1431.352546] rtk_btcoex: l2cap op 3, len 20, out 1
[ 1431.352702] rtk_btcoex: TX l2cap conn rsp, hndl 0x000b, dcid 0x0041, scid 0x0043, result 0x0000
[ 1431.385879] rtk_btcoex: l2cap op 6, len 16, out 0
[ 1431.386048] rtk_btcoex: RX l2cap disconn req, hndl 0x000b, dcid 0x0041, scid 0x0043
[ 1431.386087] rtk_btcoex: handle_l2cap_discon_req: handle 0x000b, dcid 0x0041, scid 0x0043, dir 0
[ 1431.394660] rtk_btcoex: l2cap op 2, len 16, out 0
[ 1431.394857] rtk_btcoex: RX l2cap conn req, hndl 0x000b, PSM 0x0001, scid 0x0044
[ 1431.394926] rtk_btcoex: PSM(0x0001) do not need parse
[ 1431.395036] rtk_btcoex: l2cap op 3, len 20, out 1
[ 1431.395372] rtk_btcoex: TX l2cap conn rsp, hndl 0x000b, dcid 0x0041, scid 0x0044, result 0x0000
[CHG] Device 60:AB:67:65:93:E8 Modalias: bluetooth:v038Fp1200d1436
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 60:AB:67:65:93:E8 UUIDs: 98b97136-36a2-11ea-8467-484d7e99a198
[CHG] Device 60:AB:67:65:93:E8 ServicesResolved: yes
[CHG] Device 60:AB:67:65:93:E8 Paired: yes
Pairing successful
[蓝牙测试]# [ 1431.447184] rtk_btcoex: l2cap op 6, len 16, out 0
[ 1431.447339] rtk_btcoex: RX l2cap disconn req, hndl 0x000b, dcid 0x0041, scid 0x0044
[ 1431.447363] rtk_btcoex: handle_l2cap_discon_req: handle 0x000b, dcid 0x0041, scid 0x0044, dir 0
[ 1433.990448] rtk_btcoex: l2cap op 6, len 16, out 1
[ 1433.990655] rtk_btcoex: TX l2cap disconn req, hndl 0x000b, dcid 0x0042, scid 0x0040
[ 1433.990693] rtk_btcoex: handle_l2cap_discon_req: handle 0x000b, dcid 0x0042, scid 0x0040, dir 1
[ 1436.200336] rtk_btcoex: HCI Disconnect, handle 000b, reason 0x13
[ 1436.269667] rtk_btcoex: disconn cmpl evt: status 0x00, handle 000b, reason 0x16
[ 1436.269766] rtk_btcoex: process disconn complete event.
[CHG] Device 60:AB:67:65:93:E8 ServicesResolved: no
[CHG] Device 60:AB:67:65:93:E8 Connected: no
[bluetooth]#
```

Input "yes"

The pairing code also pops up on the mobile device.

Enter the following instruction: Trust MAC.

```
trust 60:AB:67:65:93:E8        # Please enter your own Bluetooth MAC address.
```

```
[bluetooth]# trust 60:AB:67:65:93:E8
[CHG] Device 60:AB:67:65:93:E8 Trusted: yes
Changing 60:AB:67:65:93:E8 trust succeeded
[bluetooth]#
```

If the test is conducted on the desktop, simply click on the Bluetooth icon and use the graphical interface to perform the test.

2.9 CPU & GPU Temperature

Use the CAT command to check the CPU and GPU temperatures. The user layer provides two interfaces, and the displayed values are in millidegrees Celsius, as shown in the following figure.

```
cat /sys/class/thermal/thermal_zone0/temp    # Check the CPU temperature
cat /sys/class/thermal/thermal_zone1/temp    # Check the GPU temperature
```

2.10 CPU Frequency

View the current running CPU frequency

```
cat /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
```