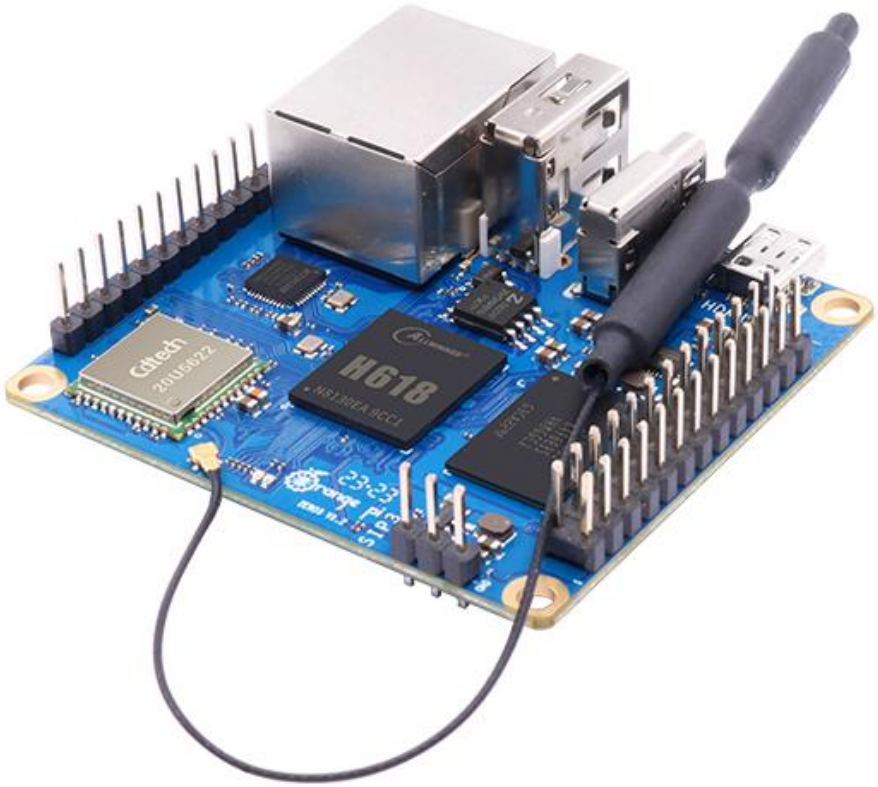
Orange Pi Zero 3

User Manual



Catalog

1. Basic features of Orange Pi Zero 3

1.1. What is Orange Pi Zero 3

Orange Pi is an open source single-board card computer, a new generation of arm64 development board, which can run Android TV 12, Ubuntu and Debian and other operating systems. Orange Pi Zero 3 uses Allwinner H618 system-on-a-chip and has 1GB or 1.5GB or 2GB or 4GB LPDDR4 memory.

1.2. Purpose of Orange Pi Zero 3

We can use it to achieve:

- A small Linux desktop computer
- A small Linux web server
- Install the Klipper host computer to control the 3D printer
- Android TV box

Of course, there are more functions. Relying on a powerful ecosystem and a variety of expansion accessories, Orange Pi can help users easily realize the delivery from idea to prototype to mass production. It is a maker, dreamer, hobby The ideal creative platform for readers.

1.3. Who is Orange Pi Zero 3 designed for?

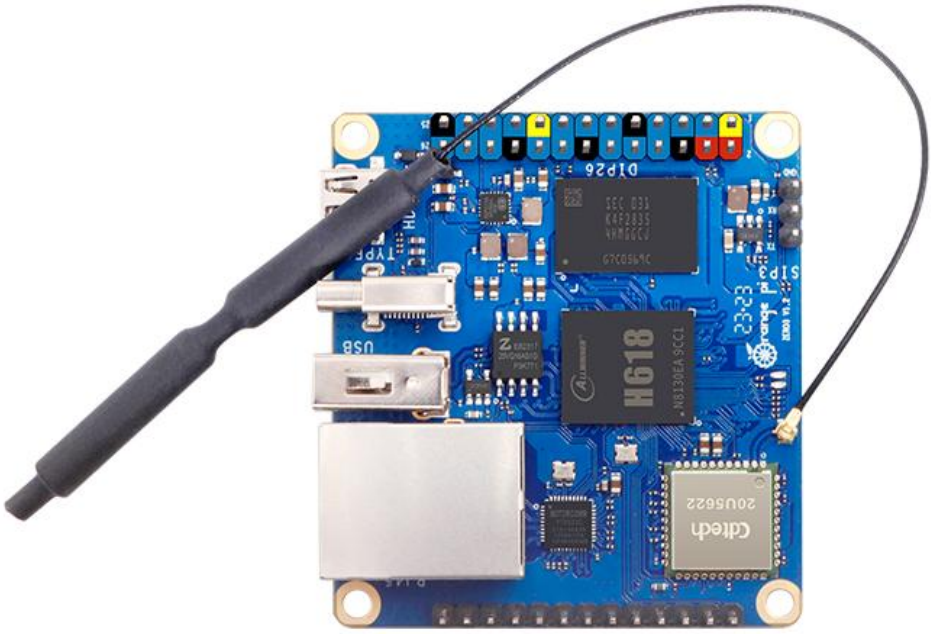
The Orange Pi development board is not just a consumer product, but is designed for anyone who wants to use technology to create and innovate. It's a simple, fun, and useful tool you can use to shape the world around you.

1. 4. Hardware Features of Orange Pi Zero 3

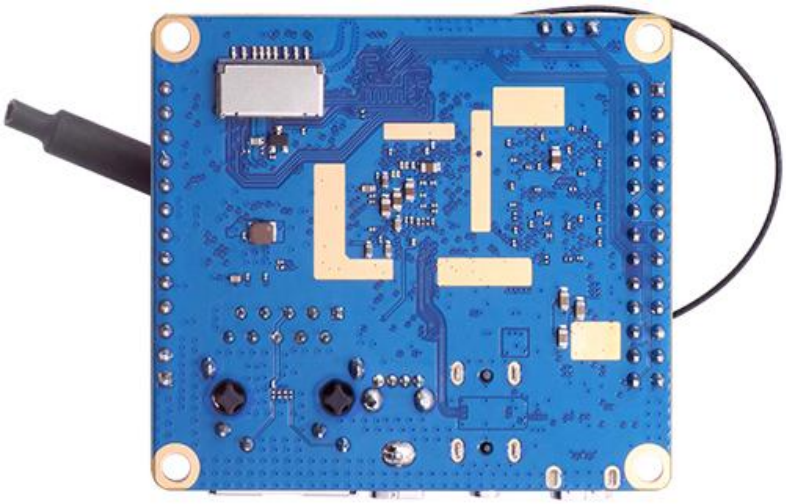
Introduction to hardware features	
CPU	Allwinner H618 quad-core 64-bit 1.5GHz high-performance Cortex-A53 processor
GPU	Mali G31 MP2 Supports OpenGL ES 1.0/2.0/3.2、OpenCL 2.0
Memory	1GB/1.5GB/2GB/4GB LPDDR4 (shared with GPU)
Onboard storage	micro SD card slot, 16MB SPI Flash
Ethernet	Support 10/100M/1000M Ethernet
WIFI+Bluetooth	• AW859A chip, support IEEE 802.11 a/b/g/n/ac, BT5.0
Video Output	• Micro HDMI 2.0a • TV CVBS output, support PAL/NTSC (via 13pin expansion board)
Audio output	• Micro HDMI output • 3.5mm audio port (via 13pin expansion board)
Power Supply	USB Type C interface input
USB 2.0 port	3* USB 2.0 HOST (two of them are through 13pin expansion board)
26pin connector	With I2Cx1, SPIx1, UARTx1 and multiple GPIO ports
13pin connector	With USB 2.0 HOSTx2, TV-OUT, LINE OUT, IR-RX, and 3 GPIO ports
Debug serial port	UART-TX, UART-RX and GND
LED light	Power light and status light
Infrared receiver	Support infrared remote control (via 13pin expansion board)
Supported OS	Android12 TV, Ubuntu, Debian, etc.
Introduction of Appearance Specifications	
Product Size	85mm×56mm
Weight	30g

1.5. Top view and bottom view of Orange Pi Zero 3

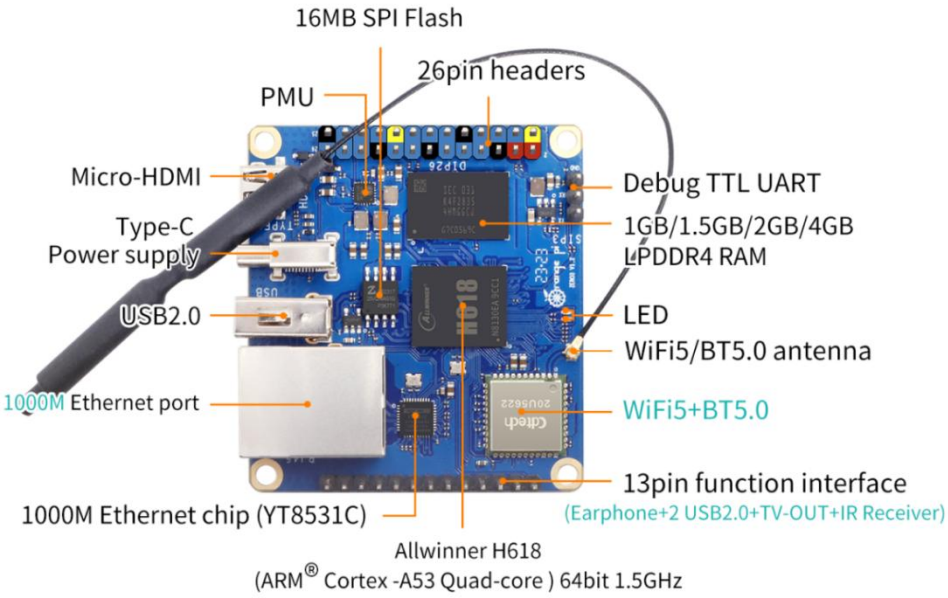
Top view：



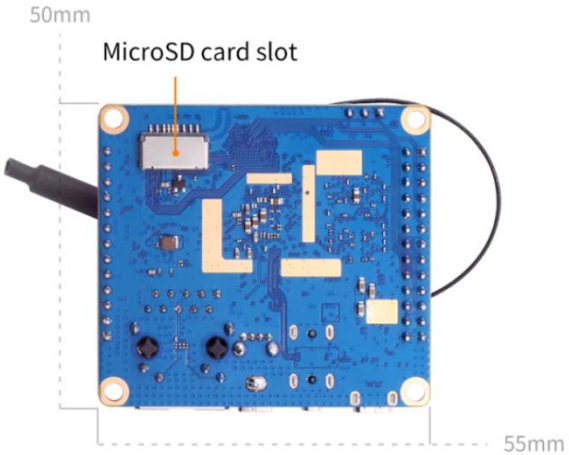
Bottom view：



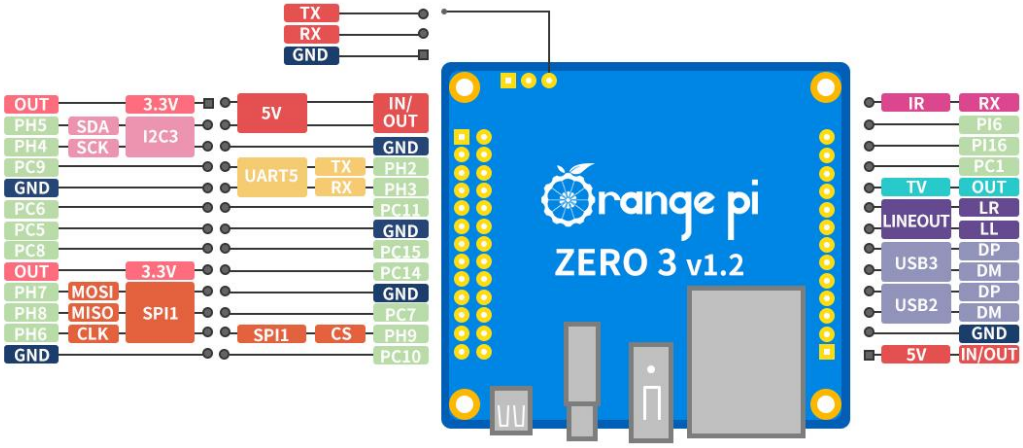
1. 6. Interface details of Orange Pi Zero 3



Top View



Bottom View



The diameters of the four positioning holes are all 3.0mm.

2. Introduction to the use of the development board

2. 1. Prepare the required accessories

1) Micro SD card, a high-speed SanDisk card of class 10 or above with a minimum capacity of 8GB

Using other brands of micro SD cards (non-SanDisk micro SD cards), as shown in the picture below (including but not limited to these cards), some friends have reported that there will be problems during the system startup process, such as the system is stuck halfway through startup, Or the reboot command cannot be used normally, and it was finally resolved after changing the SanDisk micro SD card. So if you use a non-SanDisk micro SD card and find that there is a problem with the system startup or use, please replace the SanDisk micro SD card and then test.

Current feedback is that there are some micro SD cards that have problems starting on Orange Pi Zero 3

In addition, the micro SD card that can be used normally on other types of development boards does not guarantee that it can be started normally on Orange Pi Zero 3, please pay special attention to this point.

2) Micro SD card reader, used to read and write micro SD card

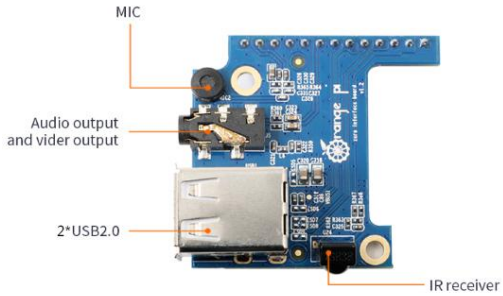
3) Micro HDMI to HDMI cable, used to connect the development board to an HDMI monitor or TV for display

Note, please do not use the relatively wide Micro HDMI adapter as shown in the figure below, because the distance between the Micro HDMI interface of the development board and the Type-C power interface is relatively small, it may cause that the two cannot be inserted into the development board at the same time. plate.

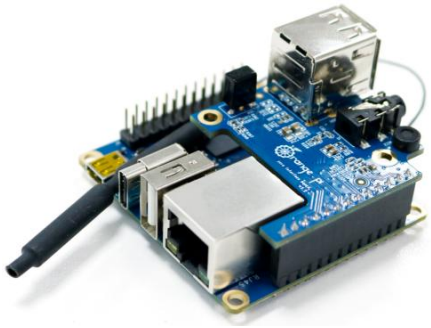
4) Power supply, if you have a 5V/2A or 5V/3A power head, you only need to prepare a USB-to-Type C interface data cable as shown in the picture on the left below, and you can also use a cable similar to the picture on the right below A 5V/2A or 5V/3A high-quality USB Typc C interface power adapter integrated with the power head.

5) 13pin expansion board

a. The physical object of the expansion board is as follows



b. The way to insert the expansion board into the development board is as follows, remember not to insert it backwards

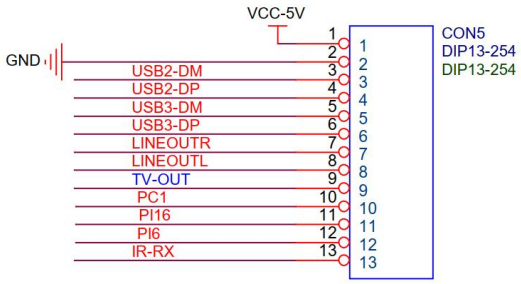


c. The 13pin pin header on the Orange Pi Zero 3 development board can be connected to the expansion board to expand the functions that are not on the development board. The functions that the expansion board can use are as follows

1	Microphone (Mic)	**No support, no support, no support! ! !**

		The 13pin expansion board is a general-purpose expansion board, which is suitable for various development boards of Orange Pi, but the 13pin interface of Orange Pi Zero3 has no Mic function, so although there is a Mic on the 13pin expansion board, it is on the Orange Pi Zero 3. Unusable, the 13pin expansion board is mainly used to expand functions other than Mic on Orange Pi Zero 3.
2	Analog audio and video output interface	Supported, it can be used to connect headphones to play music, or connect to TV through AV cable to output analog audio and video signals **(Android system only).**
3	USB 2.0 Host x 2	Support, used to connect USB keyboard, mouse and USB storage device.
4	Infrared receiving function	Support, Android system can be controlled by infrared remote control.

d. The schematic diagram of the 13pin header of the Orange Pi Zero 3 development board is shown below



6) The mouse and keyboard of the USB interface, as long as the mouse and keyboard of the standard USB interface are acceptable, the mouse and keyboard can be used to control the Orange Pi development board

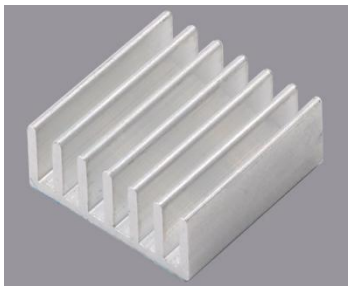
7) Infrared remote control, mainly used to control Android TV system

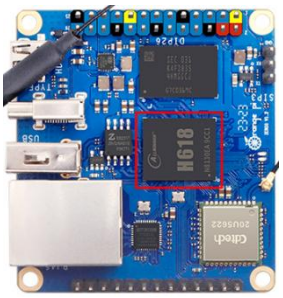
Note that the remote control of the air conditioner or the remote control of the TV cannot control the Orange Pi development board. By default, only the remote control provided by Orange Pi can.

8) 100M or 1000M network cable, used to connect the development board to the Internet

9) AV video cable, if you want to display video through the AV interface instead of the HDMI interface, then you need to connect the development board to the TV through the AV video cable

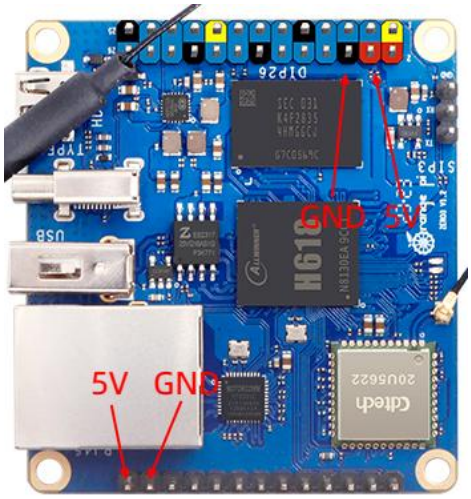
10) Heat sink, if you are worried that the temperature of the development board is too high, you can add a heat sink, and the heat sink can be pasted on the H618 chip





11) 5V cooling fan, as shown in the figure below, both 5V and GND pins on the 26pin and 13pin interfaces of the development board can be connected to the cooling fan. The spacing between the 26pin and 13pin headers is 2.54mm. Refer to this Specifications can be purchased.

Note that the 5V pin can be used directly after the development board is plugged in without other settings, and the output voltage of the 5V pin cannot be adjusted or turned off by software.



12) Matching shell **(pictures to be added)**

13) USB to TTL module and DuPont line, when using the serial port debugging function, need USB to TTL module and DuPont line to connect the development board and computer

Note that the TTL level used by the development board is 3.3v. In addition to the USB to TTL module shown in the above figure, other similar 3.3v USB to TTL modules are generally available.

14) X64 computer with Ubuntu and Windows operating systems installed

1	Ubuntu22.04 PC	Optional, used to compile Android and Linux source code
2	Windows PC	For burning Android and Linux images

2.2. Download the image of the development board and related materials

1) The website for downloading the English version is

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-Zero-3.html

2) The information mainly includes

a. **Android source code：** Save on Google Drive
b. **Linux source code：** Save on Github
c. **Android source code：** Save on Google Drive
d. **Ubuntu source code：** Save on Google Drive
e. **Debian source code：** Save on Google Drive
f. **User Manual and Schematic：** Chip-related data sheets will also be placed here
g. **Official tool：** It mainly includes the software that needs to be used during the use of the development board

2.3. Method of burning Linux image to micro SD card based on Windows PC

Note that the Linux image mentioned here specifically refers to the image of

Linux distributions such as Debian or Ubuntu downloaded from the Orange Pi data download page.

2. 3. 1. How to use balenaEtcher to burn Linux image

1) First prepare a micro SD card with a capacity of 8GB or more. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

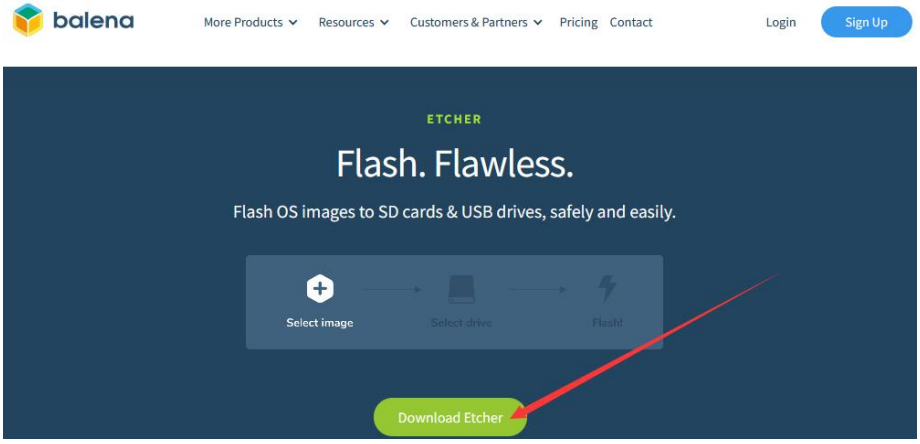
2) Then use the card reader to insert the micro SD card into the computer

3) Download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with ".img" is the image file of the operating system. The size is generally more than 1GB

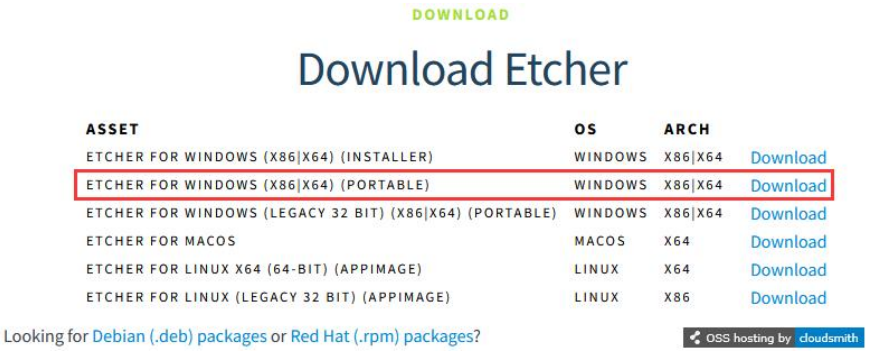
4) Then download the burning software of Linux image——**balenaEtcher,** the download address is

https://www.balena.io/etcher/

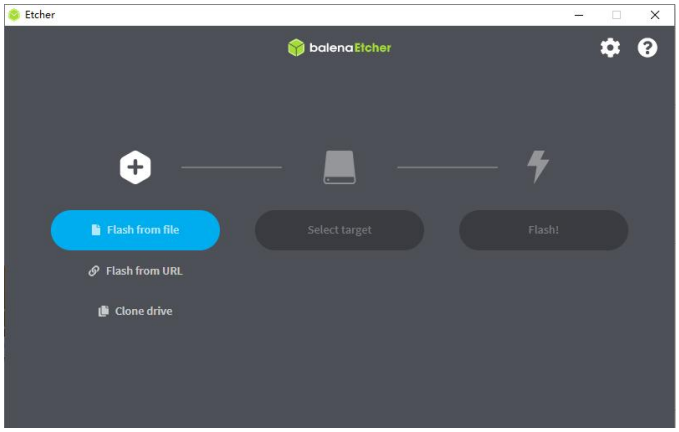
5) After entering the balenaEtcher download page, click the green download button to jump to the place where the software is downloaded



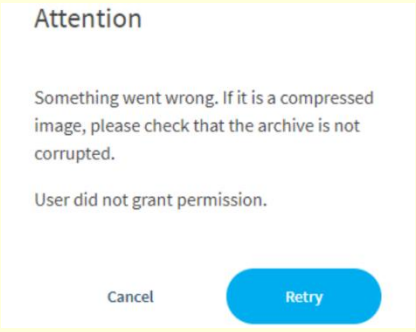
6) Then you can choose to download the Portable version of balenaEtcher software. The Portable version does not need to be installed, and you can use it by double-clicking to open it



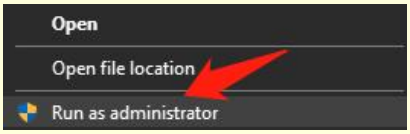
7) If the downloaded version of balenaEtcher needs to be installed, please install it before using it. If you downloaded the Portable version of balenaEtcher, just double-click to open it. The opened balenaEtcher interface is shown in the figure below



When opening balenaEtcher, if the following error is prompted:



Please select balenaEtcher, right-click, and select Run as administrator.

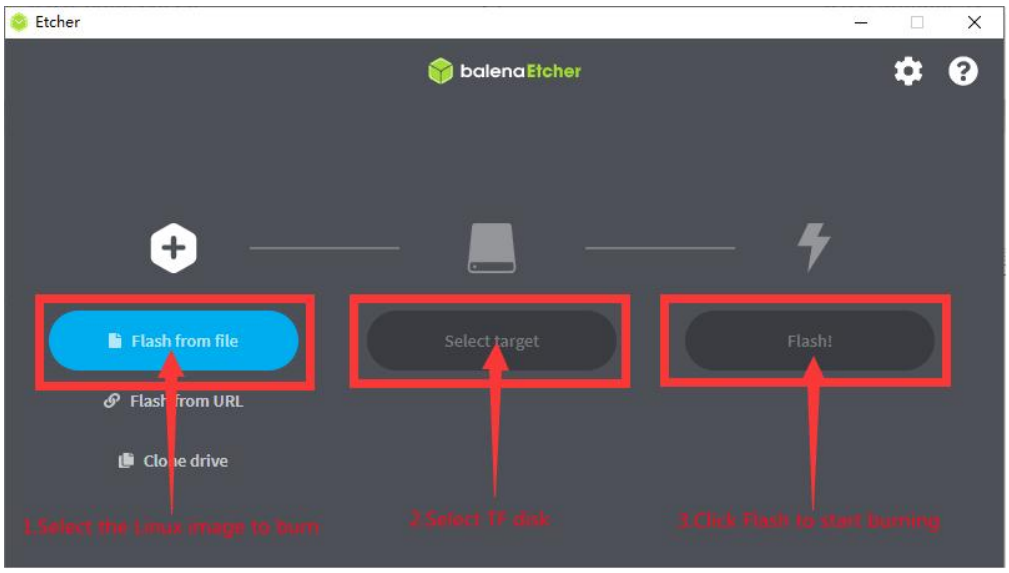


8) The specific steps to use balenaEtcher to burn the Linux image are as follows

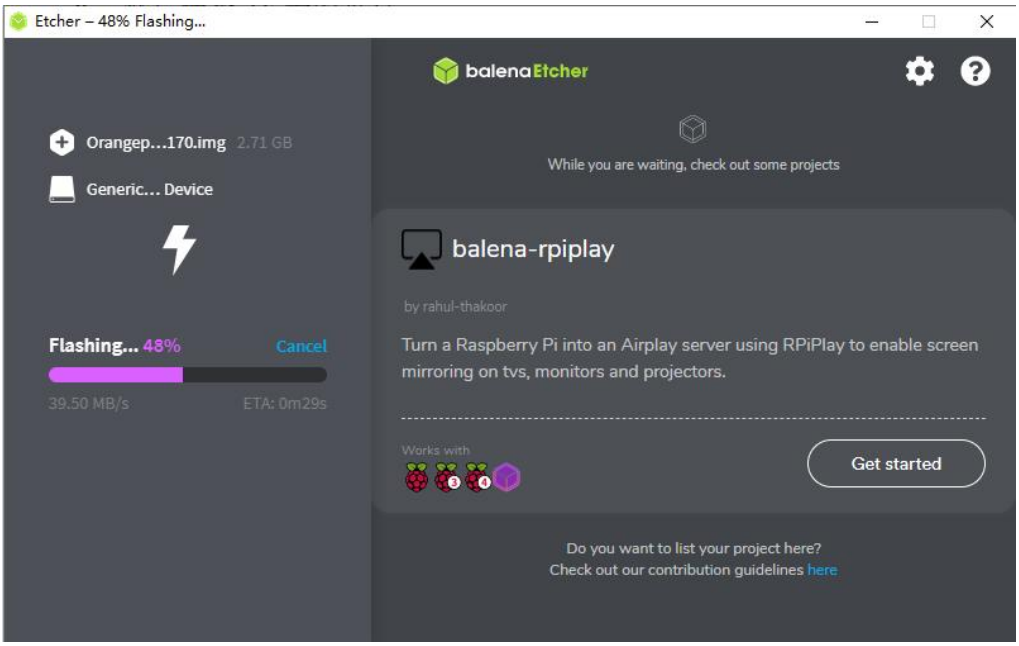
a. First select the path of the Linux image file to be burned

b. Then select the drive letter of the micro SD card

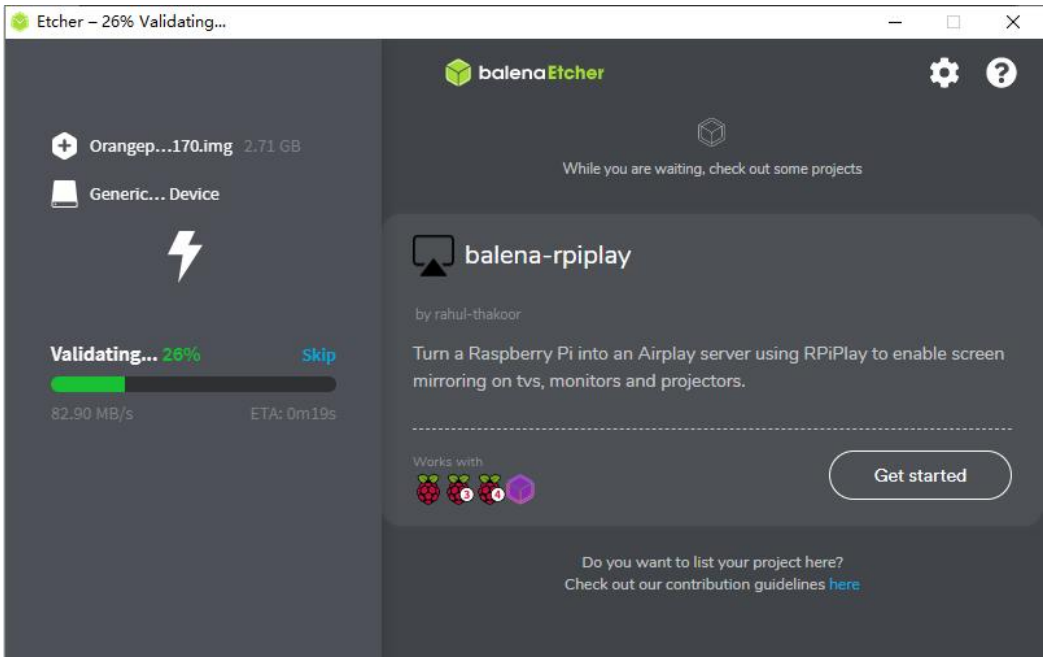
c. Finally click Flash to start burning the Linux image to the micro SD card



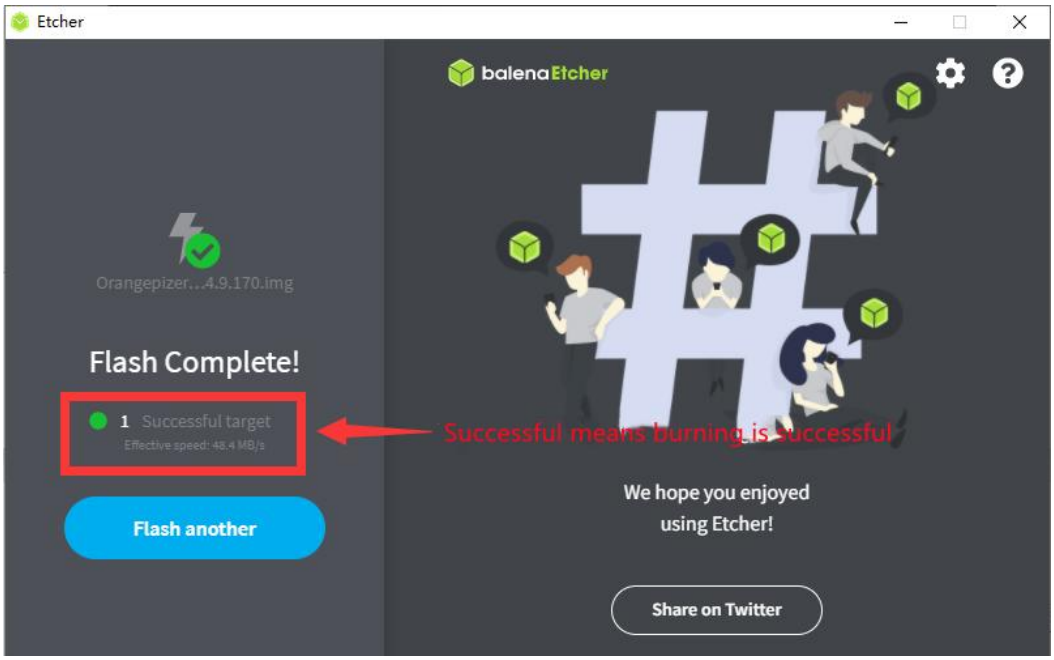
9) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the micro SD card



10) After burning the Linux image, balenaEtcher will also verify the image burned to the micro SD card by default to ensure that there is no problem in the burning process. As shown in the figure below, a green progress bar indicates that the image has been burnt, and balenaEtcher is verifying the burnt image.



11) After successful burning, the display interface of balenaEtcher is as shown in the figure below. If the green indicator icon is displayed, it means that the image burning is successful. At this time, you can exit balenaEtcher, and then pull out the micro SD card and insert it into the micro SD card slot of the development board. .



2. 3. 2. How to use Win32Diskimager to burn Linux image

1) First prepare a micro SD card with a capacity of 8GB or more. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

2) Then use the card reader to insert the micro SD card into the computer

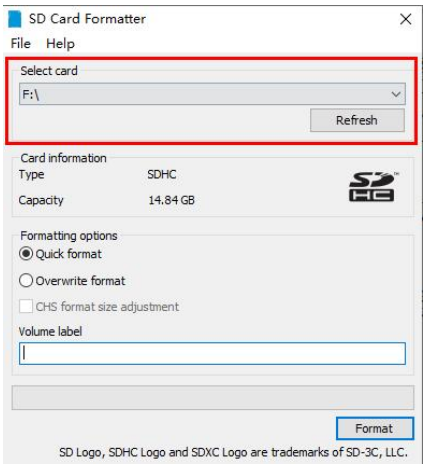
3) Then format the micro SD card

a. **SD Card Formatter** can be used to format the micro SD card. The download link is

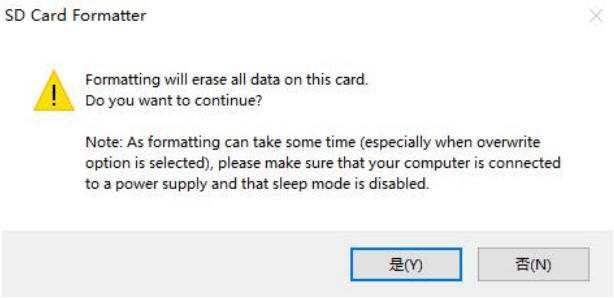
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

b. After downloading, unzip and install directly, and then open the software

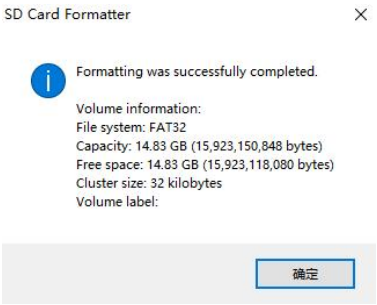
c. If only a micro SD card is inserted into the computer, the drive letter of the micro SD card will be displayed in the " **Select card** " column. If multiple USB storage devices are inserted into the computer, you can select the corresponding drive letter of the micro SD card through the drop-down box



d. hen click " **Format** ", a warning box will pop up before formatting, and formatting will start after selecting "Yes (Y)"



e. After formatting the micro SD card, the message shown in the figure below will pop up, click OK



4) Download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system. The size is generally more than 1GB

5) Use **Win32Diskimager** to burn the Linux image to the micro SD card

a. The download page of Win32Diskimager is

http://sourceforge.net/projects/win32diskimager/files/Archive/

b. After downloading, install it directly. The interface of Win32Diskimager is as follows

a) First select the path of the image file

b) Then confirm that the drive letter of the micro SD card is consistent with that displayed in the "**Device**" column

c) Finally click "**Write**" to start burning

c. After the image writing is completed, click the "**Exit**" button to exit, and then you can pull out the micro SD card and insert it into the development board to start

2. 4. Method of burning Linux image to micro SD card based on Ubuntu PC

Note that the Linux image mentioned here specifically refers to the image of Linux distributions such as Debian or Ubuntu downloaded from the Orange Pi data download page, and the Ubuntu PC refers to the personal computer with the Ubuntu system installed.

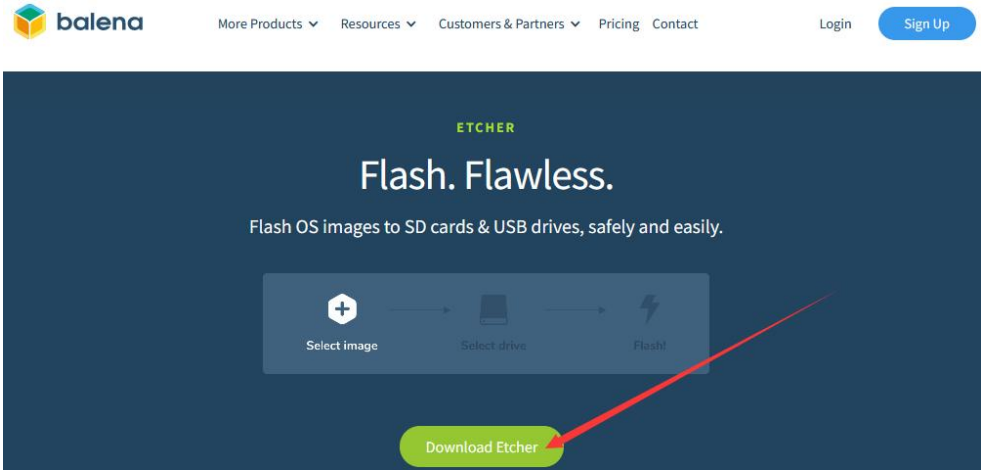
1) First prepare a micro SD card with a capacity of 8GB or more. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

2) Then use the card reader to insert the micro SD card into the computer

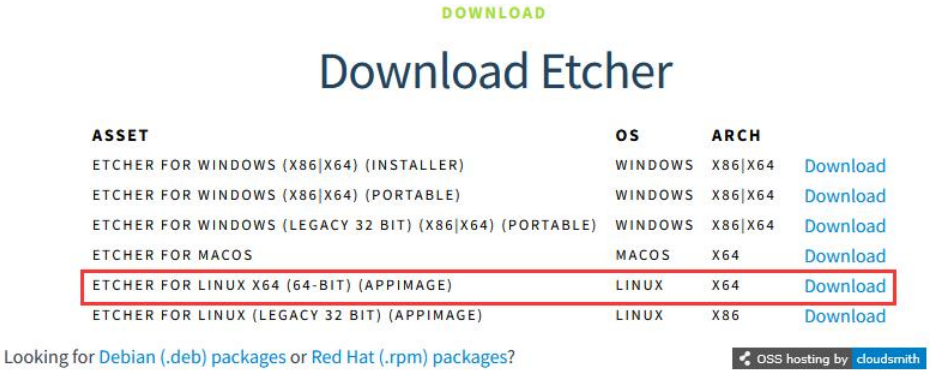
3) Download the balenaEtcher software, the download address is

https://www.balena.io/etcher/

4) After entering the balenaEtcher download page, click the green download button to jump to the place where the software is downloaded



5) Then choose to download the Linux version of the software



6) Download the image file compression package of the Linux operating system that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system. The size is generally above 1GB. The decompression command for the compressed package ending in 7z is as follows:

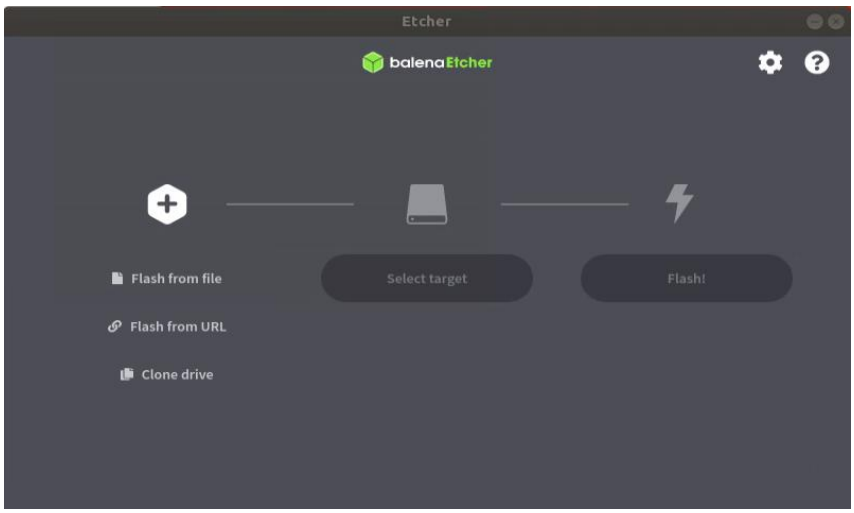
```
test@test:~$ 7z x orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.7z
```

```
test@test:~$ ls orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.*
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.7z
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.sha      # checksum file
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.img      # mirror file
```

7) After decompressing the image, you can first use the **sha256sum -c *.sha** command to calculate whether the checksum is correct. If the prompt is successful, it means that the downloaded image is correct, and you can safely burn it to the micro SD card. **If it prompts that the checksum does not match**, it means there is a problem with the downloaded image, please try to download again

```
test@test:~$ sha256sum -c *.sha
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.img: success
```

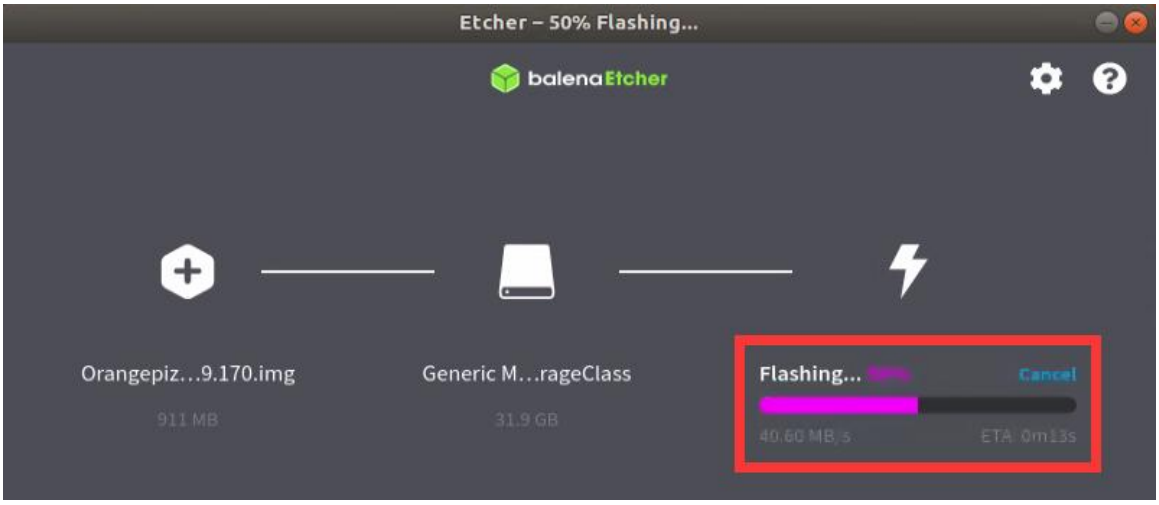
8) Then double-click **balenaEtcher-1.14.3-x64.AppImage** on the graphical interface of Ubuntu PC to open balenaEtcher (no installation required), and the interface after balenaEtcher is opened is displayed as shown in the figure below



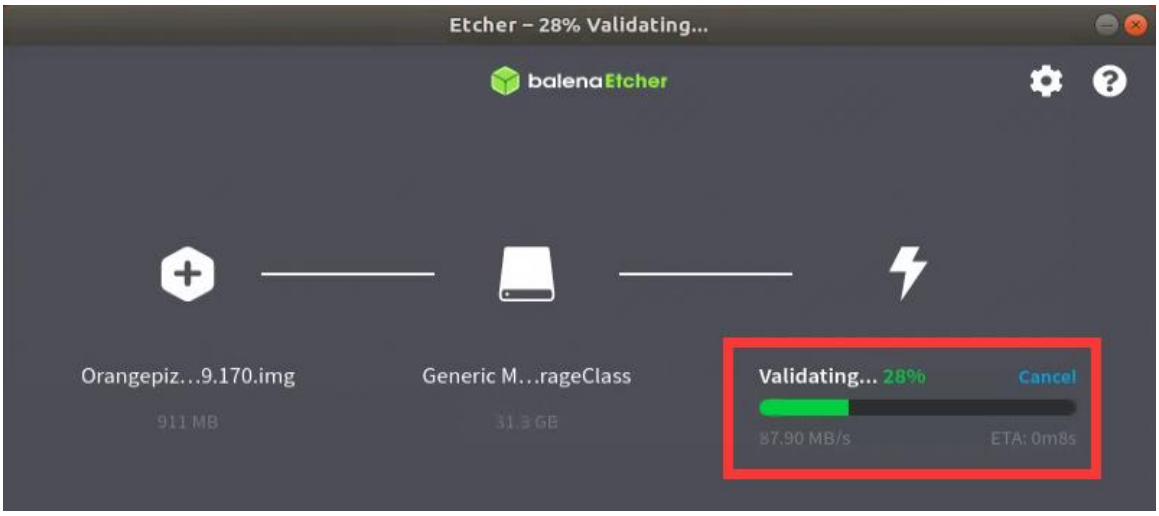
9) The specific steps to use balenaEtcher to burn the Linux image are as follows

a. First select the path of the Linux image file to be burned
b. Then select the drive letter of the micro SD card
c. Finally click Flash to start burning the Linux image to the micro SD card

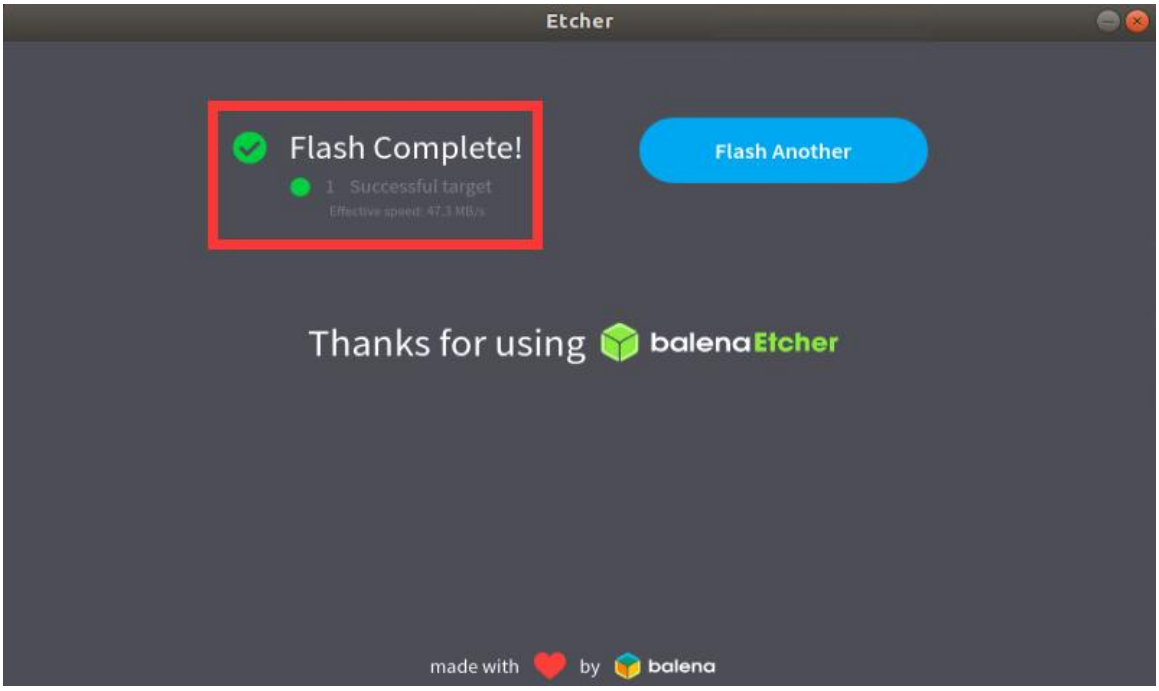
10) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the micro SD card



11)After burning the Linux image, balenaEtcher will also verify the image burned into the micro SD card by default to ensure that there is no problem in the burning process. As shown in the figure below, a green progress bar indicates that the image has been burnt, and balenaEtcher is verifying the burnt image



12) After successful burning, the display interface of balenaEtcher is as shown in the figure below. If a green indicator icon is displayed, it means that the image burning is successful. At this time, you can exit balenaEtcher, and then pull out the micro SD card and insert it into the micro SD card slot of the development board for use



2. 5. How to burn Android image to micro SD card

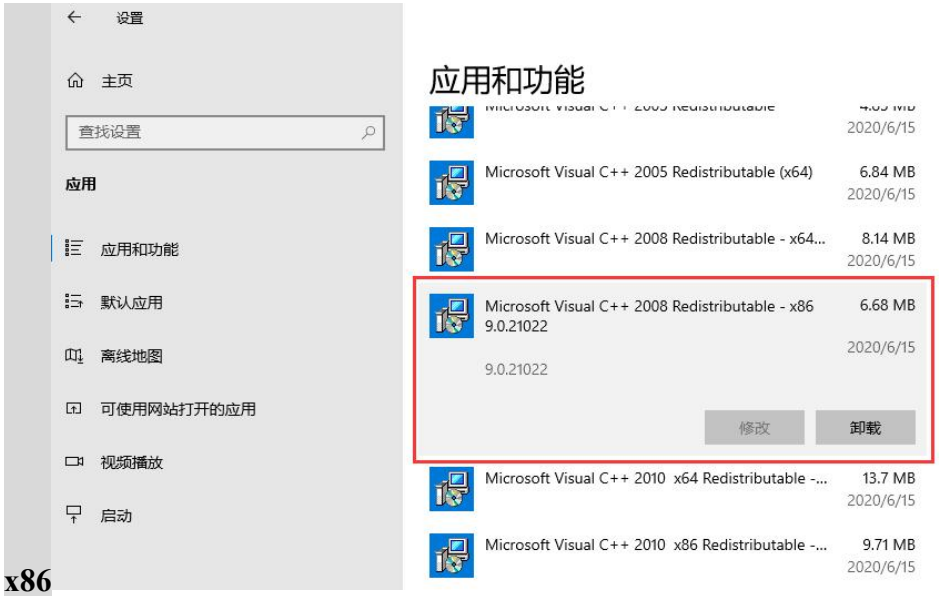
The Android image of the development board can only be burned into the micro

SD card under the Windows platform using the PhoenixCard software, and the version of the PhoenixCard software must be PhonixCard-4.2.8.

Please do not use software for burning Linux images, such as Win32Diskimager or balenaEtcher, to burn Android images.

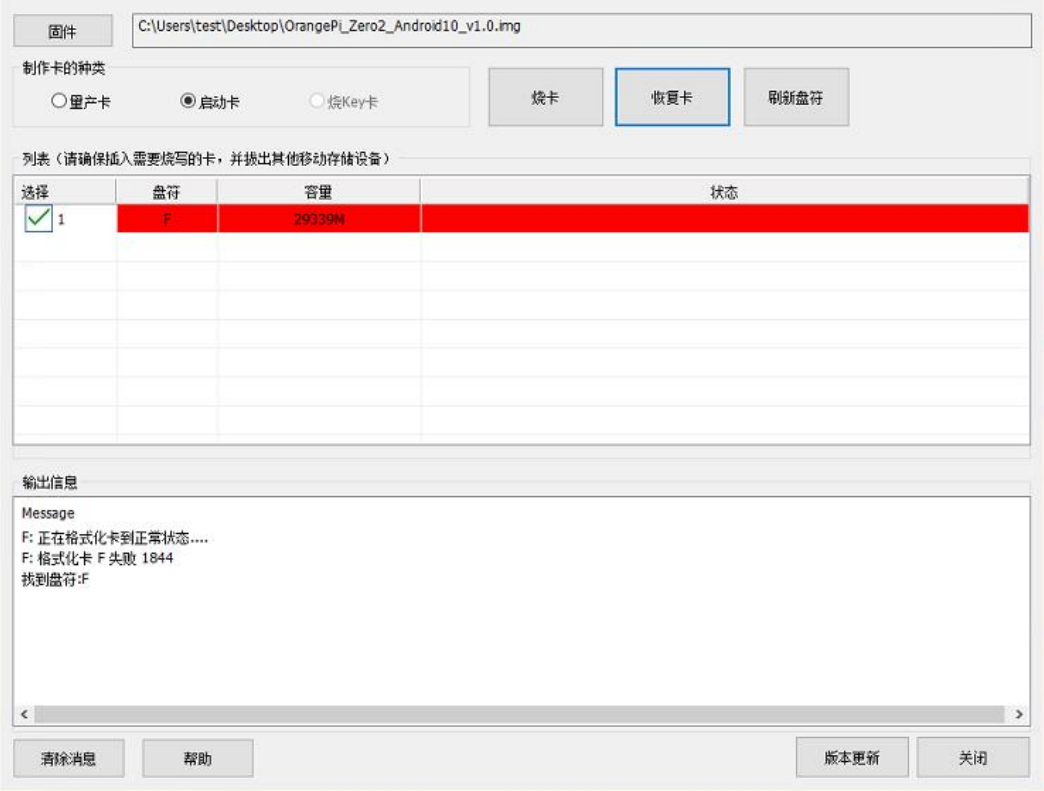
In addition, the PhoenixCard software does not have versions for Linux and Mac platforms, so it is impossible to burn Android images to micro SD cards under Linux and Mac platforms.

1) First, please make sure that the Windows system has installed **Microsoft Visual C++ 2008 Redistrbutable -**

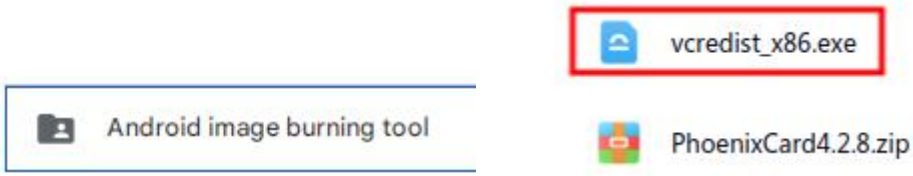


x86

2) If not installed **Microsoft Visual C++ 2008 Redistrbutable - x86**, Using PhoenixCard to format micro SD card or burn Android image will prompt the following error



3) **Microsoft Visual C++ 2008 Redistrbutable - x86** the installation package can be downloaded from the **official tool** of Orange Pi Zero 3, or you can go to the **official website of Microsoft** to download

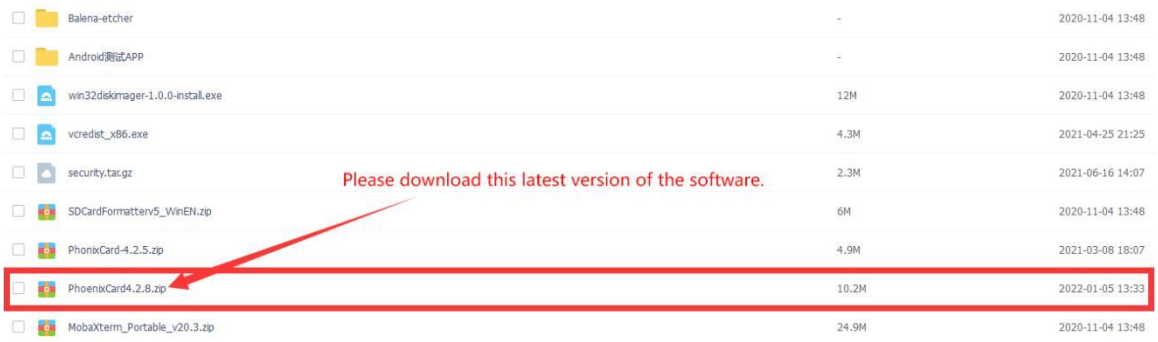


4) Then prepare a micro SD card with 8GB or larger capacity. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

5) Then use the card reader to insert the micro SD card into the computer

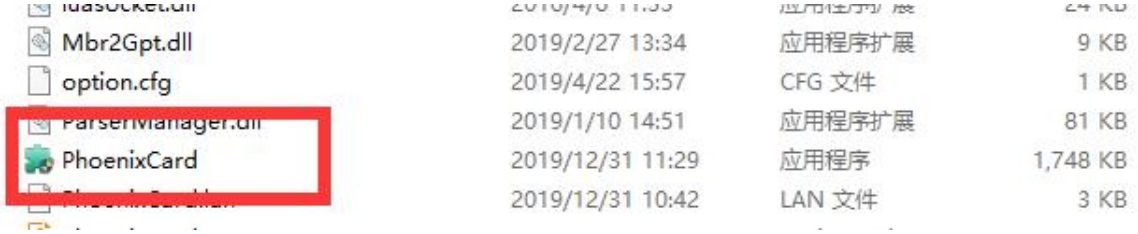
6) Download the Android image and PhoenixCard programming tool from the **Orange Pi data download page**. **Please ensure that the version of the PhonenixCrad tool is PhonixCard-4.2.8.** Please do not use the PhonixCard software version lower than 4.2.8

to burn the Android image. There may be problems with the Android image flashed by this version of the PhonixCard tool



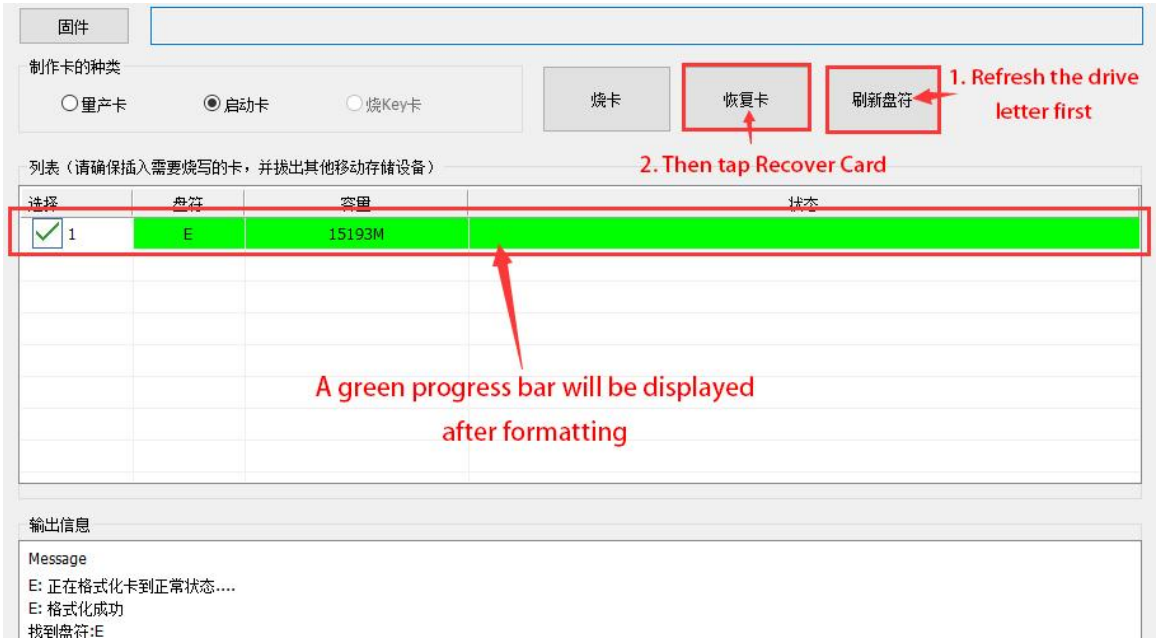
7) Then use the decompression software to decompress the compressed package of the downloaded Android image. Among the decompressed files, the file ending with ".img" is the Android image file, and the size is more than 1GB.

8) Then use decompression software to decompress **PhonixCard4.2.8.zip**, this software does not need to be installed, just find PhoenixCard in the decompressed folder and open it



9) After opening PhoenixCard, if the micro SD card is recognized normally, the drive letter and capacity of the micro SD card will be displayed in the middle list. **Please make sure that the displayed drive letter is consistent with the drive letter of the micro SD card you want to burn**. If there is no display, you can try to unplug the micro SD card, or click the "**Refresh Drive Letter**" button in PhoenixCard

10) After confirming the drive letter, format the micro SD card first, and click the "**Recover Card**" button in PhoenixCard (if the "**Recover Card**" button is gray and cannot be pressed, you can first click the "**Refresh Drive Letter**" button)



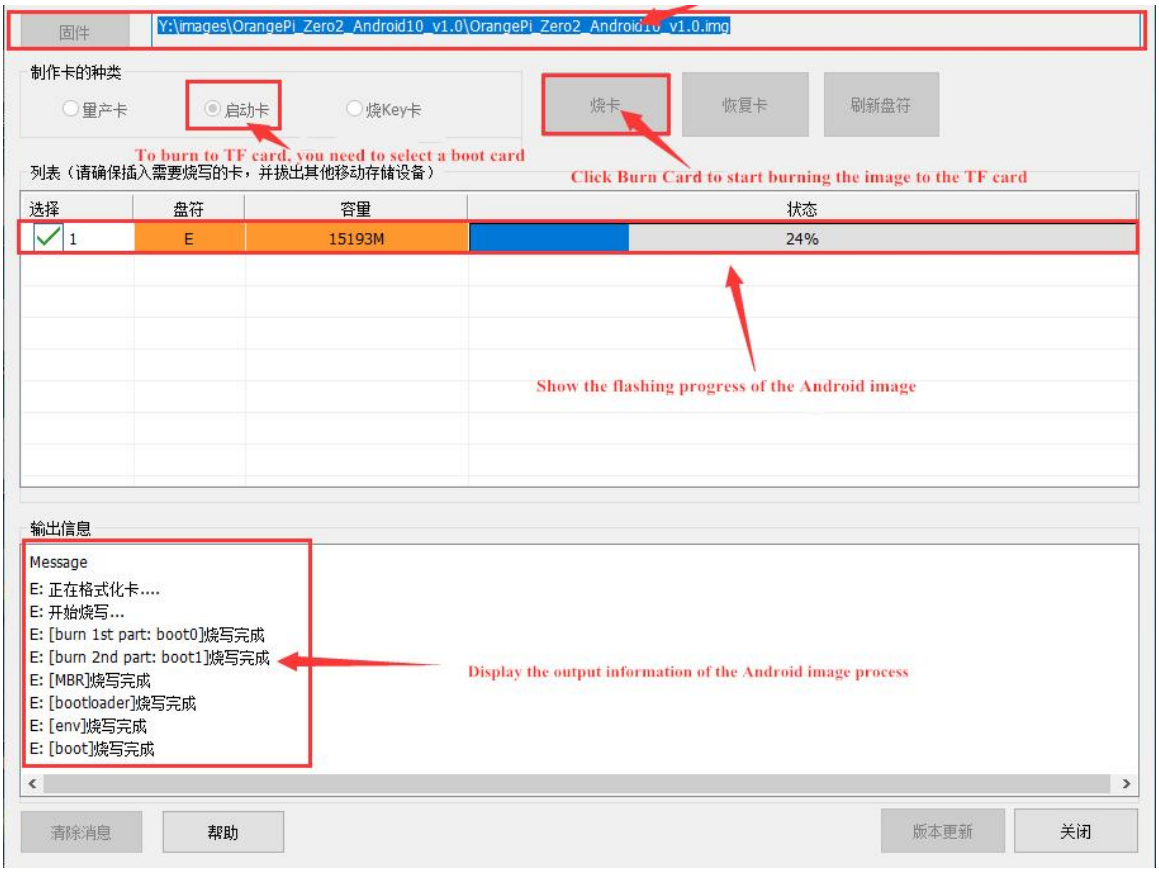
If there is a problem with formatting, please try to unplug and insert the micro SD card and then test again. If the problem still exists after re-plugging and inserting the micro SD card, you can restart the Windows computer or try another computer.

11) Then start to write the Android image into the micro SD card

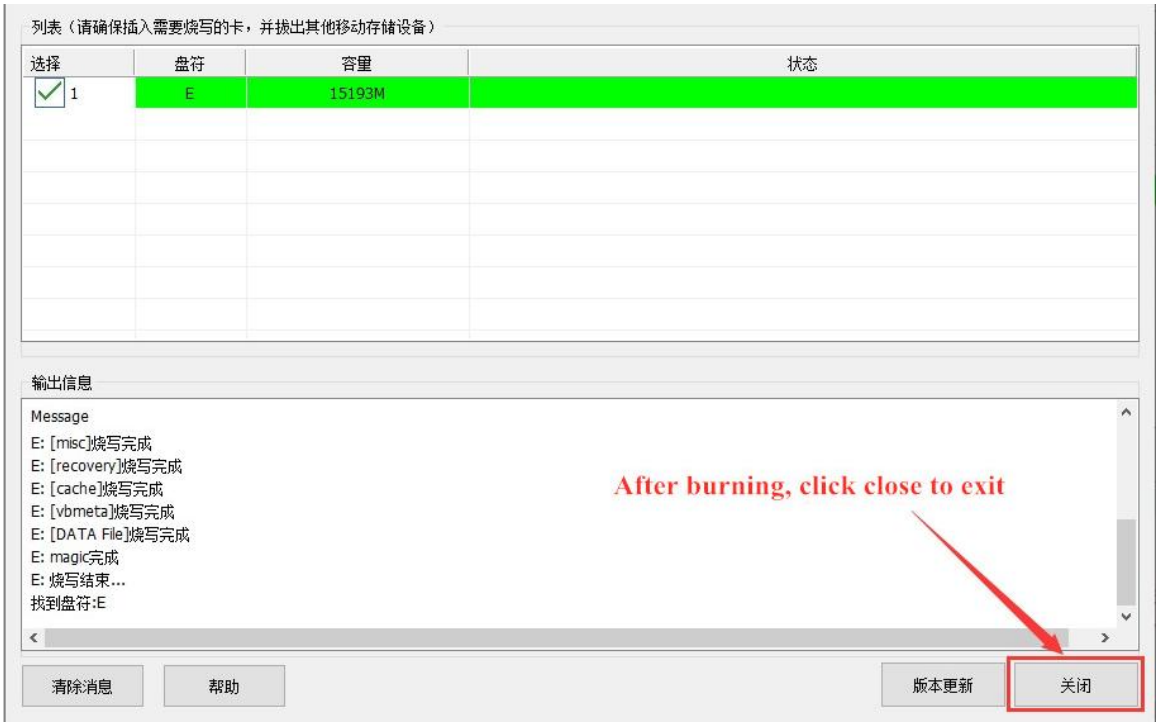
a. First select the path of the Android image in the "**Firmware**" column

b. Select "**Activation Card**" in "Card Type"

c. Then click the "**burn card**" button to start burning



12) After burning, the display of PhoenixCard is as shown in the figure below. At this time, click the "**Close**" button to exit PhoenixCard, and then you can pull out the micro SD card from the computer and insert it into the development board to start



After burning the Android system, the micro SD card can only see a 128 MB partition in Windows, and the displayed partition is as shown in the figure below (some computers may pop up more than 20 disk partitions, but only the 128 MB partition can be opened. partition), please note that this is normal, please do not think that the micro SD card is burned out. The reason for this is that the Android system has a total of more than 20 partitions, but most of them cannot be recognized normally in the Windows system. At this point, please safely unplug the micro SD card and insert it into the development board to start it.

After the Android system starts, use the following command to see the twenty or so partitions in the micro SD card:

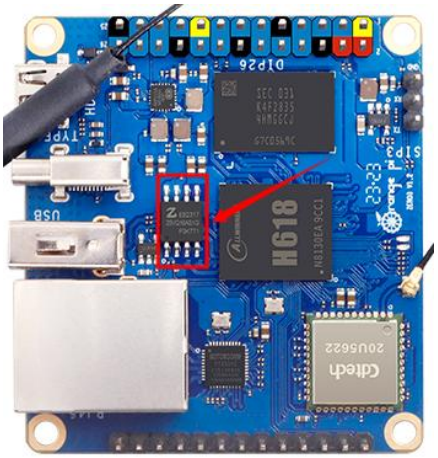
```
console:/ # ls /dev/block/mmcblk0*
/dev/block/mmcblk0      /dev/block/mmcblk0p17  /dev/block/mmcblk0p25
/dev/block/mmcblk0p1    /dev/block/mmcblk0p18  /dev/block/mmcblk0p3
/dev/block/mmcblk0p10   /dev/block/mmcblk0p19  /dev/block/mmcblk0p4
/dev/block/mmcblk0p11   /dev/block/mmcblk0p2   /dev/block/mmcblk0p5
/dev/block/mmcblk0p12   /dev/block/mmcblk0p20  /dev/block/mmcblk0p6
/dev/block/mmcblk0p13   /dev/block/mmcblk0p21  /dev/block/mmcblk0p7
/dev/block/mmcblk0p14   /dev/block/mmcblk0p22  /dev/block/mmcblk0p8
/dev/block/mmcblk0p15   /dev/block/mmcblk0p23  /dev/block/mmcblk0p9
/dev/block/mmcblk0p16   /dev/block/mmcblk0p24
console:/ #
```

Use the df -h command to see that the 16GB micro SD card has about 11 GB of space available after burning the Android system (more than 20 partitions will not be mounted to the Android system, focus on these to the partition).

```
console:/ # df -h
Filesystem                 Size Used Avail Use% Mounted on
tmpfs                      727M 1.1M  726M   1% /dev
tmpfs                      727M    0  727M   0% /mnt
/dev/block/mmcblk0p19       11M 136K   11M   2% /metadata
/dev/block/dm-0            782M 779M  2.4M 100% /
/dev/block/dm-1            104M 103M  332K 100% /vendor
/dev/block/dm-3            6.5M 6.5M   24K 100% /vendor_dlkm
/dev/block/dm-2            250M 249M  788K 100% /product
/dev/block/mmcblk0p23       16M    0   16M   0% /oem
tmpfs                      727M 8.0K  727M   1% /apex
tmpfs                      727M 532K  726M   1% /linkerconfig
/dev/block/mmcblk0p25       11G 904M   11G   8% /data
tmpfs                      727M    0  727M   0% /data_mirror
/dev/block/mmcblk0p24       16M    0   16M   0% /Reserve0
/dev/fuse                   11G 904M   11G   8% /mnt/user/0/emulated
/dev/block/vold/public:179,1 128M 5.3M  122M   5% /mnt/media_rw/0000-0000
/dev/fuse                  128M 5.3M  122M   5% /mnt/user/0/0000-0000
console:/ #
```

2.6. Instructions for using the micro linux system in the onboard SPI Flash

There is a 16MB SPI Flash on the development board, and its location is shown in the figure below:



There is a tiny linux system programmed into SPI Flash by default, which is mainly used to prove that the development board can start normally. When you get the development board, you don't need to burn the system into the micro SD card, you only need to connect the Type-C power supply to the development board to start the micro linux

system in the SPI Flash. The main functions of this system are:

a) During the u-boot start-up phase, the red LED light will be lit, and after entering the kernel, the red LED light will be turned off and the green LED light will be set to flash;

b) If the development board is connected to an HDMI screen, after the system starts up, the command line interface of the micro-linux system can be seen on the HDMI screen;

c) If the development board is connected with a USB keyboard, some simple linux commands can be run on the command line, such as ls, cd, etc.

Due to the limited functions of the tiny linux system in SPI Flash, if you want to use all the functions of the development board normally, please burn the linux image or Android image to the micro SD card, and then use it.

2. 7. Start the Orange Pi development board

1) Insert the micro SD card with the burned image into the micro SD card slot of the Orange Pi development board

2) The development board has a Micro HDMI interface, and the development board can be connected to a TV or HDMI display through a Micro HDMI to HDMI cable

3) If you have purchased a 13pin expansion board, you can plug the 13pin expansion board into the 13pin interface of the development board

4) Connect the USB mouse and keyboard to control the Orange Pi development board

5) The development board has an Ethernet port, which can be plugged into a network cable for Internet access

6) Connect a high-quality power adapter with a 5V/2A (5V/3A is also available) USB Type C interface

Remember not to plug in a power adapter with a voltage output greater than 5V, as this will burn out the development board.

Many unstable phenomena during the power-on and start-up process of the system are basically caused by power supply problems, so a reliable power adapter is very important. If you find that there is a phenomenon of continuous restart during the startup process, please replace the power supply or Type C data cable and try again.

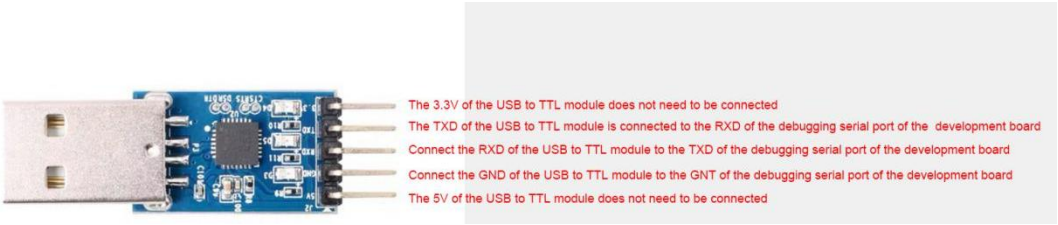
7) Then turn on the switch of the power adapter, if everything is normal, the HDMI monitor can see the system startup screen at this time

8) If you want to view the output information of the system through the debugging serial port, please use the serial cable to connect the development board to the computer. For the connection method of the serial port, please refer to **the section on how to use the debugging serial port**

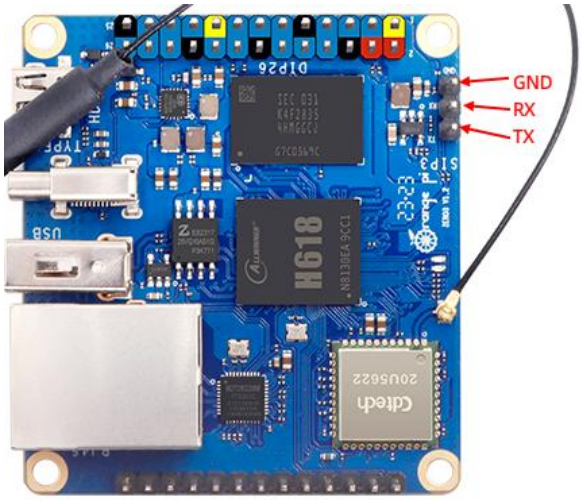
2. 8. How to use the debugging serial port

2. 8. 1. Connection instruction of debugging serial port

1) First, you need to prepare a 3.3v USB to TTL module, and then insert the USB interface end of the USB to TTL module into the USB interface of the computer



2) The corresponding relationship between GND, TX and RX pins of the debugging serial port of the development board is shown in the figure below



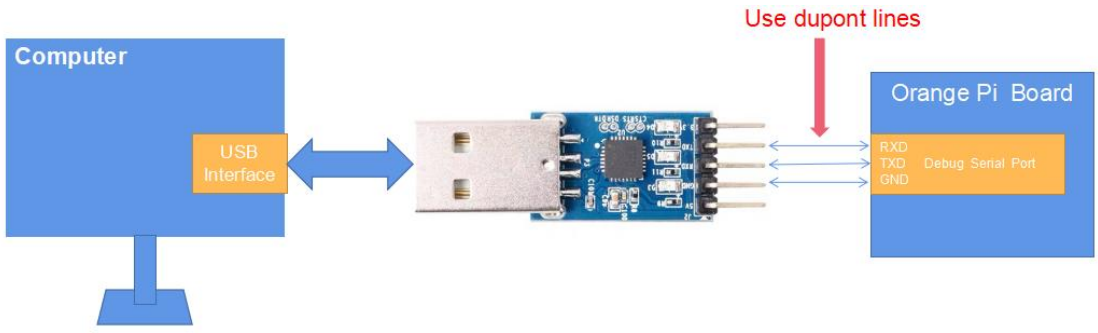
3) The GND, TX and RX pins of the USB to TTL module need to be connected to the debugging serial port of the development board through a DuPont line

a. Connect the GND of the USB to TTL module to the GND of the development board

b. The RX of the USB to TTL module is **connected to the TX of the development board**

c. Connect the TX of the USB **to TTL module to the RX of the development board**

4) The schematic diagram of connecting the USB to TTL module to the computer and the Orange Pi development board is as follows



Schematic diagram of connecting USB to TTL module to computer and Orange Pi development board

The TX and RX of the serial port need to be cross-connected. If you don't want to carefully distinguish the order of TX and RX, you can connect the TX and RX of

the serial port casually first. If there is no output from the test serial port, then exchange the order of TX and RX. One order is right.

2. 8. 2. How to use the debugging serial port on the Ubuntu platform

There are many serial port debugging software that can be used under Linux, such as putty, minicom, etc. The following demonstrates how to use putty.

1) First, insert the USB-to-TTL module into the USB port of the Ubuntu computer. If the connection and identification of the USB-to-TTL module is normal, you can see the corresponding device node name under **/dev** on the Ubuntu PC. Remember this node name, and then set the serial port software will be used

```
test@test:~$ ls /dev/ttyUSB*
/dev/ttyUSB0
```

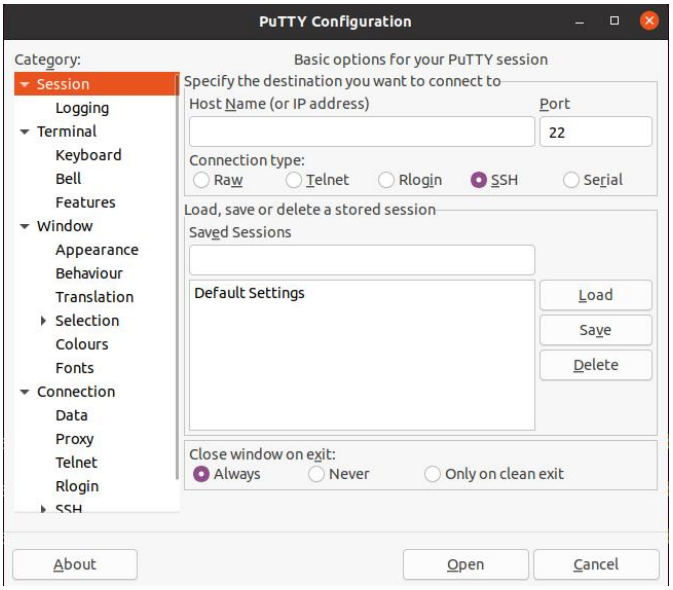
2) Then use the following command to install putty on Ubuntu PC

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install -y putty
```

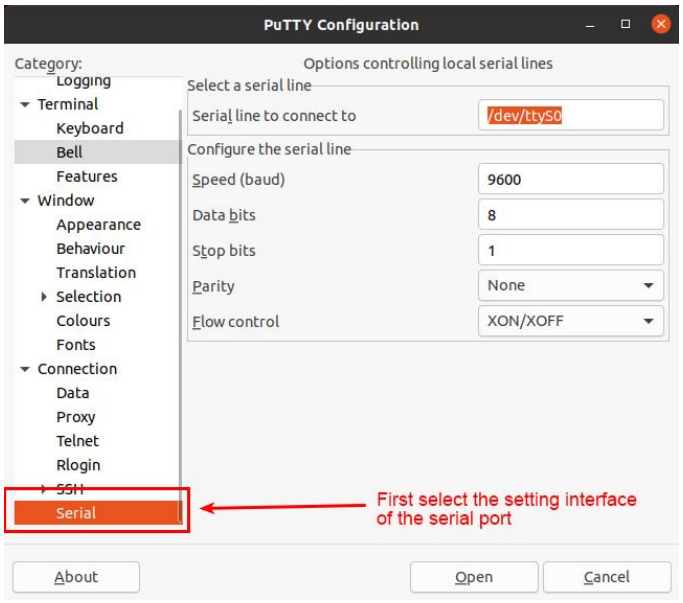
3) Then run putty, **remember to add sudo permission**

```
test@test:~$ sudo putty
```

4) After executing the putty command, the following interface will pop up



5) First select the setting interface of the serial port

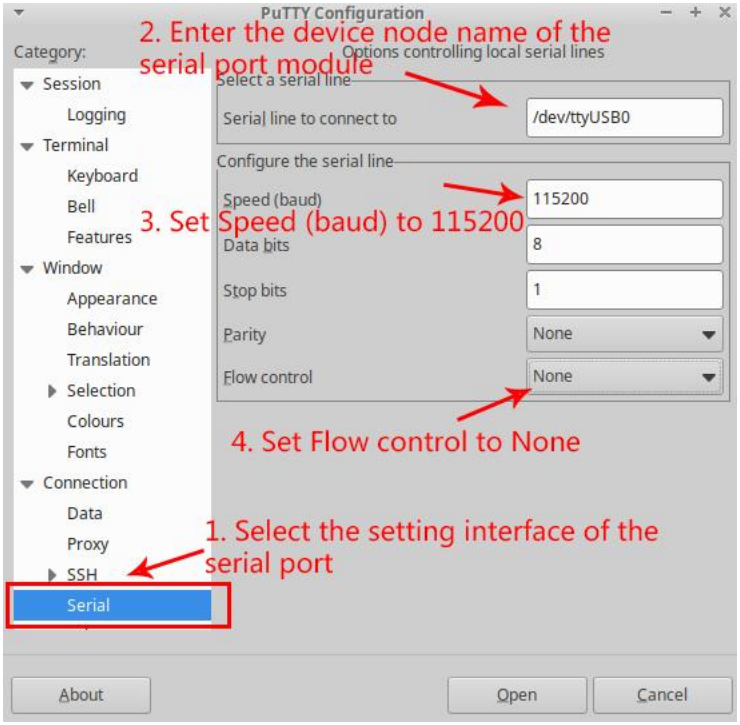


6) Then set the parameters of the serial port

a. Set **Serial line to connect to**to as **dev/ttyUSB0** (modify to the corresponding node name, usually /dev/ttyUSB0)

b. Set **Speed(baud)** to 115200 (the baud rate of the serial port)

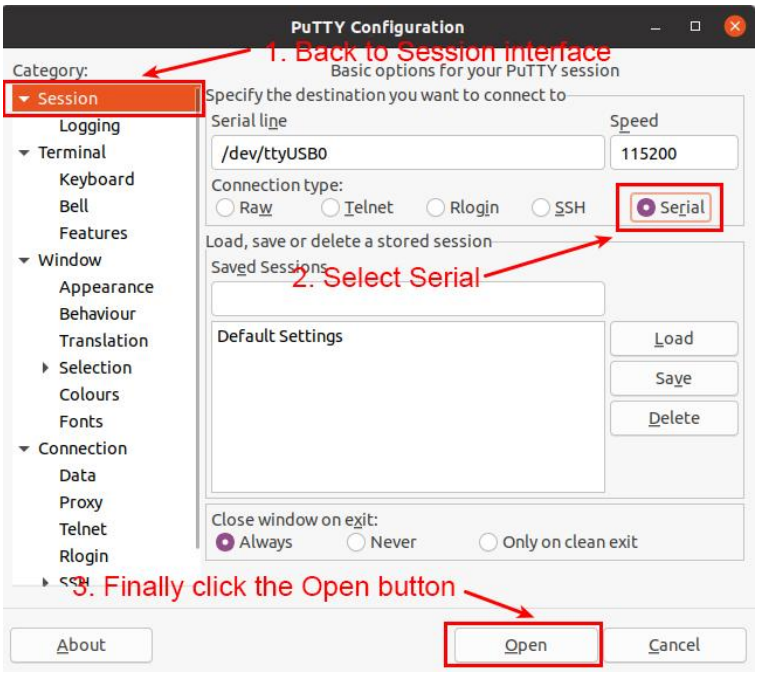
c. Set **Flow control** as **None**



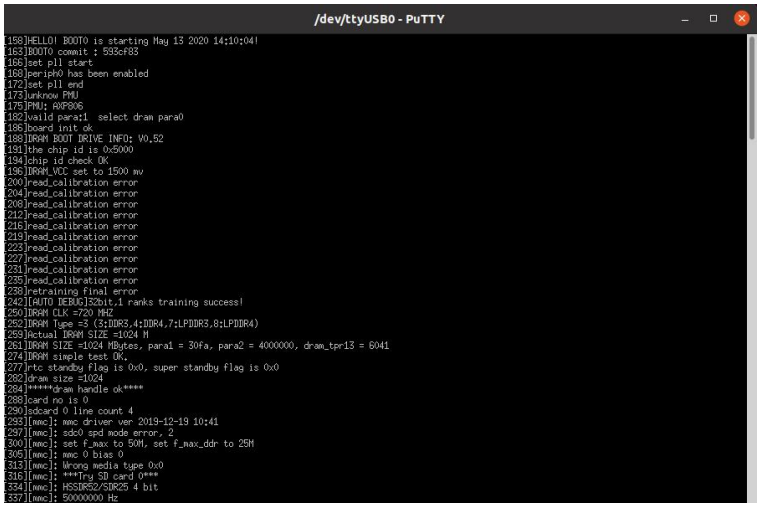
7) After setting the setting interface of the serial port, return to the Session interface

a. First select the **Connection type** as Serial

b. Then click the **Open** button to connect to the serial port



8) Then start the development board, and you can see the Log information output by the system from the opened serial port terminal



2. 8. 3. How to use the debugging serial port on Windows platform

There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use

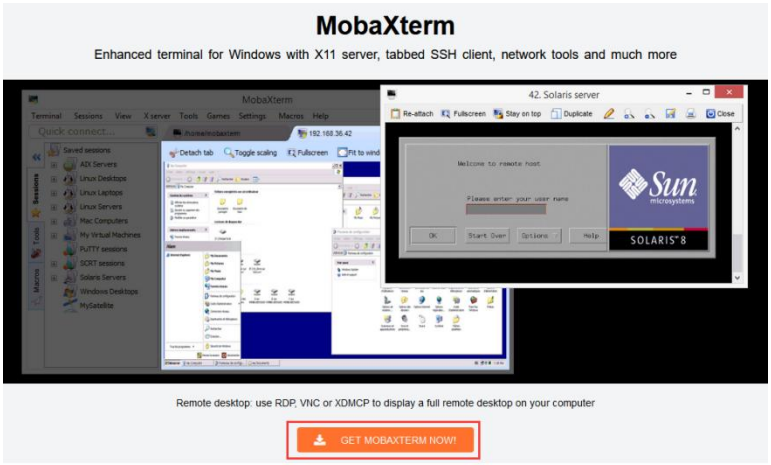
MobaXterm. This software has a free version and can be used without buying a serial number.

1) Download MobaXterm

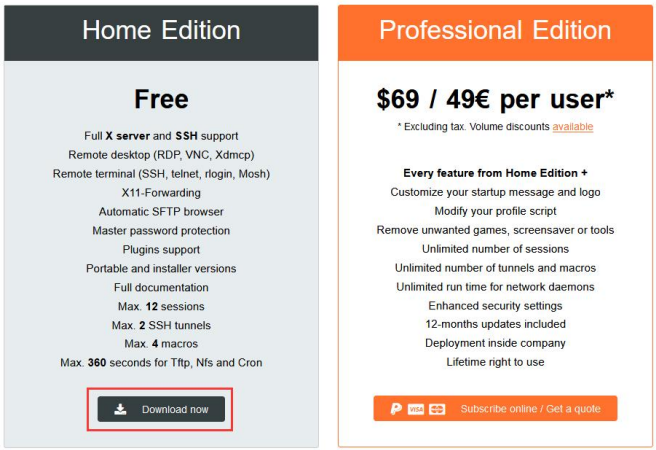
a. Download MobaXterm website as follows

https://mobaxterm.mobatek.net/

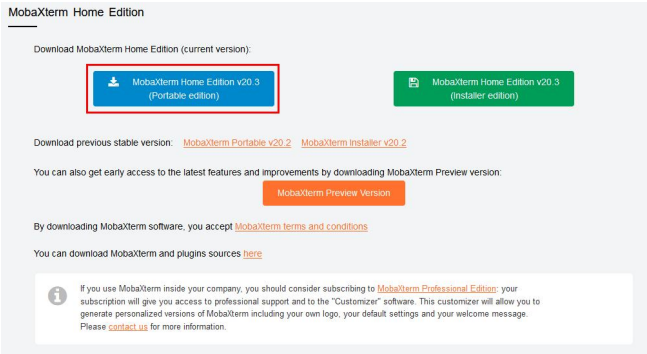
b. After entering the MobaXterm download page, click**GET XOBATERM NOW!**



c. Then choose to download the Home version



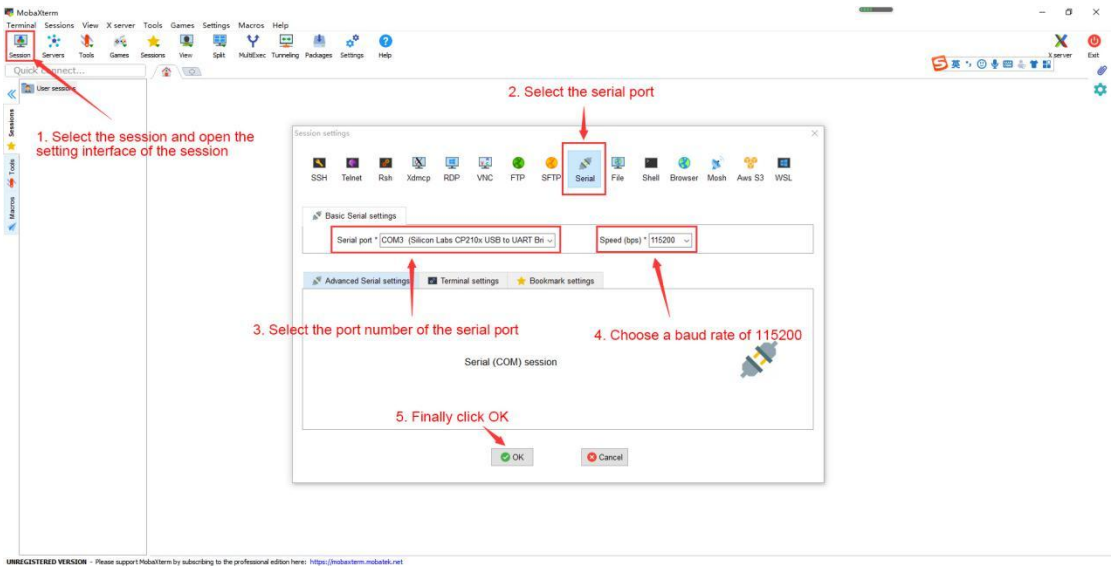
d. Then select the Portable version. After downloading, you don't need to install it, just open it and use it



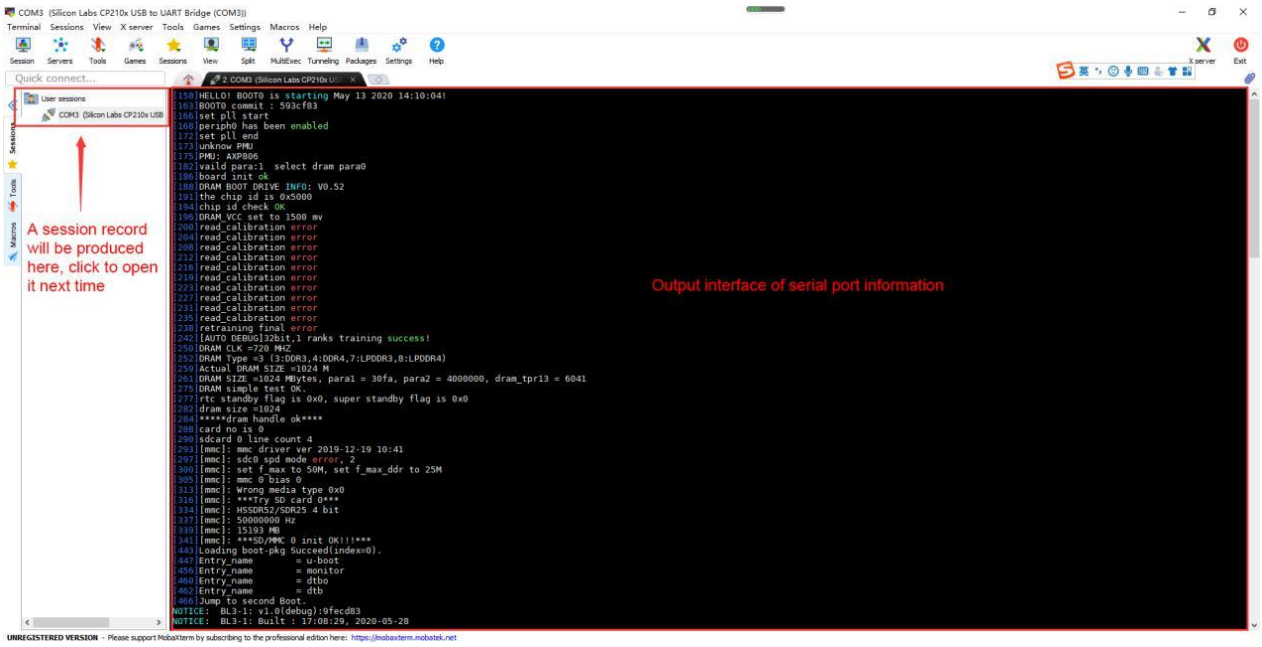
2) After downloading, use decompression software to decompress the downloaded compressed package, you can get the executable software of MobaXterm, and then double-click to open

3) After opening the software, the steps to set up the serial port connection are as follows

a. Open the session settings interface
b. Select the serial port type
c. Select the port number of the serial port (select the corresponding port number according to the actual situation), if you cannot see the port number, please use 360 **Driver Master** to scan and install the driver for the USB-to-TTL serial port chip
d. Select the baud rate of the serial port as **115200**
e. Finally click the "**OK**" button to complete the setup



4) After clicking the "**OK**" button, you will enter the following interface. At this time, start the development board and you can see the output information of the serial port



2. 9. Instructions for using the 5v pin in the 26pin or 13pin interface of the development board to supply power

The power supply method we recommend for the development board is to use the 5V/2A or 5V/3A Type C interface power cord to plug into the Type C power interface of the development board for power supply. If you need to use the 5V pin in the 26pin or 13pin interface to power the development board, please make sure that the power cable used can meet the power supply requirements of the development board. If the use is unstable, please switch back to the Type C power supply.

1) First, you need to prepare a power cord as shown in the figure below

The power cord shown in the picture above can be bought on Amazon, please search and buy by yourself.

2) Use the 5V pin in the 26pin or 13pin interface to supply power to the development board. The connection method of the power line is as follows

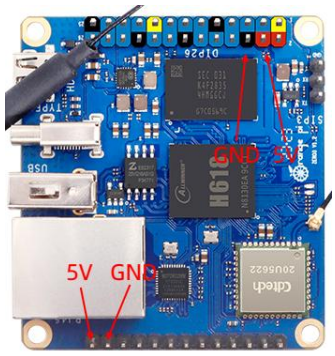
a. The USB A port of the power cord shown in the above figure needs to be plugged into the 5V/2A or 5V/3A power adapter connector (**it is not recommended to plug into the USB port of the computer for power supply. If there are too many peripherals connected to the development board, use will be unstable**)

b. The red DuPont wire needs to be plugged into the 5V pin of the 26pin or 13pin interface of the development board

c. The black Dupont wire needs to be inserted into the GND pin of the 26pin or 13pin interface

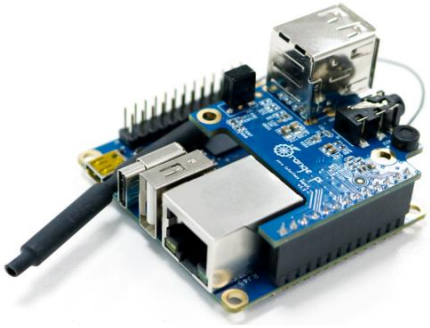
d. The positions of the 5V pin and GND pin of the 26pin and 13pin interfaces in

the development board are shown in the figure below**, remember not to reverse the connection.**



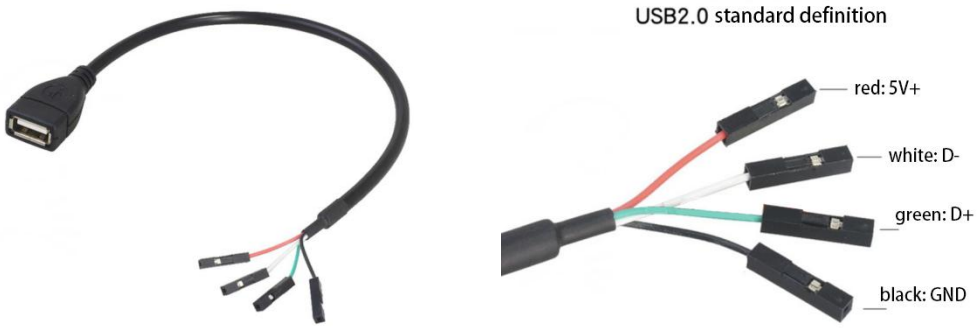
2. 10. The method of using the 13pin interface of the development board to expand the USB interface

1) If you have purchased a 13pin expansion board for Orange Pi, insert the expansion board into the 13pin interface of the development board to expand 2 USB interfaces

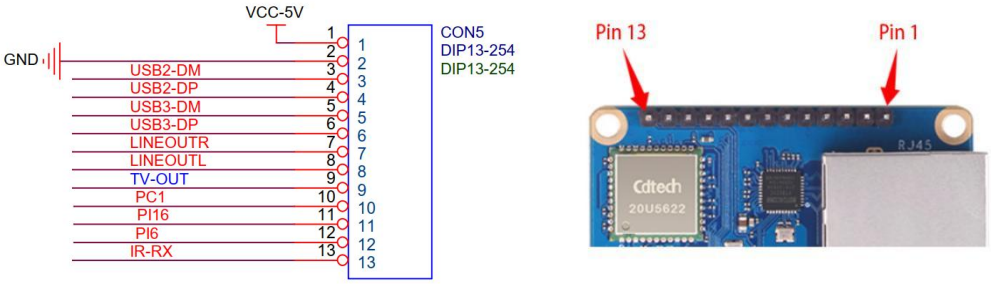


2) If there is no 13pin expansion board, you can use a 4pin 2.54mm DuPont to USB2.0 female cable to expand the USB interface. The specific method is as follows:

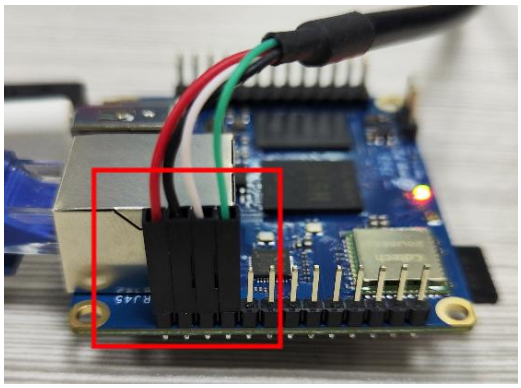
a. First, you need to prepare a 4pin 2.54mm Dupont to USB2.0 female cable (this cable can be bought on Amazon, please search and buy by yourself), as shown in the figure below:



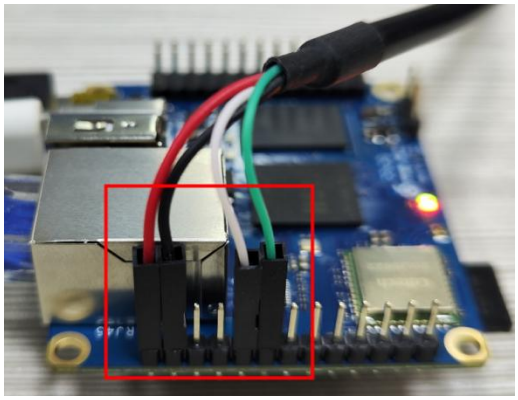
b. The schematic diagram of the 13pin interface is shown below



c. The wiring of USB2 is as follows

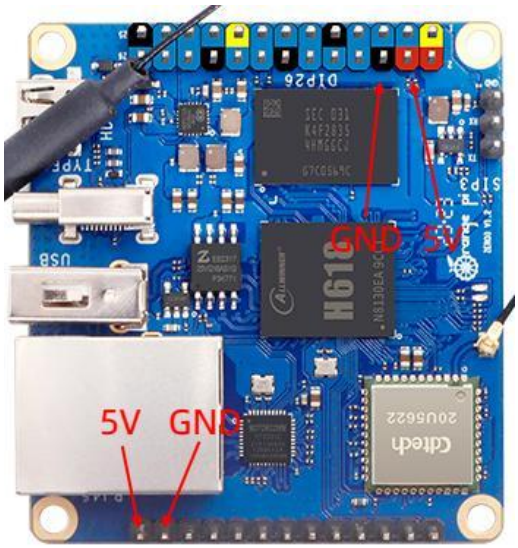


d. The wiring of USB3 is as follows



e. If you need to connect two USB devices to the 13pin interface at the same time, you will find that the 5V and GND pins on the 13pin interface are not enough. At

this time, one of the USB devices can use the 5V and GND pins in the 26pin interface. The location is as shown in the figure below Shown:



3. Instructions for use of Debian/Ubuntu Server and Xfce desktop system

3. 1. Supported linux image types and kernel versions

Linux image type	kernel version	server version	desktop version
Ubuntu 20.04 - Focal	Linux5.4	Support	Support
Ubuntu 22.04 - Focal	Linux5.4	Support	Support
Debian 11 - Bullseye	Linux5.4	Support	Support
Ubuntu 22.04 - Jammy	Linux6.1	Support	Support
Debian 11 - Bullseye	Linux6.1	Support	Support
Debian 12 - Bookworm	Linux6.1	Support	Support

After entering the download page of the corresponding development board **on the Orange Pi data download page**, you can see the following download options. In the description below, **the Ubuntu image and the Debian image are generally referred to as the Linux image**.

The naming rules of Linux images are as follows:

Development board model_version number_Linux distribution type_release code name_server or desktop_kernel version

a. **The model of the development board：** Both are **orangepizero3**. The model names of different development boards are generally different. Before burning the image, please make sure that the model name of the selected image matches the development board.

b. **Version number：** For example, **1.x.x**, this version number will increase with the update of the image function, and the last number of the version number of the Linux image on the development board is an even number.

c. **Types of Linux distributions：** **Ubuntu** and **Debian** are currently supported. Since Ubuntu is derived from Debian, there is not much difference between the two systems in terms of usage. However, there are still some differences in the default configuration of some software and the use of commands. In addition, both Ubuntu and Debian have their own software warehouses supported by maintenance, and there are also some differences in the supported and installable software packages. These need to be experienced in person to have a deeper understanding. For more details, you can refer to the official documentation provided by Ubuntu and Debian.

d. **Release code name：** Used to distinguish between different versions of a specific Linux distribution such as Ubuntu or Debian. Among them, both **focal** and **jammy** are Ubuntu distributions, focal means Ubuntu20.04, and jammy means Ubuntu22.04. The biggest difference between different versions is that the software in the software warehouse maintained by the new version of Ubuntu system is much better than that of the old version of Ubuntu system. The ones in it should be new, such as Python and GCC compilation tool chains. **bullseye** is the specific version code of Debian, bullseye means Debian11, and bookworm means Debian12.

e. **Server or Desktop：** It is used to indicate whether the system has a desktop

environment. If it is server, it means that the system does not have a desktop environment. The storage space and resources occupied by the image are relatively small, and the command line is mainly used to operate and control the system. If it is **desktop_xfce**, it means that the system is installed with the XFCE desktop environment by default. The storage space and resources occupied by the image are relatively large. You can connect the monitor, mouse and keyboard to operate the system through the interface. Of course, the desktop version of the system can also be operated through the command line like the server version of the system.

f. **Kernel version：** Used to indicate the version number of the linux kernel, currently supports **linux5.4** and **linux6.1.**

3. 2. Linux kernel driver adaptation

Function	Linux5.4	Linux6.1
HDMI video	OK	OK
HDMI audio	OK	OK
USB2.0 x 3	OK	OK
micro SD card start	OK	OK
Gigabit Ethernet	OK	OK
Infrared receiver	OK	OK
WIFI	OK	OK
Bluetooth	OK	OK
headphone audio	OK	OK
usb camera	OK	OK
LED light	OK	OK
26pin GPIO	OK	OK
26pin I2C	OK	OK
26pin SPI1	OK	OK
26pin UART	OK	OK
PWM	OK	OK
Temperature Sensor	OK	OK
hardware watchdog	OK	OK
Mali GPU	NO	NO

Video codec	NO	NO
TV-OUT	NO	NO

3. 3. The format of linux commands in this manual

1) In this manual, all commands that need to be entered in the Linux system will be marked with the following box

As shown below, the content in the yellow box indicates the content that needs special attention, except for the commands in it.

2) Description of the prompt type in front of the command

a. The prompt in front of the command refers to the content of the red part in the box below, which is not part of the linux command, so when entering the command in the linux system, please do not enter the content of the red font part.

```
orangepi@orangepi:~$ sudo apt update
root@orangepi:~# vim /boot/boot.cmd
test@test:~$ ssh root@192.168.1.xxx
root@test:~# ls
```

b. **root@orangepi:~$** The prompt indicates that this command is entered in the **linux system of the development board**. The **$** at the end of the prompt indicates that the current user of the system is a normal user. When executing a privileged command, you need to add **sudo**

c. **root@orangepi:~#** The prompt indicates that this command is entered in the linux system of the development board, and the # at the end of the prompt indicates that the current user of the system is the root user, who can execute any desired command

d. **test@test:~$** The prompt indicates that this command is entered in the Ubuntu PC or Ubuntu virtual machine, not in the linux system of the development board. The $ at the end of the prompt indicates that the current user of the system is an ordinary user. When executing privileged commands, **sudo** needs to be added

e. **root@test:~#** The prompt indicates that this command is entered in the Ubuntu PC or Ubuntu virtual machine, not in the linux system of the development board. The # at the end of the prompt indicates that the current user of the system is the

root user and can execute any command you want

3) What are the commands that need to be entered?

a. As shown below, **the black bold part** is the command that needs to be input, and the content below the command is the output content (some commands have output, some may not), and this part of the content does not need to be input

```
root@orangepi:~# cat /boot/orangepiEnv.txt
verbosity=7
bootlogo=false
console=serial
```

b. As shown below, some commands cannot be written in one line and will be placed on the next line. As long as the black and bold parts are all commands that need to be input. When these commands are entered into one line, the last "\" of each line needs to be removed, this is not part of the command. In addition, there are spaces in different parts of the command, please don't miss it

```
orangepi@orangepi:~$ echo    \
"deb [arch=$(dpkg --print-architecture)    \
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]    \
https://download.docker.com/linux/debian    \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

3. 4. Linux system login instructions

3. 4. 1. Linux system default login account and password

Account	password
root	orangepi
orangepi	orangepi

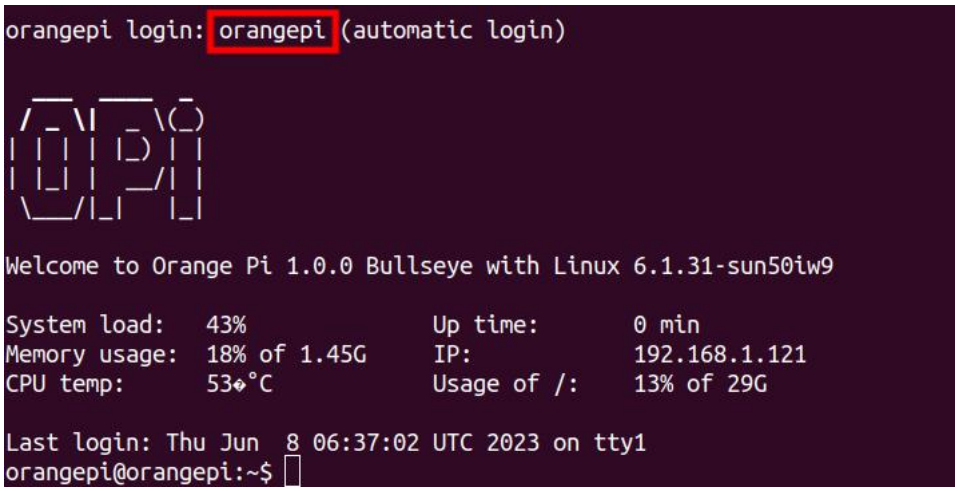
Note that when entering the password, the specific content of the entered password will not be displayed on the screen, please do not think that there is any fault, just press Enter after inputting.

When the wrong password is prompted, or there is a problem with the ssh connection, please note that as long as you are using the Linux image provided by Orange Pi, please do not suspect that the above password is wrong, but look for

other reasons.

3. 4. 2. How to set automatic terminal login in linux system

1) The Linux system automatically logs in to the terminal by default, and the default login user name is **orangepi**



2) Use the following command to set the root user to automatically log in to the terminal

```
orangepi@orangepi:~$ sudo auto_login_cli.sh root
```

3) Use the following command to disable automatic login terminal

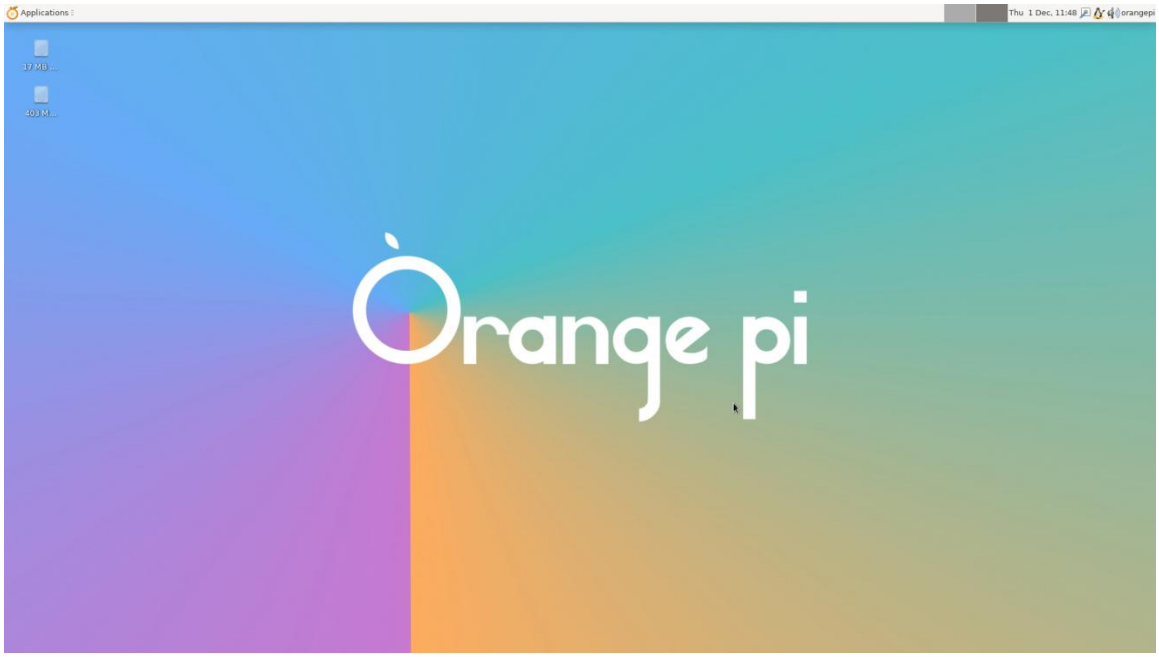
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

4) Use the following command to set the orangepi user to automatically log in to the terminal again

```
orangepi@orangepi:~$ sudo auto_login_cli.sh orangepi
```

3. 4. 3. Instructions for automatic login of linux desktop version system

1) After the desktop system starts, it will automatically log in to the desktop without entering a password



2) Run the following command to prohibit the desktop system from automatically logging into the desktop

```
orangepi@orangepi:~$ sudo disable_desktop_autologin.sh
```

3) Then restart the system and a login dialog box will appear, at which point a **Linux system default login account and password** is required to enter the system

3. 4. 4. The setting method of root user automatic login in Linux desktop version system

1) Execute the following command to set the desktop system to automatically log in as the root user

```
orangepi@orangepi:~$ sudo desktop_login.sh root
```

2) Then restart the system, and the root user will automatically log in to the desktop

Note that if you log in to the desktop system as the root user, you cannot use pulseaudio in the upper right corner to manage audio devices.

Also note that this is not a bug, since pulseaudio is not allowed to run as root.

3) Execute the following command to set the desktop system to log in automatically with the orangepi user again

```
orangepi@orangepi:~$ sudo desktop_login.sh orangepi
```

3. 4. 5. The method of disabling the desktop in the Linux desktop version system

1) First enter the following command on the command line, **Please remember to add sudo permission**

```
orangepi@orangepi:~$ sudo systemctl disable lightdm.service
```

2) Then restart the Linux system and you will find that the desktop will not be displayed

```
orangepi@orangepi:~$ sudo reboot
```

3) The command to reopen the desktop is as follows， **Please remember to add sudo permission**

```
orangepi@orangepi:~$ sudo systemctl start lightdm.service
orangepi@orangepi:~$ sudo systemctl enable lightdm.service
```

3. 5. Onboard LED Light Test Instructions

1) There are two LED lights on the development board, one is green and the other is red. When the system starts, the default display of the LED lights is as follows:

	Green light	Red light
u-boot startup phase	**off**	**on**
The kernel boots into the system	**flashing**	**off**
GPIO interface	**PC13**	**PC12**

The two LED lights on the development board are controlled by software.

When you get the development board, you may find that even if the micro SD card with the system programmed is not inserted into the development board, the two LED lights will be on after the development board is connected to the power supply. This is because the 16MB on the development board SPI Flash will burn a miniature linux system by default when leaving the factory. This system will light up the red light during the u-boot startup phase. After entering the kernel, it will turn off the red light and set the green light to flash.

If the linux system in the SPI Flash is cleared, then the two LED lights on the development board will not light up after the power is turned on without inserting the micro SD card with the system programmed

2) The method of setting the green light on and off and flashing is as follows:

Note that the following operations should be performed under the root user.

a. First enter the setting directory of the green light

```
root@orangepi:~# cd /sys/class/leds/green_led
```

b. The command to set the green light to stop flashing is as follows

```
root@orangepi:/sys/class/leds/green_led# echo none > trigger
```

c. The command to set the green light to be on is as follows

```
root@orangepi:/sys/class/leds/green_led# echo default-on > trigger
```

d. The command to set the green light to flash is as follows

```
root@orangepi:/sys/class/leds/green_led# echo heartbeat > trigger
```

3) The method of setting the red light on/off and flashing is as follows:

Note that the following operations should be performed under the root user.

a. First enter the setting directory of the red light

```
root@orangepi:~# cd /sys/class/leds/red_led
```

b. The command to set the red light to be on is as follows

```
root@orangepi:/sys/class/leds/red_led# echo default-on > trigger
```

c. The command to set the flashing red light is as follows

```
root@orangepi:/sys/class/leds/red_led# echo heartbeat > trigger
```

d. The command to set the red light to stop flashing is as follows

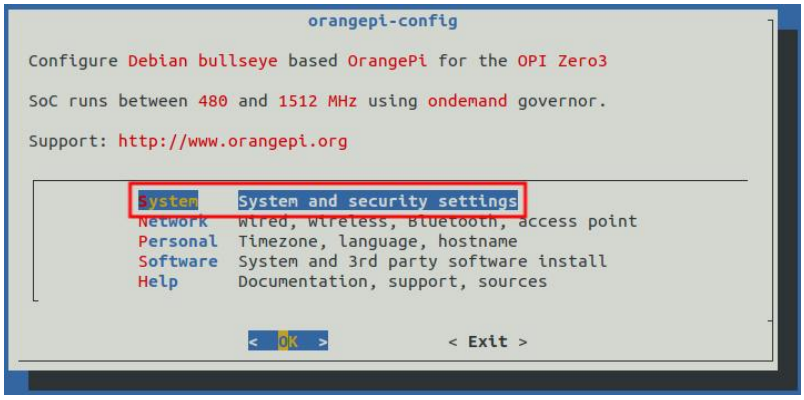
```
root@orangepi:/sys/class/leds/red_led# echo none > trigger
```

4) If you do not need the LED light to flash after booting, you can use the following method to turn off the green light flashing

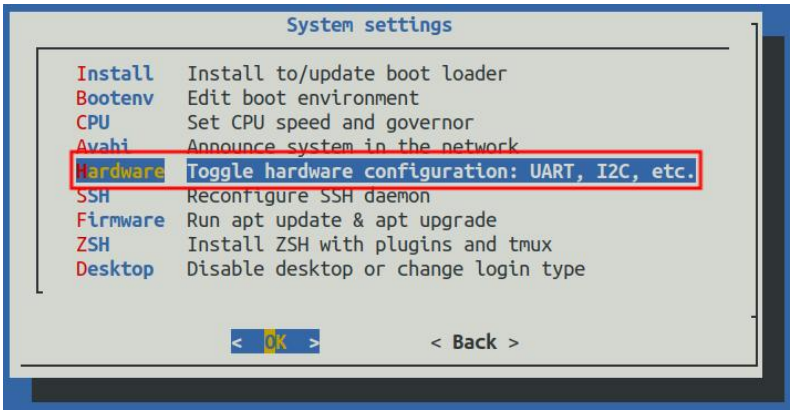
a. First run **orangepi-config**, common users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



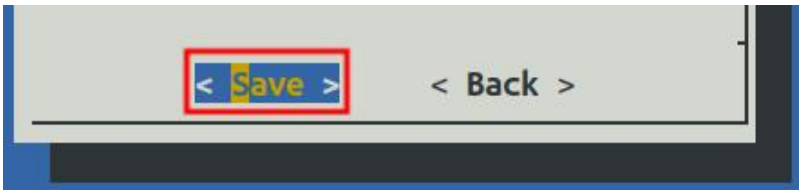
c. Then select **Hardware**



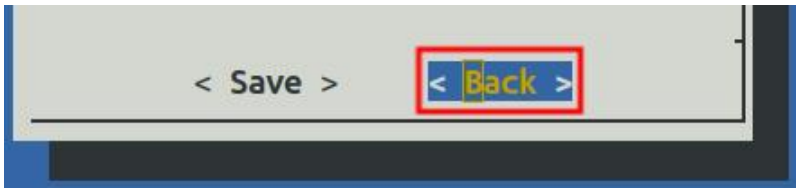
d. Then use the arrow keys on the keyboard to navigate to the position shown in the

figure below, and then use the space to select **disable-leds**

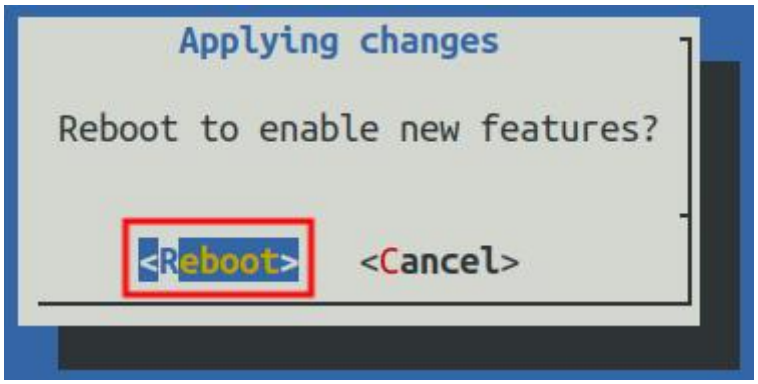
e. Then selec **<Save>**



f. Then select **<Back>**



g. Then select **<Reboot>**to restart the system for the configuration to take effect



h. After restarting and fully entering the system, you can see that the two LED lights on the development board will not light up

3. 6. Instructions for operating the rootfs partition capacity of the linux system in the micro SD card

3. 6. 1. The first startup will automatically expand the capacity of the rootfs partition in the micro SD card

1) After burning the Linux image of the development board into the micro SD card, you can check the capacity usage of the micro SD card on the **Ubuntu computer**, the steps are as follows:

Note that this step does not affect the automatic expansion of the Linux system

of the development board. Here I just want to explain how to check the capacity of the micro SD card after burning the Linux image on the micro SD card.

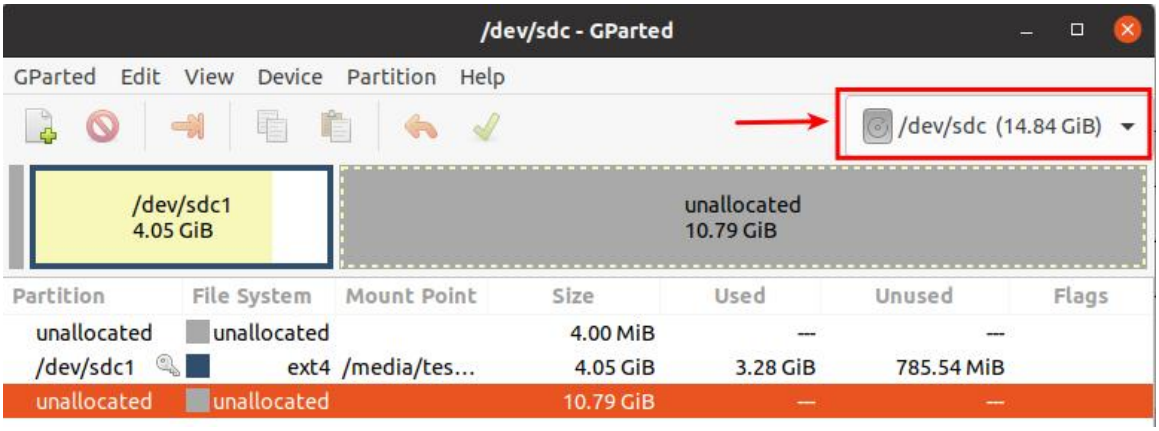
a. First install the gparted software on the Ubuntu computer

```
test@test:~$ sudo apt install -y gparted
```

b. Then open gparted

```
test@test:~$ sudo gparted
```

c. After opening gparted, you can select the micro SD card in the upper right corner, and then you can see the usage of the micro SD card capacity



d. The figure above shows the situation of the micro SD card after burning the Linux desktop version system. It can be seen that although the total capacity of the micro SD card is 16GB (displayed as 14.84GiB in GParted), the rootfs partition (/dev/ sdc1) actually allocated only 4.05GiB, leaving 10.79GiB unallocated

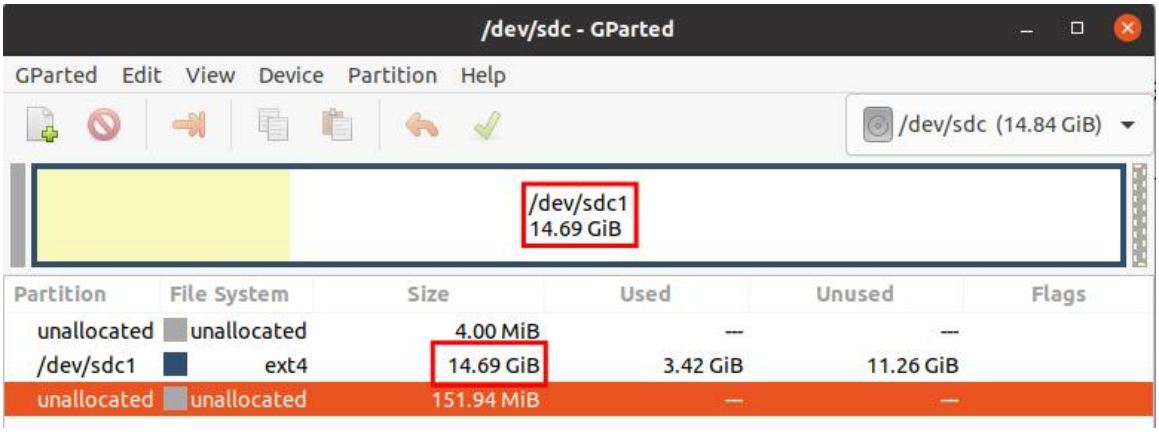
2) Then you can insert the micro SD card that has burned the Linux system into the development board to start. When the micro SD card starts the Linux system for the first time, it will call the **orangepi-resize-filesystem** script automatically through the systemd service **orangepi-resize-filesystem.service** The expansion of the rootfs partition, **so there is no need to manually expand the capacity**

3) After logging in to the system, you can use the **df -h** command to check the size of the rootfs. If it is consistent with the actual capacity of the micro SD card, it means that the automatic expansion is running correctly.

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            430M     0  430M   0% /dev
```

```
tmpfs           100M  5.6M   95M   6% /run
/dev/mmcblk0p1   15G   915M   14G   7% /
tmpfs           500M     0  500M   0% /dev/shm
```

4) After starting the Linux system for the first time, we can also remove the micro SD card from the development board and reinsert it into the **Ubuntu computer**, and then use gparted to check the status of the micro SD card again, as shown in the figure below, the rootfs partition (/dev/ The capacity of sdc1) has been expanded to 14.69GiB



It should be noted that the Linux system has only one partition in ext4 format, and does not use a separate BOOT partition to store files such as the kernel image, so there is no problem of expanding the BOOT partition.

3. 6. 2. The method of prohibiting automatic expansion of the capacity of the rootfs partition in the micro SD card

1) First burn the linux image of the development board to the micro SD card on the **Ubuntu computer** (Windows not available), **and then re-plug and insert the micro SD card**

2) Then the Ubuntu computer will automatically mount the partition of the micro SD card. If the automatic mounting is normal, use the ls command to see the following output

```
test@test:~$ ls /media/test/opi_root/
bin  boot  dev  etc  home  lib  lost+found  media  mnt  opt  proc  root  run
sbin  selinux  srv  sys  tmp  usr  var
```

3) Then switch the current user to the root user on the Ubuntu computer

```
test@test:~$ sudo -i
[sudo] password for test:
```

```
root@test:~#
```

4) Then enter the root directory of the linux system in the micro SD card and create a new file named **.no_rootfs_resize**

```
root@test:~# cd /media/test/opi_root/
root@test:/media/test/opi_root/# cd root
root@test:/media/test/opi_root/root# touch .no_rootfs_resize
root@test:/media/test/opi_root/root# ls .no_rootfs*
.no_rootfs_resize
```

5) Then you can uninstall the micro SD card, and then pull out the micro SD card and insert it into the development board to start. When the linux system starts, when the file**.no_rootfs_resize** is detected in the **/root** directory, the rootfs will not be automatically expanded.

6) After entering the Linux system after prohibiting automatic expansion of rootfs, you can see that the total capacity of the rootfs partition is only 4GB (the image of the desktop version is tested here), which is much smaller than the actual capacity of the micro SD card, indicating that the automatic expansion of rootfs is prohibited.

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            925M     0  925M   0% /dev
tmpfs           199M  3.2M  196M   2% /run
/dev/mmcblk0p1  4.0G  3.2G  686M  83% /
```

7) If you need to re-expand the capacity of the rootfs partition in the micro SD card, just execute the following command, and then restart the Linux system of the development board.

Note, please execute the following commands under the root user.

```
root@orangepi:~# rm /root/.no_rootfs_resize
root@orangepi:~# systemctl enable orangepi-resize-filesystem.service
root@orangepi:~# sudo reboot
```

After restarting, enter the Linux system of the development board again, and you can see that the rootfs partition has been expanded to the actual capacity of the micro SD card

```
root@orangepi:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            925M     0  925M   0% /dev
tmpfs           199M  3.2M  196M   2% /run
/dev/mmcblk0p1   15G  3.2G   12G  23% /
```

3. 6. 3. The method of manually expanding the capacity of the rootfs partition in the micro SD card

If the total capacity of the micro SD card is large, such as 128GB, you do not want the rootfs partition of the Linux system to use all the capacity of the micro SD card, but only want to allocate a part of the capacity, such as 16GB, to the Linux system, and then the remaining capacity of the micro SD card can be used for other use. Then you can use the content introduced in this section to manually expand the capacity of the rootfs partition in TF.

1) First burn the linux image of the development board to the micro SD card on the **Ubuntu computer** (Windows not available), **and then re-plug and insert the micro SD card**

2) Then the Ubuntu computer will automatically mount the partition of the micro SD card. If the automatic mounting is normal, use the ls command to see the following output

```
test@test:~$ ls /media/test/opi_root/
bin  boot  dev  etc  home  lib  lost+found  media  mnt  opt  proc  root  run
sbin  selinux  srv  sys  tmp  usr  var
```

3) Then switch the current user to the root user on the Ubuntu computer

```
test@test:~$ sudo -i
[sudo] password for test:
root@test:~#
```

4) Then enter the root directory of the linux system in the micro SD card and create a new file named **.no_rootfs_resize**

```
root@test:~# cd /media/test/opi_root/
root@test:/media/test/opi_root/# cd root
root@test:/media/test/opi_root/root# touch .no_rootfs_resize
root@test:/media/test/opi_root/root# ls .no_rootfs*
.no_rootfs_resize
```

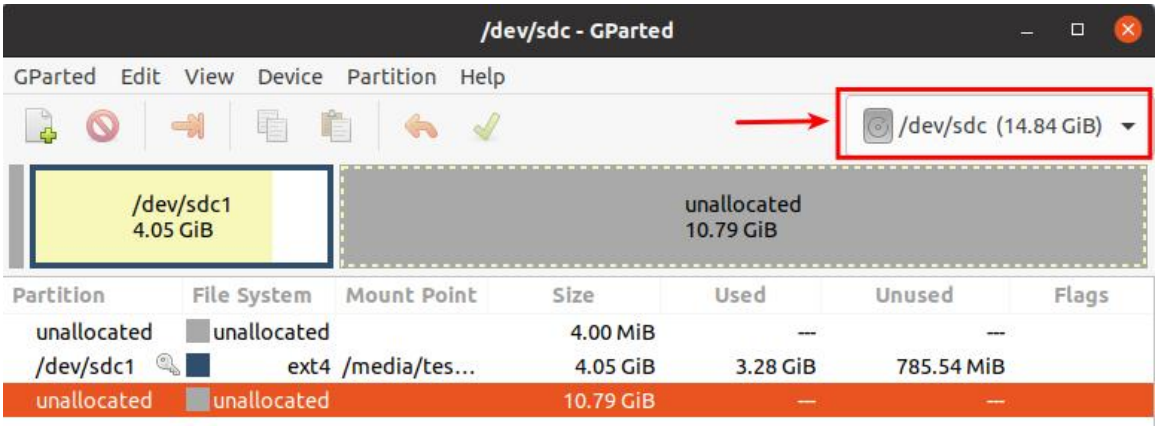
5) Then install the gparted software on the Ubuntu computer

```
test@test:~$ sudo apt install -y gparted
```

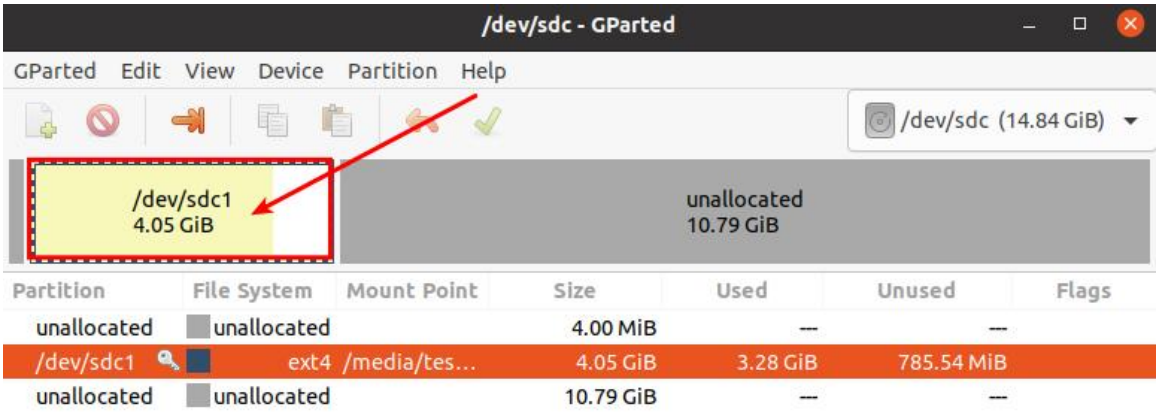
6) Then open gparted

```
test@test:~$ sudo gparted
```

7) After opening gparted, you can select the micro SD card in the upper right corner, and then you can see the usage of the micro SD card capacity. The figure below shows the situation of the micro SD card after burning the Linux desktop version system. It can be seen that although the total capacity of the micro SD card is 16GB (displayed as 14.84GiB in GParted), the rootfs partition (/dev/sdc1) Only 4.05GiB were actually allocated, leaving 10.79GiB unallocated

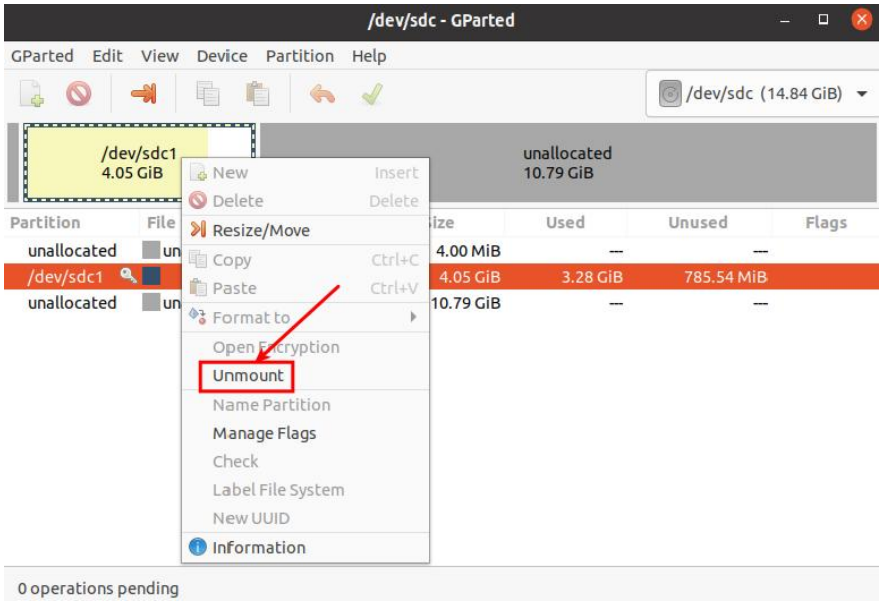


8) Then select the rootfs partition (/dev/sdc1)

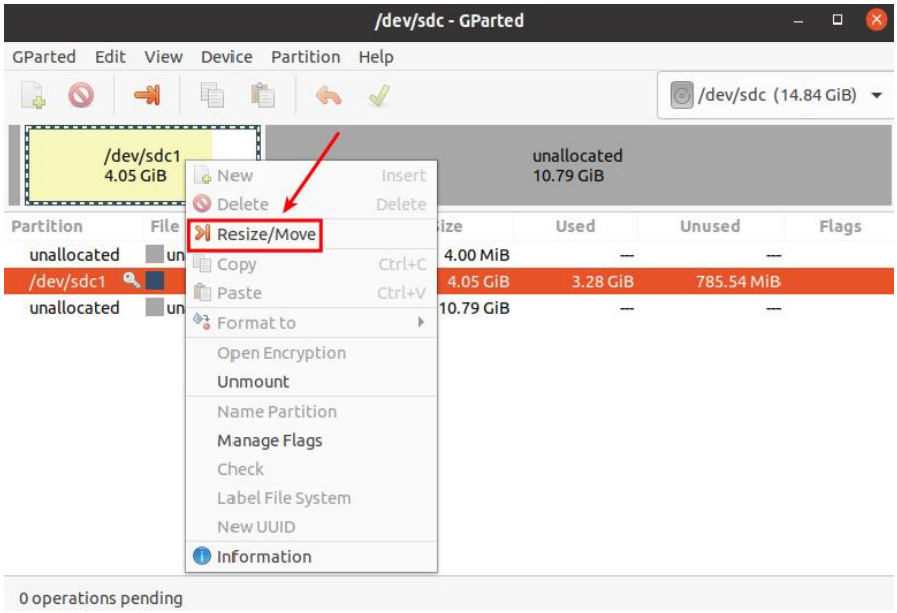


9) Click the right mouse button again to see the operation options shown in the figure below. If the micro SD card has been mounted, first you need to Umount the rootfs

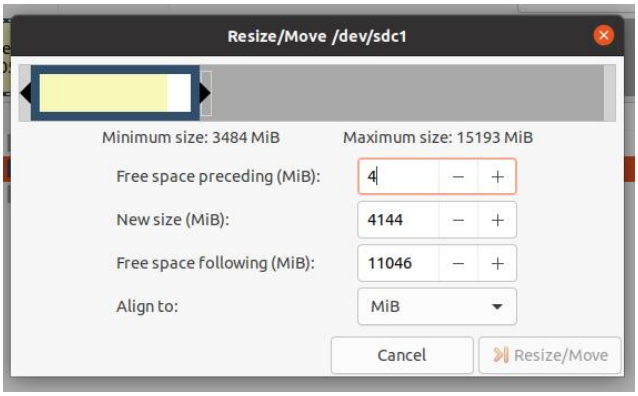
partition of the micro SD card



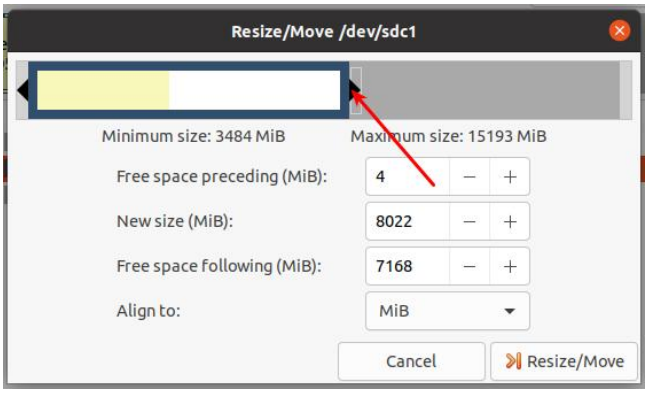
10) Then select the rootfs partition again, click the right mouse button, and select **Resize/Move** to start expanding the size of the rootfs partition



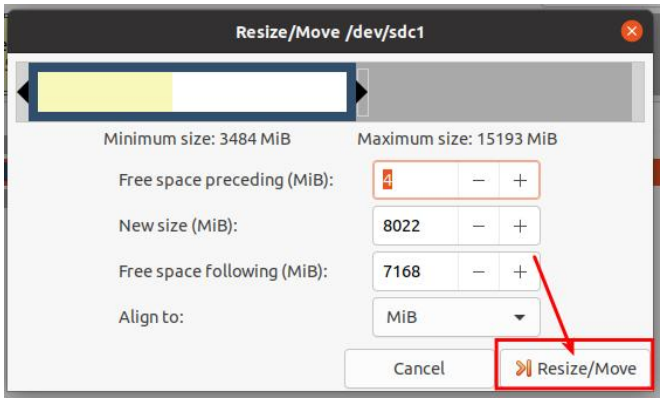
11) After the **Resize/Move** option is turned on, the following setting interface will pop up



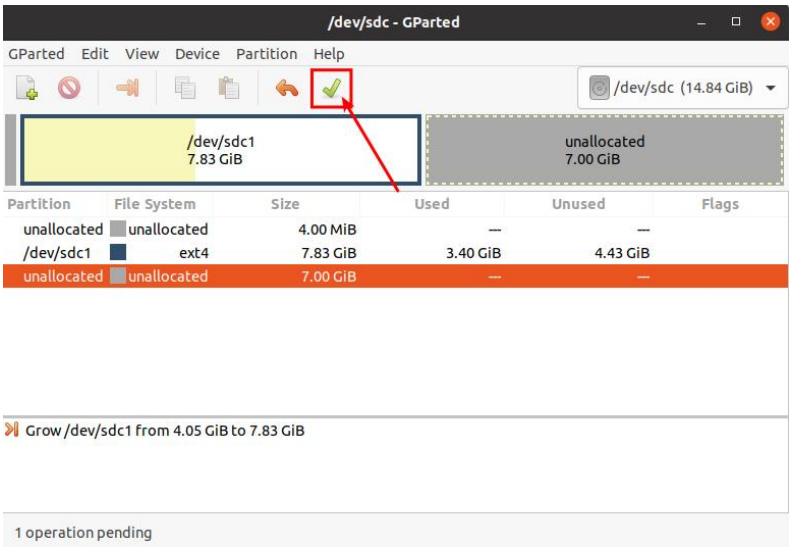
12) Then you can directly drag the location shown in the figure below to set the size of the capacity, or you can set the size of the rootfs partition by setting the number in **New size(MiB)**



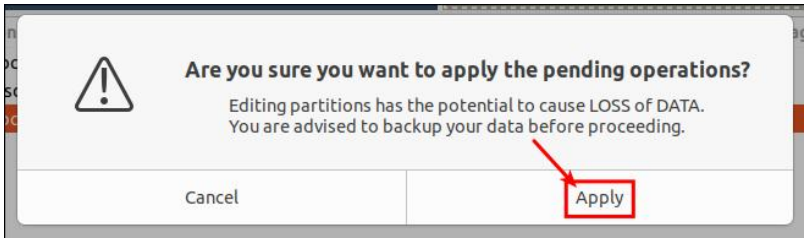
13) After setting the capacity, click **Resize/Move** in the lower right corner



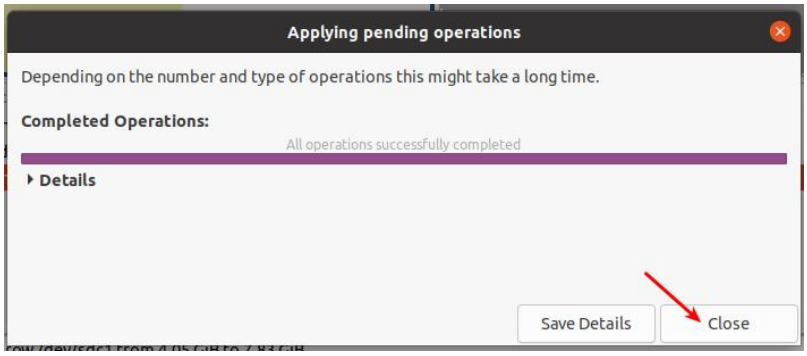
14) After confirming that it is correct, click the green button shown in the figure below √



15) Then select **Apply**, it will officially start to expand the capacity of the rootfs partition



16) After the expansion is completed, click **Close** to close



17) Then you can pull out the micro SD card, insert it into the development board and start it up. After entering the Linux system of the development board, if you can see that the size of the rootfs partition is the same as the size set before, it means manual Expansion succeeded

```
root@orangepi:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
```

```
udev            925M     0  925M   0% /dev
tmpfs           199M  3.2M  196M   2% /run
/dev/mmcblk0p1  7.7G  3.2G  4.4G  42% /
```

3. 6. 4. How to reduce the capacity of the rootfs partition in the micro SD card

After configuring the application program or other development environment in the Linux system of the micro SD card, if you want to back up the Linux system in the micro SD card, you can use the method in this section to reduce the size of the rootfs partition first, and then start the backup.

1) First insert the micro SD card you want to operate in the **Ubuntu computer** (Windows not available)

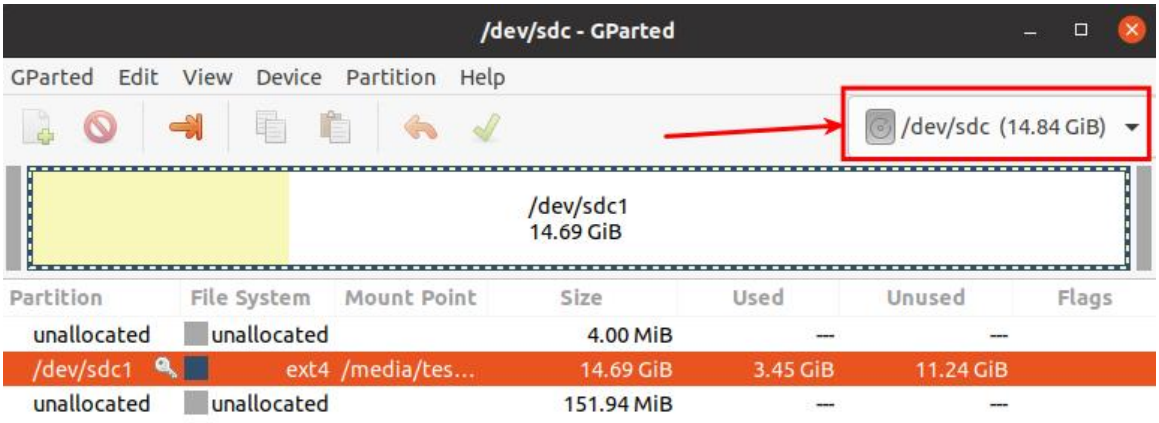
2) Then install the gparted software on the Ubuntu computer

```
test@test:~$ sudo apt install -y gparted
```

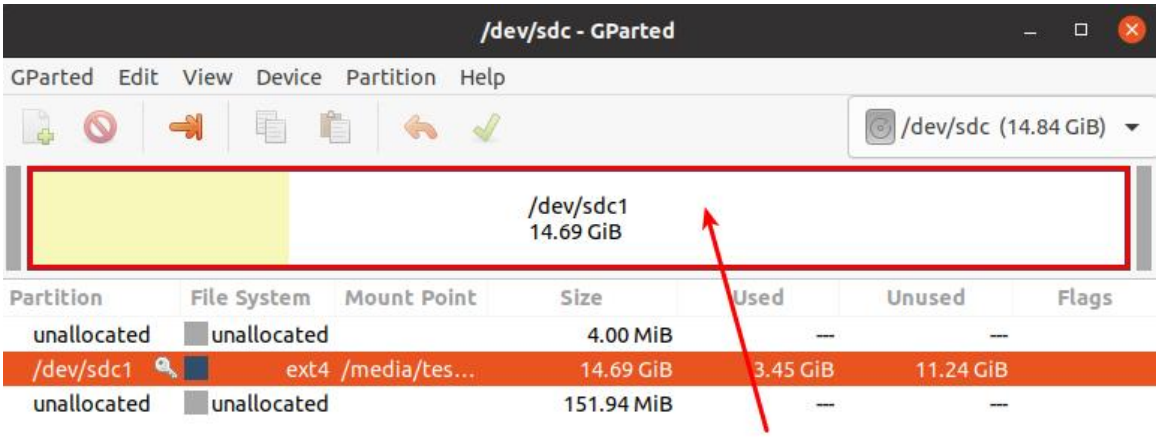
3) Then open gparted

```
test@test:~$ sudo gparted
```

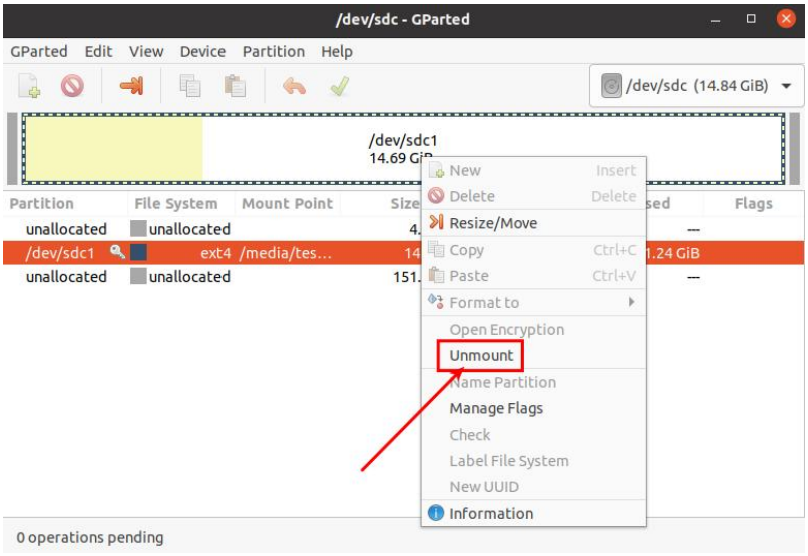
4) After opening gparted, you can select the micro SD card in the upper right corner, and then you can see the usage of the micro SD card capacity



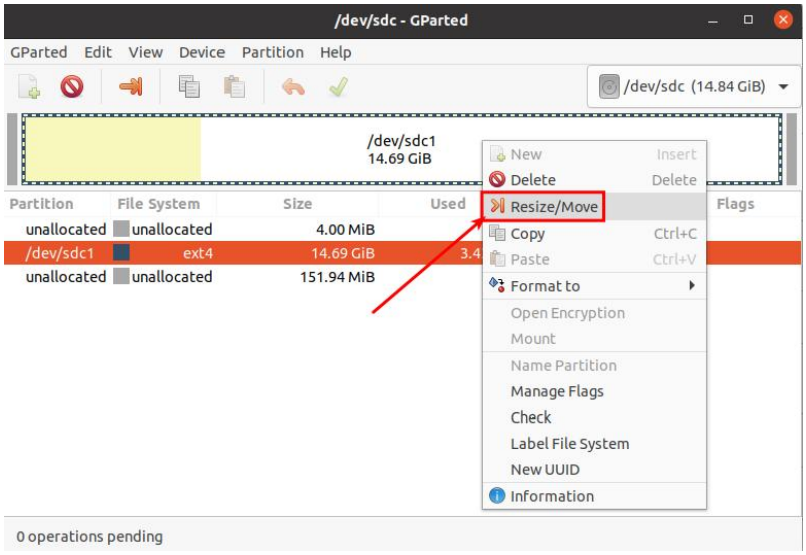
5) Then select the rootfs partition (/dev/sdc1)



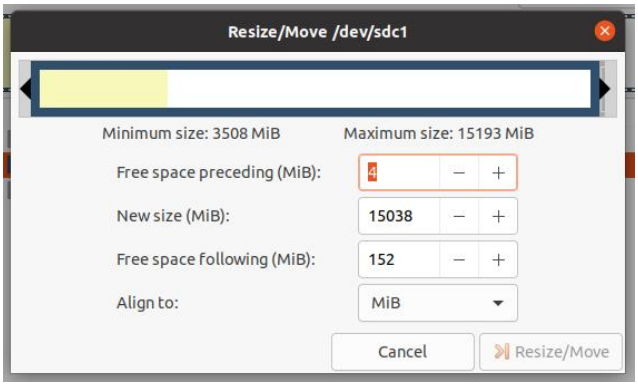
6) Click the right mouse button again to see the operation options shown in the figure below. If the micro SD card has been mounted, first you need to Umount the rootfs partition of the micro SD card



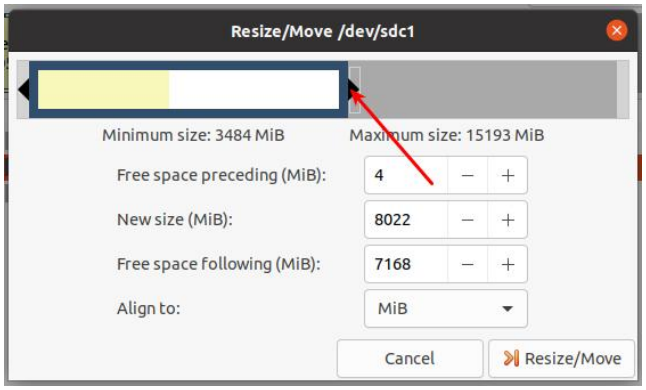
7) Then select the rootfs partition again, click the right mouse button, and select **Resize/Move** to start setting the size of the rootfs partition



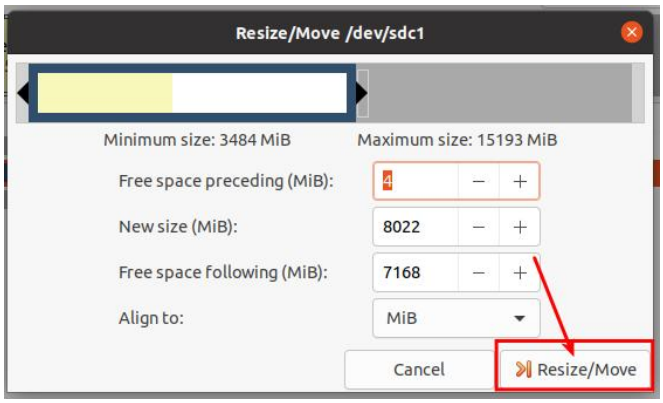
8) After the **Resize/Move** option is turned on, the following setting interface will pop up



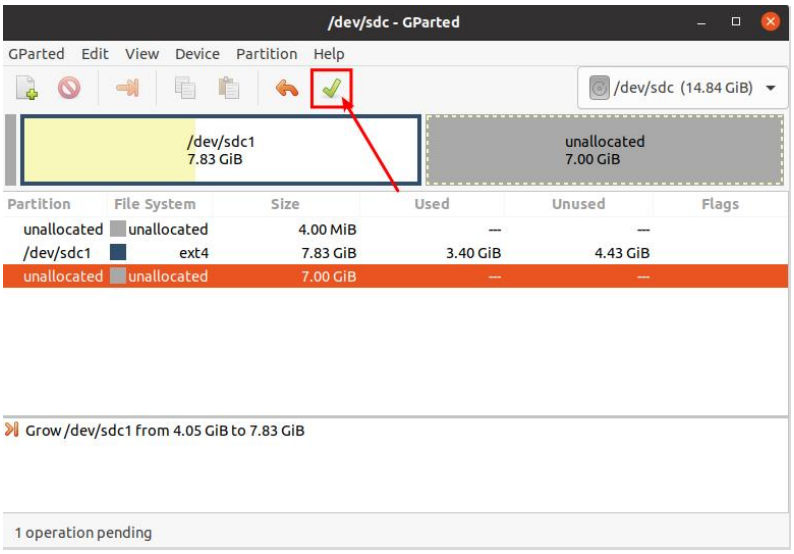
9) Then you can directly drag the location shown in the figure below to set the size of the capacity, or you can set the size of the rootfs partition by setting the number in **New sieze(MiB)**



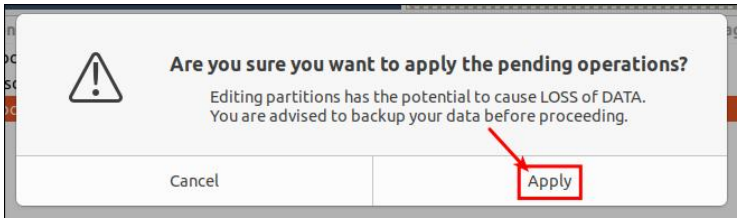
10) After setting the capacity, click **Resize/Move** in the lower right corner



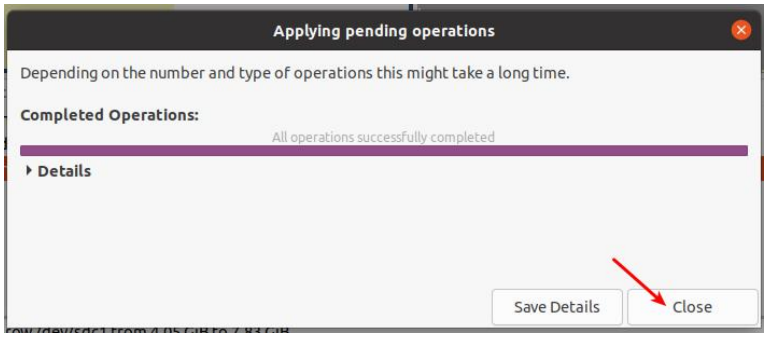
11) After confirming that it is correct, click the green button shown in the figure below √



12) Then select **Apply**, and the expansion of the rootfs partition will officially start



13) After the expansion is completed, click **Close** to close



14) Then you can pull out the micro SD card, insert it into the development board and start it up. After entering the Linux system of the development board, if you can use the **df -h** command to see that the size of the rootfs partition is the same as the size set before, it means that the size has been reduced. capacity success

```
root@orangepi:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            925M     0  925M   0% /dev
tmpfs           199M  3.2M  196M   2% /run
/dev/mmcblk0p1  7.7G  3.2G  4.4G  42% /
```

3. 7. Network connection test

3. 7. 1. Ethernet port test

1) First, insert one end of the network cable into the Ethernet interface of the development board, and connect the other end of the network cable to the router, and ensure that the network is unblocked

2) After the system starts, it will automatically assign an IP address to the Ethernet card through **DHCP, without any other configuration**

3) The command to view the IP address in the Linux system of the development board is as follows:

Please do not copy the following commands. For example, the network node name in debian12 is end0, and the following commands need to be changed to ip a s end0.

```
orangepi@orangepi:~$ ip a s eth0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
```

```
UP group default qlen 1000
    link/ether 5e:ac:14:a5:93:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.16/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
       valid_lft 259174sec preferred_lft 259174sec
    inet6 240e:3b7:3240:c3a0:e269:8305:dc08:135e/64 scope global dynamic noprefixroute
       valid_lft 259176sec preferred_lft 172776sec
    inet6 fe80::957d:bbbd:4928:3604/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

There are three ways to check the IP address after the development board starts:

1. Connect the HDMI monitor, then log in to the system and use the ip a s eth0 command to view the IP address

2. Enter the ip a s eth0 command in the debugging serial terminal to view the IP address

3. If there is no debugging serial port and no HDMI display, you can also check the IP address of the development board's network port through the router's management interface. However, in this method, some people often cannot see the IP address of the development board normally. If you can't see it, the debug method looks like this:

A） First check whether the Linux system has started normally. If the green light of the development board is flashing, it is generally started normally. If only the red light is on, or the red and green lights are not on, it means that the system has not started normally;

B）Check whether the network cable is plugged in tightly, or try another network cable;

C）Try another router (I have encountered many problems with the router, such as the router cannot assign the IP address normally, or the IP address has been assigned normally but cannot be seen in the router);

D）If there is no router to replace, you can only connect to an HDMI display or use the debugging serial port to check the IP address.

In addition, it should be noted that the development board DHCP automatically assigns an IP address without any settings.

4) The command to test the network connectivity is as follows, the ping command can be interrupted through the shortcut key of **Ctrl+C**

Please do not copy the following commands. For example, the network node name in debian12 is end0, and the following command needs to be changed to ping www.baidu.com -I end0.

```
orangepi@orangepi:~$ ping www.baidu.com -I eth0
PING www.a.shifen.com (14.215.177.38) from 192.168.1.12 eth0: 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=56 time=6.74 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=56 time=6.80 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=56 time=6.26 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=56 time=7.27 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 6.260/6.770/7.275/0.373 ms
```

3.7.2. WIFI connection test

Please do not connect to WIFI by modifying the /etc/network/interfaces configuration file. There will be problems connecting to the WIFI network in this way.

3.7.2.1. The server image connects to WIFI through commands

When the development board is not connected to Ethernet, not connected to HDMI display, but only connected to the serial port, it is recommended to use the commands demonstrated in this section to connect to the WIFI network. Because nmtui can only display characters in some serial port software (such as minicom), and cannot display the graphical interface normally. Of course, if the development board is connected to an Ethernet or HDMI display, you can also use the commands demonstrated in this section to connect to the WIFI network.

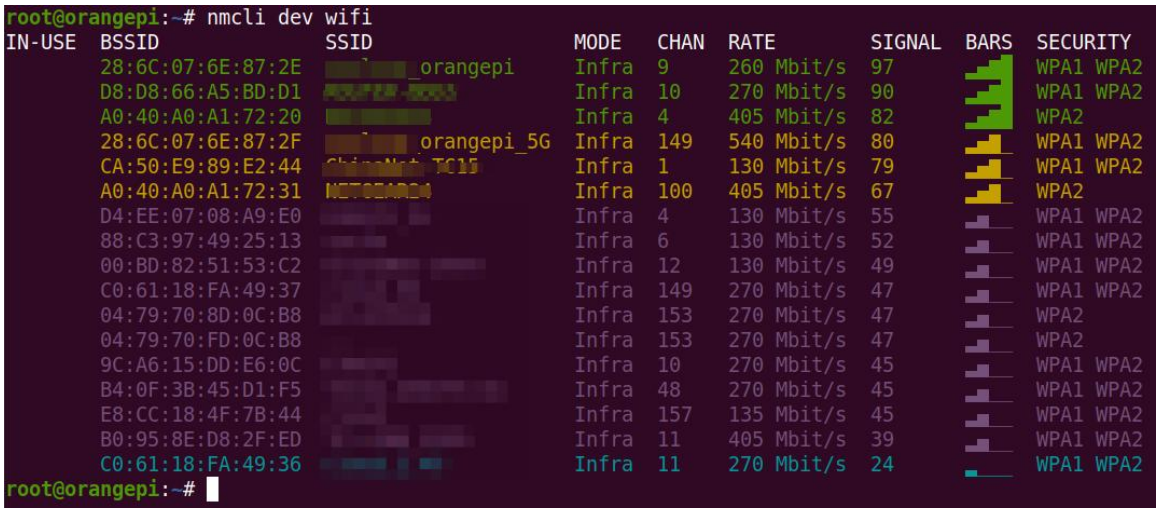
1) First log in to the linux system, there are the following three ways
   a. If the development board is connected with a network cable, **SSH remote login development board under Ubuntu**
   a. If the development board is connected to the debugging serial port, you can use

the serial port terminal to log in to the Linux system

b. If the development board is connected to the HDMI display, you can log in to the linux system through the terminal displayed on the HDMI

2) First use the **nmcli dev wifi** command to scan the surrounding WIFI hotspots

```
orangepi@orangepi:~$ nmcli dev wifi
```



3) Then use the **nmcli** command to connect to the scanned WIFI hotspot, where:

a. **wifi_name** Need to replace it with the name of the WIFI hotspot you want to connect to

b. **wifi_passwd** Need to change to the password of the WIFI hotspot you want to connect to

```
orangepi@orangepi:~$ sudo nmcli dev wifi connect wifi_name password wifi_passwd
Device 'wlan0' successfully activated with 'cf937f88-ca1e-4411-bb50-61f402eef293'.
```

4) You can view the IP address of wifi through the **ip addr show wlan0** command

```
orangepi@orangepi:~$ ip a s wlan0
11: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 23:8c:d6:ae:76:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
       valid_lft 259192sec preferred_lft 259192sec
    inet6 240e:3b7:3240:c3a0:c401:a445:5002:ccdd/64 scope global dynamic noprefixroute
```

```
        valid_lft 259192sec preferred_lft 172792sec
    inet6 fe80::42f1:6019:a80e:4c31/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

5) Use the **ping** command to test the connectivity of the wifi network, and the **ping** command can be interrupted through the shortcut key **Ctrl+C**

```
orangepi@orangepi:~$ ping www.orangepi.org -I wlan0
PING www.orangepi.org (182.92.236.130) from 192.168.1.49 wlan0: 56(84) bytes of data.
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=1 ttl=52 time=43.5 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=2 ttl=52 time=41.3 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=3 ttl=52 time=44.9 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=4 ttl=52 time=45.6 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=5 ttl=52 time=48.8 ms
^C
--- www.orangepi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

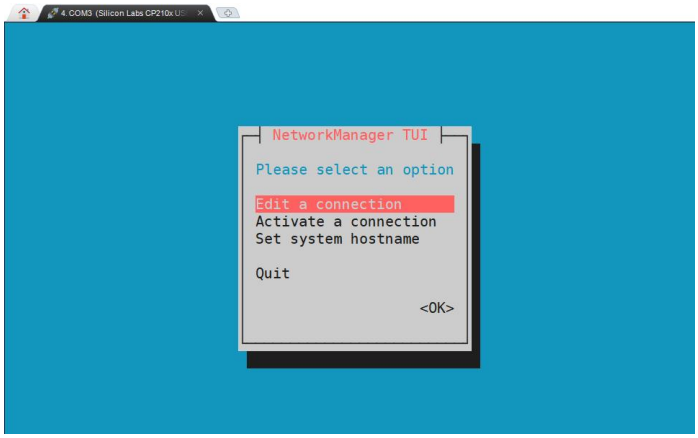
3.7.2.2. The server image connects to WIFI in a graphical way

1) First log in to the linux system, there are the following three ways
   a. **If the development board is connected with a network cable**, **SSH remote login development board under Ubuntu**
   b. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the linux system (please use MobaXterm for the serial port software, and the minicom cannot display the graphical interface)
   c. If the development board is connected to the HDMI display, you can log in to the linux system through the HDMI display terminal

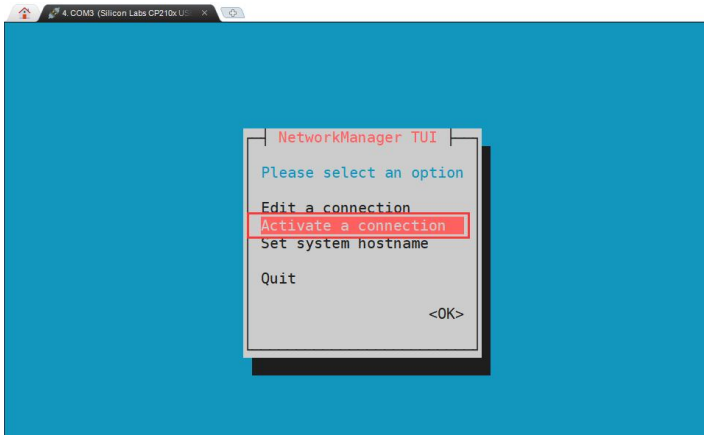
2) Then enter the nmtui command in the command line to open the wifi connection interface

```
orangepi@orangepi:~$ sudo nmtui
```

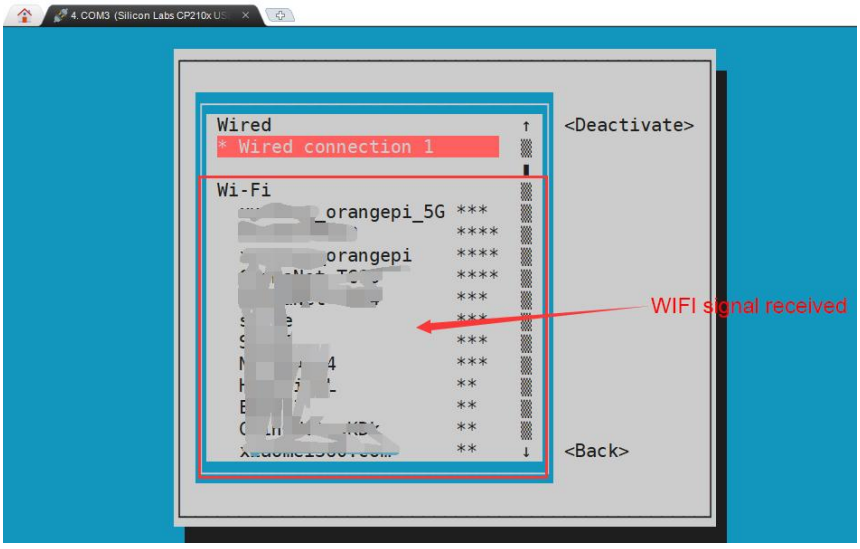
3) Enter the nmtui command to open the interface as shown below



4) Select **Activate a connect** and press Enter

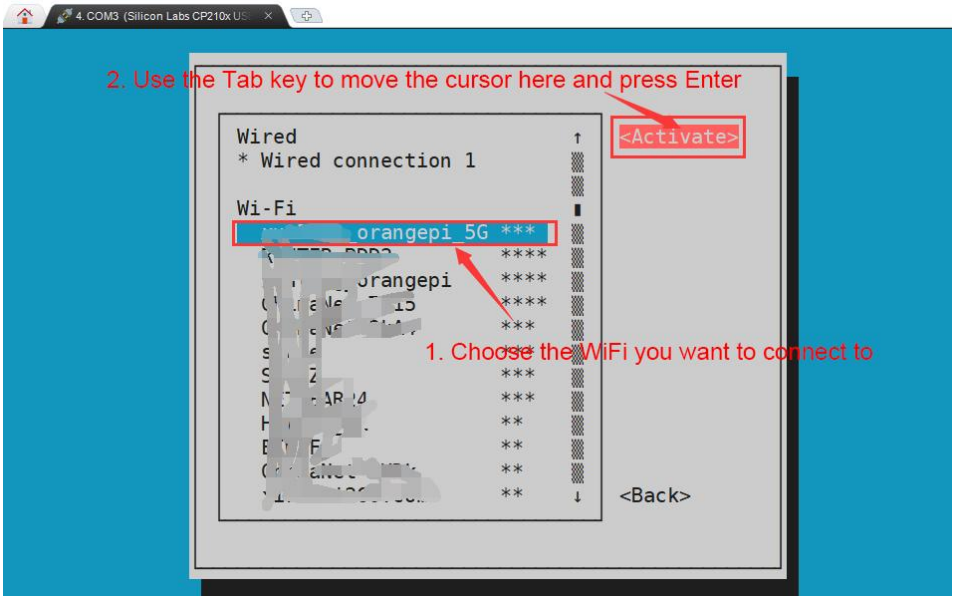


5) Then you can see all the searched WIFI hotspots

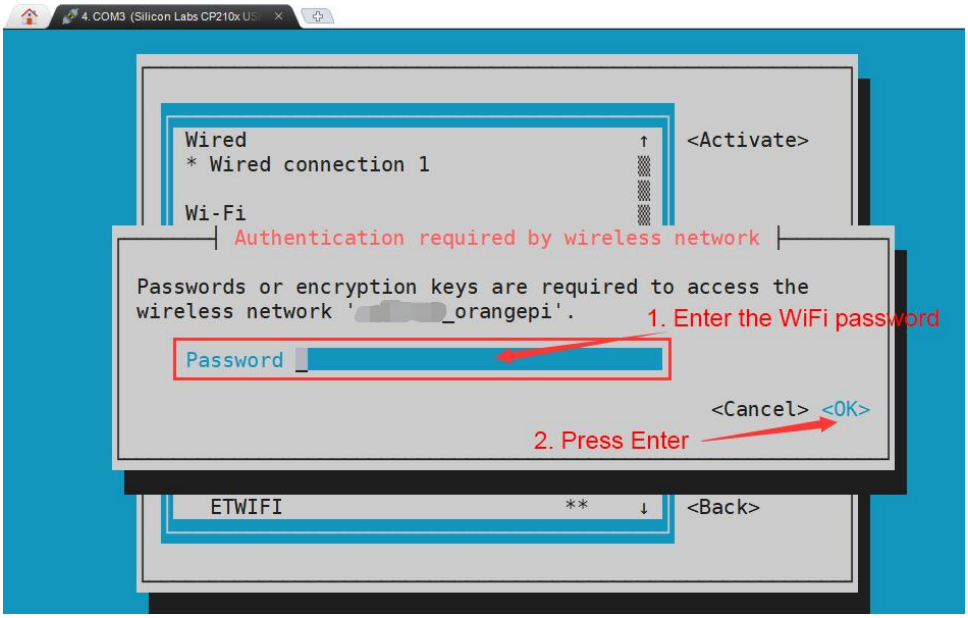


6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the

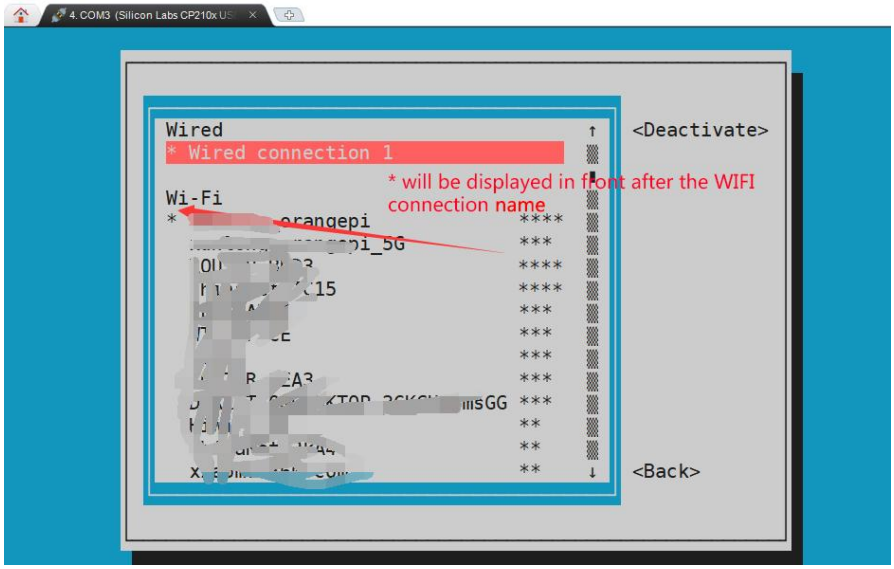
cursor on **Activate** and press Enter



7) Then a dialog box for entering a password will pop up, enter the corresponding password in **Password** and press Enter to start connecting to WIFI



8) After the WIFI connection is successful, a "*" will be displayed in front of the connected WIFI name



9) You can view the IP address of wifi through the **ip a s wlan0** command

```
orangepi@orangepi:~$ ip a s wlan0
11: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 24:8c:d3:aa:76:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
       valid_lft 259069sec preferred_lft 259069sec
    inet6 240e:3b7:3240:c4a0:c401:a445:5002:ccdd/64 scope global dynamic noprefixroute
       valid_lft 259071sec preferred_lft 172671sec
    inet6 fe80::42f1:6019:a80e:4c31/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

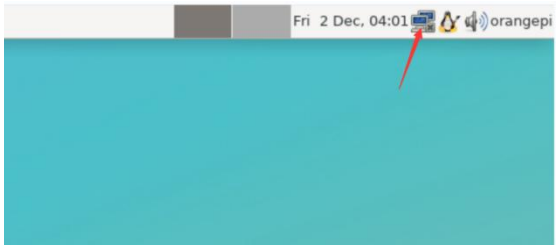
10) Use the **ping** command to test the connectivity of the wifi network, and the **ping** command can be interrupted through the shortcut key **Ctrl+C**

```
orangepi@orangepi:~$ ping www.orangepi.org -I wlan0
PING www.orangepi.org (182.92.236.130) from 192.168.1.49 wlan0: 56(84) bytes of data.
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=1 ttl=52 time=43.5 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=2 ttl=52 time=41.3 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=3 ttl=52 time=44.9 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=4 ttl=52 time=45.6 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=5 ttl=52 time=48.8 ms
```

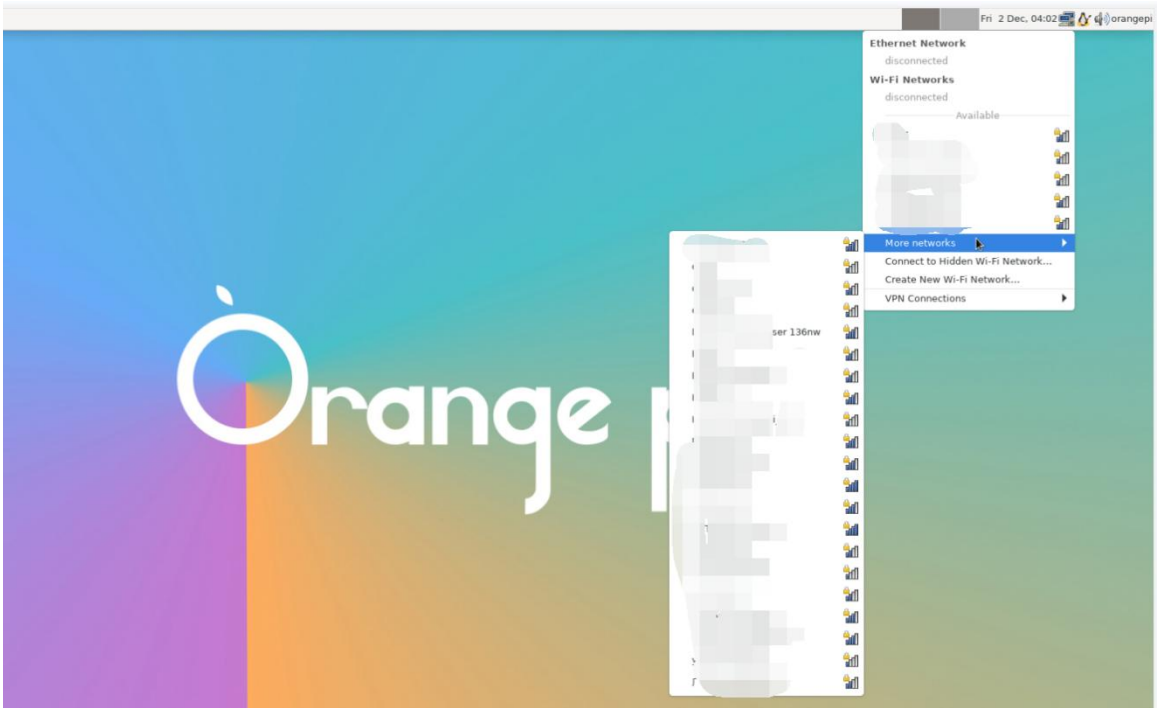
```
^C
--- www.orangepi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

3.7.2.3. Test method of desktop image

1) Click the network configuration icon in the upper right corner of the desktop (please do not connect the network cable when testing WIFI)



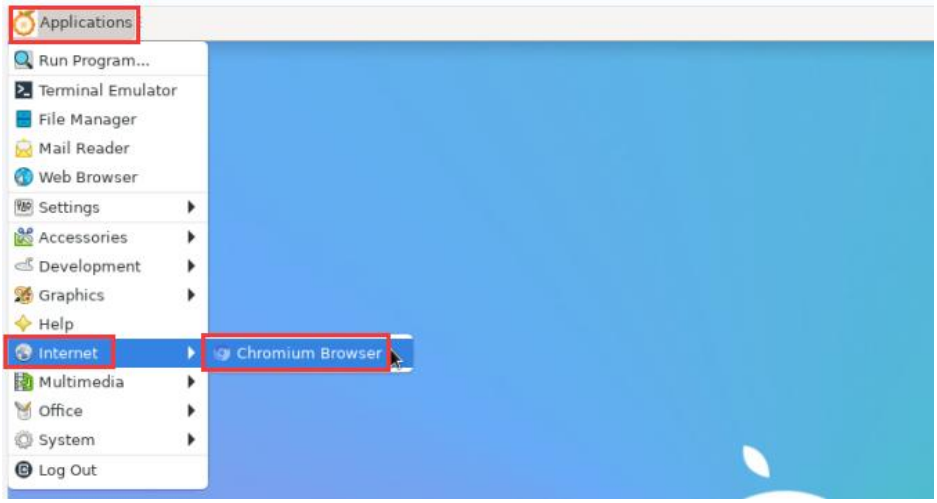
2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to



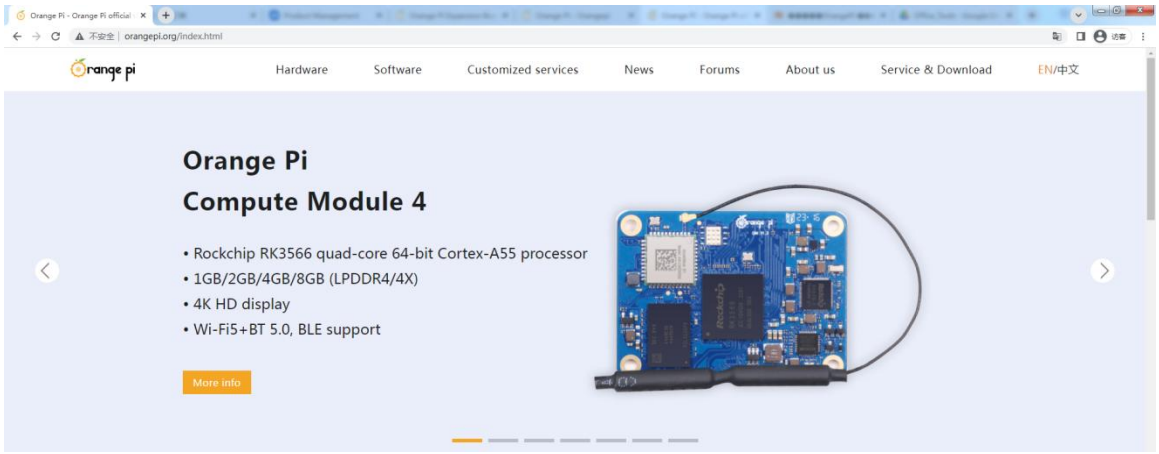
3) Then enter the password of the WIFI hotspot, and then click **Connect** start connecting

to WIFI

4) After connecting to WIFI, you can open the browser to check whether you can access the Internet. The entrance of the browser is shown in the figure below



5) If you can open other web pages after opening the browser, it means that the WIFI connection is normal



3. 7. 3. The method of creating WIFI hotspot through create_ap

create_ap is a script that helps quickly create WIFI hotspots on Linux, and supports bridge and NAT modes. It can automatically combine hostapd, dnsmasq and iptables to complete the setting of WIFI hotspots, avoiding complex configuration for users. The github address is as follows:

https://github.com/oblique/create_ap

The Linux image released by OPi has pre-installed the create_ap script. You can create a WIFI hotspot through the create_ap command. The basic command format of create_ap is as follows:

create_ap [options] <wifi-interface> [<interface-with-internet>] [<access-point-name> [<passphrase>]]

*** options： You can use this parameter to specify the encryption method, the frequency band of the WIFI hotspot, the bandwidth mode, the network sharing method, etc. You can get the options through create_ap -h**
*** wifi-interface：The name of the wireless network card**
*** interface-with-internet： The name of the network card that can be connected to the Internet, generally eth0**
*** access-point-name： hotspot name**
*** passphrase：hotspot password**

3. 7. 3. 1. create_ap method to create WIFI hotspot in NAT mode

1）Enter the following command to create a WIFI hotspot named **orangepi** and password **orangepi** in NAT mode

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --no-virt
```

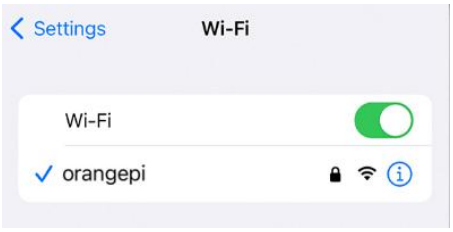
2）If the following information is output, it means that the WIFI hotspot is created successfully

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --no-virt
Config dir: /tmp/create_ap.wlan0.conf.TQkJtsz1
PID: 26139
```

```
Network Manager found, set wlan0 as unmanaged device... DONE
Sharing Internet using method: nat
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.TQkJtsz1/hostapd_ctrl
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA ce:bd:9a:dd:a5:86 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED ce:bd:9a:dd:a5:86
wlan0: STA ce:bd:9a:dd:a5:86 RADIUS: starting accounting session
D4FBF7E5C604F169
wlan0: STA ce:bd:9a:dd:a5:86 WPA: pairwise key handshake completed (RSN)
wlan0: EAPOL-4WAY-HS-COMPLETED ce:bd:9a:dd:a5:86
```

3）Take out the mobile phone at this time, you can find the WIFI hotspot named **orangepi** created by the development board in the searched WIFI list, and then you can click **orangepi** to connect to the hotspot, the password is set above **orangepi**

4）After the connection is successful, the display is as shown in the figure below



5）In NAT mode, the wireless device connected to the hotspot of the development board requests an IP address from the DHCP service of the development board, so there will be two different network segments, for example, the IP of the development board is 192.168.1.X

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.150  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::938f:8776:5783:afa2  prefixlen 64  scopeid 0x20<link>
        ether 4a:a0:c8:25:42:82  txqueuelen 1000  (Ethernet)
        RX packets 25370  bytes 2709590 (2.7 MB)
        RX errors 0  dropped 50  overruns 0  frame 0
        TX packets 3798  bytes 1519493 (1.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 83
```

By default, the DHCP service of the development board will assign an IP address of **192.168.12.0/24** to the device connected to the hotspot. At this time, click on the connected WIFI hotspot **orangepi**, and then you can see that the IP address of the mobile phone is **192.168.12.X**

6）If you want to specify a different network segment for the connected device, you can specify it through the -g parameter, such as specifying the network segment of the access point AP through the -g parameter as 192.168.2.1

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi -g 192.168.2.1 --no-virt
```

At this time, after connecting to the hotspot through the mobile phone, click on the connected WIFI hotspot **orangepi**, and then you can see that the IP address of the mobile phone is **192.168.2.X**.

7）If the **--freq-band** parameter is not specified, the hotspot created by default is in the 2.4G frequency band. If you want to create a hotspot in the 5G frequency band, you can specify the**--freq-band 5** parameter. The specific command is as follows

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --freq-band 5    --no-virt
```

8）If you need to hide the SSID, you can specify the **--hidden** parameter, the specific command is as follows

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --hidden    --no-virt
```

At this time, the mobile phone cannot search for the WIFI hotspot. You need to manually specify the name of the WIFI hotspot and enter the password to connect to the WIFI hotspot.

3.7.3.2. create_ap method to create WIFI hotspot in bridge mode

1）Enter the following command to create a WIFI hotspot named **orangepi** and password **orangepi** in bridge mode

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi    --no-virt
```

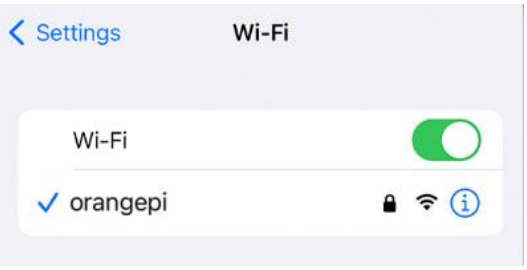
2）If the following information is output, it means that the WIFI hotspot is created successfully

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi    --no-virt
Config dir: /tmp/create_ap.wlan0.conf.zAcFlYTx
PID: 27707
Network Manager found, set wlan0 as unmanaged device... DONE
Sharing Internet using method: bridge
Create a bridge interface... br0 created.
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.zAcFlYTx/hostapd_ctrl
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA ce:bd:9a:dd:a5:86 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED ce:bd:9a:dd:a5:86
wlan0: STA ce:bd:9a:dd:a5:86 RADIUS: starting accounting session
937BF40E51897A7B
wlan0: STA ce:bd:9a:dd:a5:86 WPA: pairwise key handshake completed (RSN)
```

```
wlan0: EAPOL-4WAY-HS-COMPLETED ce:bd:9a:dd:a5:86
```

3）Take out the mobile phone at this time, and you can find the WIFI hotspot named orangepi created by the development board in the searched WIFI list, and then you can click **orangepi** to connect to the hotspot, and the password is the **orangepi** set above

4）After the connection is successful, the display is as shown in the figure below

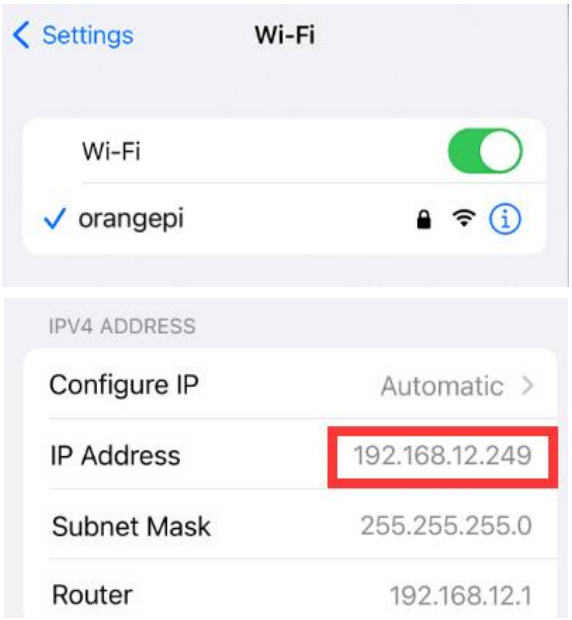


5）In bridge mode, the wireless device connected to the hotspot of the development board also requests an IP address from the DHCP service of the main router (the router connected to the development board), for example, the IP of the development board is **192.168.1.X**

```
orangepi@orangepi:~$ sudo ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.150  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::938f:8776:5783:afa2  prefixlen 64  scopeid 0x20<link>
        ether 4a:a0:c8:25:42:82  txqueuelen 1000  (Ethernet)
        RX packets 25370  bytes 2709590 (2.7 MB)
        RX errors 0  dropped 50  overruns 0  frame 0
        TX packets 3798  bytes 1519493 (1.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
device interrupt 83
```

The IP of the device connected to the WIFI hotspot is also assigned by the main router, so the mobile phone connected to the WIFI hotspot and the development board are in the same network segment. At this time, click on the connected WIFI hotspot orangepi, and then you can see the IP address of the mobile phone also **192.168.1.X**。



6）If the **--freq-band** y band. If you want to create a hotspot in the 5G frequency band, you can specify the **--freq-band 5** parameter. The specific command is as follows

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --freq-band 5    --no-virt
```

7）If you need to hide the SSID, you can specify the **--hidden** parameter, the specific command is as follows

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --hidden    --no-virt
```

At this time, the mobile phone cannot search for the WIFI hotspot. You need to manually specify the name of the WIFI hotspot and enter the password to connect to the WIFI hotspot.

3. 7. 4. How to set static IP address

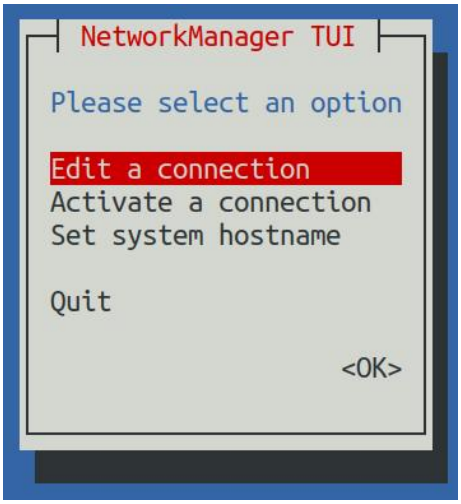
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file.

3. 7. 4. 1. Use the nmtui command to set a static IP address

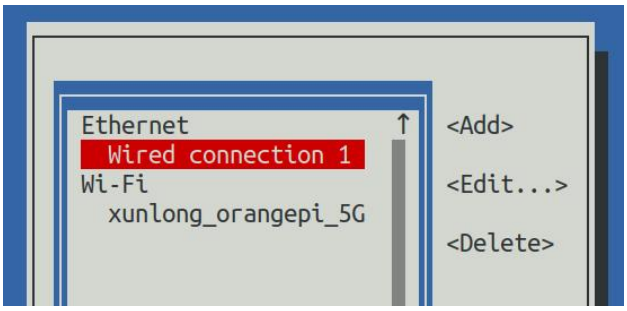
1) First run the **nmtui** command

```
orangepi@orangepi:~$ sudo nmtui
```

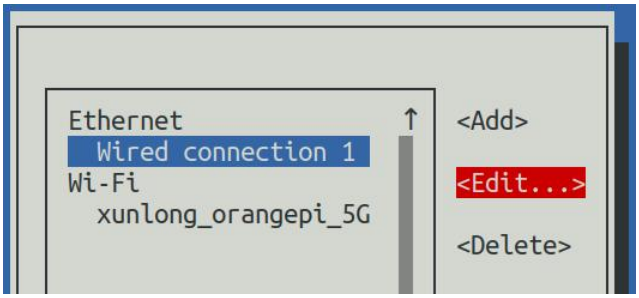
2) Then select **Edit a connection** and press Enter



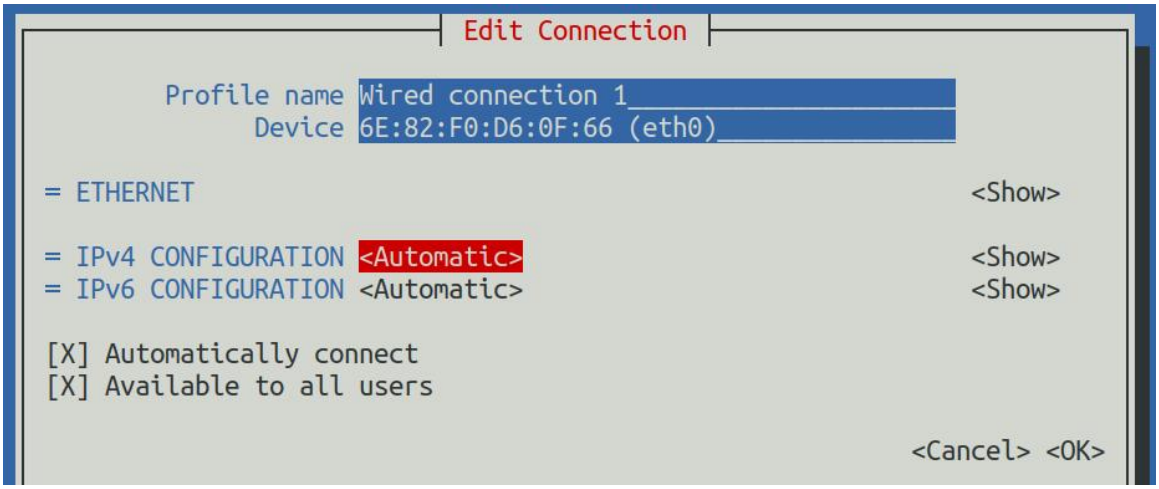
3) Then select the network interface that needs to set a static IP address, for example, to set the static IP address of the **Ethernet** interface, select **Wired connection 1**.



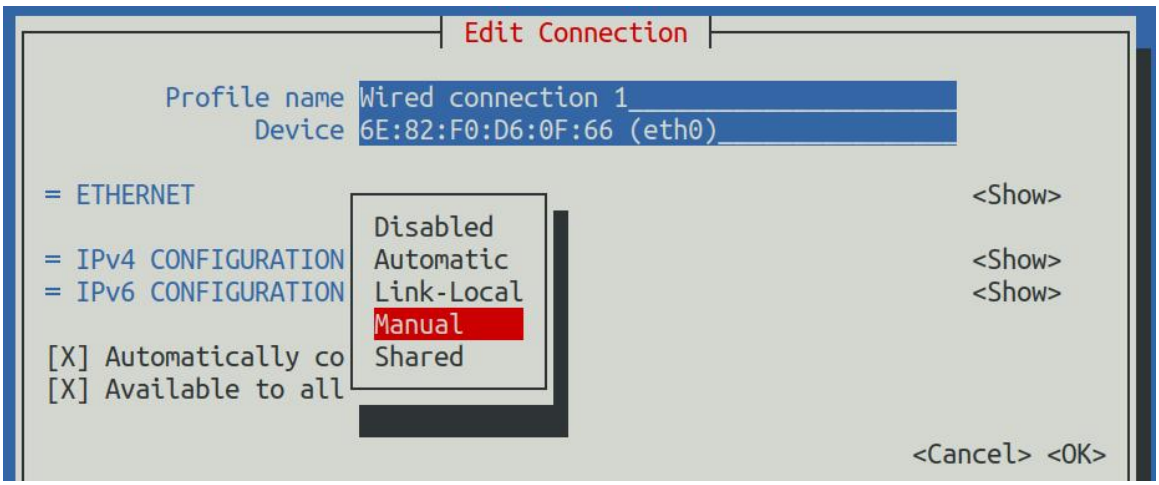
4) Then select **Edit** with the **Tab** key and press the Enter key



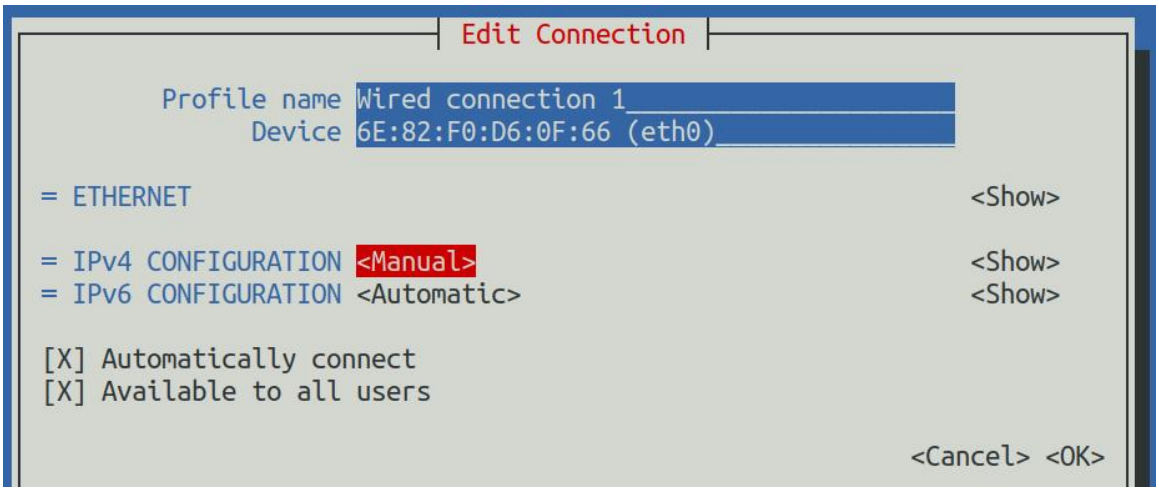
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4



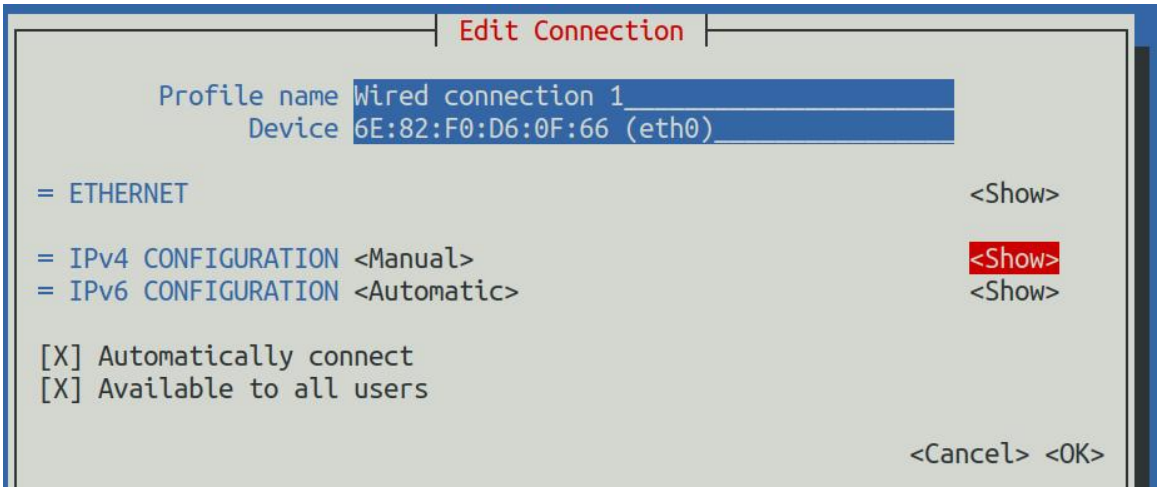
6) Then press Enter, select **Manual** through the up and down arrow keys, and press Enter to confirm



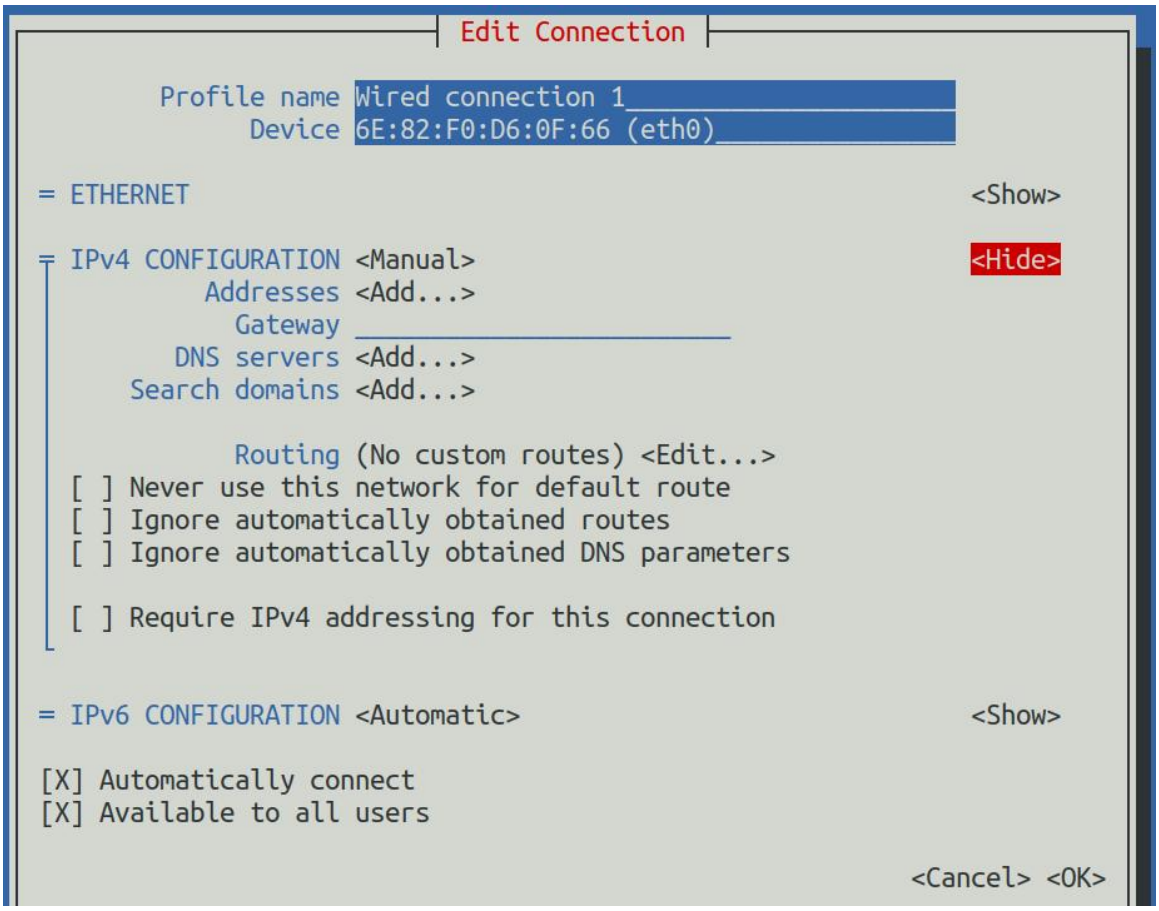
7) The display after selection is shown in the figure below



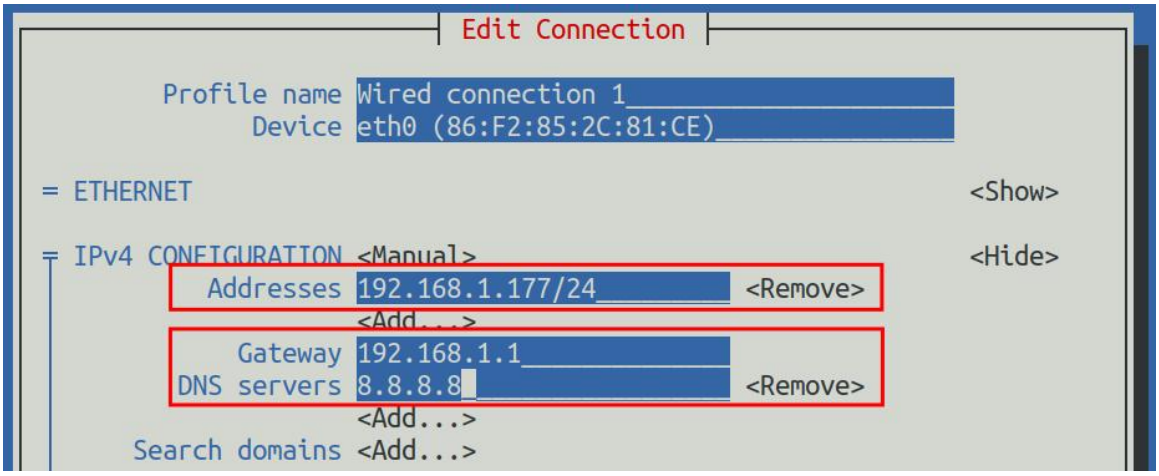
8) 8) Then move the cursor to **<Show>** via the Tab key



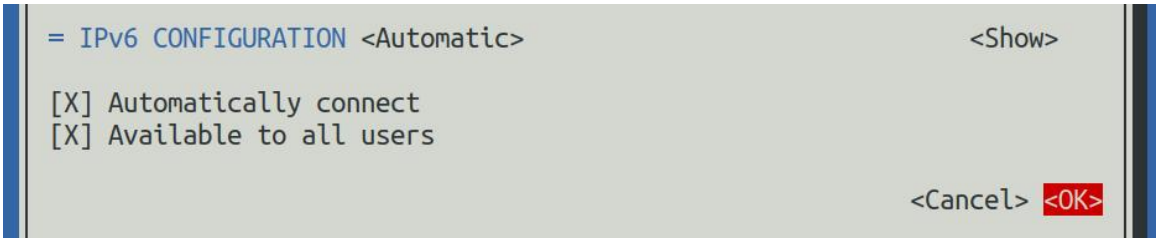
9) Then press Enter, the following setting interface will pop up after entering



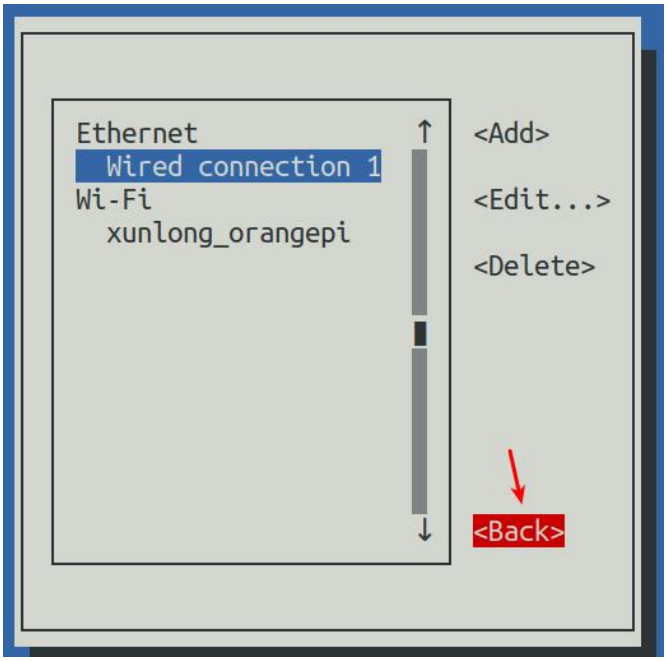
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the position shown in the figure below (there are many other setting options in it, please explore by yourself), **please set according to your specific needs, The values set in the image below are just an example**



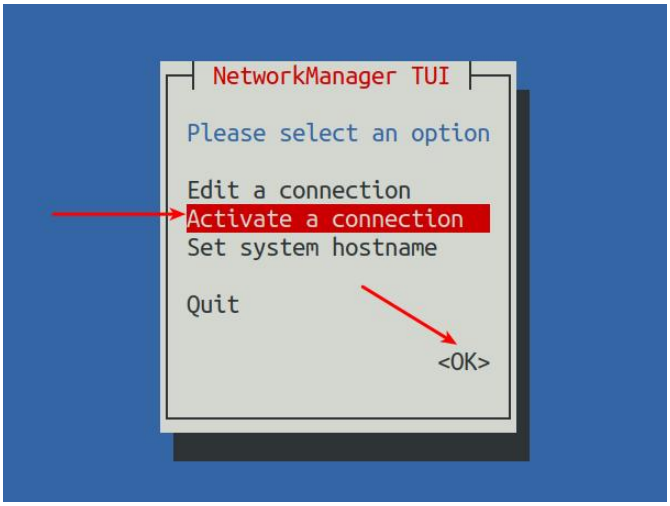
11) After setting, move the cursor to **<OK>** in the lower right corner, and press Enter to confirm



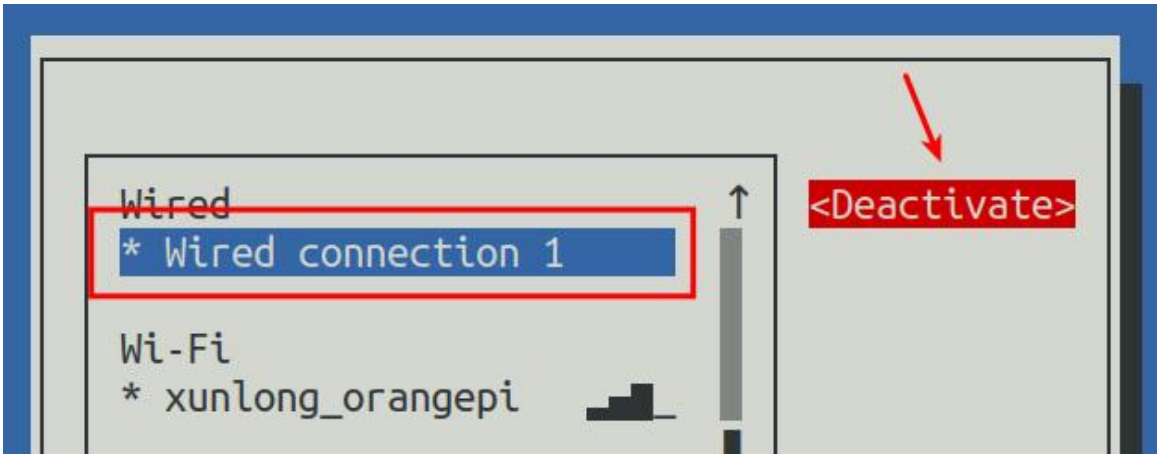
12) Then click **<Back>** to return to the previous selection interface



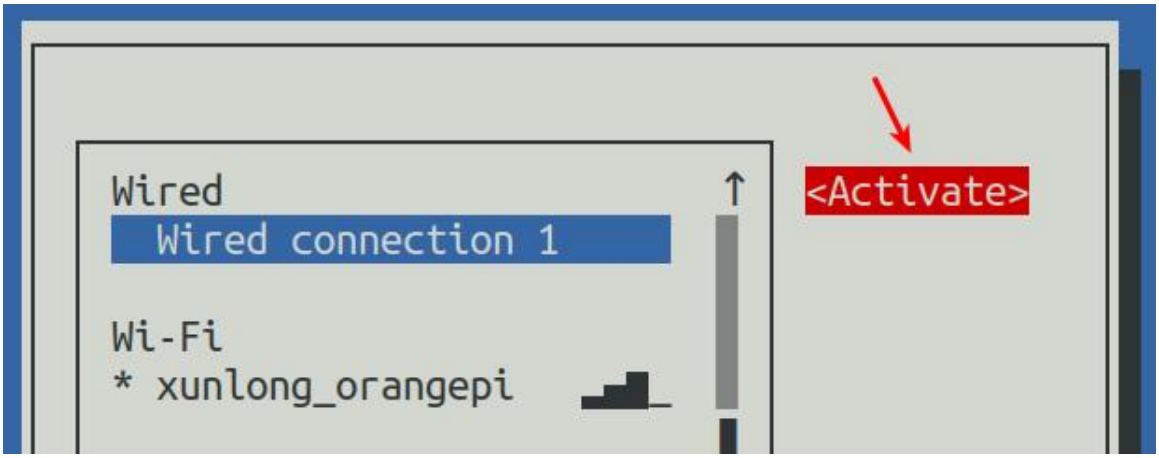
13) Then select **Activate a connection**, then move the cursor to **<OK>**, and finally click Enter



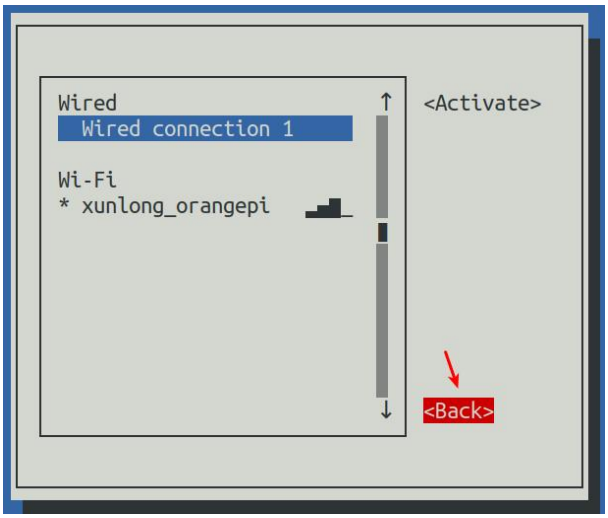
14) Then select the network interface that needs to be set, such as **Wired connection 1**, then move the cursor to <Deactivate>, and press the Enter key to disable **Wired connection 1**

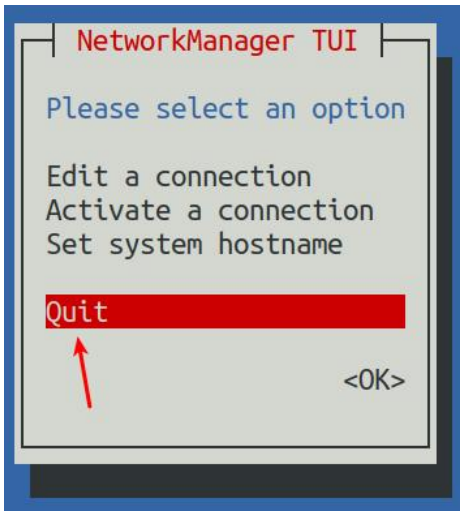


15) Then please do not move the cursor, and then press the Enter key to re-enable **Wired connection 1**, so that the static IP address set earlier will take effect



16) Then you can exit nmtui through the **<Back>** and **Quit** buttons





17) Then through **ip a s eth0**, you can see that the IP address of the network port has changed to the static IP address set earlier

Note that in the following commands, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ ip a s eth0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 5e:ac:14:a5:92:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.177/24 brd 192.168.1.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 241e:3b8:3240:c3a0:e269:8305:dc08:135e/64 scope global dynamic noprefixroute
```

```
       valid_lft 259149sec preferred_lft 172749sec
    inet6 fe80::957d:bbbe:4928:3604/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

18) Then you can test the connectivity of the network to check whether the IP address is configured OK, and the **ping** command can be interrupted through the shortcut key **Ctrl+C**

Note that in the following commands, I is capital i, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ ping 192.168.1.177 -I eth0
PING 192.168.1.47 (192.168.1.47) from 192.168.1.188 eth0: 56(84) bytes of data.
64 bytes from 192.168.1.47: icmp_seq=1 ttl=64 time=0.233 ms
64 bytes from 192.168.1.47: icmp_seq=2 ttl=64 time=0.263 ms
64 bytes from 192.168.1.47: icmp_seq=3 ttl=64 time=0.273 ms
64 bytes from 192.168.1.47: icmp_seq=4 ttl=64 time=0.269 ms
64 bytes from 192.168.1.47: icmp_seq=5 ttl=64 time=0.275 ms
^C
--- 192.168.1.47 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4042ms
rtt min/avg/max/mdev = 0.233/0.262/0.275/0.015 ms
```

3.7.4.2. Use the nmcli command to set a static IP address

1) If you want to set the static IP address of the network port, please insert the network cable into the development board first. **If you need to set the static IP address of WIFI**, please connect the WIFI first, and then start to set the static IP address

2) Then you can view the name of the network device through the **nmcli con show** command, as shown below

a. **orangepi** is the name of the WIFI network interface (the name is not necessarily the same)

b. **Wired connection 1** is the name of the Ethernet interface

```
orangepi@orangepi:~$ nmcli con show
NAME                UUID                                  TYPE      DEVICE
orangepi            cfc4f922-ae48-46f1-84e1-2f19e9ec5e2a  wifi      wlan0
```

```
Wired connection 1  9db058b7-7701-37b8-9411-efc2ae8bfa30  ethernet  eth0
```

3) Then enter the following command, where

a. **"Wired connection 1"** means to set the static IP address of the Ethernet port. If you need to set the static IP address of the WIFI, please modify it to the corresponding name of the WIFI network interface (you can get it through the **nmcli con show** command)

b. **ipv4.addresses** is followed by the static IP address to be set, which can be modified to the value you want to set

c. **ipv4.gateway** represents the address of the gateway

```
orangepi@orangepi:~$ sudo nmcli con mod "Wired connection 1"  \
ipv4.addresses "192.168.1.110"  \
ipv4.gateway "192.168.1.1"  \
ipv4.dns "8.8.8.8"  \
ipv4.method "manual"
```

4) Then restart the linux system

```
orangepi@orangepi:~$ sudo reboot
```

5) Then re-enter the linux system and use the **ip addr show eth0** command to see that the IP address has been set to the desired value

```
orangepi@orangepi:~$ ip addr show eth0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 5e:ae:14:a5:91:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.110/32 brd 192.168.1.110 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 240e:3b7:3240:c3a0:97de:1d01:b290:fe3a/64 scope global dynamic
noprefixroute
       valid_lft 259183sec preferred_lft 172783sec
    inet6 fe80::3312:861a:a589:d3c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

3.7.5. The method of setting the Linux system to automatically connect to the network for the first time

The development board has an Ethernet port. If you want to remotely log in to

the Linux system of the development board through the Ethernet port, you only need to plug in a network cable that can access the Internet normally to the Ethernet port. Assign an IP address, and then we can obtain the IP address of the Ethernet port through the HDMI screen, serial port or view the background of the router, and then we can log in to the Linux system remotely.

The development board also has wireless WIFI. If you want to remotely log in to the Linux system of the development board through WIFI, you need to remotely log in to the Linux system through the IP address of the Ethernet port ssh and then connect to WIFI through commands, or through commands on the HDMI screen or serial port. Connect to WIFI.

But if there is no HDMI screen and serial port module, although there is a network cable, the IP address of the development board cannot be checked through the background of the router. Or there is no HDMI screen, serial port module and network cable, and only WIFI can be connected, then you can use the method introduced in this section to automatically connect to WIFI and also set the static IP address of WIFI or automatically set the static IP address of the Ethernet port.

To use the method in this section, you first need to prepare a Linux system machine. For example, a computer or a virtual machine with Ubuntu system installed.

Why do you need a Linux system machine, because the root file system of the development board Linux system burned in the micro SD card is in ext4 format, and the Linux system machine can mount it normally, and then modify the configuration files in it.

If you want to modify it in the Windows system, you can use the software Paragon ExtFS for Windows. Since this software needs to be paid, and there is no similar free software that is easy to use, I will not demonstrate it here.

In addition, if you have any problems with Paragon ExtFS for Windows software, please solve it yourself, we will not answer questions.

1) First burn the Linux image of the development board you want to use into the micro SD card, and then use a card reader to insert the micro SD card that has burned the Linux image of the development board into a machine with a Linux system (such as a machine with an Ubuntu system installed) computer, the following uses Ubuntu computer as an

example to demonstrate)

2) When the micro SD card is inserted into the Ubuntu computer, the Ubuntu computer will generally automatically mount the partition of the Linux root file system in the micro SD card. You can know from the following command that **/media/test/opi_root** is the Linux root file in the micro SD card The path where the system is mounted

```
test@test:~$ df -h | grep "media"
/dev/sdd1    1.4G    1.2G    167M    88% /media/test/opi_root
test@test:~$ ls /media/test/opi_root
bin  boot  dev  etc  home  lib  lost+found  media  mnt  opt  proc  root  run
sbin  selinux  srv  sys  tmp  usr  var
```

3) Then enter the **/boot** directory of the Linux system burned in the micro SD card

```
test@test:~$ cd /media/test/opi_root/boot/
```

4) Then copy the **orangepi_first_run.txt.template** to **orangepi_first_run.txt**. Through the orangepi_first_run.txt configuration file, you can set the Linux system on the development board to automatically connect to a WIFI hotspot when it starts for the first time, and you can also set the WIFI or Ethernet port. static IP address

```
test@test:/media/test/opi_root/boot$ sudo cp orangepi_first_run.txt.template orangepi_first_run.txt
```

5) You can open the orangepi_first_run.txt file with the following command, and then you can view and modify the contents

```
test@test:/media/test/opi_root/boot$ sudo vim orangepi_first_run.txt
```

6) Instructions for using variables in the orangepi_first_run.txt file

a. **FR_general_delete_this_file_after_completion** variable is used to set whether to delete the orangepi_first_run.txt file after the first startup. The default is 1, that is, delete. If it is set to 0, orangepi_first_run.txt will be renamed to orangepi_first_run.txt after the first startup .old, generally keep the default value

b. **FR_net_change_defaults** variable is used to set whether to change the default network settings, this must be set to 1, otherwise all network settings will not take effect

c. **FR_net_ethernet_enabled** variable is used to control whether to enable the configuration of the Ethernet port. If you need to set the static IP address of the

Ethernet port, please set it to 1

d. **FR_net_wifi_enabled** variable is used to control whether to enable WIFI configuration. If you need to set the development board to automatically connect to WIFI hotspots, you must set it to 1. In addition, please note that if this variable is set to 1, the setting of the Ethernet port will be will fail. That is to say, the WIFI and Ethernet ports cannot be set at the same time (why, because it is not necessary...)
e. **FR_net_wifi_ssid** variable is used to set the name of the WIFI hotspot you want to connect to
f. **FR_net_wifi_key** variable is used to set the password of the WIFI hotspot you want to connect to
g. **FR_net_use_static** variable is used to set whether to set the static IP address of WIFI or Ethernet port
h. **FR_net_static_ip** variable is used to set static IP address, please set according to your actual situation
i. **FR_net_static_gateway** variable is used to set the gateway, please set it according to your actual situation

7) The following demonstrates several specific setting examples:
   a. For example, if you want the Linux system of the development board to automatically connect to the WIFI hotspot after the first startup, you can set it like this:
      a) Set **FR_net_change_defaults** to **1**
      b) Set **FR_net_wifi_enabled** to **1**
      c) Set **FR_net_wifi_ssid** to the name of the WIFI hotspot you want to connect to
      d) Set **FR_net_wifi_key** to the password of the WIFI hotspot you want to connect to

   b. For example, if you want the Linux system of the development board to automatically connect to the WIFI hotspot after the first startup, and set the IP address of the WIFI to a specific static IP address (so that when the Linux system is started, you can directly use the set static IP address to ssh remotely Log in to the development board without checking the IP address of the development board through the background of the router), you can set it like this:
      a) Set **FR_net_change_defaults** to **1**

b) Set **FR_net_wifi_enabled** to **1**
c) Set **FR_net_wifi_ssid** to the name of the WIFI hotspot you want to connect to
d) Set **FR_net_wifi_key** to the password of the WIFI hotspot you want to connect to
e) Set **FR_net_use_static** to **1**
f) Set **FR_net_static_ip** to the desired IP address
g) Set **FR_net_static_gateway** to the corresponding gateway address

c. For example, if you want to automatically set the IP address of the Ethernet port to the desired static IP address after the Linux system of the development board starts for the first time, you can set it like this:
a) Set **FR_net_change_defaults** to **1**
b) Set **FR_net_ethernet_enabled** to **1**
c) Set **FR_net_use_static** to **1**
d) Set **FR_net_static_ip** to the desired IP address
e) Set **FR_net_static_gateway** to the corresponding gateway address

8) After modifying the orangepi_first_run.txt file, you can exit the /boot directory of the development board Linux system in the micro SD card, uninstall the micro SD card, and then insert the micro SD card into the development board to start

9) If you have not set a static IP address, you still need to check the IP address through the background of the router. If you have set a static IP address, you can ping the set static IP address on the computer. If you can ping, it means that the system has started normally, and The network has also been set correctly, and then you can use the set IP address ssh to remotely log in to the Linux system of the development board

After the Linux system of the development board is started for the first time, orangepi_first_run.txt will be deleted or renamed to orangepi_first_run.txt.old. At this time, the orangepi_first_run.txt configuration file will be reset, and then the Linux system of the development board will be restarted, orangepi_first_run. The configuration in txt will not take effect again, because this configuration will only be effective when the Linux system is started for the first time after burning, please pay special attention to this point.

3. 8. SSH remote login development board

Linux systems enable ssh remote login by default and allow the root user to log in to the system. Before logging in with ssh, you first need to ensure that the Ethernet or wifi network is connected, and then use the ip addr command or check the router to obtain the IP address of the development board.

3. 8. 1. SSH remote login development board under Ubuntu

1) Obtain the IP address of the development board

2) Then you can remotely log in to the linux system through the ssh command

```
test@test:~$ ssh orangepi@192.168.1.xxx          (Need to be replaced with the IP address of the development board)
orangepi@192.168.1.xx's password:          （Enter the password here, the default password is orangepi）
```

Note that when entering the password, the specific content of the entered password will not be displayed on the screen, please do not think that there is any fault, just press Enter after inputting.

If you are prompted to refuse the connection, as long as you are using the image provided by Orange Pi, please do not suspect that the password orangepi is wrong, but find other reasons.

3) After successfully logging in to the system, the display is as shown in the figure below

```
test@test:~$ ssh orangepi@192.168.1.121
orangepi@192.168.1.121's password:

Welcome to Orange Pi 1.0.0 Bullseye with Linux 6.1.31-sun50iw9

System load:   39%              Up time:       21 min   Local users:   4
Memory usage:  31% of 1.45G     IP:            192.168.1.121
CPU temp:      54°C             Usage of /:    25% of 15G

Last login: Thu Jun  8 08:03:08 2023 from 192.168.1.119
orangepi@orangepi:~$
```

If ssh fails to log in to the linux system normally, first check whether the IP address of the development board can be pinged. If the ping is ok, you can log in to the linux system through the serial port or HDMI display and then enter the following command on the development board and try again. Is it possible to connect:

root@orangepi:~# **reset_ssh.sh**

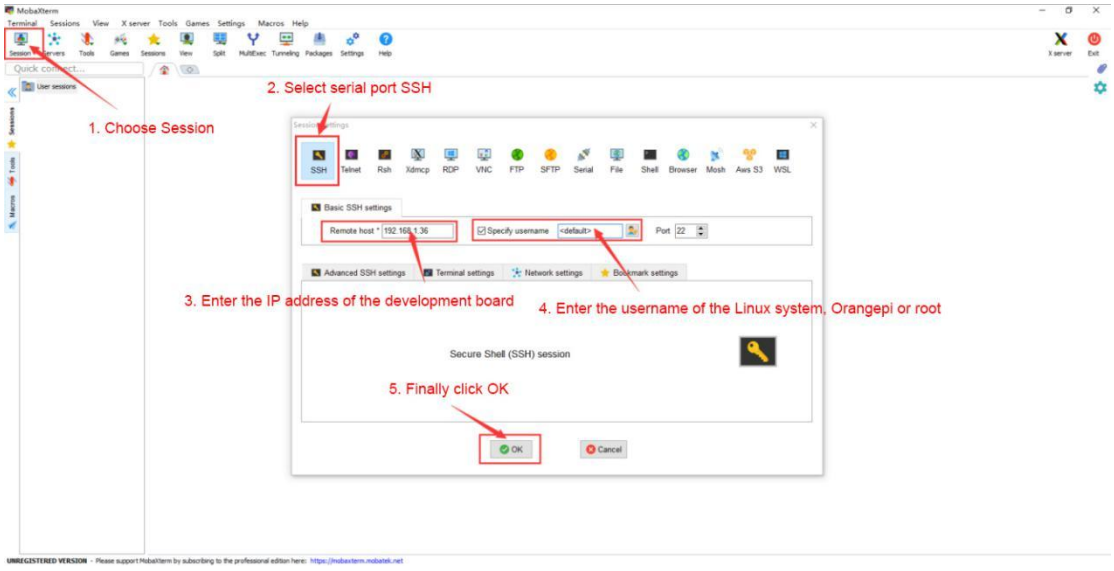
If it still doesn't work, try to reset the system.

3. 8. 2. SSH remote login development board under Windows

1) First obtain the IP address of the development board

2) Under Windows, you can use MobaXterm to remotely log in to the development board, first create a new ssh session

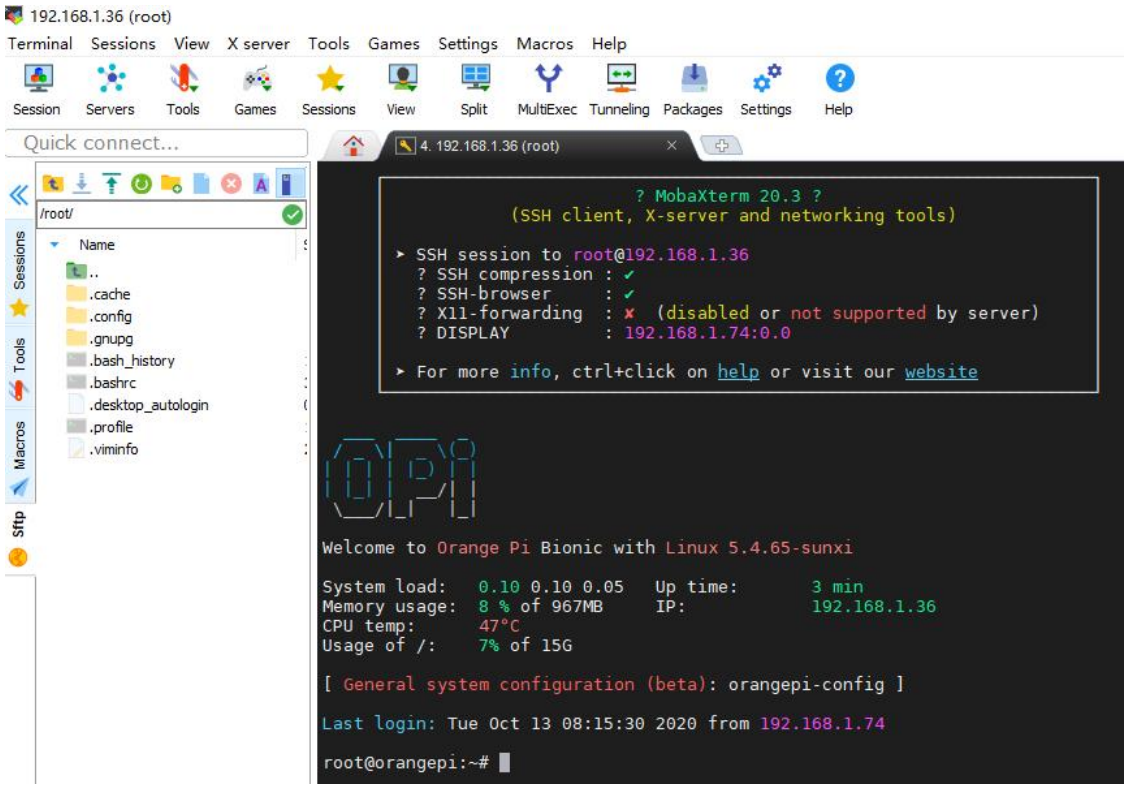
a. Open **Session**
b. Then select **SSH** in **Session Setting**
c. Then enter the IP address of the development board in the **Remote host**
d. Then enter the user name root or orangepi of the linux system in **Specify username**
e. Finally click **OK**



3) Then you will be prompted to enter a password. The default passwords for root and orangepi users are orangepi

Note that when entering the password, the specific content of the entered password will not be displayed on the screen, please do not think that there is any fault, just press Enter after inputting.

4) The display after successfully logging in to the system is shown in the figure below



3. 9. HDMI test

3. 9. 1. HDMI display test

1) Use a Micro HDMI to HDMI cable to connect the Orange Pi development board and HDMI display

2) After starting the linux system, if the HDMI monitor has image output, it means that the HDMI interface is working normally

Note that although many laptops have an HDMI interface, the HDMI interface of the notebook generally only has the output function, and does not have the function of HDMI IN, that is to say, the HDMI output of other devices cannot be displayed on the notebook screen.

When you want to connect the HDMI of the development board to the HDMI port of the laptop, please make sure that your laptop supports the HDMI IN function.

When the HDMI is not displayed, please check whether the HDMI cable is plugged in tightly. After confirming that there is no problem with the connection, you can change a different screen and try to see if it is displayed.

3. 9. 2. HDMI to VGA display test

1) First you need to prepare the following accessories

a. HDMI to VGA converter

b. A VGA cable and a Micro HDMI male to HDMI female conversion cable

c. A monitor or TV that supports VGA interface

2) HDMI to VGA display test as shown below

When using HDMI to VGA display, the development board and the Linux system of the development board do not need to make any settings, only the Micro HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI to VGA converter, VGA cable and monitor.

3. 9. 3. How to set HDMI resolution in Linux5.4 system

Note: This method is only applicable to systems with linux5.4 kernel.

1) There is a disp_mode variable in **/boot/orangepiEnv.txt** of the linux system, which can be used to set the resolution of the HDMI output. The default resolution of the linux system is 1080p60

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
console=both
disp_mode=1080p60
fb0_width=1920
fb0_height=1080
```

2) The values supported by the disp_mode variable are shown in the table below

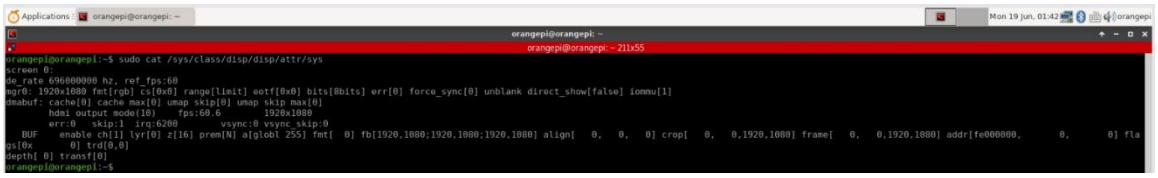
Supported values for disp_mode	HDMI resolution	HDMI refresh rate
480i	720x480	60
576i	720x480	50
480p	720x480	60
576p	720x576	60
720p50	1280x720	50
720p60	1280x720	60
1080i50	1920x1080	50
1080i60	1920x1080	60
1080p24	1920x1080	24
1080p50	1920x1080	50
1080p60	1920x1080	60

Note: Linux systems currently do not support 4K resolution.

3) Change the value of the disp_mode variable to the desired output resolution, and then restart the system, HDMI will output the set resolution

4) The method of viewing the HDMI output resolution is as follows. If the displayed resolution is the same as the set resolution, it means that the setting of the development board is correct.

```
orangepi@orangepi:~$ sudo cat /sys/class/disp/disp/attr/sys
```



3. 9. 4. How to Modify the Framebuffer Width and Height of Linux5.4 System

Note: This method is only applicable to systems with linux5.4 kernel.

There are two variables fb0_width and fb0_height in the **/boot/orangepiEnv.txt** of the linux system, which can be used to set the width and height of the Framebuffer. The default setting of the linux system is fb0_width=1920 and fb0_height=1080.

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
console=both
disp_mode=1080p60
fb0_width=1920
fb0_height=1080
```

The **reference** values corresponding to different resolutions of fb0_width and fb0_height are as follows:

HDMI resolution	fb0_width	fb0_height
480p	720	480
576p	720	576
720p	1280	720
1080p	1920	1080

Under the same HDMI resolution, the larger the value of fb0_width and fb0_height, the smaller the text displayed on the screen, and the smaller the value of fb0_width and fb0_height, the larger the text displayed on the screen.

3.9.5. Framebuffer Cursor Setting

1) The softcursor used by Framebuffer, the method of setting the cursor to blink or not is as follows

```
root@orangepi:~# echo 1 > /sys/class/graphics/fbcon/cursor_blink #cursor blinking
root@orangepi:~# echo 0 > /sys/class/graphics/fbcon/cursor_blink #cursor not blinking
```

2) If you need to hide the cursor, you can add **vt.global_cursor_default=0** in the **extraargs** variable of **/boot/orangepiEnv.txt** (the value of extraargs will be assigned to the **bootargs** environment variable and finally passed to the kernel) (if **vt.global_cursor_default=1**, the cursor will be displayed) , and then restart the system to see that the cursor has disappeared

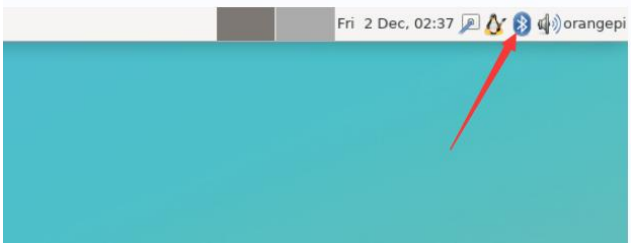
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
console=both
```

```
disp_mode=1080p60
fb0_width=1920
fb0_height=1080
extraargs=vt.global_cursor_default=0
```

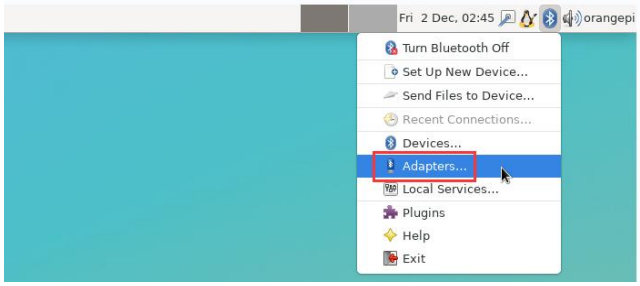
3. 10. How to use Bluetooth

3. 10. 1. Test method of desktop image

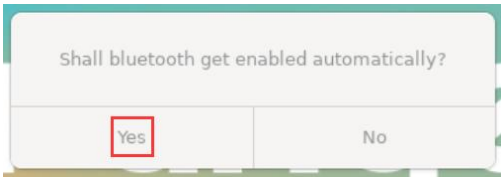
1) Click the Bluetooth icon in the upper right corner of the desktop



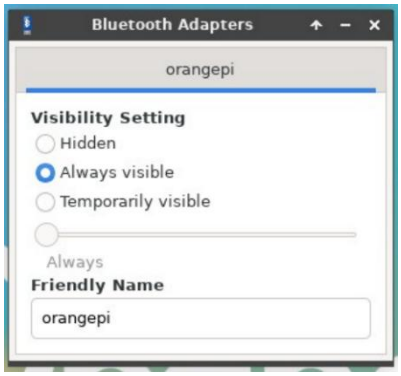
2) Then select the adapter



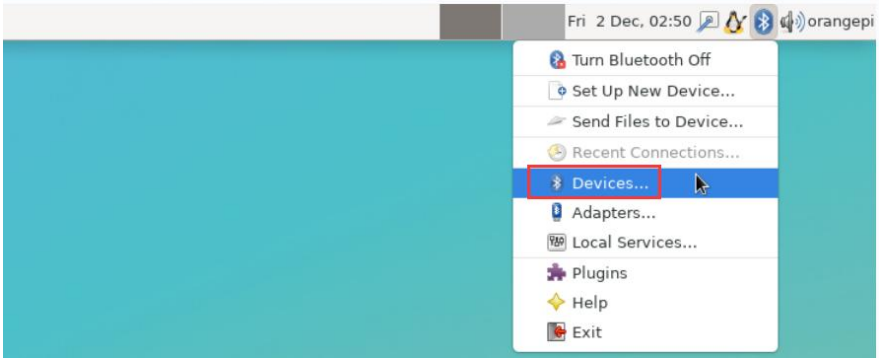
3) If there is a prompt on the following interface, please select **Yes**



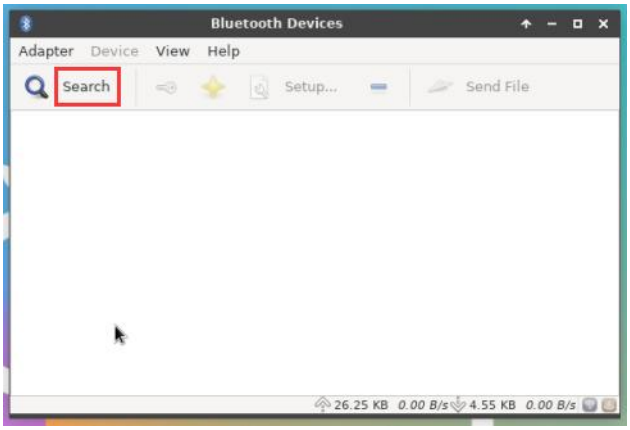
4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter setting interface, and then close it



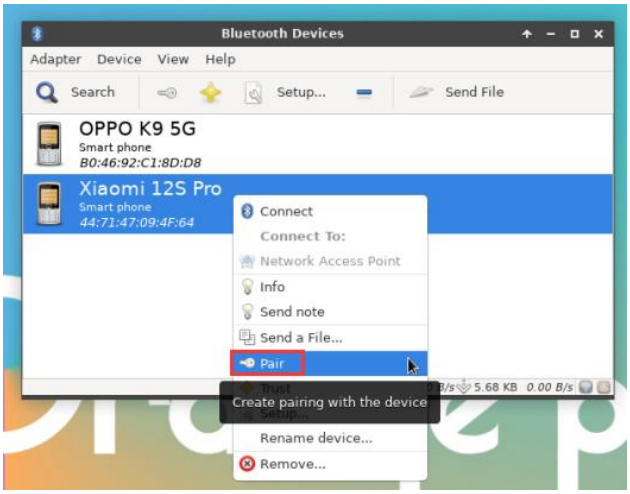
5) Then open the configuration interface of the Bluetooth device



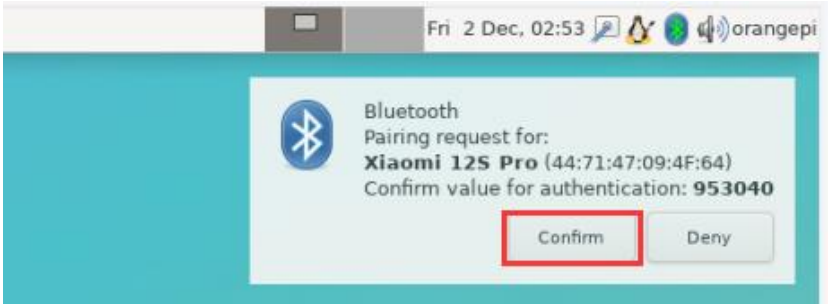
6) Click **Search** to start scanning the surrounding Bluetooth devices



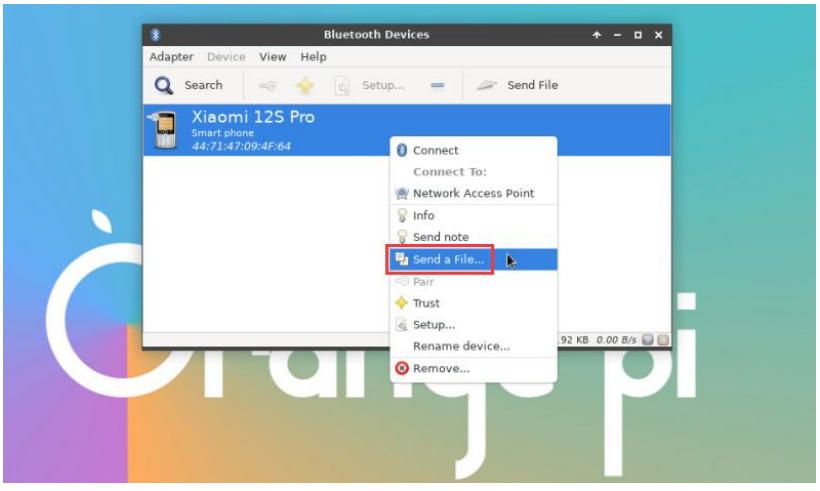
7) Then select the Bluetooth device you want to connect to, and then click the right button of the mouse to pop up the operation interface for this Bluetooth device, select **Pair** to start pairing, and the demonstration here is to pair with an Android phone



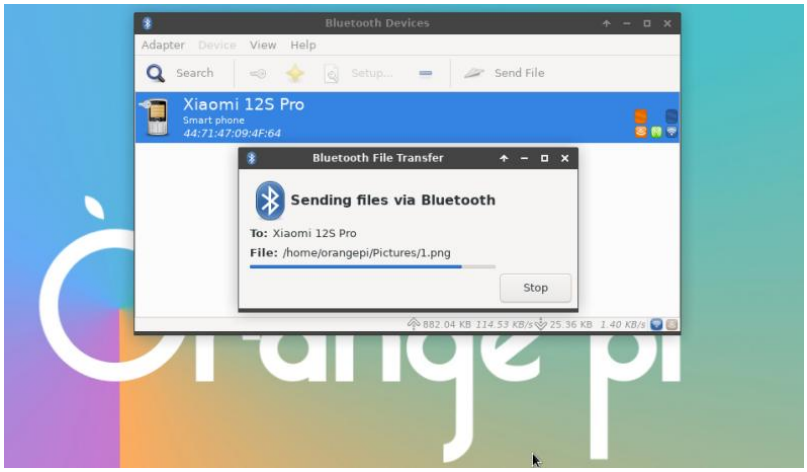
8) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop, just select **Confirm** to confirm, and the phone also needs to confirm at this time



9) After pairing with the mobile phone, you can select the paired Bluetooth device, then right click and select **Send a File** to start sending a picture to the mobile phone



10) The interface for sending pictures is as follows



3. 10. 2. How to use the server image

1) After entering the system, you can first check whether there is a Bluetooth device node through the **hciconfig** command. If it exists, it means that the Bluetooth initialization is normal.

```
orangepi@orangepi:~$ sudo apt update && sudo apt install -y bluez
orangepi@orangepi:~$ hciconfig -a
hci0:   Type: Primary  Bus: UART
        BD Address: 3E:61:3D:19:0E:52  ACL MTU: 1021:8  SCO MTU: 240:3
        UP RUNNING
        RX bytes:925 acl:0 sco:0 events:72 errors:0
        TX bytes:5498 acl:0 sco:0 commands:72 errors:0
        Features: 0xbf 0xff 0x8d 0xfe 0xdb 0x3d 0x7b 0xc7
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH SNIFF
        Link mode: SLAVE ACCEPT
        Name: 'orangepi'
        Class: 0x3c0000
        Service Classes: Rendering, Capturing, Object Transfer, Audio
        Device Class: Miscellaneous,
        HCI Version: 5.0 (0x9)  Revision: 0x400
        LMP Version: 5.0 (0x9)  Subversion: 0x400
        Manufacturer: Spreadtrum Communications Shanghai Ltd (492)
```

2) Use **bluetoothctl** to scan bluetooth devices

```
orangepi@orangepi:~$ sudo bluetoothctl
```

```
[NEW] Controller 10:11:12:13:14:15 orangepizero3 [default]
Agent registered
[bluetooth]# power on          #enable controller
Changing power on succeeded
[bluetooth]# discoverable on      #Make the controller discoverable
Changing discoverable on succeeded
[CHG] Controller 10:11:12:13:14:15 Discoverable: yes
[bluetooth]# pairable on     #Set the controller as pairable
Changing pairable on succeeded
[bluetooth]# scan on          #Start scanning for nearby Bluetooth devices
Discovery started
[CHG] Controller 10:11:12:13:14:15 Discovering: yes
[NEW] Device 76:60:79:29:B9:31 76-60-79-29-B9-31
[NEW] Device 9C:2E:A1:42:71:11 MiPhone
[NEW] Device DC:72:9B:4C:F4:CF orangepi
[bluetooth]# scan off          #After scanning to the Bluetooth device you want to connect, you can close the scan, and then write down the MAC address of the Bluetooth device. The Bluetooth device tested here is an Android phone, the name of the Bluetooth is orangepi, and the corresponding MAC address is DC:72:9B:4C :F4:CF
Discovery stopped
[CHG] Controller 10:11:12:13:14:15 Discovering: no
[CHG] Device DC:72:9B:4C:F4:CF RSSI is nil
```

3) After scanning to the device you want to pair, you can pair it. You need to use the MAC address of the device for pairing

```
[bluetooth]# pair DC:72:9B:4C:F4:CF      #Use the MAC address of the scanned Bluetooth device for pairing
Attempting to pair with DC:72:9B:4C:F4:CF
[CHG] Device DC:72:9B:4C:F4:CF Connected: yes
Request confirmation
[leeb1m[agent] Confirm passkey 764475 (yes/no): yes   #Enter yes here, and you need to confirm on the mobile phone
[CHG] Device DC:72:9B:4C:F4:CF Modalias: bluetooth:v010Fp107Ed1436
[CHG] Device DC:72:9B:4C:F4:CF UUIDs: 0000046a-0000-1000-8000-00805f9b34fb
[CHG] Device DC:72:9B:4C:F4:CF ServicesResolved: yes
```

```
[CHG] Device DC:72:9B:4C:F4:CF Paired: yes
Pairing successful      #Prompt that the pairing is successful
[CHG] Device DC:72:9B:4C:F4:CF ServicesResolved: no
[CHG] Device DC:72:9B:4C:F4:CF Connected: no
```

4) After the pairing is successful, the Bluetooth interface of the mobile phone will be displayed as follows

5) To connect a Bluetooth device, you need to install the **pulseaudio-module-bluetooth** package, and then start the **pulseaudio** service

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt -y install pulseaudio-module-bluetooth
orangepi@orangepi:~$ pulseaudio --start
```

6) How to connect to a Bluetooth device

```
orangepi@orangepi:~$ sudo bluetoothctl
Agent registered
[bluetooth]# paired-devices    #View the MAC address of the paired Bluetooth device
Device DC:72:9B:4C:F4:CF orangepi
[bluetooth]# connect DC:72:9B:4C:F4:CF     #Connect to bluetooth device using MAC address
Attempting to connect to DC:72:9B:4C:F4:CF
[CHG] Device DC:72:9B:4C:F4:CF Connected: yes
Connection successful
[CHG] Device DC:72:9B:4C:F4:CF ServicesResolved: yes
[CHG] Controller 10:11:12:13:14:15 Discoverable: no
[orangepi]#        #This prompt appears to indicate that the connection is successful
```

7) After connecting the bluetooth device, the bluetooth configuration interface of the

Android mobile phone can see **the prompt that the audio has been connected for calls and media**

3. 11. USB interface test

The USB interface can be connected to a USB hub to expand the number of USB interfaces.

3. 11. 1. Connect USB mouse or keyboard to test

1) Insert the USB interface keyboard into the USB interface of the Orange Pi development board

2) Connect the Orange Pi development board to the HDMI display

3) If the mouse or keyboard can operate normally, it means that the USB interface is working normally (the mouse can only be used in the desktop version of the system)

3. 11. 2. Connect USB storage device test

1) First insert the U disk or USB mobile hard disk into the USB interface of the Orange Pi development board

2) Execute the following command, if you can see the output of sdX, it means that the U disk is recognized successfully

```
orangepi@orangepi:~$ cat /proc/partitions | grep "sd*"
major minor  #blocks  name
   8        0   30044160 sda
   8        1   30043119 sda1
```

3) Use the mount command to mount the U disk to **/mnt**, and then you can view the files in the U disk

```
orangepi@orangepi:~$ sudo mount /dev/sda1 /mnt/
orangepi@orangepi:~$ ls /mnt/
test.txt
```

4) After mounting, you can view the capacity usage and mount point of the U disk through the **df -h** command

```
orangepi@orangepi:~$ df -h | grep "sd"
/dev/sda1               29G  208K   29G   1% /mnt
```

3.11.3. USB Ethernet card test

1) The usable USB Ethernet cards that **have been tested** so far are as follows. Among them, the RTL8153 USB Gigabit network card can be used normally when it is inserted into the USB 2.0 Host interface of the development board, but the speed cannot reach Gigabit. Please note that

serial number	model
1	RTL8152B USB 100M network card
2	RTL8153 USB Gigabit LAN

2) First insert the USB network card into the USB interface of the development board, and then insert the network cable into the USB network card to ensure that the network cable can access the Internet normally. If you can see the following log information through the **dmesg** command, it means that the USB network card is recognized normally.

```
orangepi@orangepi:~$ dmesg | tail
[  121.985016] usb 3-1: USB disconnect, device number 2
[  126.873772] sunxi-ehci 5311000.ehci3-controller: ehci_irq: highspeed device connect
[  127.094054] usb 3-1: new high-speed USB device number 3 using sunxi-ehci
[  127.357472] usb 3-1: reset high-speed USB device number 3 using sunxi-ehci
[  127.557960] r8152 3-1:1.0 eth1: v1.08.9
[  127.602642] r8152 3-1:1.0 enx00e04c362017: renamed from eth1
[  127.731874] IPv6: ADDRCONF(NETDEV_UP): enx00e04c362017: link is not ready
[  127.763031] IPv6: ADDRCONF(NETDEV_UP): enx00e04c362017: link is not ready
```

```
[    129.892465] r8152 3-1:1.0 enx00e04c362017: carrier on
[     129.892583] IPv6: ADDRCONF(NETDEV_CHANGE): enx00e04c362017: link becomes ready
```

3) Then you can see the device node of the USB network card and the automatically assigned IP address through the ifconfig command

```
orangepi@orangepi:~$ sudo ifconfig
enx00e04c362017: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.177  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::681f:d293:4bc5:e9fd  prefixlen 64  scopeid 0x20<link>
        ether 00:e0:4c:36:20:17  txqueuelen 1000  (Ethernet)
        RX packets 1849  bytes 134590 (134.5 KB)
        RX errors 0  dropped 125  overruns 0  frame 0
        TX packets 33  bytes 2834 (2.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

4) The command to test network connectivity is as follows

```
orangepi@orangepi:~$ ping www.baidu.com -I enx00e04c362017
PING www.a.shifen.com (14.215.177.38) from 192.168.1.12 eth0: 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=56 time=6.74 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=56 time=6.80 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=56 time=6.26 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=56 time=7.27 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 6.260/6.770/7.275/0.373 ms
```

3.11.4. USB camera test

1) First insert the USB camera into the USB port of the Orange Pi development board

2) Then through the lsmod command, you can see that the kernel has automatically loaded the following modules

```
orangepi@orangepi:~$ lsmod
```

```
Module                  Size  Used by
uvcvideo              106496  0
```

3) 3) Through the v4l2-ctl command, you can see that the device node information of the USB camera is /dev/video0

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y v4l-utils
orangepi@orangepi:~$ v4l2-ctl --list-devices
USB 2.0 Camera (usb-sunxi-ehci-1):
    /dev/video0
```

Note that the l in v4l2 is a lowercase letter l, not the number 1.

In addition, the serial number of the video is not necessarily video0, please refer to what you actually see.

4) Use fswebcam to test the USB camera

a. Install fswebcam

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get install -y fswebcam
```

b. After installing fswebcam, you can use the following command to take pictures

a) -d option is used to specify the device node of the USB camera

b) --no-banner is used to remove the watermark of the photo

c) -r option is used to specify the resolution of the photo

d) -S option is used to set the number of previous frames to skip

e) ./image.jpg is used to set the name and path of the generated photo

```
orangepi@orangepi:~$ sudo fswebcam -d /dev/video0   \
--no-banner -r 1280x720 -S 5 ./image.jpg
```

c. In the server version of the linux system, you can use the scp command to transfer the taken pictures to the Ubuntu PC for mirror viewing after taking pictures

```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test (Modify the IP address and path according to the actual situation)
```

d. In the desktop version of the linux system, you can directly view the captured pictures through the HDMI display

5) Use mjpg-streamer to test USB camera

a. Download mjpg-streamer

a) Github download address:

```
orangepi@orangepi:~$ git clone https://github.com/jacksonliam/mjpg-streamer
```

b) The image download address of Gitee is:

```
orangepi@orangepi:~$ git clone https://gitee.com/leeboby/mjpg-streamer
```

b. Install dependent packages

a) Ubuntu system

```
orangepi@orangepi:~$ sudo apt-get install -y cmake libjpeg8-dev
```

b) Debian system

```
orangepi@orangepi:~$ sudo apt-get install -y cmake libjpeg62-turbo-dev
```

c. Compile and install mjpg-streamer

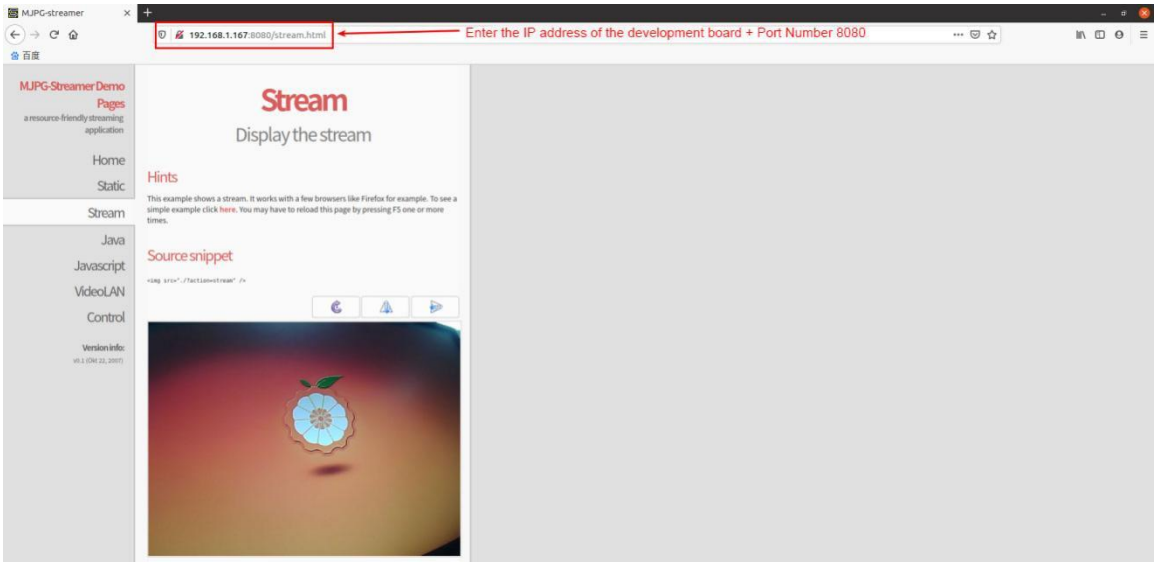
```
orangepi@orangepi:~$ cd mjpg-streamer/mjpg-streamer-experimental
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ make -j4
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ sudo make install
```

d. Then enter the following command to start mjpg_streamer

Note that the serial number of the video is not necessarily video0, please refer to what you actually see.

```
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ export LD_LIBRARY_PATH=.
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ sudo ./mjpg_streamer -i "./input_uvc.so -d \
/dev/video0 -u -f 30" -o "./output_http.so -w ./www"
```

e. Then enter **[the IP address of the development board: 8080]** in the browser of the Ubuntu PC or Windows PC or mobile phone on the same LAN as the development board to see the video output by the camera

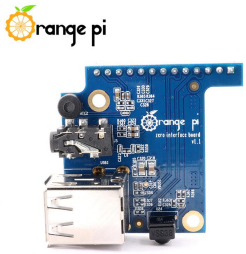


3. 12. Audio Test

3. 12. 1. How to play audio using the command line

3. 12. 1. 1. Headphone jack playback audio test

1) First, you need to insert the 13pin expansion board into the 13pin interface of the Orange Pi development board, and then insert the earphone into the audio interface



2) Through the **aplay -l** command, you can view the sound card devices supported by the Linux system

   a. The output of the linux5.4 system is as follows, where **card 0: audiocodec** is the sound card device required for earphone playback

```
root@orangepi:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: audiocodec [audiocodec], device 0: SUNXI-CODEC sun50iw9-codec-0 []
```

```
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

b. The output of the linux6.1 system is as follows, where **audiocodec** is the sound card device required for earphone playback

```
root@orangepi:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: audiocodec [audiocodec], device 0: CDC PCM Codec-0 [CDC PCM Codec-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio, and the headset can hear the sound

```
root@orangepi:~# aplay -D hw:0,0 /usr/share/sounds/alsa/audio.wav
Playing WAVE 'audio.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

If there is noise in the earphone test, please pull out some earphones, do not insert all the earphones.

3.12.1.2. HDMI audio playback test

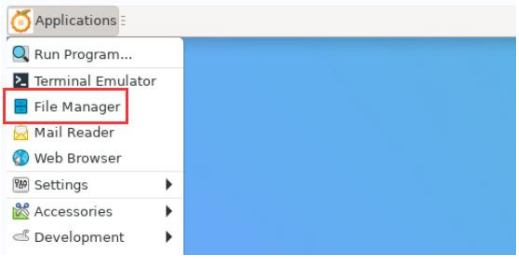
1) First use the Micro HDMI to HDMI cable to connect the Orange Pi development board to the TV (other HDMI monitors need to ensure that they can play audio)

2) HDMI audio playback does not require other settings, just use the **aplay** command to play directly

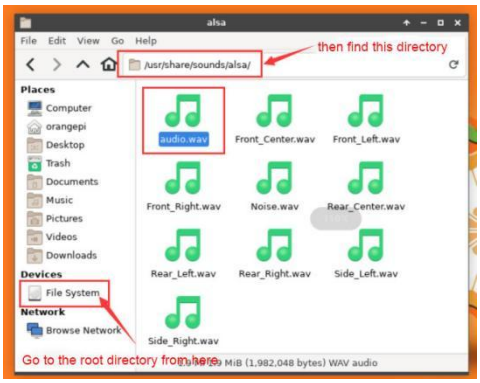
```
root@orangepi:~# aplay -D hw:2,0 /usr/share/sounds/alsa/audio.wav
```

3.12.2. Testing audio methods on desktop systems

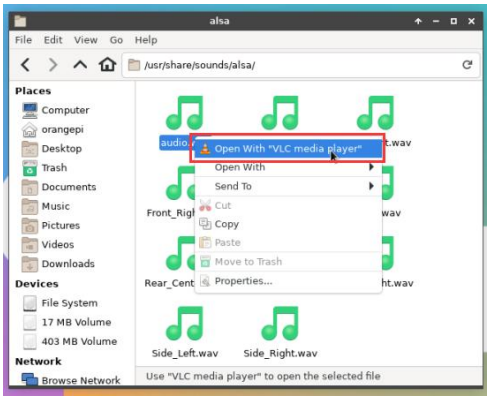
1) First open the file manager



2) Then find the following file (if there is no audio file in the system, you can upload an audio file to the system yourself)

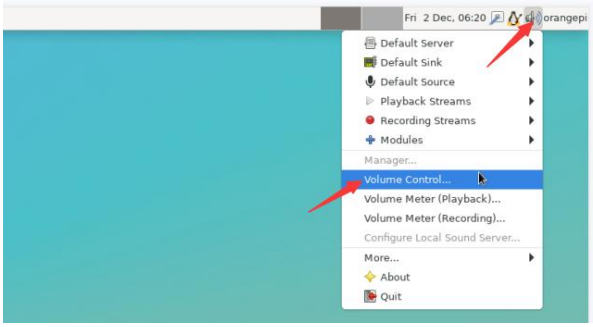


3) Then select the audio.wav file, right click and select open with vlc to start playing

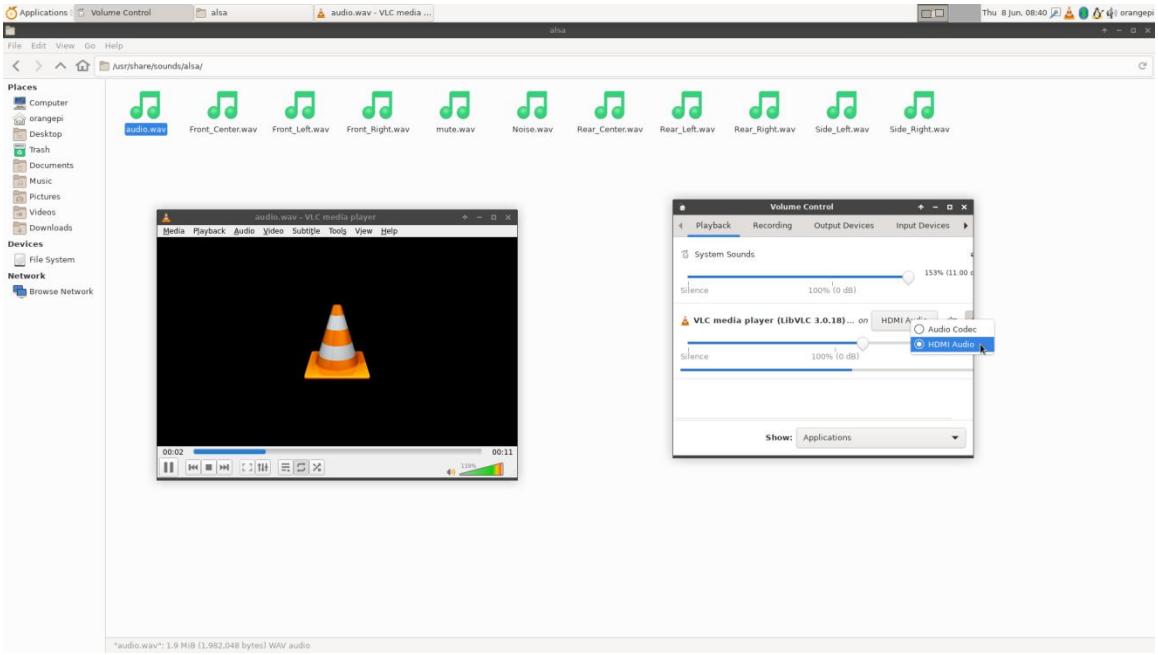


4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First open the volume control interface

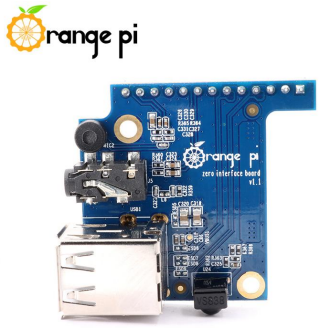


b. When playing audio, the audio device options that the playback software can use will be displayed in **Playback**, as shown in the figure below, where you can set which audio device to play to



3. 13. Infrared receiving test

1) First, you need to insert the 13pin expansion board into the 13pin interface of the Orange Pi development board. After inserting the expansion board, the development board can use the infrared receiving function



2) Install ir-keytable infrared testing software

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get install -y ir-keytable
```

3) Then execute ir-keytable to view the information of the infrared device

a. linux5.4 system output is as follows

```
orangepi@orangepi:~$ ir-keytable
Found /sys/class/rc/rc0/ (/dev/input/event1) with:
```

```
        Driver: sunxi-rc-recv, table: rc_map_sunxi
        lirc device: /dev/lirc0
        Supported protocols: lirc nec
        Enabled protocols: lirc nec
        Name: sunxi_ir_recv
        bus: 25, vendor/product: 0001:0001, version: 0x0100
        Repeat delay = 500 ms, repeat period = 125 ms
```

b. The output of the linux6.1 system is as follows

```
orangepi@orangepi:~$ ir-keytable
Found /sys/class/rc/rc0/ with:
    Name: sunxi-ir
    Driver: sunxi-ir
    Default keymap: rc-empty
    Input device: /dev/input/event5
    LIRC device: /dev/lirc0
    Attached BPF protocols: Operation not permitted
    Supported kernel protocols: lirc rc-5 rc-5-sz jvc sony nec sanyo mce_kbd rc-6 sharp xmp imon rc-mm
    Enabled kernel protocols: lirc
    bus: 25, vendor/product: 0001:0001, version: 0x0100
    Repeat delay = 500 ms, repeat period = 125 ms
```

4) Before testing the infrared receiving function, you need to prepare an infrared remote controller dedicated to Orange Pi, **other remote controllers do not support it**

5) Then enter the **ir-keytable -t** command in the terminal, and then use the infrared remote control to press the button against the infrared receiver of the Orange Pi development board to see the received key code in the terminal

a. linux5.4 system output is as follows

```
orangepi@orangepi:~$ sudo ir-keytable -t
Testing events. Please, press CTRL-C to abort.
1598339152.260376: event type EV_MSC(0x04): scancode = 0xfb0413
1598339152.260376: event type EV_SYN(0x00).
1598339152.914715: event type EV_MSC(0x04): scancode = 0xfb0410
```

b. linux6.1 system output is as follows

```
orangepi@orangepi:~$ sudo ir-keytable -c -p NEC -t
Old keytable cleared
Protocols changed to nec
Testing events. Please, press CTRL-C to abort.
202.063219: lirc protocol(nec): scancode = 0x45c
202.063249: event type EV_MSC(0x04): scancode = 0x45c
202.063249: event type EV_SYN(0x00).
```

3. 14. Temperature sensor

1) H618 has a total of 4 temperature sensors, the command to view the temperature is as follows:

The displayed temperature value needs to be divided by 1000, the unit is Celsius.

a. sensor0: CPU temperature sensor, the first command is used to view the type of temperature sensor, and the second command is used to view the value of the temperature sensor

```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone0/type
cpu_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone0/temp
57734
```

b. sensor1: DDR temperature sensor, the first command is used to view the type of temperature sensor, and the second command is used to view the value of the temperature sensor

```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone1/type
ddr_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone1/temp
57410
```

c. sensor2: GPU temperature sensor, the first command is used to view the type of

temperature sensor, the second command is used to view the value of the temperature sensor

```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone2/type
gpu_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone2/temp
59273
```

d. sensor3: VE temperature sensor, the first command is used to view the type of temperature sensor, and the second command is used to view the value of the temperature sensor

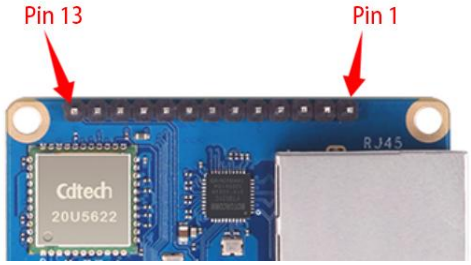
```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone3/type
ve_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone3/temp
58949
```

2) The system with Linux6.1 kernel can also use the **sensors** command to view the current temperature of the system

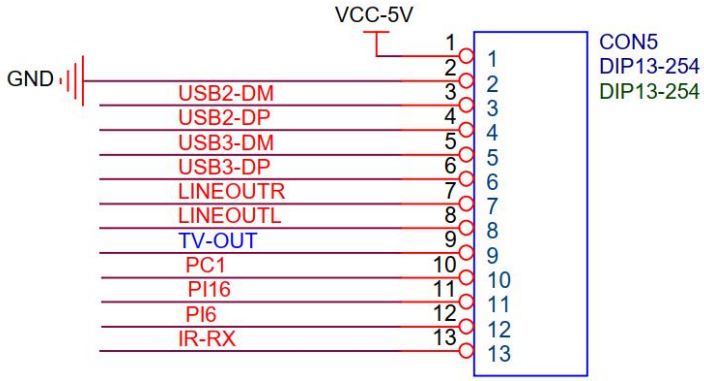
```
orangepi@orangepi:~$ sensors
cpu_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.4°C  (crit = +110.0°C)

gpu_thermal-virtual-0
Adapter: Virtual device
temp1:          +48.7°C  (crit = +110.0°C)

ddr_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.8°C  (crit = +110.0°C)

ve_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.2°C  (crit = +110.0°C)
```

3. 15. 13 Pin Description of the Pin Expansion Board Interface

1) Please refer to the figure below for the order of the interface pins of the 13 pin expansion board on the development board



2) The schematic diagram of the 13pin interface of the development board is as follows



3) The function description of the 13 pin expansion board interface pins of the development board is as follows

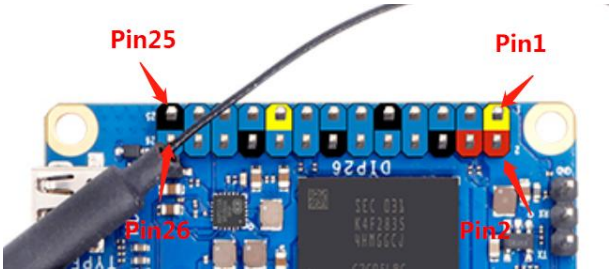
a. When the 13pin pin is connected to the expansion board, it can be provided additionally
   a) 2 个 USB 2.0 Host
   b) Audio output of the left and right audio channels of the earphones
   c) TV-OUT video output
   d) Infrared receiving function
   e) After the expansion board is connected, pins 10, 11 and 12 of the 13pin interface cannot be used
   f) **In addition, it should be noted that the MIC on the 13pin expansion board cannot be used on the Orange Pi Zero 3**

b. When pin 13 is not connected to the expansion board, pins 10, 11, 12 and 13 can be used as ordinary GPIO ports

GPIO serial number	Function	Pin
	5V	1
	GND	2
	USB2-DM	3
	USB2-DP	4
	USB3-DM	5
	USB3-DP	6
	LINEOUTR	7
	LINEOUTL	8
	TV-OUT	9
65	PC1	10
272	PI16	11
262	PI6	12
234	IR-RX/PH10	13

3.16. 26 Pin Interface Pin Description

1) For the order of the 26-pin interface pins on the Orange Pi Zero 3 development board, please refer to the silkscreen diagram on the development board



2) The functions of the 26 pin interface pins on the development board are shown in the table below

GPIO S/N	GPIO	Function	Pin
		3.3V	1
229	PH5	TWI3-SDA	3
228	PH4	TWI3-SCK	5

Pin	Function	GPIO	GPIO S/N
2	5V		
4	5V		
6	GND		

73	PC9	PC9	7	8	UART5_TX	PH2	226
		GND	9	10	UART5_RX	PH3	227
70	PC6	PC6	11	12	PC11	PC11	75
69	PC5	PC5	13	14	GND		
72	PC8	PC8	15	16	PC15	PC15	79
		3.3V	17	18	PC14	PC14	78
231	PH7	SPI1_MOSI	19	20	GND		
232	PH8	SPI1_MISO	21	22	PC7	PC7	71
230	PH6	SPI1_CLK	23	24	SPI1_CS	PH9	233
		GND	25	26	PC10	PC10	74

3) 3) There are a total of 17 GPIO ports in the 26pin interface, and the voltage of all GPIO ports is **3.3v**

3.17. How to install wiringOP

Note that wiringOP has been pre-installed in the linux image released by Orange Pi. Unless the code of wiringOP is updated, there is no need to re-download, compile and install, just use it directly.

The storage path of the compiled wiringOP deb package in orangepi-build is: orangepi-build/external/cache/debs/arm64/wiringpi_x.xx.deb

After entering the system, you can run the gpio readall command. If you can see the following output, it means that wiringOP has been pre-installed and can be used normally.

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
 |   70 |   5 |      PC6 |   ALT5 | 0 | 11 || 12 | 0 | OFF    | PC11     | 6   | 75   |
 |   69 |   7 |      PC5 |   ALT5 | 0 | 13 || 14 |   |        | GND      |     |      |
 |   72 |   8 |      PC8 |    OFF | 0 | 15 || 16 | 0 | OFF    | PC15     | 9   | 79   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PC14     | 10  | 78   |
 |  231 |  11 |   MOSI.1 |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  232 |  12 |   MISO.1 |    OFF | 0 | 21 || 22 | 0 | OFF    | PC7      | 13  | 71   |
 |  230 |  14 |   SCLK.1 |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 233  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | PC10     | 16  | 74   |
 |   65 |  17 |      PC1 |    OFF | 0 | 27 || 28 |   |        |          |     |      |
 |  272 |  18 |     PI16 |   ALT2 | 0 | 29 || 30 |   |        |          |     |      |
 |  262 |  19 |      PI6 |    OFF | 0 | 31 || 32 |   |        |          |     |      |
 |  234 |  20 |     PH10 |   ALT3 | 0 | 33 || 34 |   |        |          |     |      |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
orangepi@orangepi:~$
```

wiringOP is currently mainly adapted to the functions of setting GPIO port input and output, setting GPIO port output high and low levels, and setting up and down pull-down resistors. Functions such as hardware PWM are not available.

1) Download the code of wiringOP

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y git
orangepi@orangepi:~$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

Note that the source code needs to download the code of the wiringOP next branch, please don't miss the -b next parameter.

If you have problems downloading the code from GitHub, you can directly use the wiringOP source code that comes with the Linux image, and the storage location is: /usr/src/wiringOP.

2) Compile and install wiringOP

```
orangepi@orangepi:~$ cd wiringOP
orangepi@orangepi:~/wiringOP$ sudo ./build clean
```

```
orangepi@orangepi:~/wiringOP$ sudo ./build
```

3) Test the output of the gpio readall command as follows

a. There is a one-to-one correspondence between pins 1 to 26 and 26 Pins on the development board

b. Pin 27 corresponds to pin 10 of 13pin on the development board

c. Pin 29 corresponds to pin 11 of 13pin on the development board

d. Pin 31 corresponds to pin 12 of 13pin on the development board

e. Pin 33 corresponds to pin 13 of 13pin on the development board

f. Pins 28, 30, 32, and 34 are empty, please ignore them

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
 |   70 |   5 |      PC6 |   ALT5 | 0 | 11 || 12 | 0 | OFF    | PC11     | 6   | 75   |
 |   69 |   7 |      PC5 |   ALT5 | 0 | 13 || 14 |   |        | GND      |     |      |
 |   72 |   8 |      PC8 |    OFF | 0 | 15 || 16 | 0 | OFF    | PC15     | 9   | 79   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PC14     | 10  | 78   |
 |  231 |  11 |   MOSI.1 |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  232 |  12 |   MISO.1 |    OFF | 0 | 21 || 22 | 0 | OFF    | PC7      | 13  | 71   |
 |  230 |  14 |   SCLK.1 |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 233  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | PC10     | 16  | 74   |
 |   65 |  17 |      PC1 |    OFF | 0 | 27 || 28 |   |        |          |     |      |
 |  272 |  18 |     PI16 |   ALT2 | 0 | 29 || 30 |   |        |          |     |      |
 |  262 |  19 |      PI6 |    OFF | 0 | 31 || 32 |   |        |          |     |      |
 |  234 |  20 |     PH10 |   ALT3 | 0 | 33 || 34 |   |        |          |     |      |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
orangepi@orangepi:~$
```

3. 18. 26pin interface GPIO, I2C, UART, SPI and PWM test

3. 18. 1. 26pin GPIO port test

1) The following takes pin 7—corresponding to GPIO as PC9—corresponding to wPi number 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

2) First set the GPIO port to output mode, where the third parameter requires the serial number of the wPi corresponding to the input pin

```
root@orangepi:~/wiringOP# gpio mode 2 out
```

3) Then set the GPIO port to output a low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 0v, it means that the low level is set successfully

```
root@orangepi:~/wiringOP# gpio write 2 0
```

Use gpio readall to see that the value (V) of pin 7 has changed to 0

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OUT | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

4) Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

```
root@orangepi:~/wiringOP# gpio write 2 1
```

Use gpio readall to see that the value (V) of pin 7 has changed to 1

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OUT | 1 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

5) The setting method of other pins is similar, just modify the serial number of wPi to the corresponding serial number of the pin

3.18.2. How to set pull-up and pull-down resistors on 26 pin GPIO ports

1) The following takes pin 7—the corresponding GPIO is PC9—the corresponding wPi number is 2 — as an example to demonstrate how to set the pull-up and pull-down resistors of the GPIO port

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 | 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

2) First, you need to set the GPIO port to the input mode, and the third parameter needs to be the serial number of the wPi corresponding to the input pin

```
root@orangepi:~/wiringOP# gpio mode 2 in
```

3) After setting to input mode, execute the following command to set the GPIO port to pull-up mode

```
root@orangepi:~/wiringOP# gpio mode 2 up
```

4) Then enter the following command to read the level of the GPIO port, if the level is 1, it means that the pull-up mode is set successfully

```
root@orangepi:~/wiringOP# gpio read 2
1
```

5) Then execute the following command to set the GPIO port to pull-down mode

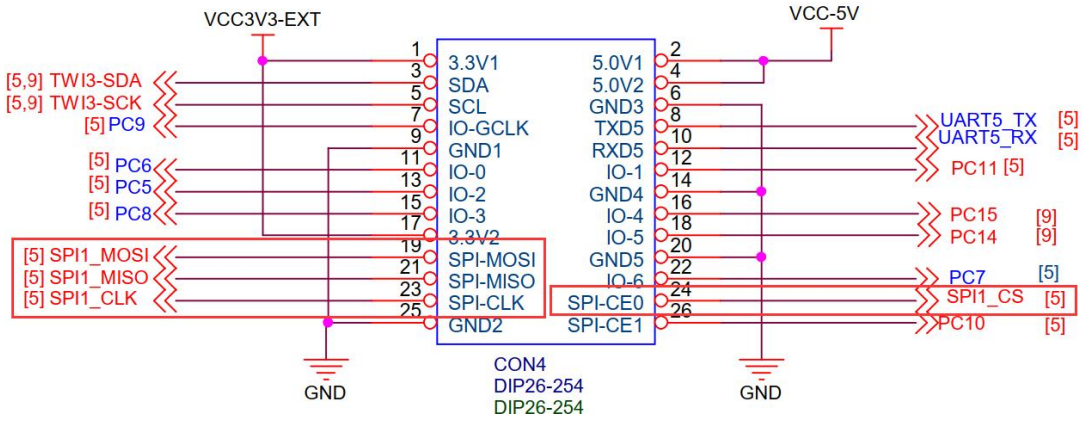
```
root@orangepi:~/wiringOP# gpio mode 2 down
```

6) Then enter the following command to read the level of the GPIO port, if the level is 0, the pull-down mode is set successfully

```
root@orangepi:~/wiringOP# gpio read 2
0
```

3. 18. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the available spi is spi1

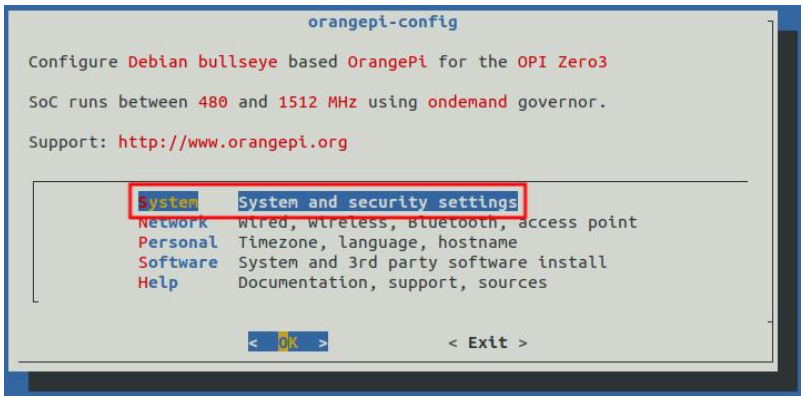


2) The spi1 is disabled by default in the Linux system and needs to be manually enabled before it can be used. The steps to open are as follows:

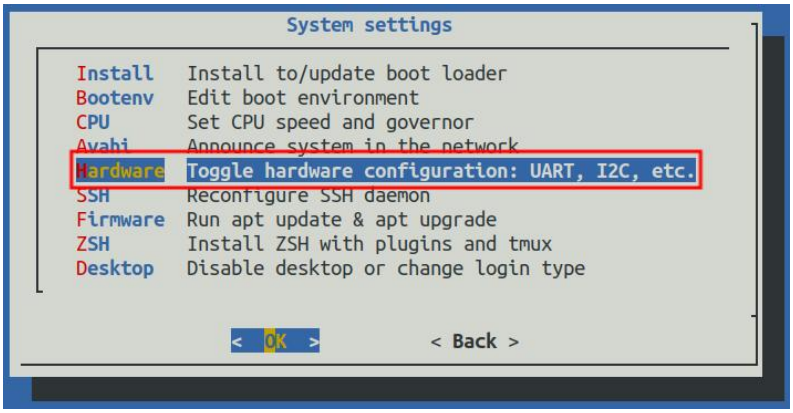
a. First run **orangepi-config**, normal users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

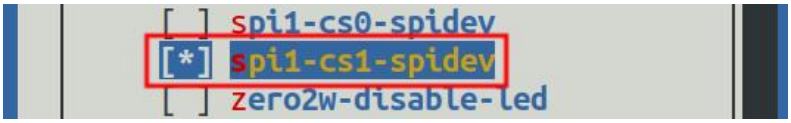
b. Then select **System**



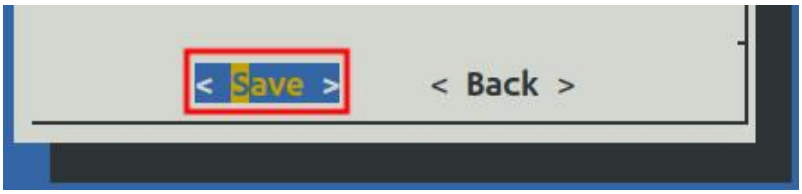
c. Then select **Hardware**



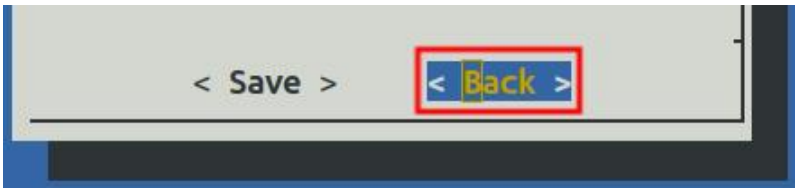
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **spi1-cs1-spidev**



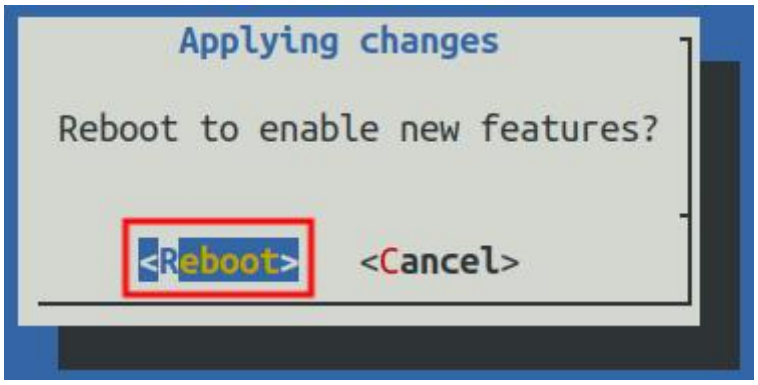
e. Then select **<Save>** to save



f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration take effect



3) Then check whether there is a **spidev1.1** device node in the Linux system. If it exists, it means that the configuration of SPI1 has taken effect

```
orangepi@orangepi:~$ ls /dev/spidev1*
/dev/spidev1.1
```

4) Do not short-circuit the mosi and miso pins of SPI1 first, the output result of running spidev_test is as follows, you can see that the data of TX and RX are inconsistent

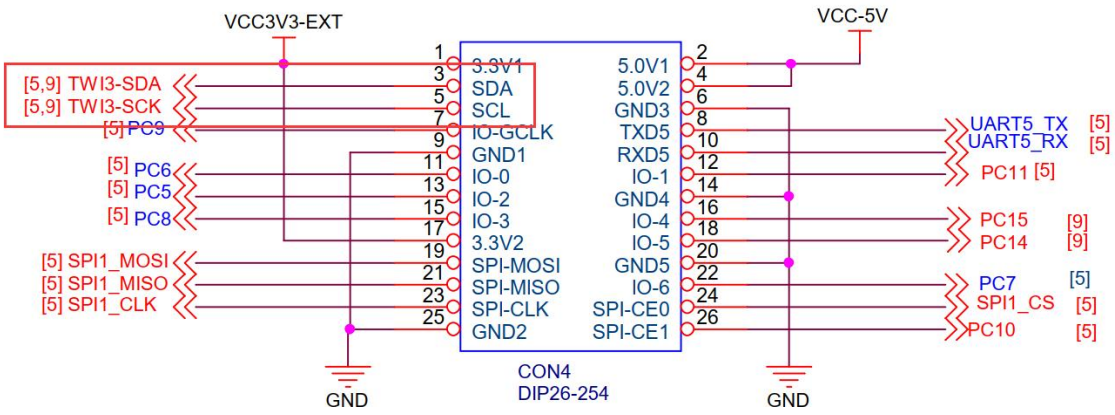
```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    | ......@....▒.....................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF    | ................................
```

5) Then short the two pins of mosi (the 19th pin in the 26pin interface) and miso (the 21st pin in the 26pin interface) of SPI1, and then run the output of spidev_test as follows, you can see the sending and receiving same data

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    | ......@....▒.....................▒.
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    | ......@....▒.....................▒.
```

3. 18. 4. 26pin I2C test

1) According to the schematic diagram of 26pin, the available i2c is i2c3

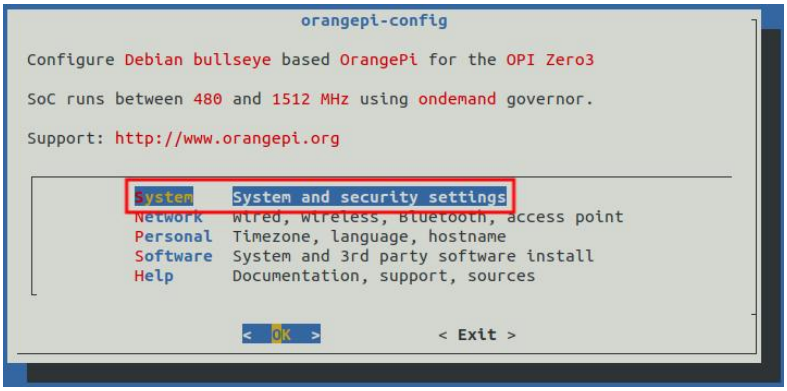


2) i2c3 is disabled by default in the Linux system, and it needs to be manually enabled before it can be used. The steps to open are as follows:

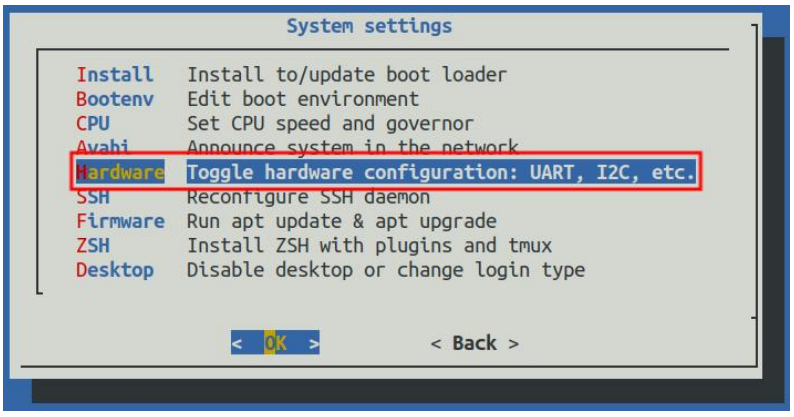
a. First run **orangepi-config**, normal users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**

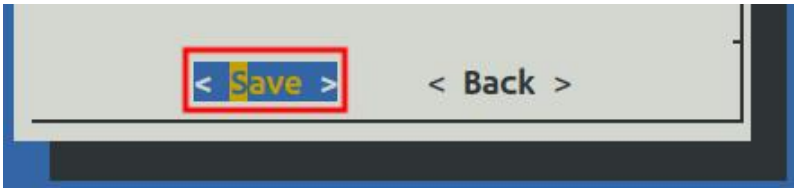


c. Then select **Hardware**

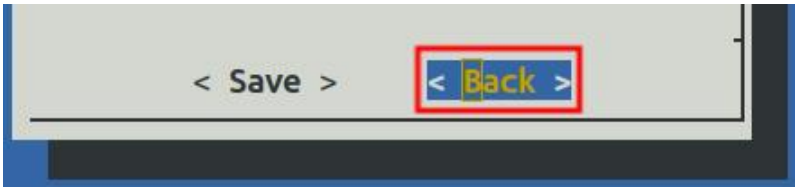


d. d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-i2c3**

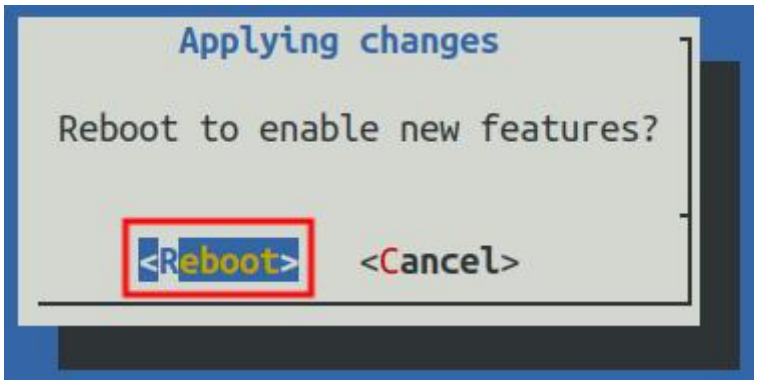
e. Then select **<Save>** to save



f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration take effect



3) After starting the linux system, first confirm that there is an i2c3 device node under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-3
/dev/i2c-3
```

4) Then start testing i2c, first install i2c-tools

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y i2c-tools
```

5) Then connect an i2c device to the i2c3 pin of the 26pin connector

Please select the 5V and 3.3V pins according to the specific i2c device, and the voltage value required by different i2c devices may be different.

sda pin	**Corresponding to pin 3 in 26pin**
sck pin	**Corresponding to pin 5 in 26pin**
5v pin	**Corresponding to pin 2 in 26pin**
3.3v pin	**Corresponding to pin 1 in 26pin**
gnd pin	**Corresponding to pin 6 in 26pin**

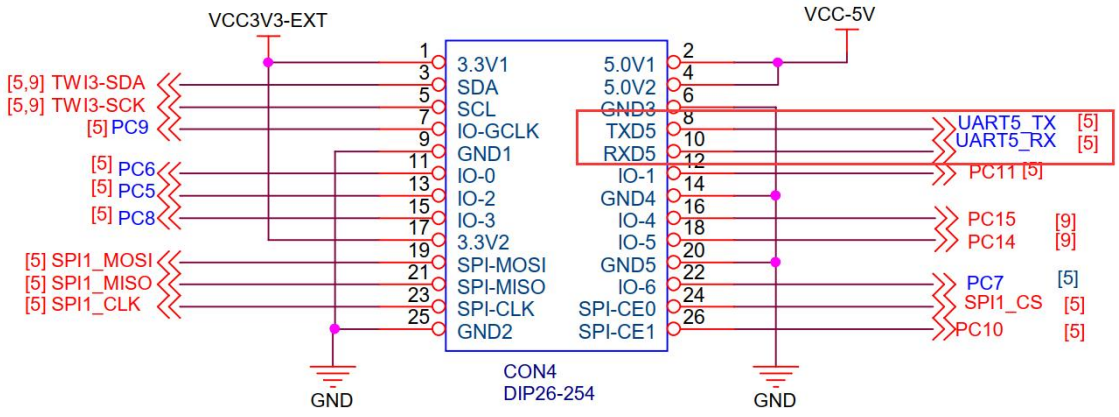
6) Then use the **i2cdetect -y 3** command, if the address of the connected i2c device can be detected, it means that i2c can be used normally

Different i2c device addresses are different, the address 0x50 in the figure below is just an example. Please refer to what you actually see.

```
root@orangepi:~# i2cdetect -y 3
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

3. 18. 5. 26pin UART test

1) According to the schematic diagram of the 26pin interface, the available uart is uart5

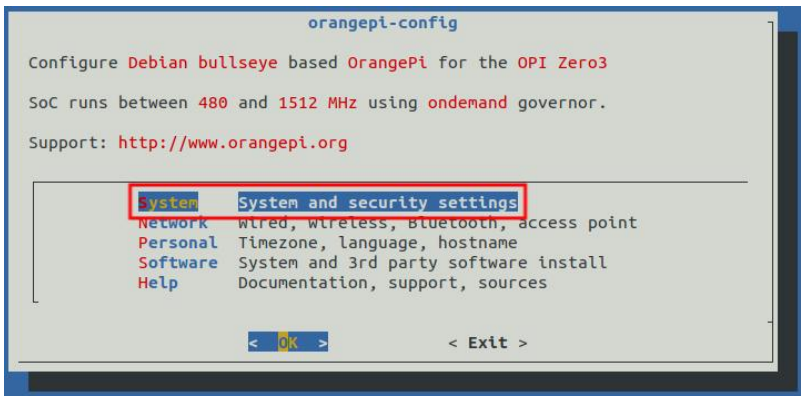


2) Uart5 is disabled by default in the Linux system, and it needs to be opened manually to use it. The steps to open are as follows:

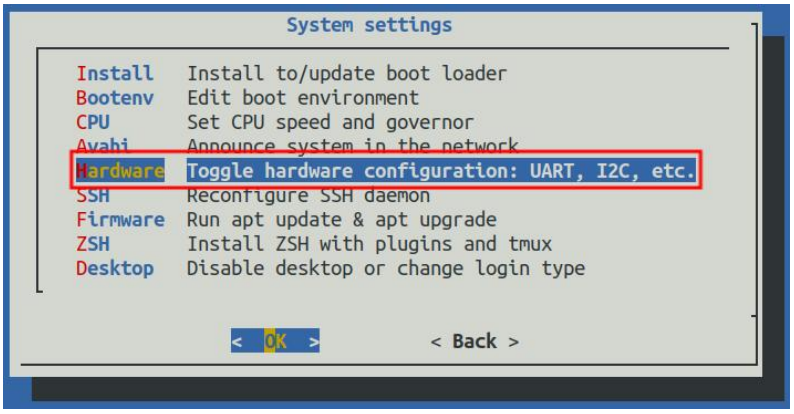
a. First run **orangepi-config**, normal users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



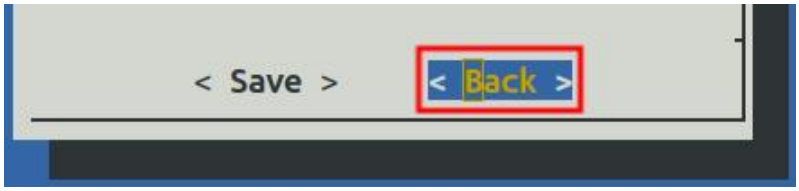
c. Then select **Hardware**



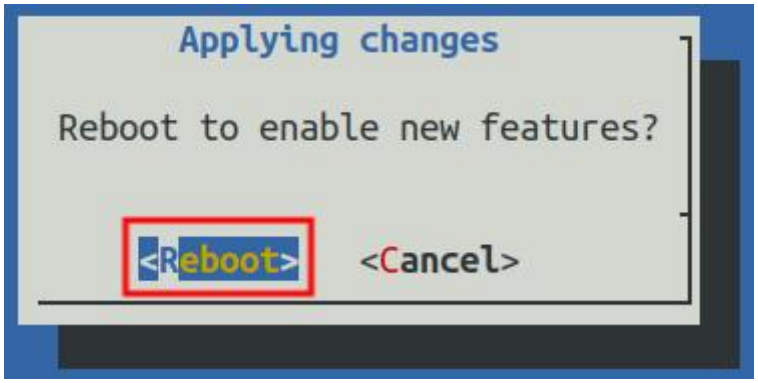
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-uart5**

e. Then select **<Save>** to save

f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration take effect



3) After entering the linux system, first confirm whether there is a uart5 device node under **/dev**

Note that the linux5.4 system is /dev/ttyAS5.

```
orangepi@orangepi:~$ ls /dev/ttyS5
/dev/ttyS5
```

4) Then start to test the uart5 interface, first use the DuPont line to short the rx and tx of the uart5 interface to be tested

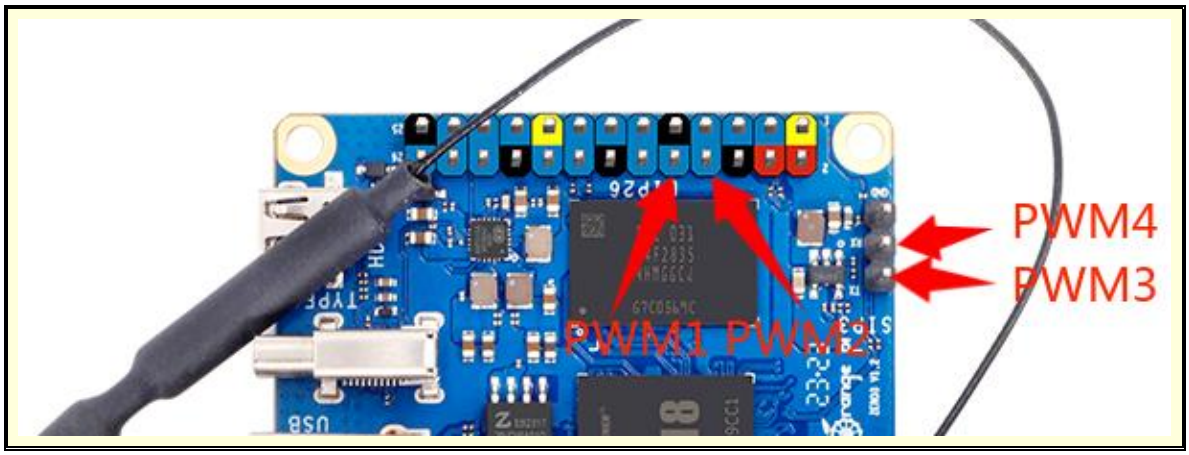
	uart5
tx pin	Corresponding to pin 8 of 26pin
rx pin	Corresponding to pin 10 of 26pin

5) Use the **gpio** command in wiringOP to test the loopback function of the serial port as shown below. If you can see the following print, it means that the serial port communication is normal

```
orangepi@orangepi:~$ gpio serial /dev/ttyS5          # linux-6.1 test command
orangepi@orangepi:~$ gpio serial /dev/ttyAS5         # linux-5.4 test command

Out:    0:  ->  0
Out:    1:  ->  1
Out:    2:  ->  2
Out:    3:  ->  3^C
```

3. 18. 6. PWM test method

The development board can use up to 4 channels of PWM, and the positions of their pins are shown in the figure below:

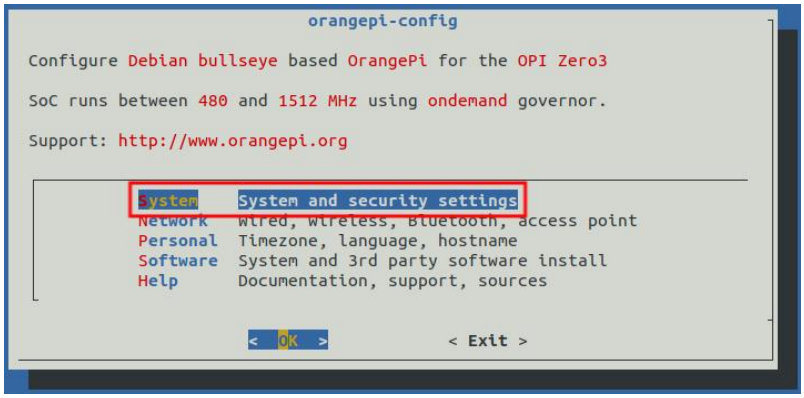


1) In the Linux system, the pwm is closed by default, and it needs to be opened manually to use it. The steps to open are as follows:

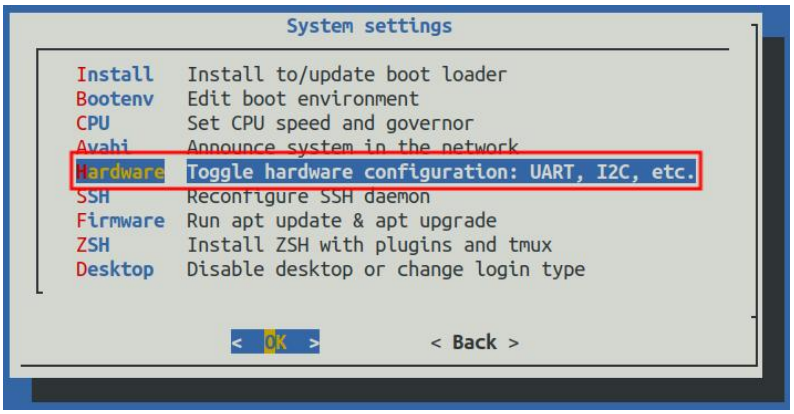
a. First run **orangepi-config**, normal users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



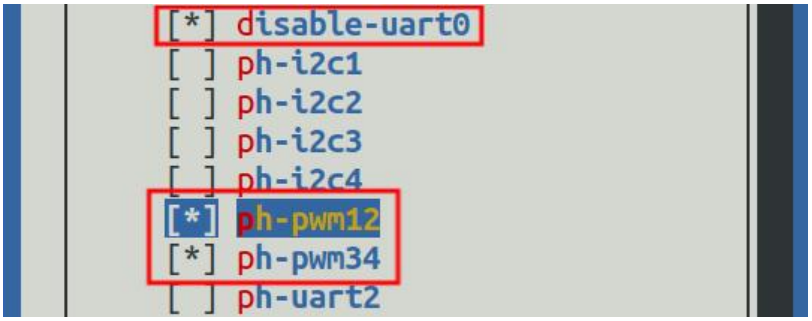
c. Then select **Hardware**



d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the configuration corresponding to pwm

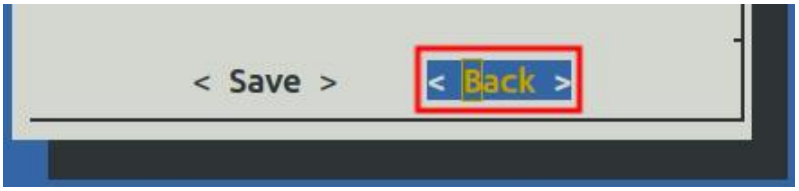
Since the RX and TX pins of UART5 in the PWM1, PWM2 and 26pin interfaces are multiplexed, please make sure that the UART5 configuration is not selected (do not check ph-uart5) when turning on PWM1 and PWM2 (you need to select ph-pwm12).

PWM3, PWM4 and the TX and RX pins in the debug serial port are multiplexed, so when using PWM3 and PWM4 (need to select ph-pwm34), please turn off the configuration of UART0 (need to select disable-uart0), after turning off UART0 The debug serial port cannot be used.

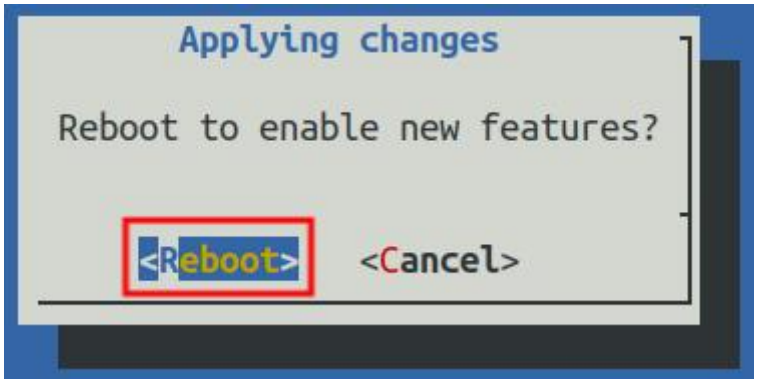


e. Then select **<Save>** to save

f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration take effect



2) After restarting, the PWM test can be started

Please execute the following commands under the root user.

a. Enter the following command on the command line to make pwm1 output a 50Hz square wave

```
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm1/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm1/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm1/enable
```

b. Enter the following command on the command line to make pwm2 output a 50Hz square wave

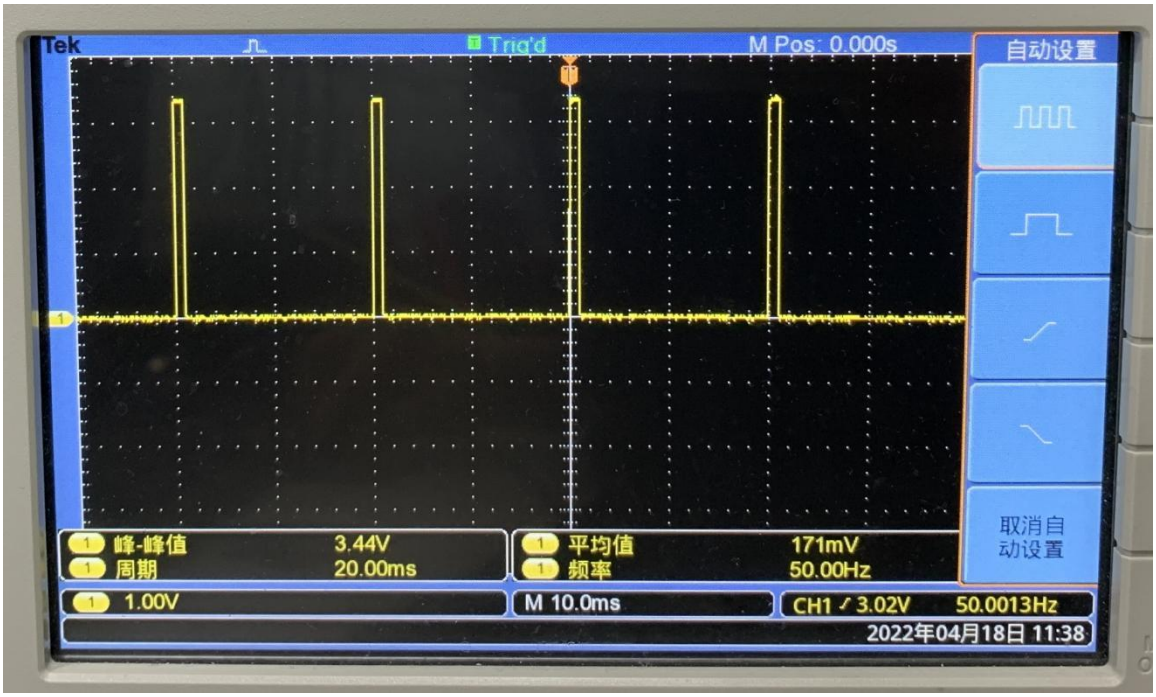
```
root@orangepi:~# echo 2 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm2/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm2/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm2/enable
```

c. Enter the following command on the command line to make pwm3 output a 50Hz square wave

```
root@orangepi:~# echo 3 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm3/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm3/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm3/enable
```

d. Enter the following command on the command line to make pwm4 output a 50Hz square wave

```
root@orangepi:~# echo 4 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm4/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm4/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm4/enable
```



3. 19. How to install and use wiringOP-Python

wiringOP-Python is the Python language version of wiringOP, which is used to operate the hardware resources of the development board, such as GPIO, I2C, SPI and UART, in the Python program.

In addition, please note that all the following commands are operated under the root user.

3. 19. 1. How to install wiringOP-Python

1) First install the dependency package

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python

Note that the following git clone --recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on wiringOP. Please make sure that the download process does not report errors due to network problems.

If you have problems downloading the code from GitHub, you can directly use the wiringOP-Python source code that comes with the Linux image, and the storage location is: /usr/src/wiringOP-Python

```
root@orangepi:~# git clone --recursive https://github.com/orangepi-xunlong/wiringOP-Python -b next
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# git submodule update --init --remote
```

3) Then use the following command to compile wiringOP-Python and install it into the Linux system of the development board

```
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# python3 generate-bindings.py > bindings.i
root@orangepi:~/wiringOP-Python# sudo python3 setup.py install
```

4) Then enter the following command, if there is help information output, it means that wiringOP-Python is installed successfully, press the **q** key to exit the help information interface

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi;  help(wiringpi)"
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.
```

5) The steps to test whether wiringOP-Python is successfully installed under the python command line are as follows:

a. First use the python3 command to enter the command line mode of python3

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi;
```

c. Finally, enter the following command to view the help information of wiringOP-Python, and press the q key to exit the help information interface

```
>>> help(wiringpi)
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

    class GPIO(builtins.object)
     |  GPIO(pinmode=0)
     |

>>>
```

3. 19. 2. 26pin GPIO port test

wiringOP-Python is the same as wiringOP, you can also determine which GPIO pin to operate by specifying the wPi number, because there is no command to check the wPi number in wiringOP-Python, so you can only check the board wPi number and physical Correspondence between pins.

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
 |   70 |   5 |      PC6 |   ALT5 | 0 | 11 || 12 | 0 | OFF    | PC11     | 6   | 75   |
 |   69 |   7 |      PC5 |   ALT5 | 0 | 13 || 14 |   |        | GND      |     |      |
 |   72 |   8 |      PC8 |    OFF | 0 | 15 || 16 | 0 | OFF    | PC15     | 9   | 79   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PC14     | 10  | 78   |
 |  231 |  11 |   MOSI.1 |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  232 |  12 |   MISO.1 |    OFF | 0 | 21 || 22 | 0 | OFF    | PC7      | 13  | 71   |
 |  230 |  14 |   SCLK.1 |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 233  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | PC10     | 16  | 74   |
 |   65 |  17 |      PC1 |    OFF | 0 | 27 || 28 |   |        |          |     |      |
 |  272 |  18 |     PI16 |   ALT2 | 0 | 29 || 30 |   |        |          |     |      |
 |  262 |  19 |      PI6 |    OFF | 0 | 31 || 32 |   |        |          |     |      |
 |  234 |  20 |     PH10 |   ALT3 | 0 | 33 || 34 |   |        |          |     |      |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
orangepi@orangepi:~$
```

1) The following takes pin 7—the corresponding GPIO is PC9—the corresponding wPi number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

2) The steps to test directly with the command are as follows:

a. First set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.pinMode(2, GPIO.OUTPUT) ;  "
```

b. Then set the GPIO port to output low level. After setting, you can use a

multimeter to measure the voltage value of the pin. If it is 0v, it means that the low level is set successfully.

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.LOW)"
```

c. Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.HIGH)"
```

3) The steps to test in the command line of python3 are as follows:

a. First use the python3 command to enter the command line mode of python3

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

c. Then set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
>>> wiringpi.wiringPiSetup()
0
>>> wiringpi.pinMode(2, GPIO.OUTPUT)
```

d. Then set the GPIO port to output low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 0v, it means that the low level is set successfully.

```
>>> wiringpi.digitalWrite(2, GPIO.LOW)
```

e. Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

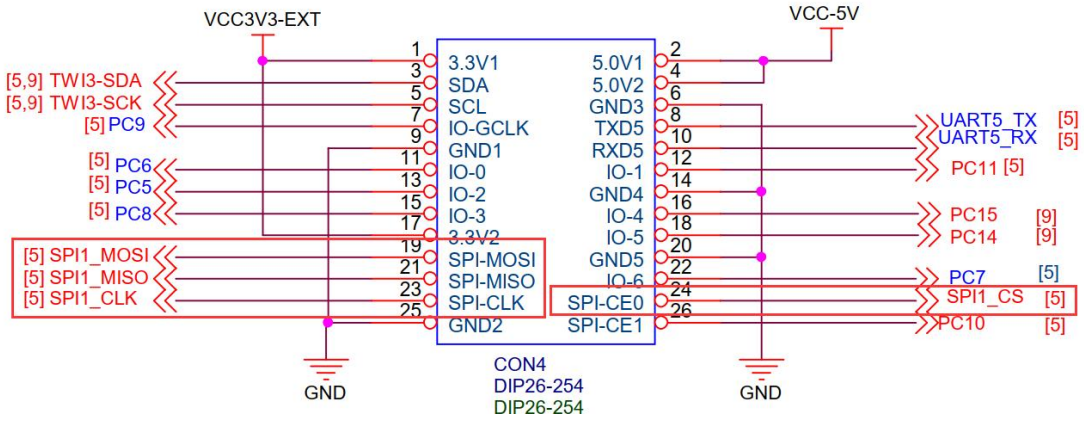
4) The method of wiringOP-Python to set GPIO high and low levels in python code can refer to the **blink.py** test program in the examples below. The **blink.py** test program will

set the voltage of all GPIO ports in the 26 pins of the development board to change continuously.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# ls blink.py
blink.py
root@orangepi:~/wiringOP-Python/examples# python3 blink.py
```

3.19.3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the available spi is spi1

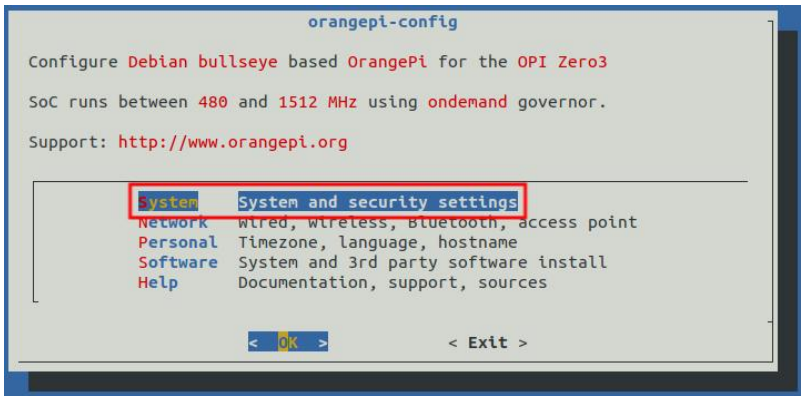


2) The spi1 is disabled by default in the Linux system and needs to be manually enabled before it can be used. The steps to open are as follows:

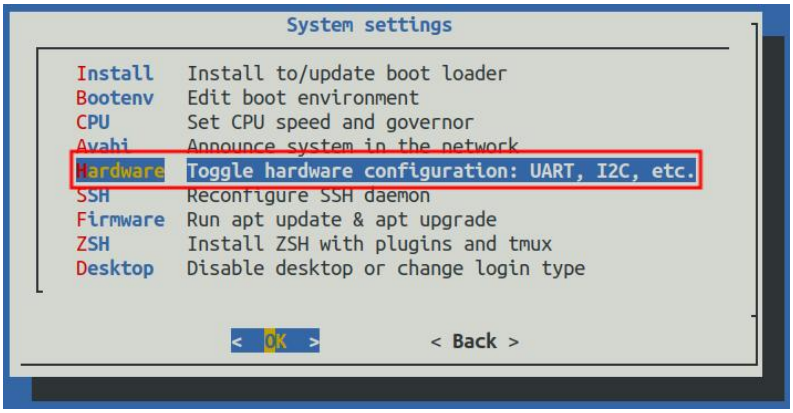
a. First run **orangepi-config**, normal users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

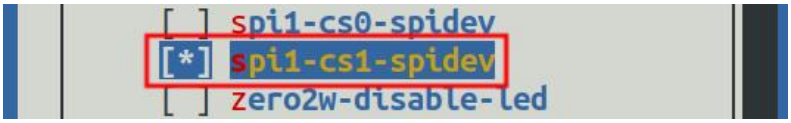
b. Then select **System**



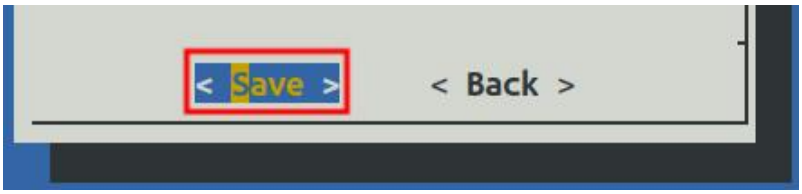
c. Then select **Hardware**



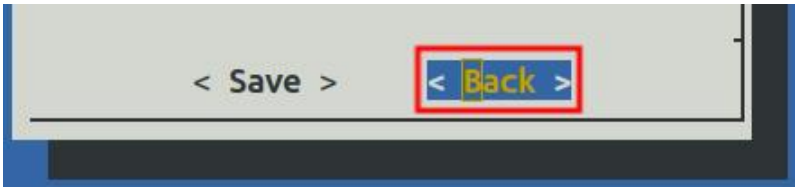
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **spi1-cs1-spidev**



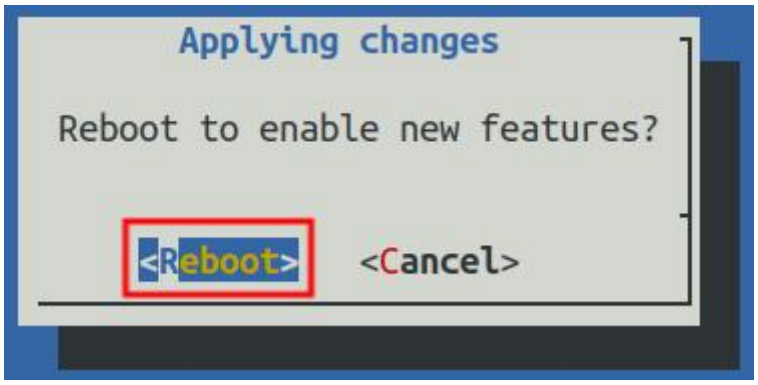
e. Then select **<Save>** to save



f. Then select **<Back>**然后选择**<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration take effect



3) Then check whether there is a **spidev1.1** device node in the Linux system. If it exists, it means that the configuration of SPI1 has taken effect

```
orangepi@orangepi:~$ ls /dev/spidev1*
/dev/spidev1.1
```

4) Then you can use the **spidev_test.py** program in the examples to test the loopback function of the SPI. The **spidev_test.py** program needs to specify the following two parameters:

a. **--channel**: Specify the channel number of SPI
b. **--port**: Specify the port number of SPI

5) Do not short-circuit the mosi and miso pins of SPI1, the output result of running spidev_test.py is as follows, you can see that the data of TX and RX are inconsistent

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 1 --port 1
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev1.1
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@.....|
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF    |................|
```

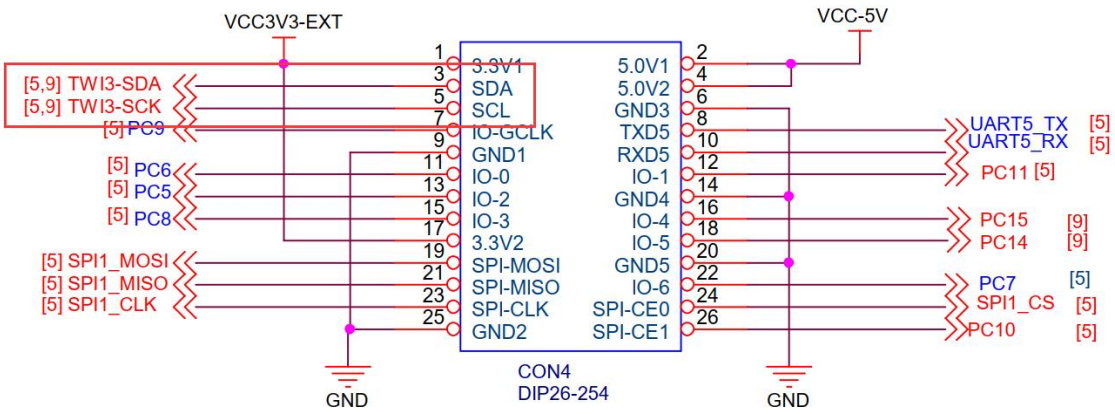
6) Then use the Dupont wire to short-circuit the two pins of txd (pin 19 in the 26pin interface) and rxd (pin 21 in the 26pin interface) of SPI1 and then run the output of spidev_test.py as follows, you can see The data sent and received are the same, indicating that the SPI1 loopback test is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 1 --port 1
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev1.1
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@.....|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
```

```
FF FF FF FF FF F0 0D   |.....@.....|
```

3. 19. 4. 26pin I2C test

1) According to the schematic diagram of 26pin, the available i2c is i2c3

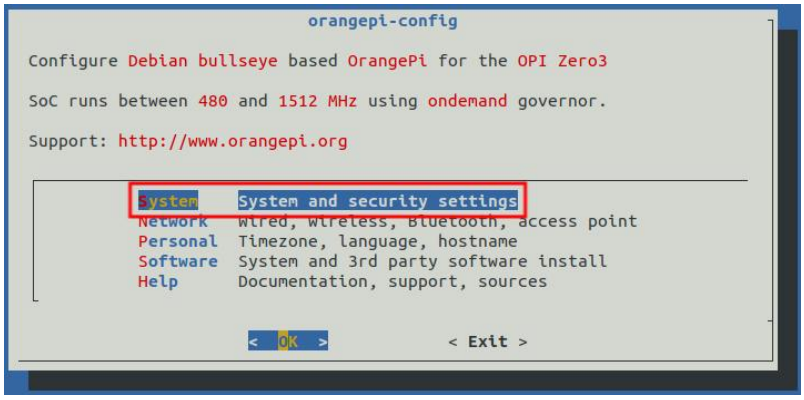


2) i2c3 is disabled by default in the Linux system, and it needs to be manually enabled before it can be used. The steps to open are as follows:

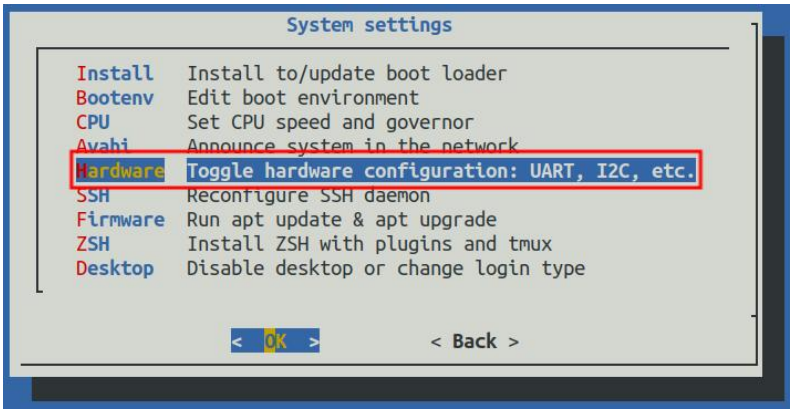
a. First run **orangepi-config**, normal users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**

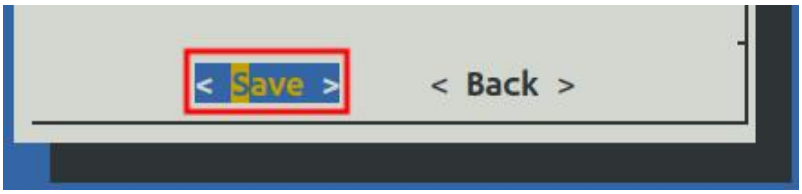


c. Then select **Hardware**

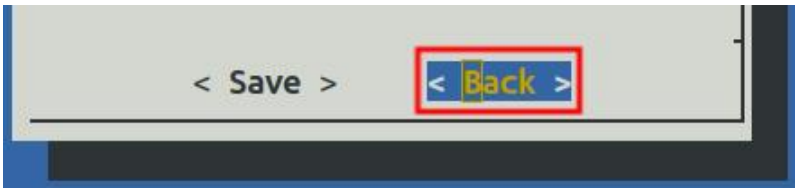


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-i2c3**

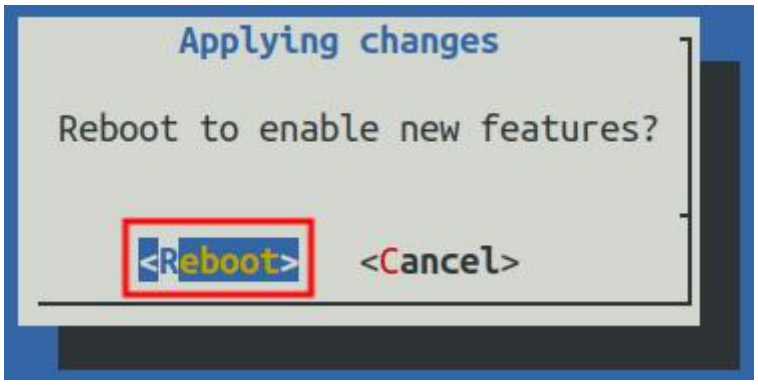
e. Then select **<Save>** to save



f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration take effect



3) After starting the linux system, first confirm that there is an i2c3 device node under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-3
/dev/i2c-3
```

4) Then start testing i2c, first install i2c-tools

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y i2c-tools
```

5) Then connect an i2c device to the i2c3 pin of the 26pin connector, here we take the DS1307 RTC module as an example

Pins of the RTC module	The pin corresponding to the 26pin of the development board
5V	**Pin 2**
GND	**Pin 6**
SDA	**Pin 3**
SCL	**Pin 5**

6) Then use the **i2cdetect -y 3** command, if the address of the connected i2c device can be detected, it means that the i2c device is connected correctly

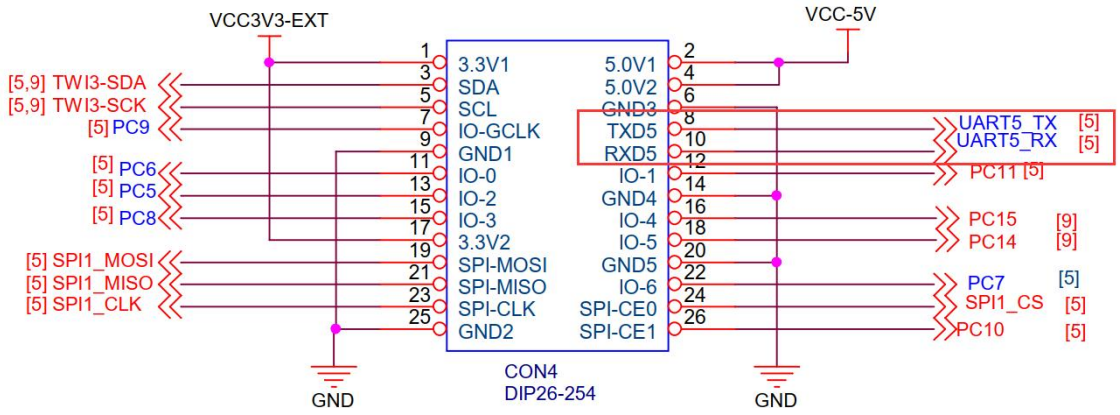
```
orangepi@orangepi:~$ sudo i2cdetect -y 3
[sudo] password for orangepi:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
orangepi@orangepi:~$
```

7) Then you can run the **ds1307.py** test program in the **examples** to read the RTC time

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device \
"/dev/i2c-3"
Thu 2022-06-16  04:35:46
Thu 2022-06-16  04:35:47
Thu 2022-06-16  04:35:48
^C
exit
```

3.19.5. 26pin UART test

1) According to the schematic diagram of the 26pin interface, the available uart is uart5

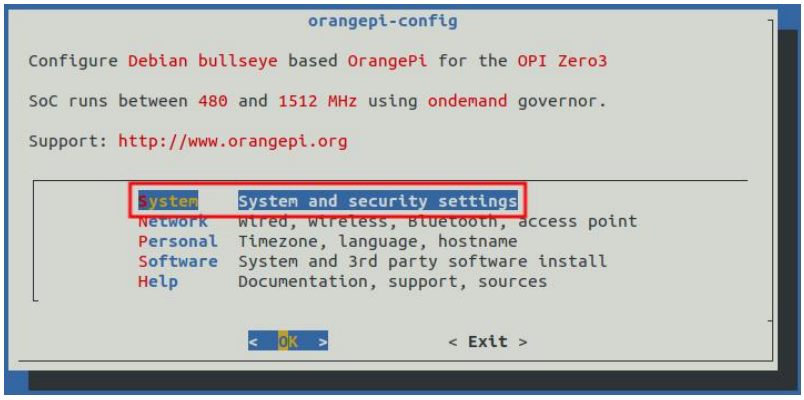


2) uart5 is disabled by default in the Linux system, and it needs to be opened manually to use it. The steps to open are as follows:

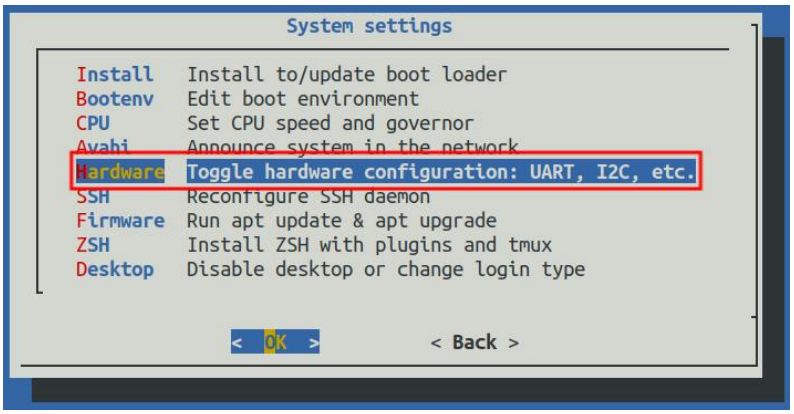
a. First run **orangepi-config**, normal users remember to add **sudo** permission

```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**

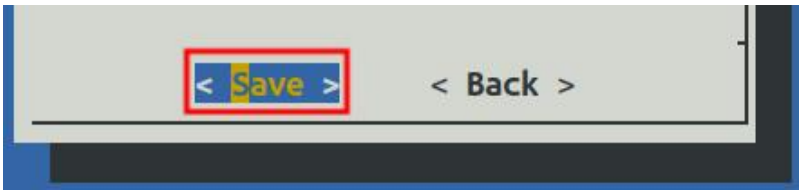


c. Then select **Hardware**

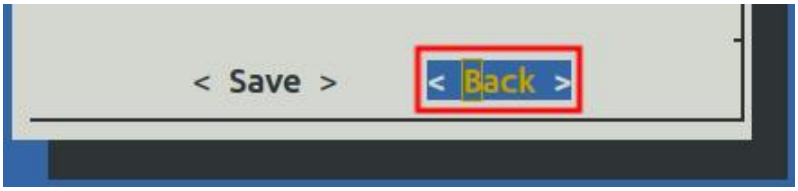


d. d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-uart5**

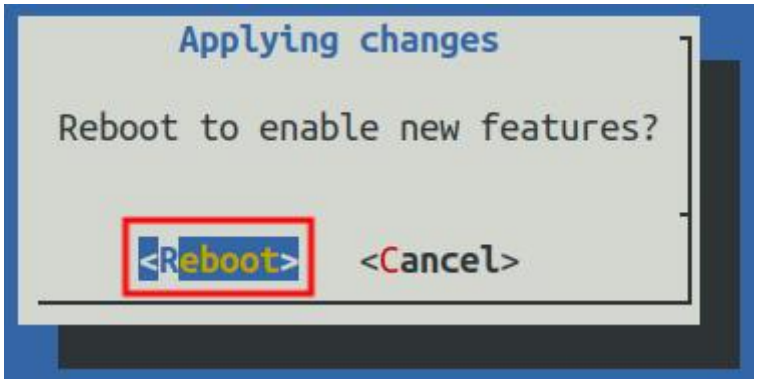
e. Then select **<Save>** to save



f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration take effect



3) After entering the linux system, first confirm whether there is a uart5 device node

under **/dev**

Note that the linux5.4 system is /dev/ttyAS5.

```
orangepi@orangepi:~$ ls /dev/ttyS5
/dev/ttyS5
```

4) Then start to test the uart5 interface, first use the DuPont line to short the rx and tx of the uart5 interface to be tested

	uart5
tx pin	Corresponding to pin 8 in 26pin
rx pin	Corresponding to pin 10 in 26pin

5) Finally, you can run the **serialTest.py** program in the examples to test the loopback function of the serial port. If you can see the following print, it means that the loopback test of the serial port is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device "/dev/ttyS5"        # linux6.1 使用
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device "/dev/ttyAS5"       # linux5.4 使用

Out:    0: ->    0
Out:    1: ->    1
Out:    2: ->    2
Out:    3: ->    3
Out:    4:^C
exit
```

3. 20. Hardware Watchdog Test

The watchdog_test program is pre-installed in the linux system released by Orange PI, which can be tested directly.。

The method of running the watchdog_test program is shown below：

a. The second parameter 10 indicates the counting of the door dog. If the dog is not fed in this time, the system will restart
b. We can feed the dog by pressing any key (except ESC) on the keyboard. After the dog is fed, the program will print a line of Keep Alive to indicate that the dog

is successful.

```
orangepi@orangepi:~$ sudo watchdog_test 10
open success
options is 33152,identity is sunxi-wdt
put_usr return,if 0,success:0
The old reset time is: 16
return ENOTTY,if -1,success:0
return ENOTTY,if -1,success:0
put_user return,if 0,success:0
put_usr return,if 0,success:0
keep alive
keep alive
keep alive
```

3.21. Check the chipid of the H618 chip

The command of the H618 chip Chipid is shown below. The chipid of each chip is different, so you can use Chipid to distinguish multiple development boards.

```
orangepi@orangepi:~$ cat /sys/class/sunxi_info/sys_info | grep "chipid"
sunxi_chipid            : 338020004c004808014747882468 1ed1
```

3.22. Python related instructions

3.22.1. Python source code compilation and installation method

If the Python version in the Ubuntu or Debian system software warehouse does not meet the requirements of development, if you want to use the latest version of Python, you can use the following method to download the source code package of Python to compile and install the latest version of Python.

The following demonstration is the latest version of the compilation and installation of Python3.9. If you want to compile and install other versions of Python, the method is the same (need to download the source code corresponding to the Python you want to install).

1) First install the dependency package required to compile Python

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y build-essential zlib1g-dev  \
libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libsqlite3-dev  \
libreadline-dev libffi-dev curl libbz2-dev
```

2) Then download the latest version of the Python3.9 source code and decompress it

```
orangepi@orangepi:~$ wget  \
https://www.python.org/ftp/python/3.9.10/Python-3.9.10.tgz
orangepi@orangepi:~$ tar xvf Python-3.9.10.tgz
```

3) Then run the configuration command

```
orangepi@orangepi:~$ cd Python-3.9.10
orangepi@orangepi:~$ ./configure --enable-optimizations
```

4) Then compile and install python3.9, the compilation time takes about half an hour

```
orangepi@orangepi:~$ make -j4
orangepi@orangepi:~$ sudo make altinstall
```

5) After installation, you can use the following command to view the Python version number just installed

```
orangepi@orangepi:~$ python3.9 --version
Python 3.9.10
```

6) Then update pip

```
orangepi@orangepi:~$ /usr/local/bin/python3.9 -m pip install --upgrade pip
```

3. 22. 2. Python to replace the pip source method

The Linux system PIP's default source is the official source of Python, but the source of the official Python official visits is very slow, and the Python software packaging failure is often caused by network reasons. So when installing the Python library with PIP, remember to replace the pip source

1) First install **python3-pip**

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3-pip
```

2) Method of permanent replacement of pip source under Linux

a. Create a new **~/.pip** directory first, then add the **pip.conf** configuration file, and set the source of the pip as Tsinghua source

```
orangepi@orangepi:~$ mkdir -p ~/.pip
orangepi@orangepi:~$ cat <<EOF > ~/.pip/pip.conf
[global]
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
EOF
```

b. Then install the Python library with pip3, it will be fast

3) The method of temporarily replacing the pip source under Linux, the **<packagename>** needs to be replaced with a specific package name

```
orangepi@orangepi:~$ pip3 install <packagename> -i  \
https://pypi.tuna.tsinghua.edu.cn/simple --trusted-host pypi.tuna.tsinghua.edu.cn
```

3. 23. The Method Of Installing The Docker

Orange Pi's Linux image has been pre-installed with Docker, but the Docker service has not been opened by default. Use **enable_docker.sh** script to enable the docker service, and then you can start using the docker command, and the docker service will be automatically activated at the next start of the system.

```
orangepi@orangepi:~$ enable_docker.sh
```

You can use the following command to test the docker,If you can run **hello-world** to indicate that docker can be used normally.

```
orangepi@orangepi:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
256ab8fe8778: Pull complete
Digest:
sha256:7f0a9f93b4aa3022c3a4c147a449ef11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
This message shows that your installation appears to be working correctly.
.....
```

When using the docker command, if it prompts **permission denied**, please add the current user to the docker user group, so that the docker command can be run without sudo。

```
orangepi@orangepi:~$ sudo usermod -aG docker $USER
```

Note: You need to log out of the system and log in again to take effect, and restarting the system is also OK.

3. 24. How to install Home Assistant

Note that only the method of installing Home Assistant in Ubuntu or Debian system will be provided here. For detailed usage of Home Assistant, please refer to official documents or corresponding books。

3. 24. 1. Install via docker

1) First of all, please install docker and make sure that docker can run normally. The Method Of Installing The Docker, please refer to the instructions in the section on how to install Docker。

2) Then you can search for the docker image of Home Assistant

```
orangepi@orangepi:~$ docker search homeassistant
```

3) Then use the following command to download the docker image of Home Assistant to the local. The size of the image is about 1GB, and the download time will be relatively long. Please wait patiently for the download to complete

```
orangepi@orangepi:~$ docker pull homeassistant/home-assistant
Using default tag: latest
latest: Pulling from homeassistant/home-assistant
be307f383ecc: Downloading
5fbc4c07ac88: Download complete
..... (omit some output)
```

```
3cc6a1510c9f: Pull complete
7a4e4d5b979f: Pull complete
Digest:
sha256:81d381f5008c082a37da97d8b08dd8b358dae7ecf49e62ce3ef1eeaefc4381bb
Status: Downloaded newer image for homeassistant/home-assistant:latest
docker.io/homeassistant/home-assistant:latest
```

4) Then you can use the following command to view the docker image of Home Assistant just downloaded

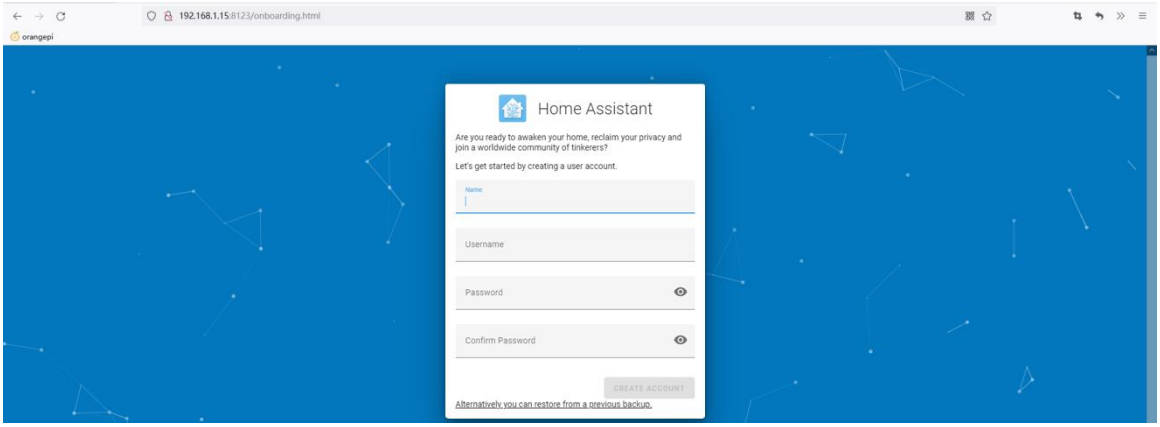
```
orangepi@orangepi:~$ docker images homeassistant/home-assistant
REPOSITORY                  TAG       IMAGE ID       CREATED        SIZE
homeassistant/home-assistant  latest    bfa0ab9e1cf5   2 months ago   1.17GB
```

5) Now you can run the docker container of Home Assistant

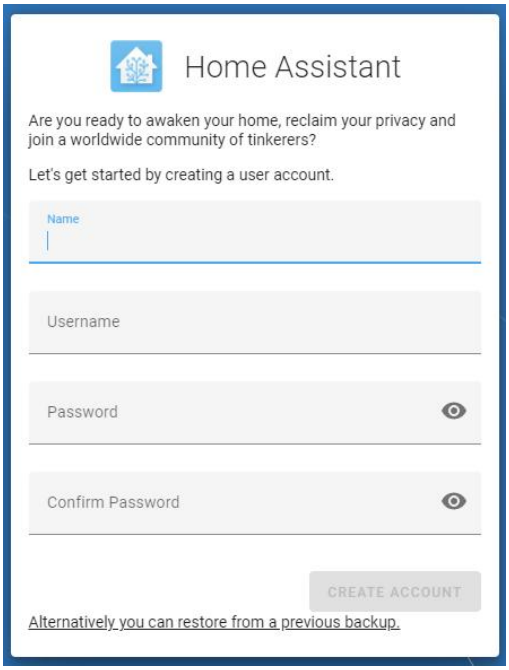
```
orangepi@orangepi:~$ docker run -d \
  --name homeassistant \
  --privileged \
  --restart=unless-stopped \
  -e TZ=Asia/Shanghai \
  -v /home/orangepi/home-assistant:/config \
  --network=host \
  homeassistant/home-assistant:latest
```

6) Then enter【IP address of the development board: 8123】in the browser to see the Home Assistant interface

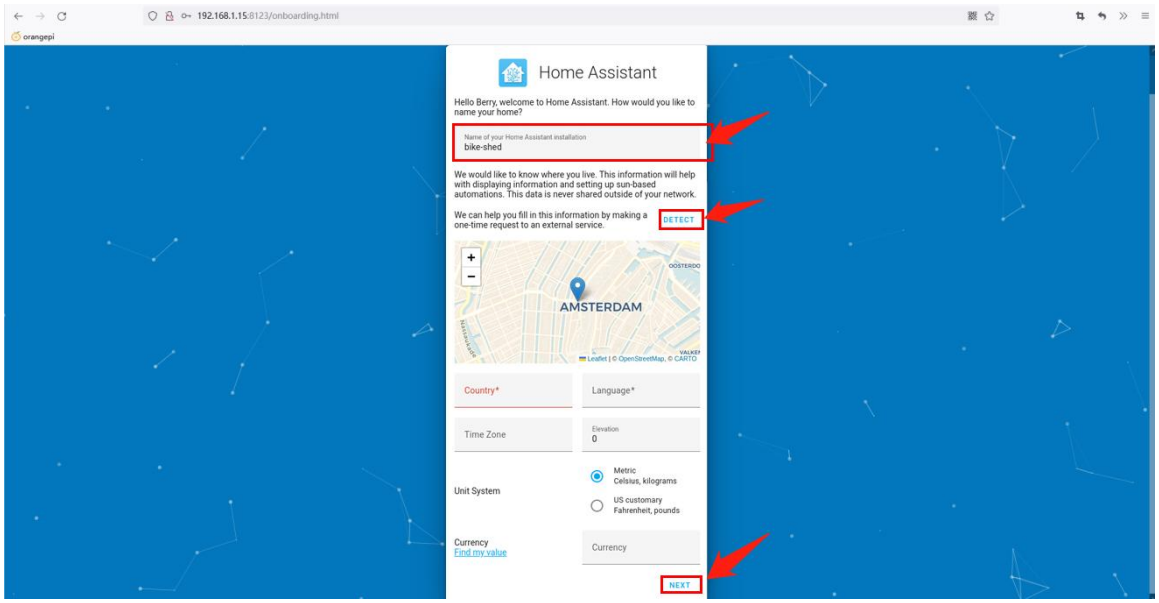
It takes a while for the Home Assistant container to start. If the following interface is not displayed normally, please wait for a few seconds and then refresh. If the following interface is not displayed normally after waiting for more than one minute, it means that there is a problem with the installation of Home Assistant. At this time, you need to check whether there is a problem with the previous installation and setting process。



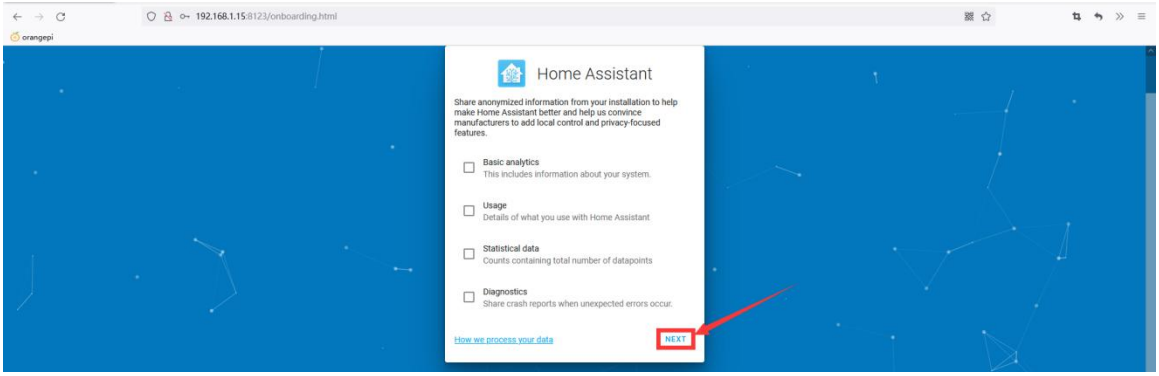
7) Then enter **your name**, **user name** and **password** and click **Create Account**



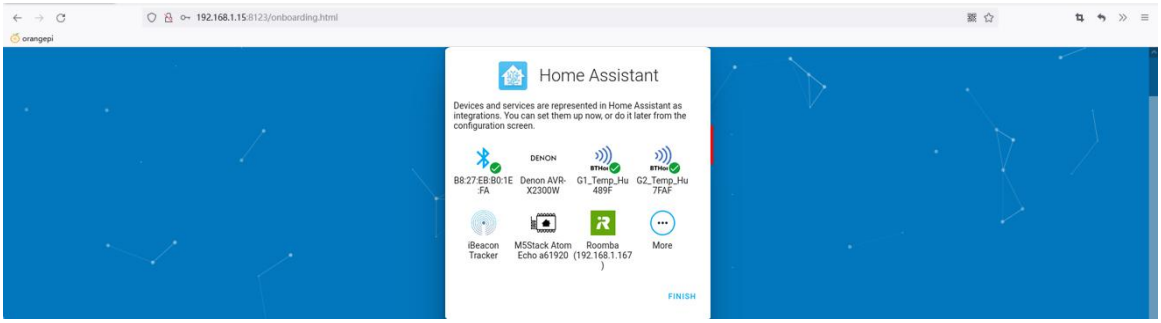
8) Then follow the interface prompts to set according to your preferences, and then click Next



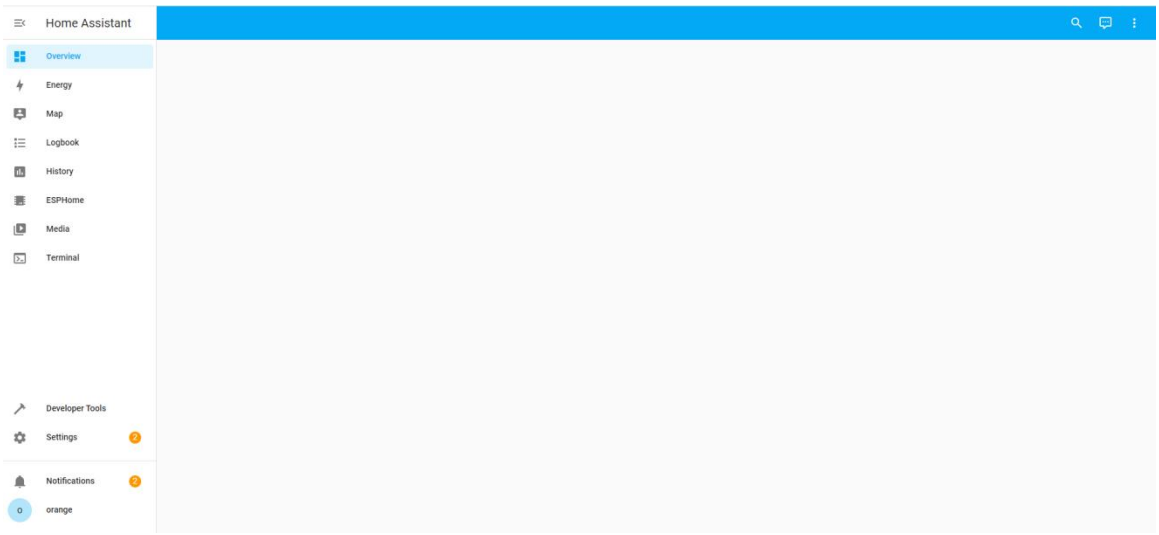
9) Then click Next



10) Then click Finish



11) The final main interface displayed by Home Assistant is shown in the figure below



12) How to stop the Home Assistant container

a. The command to view the docker container is as follows

```
orangepi@orangepi:~$ docker ps -a
```

b. The command to stop the Home Assistant container is as follows

```
orangepi@orangepi:~$ docker stop homeassistant
```

c. The command to delete the Home Assistant container is as follows

```
orangepi@orangepi:~$ docker rm homeassistant
```

3. 24. 2. Install via python

Before installation, please replace the source of pip with a domestic source to speed up the installation of the Python package. For the configuration method, see the instructions in the section on Python to replace the pip source method.

1) First install the dependency package

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3 python3-dev python3-venv    \
python3-pip libffi-dev libssl-dev libjpeg-dev zlib1g-dev autoconf build-essential    \
libopenjp2-7 libtiff5 libturbojpeg0-dev tzdata
```

If it is debian12, please use the following command：

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3 python3-dev python3-venv    \
python3-pip libffi-dev libssl-dev libjpeg-dev zlib1g-dev autoconf build-essential    \
```

libopenjp2-7 libturbojpeg0-dev tzdata

2) Then you need to compile and install Python3.9. For the method, please refer to the section on **how to compile and install Python source code**

The default Python version of Debian Bullseye is Python3.9, so there is no need to compile and install。

The default Python version of Ubuntu Jammy is Python3.10, so there is no need to compile and install。

The default Python version of Debian Bookworm is Python3.11, so there is no need to compile and install。

3) Then create a Python virtual environment

Debian Bookworm is python3.11, please remember to replace the corresponding command。

```
orangepi@orangepi:~$ sudo mkdir /srv/homeassistant
orangepi@orangepi:~$ sudo chown orangepi:orangepi /srv/homeassistant
orangepi@orangepi:~$ cd /srv/homeassistant
orangepi@orangepi:~$ python3.9 -m venv .
orangepi@orangepi:~$ source bin/activate
(homeassistant) orangepi@orangepi:/srv/homeassistant$
```

4) Then install the required Python package

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ python3 -m pip install wheel
```

5) Then you can install Home Assistant Core

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ pip3 install homeassistant
```

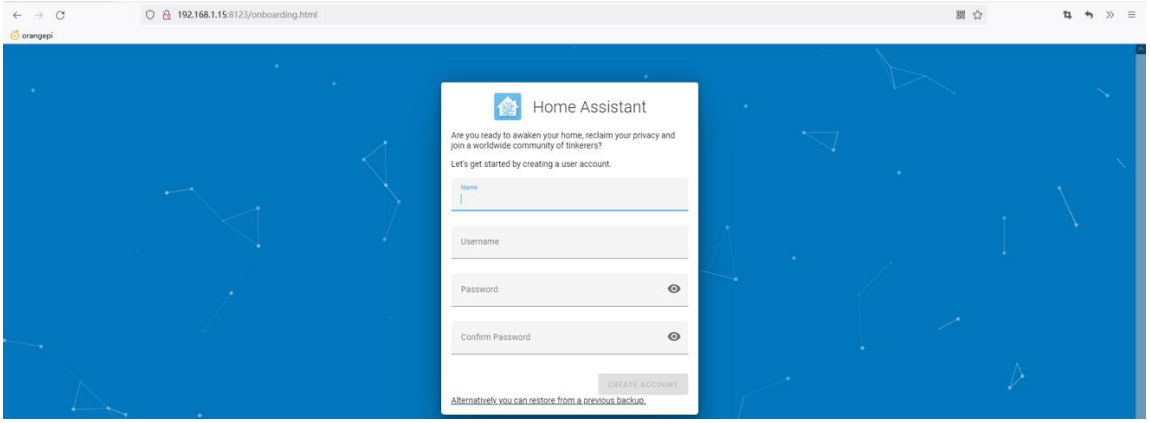
6) Then enter the following command to run Home Assistant Core

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ hass
```

7) Then enter **[IP address of the development board: 8123]** in the browser to see the interface of Home Assistant

When running the hass command for the first time, you will download and install and cache some necessary libraries and dependencies to run. This process

may take a few minutes. Note that at this time, the interface of Home Assistant cannot be seen in the browser. Please wait for a while before refreshing。



3. 25. OpenCV installation method

3. 25. 1. Use apt to install OpenCV

1) The installation command is shown below

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y libopencv-dev python3-opencv
```

2) Then use the following command to print the output of the version number of OpenCV, indicating that the installation of OpenCV is successful

a. The version of OpenCV in ubuntu22.04 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.5.4
```

b. The version of OpenCV in Ubuntu 20.04 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.2.0
```

c. The version of OpenCV in Debian11 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.5.1
```

d. The version of OpenCV in Debian12 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.6.0
```

3. 26. The installation method of the aapanel Linux panel

aapanel Linux panel is a server management software that improves operation and maintenance efficiency. It supports more than 100 server management functions such as one -click LAMP/LNMP/cluster/monitoring/website/FTP/database/Java (excerpted from Baota official website)

1) First of all, you need to expand the/**tmp** memory. After setting, you need to **restart the linux system of the development board，** and the command is shown below：

```
orangepi@orangepi:~$ sudo sed -i 's/nosuid/&,size=2G/' /etc/fstab
orangepi@orangepi:~$ sudo reboot
```

2) After restarting, you can see that the size of the/**tmp** space has become 2G

```
orangepi@orangepi:~$ df -h | grep "/tmp"
tmpfs                   2.0G     12K  2.0G      1% /tmp
```

3) Then enter the following command in the Linux system to start the installation of the aapanel

```
orangepi@orangepi:~$ sudo install_bt_panel.sh
```

4) Then the aapanel installation program reminds whether to install the **Bt-Panel** to the/**www** folder, and enter Y at this time

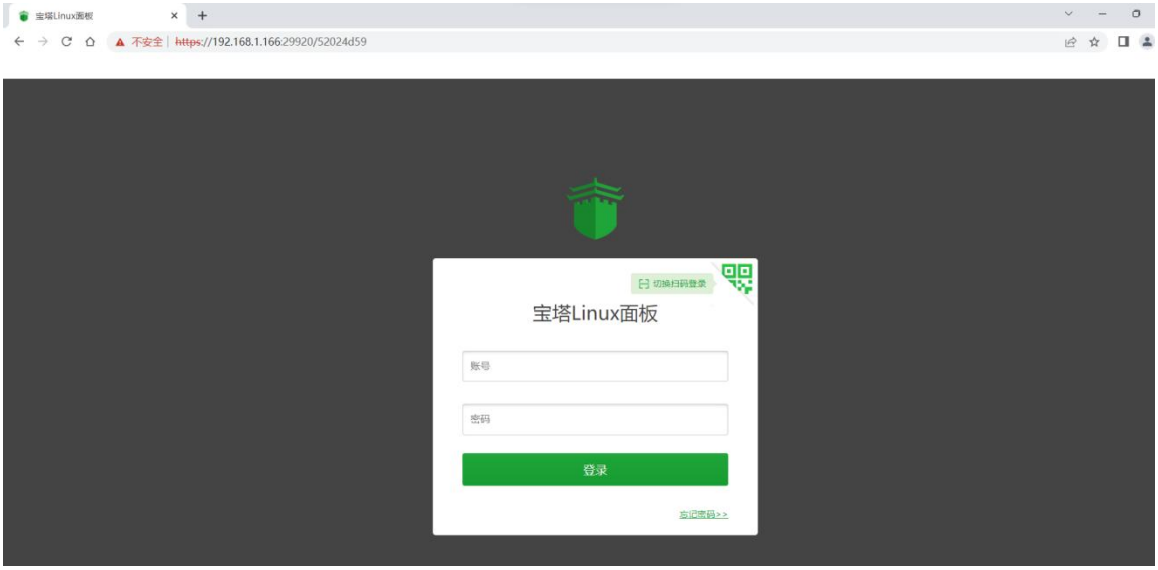
```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| The WebPanel URL will be http://SERVER_IP:8888 when installed.
+----------------------------------------------------------------------

Do you want to install Bt-Panel to the /www directory now?(y/n): y
```

5) Then you have to wait patiently. When you see the printing information below the terminal output, it means that the pagoda has been installed. The entire installation

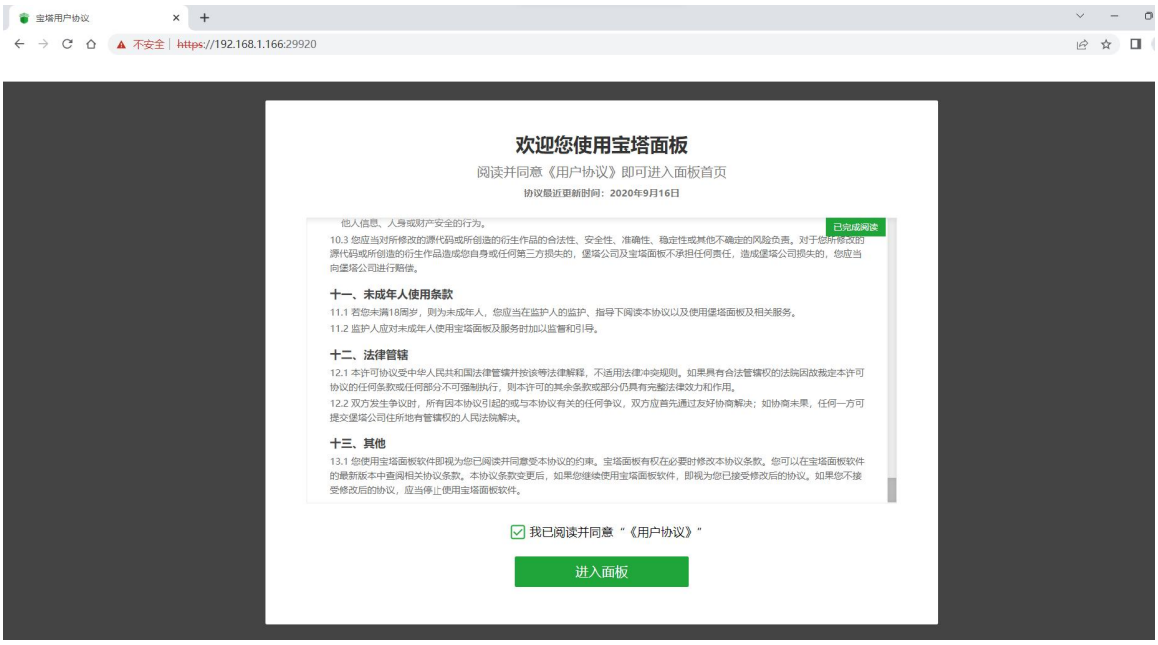
process takes about 34 minutes. There may be some differences according to the difference in network speed

```
==========================================================
Congratulations! Installed successfully!
==========================================================
外网面板地址: https://183.15.204.194:29920/52024d59
内网面板地址: https://192.168.1.166:29920/52024d59
username: 4qhagfrc
password: 27b2d026
If you cannot access the panel,
release the following panel port [29920] in the security group
若无法访问面板，请检查防火墙/安全组是否有放行面板[29920]端口
因已开启面板自签证书，访问面板会提示不匹配证书，请参考以下链接配置证书
https://www.bt.cn/bbs/thread-105443-1-1.html
==========================================================
Time consumed: 34 Minute!
orangepi@orangepi:~$
```

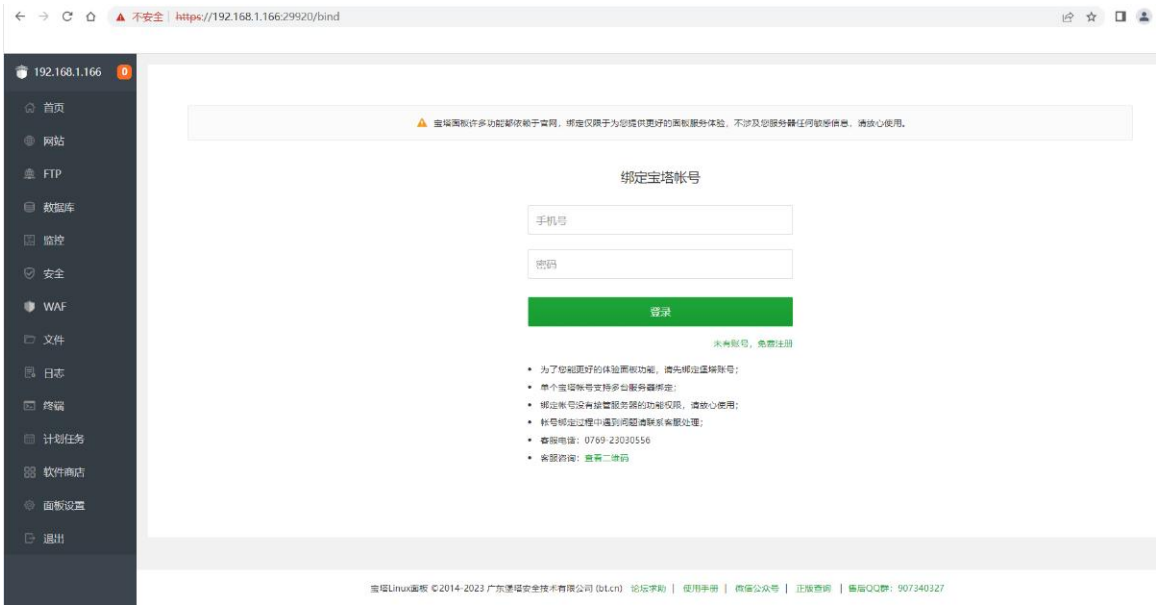
6) At this time, enter the **panel address** displayed above in the browser to open the login interface of the aapanel Linux panel, and then enter the **username** and **password** displayed in the corresponding position to log in to the aapanel



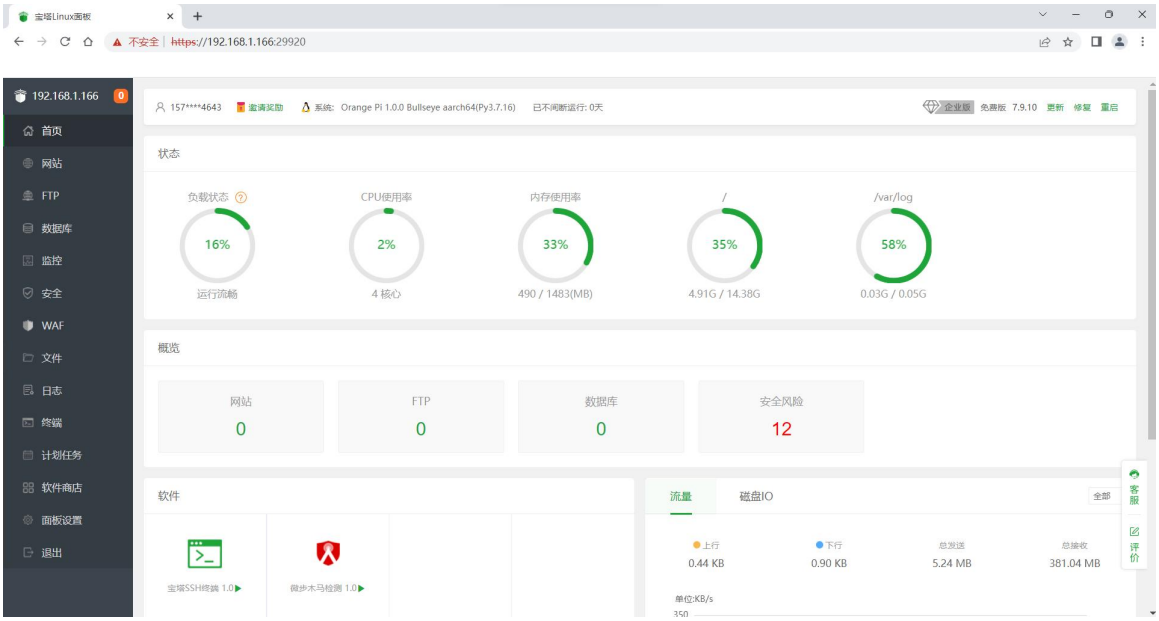
7) After successfully logging in to the aapanel, the following welcome interface will pop up. First, please take the intermediate user notice to read to the bottom, and then you can choose "I have agreed and read" User Agreement ", and then click" Enter the panel " You can enter the aapanel



8) After entering the aapanel, you will first prompt that you need to bind the account of the aapanel official website. If you do n't have an account, you can go to the aapanel 's official website (**https://www.bt.cn**) to register one



9) The final display interface is shown in the figure below. You can intuitively see some status information of the development board Linux system, such as load state, CPU usage, memory usage and storage space usage



10) For more functions of the aapanel, please refer to the following information to explore by yourself

Use manual：**http://docs.bt.cn**
Forum website：**https://www.bt.cn/bbs**
GitHub Link：**https://github.com/aaPanel/BaoTa**

3. 27. "face_recognition" The installation and testing method of the face recognition library

Note that the contents of this section are tested in the Linux system of the desktop version, so please make sure that the system used by the development board is a desktop version system。

In addition, the installation test below is performed under OrangePi users. Please keep the environment consistent.

Debian12 is currently not adapted.

The address of the source code warehouse of Face_recognition is：
https://github.com/ageitgey/face_recognition

Face_recognition Chinese version of the explanation document is：
https://github.com/ageitgey/face_recognition/blob/master/README_Simplified_Chi

nese.md

3.27.1. The method of using script to automatically install face_recognition

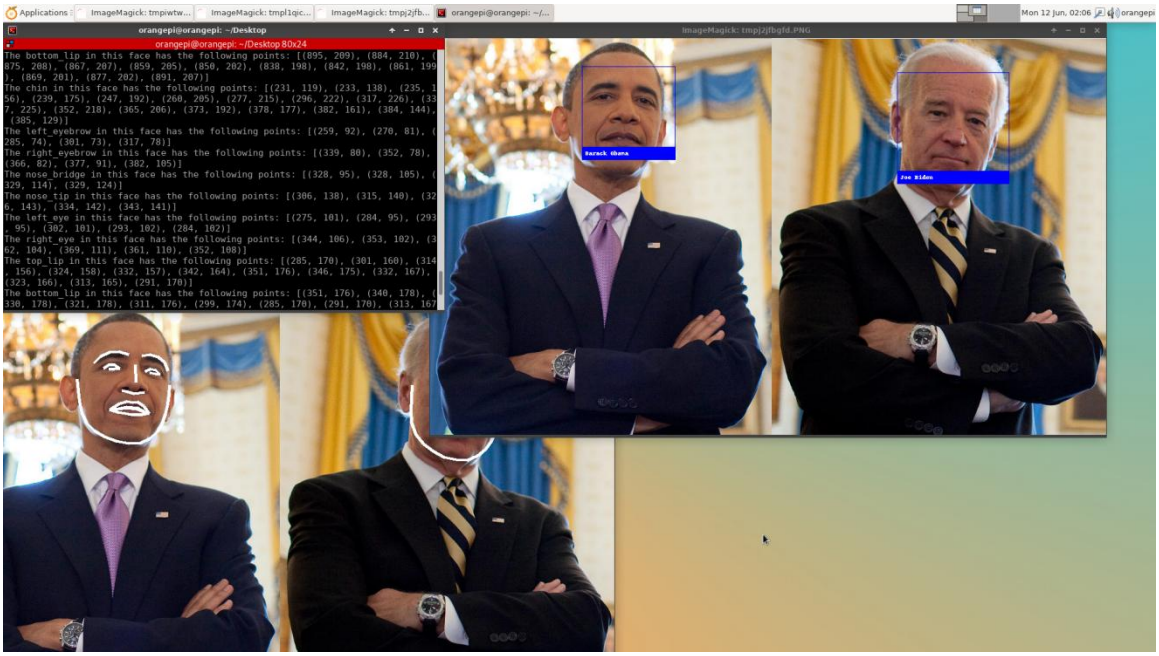
1) First open a terminal on the desktop**, then download face_recognition_install.sh**

```
orangepi@orangepi:~/Desktop$ wget \
https://gitee.com/leeboby/face_recognition_install/raw/master/face_recognition_install.sh
```

2) Then execute the following command to start the installation **face_recognition**

```
orangepi@orangepi:~/Desktop$ bash face_recognition_install.sh
```

3) After face_recognition is installed, it will automatically download the source code of face_recognition, and then automatically run some examples in face_recognition. If you can finally see the following pictures popping up on the desktop, it means that the face_recognition installation test is successful.



3.27.2. Manually install face_recognition

1) First create a new **~/.pip** directory, then add the **pip.conf** configuration file, and set the mirror source of pip to Tsinghua source in it, the commands to be executed are as follows：

```
orangepi@orangepi:~$ mkdir -p ~/.pip
orangepi@orangepi:~$ cat <<EOF > ~/.pip/pip.conf
[global]
```

```
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
EOF
```

2) Then install dependencies

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y python3-pip libopencv-dev \
python3-opencv imagemagick python3-scipy python3-setuptools python3-wheel \
python3-dev cmake python3-testresources
```

3) Then update pip3

```
orangepi@orangepi:~$ python3 -m pip install -U pip setuptools wheel
```

4) Before installing **face_recognition**, you first need to install the **dlib** library. Since the compilation and installation of the dlib library on the development board is relatively slow, I saved a compiled dlib whl file on **gitee**, and you can install it directly after downloading. The download address of the dlib whl file is as follows：

```
https://gitee.com/leeboby/python_whl
```

a. First download the python_whl warehouse to the Linux system of the development board

```
orangepi@orangepi:~$ git clone --depth=1 https://gitee.com/leeboby/python_whl
```

b. In the python_whl folder, you can see that there are multiple versions of dlib installation packages. The Linux systems corresponding to different versions of dlib are as follows：

Ubuntu20.04	**dlib-19.24.0-cp38-cp38-linux_aarch64.whl**
Ubuntu22.04	**dlib-19.24.0-cp310-cp310-linux_aarch64.whl**
Debian11	**dlib-19.24.0-cp39-cp39-linux_aarch64.whl**

c. Then you can start installing dlib, the command is as follows

a) Ubuntu20.04

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install dlib-19.24.0-cp38-cp38-linux_aarch64.whl
```

b) Ubuntu22.04

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install dlib-19.24.0-cp310-cp310-linux_aarch64.whl
```

c) Debian11

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install dlib-19.24.0-cp39-cp39-linux_aarch64.whl
```

d. After installation, if the version number of dlib can be printed normally by using the following command, it means that dlib is installed correctly

```
orangepi@orangepi:~/python_whl$ python3 -c "import dlib; print(dlib.__version__)"
19.24.0
```

5) Then install **face_recognition_models-0.3.0-py2.py3-none-any.whl**

```
orangepi@orangepi:~/python_whl$ python3 -m pip install face_recognition_models-0.3.0-py2.py3-none-any.whl
```

6) Then install **face_recognition**

```
orangepi@orangepi:~$ python3 -m pip install face_recognition
```

7) Then you need to **reopen a terminal** to find and run the two commands **face_detection** and **face_recognition**

a. The face_recognition command is used to recognize whose face is in a single image or a folder of images.

b. The face_detection command is used to locate the face in a single picture or a picture folder

```
orangepi@orangepi:~$ which face_detection
/usr/local/bin/face_detection
orangepi@orangepi:~$ which face_recognition
/usr/local/bin/face_recognition
```

If you can't find the above two commands after reopening the terminal, please try to manually import environment variables, and then test again

```
orangepi@orangepi:~$ export PATH=/home/orangepi/.local/bin:$PATH
```

3. 27. 3. The test method of face_recognition

Note that the following operations are demonstrated on the desktop, so please connect the HDMI monitor first, or use NoMachine/VNC to remotely log in to the Linux desktop to test。

1) There are some sample codes in the source code of **face_recognition**, which we can use directly for testing. The download address of the source code of face_recognition is

as follows：

a. GitHub official download address

```
orangepi@orangepi:~$ git clone https://github.com/ageitgey/face_recognition.git
```

b. Gitee image download address

```
orangepi@orangepi:~$ git clone https://gitee.com/leeboby/face_recognition.git
```

2) The path to the face_recognition sample code is as follows

```
face_recognition/examples
```

3) The Chinese description document link of face_recognition is as follows, please read it carefully before using face_recognition

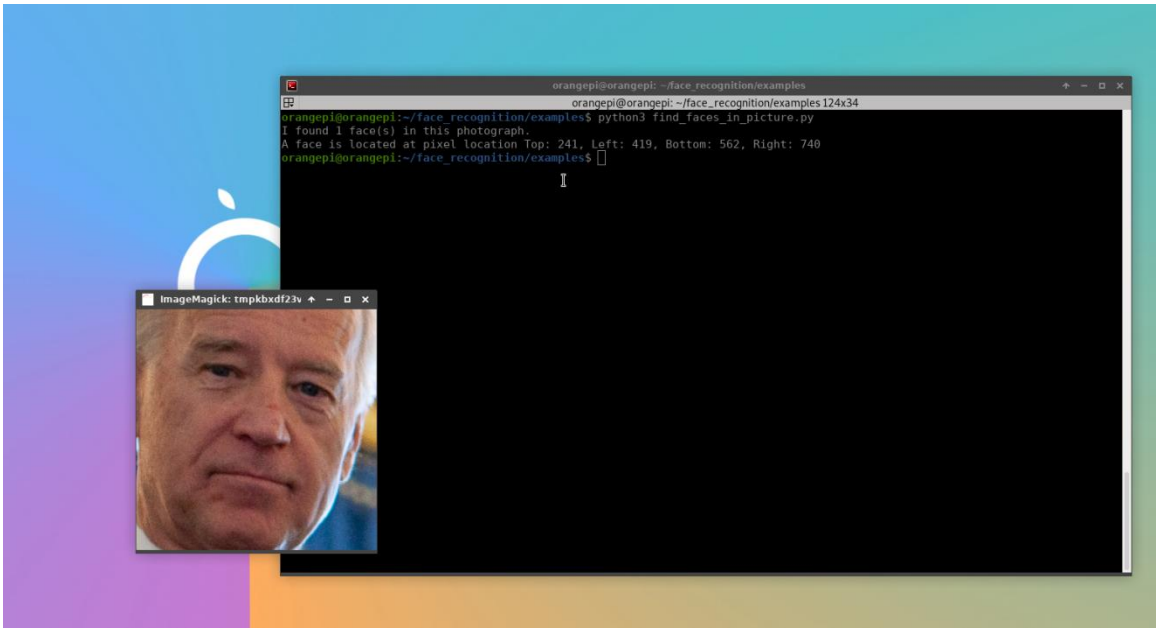
```
https://github.com/ageitgey/face_recognition/blob/master/README_Simplified_Chinese.md
```

4) **find_faces_in_picture.py** is used to locate the position of the face in the picture, the test steps are as follows

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 find_faces_in_picture.py
I found 1 face(s) in this photograph.
A face is located at pixel location Top: 241, Left: 419, Bottom: 562, Right: 740
```

b. Wait for a while and the following picture will pop up, which is the face located in the test picture

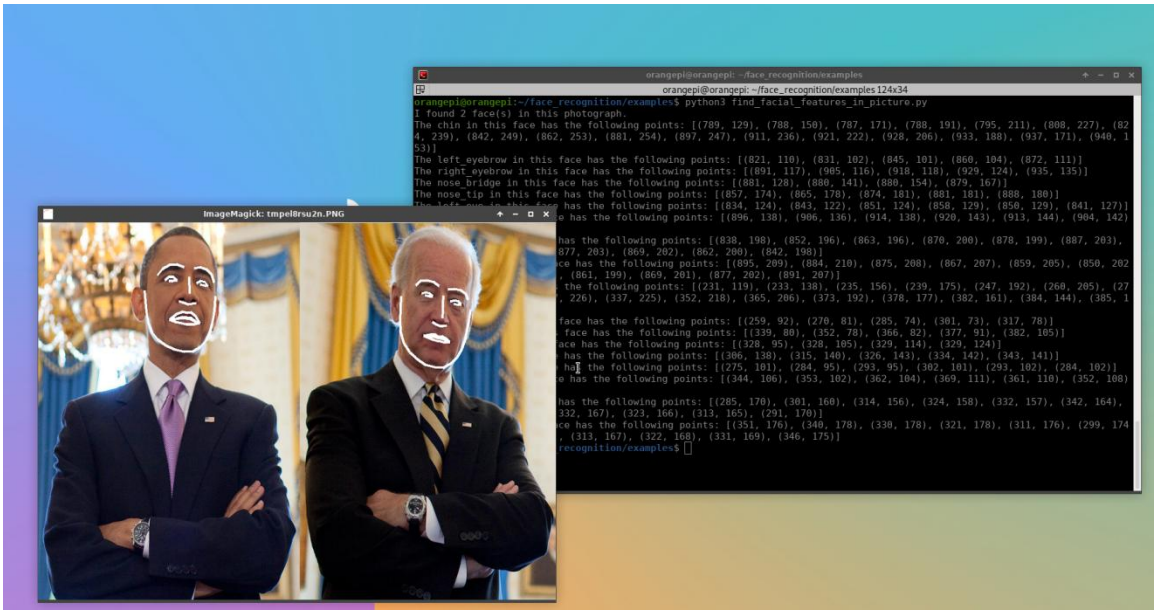


5) **find_facial_features_in_picture.py** is used to identify the key points of the face in a single picture, and the test steps are as follows

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 find_facial_features_in_picture.py
```

b. After waiting for a while, the following picture will pop up, and you can see that the outline of the face is marked

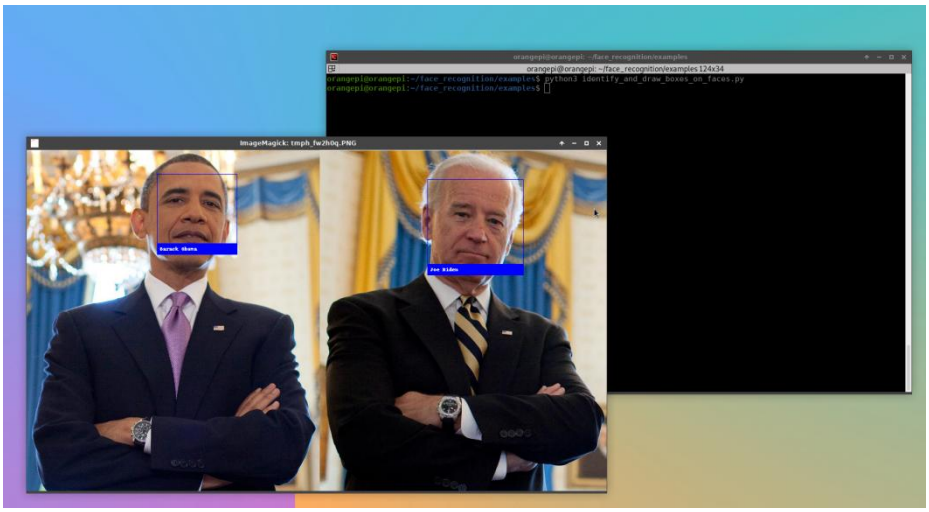


6) **identify_and_draw_boxes_on_faces.py** is used to identify faces and mark them with boxes. The test steps are as follows

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 identify_and_draw_boxes_on_faces.py
```

b. After waiting for a while, the following picture will pop up. You can see that the faces in the picture are marked with boxes, and the names of the characters are displayed correctly



7) **face_distance.py** is used to compare whether two faces belong to the same person at different precisions. First open a terminal, then enter the **face_recognition/examples** directory, and then execute the following command to see the output of the test

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 face_distance.py
The test image has a distance of 0.35 from known image #0
- With a normal cutoff of 0.6, would the test image match the known image? True
- With a very strict cutoff of 0.5, would the test image match the known image? True

The test image has a distance of 0.82 from known image #1
- With a normal cutoff of 0.6, would the test image match the known image? False
- With a very strict cutoff of 0.5, would the test image match the known image?
  False
```

8) **recognize_faces_in_pictures.py** is used to identify who the face in the unknown

picture is. First open a terminal, then enter the **face_recognition/examples** directory, and then execute the following command, and you can see the test results after waiting for a while

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 recognize_faces_in_pictures.py
Is the unknown face a picture of Biden? False
Is the unknown face a picture of Obama? True
Is the unknown face a new person that we've never seen before? False
```

9) **facerec_from_webcam_faster.py** is used to recognize the face in the USB camera, the test steps are as follows：

a. First, please insert the USB camera into the USB interface of the development board, and then use the **v4l2-ctl** (**note that l in v4l2 is a lowercase letter l, not the number 1**) command to check the serial number of the device node of the USB camera

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y v4l-utils
orangepi@orangepi:~$ v4l2-ctl --list-devices
cedrus (platform:cedrus):
        /dev/video0

USB2.0 UVC PC Camera: USB2.0 UV (usb-5311000.usb-1):
        /dev/video1
        /dev/video2
```

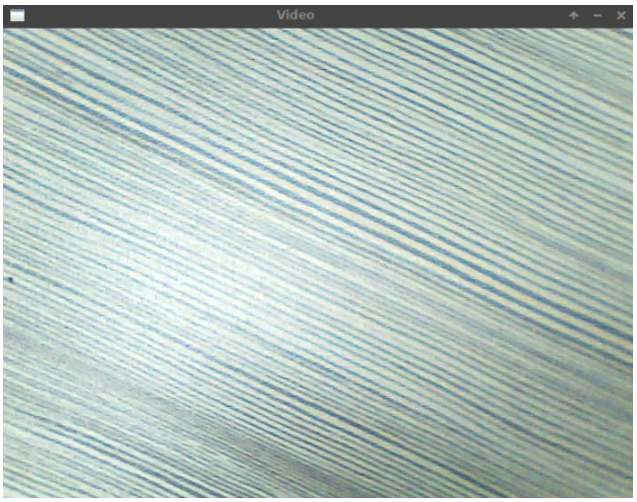
b. Then open a terminal on the desktop, enter the **face_recognition/examples** directory, and first modify the device serial number of the camera used in **facerec_from_webcam_faster.py.** For example, through the **v4l2-ctl --list-devices** command above, you can see that the USB camera is **/dev/video1**, then modify the **0** in **cv2.VideoCapture(0)** to **1**

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ vim facerec_from_webcam_faster.py
video_capture = cv2.VideoCapture(1)
```

c. Then execute the following command to run **facerec_from_webcam_faster.py**

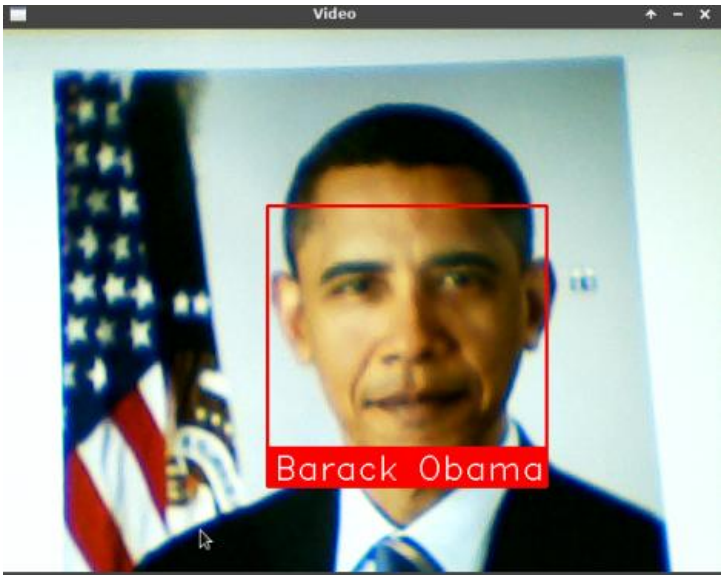
```
orangepi@orangepi:~/face_recognition/examples$ python3 facerec_from_webcam_faster.py
```

d. Wait for a while and the camera display screen will pop up



e. At this point, you can point the camera at yourself. When the camera detects a face, it will frame the detected face with a square. **Note that when detecting a face, the image displayed by the camera will be relatively slow, please do not move too fast**

f. You can also open a picture of Obama, and then use the camera to point at the opened picture. You can see that not only the face can be marked, but also the name of the detected face can be displayed correctly. **Note that when detecting a face, the image displayed by the camera will be relatively slow, please do not move too fast**



10) **web_service_example.py** is a very simple case of using a web service to upload a picture to run face recognition. The backend server will identify whether the picture is Obama, and output the recognition result as a json key-value pair. The test steps are as

follows：

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command (**if face_recognition is automatically installed using a script, then there is no need to install flask**)

```
orangepi@orangepi:~$ python3 -m pip install flask
orangepi@orangepi:~$ cd face_recognition/examples
root@orangepi:~/face_recognition/examples$ python3 web_service_example.py
 * Serving Flask app 'web_service_example' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses (0.0.0.0)
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://127.0.0.1:5001
 * Running on http://192.168.1.79:5001 (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 500-161-390
```

b. Then open another terminal and run the following command to return the result of image recognition (note that the execution path of the following command is **face_recognition/examples**)

```
orangepi@orangepi:~/face_recognition/examples$ curl -XPOST -F \
"file=@obama2.jpg" http://127.0.0.1:5001
{
  "face_found_in_image": true,
  "is_picture_of_obama": true
}
```

c. We can also copy the picture **face_recognition/examples/obama2.jpg** to other Linux computers. Of course, we can also prepare a picture named **obama2.jpg** by ourselves, and then use the following command on the Linux computer to remotely Use the service running on the development board to recognize faces (**note that the IP address in the command needs to be replaced with the IP address of the development board, and the file name after file needs to be replaced with the name of the picture you want to test**).

```
test@test:~$ curl -XPOST -F "file=@obama2.jpg" http://192.168.1.79:5001
{
  "face_found_in_image": true,
  "is_picture_of_obama": true
}
```

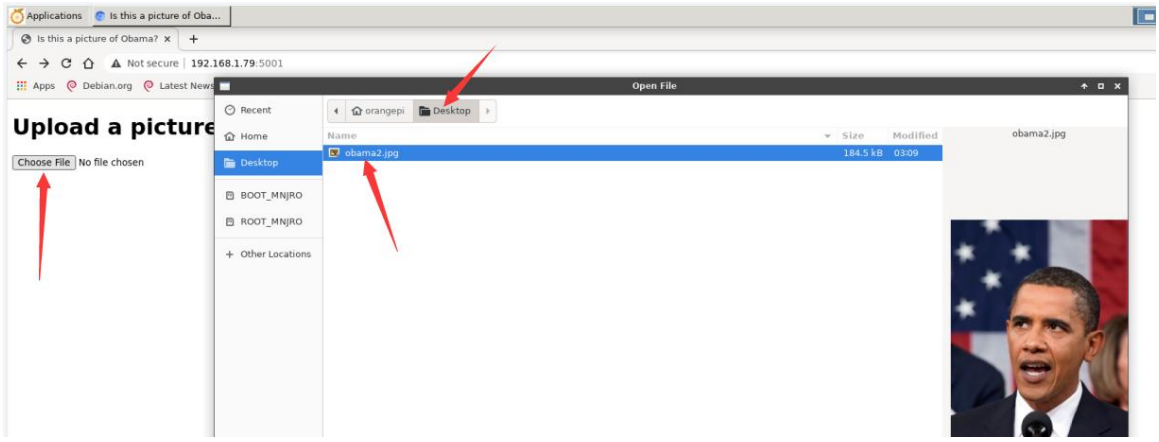
d. The method to test using the browser is as follows：

a) First open the browser, then enter the **IP address of the development board: 5001** in the address bar of the browser, and then you can see the following page

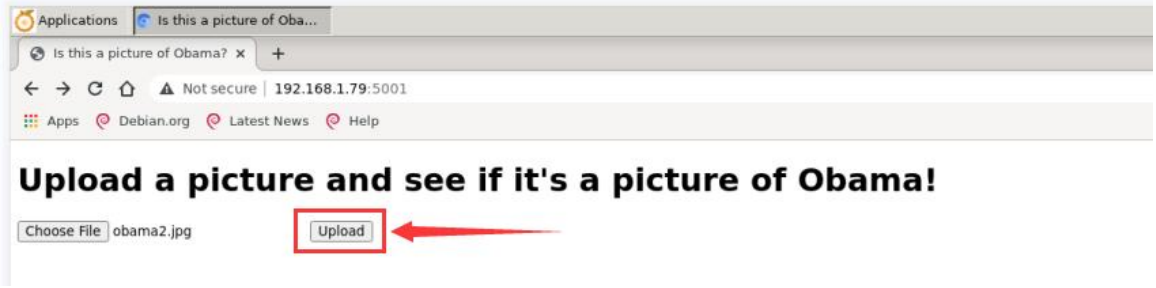
b) Then copy obama2.jpg to the desktop

```
orangepi@orangepi:~/face_recognition/examples$ cp obama2.jpg /home/orangepi/Desktop/
```

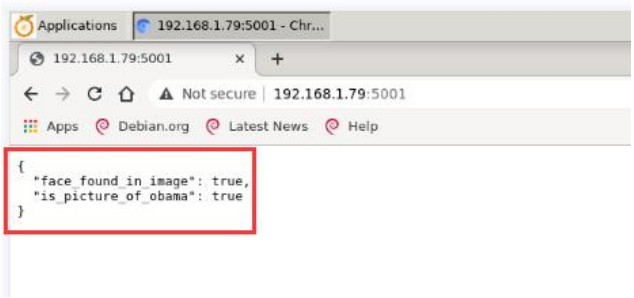
c) Then select the picture you just copied in your browser



d) Then click **Upload** to upload the picture you just selected for face recognition



e) After waiting for a while, the detection result will be displayed



11) **face_detection** command test example

a. The face_detection command-line tool can locate the face position (output pixel coordinates) in a single picture or a picture folder. Use **face_detection --help** to view the help information of the face_detection command

```
orangepi@orangepi:~$ face_detection --help
Usage: face_detection [OPTIONS] IMAGE_TO_CHECK

Options:
  --cpus INTEGER    number of CPU cores to use in parallel. -1 means "use all in
                    system"
  --model TEXT      Which face detection model to use. Options are "hog" or
                    "cnn".
  --help            Show this message and exit.
```

b. An example of detecting a single image is shown below：

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ face_detection obama2.jpg
obama2.jpg,302,474,611,164
```

c. An example of using multiple cores to detect multiple images in parallel is shown below：

a) First enter the **face_recognition/examples** folder

b) Then create a new test folder
c) Then copy the jpg images to the test folder
d) Then use all cpus to run **face_detection** in parallel to check the pictures in the test folder, where **--cpus -1** means to use all cpus

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ mkdir test
orangepi@orangepi:~/face_recognition/examples$ cp *.jpg test
orangepi@orangepi:~/face_recognition/examples$ face_detection --cpus -1 test
test/obama-240p.jpg,29,261,101,189
test/obama_small.jpg,65,215,169,112
test/obama2.jpg,302,474,611,164
test/two_people.jpg,62,394,211,244
test/two_people.jpg,95,941,244,792
test/obama.jpg,136,624,394,366
test/obama-480p.jpg,65,507,189,383
test/obama-720p.jpg,94,751,273,572
test/obama-1080p.jpg,136,1140,394,882
test/biden.jpg,233,749,542,439
```

12) **face_recognition** command test example
a. **face_recognition** command-line tool can recognize whose face is in a single picture or a picture folder. Use **face_recognition --help** to view the help information of the face_recognition command

```
orangepi@orangepi:~$ face_recognition --help
Usage: face_recognition [OPTIONS] KNOWN_PEOPLE_FOLDER
IMAGE_TO_CHECK

Options:
  --cpus INTEGER                  number of CPU cores to use in parallel (can speed
                                  up processing lots of images). -1 means "use all in
                                  system"
  --tolerance FLOAT               Tolerance for face comparisons. Default is 0.6.
                                  Lower this if you get multiple matches for the same
                                  person.
  --show-distance BOOLEAN         Output face distance. Useful for tweaking tolerance
                                  setting.
```

--help	Show this message and exit.

b. First create a new face picture folder **known_people** with a known name, then copy two pictures to **known_people**, and then copy **obama2.jpg** to **unknown.jpg**, which is the picture we want to identify

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ mkdir known_people
orangepi@orangepi:~/face_recognition/examples$ cp biden.jpg obama.jpg known_people
orangepi@orangepi:~/face_recognition/examples$ cp obama2.jpg unkown.jpg
```

c. Then you can use the following command to identify the name of the person in the **unknown.jpg** picture, and you can see that the unknown.jpg picture is recognized as obama

```
orangepi@orangepi:~/face_recognition/examples$ face_recognition known_people \
unkown.jpg
unkown.jpg,obama
```

d. If we identify an irrelevant image, unknown_person will be displayed

```
root@orangepi:~/face_recognition/examples$ face_recognition known_people \
alex-lacamoire.png
alex-lacamoire.png,unknown_person
```

e. We can also create a new test folder and put multiple pictures in it, and then we can use all the CPUs to recognize all the pictures in parallel

```
orangepi@orangepi:~/face_recognition/examples$ mkdir test
orangepi@orangepi:~/face_recognition/examples$ cp *.jpg *.png test
orangepi@orangepi:~/face_recognition/examples$ face_recognition --cpus -1 \
known_people test
test/obama-240p.jpg,obama
test/alex-lacamoire.png,unknown_person
test/obama_small.jpg,obama
test/unkown.jpg,obama
test/obama2.jpg,obama
test/lin-manuel-miranda.png,unknown_person
test/two_people.jpg,biden
test/two_people.jpg,obama
test/obama-720p.jpg,obama
test/obama.jpg,obama
test/obama-480p.jpg,obama
```

```
test/biden.jpg,biden
test/obama-1080p.jpg,obama
```

3.28. Setting Chinese environment and installing Chinese input method

Note, please make sure that the Linux system used by the development board is a desktop system before installing the Chinese input method。

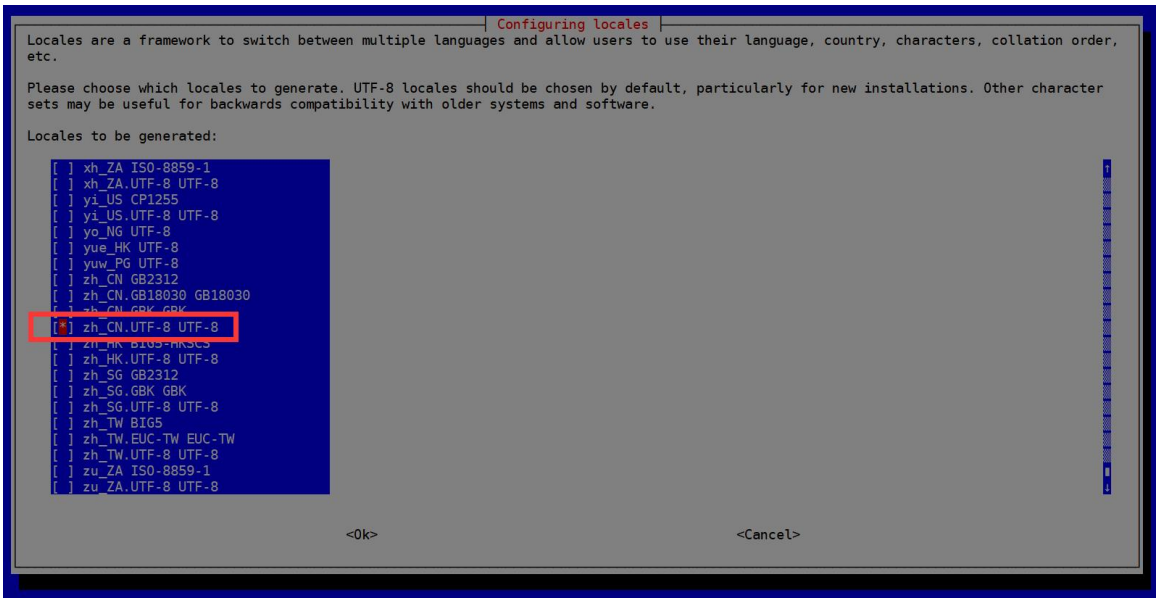
3.28.1. How to install Debian system

1) First set the default **locale** to Chinese

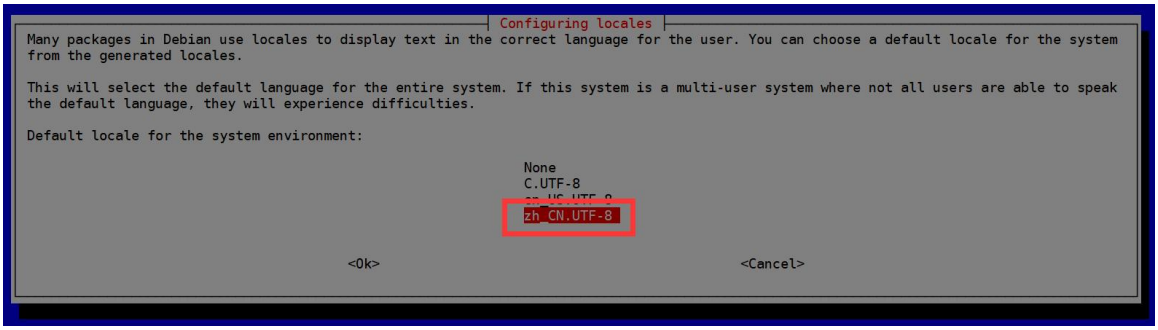
a. Enter the following command to start configuring the **locale**

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
```

b. Then select **zh_CN.UTF-8 UTF-8** in the pop-up interface (move up and down through the up and down direction keys on the keyboard, select through the space bar, and finally move the cursor to **<OK>**through the Tab key, and then press Enter.)



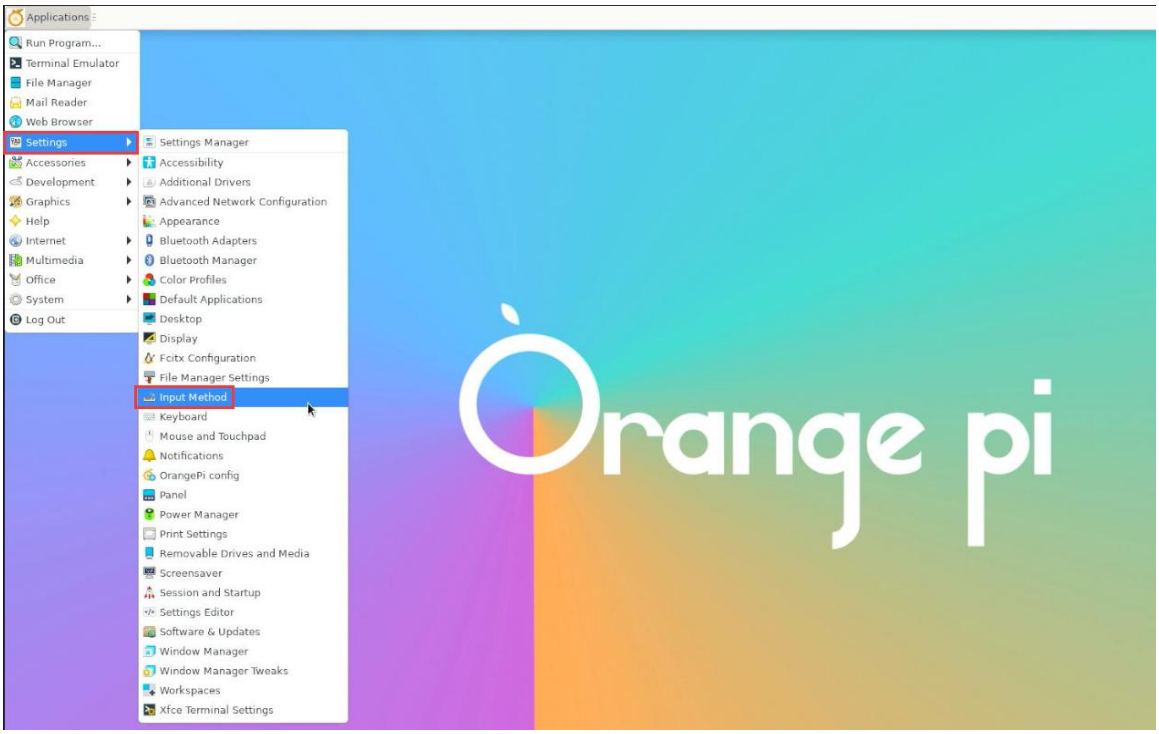
c. Then set the default **locale** to **zh_CN.UTF-8**



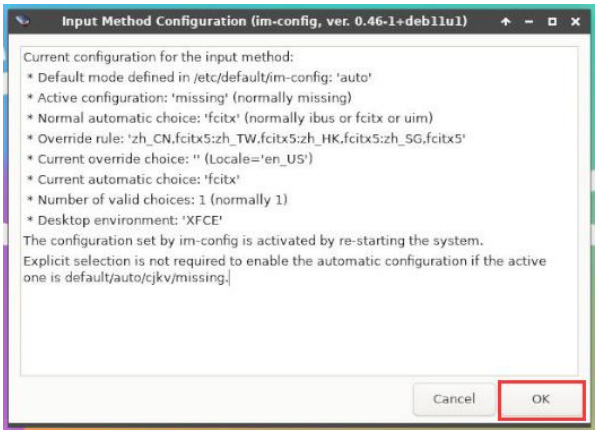
d. After exiting the interface, the **locale** setting will start, and the output displayed on the command line is as follows

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
  en_US.UTF-8... done
  zh_CN.UTF-8... done
Generation complete.
```

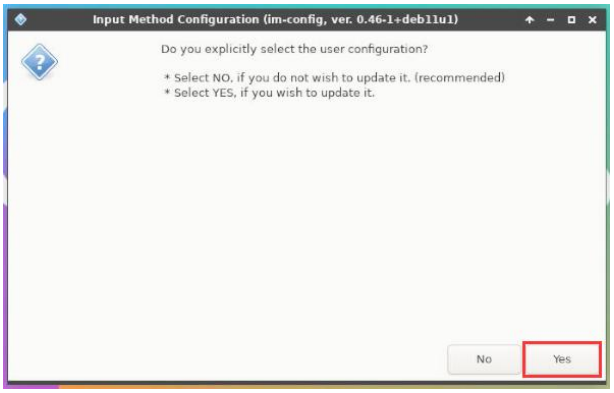
2) Then open the **Input Method**



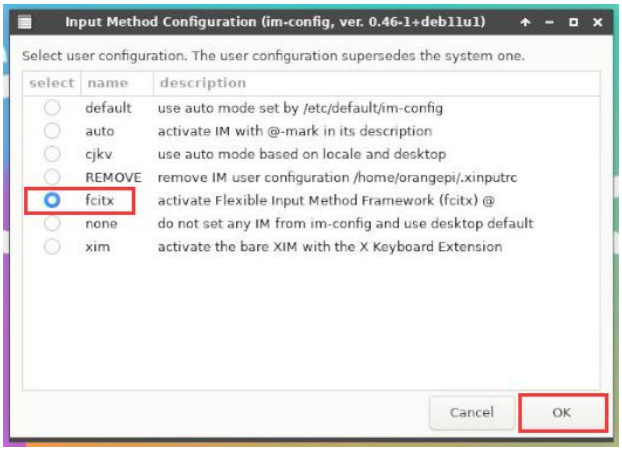
3) Then choose **OK**



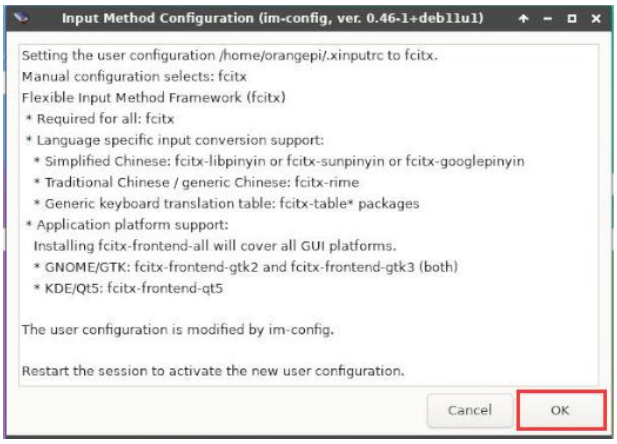
4) Then choose**Yes**



5) Then choose**fcitx**

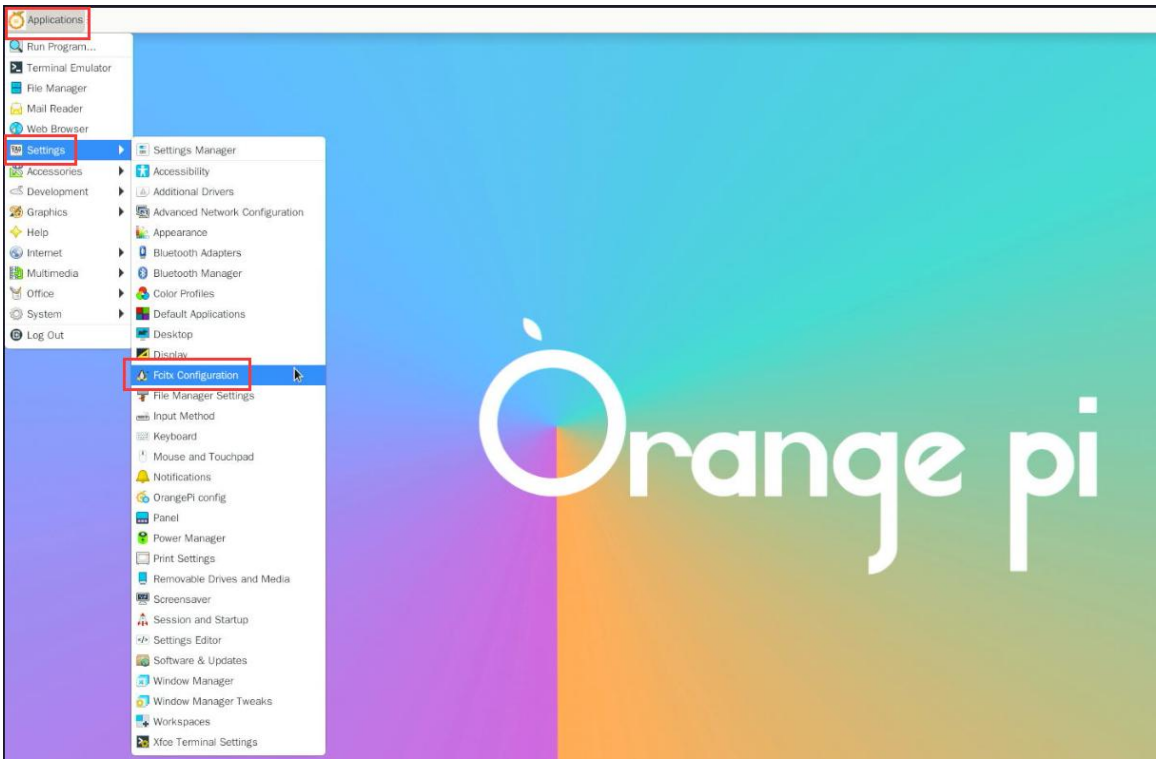


6) Then choose**OK**

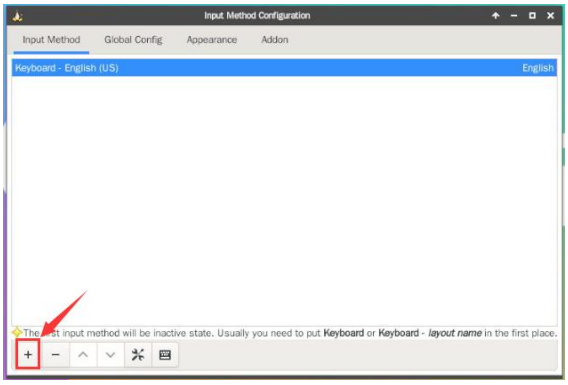


7) **Then restart the Linux system to make the configuration take effect**

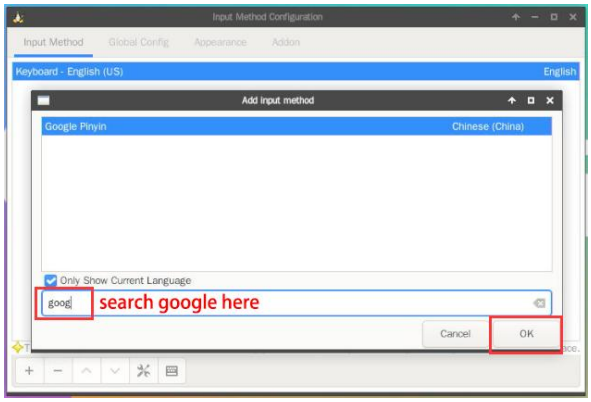
8) Then open **Fcitx configuration**



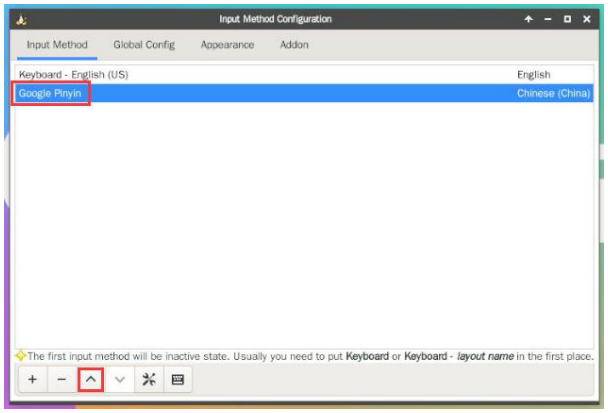
9) Then click the + sign in the position shown in the figure below

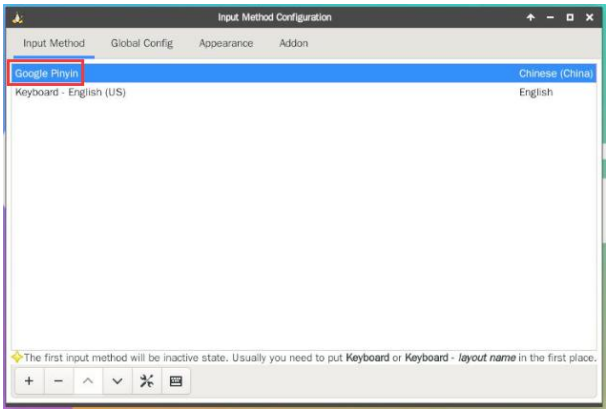


10) Then search **Google Pinyin** and click **OK**



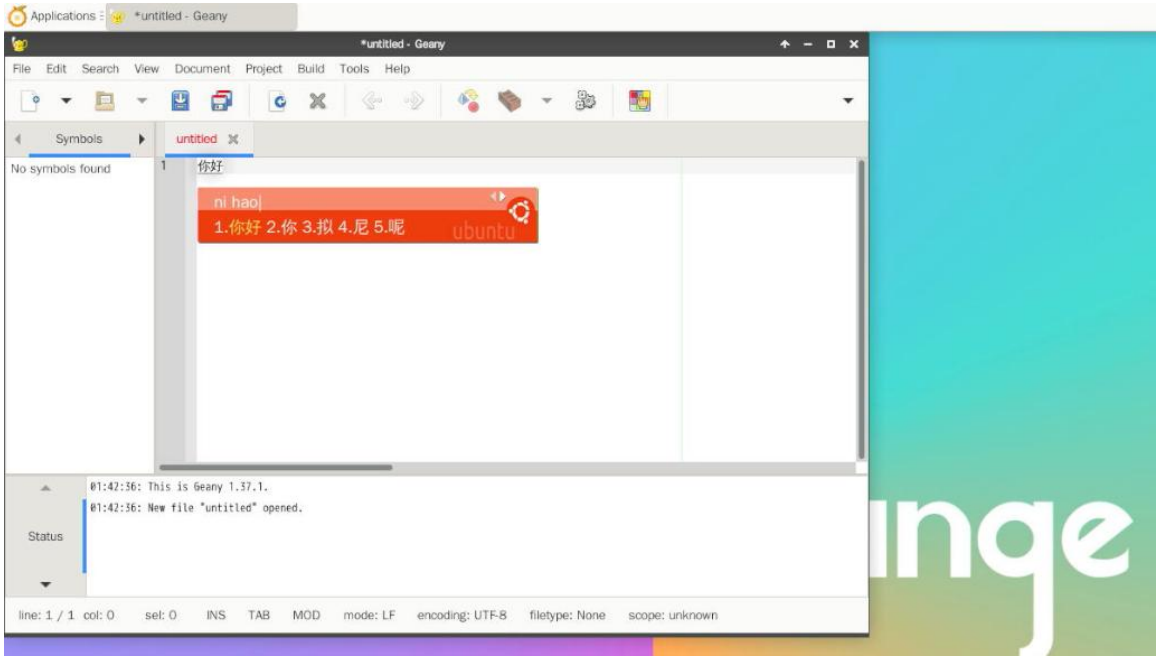
11) Then bring **Google Pinyin** to the front





12) Then open the **Geany** editor to test the Chinese input method

13) The Chinese input method test is as follows

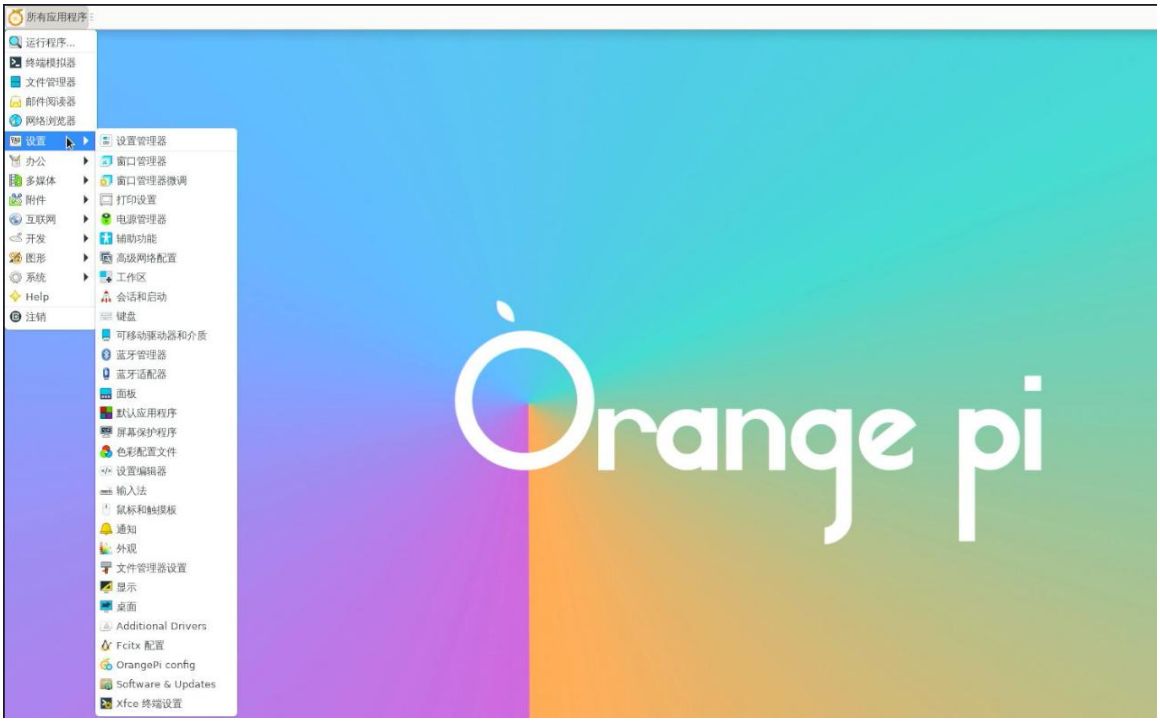


14) The Chinese and English input methods can be switched through the **Ctrl+Space** shortcut key

15) If the entire system needs to be displayed in Chinese, you can set the variables in **/etc/default/locale** to **zh_CN.UTF-8**

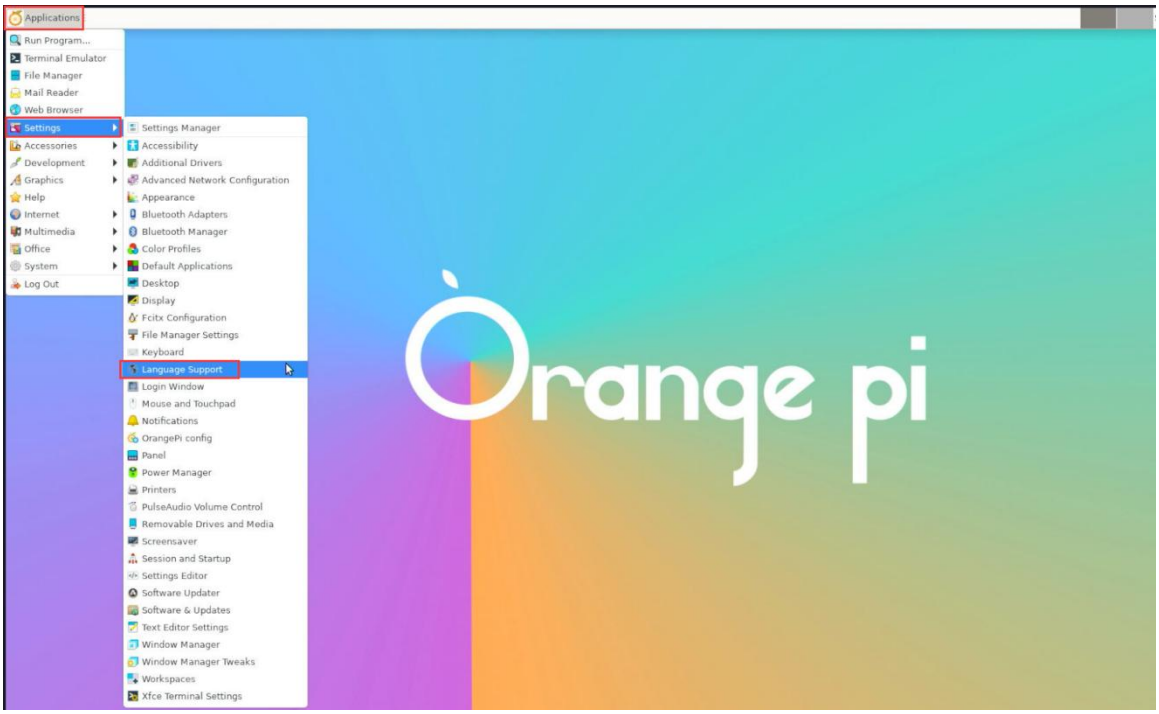
```
orangepi@orangepi:~$ sudo vim /etc/default/locale
#  File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** and you can see that the system is displayed in Chinese

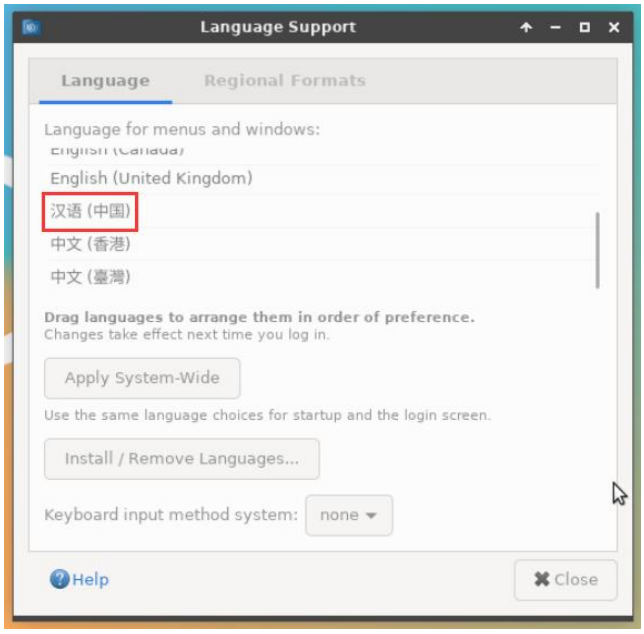


3. 28. 2. How to install Ubuntu 20.04 system

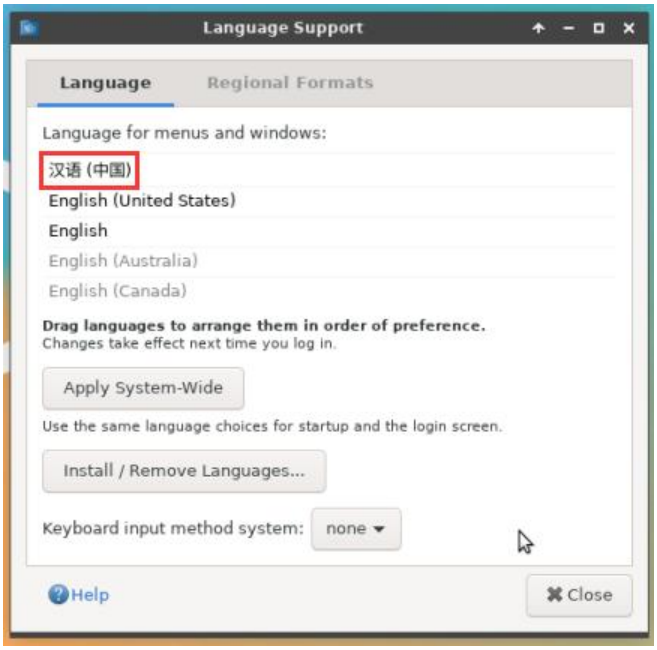
1) First open **Language Support**



2) Then find the **Chinese (China)** option

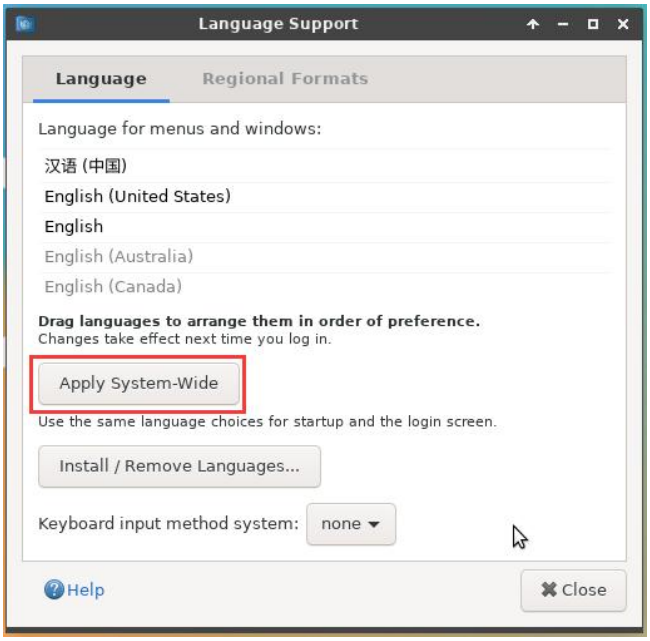


3) Then please use the left mouse button to select **Chinese (China)** and hold it down, then drag it up to the initial position, the display after dragging is as shown in the figure below：



Note that this step is not easy to drag, please be patient and try a few more times.

4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system

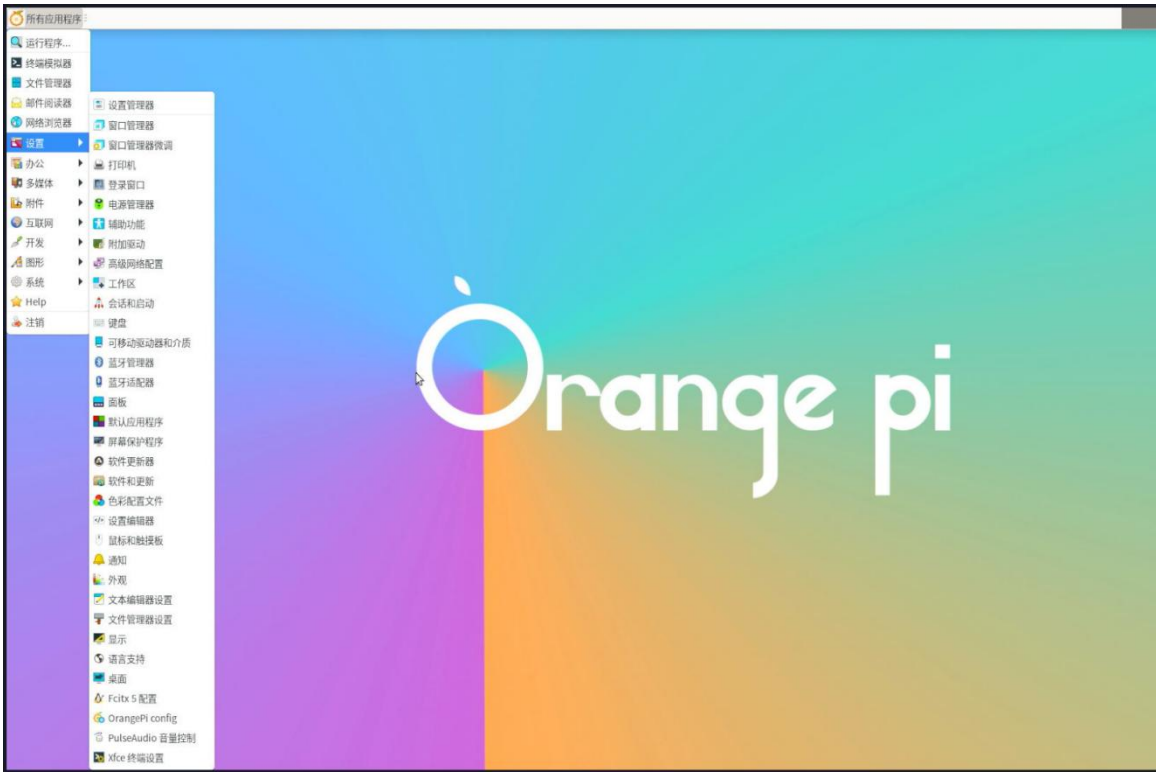


5) Then set**Keyboard input method system** to **fcitx**

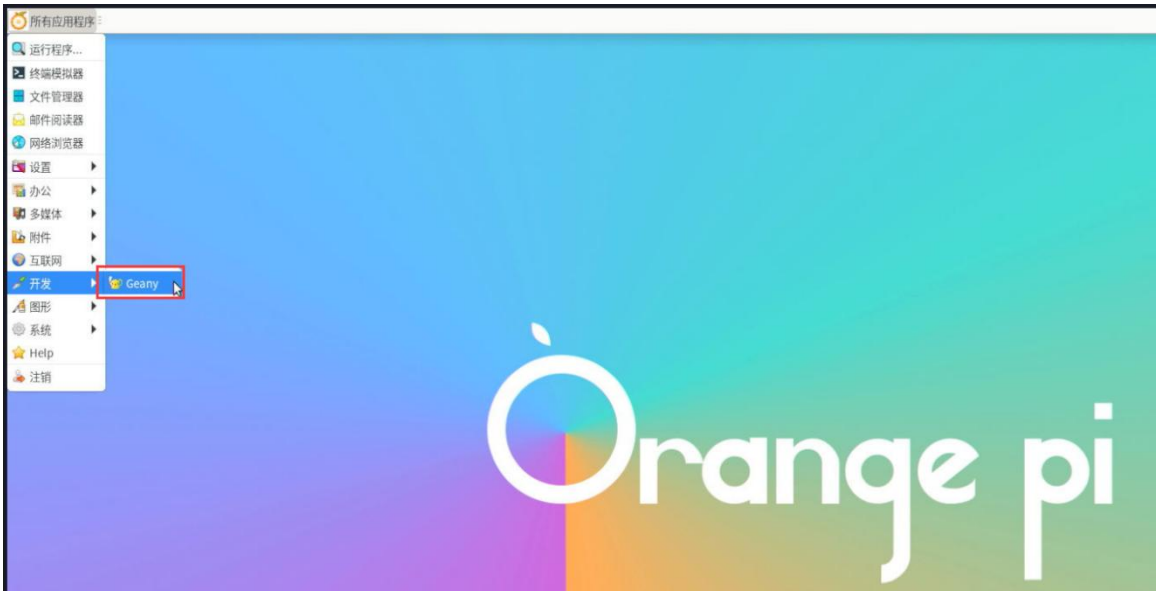
6) **Then restart the Linux system to make the configuration take effect**

7) After re-entering the system, **please choose not to ask me again** in the following interface, and then please decide whether the standard folder should also be updated to Chinese according to your preferences

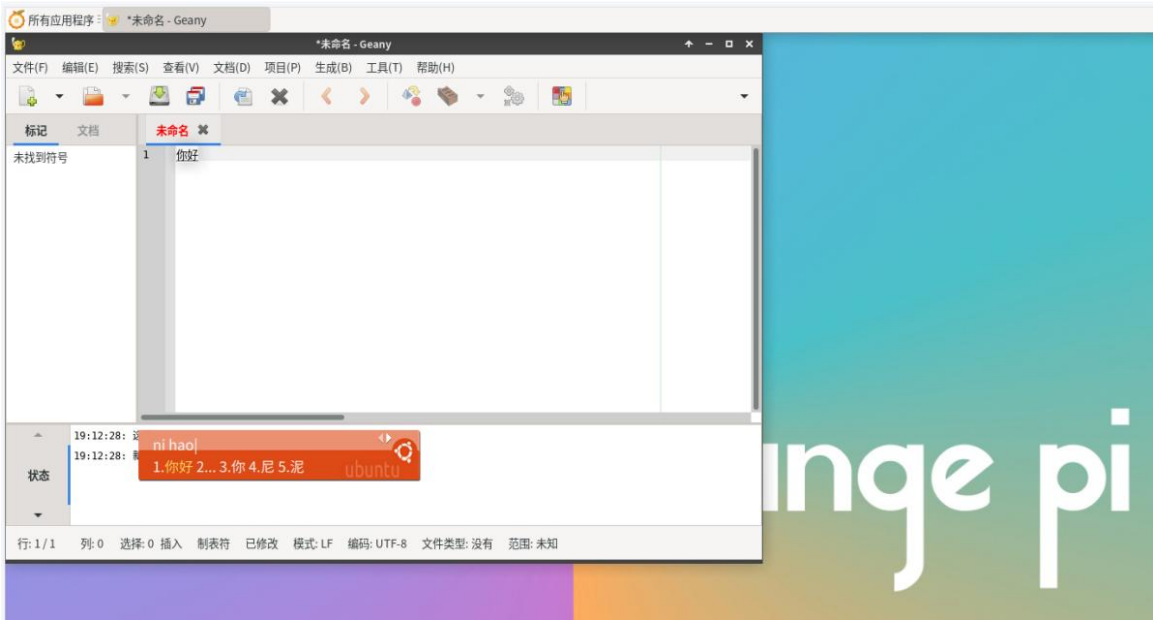
8) Then you can see that the desktop is displayed in Chinese



9) Then we can open **Geany** to test the Chinese input method, as shown in the figure below

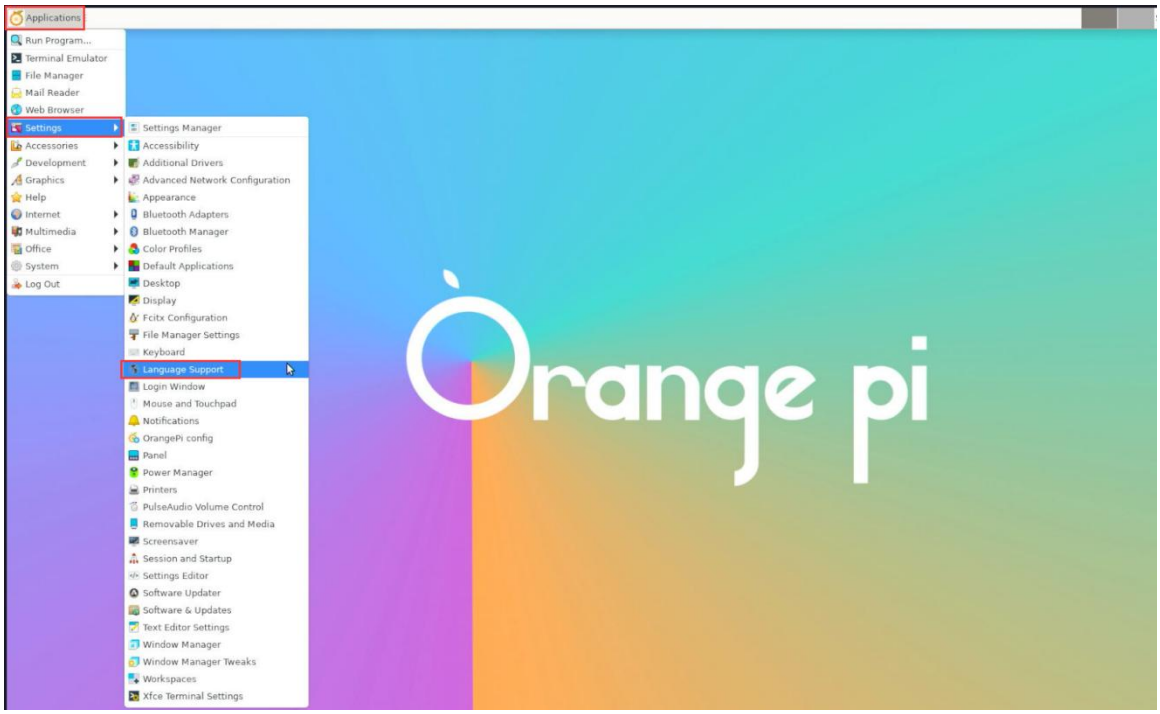


10) After opening **Geany**, the default is English input method, we can switch to Chinese input method through **Ctrl+Space** shortcut key, and then we can input Chinese



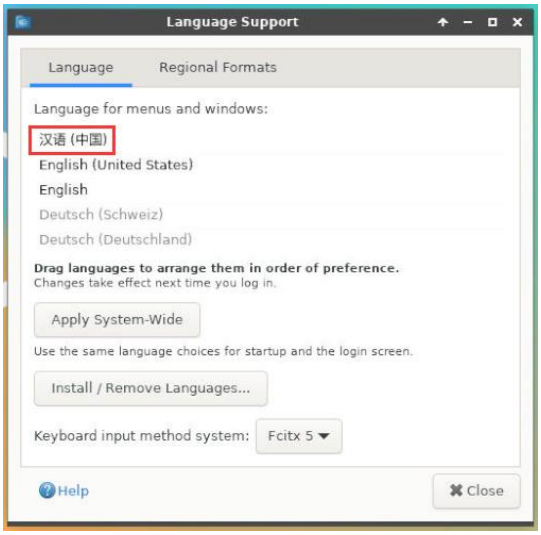
3. 28. 3. How to install Ubuntu 22.04 system

1) First open **Language Support**



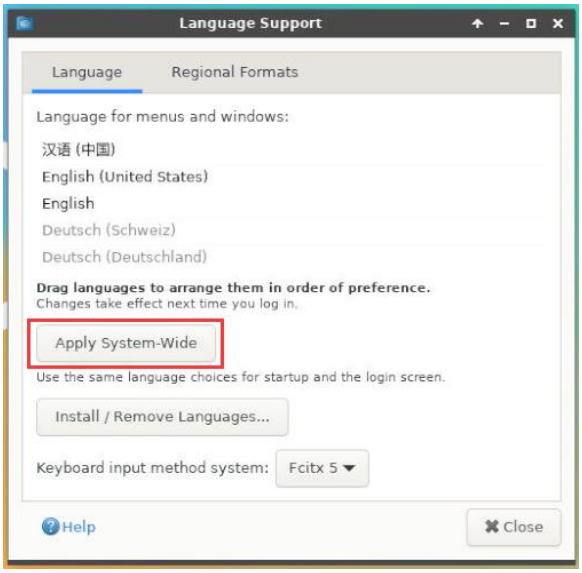
2) Then find the **Chinese (China)** option

3) Then please use the left mouse button to select **Chinese (China)** and hold it down, then drag it up to the initial position, the display after dragging is as shown in the figure below：



Note that this step is not easy to drag, please be patient and try a few more times。

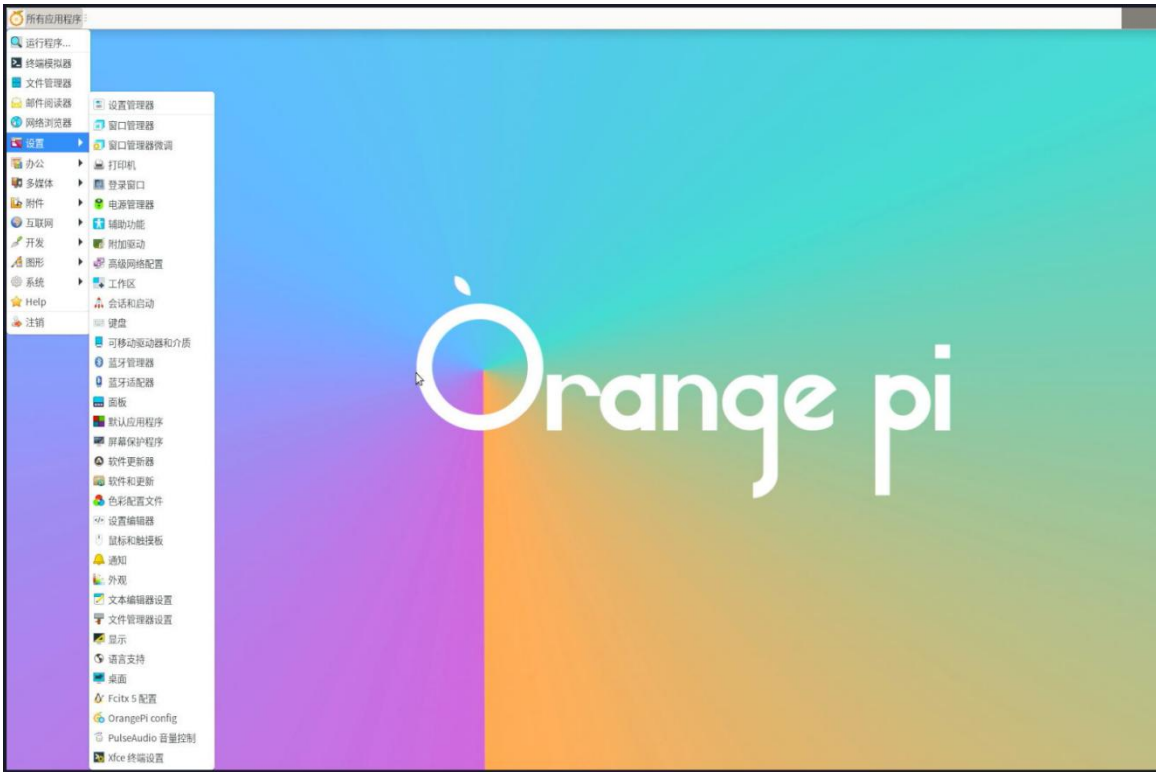
4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



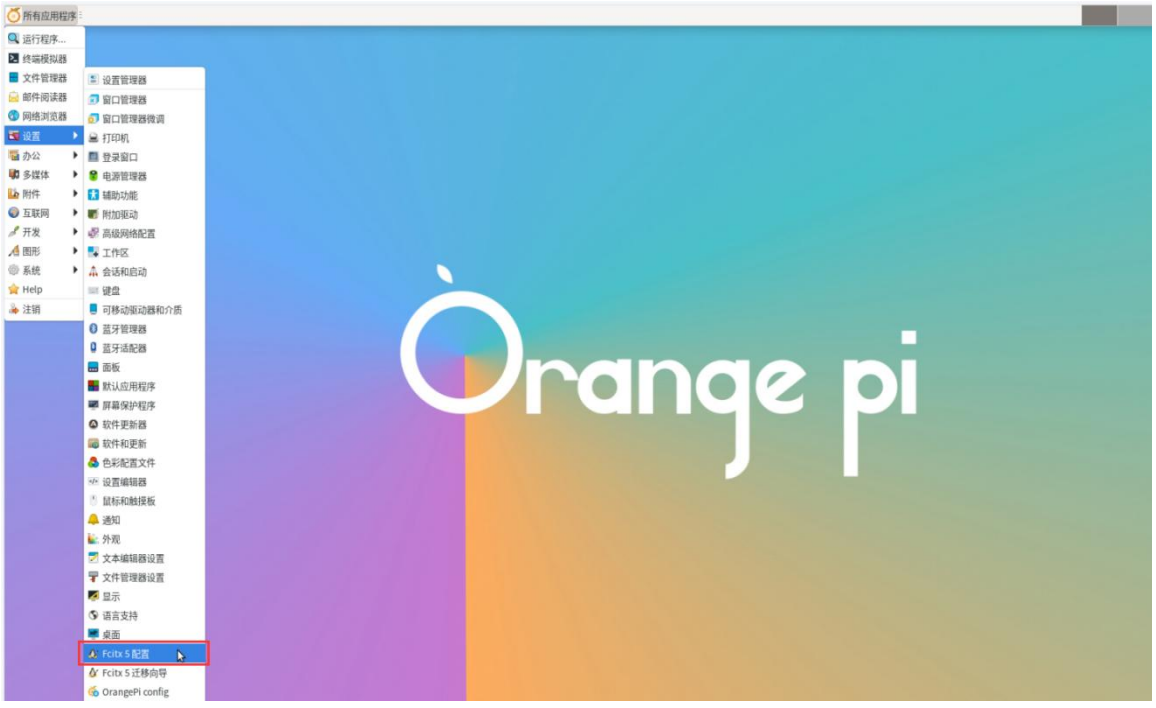
5) **Then restart the Linux system to make the configuration take effect**

6) After re-entering the system, please choose **not to ask me again** in the following interface, and then please decide whether the standard folder should also be updated to Chinese according to your preferences

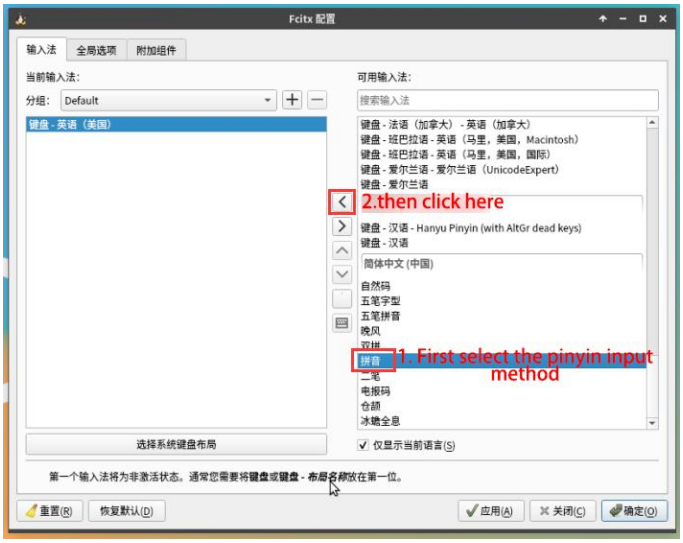
7) Then you can see that the desktop is displayed in Chinese



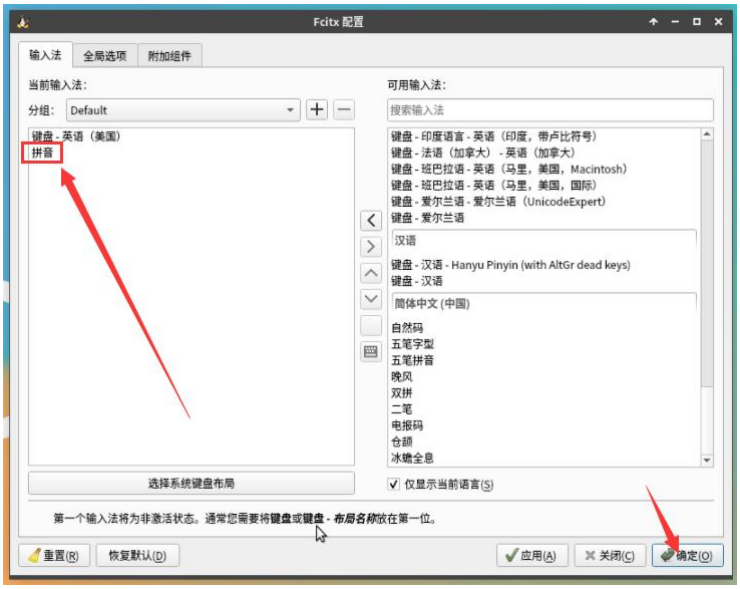
8) Then open the Fcitx5 configuration program



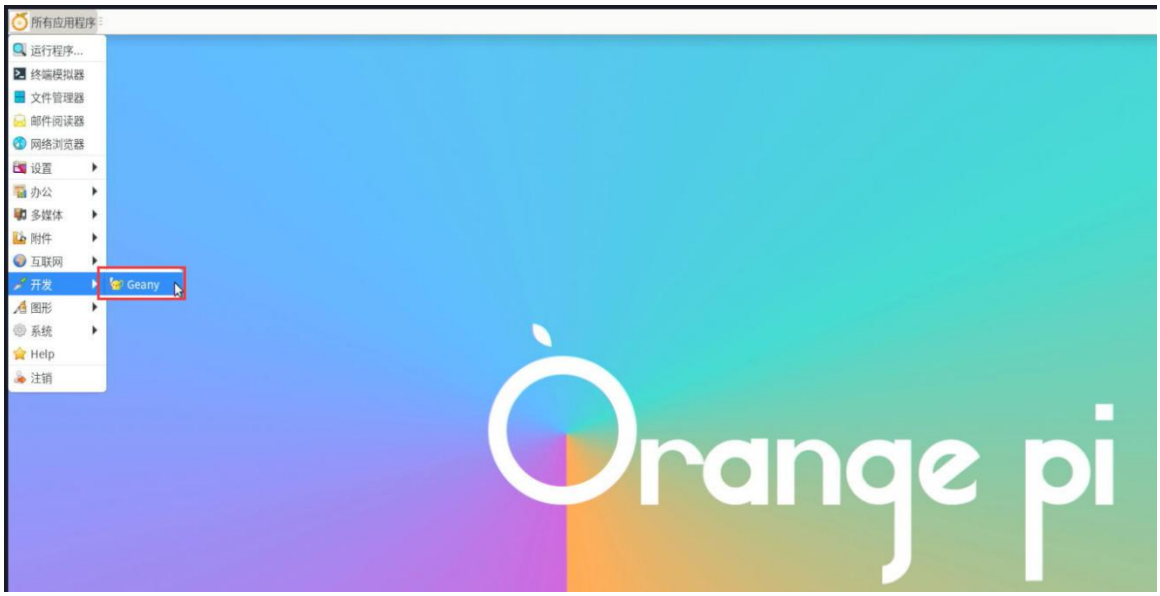
9) Then choose to use Pinyin input method



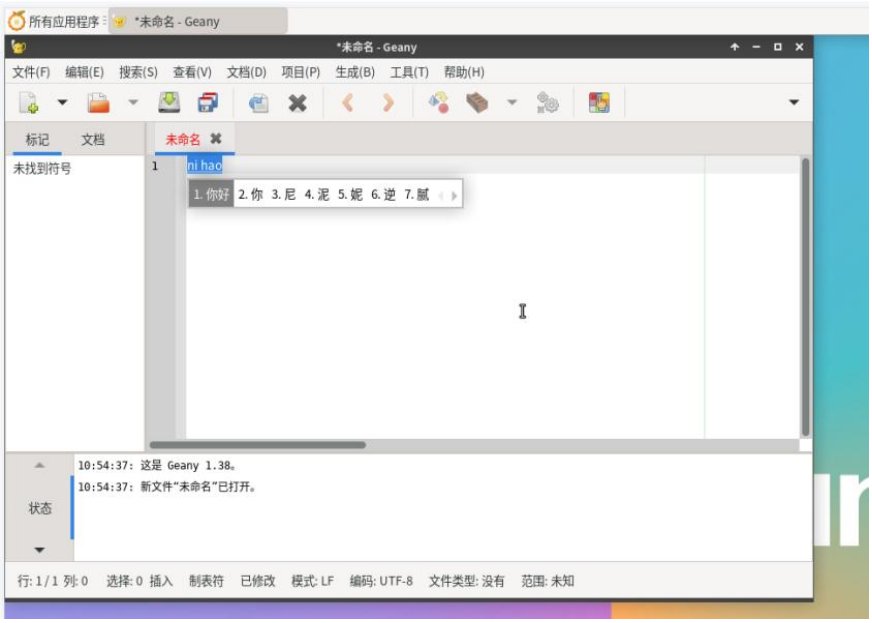
10) The interface after selection is as shown below, and then click OK



11) Then we can open **Geany** to test the Chinese input method, the opening method is shown in the figure below



12) After opening **Geany**, the default is English input method, we can switch to Chinese input method through **Ctrl+Space** shortcut key, and then we can input Chinese



3. 29. How to Remotely Log In to the Desktop of Linux System

3. 29. 1. Remote login using NoMachine

Make sure the Ubuntu or Debian system installed on the development board is a desktop system version. In addition, NoMachine also provides detailed documents.

It is strongly recommended to read this document to be familiar with the use of NoMachine. The document link is shown below：

https://knowledgebase.nomachine.com/DT10R00166

NoMachine supports Windows, Mac, Linux, iOS and Android platforms, so we can remotely log in and control the Orange Pi development board through NoMachine on a variety of devices. The following demonstrates how to remotely log in to the Linux system desktop of the Orange Pi development board through NoMachine in Windows. For installation methods on other platforms, please refer to the official documentation of NoMachine

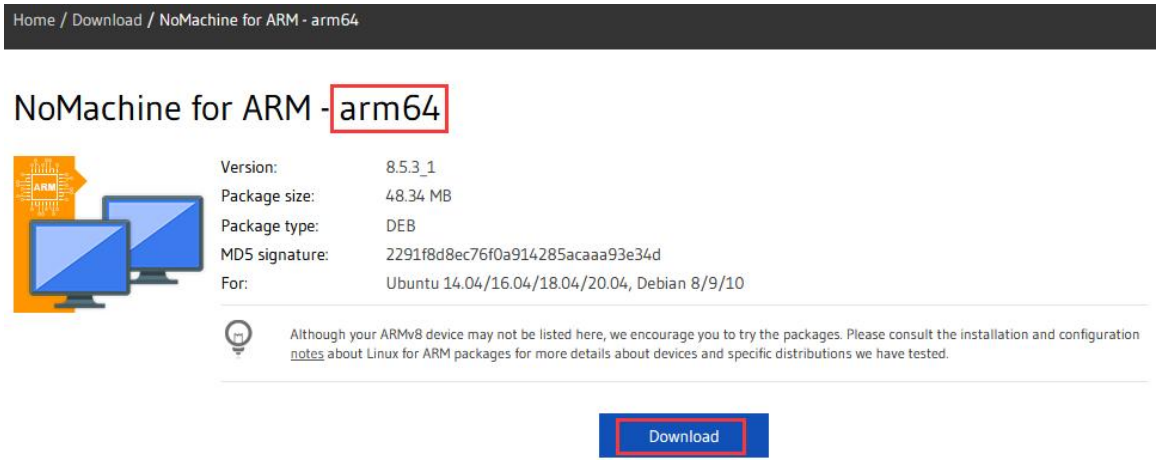
Before operation, please make sure that the Windows computer and the development board are in the same LAN, and can log in to the Ubuntu or Debian system of the development board through ssh

1) First download the installation package of the NoMachine software Linux **arm64** deb version, and then install it into the Linux system of the development board

a. Since H618 is a SOC of the ARMV8 architecture, the system we use is Ubuntu or Debian, so you need to download the **NoMachine for ARM ARMv8 DEB** installation package. The download link is shown below：

Note that this download link may change, please look for the deb package of the Armv8/Arm64 version。

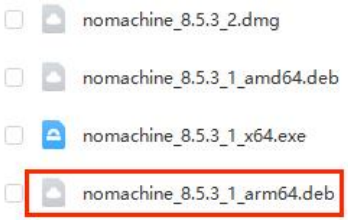
https://downloads.nomachine.com/download/?id=118&distro=ARM



b. In addition, the installation package of **NoMachine** can also be downloaded in the **official tool**

First enter the **remote login software-NoMachine** folder

Then download the arm64 version of the deb installation package



c. Then upload the downloaded **nomachine_x.x.x_x_arm64.deb** to the Linux system of the development board

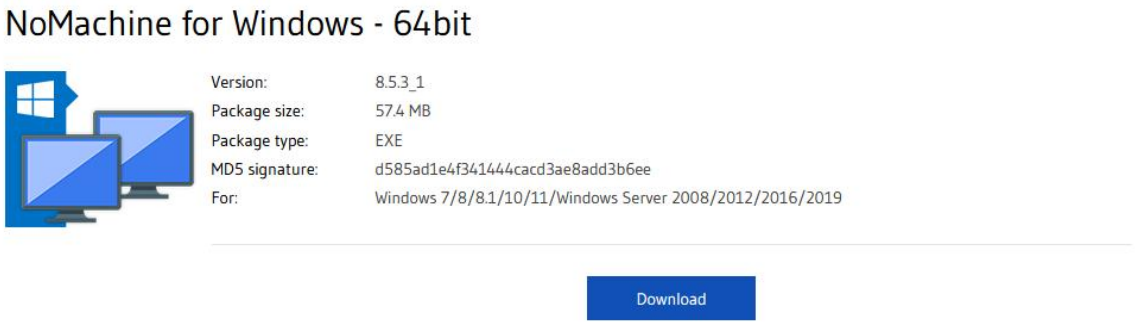
d. Then use the following command to install **NoMachine** in the Linux system of the development board

```
orangepi@orangepi:~$ sudo dpkg -i nomachine_x.x.x_x_arm64_arm64.deb
```

2) Then download the installation package of the Windows version of the NoMachine software, the download address is as follows

Note that this download link may change。

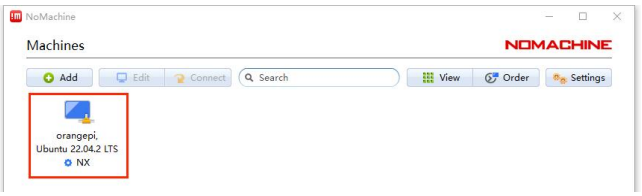
https://downloads.nomachine.com/download/?id=9



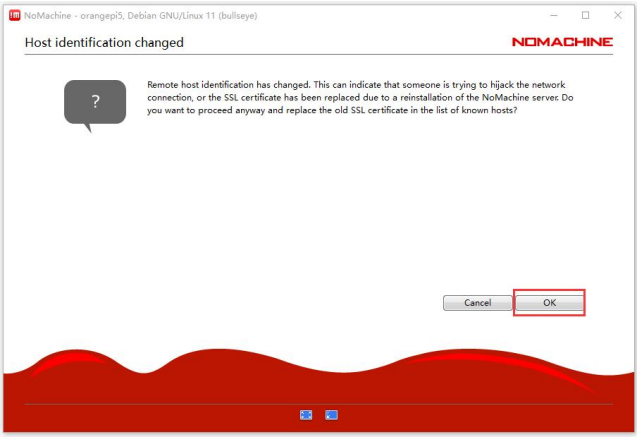
3) Then install NoMachine in Windows, **please restart the computer after installation**

4) Then open **NoMachine** in Window

5) After Nomachine starts, it will automatically scan other devices installed in the local area network. After entering the main interface of Nomachine, you can see that the development board is already in the connected device list, and then click the location shown in the red box below. Start log in to the linux system desktop of the development board



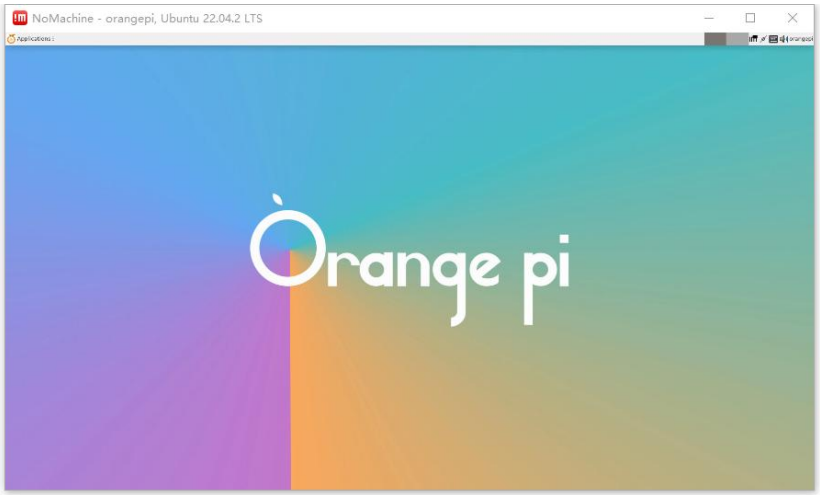
6) Then click **OK**



7) Then enter the username and password of the linux system in the corresponding position in the figure below, and then click **OK** to start logging in

8) Then click OK in the next interface

9) Finally, you can see the desktop of the development board Linux system



3. 29. 2. Log in remotely using VNC

Before operation, please make sure that the Windows computer and the development board are in the same local area network, and can log in to the Ubuntu or Debian system of the development board normally with ssh.

Ubuntu20.04 has many problems testing VNC, please do not use this method

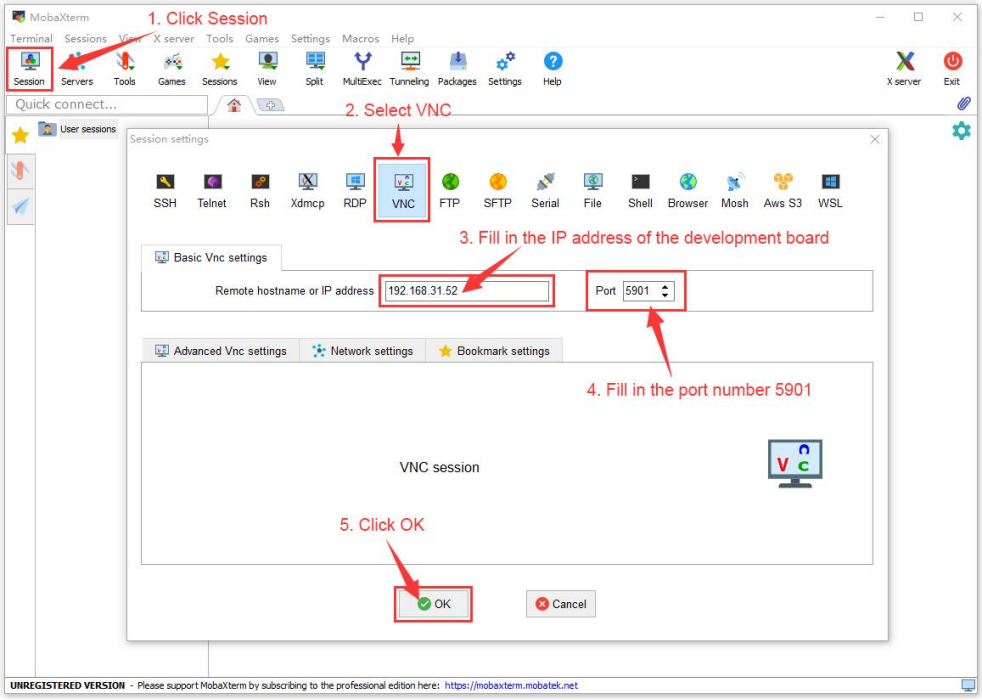
1) First run the **set_vnc.sh** script to set up vnc, remember to **add sudo permission**

```
orangepi@orangepi:~$ sudo set_vnc.sh
You will require a password to access your desktops.
```

```
Password:          #Set the vnc password here, 8 -bit characters
Verify:            #Set the vnc password here, 8 characters
Would you like to enter a view-only password (y/n)? n
xauth:    file /root/.Xauthority does not exist

New 'X' desktop is orangepi:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi:1.log

Killing Xtightvnc process ID 3047

New 'X' desktop is orangepi:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi:1.log
```

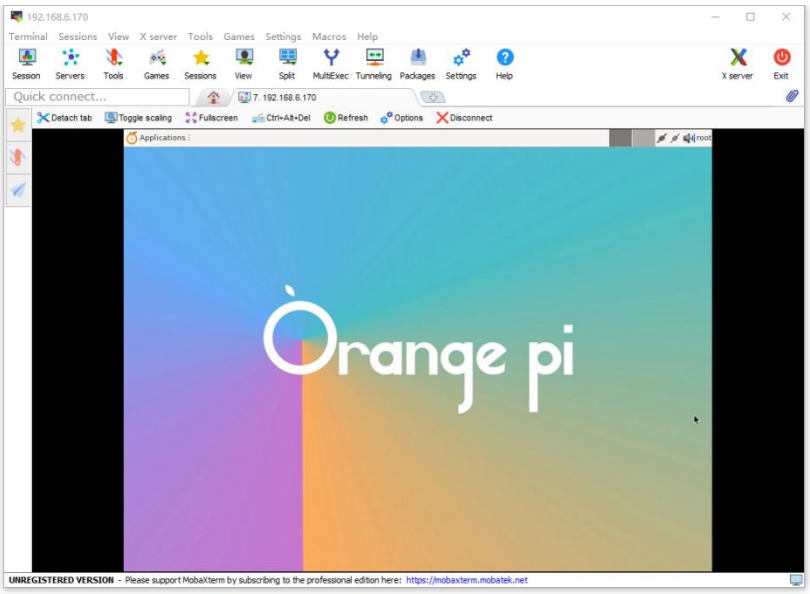
2) The steps to use the MobaXterm software to connect to the desktop of the Linux system of the development board are as follows：

a. First click on Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm



b. Then enter the VNC password set earlier

c. After successful login, the interface is displayed as shown in the figure below, and then you can remotely operate the desktop of the development board linux system



3. 30. How to install QT

1) Use the script below to install QT5 and QT Creator

```
orangepi@orangepi:~$ install_qt.sh
```

2) After installation, the version number of QT will be automatically printed

a. The qt version that comes with Ubuntu 20.04 is **5.12.8**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/aarch64-linux-gnu
```

b. The QT version that comes with Ubuntu 22.04 is **5.15.3**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.3 in /usr/lib/aarch64-linux-gnu
```

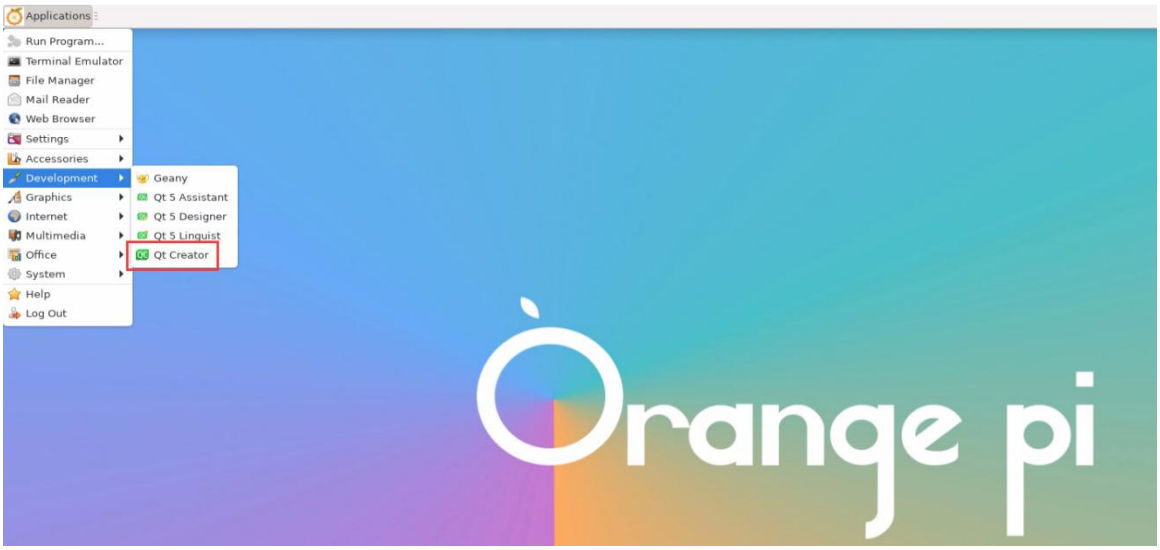
c. The QT version that comes with Debian11 is **5.15.2**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

d. The QT version that comes with Debian12 is **5.15.8**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.8 in /usr/lib/aarch64-linux-gnu
```

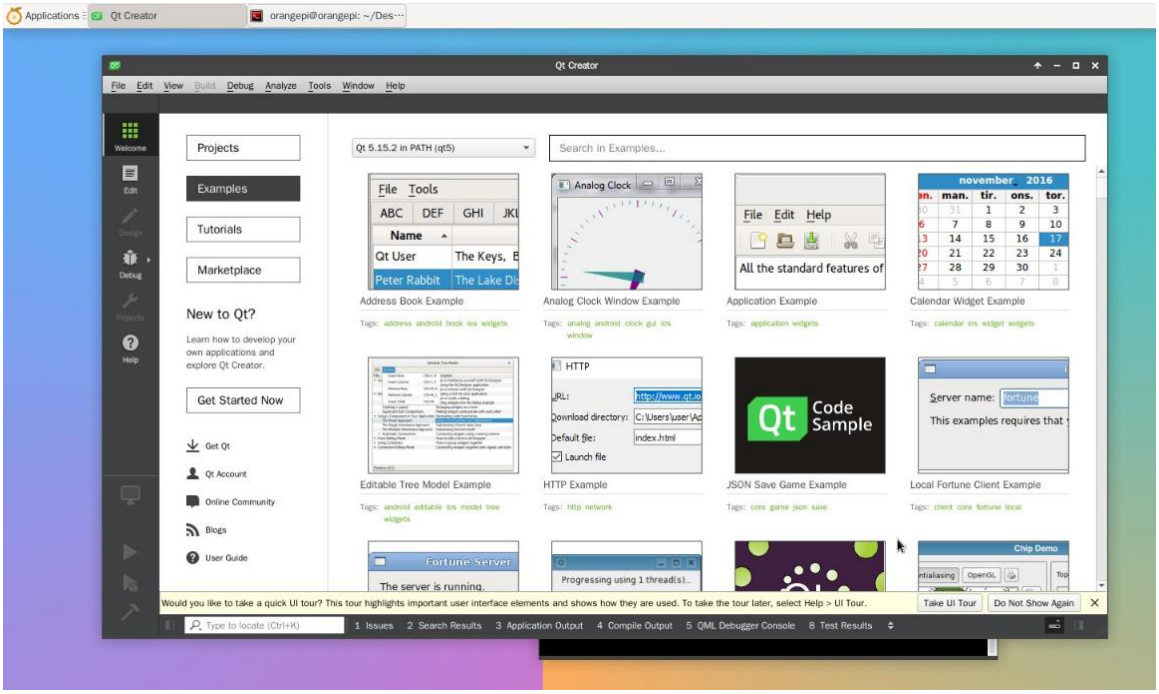
3) Then you can see the QT Creator startup icon in **Applications**



You can also use the following command to open QT Creator

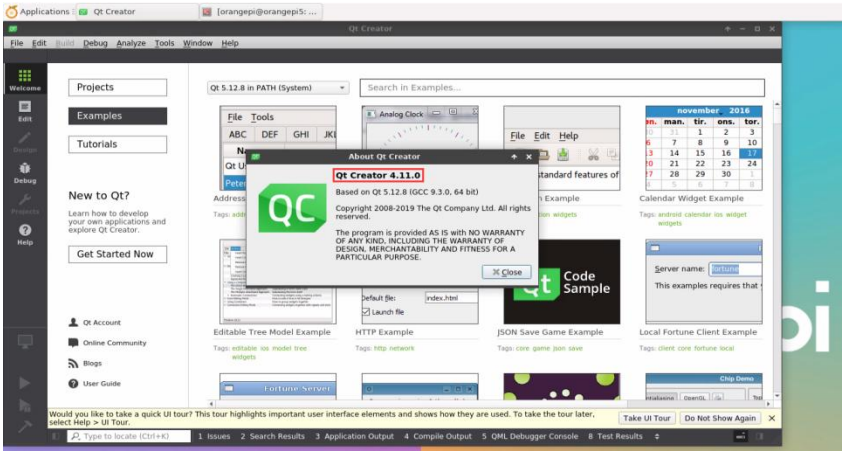
```
orangepi@orangepi:~$ qtcreator
```

4) The interface after QT Creator is opened is as follows

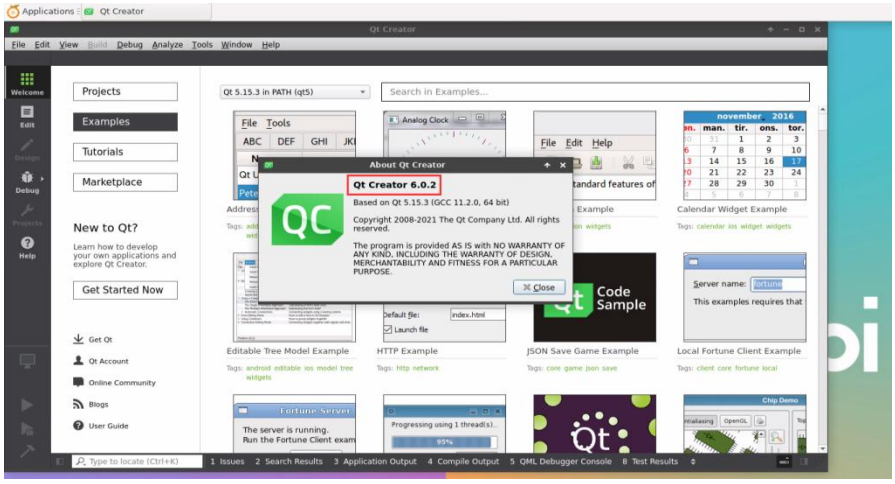


5) The version of QT Creator is as follows

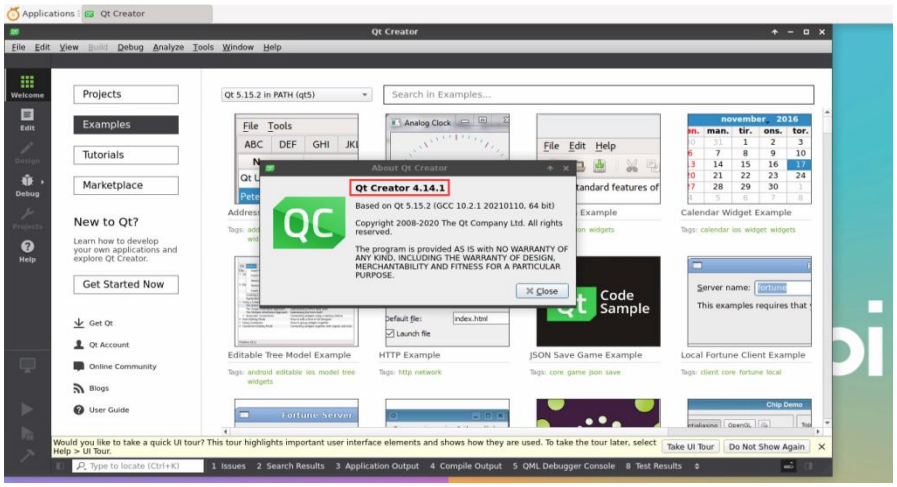
a. The default version of QT Creator in **Ubuntu20.04** is as follows



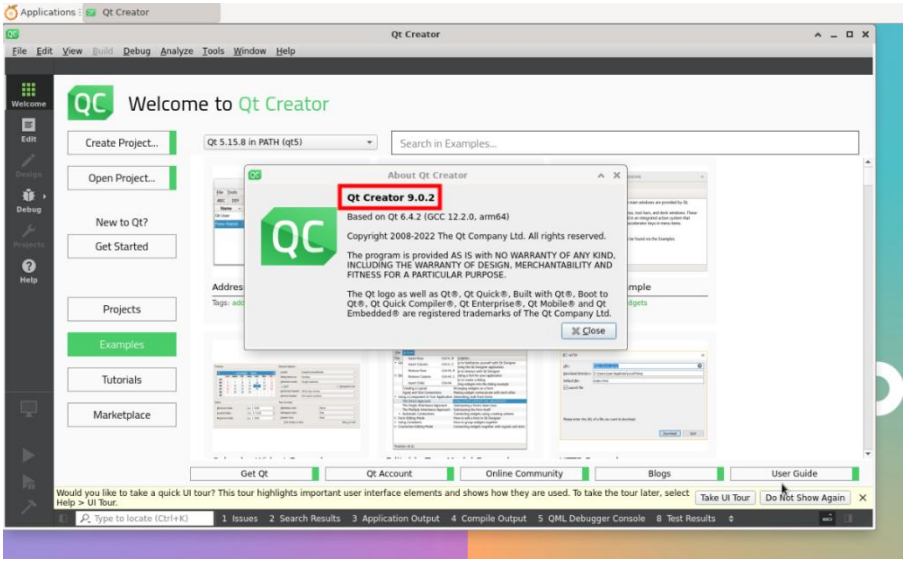
b. The default version of QT Creator in **Ubuntu22.04** is as follows



c. The default version of QT Creator in **Debian11** is as follows

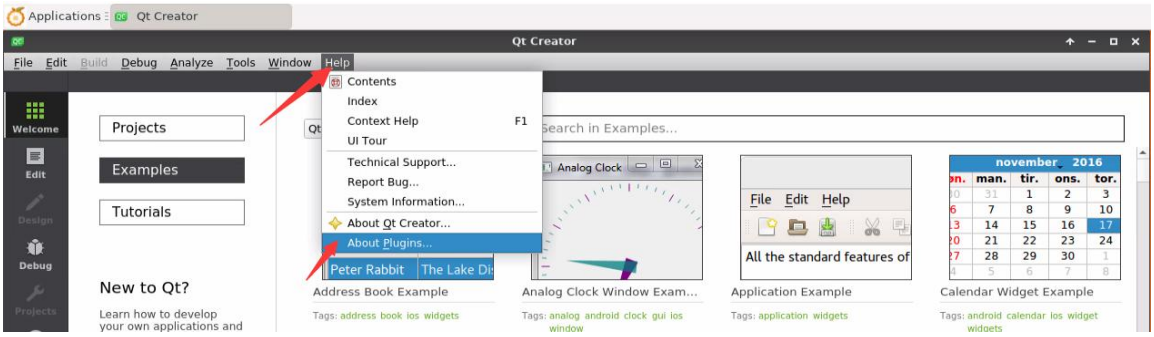


d. The default version of QT Creator in **Debian12** is as follows

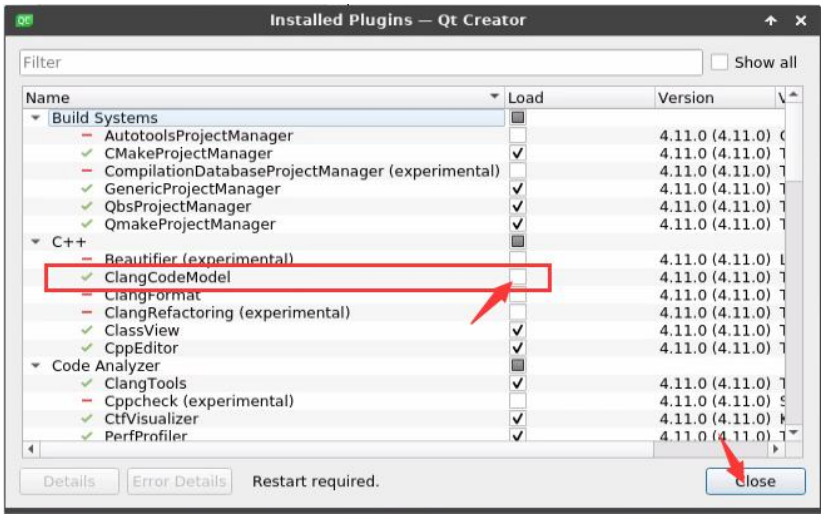


6) Then set up QT

a. First open **Help->About Plugins....**



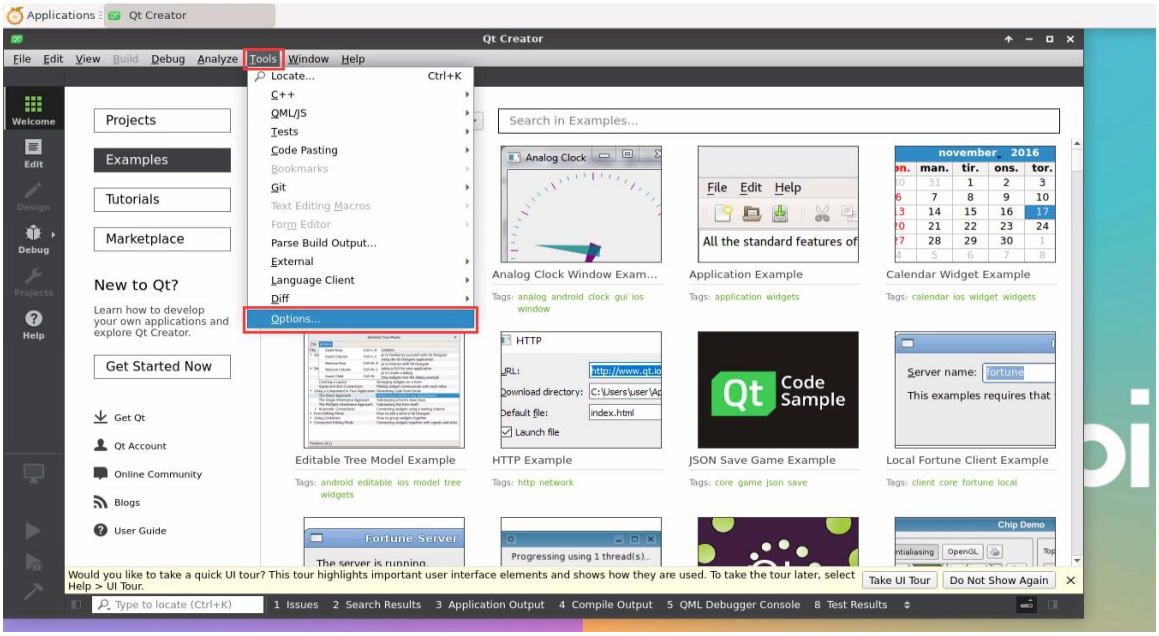
b. Then remove the tick of **ClangCodeModel**

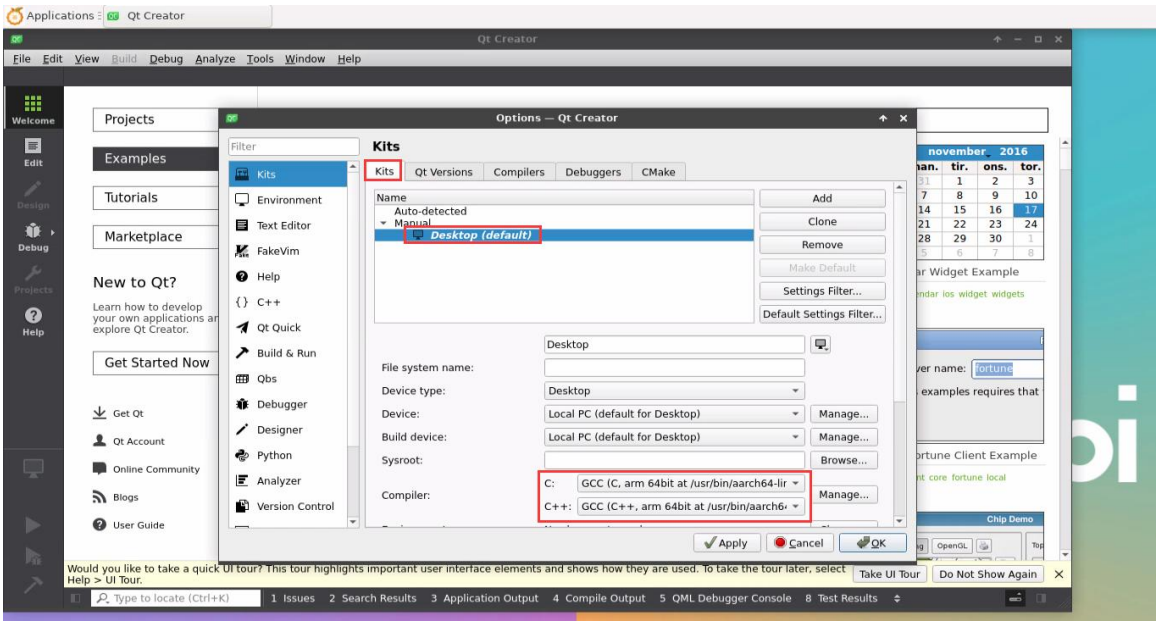


c. **After setting, you need to restart QT Creator**

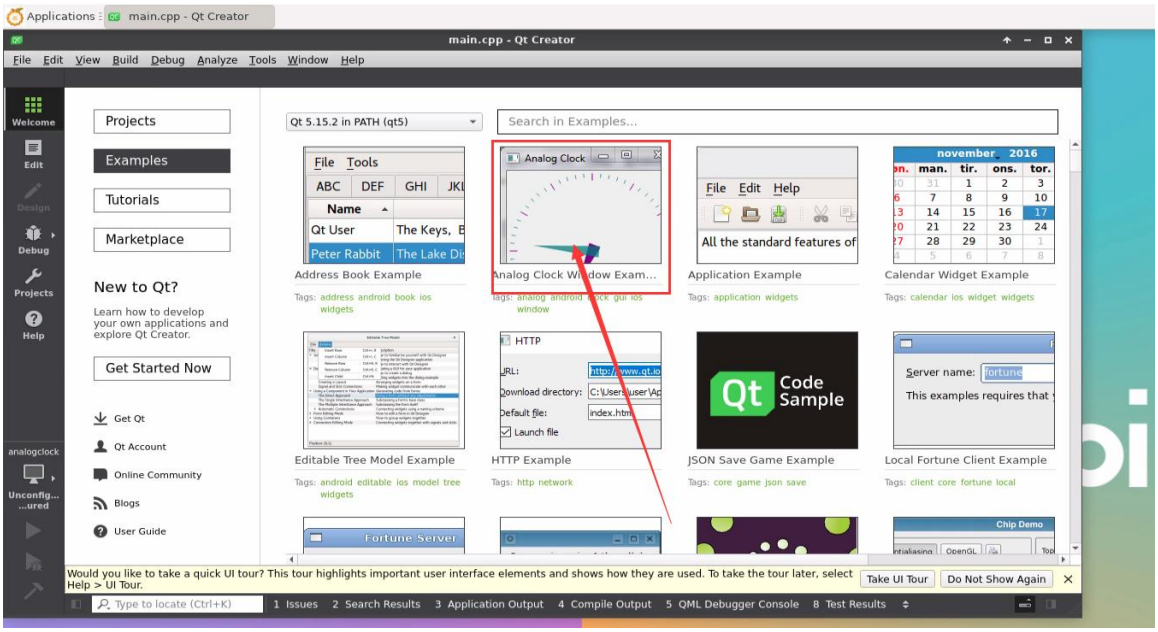
d. Then make sure the GCC compiler used by QT Creator, if the default is Clang, please modify it to GCC

Debian12 please skip this step.

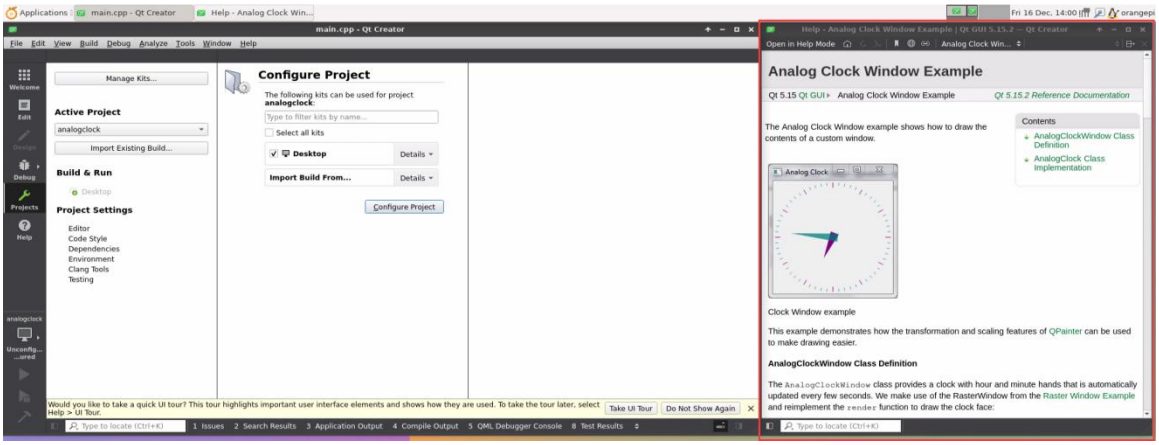




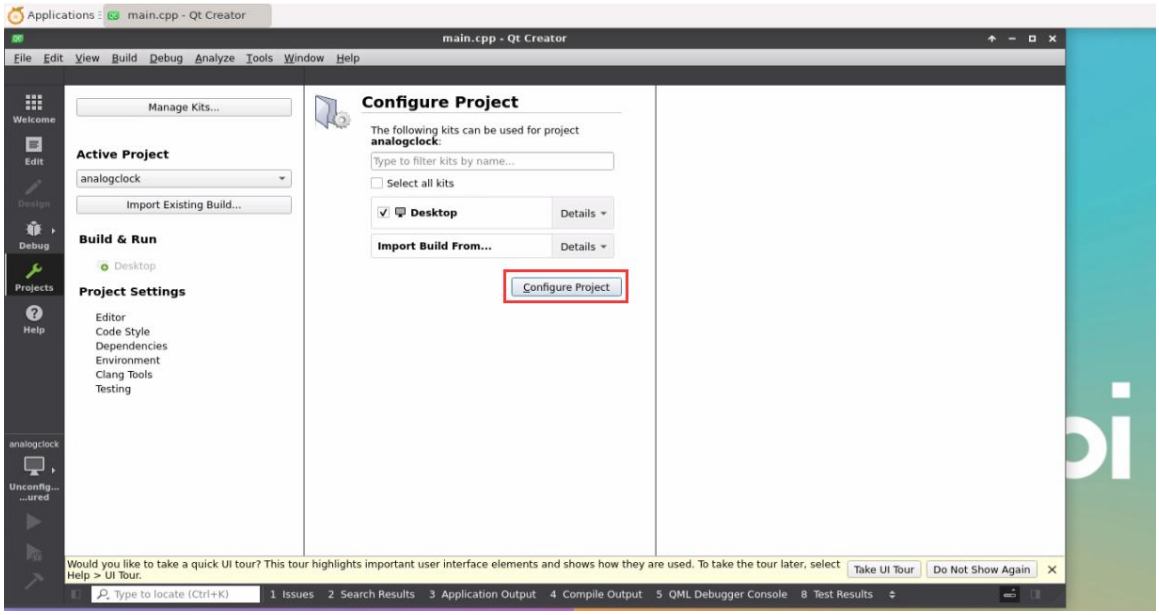
7) Then you can open a sample code



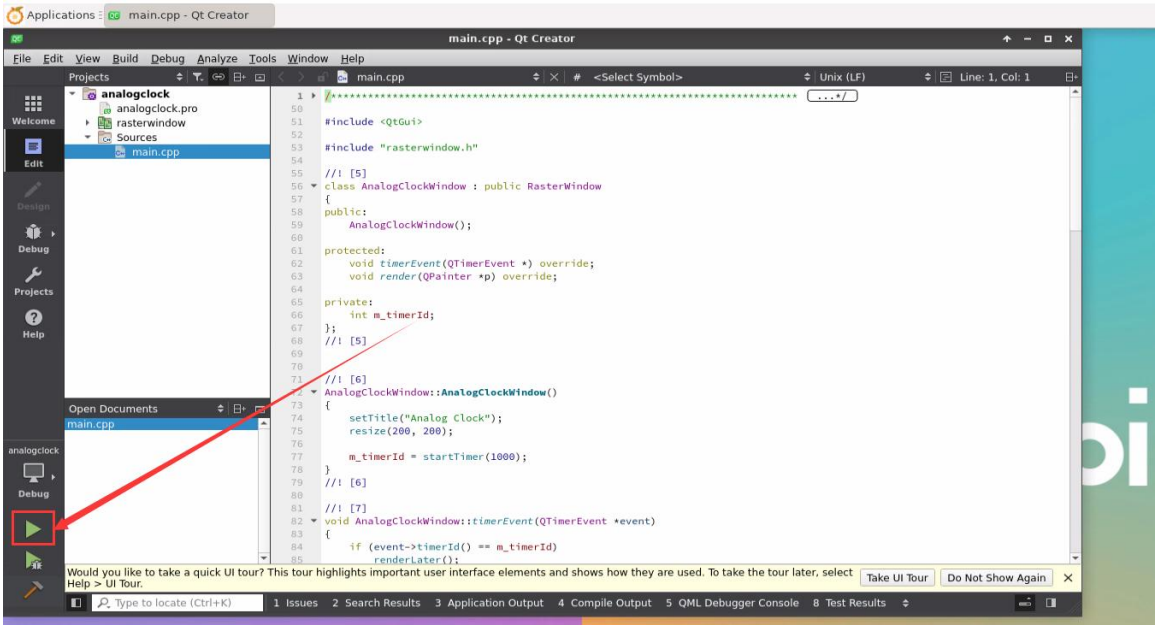
8) After clicking the sample code, the corresponding instruction document will be opened automatically, you can read the instructions carefully



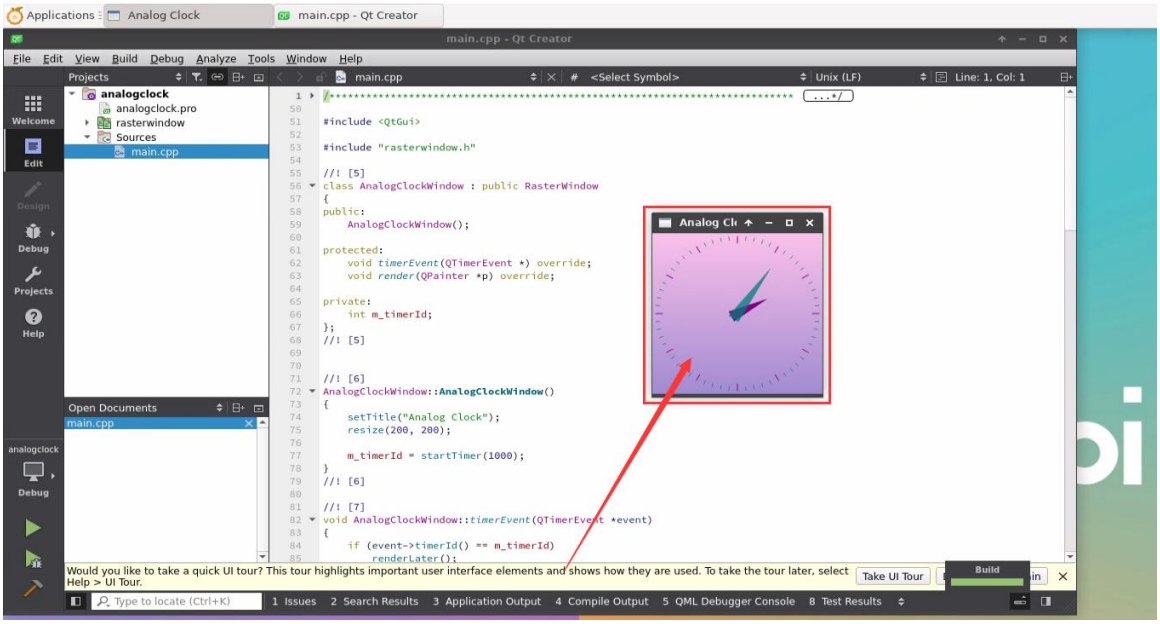
9) Then click **Configure Project**



10) Then click the green triangle in the lower left corner to compile and run the sample code



11) After waiting for a period of time, the interface shown in the figure below will pop up, which means that QT can compile and run normally



12) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu
https://download.qt.io/archive/qtcreator
https://download.qt.io/archive/qt

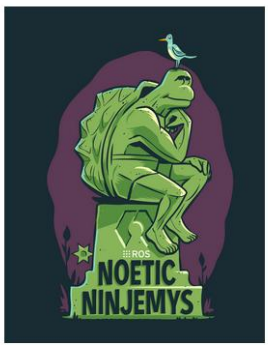
3. 31. How to install ROS

3. 31. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The currently active version of ROS 1 is shown below, and the recommended version is **Noetic Ninjemys**

Active ROS 1 distributions

Recommended



Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

http://docs.ros.org
https://wiki.ros.org/Distributions

2) ROS 1 **Noetic Ninjemys** official installation documentation link is as follows：

http://wiki.ros.org/noetic/Installation/Ubuntu

3) In the ROS **Noetic Ninjemys** official installation document, Ubuntu recommends using Ubuntu20.04, so please make sure that the system used by the development board is the **Ubuntu20.04 desktop system**

http://wiki.ros.org/noetic/Installation

4) Then use the script below to install ros1

orangepi@orangepi:~$ **install_ros.sh ros1**

5) Before using the ROS tool, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code

Note that running the following command needs to ensure that the development board can access github normally, otherwise an error will be reported due to network problems

The install_ros.sh script will try to modify /etc/hosts and run the following commands automatically. However, this method cannot guarantee that github can be accessed normally every time. If install_ros.sh prompts the following error after installing ros1, please find other ways to allow the linux system of the development board to access github normally, and then manually run the following command。

https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
ERROR: error loading sources list:
The read operation timed out

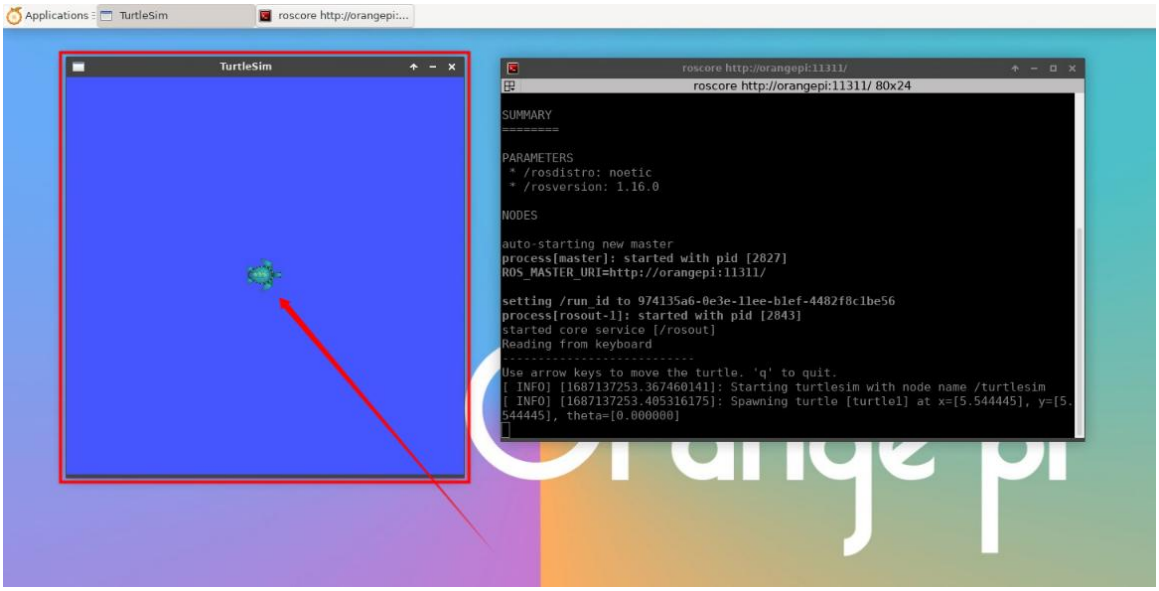
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

        rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
```

```
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

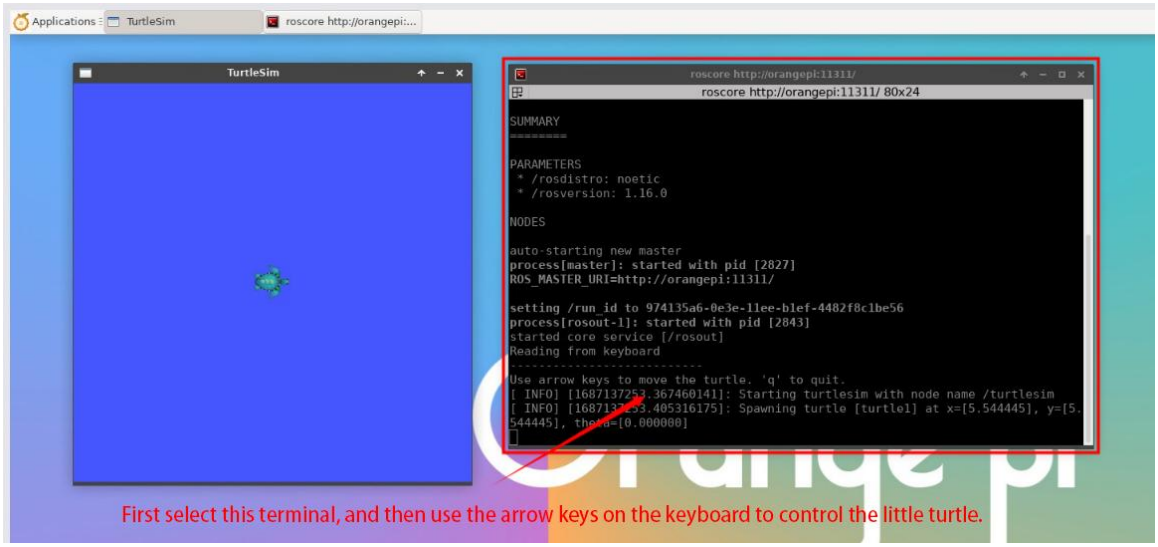
6) Then open a command line terminal window on the **desktop**, and use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally

```
orangepi@orangepi:~$ test_ros.sh
```

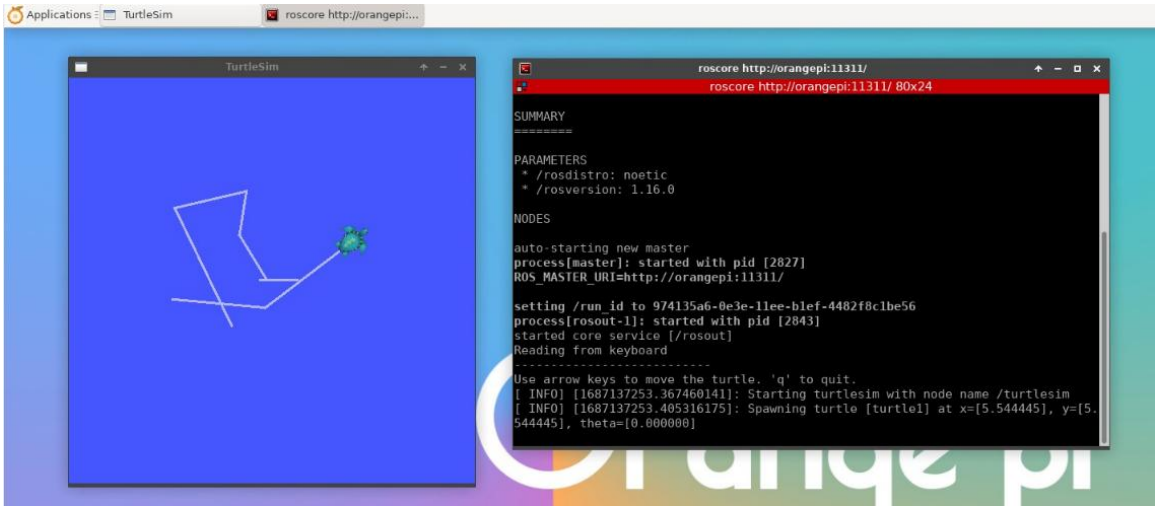
7) After running the **test_ros.sh** script, a little turtle as shown in the figure below will pop up



8) Then please keep the terminal window you just opened on top



9) At this time, press the direction keys on the keyboard to control the little turtle to move up, down, left, and right

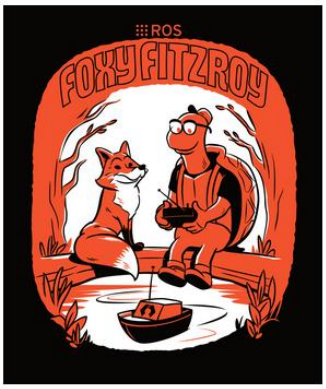


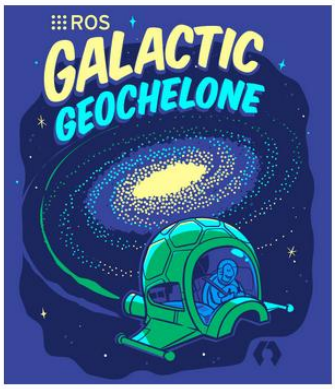
3. 31. 2. How to install ROS 2 Galactic on Ubuntu 20.04

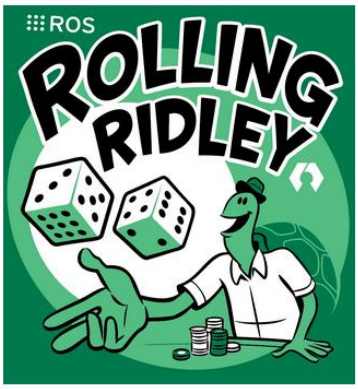
1) The currently active version of ROS 2 is shown below, and the recommended version is **Galactic Geochelone**

Active ROS 2 distributions

Recommended **Development**







Distro	Release date	Logo	EOL date
Humble Hawksbill	May 23rd, 2022		May 2027
Galactic Geochelone	May 23rd, 2021		November 2022
Foxy Fitzroy	June 5th, 2020		May 2023

http://docs.ros.org
http://docs.ros.org/en/galactic/Releases.html

2) The link to the official ROS 2 **Galactic Geochelone** installation documentation is as follows：

docs.ros.org/en/galactic/Installation.html
http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html

3) In the official ROS 2 **Galactic Geochelone** installation document, Ubuntu Linux recommends using Ubuntu 20.04, so please make sure that the system used by the development board is the **Ubuntu 20.04 desktop system**. There are several ways to install ROS 2. The following demonstrates how to install ROS 2 **Galactic Geochelone**

through **Debian packages**.

4) Ros2 can be installed using the **install_ros.sh** script

```
orangepi@orangepi:~$ install_ros.sh ros2
```

5) The **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete

```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

  Call `ros2 <command> -h` for more detailed usage.
```

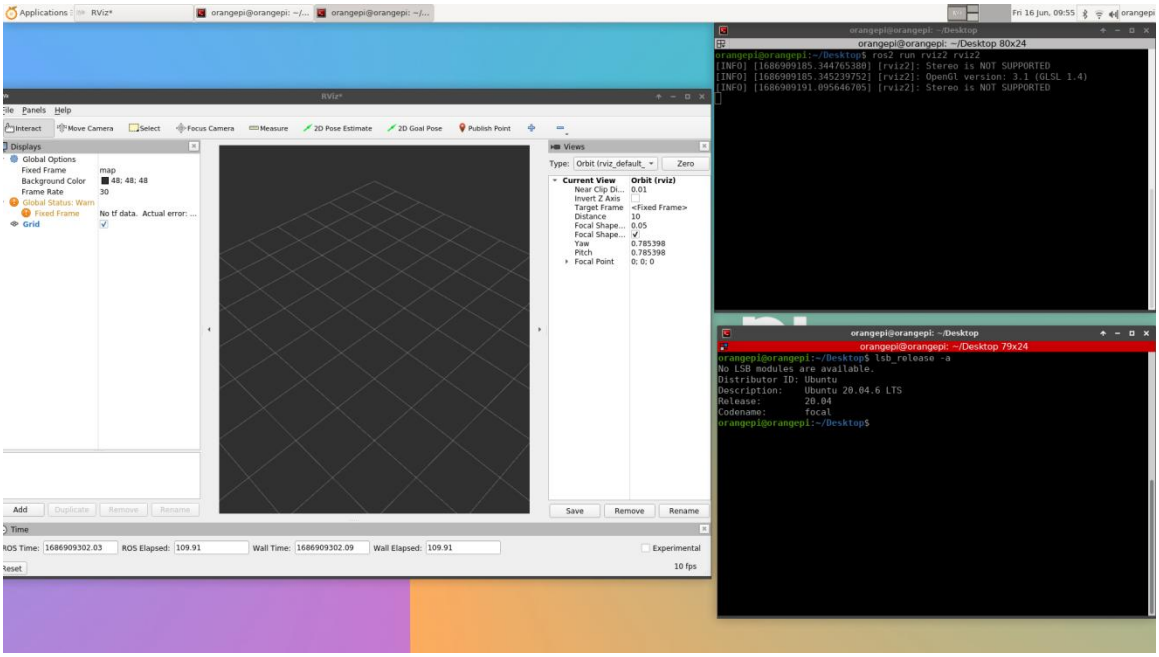
6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully.

If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepi:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



8) For how to use ROS, please refer to the documentation of ROS 2

http://docs.ros.org/en/galactic/Tutorials.html

3. 31. 3. How to install ROS 2 Humble on Ubuntu 22.04

1) Ros2 can be installed using the **install_ros.sh** script

```
orangepi@orangepi:~$ install_ros.sh ros2
```

2) The **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete

```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

  Call `ros2 <command> -h` for more detailed usage.
```

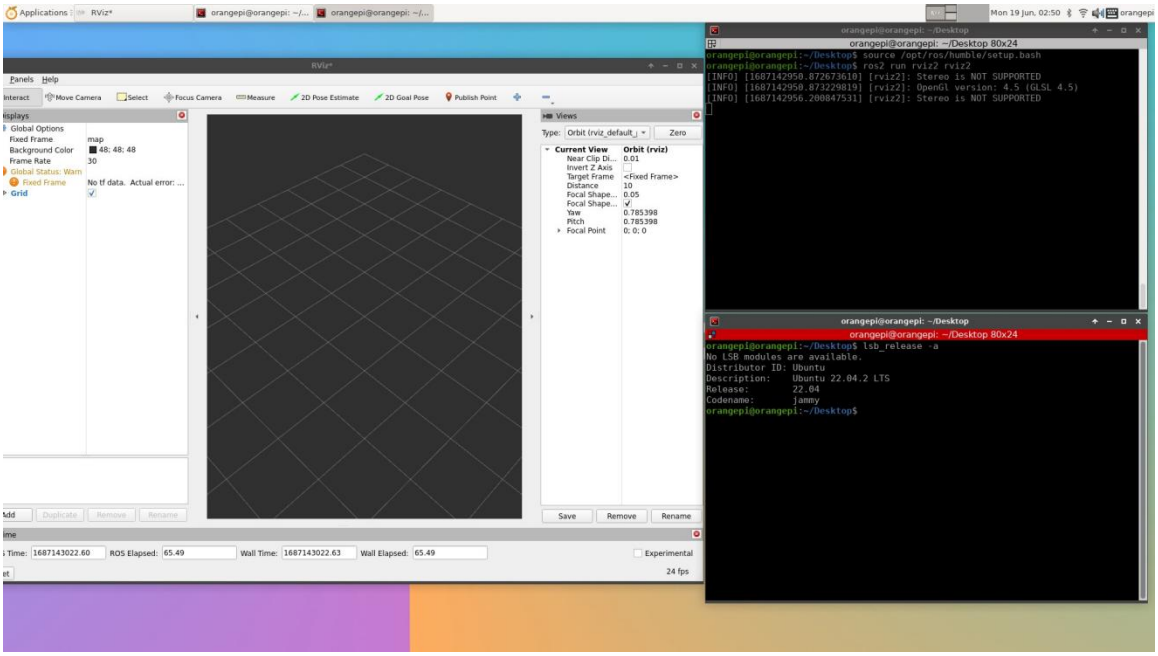
3) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepi:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
```

```
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/humble/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



5) Reference documents

```
http://docs.ros.org/en/humble/index.html
http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html
```

3. 32. How to install the kernel header file

The Debian 11 system with the Linux6.1 kernel will report a GCC error when compiling the kernel module. So if you want to compile the kernel module, please use Debian12 or Ubuntu22.04

1) The Linux image released by OPi comes with the deb package of the kernel header file by default, and the storage location is **/opt/**

```
orangepi@orangepi:~$ ls /opt/linux-headers*
/opt/linux-headers-xxx-sun50iw9_x.x.x_arm64.deb
```

2) Use the following command to install the deb package of the kernel header file

```
orangepi@orangepi:~$ sudo dpkg -i /opt/linux-headers*.deb
```

3) After installation, you can see the folder where the kernel header files are located under **/usr/src**

```
orangepi@orangepi:~$ ls /usr/src
linux-headers-x.x.x
```

4) Then you can compile the source code of the hello kernel module that comes with the Linux image. The source code of the hello module is in **/usr/src/hello**. After entering this directory, use the make command to compile。

```
orangepi@orangepi:~$ cd /usr/src/hello/
orangepi@orangepi:/usr/src/hello$ sudo make
make -C /lib/modules/5.4.125/build M=/usr/src/hello modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.125'
  CC [M]  /usr/src/hello/hello.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /usr/src/hello/hello.mod.o
  LD [M]  /usr/src/hello/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.125'
```

5) After compiling, the **hello.ko** kernel module will be generated

```
orangepi@orangepi:/usr/src/hello$ ls *.ko
hello.ko
```

6) Using the **insmod** command, you can insert the **hello.ko** kernel module into the kernel

```
orangepi@orangepi:/usr/src/hello$ sudo insmod hello.ko
```

7) Then use the **demsg** command to view the output of the **hello.ko** kernel module. If you can see the output instructions below, the **hello.ko** kernel module is loaded correctly

```
orangepi@orangepi:/usr/src/hello$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
```

8) Use the **rmmod** command to uninstall the **hello.ko** kernel module

```
orangepi@orangepi:/usr/src/hello$ sudo rmmod hello
orangepi@orangepi:/usr/src/hello$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
[ 3173.800892] Hello Orange Pi -- exit
```

3. 33. Some programming language tests supported by Linux system

3. 33. 1. Debian Bullseye System

1) Debian Bullseye has a gcc compilation tool chain installed by default, which can directly compile C language programs in the Linux system of the development board

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 10.2.1-6) 10.2.1 20210110
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bullseye comes with Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Use the Ctrl+D shortcut to exit python's interactive mode。

b. Write **hello_world.py** program in Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bullseye does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk, the latest version in Debian Bullseye is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
```

c. Edit **hello_world.java of java version**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run**hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 33. 2. Debian Bookworm System

1) Debian Bookworm is installed with a gcc compilation tool chain by default, which can

directly compile C language programs in the Linux system of the development board

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 12.2.0-14) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bookworm has Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Use the Ctrl+D shortcut to exit python's interactive mode。

b. Write **hello_world.py** program in Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
```

```
Hello World!
```

3) Debian Bookworm does not install Java compilation tools and operating environment by default

a. You can use the following command to install openjdk, the latest version in Debian Bookworm is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
```

c. Edit the Javaverison's **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run**hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3.33.3. Ubuntu Focal System

1) Ubuntu Focal has a gcc compilation tool chain installed by default, which can directly compile C language programs in the Linux system of the development board

a. The version of a.gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>
```

```
int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Ubuntu Focal has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.8.10 (default, Nov 14 2022, 12:59:47)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Use the Ctrl+D shortcut to exit python's interactive mode.

b. Write the **hello_world.py** program in Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Ubuntu Focal does not install Java compilation tools and operating environment by default

a. You can use the following command to install openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
openjdk 17.0.2 2022-01-18
OpenJDK Runtime Environment (build 17.0.2+8-Ubuntu-120.04)
```

```
OpenJDK 64-Bit Server VM (build 17.0.2+8-Ubuntu-120.04, mixed mode, sharing)
```

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 33. 4. Ubuntu Jammy System

1) Ubuntu Jammy is installed with a gcc compilation tool chain by default, which can directly compile C language programs in the Linux system of the development board

a. The version of a.gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.3.0-1ubuntu1~22.04.1) 11.3.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Ubuntu Jammy has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.6 (main, May 29 2023, 11:10:38) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Use the Ctrl+D shortcut to exit python's interactive mode.

b. Write the **hello_world.py** program in Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Ubuntu Jammy does not install Java compilation tools and operating environment by default

a. You can use the following command to install openjdk-18

```
orangepi@orangepi:~$ sudo apt install -y openjdk-18-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
openjdk 18.0.2-ea 2022-07-19
OpenJDK Runtime Environment (build 18.0.2-ea+9-Ubuntu-222.04)
OpenJDK 64-Bit Server VM (build 18.0.2-ea+9-Ubuntu-222.04, mixed mode, sharing)
```

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
```

```
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 34. The method of uploading files to the Linux system of the development board

3. 34. 1. How to upload files to the development board Linux system in Ubuntu PC

3. 34. 1. 1. How to upload files using the scp command

1) Use the scp command to upload files from the Ubuntu PC to the Linux system of the development board. The specific commands are as follows

a. **file_path：** Need to be replaced with the path of the file to be uploaded

b. **orangepi：** It is the user name of the Linux system of the development board, and it can also be replaced with other ones, such as root

c. **192.168.xx.xx:** It is the IP address of the development board, please modify it according to the actual situation

d. **/home/orangepi:** The path in the Linux system of the development board can also be modified to other paths

```
test@test:~$ scp    file_path    orangepi@192.168.xx.xx:/home/orangepi/
```

2) If you want to upload a folder, you need to add the -r parameter

```
test@test:~$ scp    -r    dir_path    orangepi@192.168.xx.xx:/home/orangepi/
```

3) There are more usages of scp, please use the following command to view the man manual

```
test@test:~$ man scp
```

3. 34. 1. 2. How to upload files using filezilla

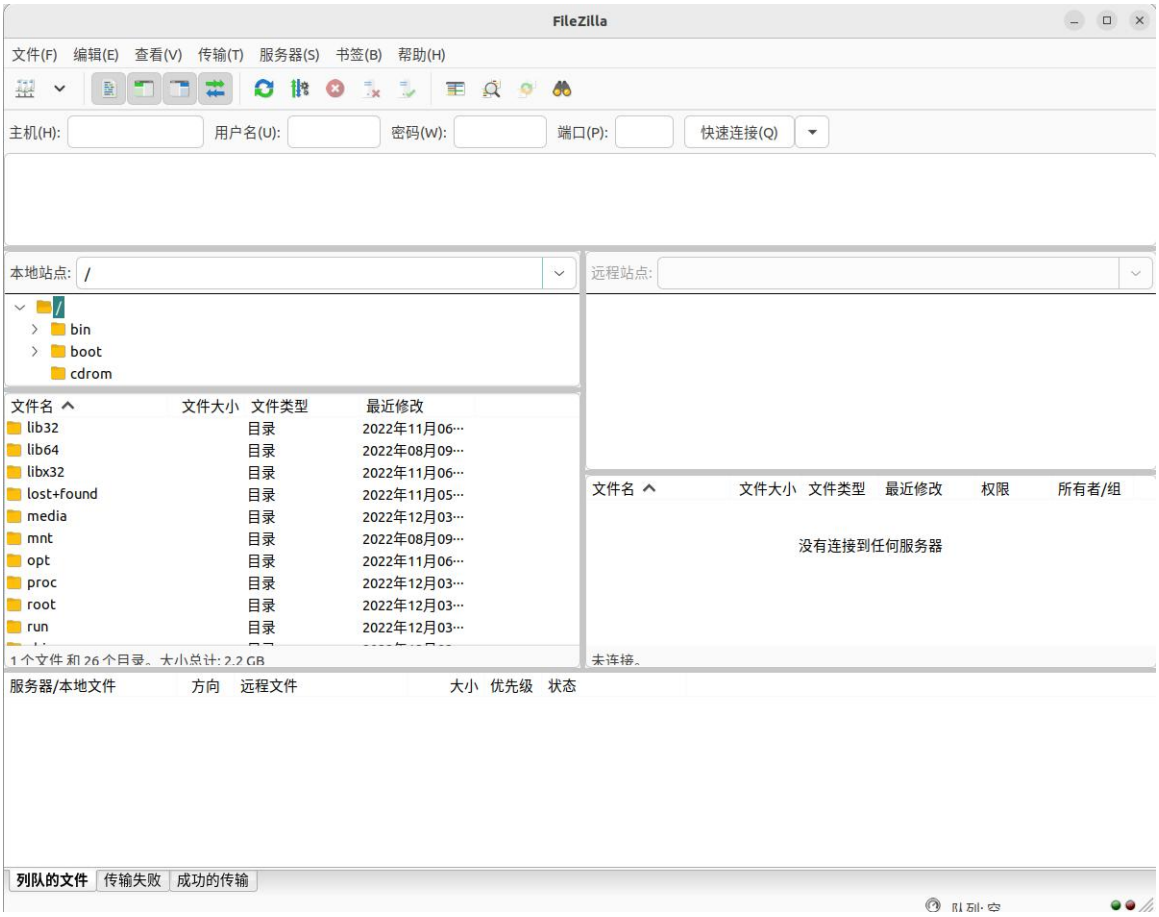
1) First install filezilla in Ubuntu PC

```
test@test:~$ sudo apt install -y filezilla
```

2) Then use the following command to open filezilla

```
test@test:~$ filezilla
```

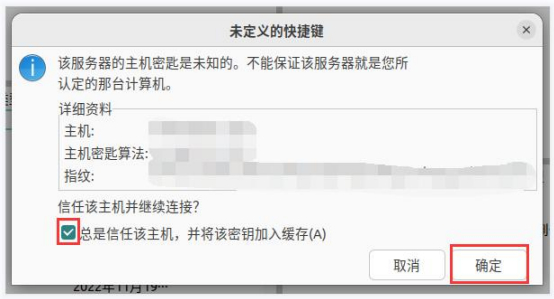
3) The interface after filezilla is opened is as follows, at this time, the display under the remote site on the right is empty



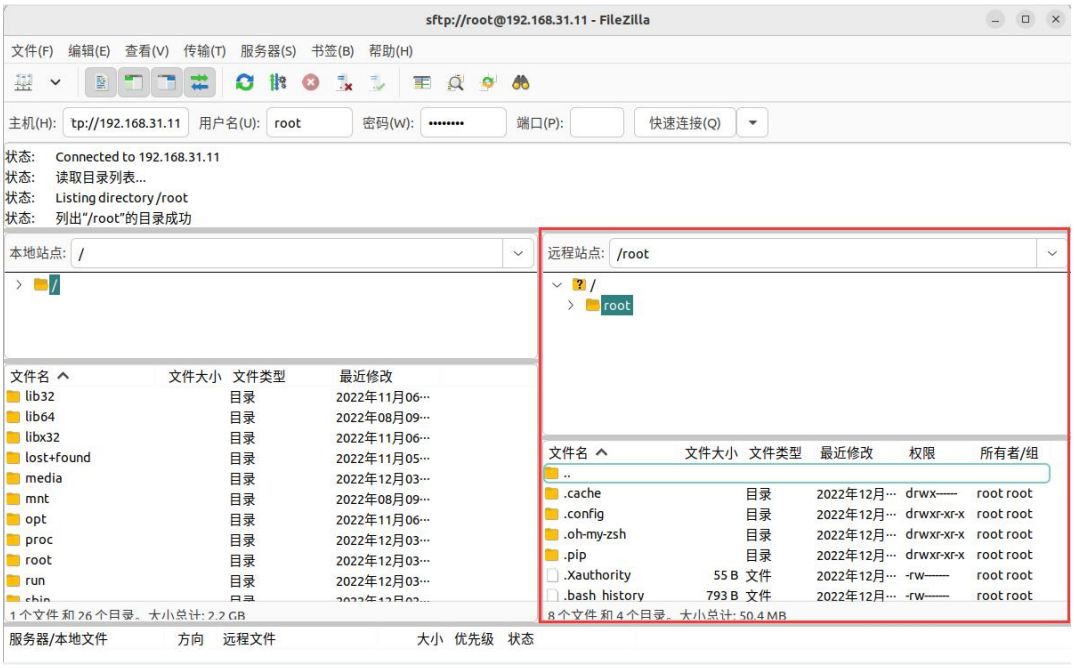
4) The method of connecting the development board is shown in the figure below

5) Then choose to **save the password**, and then click **OK**

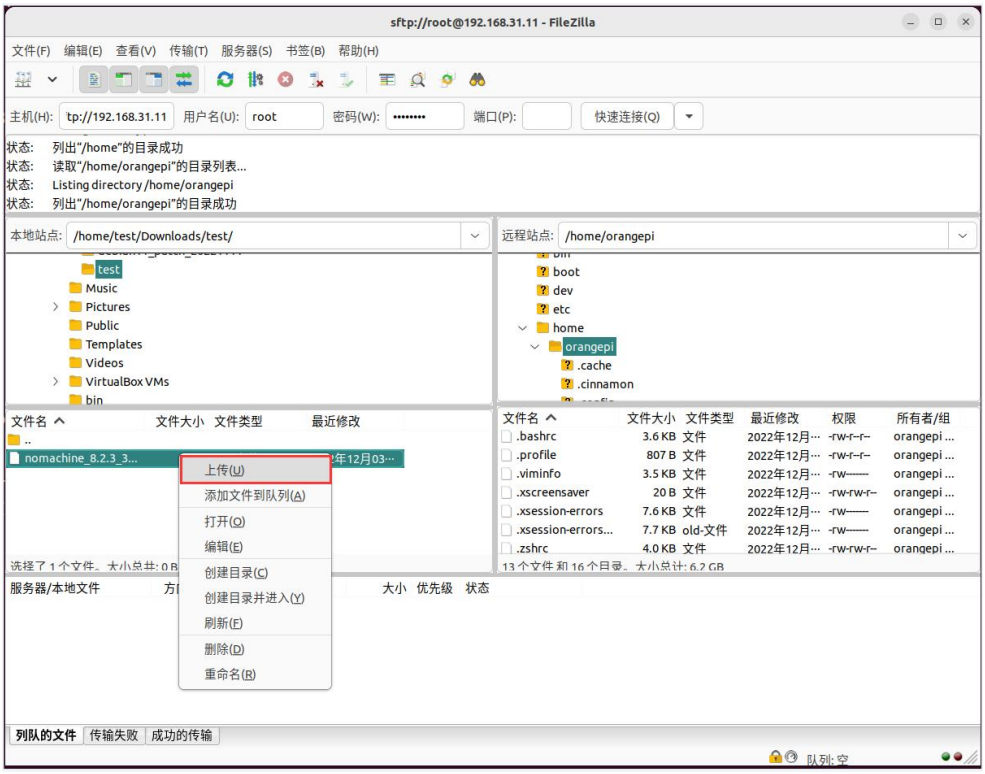
6) Then choose to **always trust this host**, and then click **OK**



7) After the connection is successful, you can see the directory structure of the development board linux file system on the right side of the filezilla software



8) Then select the path to be uploaded to the development board on the right side of the filezilla software, and then select the file to be uploaded on the Ubuntu PC on the left side of the filezilla software, then click the right mouse button, and then click the upload option to start uploading the file to the development board.



9) After the upload is complete, you can go to the corresponding path in the Linux system of the development board to view the uploaded files

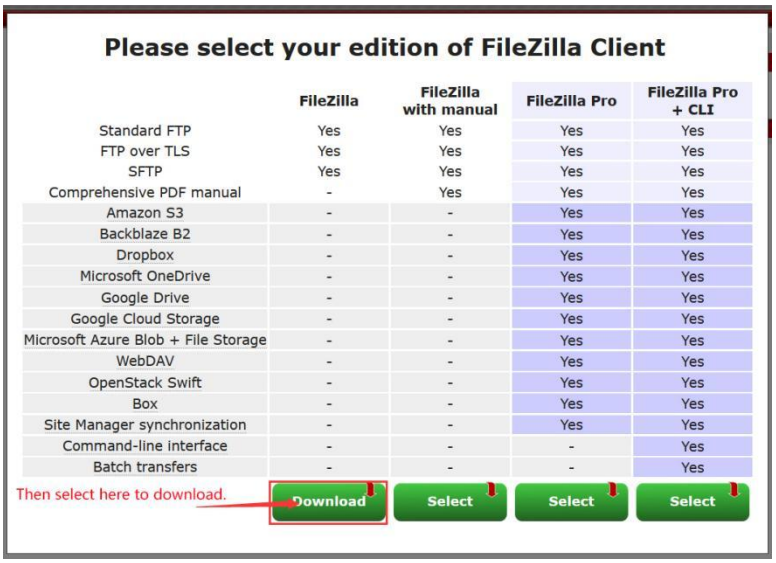
10) The method of uploading a folder is the same as that of uploading a file, so I won't go into details here

3.34.2. How to upload files to the development board Linux system in Windows PC

3.34.2.1. How to upload files using filezilla

1) First download the installation file of the Windows version of the filezilla software, the download link is as follows

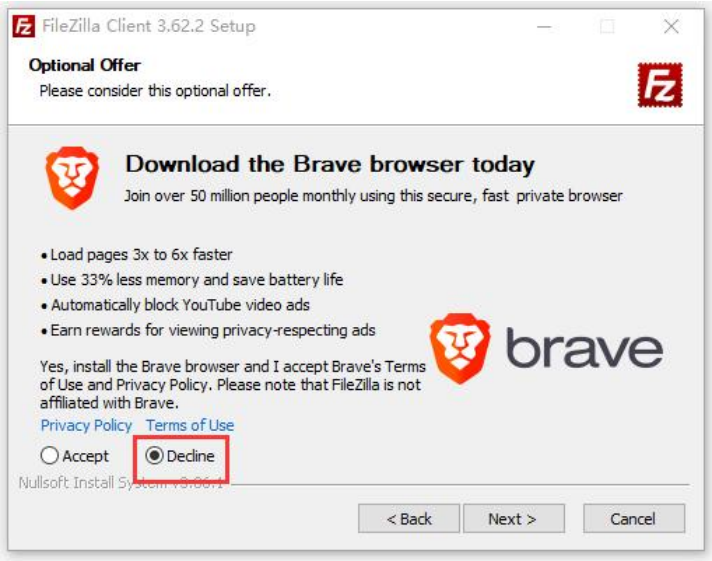
https://filezilla-project.org/download.php?type=client



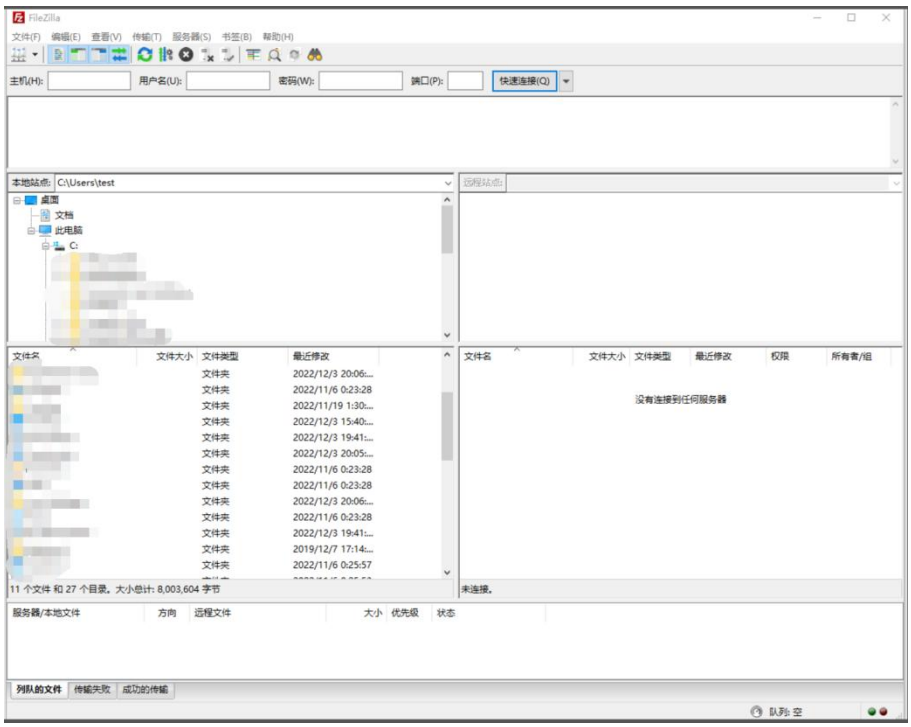
2) The downloaded installation package is as follows, and then double-click to install directly

FileZilla_Server_1.5.1_win64-setup.exe

During the installation process, please select Decline on the following installation interface, and then select Next>



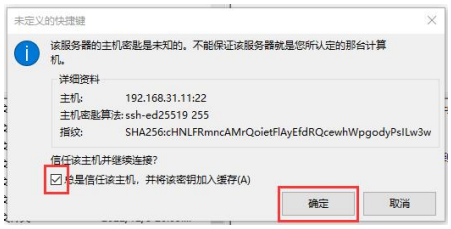
3) The interface after filezilla is opened is as follows, and the display under the remote site on the right is empty



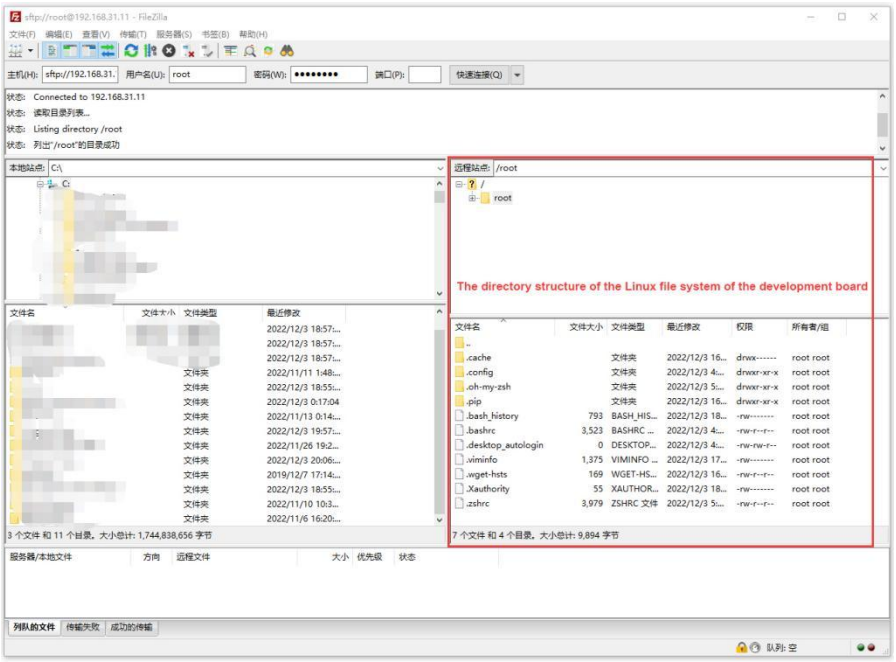
4) The method of connecting the development board is shown in the figure below:

5) Then choose to **save the password**, and then click **OK**

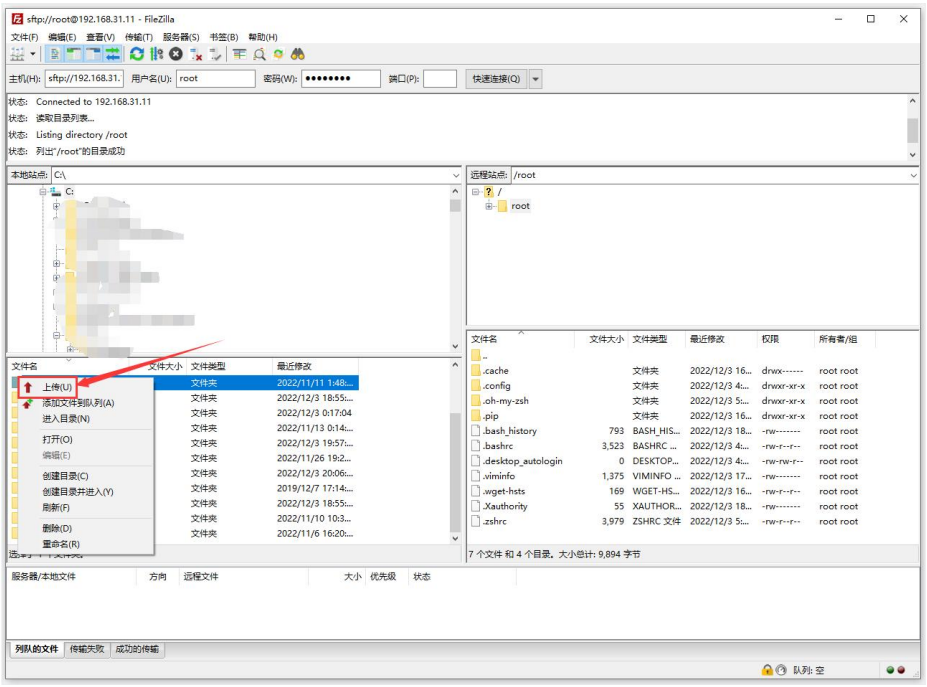
6) Then select **Always trust this host**, and click **OK**



7) After the connection is successful, you can see the directory structure of the development board linux file system on the right side of the filezilla software



8) Then select the path to be uploaded to the development board on the right side of the filezilla software, and then select the file to be uploaded on the Windows PC on the left side of the filezilla software, then click the right mouse button, and then click the upload option to start uploading the file to the development board.



9) After the upload is complete, you can go to the corresponding path in the Linux system of the development board to view the uploaded files

10) The method of uploading a folder is the same as that of uploading a file, so I won't go into details here

3. 35. Instructions for using the switch logo

1) By default, the switch logo will only be displayed in the desktop version of the system

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the switch logo

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=false
```

3) Set the **bootlogo** variable to **true** in **/boot/orangepiEnv.txt** to enable the switch logo

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
```

4) The location of the boot logo image in the Linux system is

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

5) After replacing the boot logo picture, you need to run the following command to take effect

```
orangepi@orangepi:~$ sudo update-initramfs -u
```

3. 36. How to shut down and restart the development board

1) During the running of the Linux system, if the power is directly unplugged, the file system may lose some data. It is recommended to use the **poweroff** command to shut down the Linux system of the development board before power off, and then unplug the power

```
orangepi@orangepi:~$ sudo poweroff
```

Note that after the development board is turned off, the power supply needs to be re-plugged to turn it on.

2) Use the **reboot** command to restart the Linux system on the development board

```
orangepi@orangepi:~$ sudo reboot
```

4. Linux SDK——orangepi-build instruction

4. 1. Compilation system requirements

The Linux SDK, **orangepi-build**, only supports running on X64 computers with **Ubuntu 22.04** installed, so before downloading orangepi-build, please make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on the computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please replace the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04 LTS
Release:        22.04
Codename:       jammy
```

If the computer is installed with a Windows system and there is no computer with Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in the Windows system. But please note, do not compile orangepi-build on the WSL virtual machine, because orangepi-build has not been tested in the WSL virtual machine, so it cannot be guaranteed that orangepi-build can be used normally in WSL, and please do not compile it on the Linux system of the development

board Use orangepi-build in. The download address of the installation image of Ubuntu 22.04 **amd64** version is:

https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso

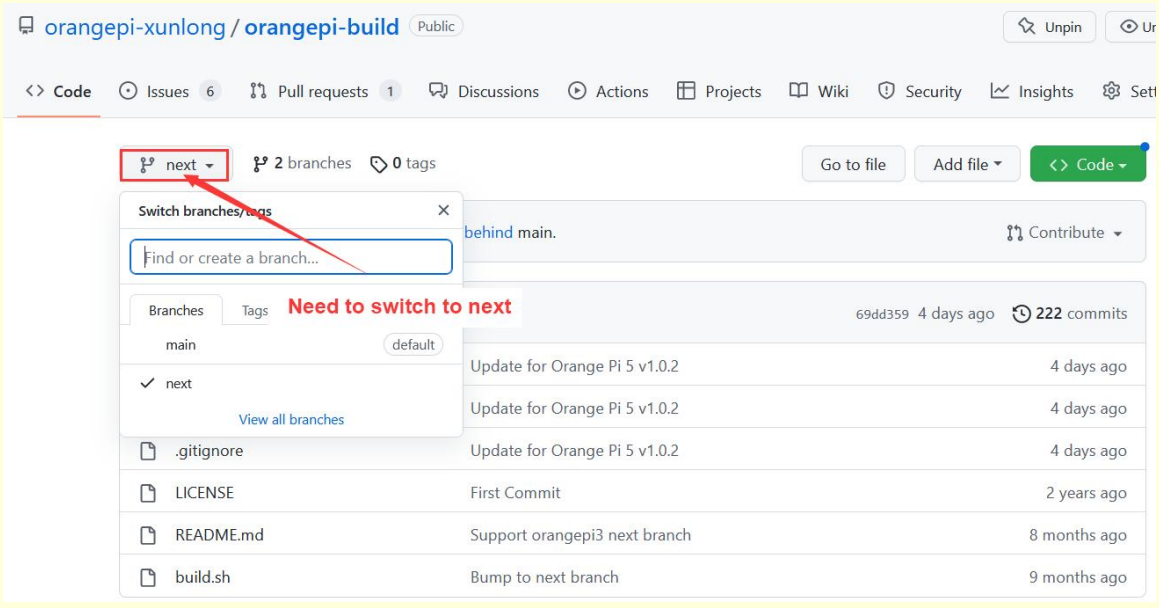
4. 2. Get the source code of linux sdk

4. 2. 1. Download orangepi-build from github

The linux sdk refers to the code of orangepi-build. orangepi-build is modified based on the armbian build system. Using orangepi-build, multiple versions of linux images can be compiled. Use the following command to download the orangepi-build code:

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y git
test@test:~$ git clone https://github.com/orangepi-xunlong/orangepi-build.git -b next
```

Note that the development board using the H618 Soc needs to download the source code of the next branch of orangepi-build. The above git clone command needs to specify the branch of the orangepi-build source code as next.



Downloading the orangepi-build code through the git clone command does not require entering the user name and password of the github account (the same is true for downloading other codes in this manual), if the Ubuntu PC prompts the user to

enter the github account after entering the git clone command The name and password are usually entered incorrectly in the address of the orangepi-build warehouse behind the git clone. Please check the spelling of the command carefully, instead of thinking that we forgot to provide the username and password of the github account.

The u-boot and linux kernel versions currently used by the H618 series development boards are as follows:

branch	u-boot version	Linux kernel version
current	u-boot v2018.05	linux5.4
next	u-boot v2021.07	linux6.1

The branch mentioned here is not the same thing as the branch of the orangepi-build source code, please do not confuse it. This branch is mainly used to distinguish different kernel source code versions.

We currently define the linux5.4 bsp kernel provided by Allwinner as the current branch. The latest linux6.1 LTS kernel is defined as the next branch.

After orangepi-build is downloaded, it will contain the following files and folders:

a. **build.sh**: Compile the startup script
b. **external**: Contains the configuration files needed to compile the image, specific scripts, and the source code of some programs, etc.
c. **LICENSE**: GPL 2 license file
d. README.md: orangepi-build documentation
e. **scripts**: General script for compiling linux images

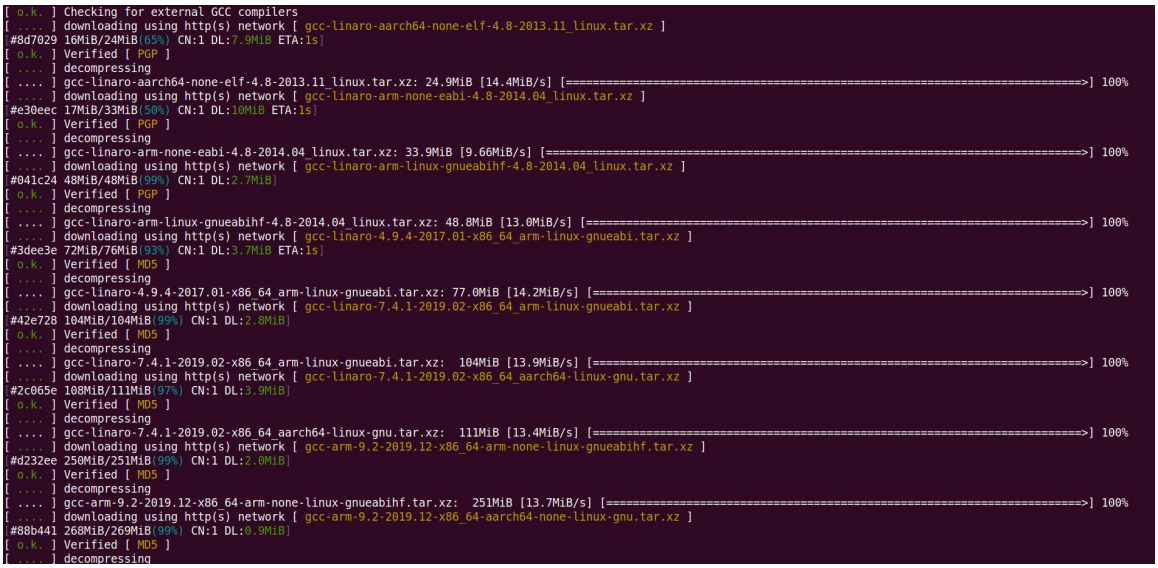
```
test@test:~/orangepi-build$ ls
build.sh  external  LICENSE  README.md  scripts
```

If you downloaded the code of orangepi-build from github, after downloading, you may find that orangepi-build does not contain the source code of u-boot and linux kernel, nor does u-boot and linux kernel need to use cross-compilation tools Chain, this is normal, because these things are stored in other separate github warehouses or some servers (the addresses will be detailed below). orangepi-build will specify the address of u-boot, linux kernel and cross-compilation toolchain in the script and configuration file. When running orangepi-build, when it finds that

there are no such things locally, it will automatically go to the corresponding place to download them.

4. 2. 2. Download the cross-compilation toolchain

When orangepi-build runs for the first time, it will automatically download the cross-compilation toolchain and put it in the **toolchains** folder. Every time you run the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists. If If it does not exist, the download will be restarted, if it exists, it will be used directly, and the download will not be repeated.



The mirror URL of the cross-compilation toolchain in China is the open source software mirror site of armbian University:

https://imola.armbian.com/dl/_toolchain/

After the **toolchains** are downloaded, it will contain multiple versions of the cross-compilation toolchain:

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
gcc-linaro-4.9.4-2017.01-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-arm-11.2-2022.02-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

```
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

The cross-compilation toolchain used to compile the H618 Linux kernel source code is:

a. linux5.4

gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu

b. linux6.1

gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu

The cross-compilation toolchain used to compile the H618 u-boot source code is:

a. v2018.05

gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi

b. v2021.07

gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu

4.2.3. orangepi-build complete directory structure description

1) The orangepi-build repository does not contain the source code of the linux kernel, u-boot, and cross-compilation toolchain after downloading. The source code of the linux kernel and u-boot is stored in an independent git repository

a. The git warehouse where the linux kernel source code is stored is as follows, pay attention to switch the branch of the linux-orangepi warehouse to

a) Linux5.4

https://github.com/orangepi-xunlong/linux-orangepi/tree/**orange-pi-5.4-sun50iw9**

b) Linux6.1

https://github.com/orangepi-xunlong/linux-orangepi/tree/**orange-pi-6.1-sun50iw9**

b. The git warehouse where the u-boot source code is stored is as follows, pay attention to switch the branch of the u-boot-orangepi warehouse to

a) v2018.05

https://github.com/orangepi-xunlong/u-boot-orangepi/tree/**v2018.05-h618**

b) v2021.07

https://github.com/orangepi-xunlong/u-boot-orangepi/tree/**v2021.07-sunxi**

2) When orangepi-build runs for the first time, it will download the cross-compilation toolchain, u-boot and linux kernel source code. After successfully compiling a linux image, the files and folders that can be seen in orangepi-build are:

a. **build.sh**: Compile the startup script
b. **external**: Contains the configuration files needed for compiling the image, scripts for specific functions, and the source code of some programs. The rootfs compressed package cached during the compiling process is also stored in external
c. **kernel**: Store the source code of the linux kernel
d. **LICENSE**: GPL 2 license file
e. **README.md**: orangepi-build documentation
f. **output**: Store compiled u-boot, linux and other deb packages, compilation logs, and compiled images and other files
g. **scripts**: General script for compiling linux images
h. **toolchains**: Store the cross-compilation toolchain
i. **u-boot**: Store the source code of u-boot
j. **userpatches**: Store the configuration files needed to compile the script

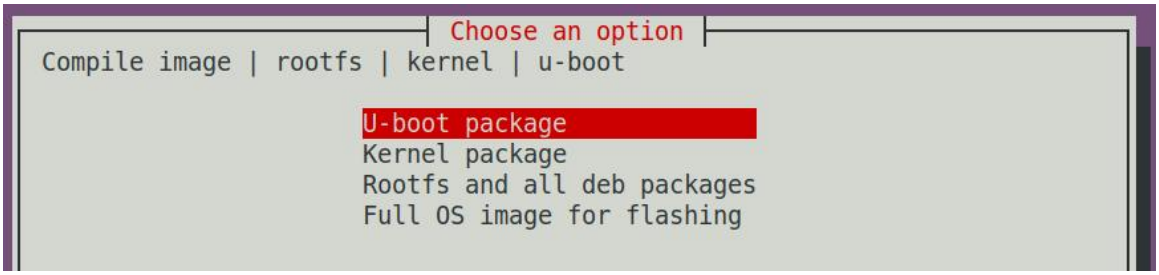
```
test@test:~/orangepi-build$ ls
build.sh  external  kernel  LICENSE  output  README.md  scripts  toolchains  u-boot  userpatches
```

4. 3. Compile u-boot

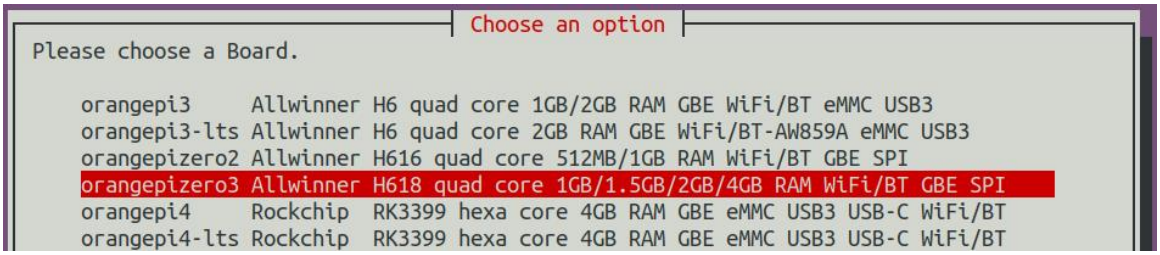
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package**, then press Enter



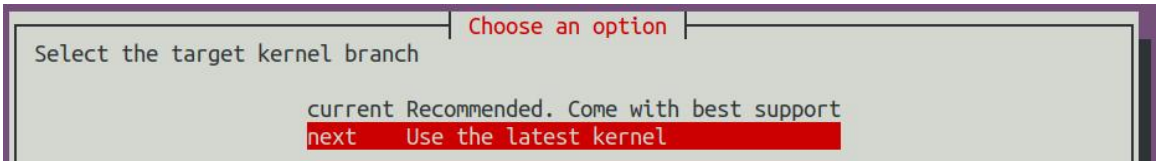
3) Then select the model of the development board



4) Then select the branch type of u-boot

a. The current branch will compile the u-boot v2018.05 code that needs to be used in the linux5.4 image

b. The next branch will compile the u-boot v2021.07 version code that needs to be used in the linux6.1 image



5) Then it will start to compile u-boot, and part of the information prompted when compiling the next branch is as follows:

a. u-boot source code version

[o.k.] Compiling u-boot [**v2021.07**]

b. The version of the cross-compilation toolchain

[o.k.] Compiler version [**aarch64-linux-gnu-gcc 11**]

c. Path to the generated u-boot deb package

[o.k.] Target directory [**orangepi-build/output/debs/u-boot**]

d. The package name of the generated u-boot deb package

[o.k.] File name [**linux-u-boot-next-orangepizero3_x.x.x_arm64.deb**]

e. Compilation time

[o.k.] Runtime [**1 min**]

f. Repeat the command to compile u-boot, use the following command to start compiling u-boot directly without selecting through the graphical interface

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepizero3 BRANCH=next BUILD_OPT=u-boot**]

6) View the u-boot deb package generated by compilation

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-next-orangepizero3_x.x.x_arm64.deb
```

7) When the orangepi-bulid compilation system compiles the u-boot source code, it will first synchronize the u-boot source code with the u-boot source code of the github server, so if you want to modify the u-boot source code, you first need to turn off the download and update function of the source code **(This function needs to be fully compiled once u-boot, otherwise it will prompt that the source code of u-boot cannot be found)**, otherwise the changes made will be restored, the method is as follows:

Set the IGNORE_UPDATES variable in **userpatches/config-default.conf** to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
.....
IGNORE_UPDATES="yes"
.....
```

8) When debugging u-boot code, you can use the following method to update u-boot in the linux image for testing

a. First upload the compiled u-boot deb package to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-next-orangepizero3_x.x.x_arm64.deb root@192.168.1.xxx:/root
```

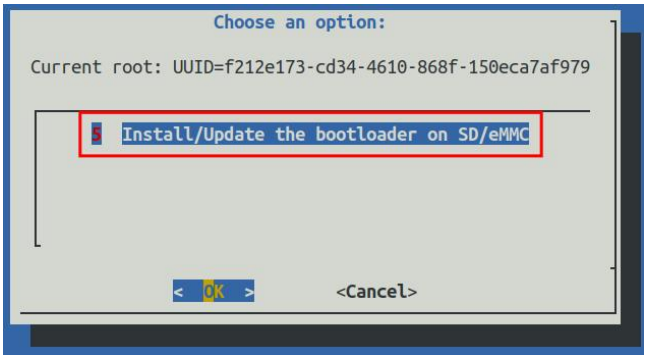
b. Install the new u-boot deb package just uploaded

```
orangepi@orangepi:~$ sudo dpkg -i linux-u-boot-next-orangepizero3_x.x.x_arm64.deb
```

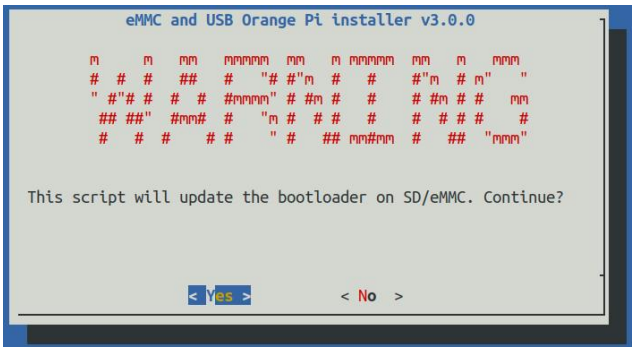
c. Then run the nand-sata-install script

```
orangepi@orangepi:~$ sudo nand-sata-install
```

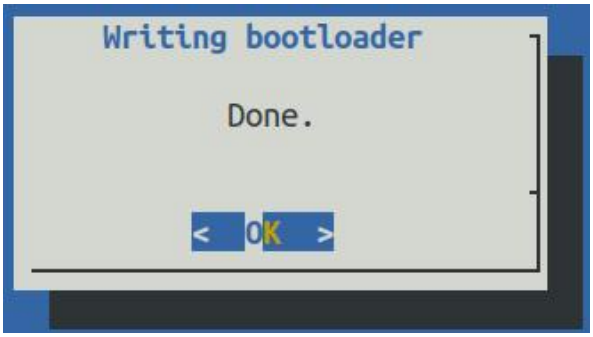
d. Then select **5 Install/Update the bootloader on SD/eMMC**



e. After pressing the Enter key, a Warning will pop up first



f. Press the Enter key again to start updating u-boot, and the following information will be displayed after the update



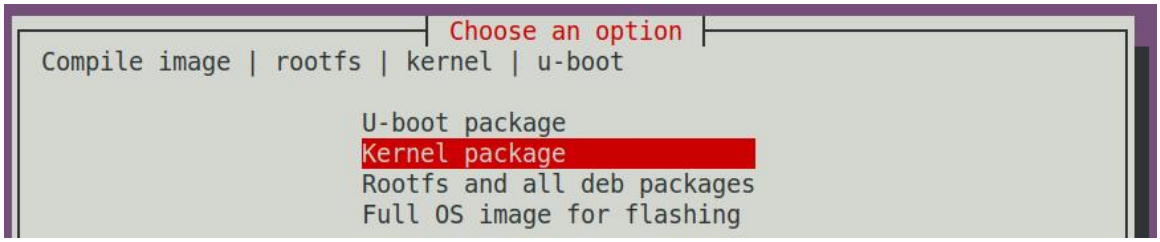
g. Then you can restart the development board to test whether the modification of u-boot takes effect

4. 4. Compile the linux kernel

1) Run the **build.sh** script, remember to add sudo permission

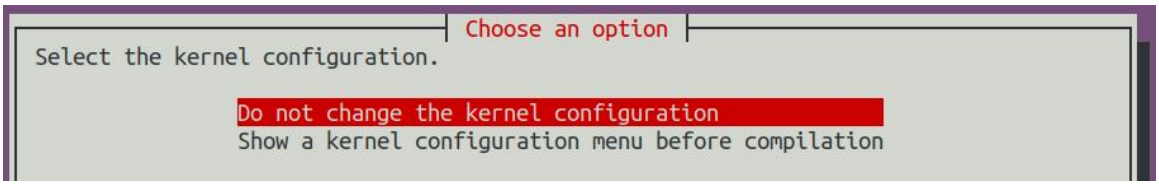
```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Kernel package**, then press Enter

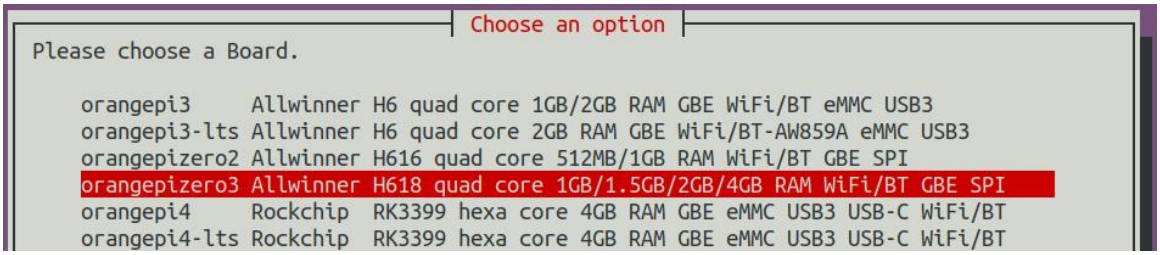


3) Then you will be prompted whether you need to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If

you need to modify the kernel configuration, select the second one.



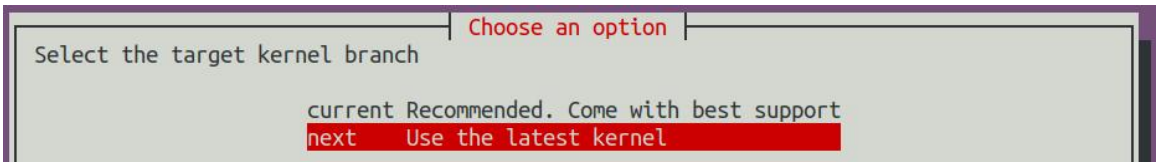
4) Then select the model of the development board



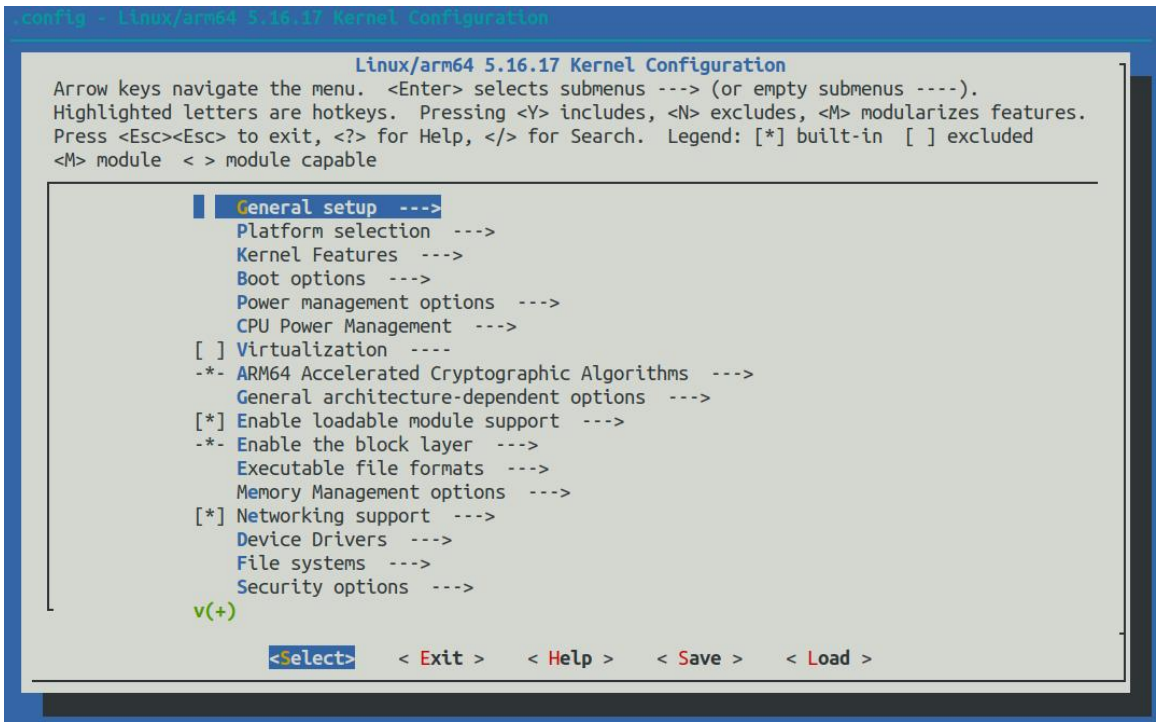
5) Then select the branch type of the kernel source code

a. current branch will compile linux5.4 kernel source code

b. The next branch will compile the linux6.1 kernel source code



6) If you choose to display the kernel configuration menu (the second option) in step 3), the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration, save and exit after modification. Yes, after exiting, it will start compiling the kernel source code.



a. If you do not need to modify the configuration options of the kernel, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily block the pop-up kernel configuration interface

```
test@test:~/orangepi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

b. You can also set **KERNEL_CONFIGURE=no** in the orangepi-build/userpatches/config-default.conf configuration file, which can permanently disable this function

c. If the following error is displayed when compiling the kernel, it is because the terminal interface of the Ubuntu PC is too small to display the make menuconfig interface. Please maximize the terminal of the Ubuntu PC and run the build.sh script again

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

7) Part of the information prompted when compiling the next branch kernel source code is as follows：

a. The version of the linux kernel source code

```
[ o.k. ] Compiling current kernel [ 6.1.31 ]
```

b. The version of the cross-compilation toolchain used

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 11 ]
```

c. The configuration file used by the kernel by default and the path where it is stored are as follows

```
[ o.k. ] Using kernel config file
[ orangepi-build/external/config/kernel/linux-6.1-sun50iw9-next.config ]
```

d. The path of the deb package related to the kernel generated by compiling

```
[ o.k. ] Target directory [ output/debs/ ]
```

e. The package name of the compiled kernel image deb package

```
[ o.k. ] File name [ linux-image-next-sun50iw9_x.x.x_arm64.deb ]
```

f. The time used for compilation

```
[ o.k. ] Runtime [ 10 min ]
```

g. Finally, the compilation command to repeatedly compile the kernel selected last time will be displayed. Use the following command to start compiling the kernel source code directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepizero3
BRANCH=next BUILD_OPT=kernel KERNEL_CONFIGURE=no ]
```

8) View the deb package related to the kernel generated by compilation

a. **linux-dtb-next-sun50iw9_x.x.x_arm64.deb** Contains dtb files used by the kernel

b. **linux-headers-next-sun50iw9_x.x.x_arm64.deb** Include kernel headers

c. **linux-image-next-sun50iw9_x.x.x_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-next-sun50iw9_x.x.x_arm64.deb
output/debs/linux-headers-next-sun50iw9_x.x.x_arm64.deb
output/debs/linux-image-next-sun50iw9_x.x.x_arm64.deb
```

9) When the orangepi-bulid compilation system compiles the linux kernel source code, it

first synchronizes the linux kernel source code with the linux kernel source code of the github server, so if you want to modify the linux kernel source code, you first need to turn off the update function of the source code **(the linux kernel needs to be fully compiled once This function can only be turned off after the source code, otherwise it will prompt that the source code of the linux kernel cannot be found)**, otherwise the changes made will be restored, the method is as follows：

Set the IGNORE_UPDATES variable in **userpatches/config-default.conf** to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If the kernel has been modified, the following method can be used to update the kernel and kernel modules of the development board linux system

a. Upload the deb package of the compiled linux kernel to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-next-sun50iw9_x.x.x_arm64.deb root@192.168.1.xxx:/root
```

b. Install the deb package of the new linux kernel just uploaded

```
orangepi@orangepi:~$ sudo dpkg -i linux-image-next-sun50iw9_x.x.x_arm64.deb
```

c. Then restart the development board, and then check whether the kernel-related modifications have taken effect

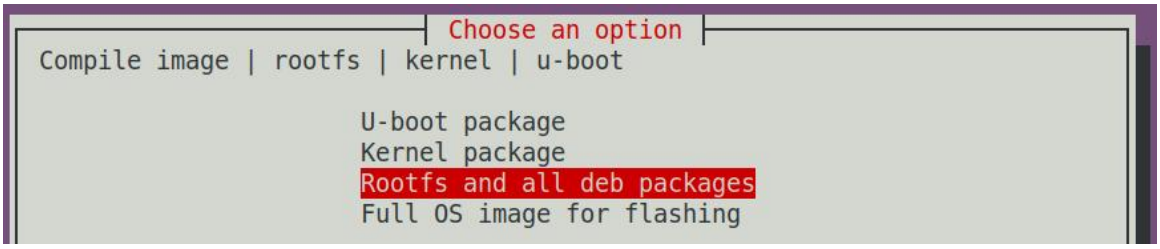
```
orangepi@orangepi:~$ sudo reboot
```

4. 5. Compile rootfs

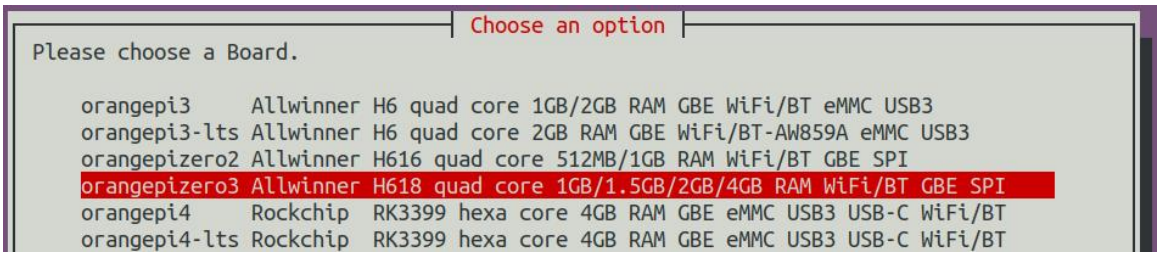
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Rootfs and all deb packages**, then press Enter

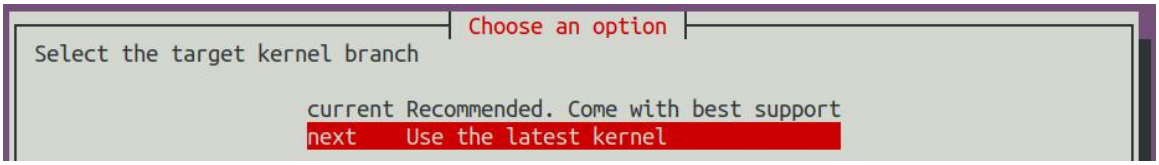


3) Then select the model of the development board

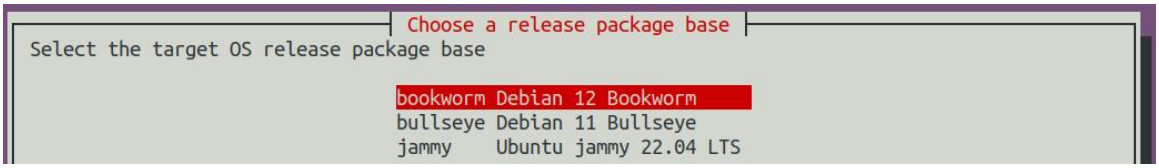


4) Then select the branch type of the kernel source code, the type of rootfs maintained by different versions of the kernel source code is different

a. The current branch can see debian11, ubuntu20.04, ubuntu22.04 three options
b. The next branch can see debian11, debian12, ubuntu22.04 three options

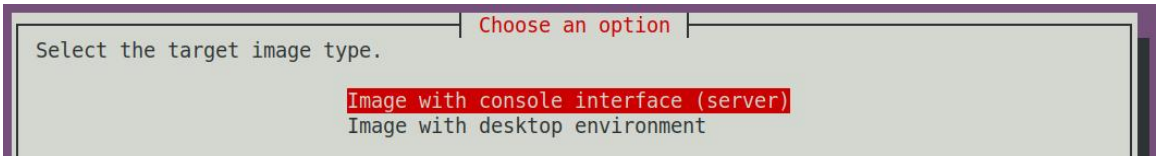


5) Then select the type of rootfs



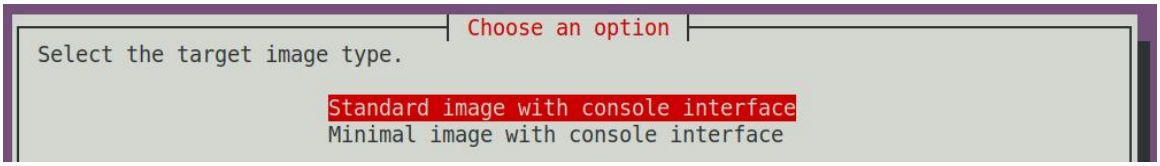
6) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small
b. **Image with desktop environment** Indicates a mirror image with a desktop, which is relatively large

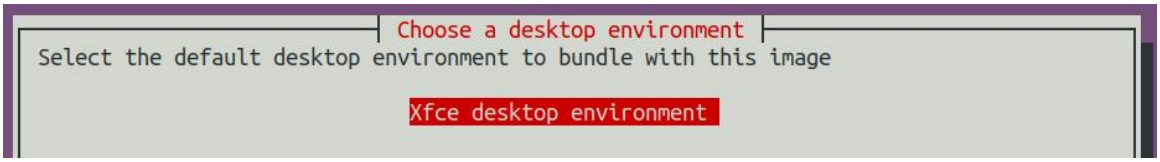


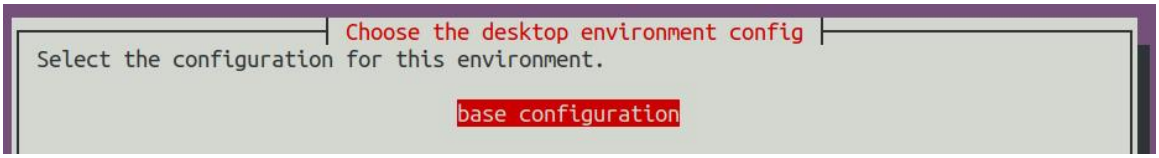
7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the**

Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)

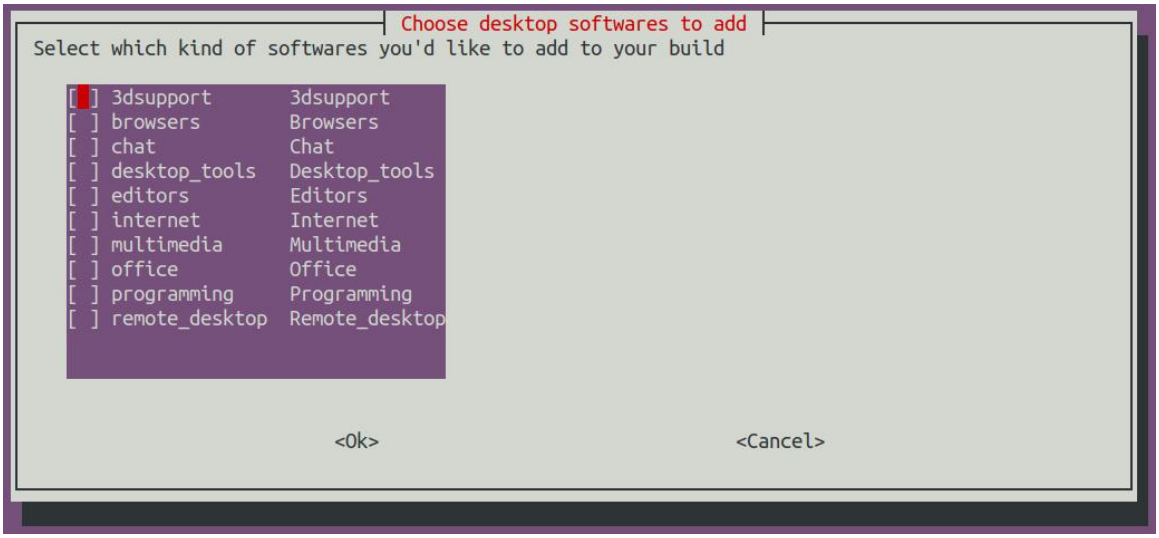


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently only XFCE is maintained, so please choose the XFCE type desktop





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



9) Then it will start to compile rootfs, and some information prompted during compilation is explained as follows

a. The type of rootfs

```
[ o.k. ] local not found [ Creating new rootfs cache for bullseye ]
```

b. The storage path of the compiled rootfs compressed package

```
[ o.k. ] Target directory [ orangepi-build/external/cache/rootfs ]
```

c. The name of the rootfs compressed package generated by compilation

```
[ o.k. ] File name [ bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4 ]
```

10) View the rootfs compressed package generated by compilation

a. **bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4** is the rootfs compressed package, the meaning of each field of the name is

a) **bullseye** indicates the type of linux distribution of rootfs

b) **xfce** means rootfs is the type of desktop version, if it is **cli**, it means the type of server version

c) **arm64** represents the architecture type of rootfs

d) **25250ec7002de9e81a41de169f1f89721** is the MD5 hash value generated by the package names of all software packages installed by rootfs. As long as the list of software packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to generate Determine whether rootfs needs to be recompiled

b. **bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.current
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list
```

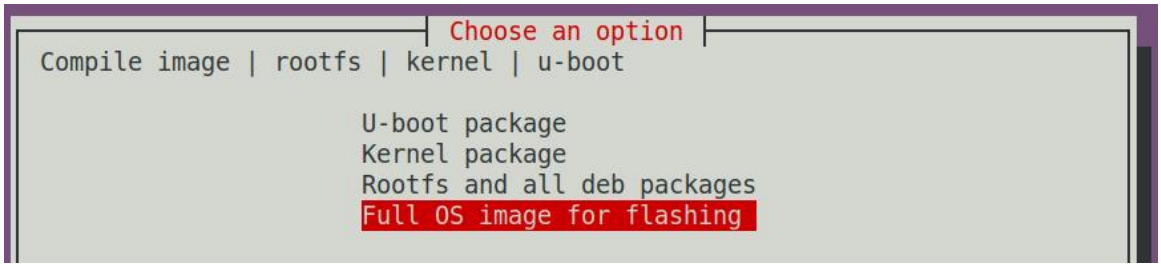
11) If the required rootfs already exists under **external/cache/rootfs**, then compiling rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has If there is rootfs available in the cache, use it directly, which can save a lot of download and compilation time

4. 6. Compile linux image

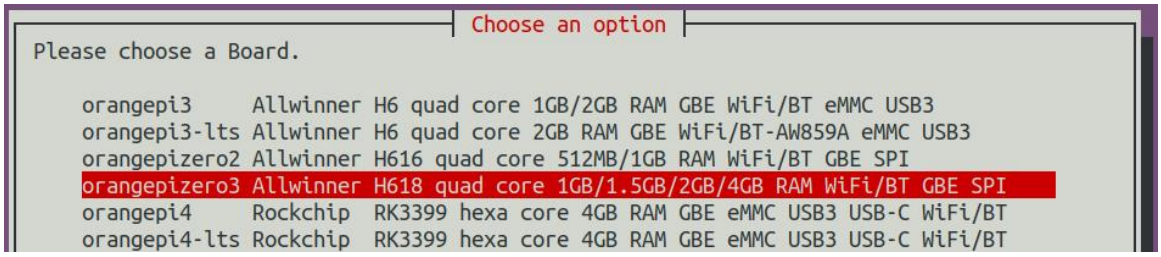
1) Run the **build.sh** script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Full OS image for flashing**, then press Enter

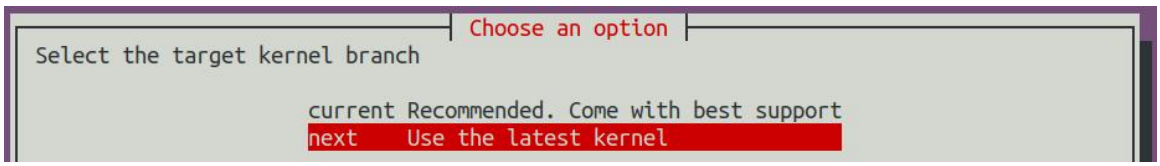


3) Then select the model of the development board

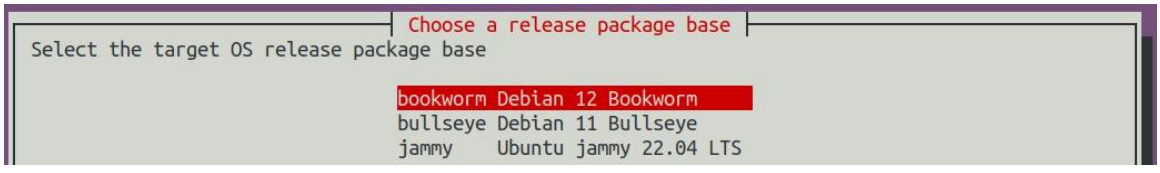


4) Then select the branch type of the kernel source code, the type of rootfs maintained by different versions of the kernel source code is different

a. The current branch can see debian11, ubuntu20.04, ubuntu22.04 three options
b. The next branch can see debian11, debian12, ubuntu22.04 three options



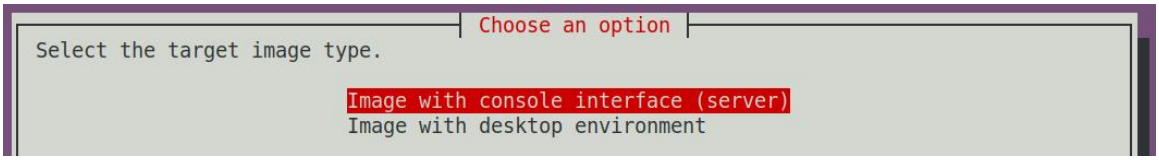
5) Then select the type of rootfs



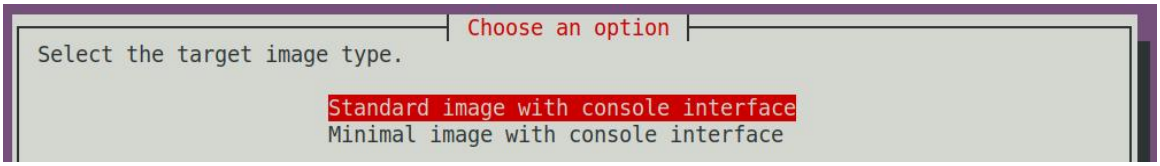
6) Then select the type of image

c. **Image with console interface (server)** Indicates the image of the server version, which is relatively small

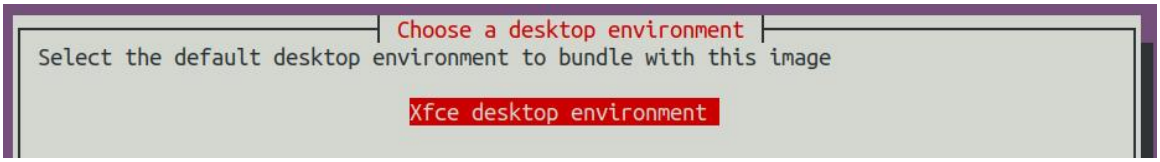
d. **Image with desktop environment** Indicates a mirror image with a desktop, which is relatively large

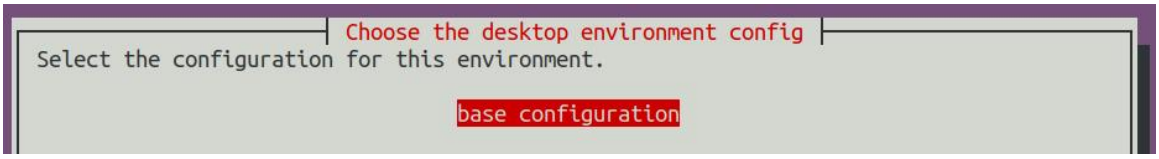


7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**

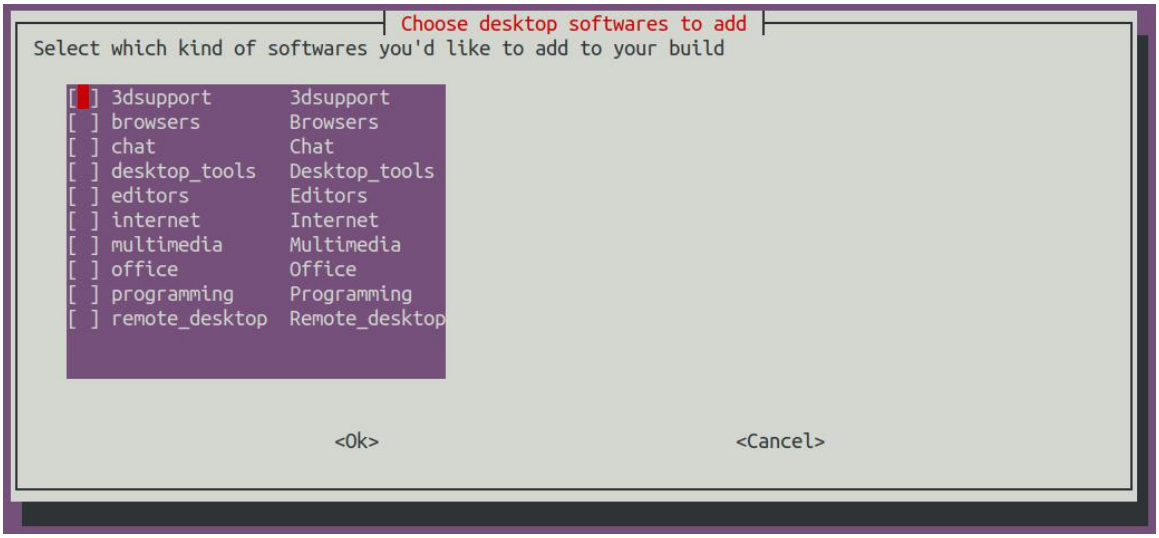


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently only XFCE is maintained, so please select the XFCE type desktop





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



9) Then it will start to compile the linux image. The general process of compilation is as follows

a. Initialize the compilation environment of Ubuntu PC and install the software packages required for the compilation process

b. Download the source code of u-boot and linux kernel (if cached, only update the code)

c. Compile u-boot source code and generate u-boot deb package

d. Compile the linux source code and generate linux-related deb packages

e. Make the deb package of linux firmware

f. Make the deb package of the orangepi-config tool

g. Create a deb package supported by the board

h. If you are compiling the desktop image, you will also create desktop-related deb packages

i. Check whether the rootfs has been cached, if not, recreate the rootfs, if it has been cached, directly decompress and use

j. Install the previously generated deb package into rootfs

k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.

l. Then make an image file and format the partition, the default type is ext4

m. Then copy the configured rootfs to the mirrored partition

n. Then update initramfs

o. Finally, write the bin file of u-boot into the image through the dd command

10) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

[o.k.] Done building
[**output/images/orangepizero3_x.x.x_debian_bullseye_linux6.1.xx_xfce_desktop/orangepizero3_x.x.x_debian_bullseye_linux6.1.xx_xfce_desktop.img**]

b. Compilation time

[o.k.] Runtime [19 min]

c. Repeat the command to compile the image, and use the following command to start compiling the image directly without selecting through the graphical interface

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepizero3 BRANCH=next BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes**]

5. Android 12 TV system instructions

5. 1. Supported Android versions

Android version	kernel version
Android 12 TV version	**linux5.4**

5. 2. Android 12 TV function adaptation

Function	Android12
HDMI Video	OK
HDMI Audio	OK
USB2.0 x 3	OK
TFcard boot	OK
network card	OK
infrared	OK
WIFI	OK
WIFI hotsport	OK
Bluetooth	OK
BLE Bluetooth	OK
headphone audio	OK
TV-OUT	OK
USB camera	OK
LED light	OK
Temperature Sensor	OK
Mali GPU	OK
video codec	OK

5. 3. Onboard LED light display instructions

	Green Light	**Red Light**
u-boot startup phase	**Off**	**Bright**

The kernel boots into the system	Bright	Off

5. 4. Android method to return to the previous interface

1) We generally use the mouse and keyboard to control the Android system of the development board. When entering certain interfaces and need to return to the previous interface or desktop, we can only return through the **right mouse button**, and the keyboard cannot return.

2) If you have purchased the infrared remote control (other remote control does not work) and the expansion board that match the development board, after inserting the expansion board into the development board, you can return to the previous menu through the return key in the remote control. The position of the return key is shown in the figure below shown:

5. 5. How to use ADB

5. 5. 1. Use network connection adb debugging

Using the network adb does not require a USB Type C interface data cable to connect the computer and the development board, but to communicate through the network, so first make sure that the wired or wireless network of the development board has been connected, and then obtain the IP address of the development board, and then to use.

1) Make sure that the **service.adb.tcp.port** of the Android system is set to port number 5555

```
apollo-p2:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the port number of network adb in the serial port

```
apollo-p2:/ # setprop service.adb.tcp.port 5555
apollo-p2:/ # stop adbd
apollo-p2:/ # start adbd
```

3) Install adb tool on Ubuntu PC

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y adb
```

4) Then connect network adb on Ubuntu PC

```
test@test:~$ adb connect 192.168.1.xxx:5555      (It needs to be modified to the IP address of the development board)
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xxx:5555

test@test:~$ adb devices
List of devices attached
192.168.1.xxx:5555          device
```

5) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell
apollo-p2:/ #
```

5.5.2. Use the data cable to connect to adb debugging

1) Prepare a data cable with USB Type C interface, plug one end of the USB interface into the USB interface of the computer, and insert one end of the USB Type C interface into the power interface of the development board. In this case, the development board is powered by the USB interface of the computer, so please make sure that the USB interface of the computer can provide the most power to drive the development board

2) Install adb tool on Ubuntu PC

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y adb
```

3) View the identified ADB device

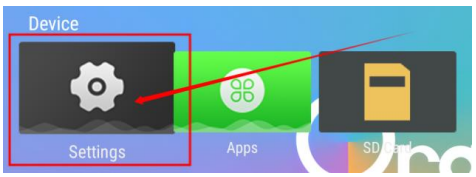
```
test@test:~$ adb devices
List of devices attached
4c00146473c28651dd0     device
```

4) Then you can log in to the android system through the adb shell on the Ubuntu PC

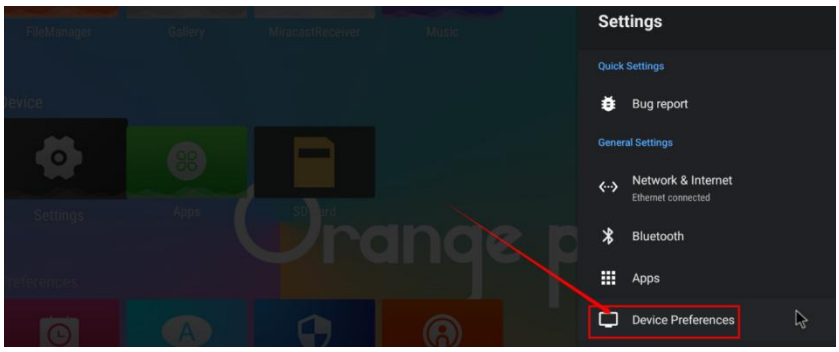
```
test@test:~$ adb shell
apollo-p2:/ $
```

5. 6. View the method of setting HDMI display resolution

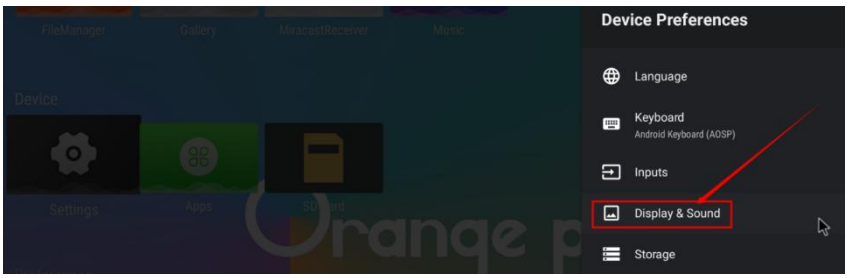
1) First enter **Settings**



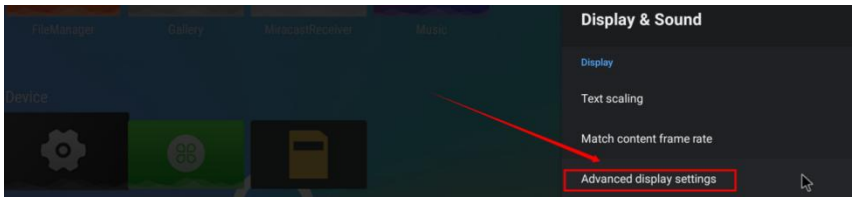
2) Then select **Device Preferences**



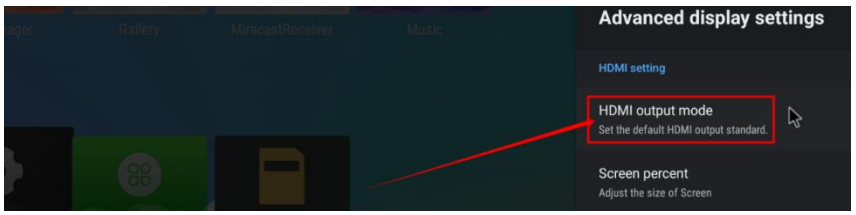
3) Then select **Display & Sound**



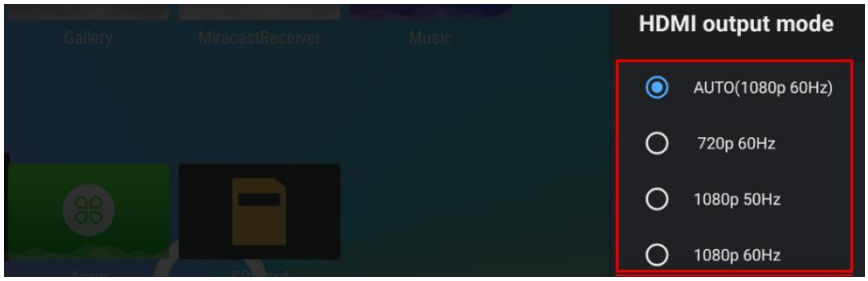
4) Then select **Advanced display settings**



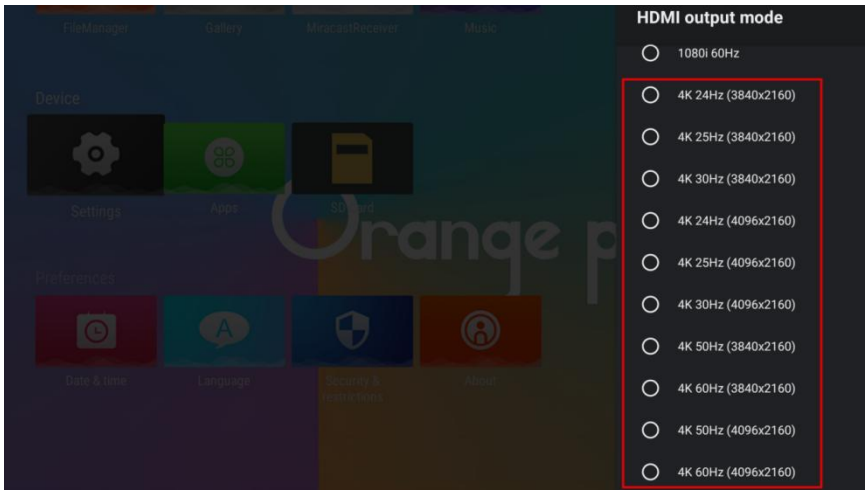
5) Then select **HDMI outpu mode**



6) Then you can see the list of resolutions supported by the display. At this time, click the corresponding option to switch to the corresponding resolution. Please note that the resolutions supported by different monitors may be different. If you connect to a TV, you will generally see more resolution options than the picture below.



7) The HDMI output of the development board supports 4K display. When connected to a 4K TV, you can see the option of 4K resolution



5. 7. HDMI to VGA display test

1) First, you need to prepare the following accessories

a. HDMI to VGA Converter

b. A VGA cable and a Micro HDMI male to HDMI female conversion cable

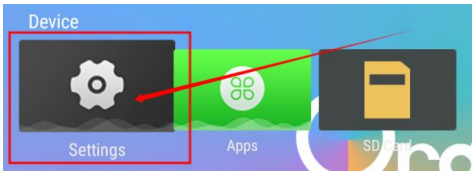
c. A monitor or TV that supports VGA interface

2) HDMI to VGA display test as shown below

When using HDMI to VGA display, the development board and the Android system of the development board do not need to make any settings, only the Micro HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI to VGA converter, VGA cable and monitor.

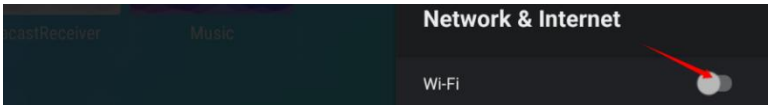
5. 8. WI-FI connection method

1) First select **Settings**

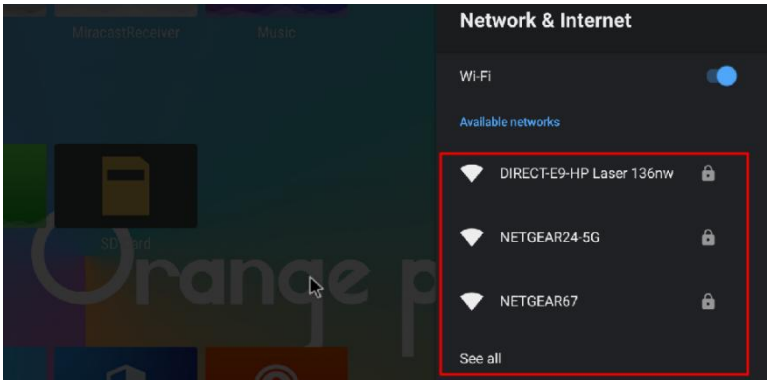


2) Then select **Network & Internet**

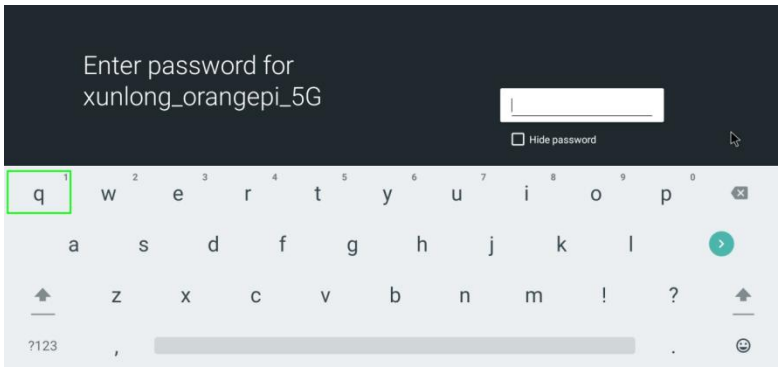
3) Then open WI-FI



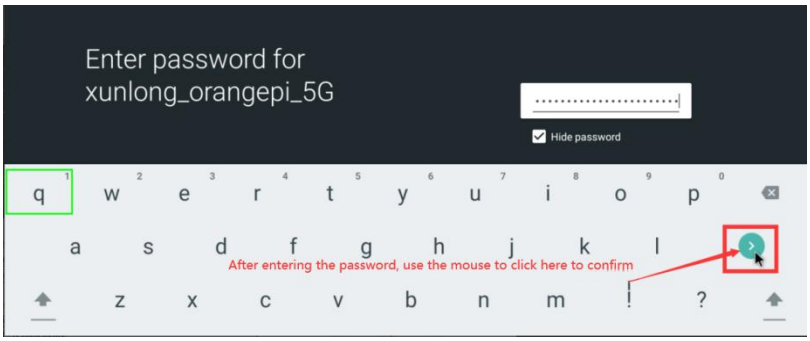
4) After turning on WI-FI, you can see the searched signal under **Available networks**



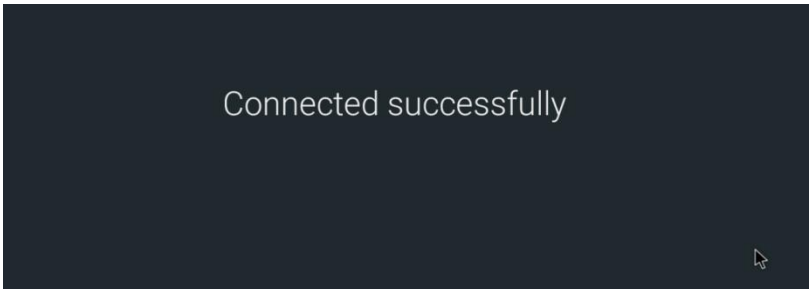
5) After selecting the WI-FI you want to connect to, the password input interface shown in the figure below will pop up



6) Then use the keyboard to enter the password corresponding to the WI-FI, and then use the **mouse** to click the Enter button in the virtual keyboard to start connecting to the WI-FI



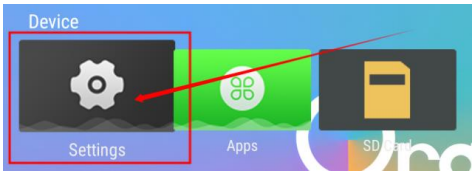
7) After the WI-FI connection is successful, the display is as shown in the figure below



5. 9. How to use WI-FI hotspot

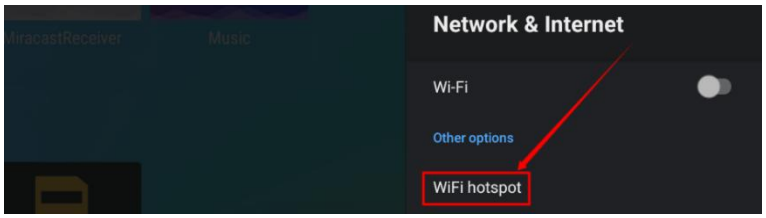
1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally

2) Then select **Settings**

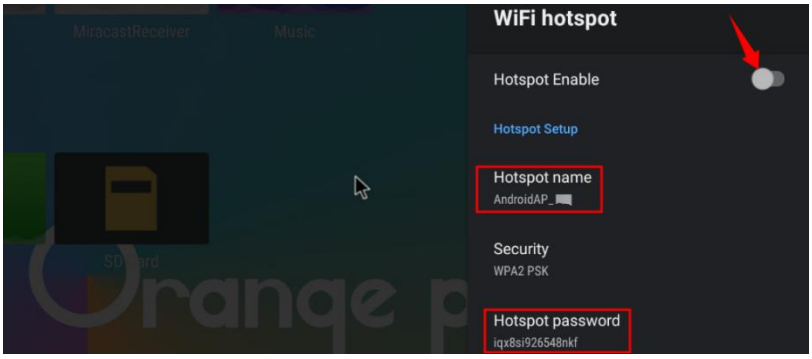


3) Then select **Network & Internet**

4) Then select **WIFI hotspot**

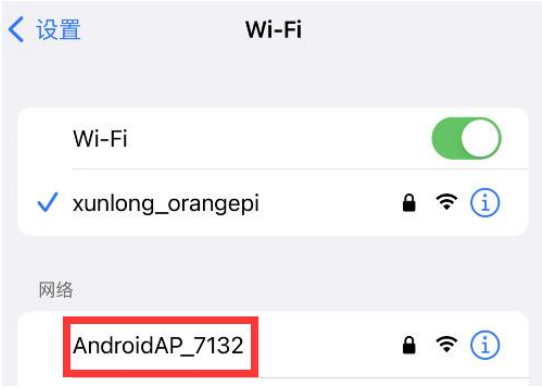


5) Then turn on **Hotspot Enable**, you can also see the name and password of the generated hotspot in the figure below, remember them and use them when connecting to the hotspot (if you need to modify the name and password of the hotspot, you need to turn off **Hotspot Enable** first, before modifying)

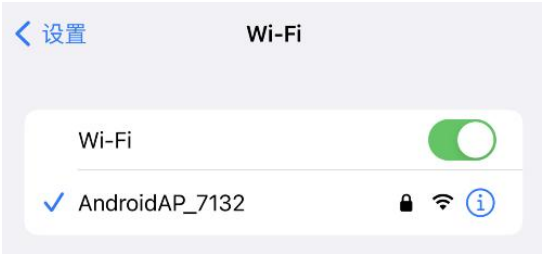


6) At this point, you can take out your mobile phone. If everything is normal, you can find the WIFI hotspot with the same name **(here AndroidAP_7132)** displayed under the **Hotspot name** in the above picture in the WI-FI list searched by the mobile phone. Then you can click **AndroidAP_7132** to connect to the hotspot, and the password can be seen

under the **Hotspot password** in the above picture



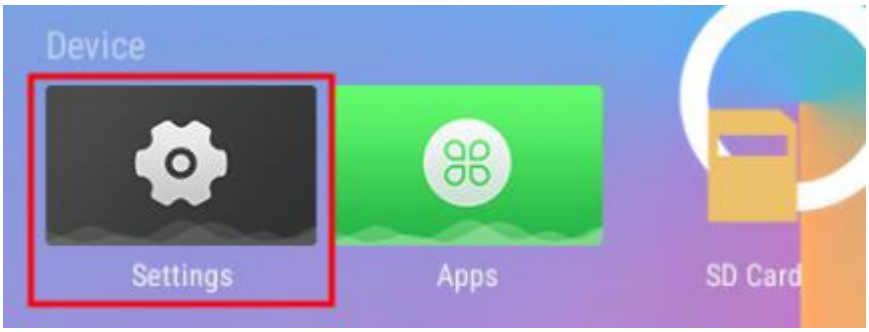
7) After the connection is successful, it will be displayed as shown in the figure below (the interface of different mobile phones will be different, the specific interface is subject to the display of your mobile phone). At this point, you can open a webpage on your mobile phone to see if you can access the Internet. If you can open the webpage normally, it means that the **WI-FI Hotspot** of the development board can be used normally.



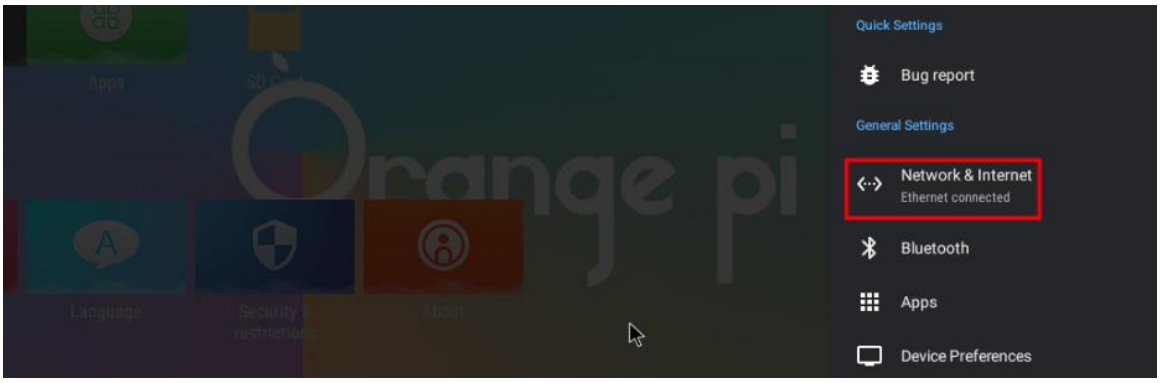
5. 10. How to check the IP address of the Ethernet port

1) First, please make sure that the Gigabit Ethernet port of the development board is connected to a router or switch

2) Then open **Settings**



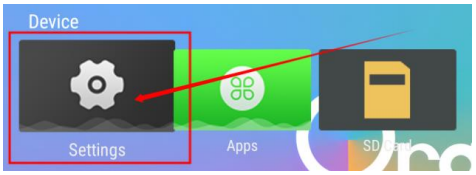
3) Then select **Network & Internet**



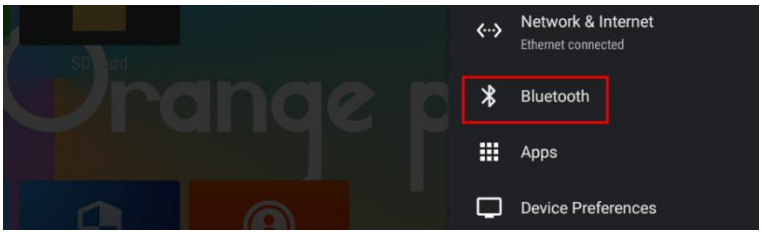
4) Then you can see the IP address of the development board's wired network port at the position shown in the figure below

5. 11. Bluetooth connection method

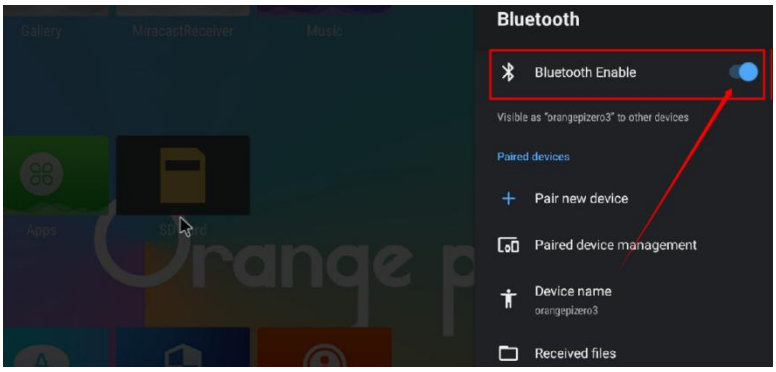
1) First select **Settings**



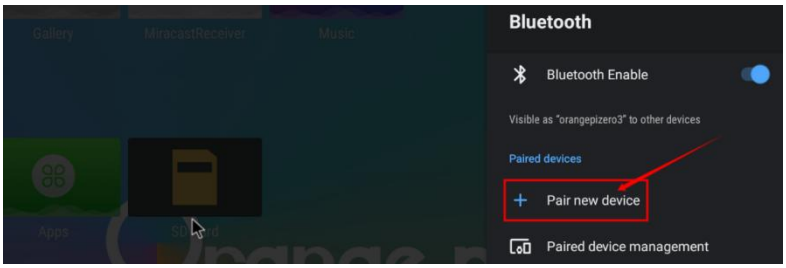
2) Then select **Bluetooth**



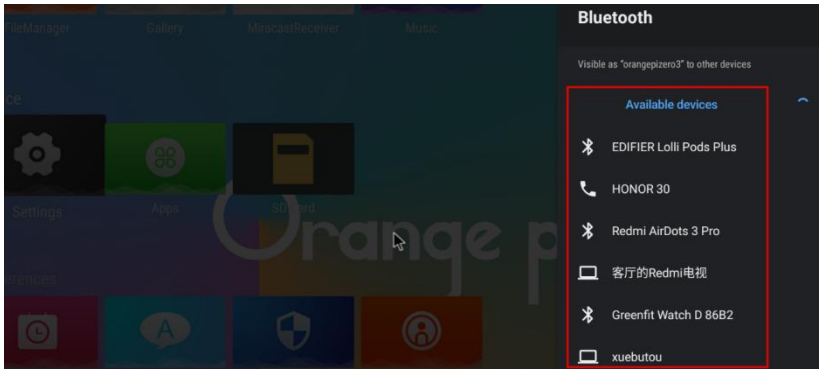
3) Then open **Bluetooth Enable**



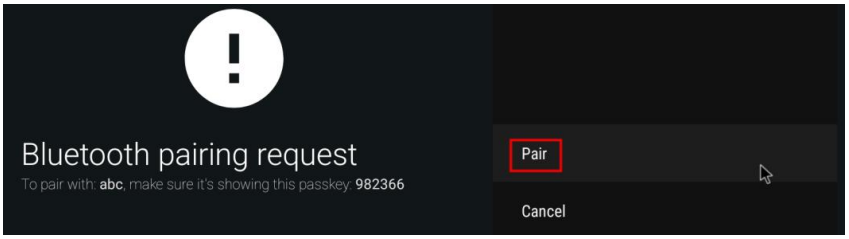
4) Then click **Pair new device** to start scanning the surrounding Bluetooth devices



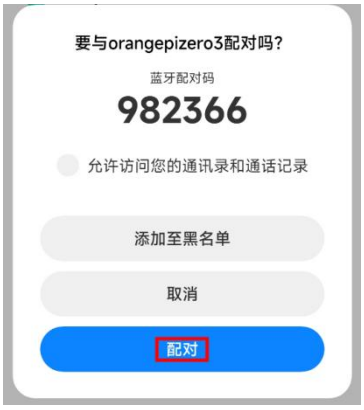
5) The searched Bluetooth devices will be displayed under **Available devices**



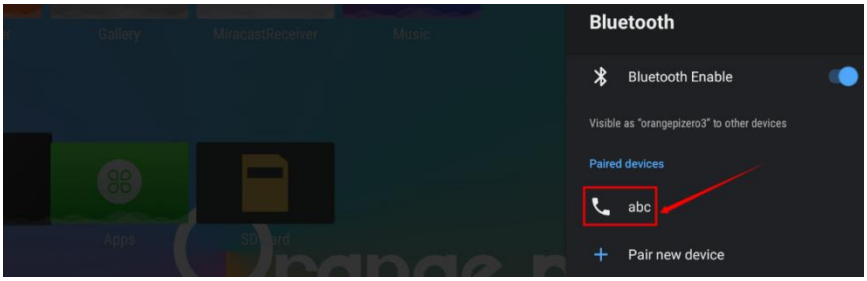
6) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option



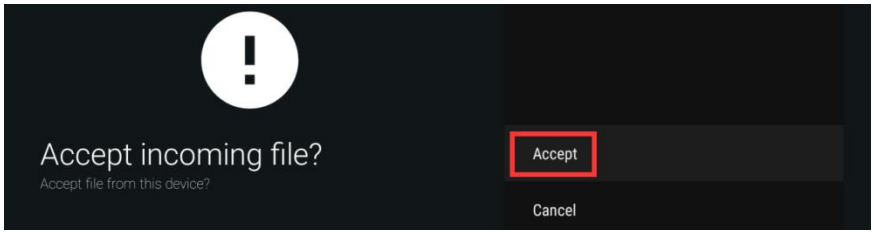
7) The test here is the configuration process of the development board and the Bluetooth of the **Android mobile phone**. At this time, the following confirmation interface will pop up on the mobile phone. After clicking the pairing button on the mobile phone, the pairing process will start



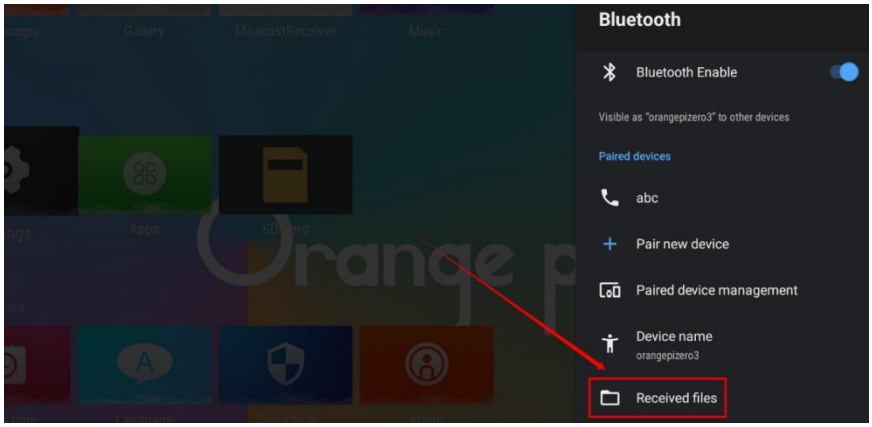
8) After the pairing is complete, open **Paired devices** and you can see the paired Bluetooth devices



9) At this time, you can use the Bluetooth of your mobile phone to send a picture to the development board. After sending, you can see the following confirmation interface in the Android system of the development board, and then click **Accept** to start receiving the picture sent by the mobile phone.



10) The pictures received by the Android system Bluetooth of the development board can be viewed in **Received files**



5. 12. How to use the USB camera

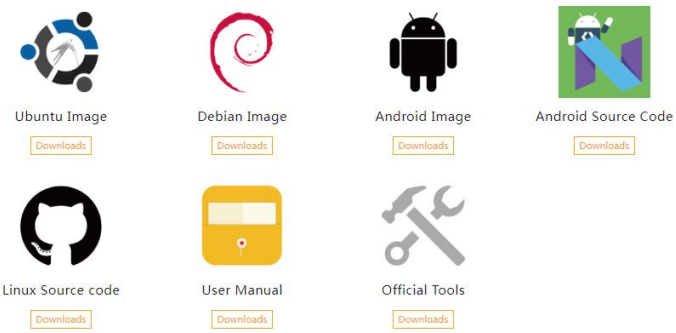
1) First insert the USB (UVC protocol) camera into the USB interface of the development board

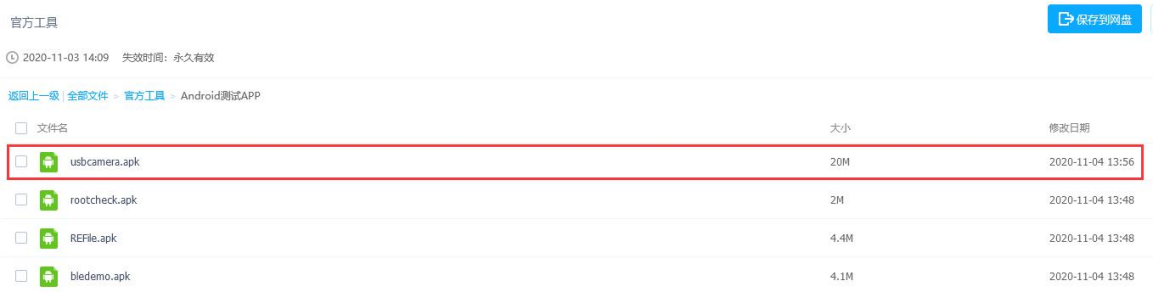
2) If the USB camera is recognized normally, a corresponding video device node will be generated under /dev

```
console:/ # ls /dev/video0
/dev/video0
```

3) Then make sure that the adb connection between the Ubuntu PC and the development board is normal. For how to use adb, please refer to the instructions in the section on **how to use ADB**

4) Download the USB camera test APP in the official tool on the development board data download page

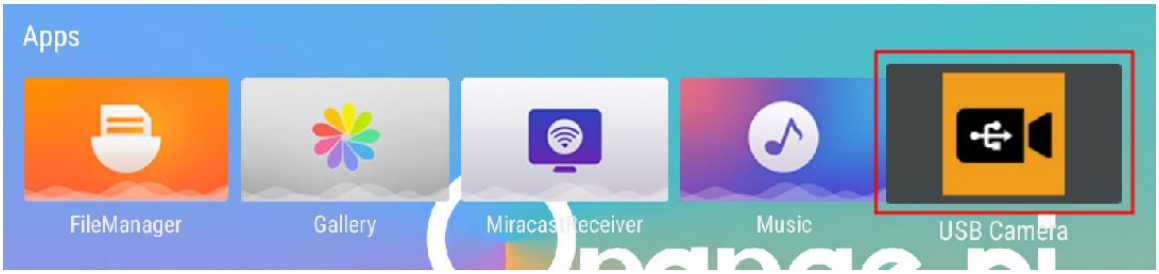




5) Then use the adb command to install the USB camera test APP to the Android system, of course, you can also use the U disk to copy the installation

```
test@test:~$ adb install usbcamera.apk
```

6) After installation, you can see the startup icon of the USB camera on the Android desktop



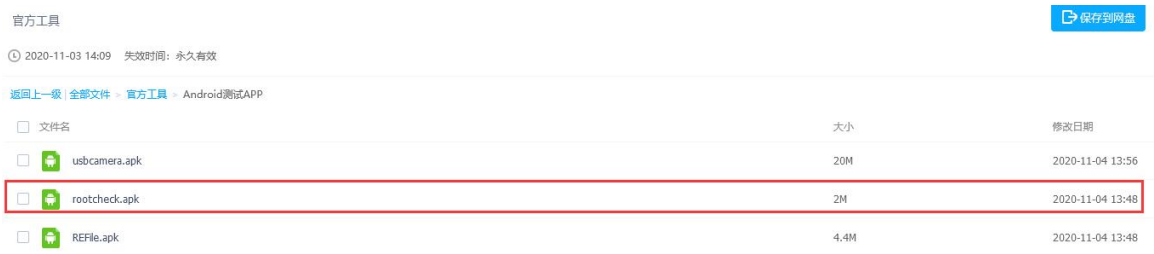
7) Then double-click to open the USB camera APP and you can see the output video of the USB camera

5. 13. Android system ROOT instructions

The Android system released by Orange Pi has been ROOT, you can use the

following method to test.

1) Download **rootcheck.apk** from the official tool on the development board data download page

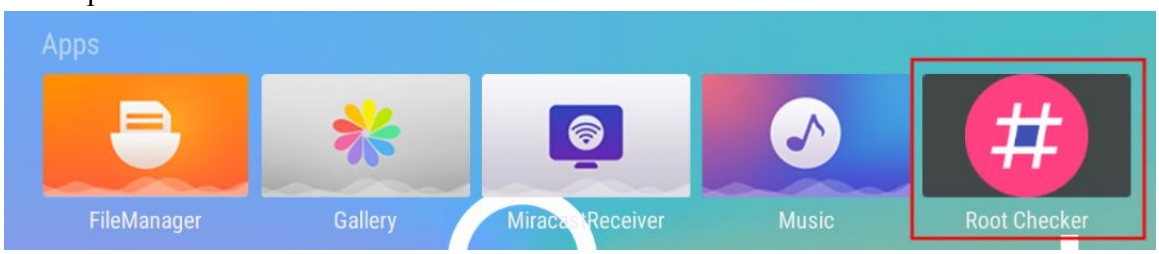


2) Then make sure that the adb connection between the Ubuntu PC and the development board is normal. For how to use adb, please refer to the instructions in the section on **how to use ADB**

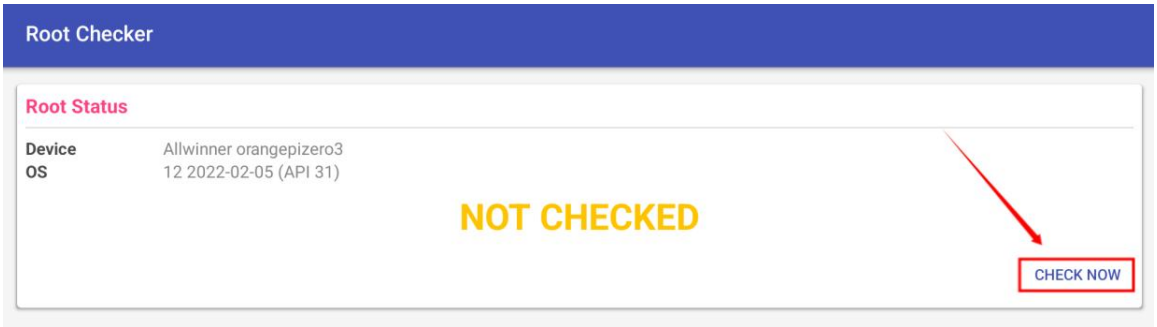
3) Then use the adb command to install rootcheck.apk to the Android system, of course, you can also use the U disk to copy the installation

```
test@test:~$ adb install rootcheck.apk
```

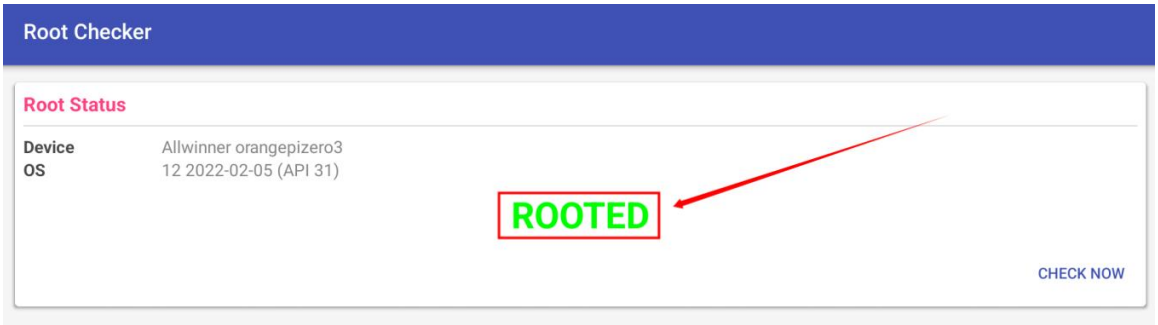
4) After installation, you can see the startup icon of the ROOT test tool on the Android desktop



5) The display interface after opening the **ROOT test tool** for the first time is shown in the figure below



6) Then you can click **CHECK NOW** to start checking the ROOT status of the Android system. After the check, the display is as follows, and you can see that the Android system has obtained the ROOT permission

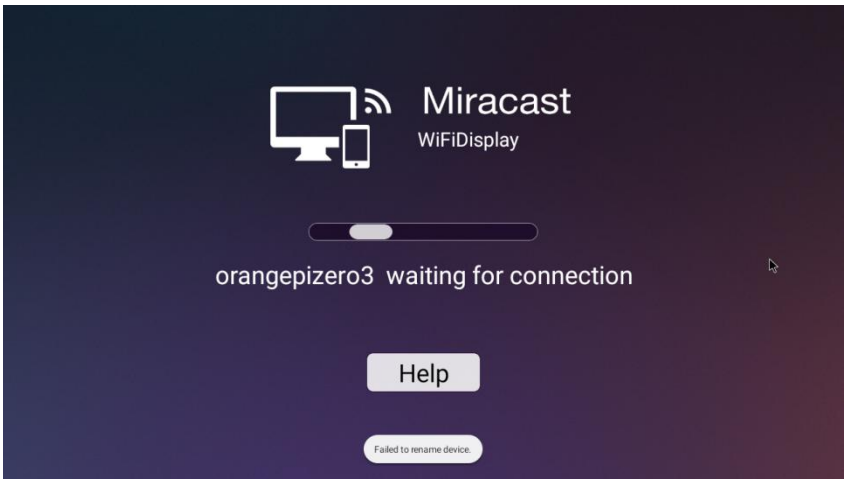


5. 14. The method of using MiracastReceiver to cast the mobile phone screen to the development board

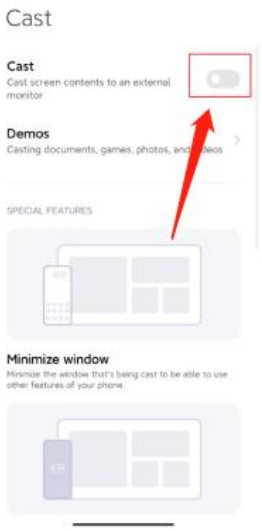
1) First, please make sure that the development board and the mobile phone are connected to the same WIFI hotspot. For the method of connecting the development board to WIFI, please refer to the description in the section of **WI-FI connection method**

2) Then open the **MiracastReceiver** application in the Android system of the development board

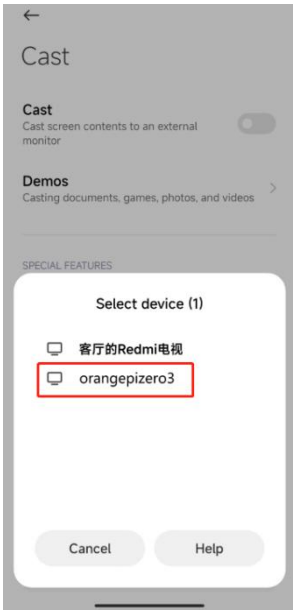
3) The interface after **MiracastReceiver** is opened is as follows



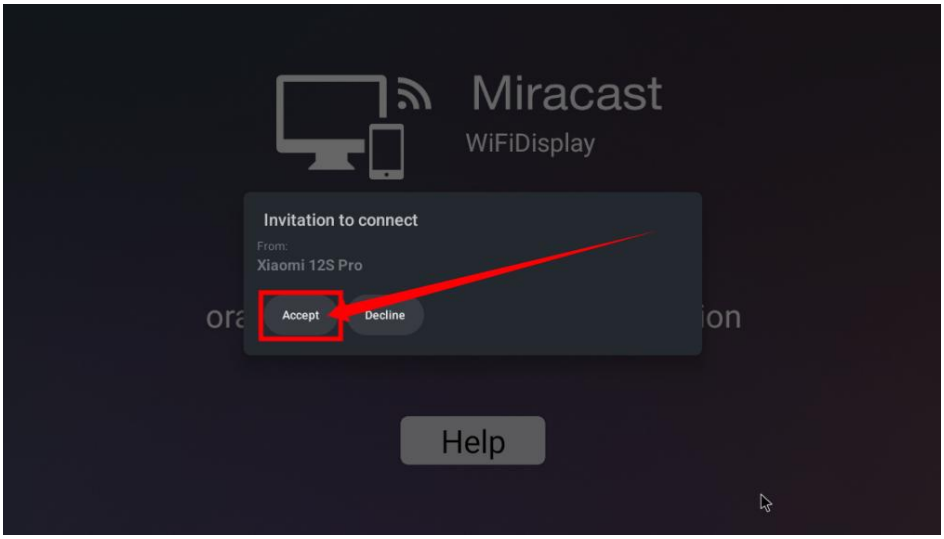
4) Then find the screen projection function in the mobile phone settings. **Here we take the Mi 12S Pro mobile phone as an example**. Please research other brands of mobile phones by yourself. As shown in the figure below, click the button in the red box to open the screen projection function of the mobile phone



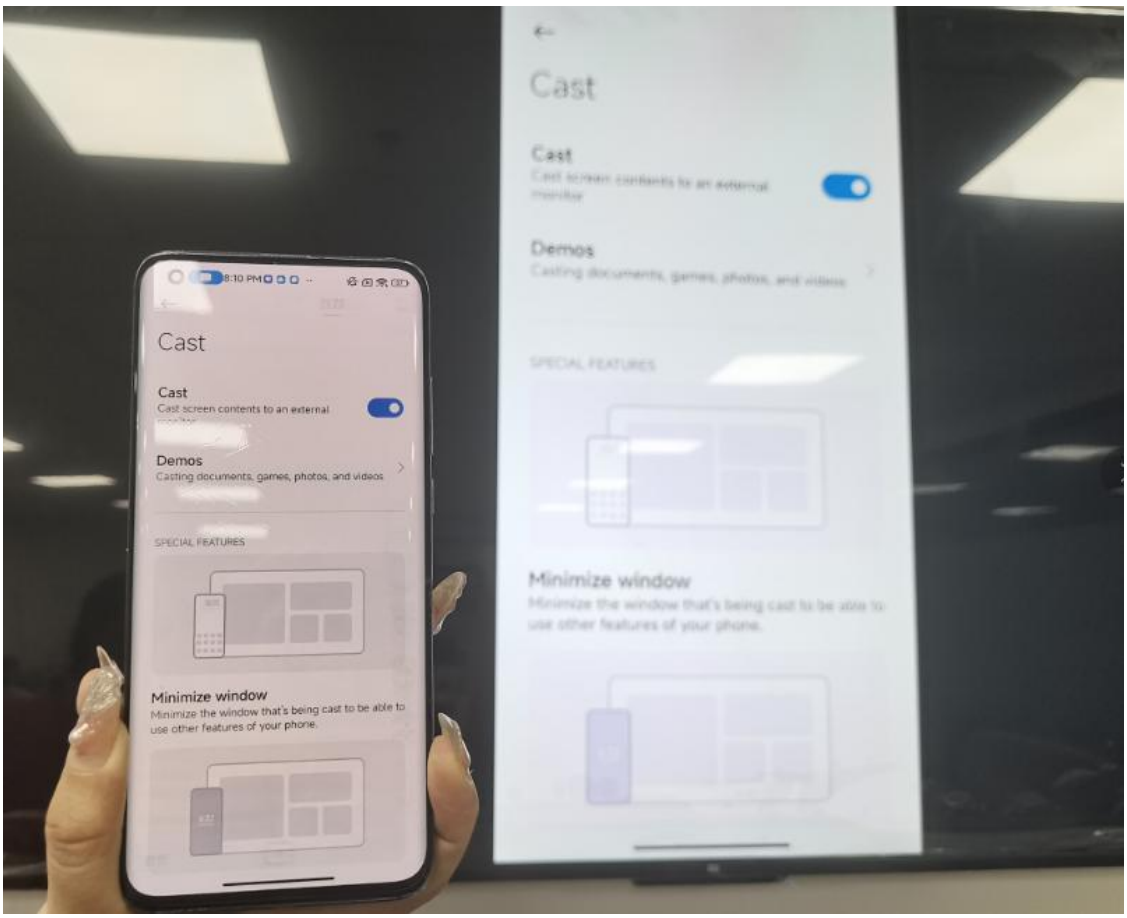
5) After waiting for a while, you can see the searched and connectable devices on the mobile phone, and then we can select the device corresponding to the development board to connect



6) Then the selection box shown in the figure below will pop up on the **MiracastReceiver** application interface of the development board, here we can select **Accept**



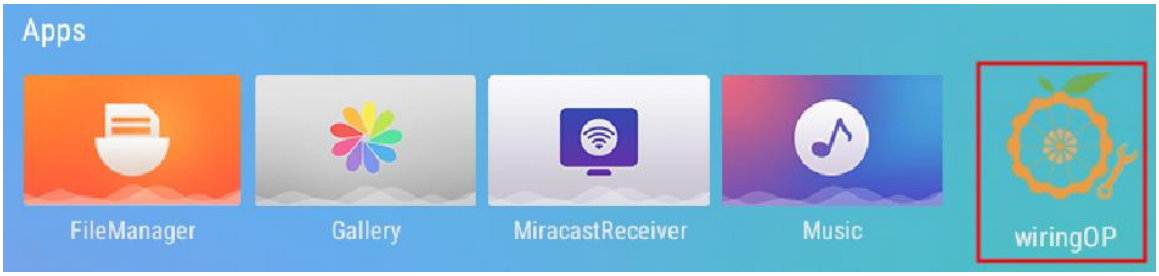
7) Then you can see the content of the mobile phone screen on the HDMI screen connected to the development board



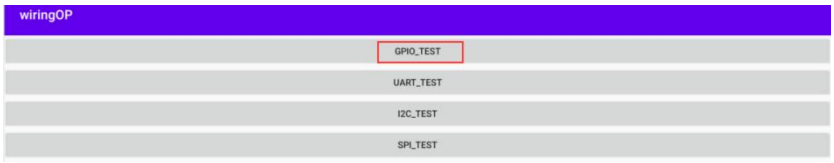
5. 15. 26pin interface GPIO, UART, SPI test

5. 15. 1. 26pin GPIO port test method

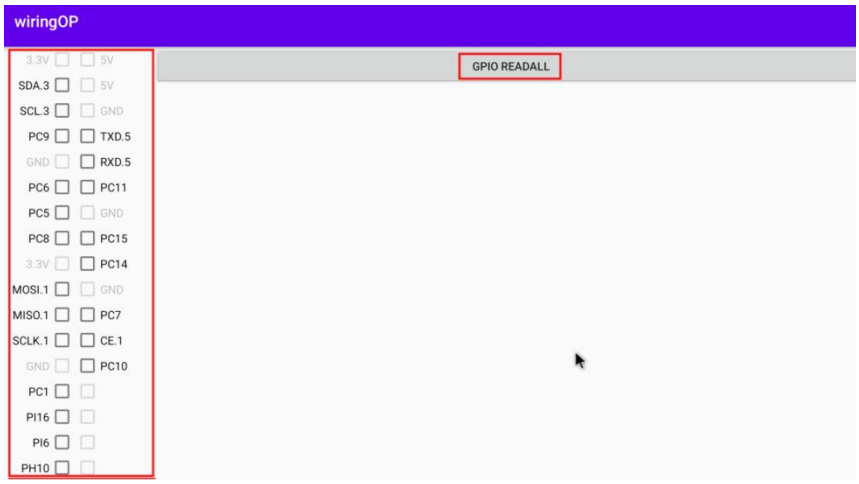
1) First open the wiringOP APP on the desktop



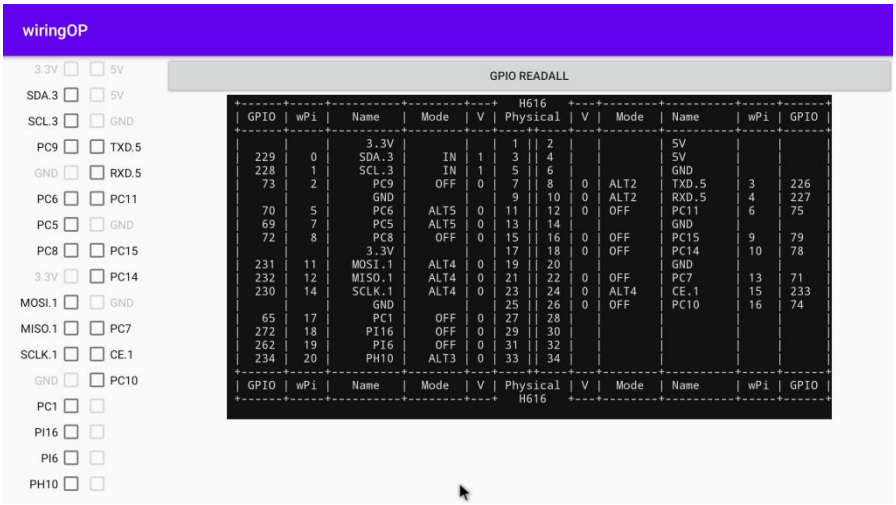
2) Then click the **GPIO_TEST** button to open the GPIO test interface



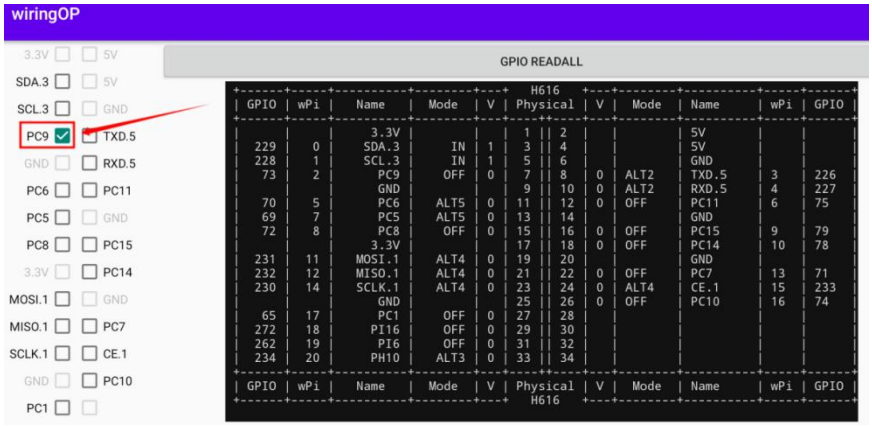
3) The GPIO test interface is shown in the figure below. There is a one-to-one correspondence between the two rows of **CheckBox** buttons on the left and the 26pin pins **(PC1/PI16/PI6/PH10 in the lower left corner are GPIO ports in the 13pin pins)**. When the **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT** mode, and the pin level will be set to high level; when the checkbox is unchecked, the GPIO pin level will be set to low level; When the **GPIO READALL** button is pressed, information such as the wPi number, GPIO mode, and pin level can be obtained.



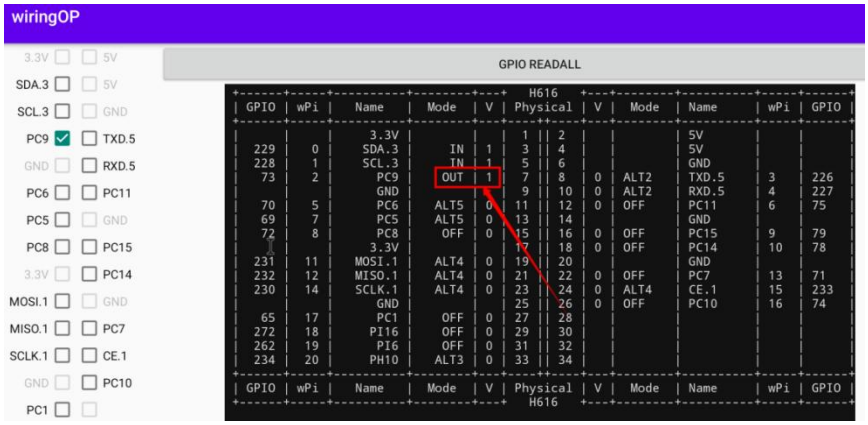
4) Then click the **GPIO READALL** button, the output information is as shown in the figure below：



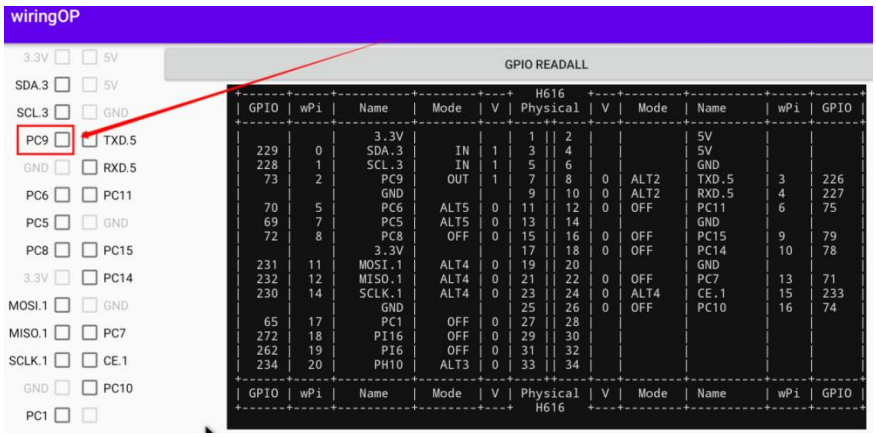
5) There are a total of 17 GPIO ports in the 26pin of the development board that can be used **(if you add 4 GPIO ports in the 13pin, then there are 21 in total)**, and the following pin No. 7 - corresponding to GPIO is PC9 - corresponding to wPi The serial number is 2——as an example to demonstrate how to set the high and low levels of the GPIO port. First click the **CheckBox** button corresponding to pin 7. When the button is selected, pin 7 will be set to high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **3.3v**, it means setting high level success



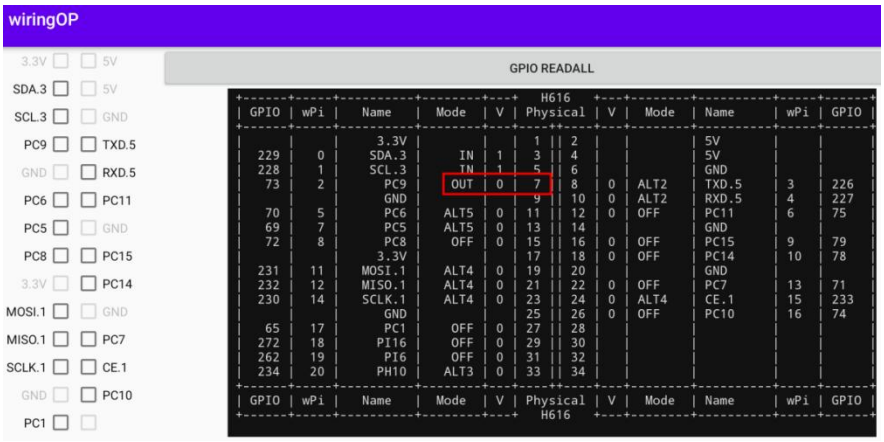
6) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is **OUT**, and the pin level is high



7) Click the **CheckBox** button in the figure below again to cancel the check status. Pin 7 will be set to low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level is set successfully.



8) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is OUT, and the pin level is low

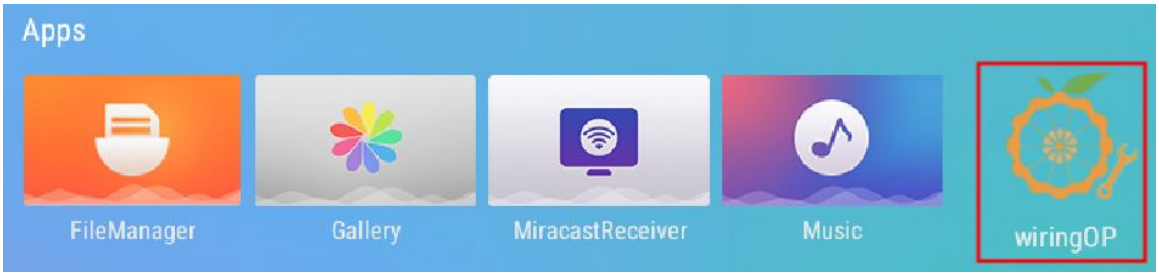


5. 15. 2. 26pin UART test method

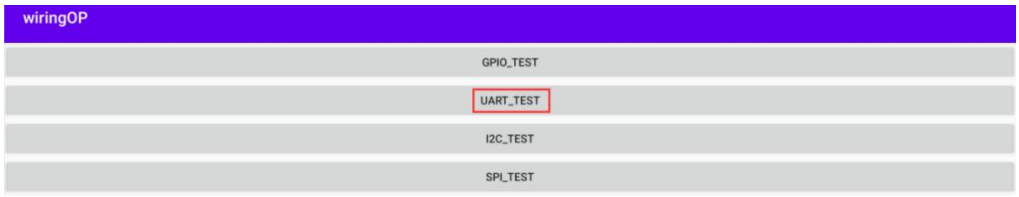
1) **UART5** is enabled by default in Android, and the corresponding device node is **/dev/ttyAS5**

```
apollo-p2:/ # ls /dev/ttyAS5
/dev/ttyAS5
```

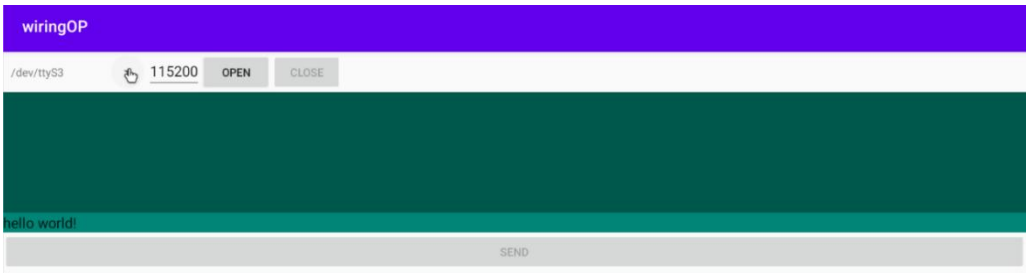
2) First open the wiringOP APP on the desktop



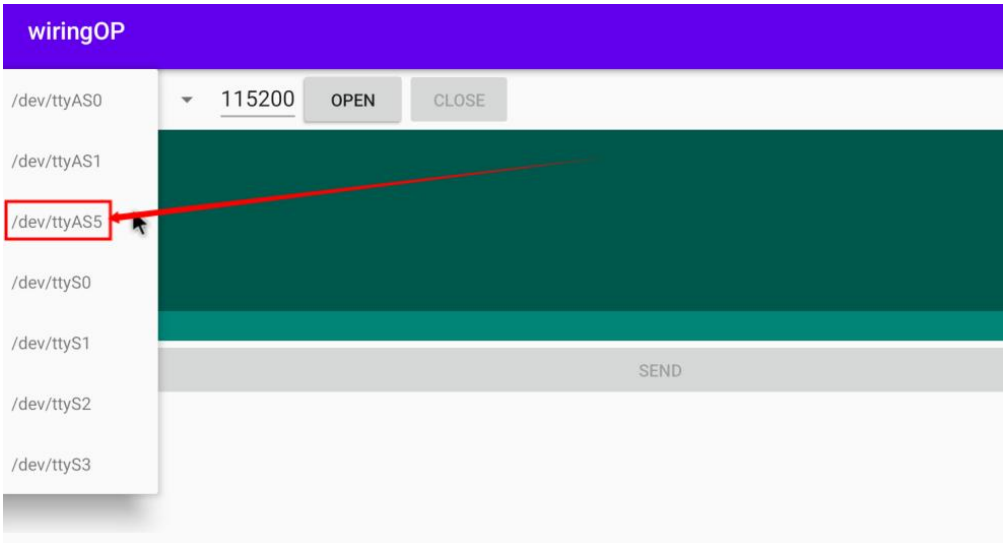
3) Then click the **UART_TEST** button to open the UART test interface



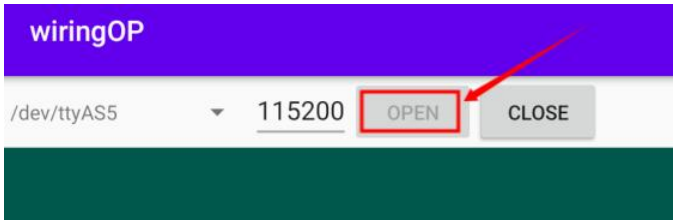
4) The serial port test interface of wiringOP is shown in the figure below



5) Then select the **/dev/ttyAS5** node in the selection box



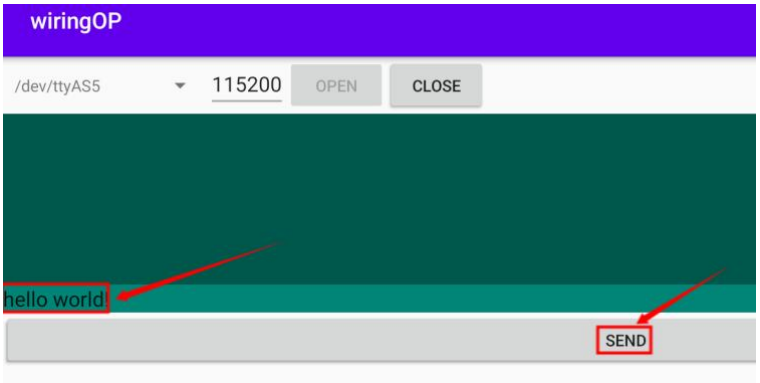
6) Enter the baud rate you want to set in the edit box, and then click the **OPEN** button to open the **/dev/ttyAS5** node. After the opening is successful, the **OPEN** button becomes unselectable, and the **CLOSE** button and **SEND** button become selectable.



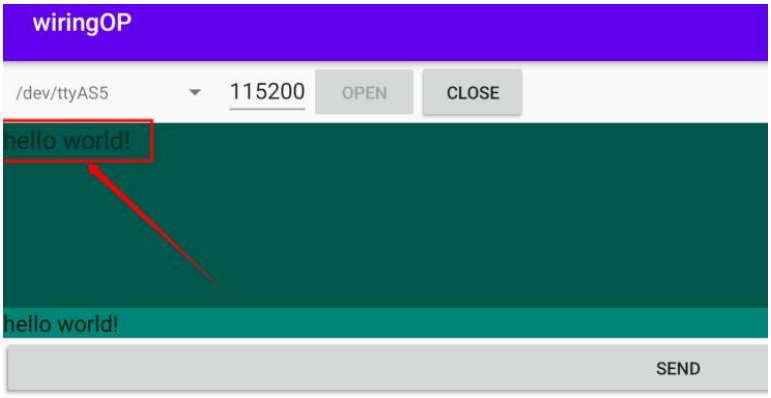
7) Then use Dupont wire to short the rx and tx pins of uart5

	uart5
Tx pin	Corresponding to pin 8 of 26pin
Rx pin	Corresponding to pin 10 of 26pin

8) Then you can enter a character in the send edit box below, click the **SEND** button to start sending

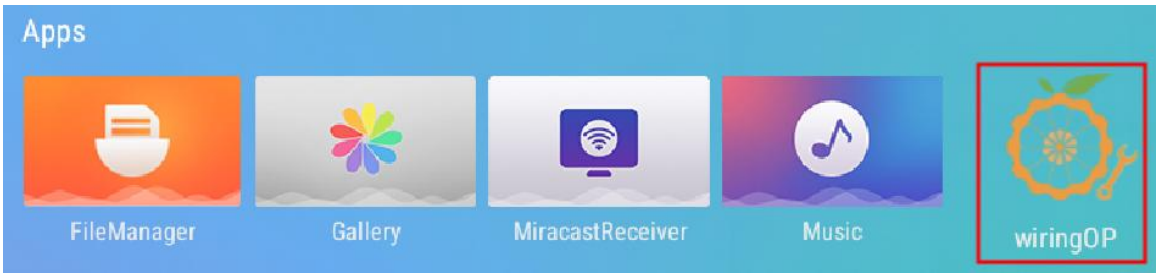


9) If everything is normal, the received string will be displayed in the receiving box



5. 15. 3. 26pin SPI test method

1) The SPI that can be used in 26pin is SPI1, and the corresponding device node is **/dev/spidev1.1**

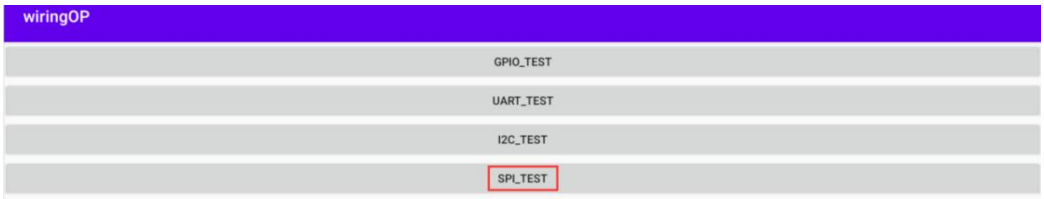


2) Here is a demonstration to test the SPI1 interface through the **w25q64** module, first connect the w25q64 module to the SPI1 interface

It doesn't matter if there is no w25q64 module, because there is a SPIFlash connected to SPI0 on the development board, and the configuration of SPI0 is also enabled by default in Android, so we can also directly use the onboard SPIFlash test.

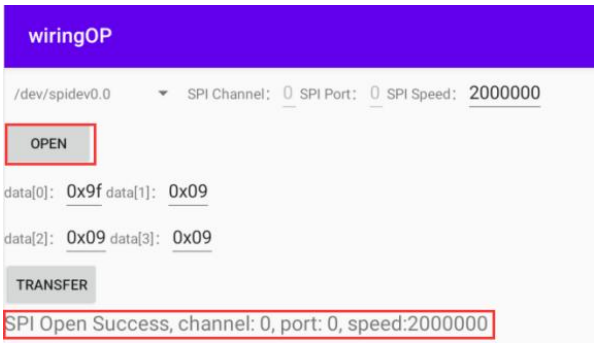
3) Then open the wiringOP APP on the desktop

4) Then click the **SPI_TEST** button to open the SPI test interface



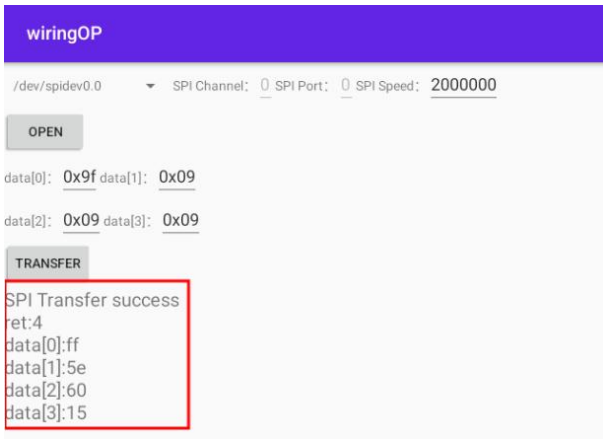
5) Then select the spi device node in the upper left corner. If you directly test the onboard SPIFlash, then keep the default **/dev/spidev0.0**. If you connect the **w25q64** module to the 26pin spi1, then please select /dev **/dev/spidev1.1**

6) Then click the **OPEN** button to initialize the SPI

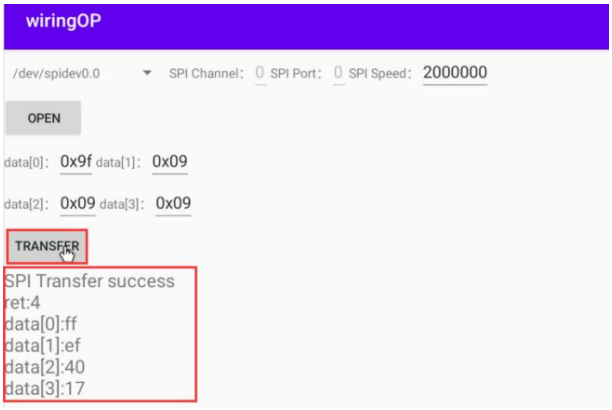


7) Then fill in the bytes that need to be sent, such as reading the ID information of the onboard SPIFlash, fill in the address 0x9f in data[0], and then click the **TRANSFER** button

8) Finally, the APP will display the read ID information of the onboard SPI Flash



9) If it is to read the w25q64 module connected to 26pin SPI1, then the read ID information is shown in the figure below



10) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, corresponding to the value read above (h stands for hexadecimal)

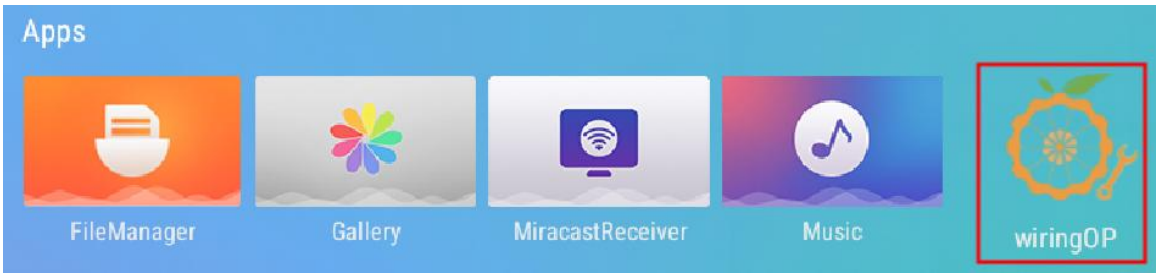
MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

5. 15. 4. 26pin I2C test method

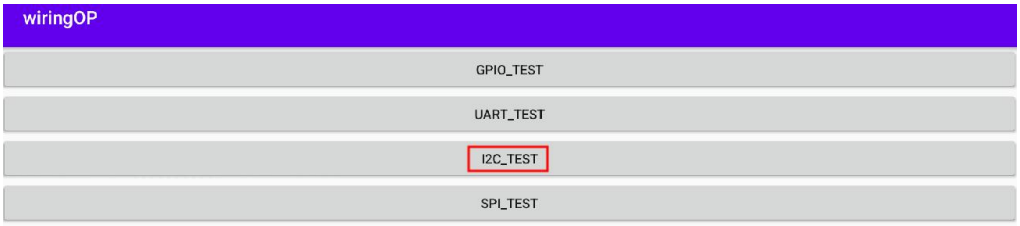
1) The i2c3 in 26pin is enabled by default in Android, and the corresponding device node is **/dev/i2c-3**

```
apollo-p2:/ # ls /dev/i2c-3
/dev/i2c-3
```

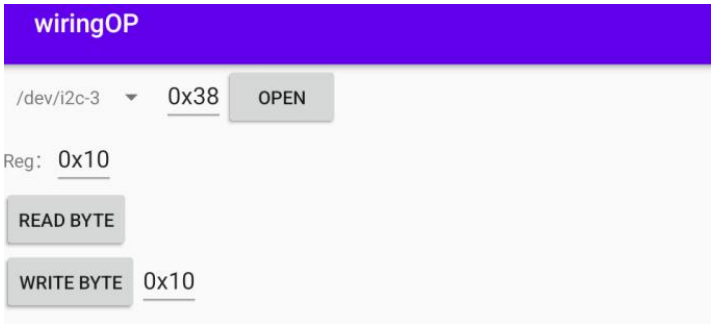
2) First open the wiringOP APP on the desktop



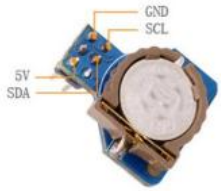
3) Then click the **I2C_TEST** button to open the i2c test interface



4) The i2c test interface of wiringOP is shown in the figure below. You can see that the default i2c is **/dev/i2c-3**, so we don’t need to re-select



5) Then connect an i2c device to the 26pin i2c3 pin, here we take the ds1307 rtc module as an example



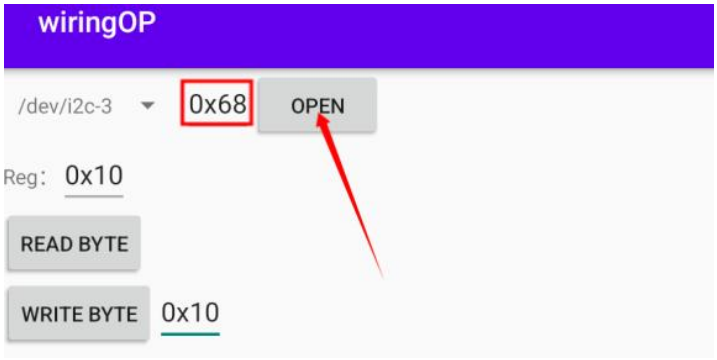
Pins of the RTC module	The pin corresponding to the 26pin of the development board
5V	**pin 2**
GND	**pin 6**
SDA	**pin 3**
SCL	**pin 5**

6) The i2c address of the ds1307 rtc module is 0x68. After connecting the wires, we can use the **i2cdetect -y 3** command in the serial port command line to check whether the i2c address of the ds1307 rtc module can be scanned. As shown in the figure below, if you can see the address 0x68, it means that the wiring of the ds1307 rtc module is correct.

```
apollo-p2:/ # i2cdetect -y 3
```

```
apollo-p2:/ # i2cdetect -y 3
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
apollo-p2:/ #
```

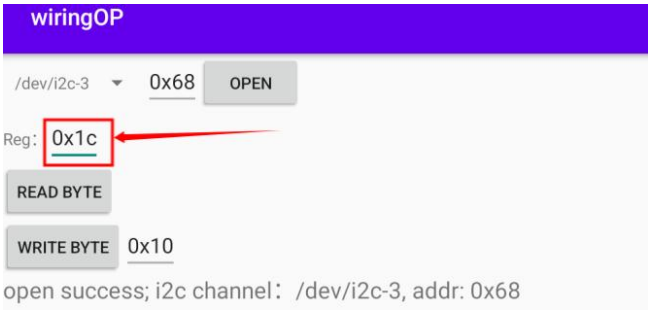
7) Then set the address of i2c to 0x68 in wiringOP, and then click the **OPEN** button to open i2c3



8) The display after clicking the **OPEN** button to open i2c3 is as follows：

9) Then we test to write a value into the register of the rtc module, for example, write 0x55 to the 0x1c address

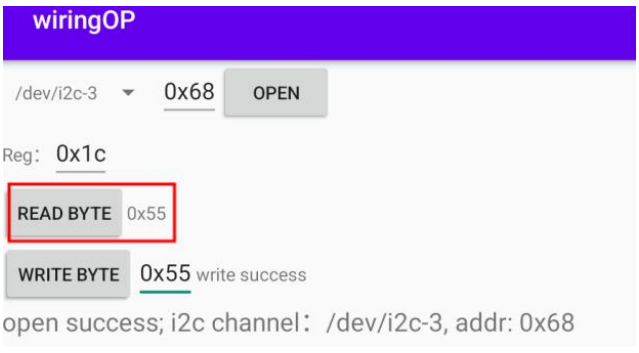
a. We first set the address of the register to be written to 0x1c



b. Then set the value to be written to 0x55

c. Then click the **WRITE BYTE** button to execute the write action

10) Then click the **READ BYTE** button to read the value of the 0x1c register, if it is displayed as 0x55, it means that the i2c read and write test has passed



6. How to compile Android 12 source code

6. 1. Download the source code of Android 12

1) First download the Android 12 source code sub-volume compressed package from Baidu or Google Disk

a. Baidu Netdisk

b. Google Disk

2) After downloading the sub-volume compression package of the Android 12 source code, please check whether the MD5 checksum is correct. If not, please download the source code again. The way to check the MD5 checksum is as follows:

```
test@test:~$ md5sum -c H618-Android12-Src.tar.gz.md5sum
H618-Android12-Src.tar.gzaa: OK
H618-Android12-Src.tar.gzab: OK
.....
```

3) Then you need to combine multiple compressed files into one, and then extract the Android source code. The command looks like this:

```
test@test:~$ cat H618-Android12-Src.tar.gza* > H618-Android12-Src.tar.gz
test@test:~$ tar -xvf H618-Android12-Src.tar.gz
```

6. 2. Compile the source code of Android 12

The compilation of Android12 is carried out on an x86_64 computer with Ubuntu 22.04 installed. Other versions of Ubuntu may have some differences in system package dependencies. The mirror download address of Ubuntu 22.04 amd64 version is as follows:
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.2-desktop-amd64.iso

The x86_64 computer hardware configuration for compiling the Android12

source code is recommended to have 16GB or more memory, and 200GB or more hard disk space is recommended. The more CPU cores, the better.

1) First install the software packages required to compile the Android12 source code

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y git gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip u-boot-tools python-is-python3 \
libssl-dev libncurses5 clang gawk
```

2) Then compile the code in the **longan** folder, which mainly contains u-boot and linux kernel

a. First run **./build.sh config** to set compilation options

```
test@test:~$ cd H618-Android12-Src/longan
test@test:~/H618-Android12-Src/longan$ ./build.sh config

Welcome to mkscript setup progress
All available platform:
    0. android
    1. linux
Choice [android]: 0
All available ic:
    0. h618
Choice [h618]: 0
All available board:
    0. ft
    1. p1
    2. p2
    3. p7
    4. p7l
    5. perf1
    6. perf2
    7. perf3
    8. qa
```

```
Choice [p2]: 2
All available flash:
    0. default
    1. nor
Choice [default]: 0
All available kern_ver:
    0. linux-5.4
Choice [linux-5.4]: 0
All available arch:
    0. arm
    1. arm64
Choice [arm64]: 1
......
*** Default configuration is based on 'sun50iw9p1smp_h618_android_defconfig'
#
# configuration written to .config
#
make[1]: Leaving directory '/home/test/H618-Android12-Src/longan/out/kernel/build'
make: Leaving directory '/home/test/H618-Android12-Src/longan/kernel/linux-5.4'
INFO: clean buildserver
INFO: prepare_buildserver
```

b. Then run the **./build.sh** script to start compiling

```
test@test:~/H618-Android12-Src/longan$ ./build.sh
```

c. After the compilation is complete, you will see the following output

```
sun50iw9p1 compile Kernel successful


INFO: Prepare toolchain ...
......
INFO: build kernel OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: ----------------------------------------
INFO: build lichee OK.
INFO: ----------------------------------------
```

3) Then use the following command to compile the Android source code and generate the final Android image

```
test@test:~$ cd H618-Android12-Src
test@test:~/H618-Android12-Src$ source build/envsetup.sh
test@test:~/H618-Android12-Src$ lunch apollo_p2-userdebug
test@test:~/H618-Android12-Src$ make -j8
test@test:~/H618-Android12-Src$ pack
```

4) The storage path of the compiled and generated Android image is:

```
longan/out/h618_android12_p2_uart0.img
```

7. Appendix

7.1. User Manual Update History

Version	Date	Update Notes
v1.0	2023-07-05	initial version
v1.1	2023-07-14	Add the compilation method of Android 12 source code

7.2. Image update history

Date	Update Notes
2023-07-05	Orangepizero3_1.0.0_ubuntu_jammy_server_linux5.4.125.7z Orangepizero3_1.0.0_debian_bullseye_server_linux5.4.125.7z Orangepizero3_1.0.0_ubuntu_focal_desktop_xfce_linux5.4.125.7z Orangepizero3_1.0.0_ubuntu_jammy_desktop_xfce_linux5.4.125.7z Orangepizero3_1.0.0_debian_bullseye_desktop_xfce_linux5.4.125.7z Orangepizero3_1.0.0_ubuntu_jammy_server_linux6.1.31.7z Orangepizero3_1.0.0_debian_bookworm_server_linux6.1.31.7z Orangepizero3_1.0.0_debian_bullseye_server_linux6.1.31.7z Orangepizero3_1.0.0_ubuntu_jammy_desktop_xfce_linux6.1.31.7z

	Orangepizero3_1.0.0_debian_bookworm_desktop_xfce_linux6.1.31.7z Orangepizero3_1.0.0_debian_bullseye_desktop_xfce_linux6.1.31.7z OrangePi_Zero3_Android12_v1.0.tar.gz *initial version
2023-07-13	Opios-arch-aarch64-xfce-opizero3-23.07-linux6.1.31.img.xz * initial version