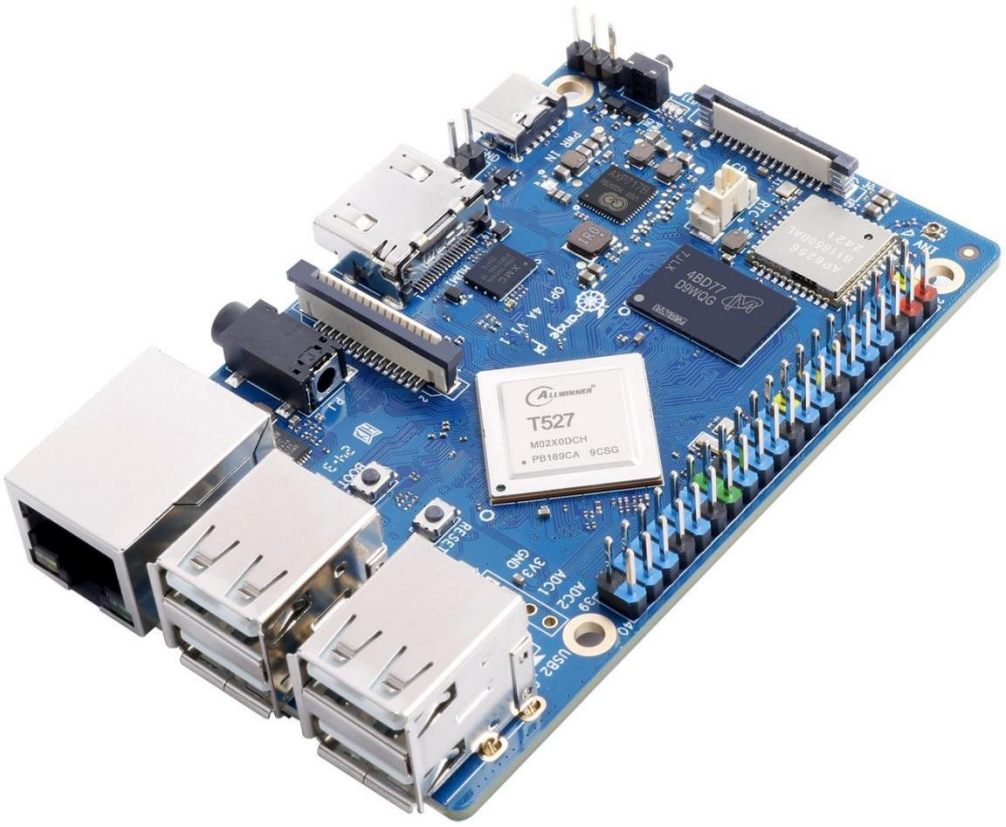
Orange Pi 4A

User Manual



Catalogue

1. Basic characteristics of Orange Pi 4A

1.1. What is Orange Pi 4A

The Orange Pi 4A adopts the Allwinner T527 eight core Cortex-A55+HiFI4 DSP+RISV-V multi-core heterogeneous industrial grade processor, supporting 2TOPS NPU to meet the needs of edge intelligent AI acceleration applications; Supports 2GB/4GB LPDDR4/4X and provides H.265 4K@60fps And H.264 4K@60fps Video decoding, H.264 4K@25fps code; Rich interfaces, including commonly used functional interfaces such as Gigabit Ethernet, PCIe 2.0, USB 2.0, MIPI-CSI, MIPI-DSI, 40Pin expansion interface, etc. Supports operating systems such as Ubuntu, Debian, Android 13, etc.

Orange Pi 4A can provide a solid hardware foundation for the scenario landing of generative AI and artificial intelligence algorithms, and can be widely used in intelligent industrial control, intelligent business display, retail payment, intelligent education, commercial robots, vehicle terminals, visual assistant driving, edge computing, intelligent power distribution terminals, etc.

1.2. The purpose of Orange Pi 4A

We can use it to achieve:

- A small Linux desktop computer
- A small Linux network server
- Android tablet
- Android game consoles, etc

Of course, there are many other features as well. With a powerful ecosystem and various expansion accessories, Orange Pi can help users easily achieve delivery from creativity to prototype to mass production. It is an ideal creative platform for makers, dreamers, and hobbyists.

1.3. Who is Orange Pi 4A designed for

Orange Pi development board is not only a consumer product, but also designed for anyone who wants to use technology for creative innovation. It is a simple, fun, and practical tool that you can use to create the world around you.

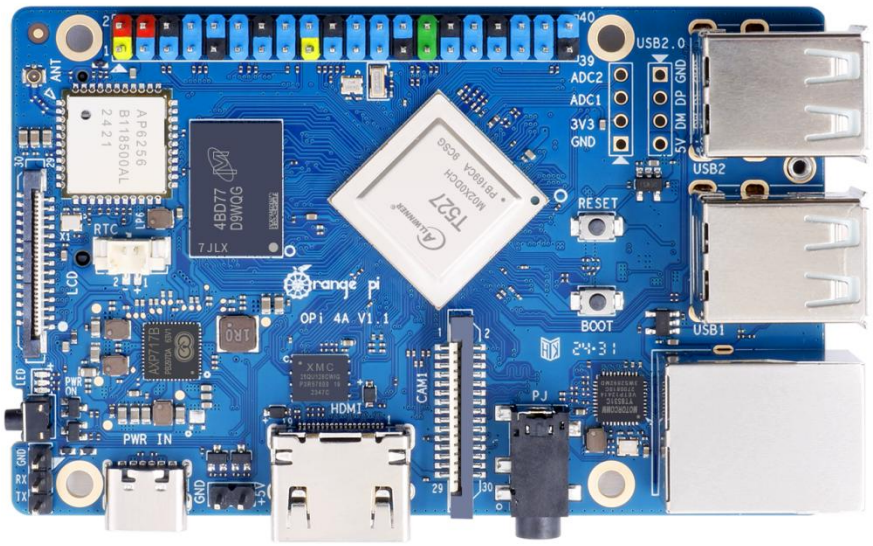
1.4. Hardware Features of Orange Pi 4A

Introduction to Hardware Features	
Processor	T527，8-core ARM CortexTM-A55@1.8GHz HIFI4 Audio DSP@600MHz RISC-V@200MHz GPU：G57 MC1 VPU： H.265 4K@60fps Decoding, H.264 4K@60fps Decoding, H.264 4K@25fps code NPU：2TOPS
Memory	LPDDR4/4X:2GB/4GB optional
Storage	EMMC module optional: 16GB/32GB/64GB/128GB optional SPI Flash：128Mb (default paste), 256Mb optional M.2 M-KEY Socket：PCIe2.0 NVMe SSD uSD card slot：supports up to 128GB uSD card
Wi Fi+Bluetooth	Wi Fi+Bluetooth two in one module Wi-Fi5.0+BT 5.0，BLE
Ethernet	10/100/1000Mbps Ethernet
Display	1x HDMI TX 2.0 interface up to 4K@60fps 1x 4-lane MIPI-DSI 1x eDP1.3
Camera	1x 2-lane MIPI-CSI camera interface 1x 4-lane MIPI-CSI camera interface

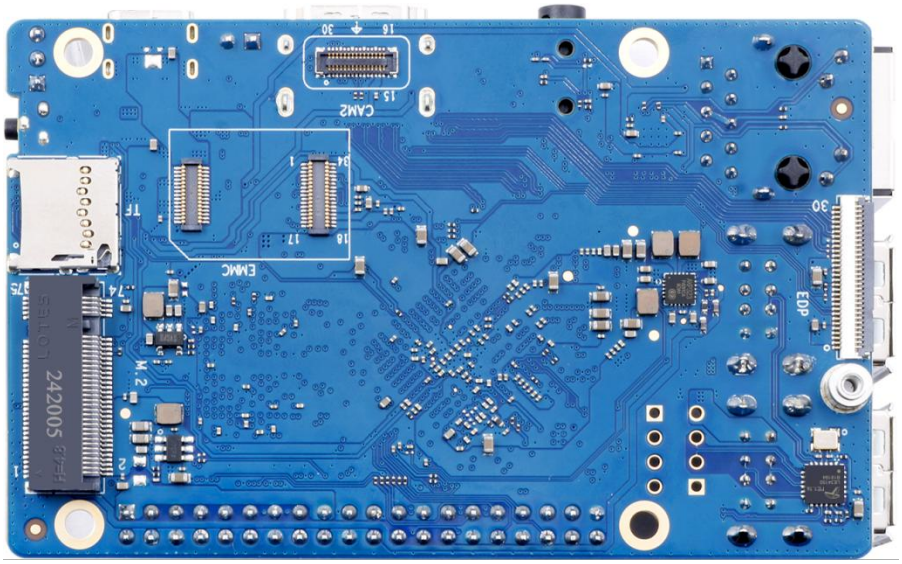
USB	1xUSB Type-A 2.0 3xUSB Type-A 2.0 HOST 1xUSB 2.0 HOST reserved for customer expansion
ADC	Reserved 4pin interface, capable of connecting 2 ADCs, with a maximum input of 1.8V
Audio frequency	3.5mm headphone jack audio input/output
Key	1* BOOT，1*RESET，1 *PWR ON
RTC	2Pin backup battery interface (Pitch=1.27mm)
40Pin	40Pin function extension interface, supporting the following interface types: GPIO、UART、I2C、SPI、PWM
DEBUG	3Pin debugging serial port
Power Supply	Type-C 5V 5A DCIN
Supported OS	Ubuntu、Debian、Android13, etc
Introduction to appearance specifications	
PCB	89mm*56mm*1.6mm
Weight	52g

1. 5. Top and Bottom Views of Range Pi 4A

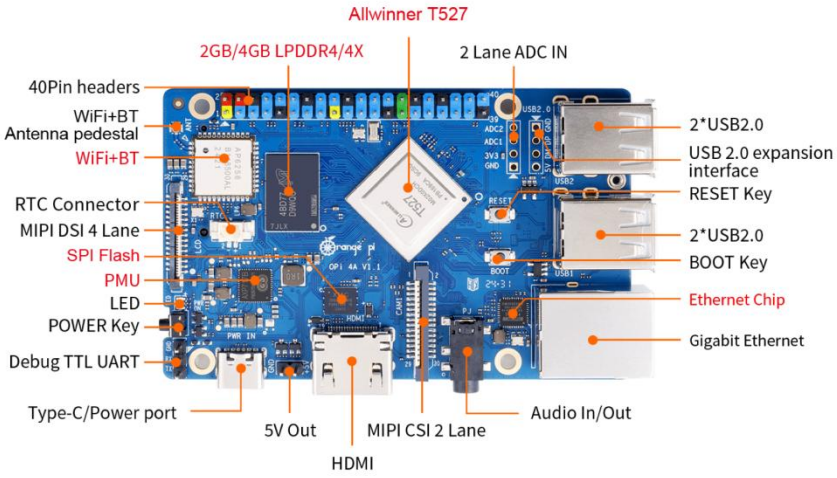
Top level view:



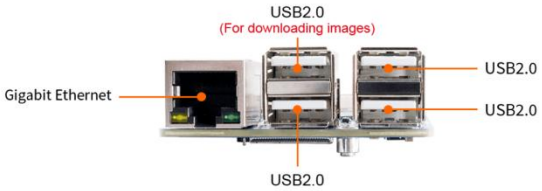
Bottom level view:



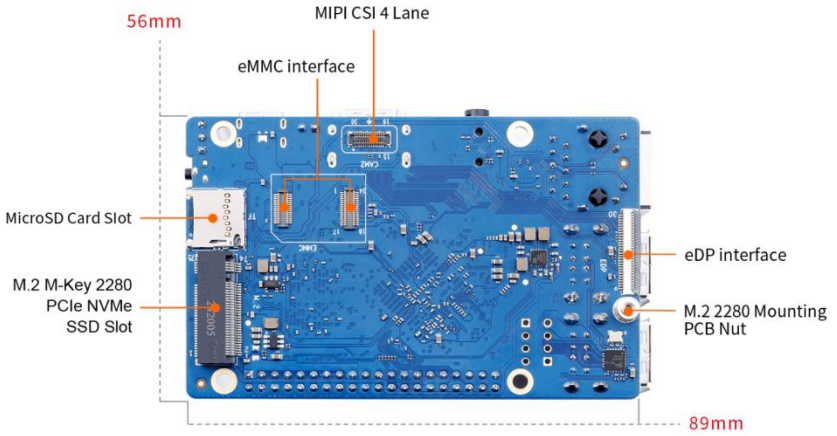
1. 6. Interface Details of Range Pi 4A

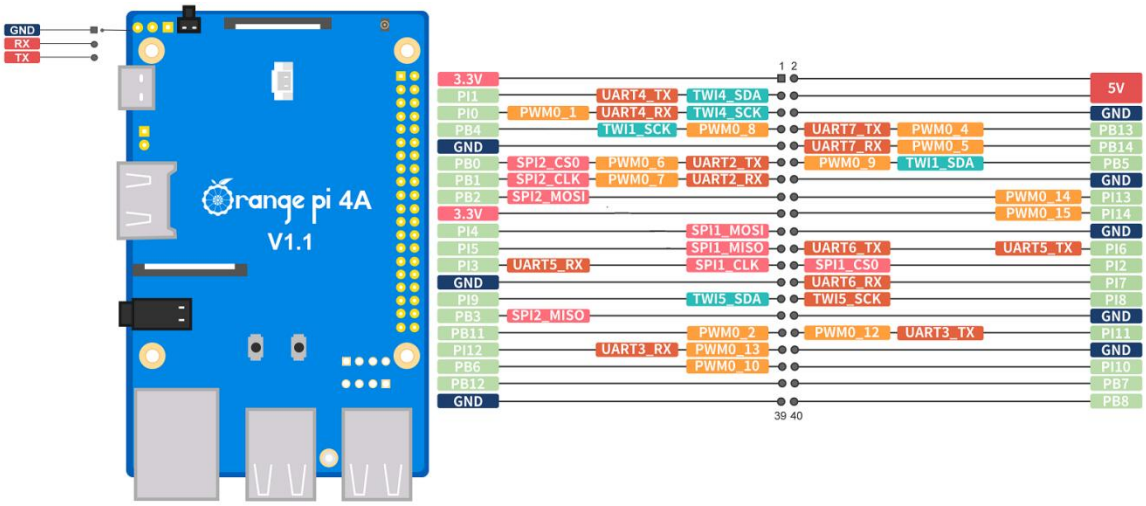


Top View



Side View





The diameter of the four positioning holes is 3.0mm.

2. Introduction to using the development board

2. 1. Prepare the necessary accessories

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

2) TF card reader, used for reading and writing TF cards

3) HDMI interface display

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

5) 10.1-inch MIPI screen, used to display the system interface of the development board (this screen includes adapter board and OPi5Plus/OPi5B/OPi5Pro/OPi5Max/OPi4A universal)

6) Power adapter, Orange Pi 4A recommends using a 5V/5A Type-C power supply for power supply

The Type-C power interface of the development board does not support PD negotiation function and only supports a fixed 5V voltage input.

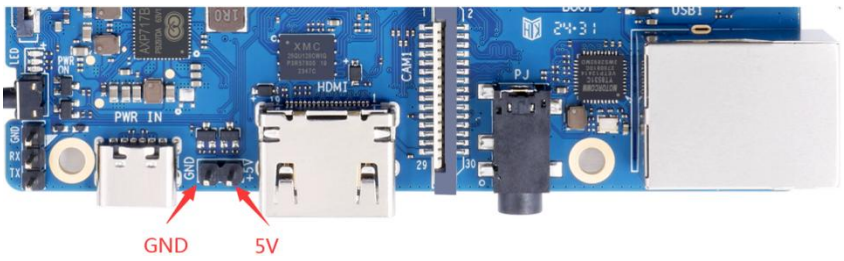
7) USB interface mouse and keyboard, any standard USB interface mouse and keyboard can be used to control the Orange Pi development board

8) USB camera

9) 100Mbps or 1G Ethernet cable, used to connect the development board to the Internet

10) A 5V cooling fan, as shown in the figure below, has a dedicated 5V output interface on the development board for connecting to the cooling fan, with a spacing of **2.54mm**. The power interface of the cooling fan can be purchased according to this specification.

Note that once the development board is plugged in, the 5V pin can be used directly without any additional settings. Additionally, the voltage output from the 5V pin cannot be adjusted or turned off through software.



11) USB 2.0 male to male data cable, used for adb debugging, burning images to eMMC and other functions

12) When using the serial port debugging function, USB to TTL module and DuPont cable are required 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 acceptable.

13) X64 computer with Ubuntu and Windows operating systems installed

1	Ubuntu22.04 PC	Optional, used for compiling Android and Linux source code
2	Windows PC	Used for burning Android and Linux images

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

1) The download link for the Chinese version of the materials is

http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-4A.html

2) The download link for the English version of the material is

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-4A.html

3) The information mainly includes

a. **Linux source code**：saved on Github
b. **Android image**：saved on Google Drive
c. **Ubuntu image**：saved on Google Drive
d. **Debian image**：saved on Google Drive
e. **User manual and schematic diagram**：saved on Google Drive
f. **Official tools**：saved on Google Drive. Mainly including the software required during the use of the development board

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

Note that the Linux image referred to here specifically refers to Linux distribution images such as Debian or Ubuntu downloaded from the Orange Pi data download page.

2.3.1. Method of burning Linux images using BalenaEtcher

1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brands

2) Then use the card reader to insert the TF card into the computer

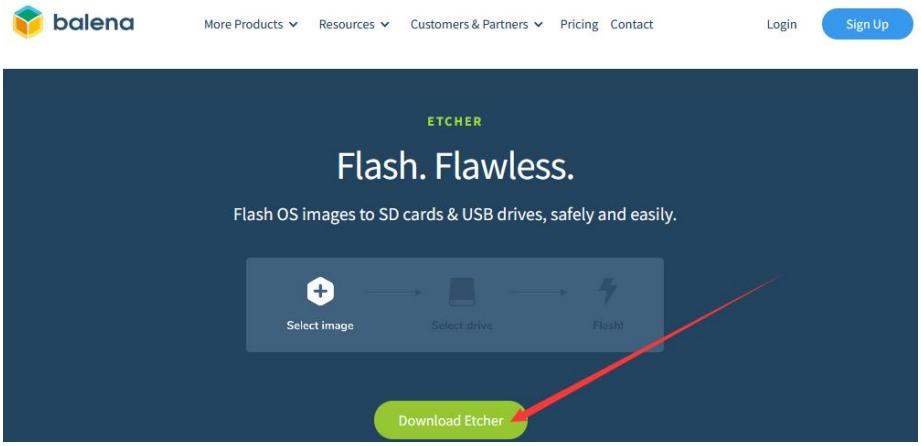
3) Download the compressed file of the Linux operating system image that you want to

burn from the **Orange Pi's download page**, and then use decompression software to decompress it. In the decompressed file, the file ending with "**.img**" is the operating system image file, which is usually over 1GB in size

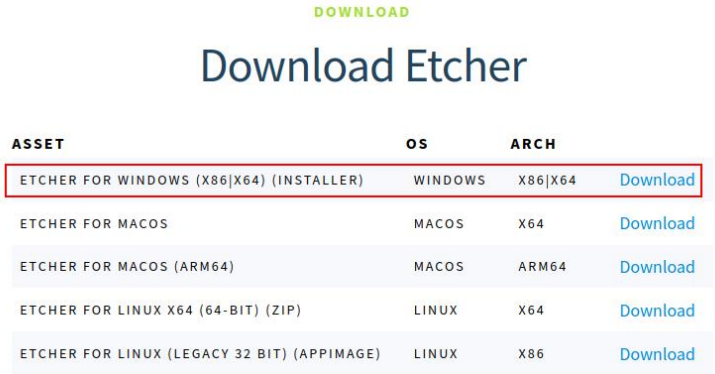
4) Then download the Linux image burning software - **balenaEtcher**, from the download link

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

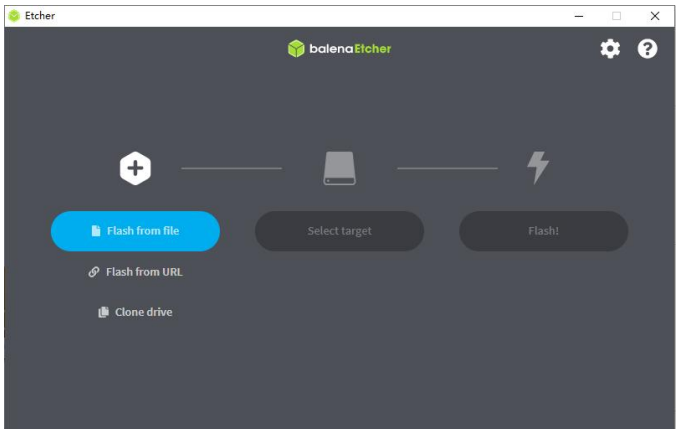
5) After entering the balenaEtcher download page, clicking the green download button will jump to the software download selection interface



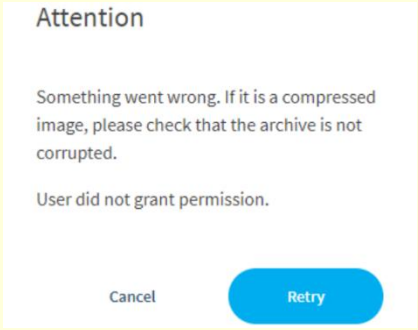
6) Then download the installation package for balenaEtcher Windows version.



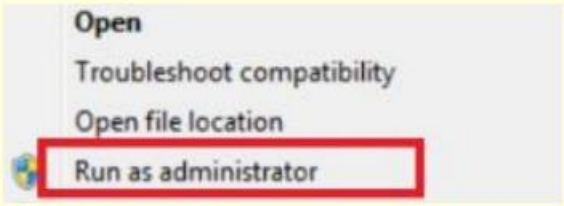
7) Then install balenaEtcher and open it again. The balenaEtcher interface after opening is shown in the following figure



When opening balenaEtcher, if prompted with the following error:

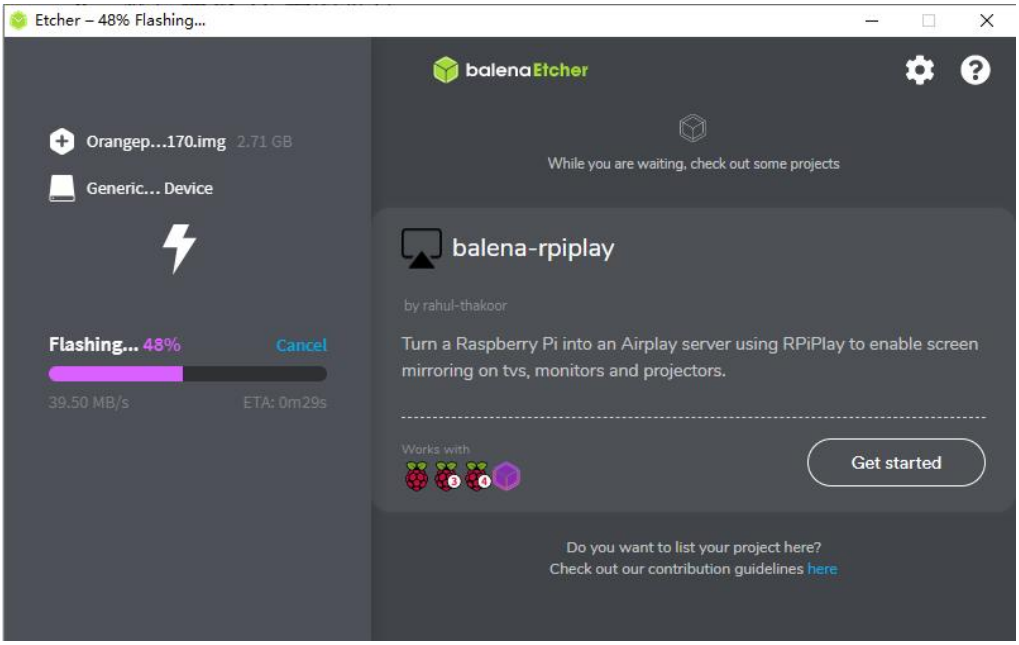


Please select balenaEtcher and right-click, then choose to run as administrator.

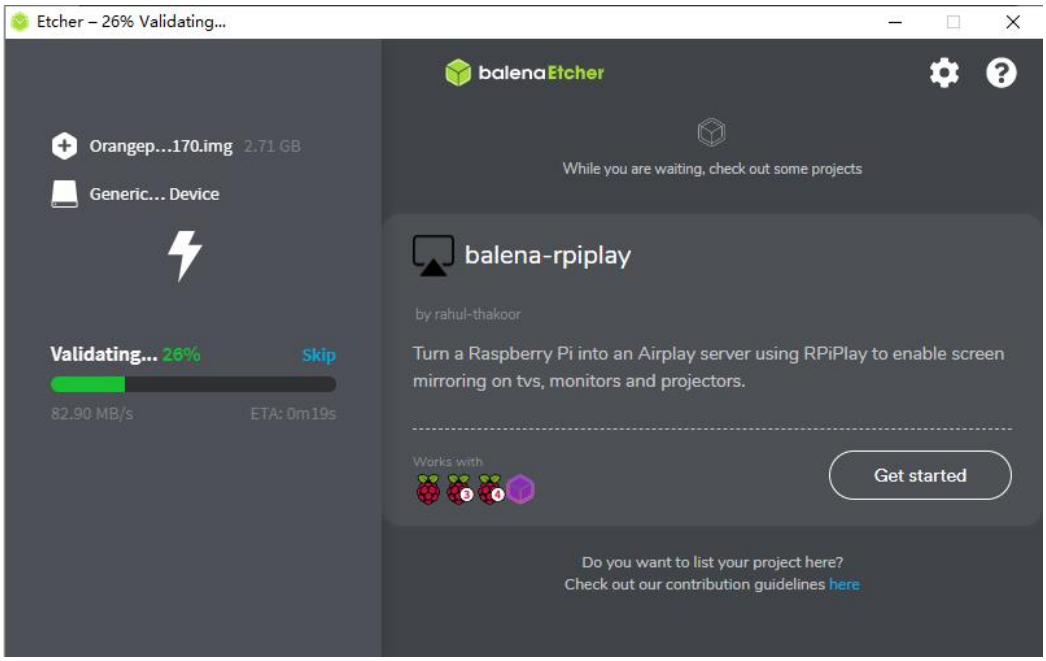


8) The specific steps to use balenaEtcher to burn the Linux image are aa follow
   a. First select the path of the Linux image file to burned
   b. Then select the drive letter of the TF card
   c. Finally, click Flash to start burning the Linux image to the TF card

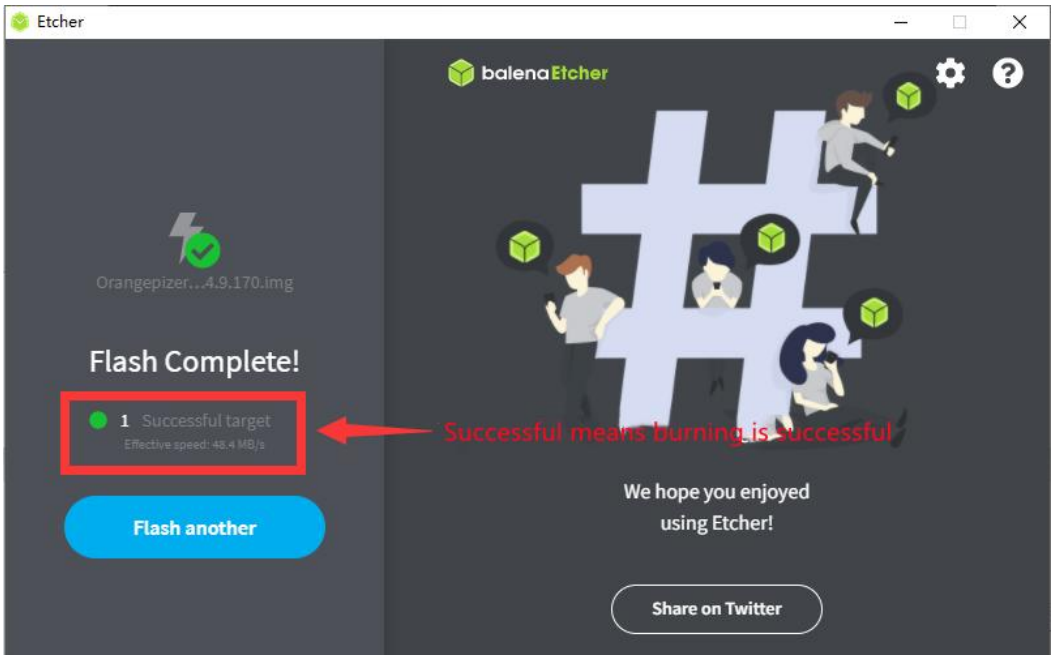
9) The interface displayed during the process of burning a Linux image by balenaEtcher is shown in the following figure. In addition, the progress bar displaying purple indicates that the Linux image is being burned to the TF card



10) After the Linux image is burned, balenaEtcher will also verify the image burned to the TF card by default to ensure that there are no problems during the burning process. As shown in the following figure, a green progress bar indicates that the image has been burned and balenaEtcher is verifying the burned image



11) After successful burning, the display interface of balenaEtcher is shown in the following figure. If a green indicator icon is displayed, it indicates that the image burning is successful. At this time, you can exit balenaEtcher, then unplug the TF card and insert it into the TF card slot of the development board for use



2. 3. 2. Method of burning Linux images using Win32Diskimager

1) First prepare an 8GB or larger capacity TF card, TF card transmission speed must be **class10** or above, it is recommended to use Sandisk and other brands of TF card

2) Then use the card reader to insert the TF card into the computer

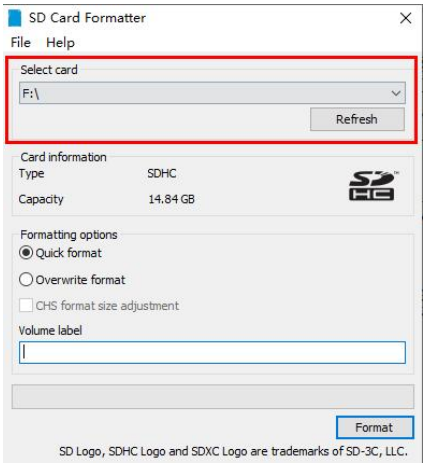
3) Then format the TF card

a. You can use **SD Card Formatter** to format TF cards. The download address is

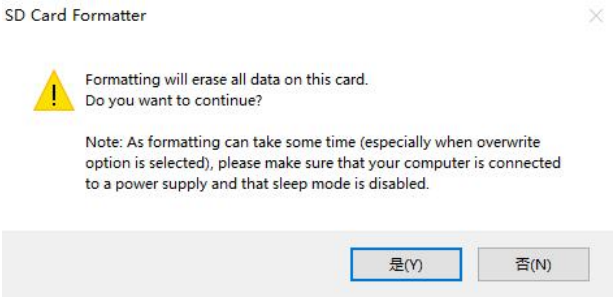
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

b. After downloading the software, decompress and install it. Then open the software

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

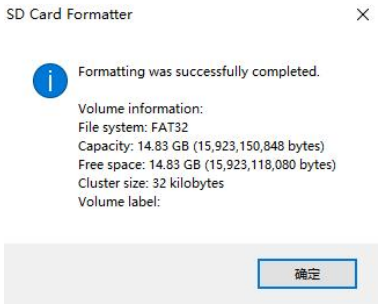


d. Then click "**Format**", a warning box will pop up before formatting, select "**Yes (Y)**" will start formatting



e. After formatting the TF card, the message as shown in the following figure will

pop up. Click OK



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

5) Burn Linux image to TF card using **Win32Diskimager**

a. The download page for Win32Diskimager is

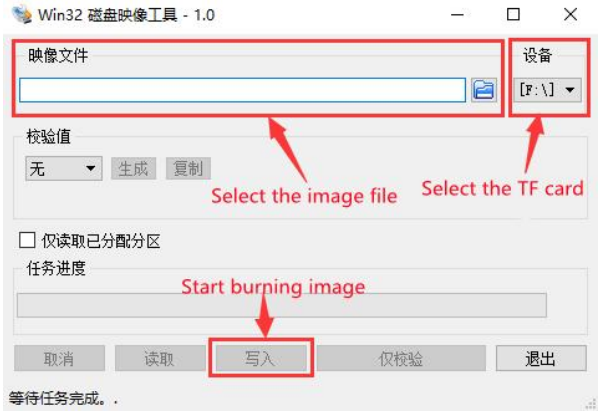
http://sourceforge.net/projects/win32diskimager/files/Archive/

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

a) First select the path of the image

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

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



c. After the image writing is completed, click the "**Exit**" button to exit, and then

you can pull out the TF card and insert it into the development board to start

2. 4. Method for burning Linux images to TF cards based on Ubuntu PC

Note that the Linux image referred to here specifically refers to Linux distribution images such as Debian or Ubuntu downloaded from the Orange Pi data download page, while Ubuntu PC refers to a personal computer with the Ubuntu system installed.

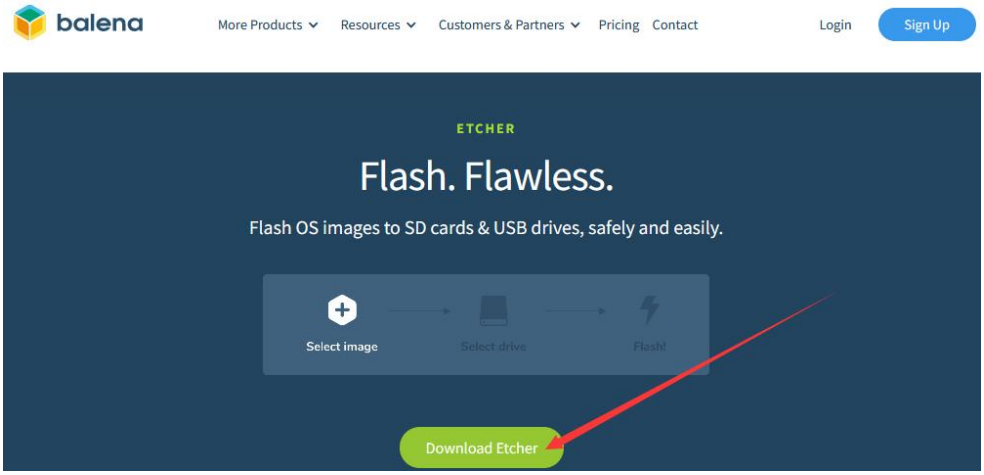
1) Firstly, prepare a TF card with 8GB or larger capacity, and the transfer speed of the TF card must be **class10** or above. It is recommended to use TF cards from brands such as SanDisk

2) Then use a card reader to insert the TF card into the computer

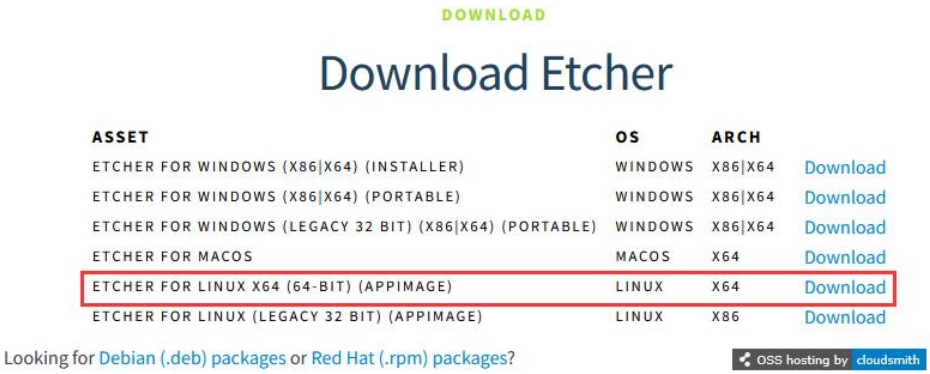
1) Download the balenaEtcher software from the following link:

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

2) After entering the balenaEtcher download page, clicking the green download button will jump to the software download location



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



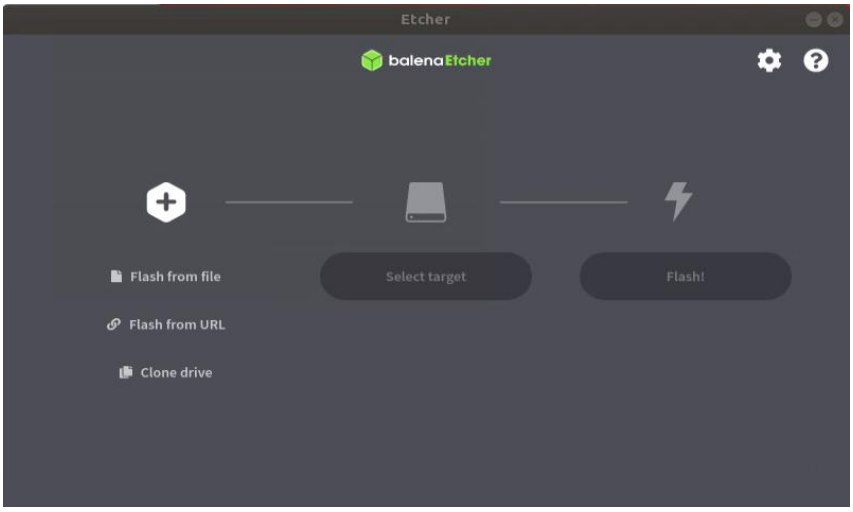
4) Download the compressed file of the Linux operating system image that you want to burn from the **Orange Pi download page**, and then use decompression software to decompress it. In the decompressed file, the file ending with "**.img**" is the operating system image file, which is usually over 1GB in size. The decompression command for the compressed file ending in 7z is as follows:

```
test@test:~$ 7z x Orangepi4a_1.0.0_ubuntu_jammy_desktop_linux5.15.147.7z
test@test:~$ ls Orangepi4a_1.0.0_ubuntu_jammy_desktop_linux5.15.147.*
Orangepi4a_1.0.0_ubuntu_jammy_desktop_linux5.15.147.7z
Orangepi4a_1.0.0_ubuntu_jammy_desktop_linux5.15.147.sha      #Verification and file
Orangepi4a_1.0.0_ubuntu_jammy_desktop_linux5.15.147.img      #Image file
```

5) After decompressing the image, you can first use the **sha256sum -c *.sha** command to calculate if the checksum is correct. If the prompt is **successful**, it means that the downloaded image is correct and can be safely burned to the TF card. If the prompt is that the **checksum does not match**, it means that the downloaded image has a problem. Please try downloading it again

```
test@test:~$ sha256sum -c *.sha
Orangepi4a_1.0.0_ubuntu_jammy_desktop_linux5.15.147.img: success
```

6) Then double-click **balenaEtcher-1.14.3-x64.AppImage** on the graphical interface of Ubuntu PC to open balenaEtcher (**no installation required**). The interface displayed after opening balenaEtcher is shown in the following figure

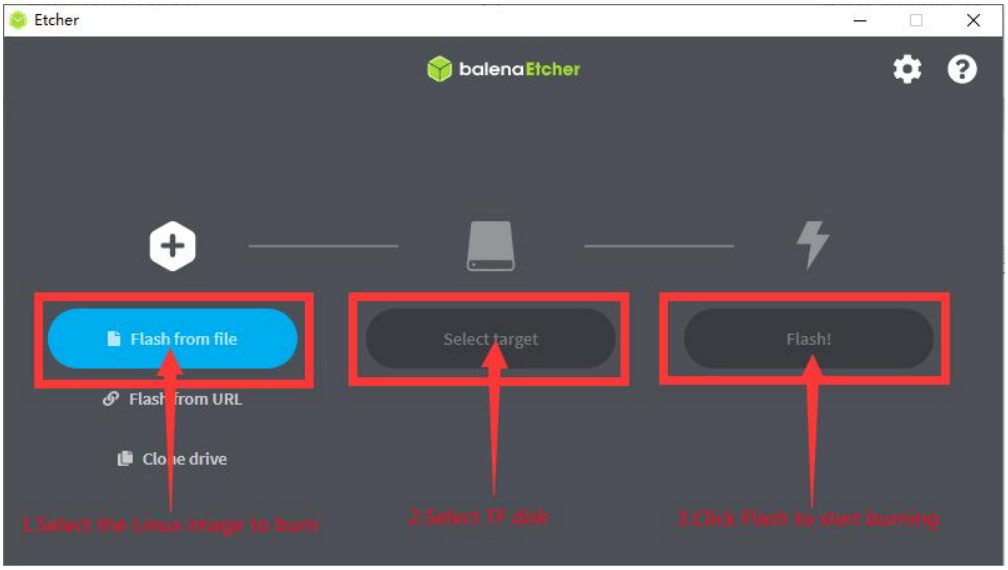


7) The specific steps for burning a Linux image using balenaEtcher are as follows

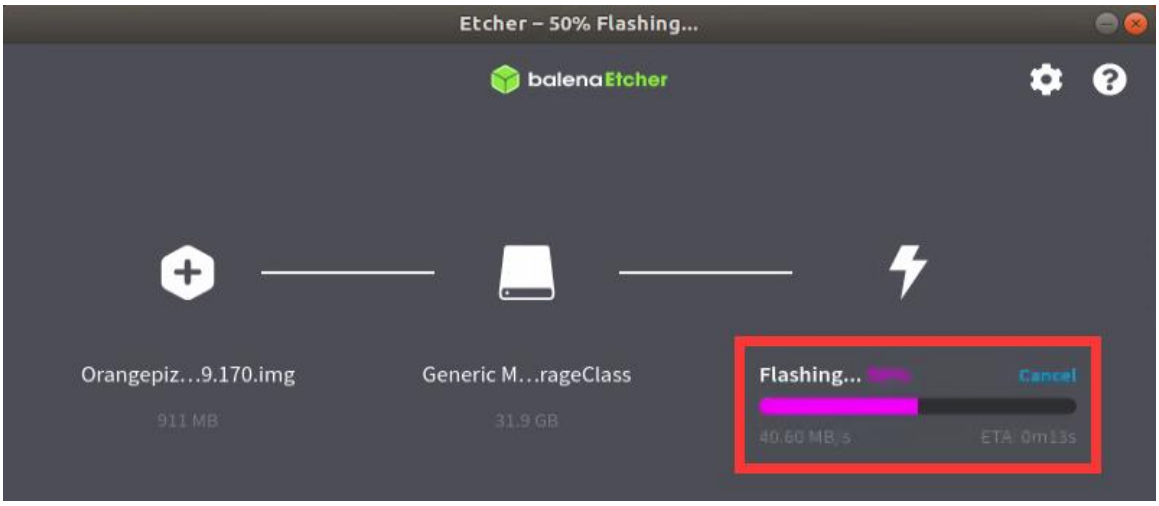
a. First select the path of the Linux image file to burned

b. Then select the drive letter of the TF card

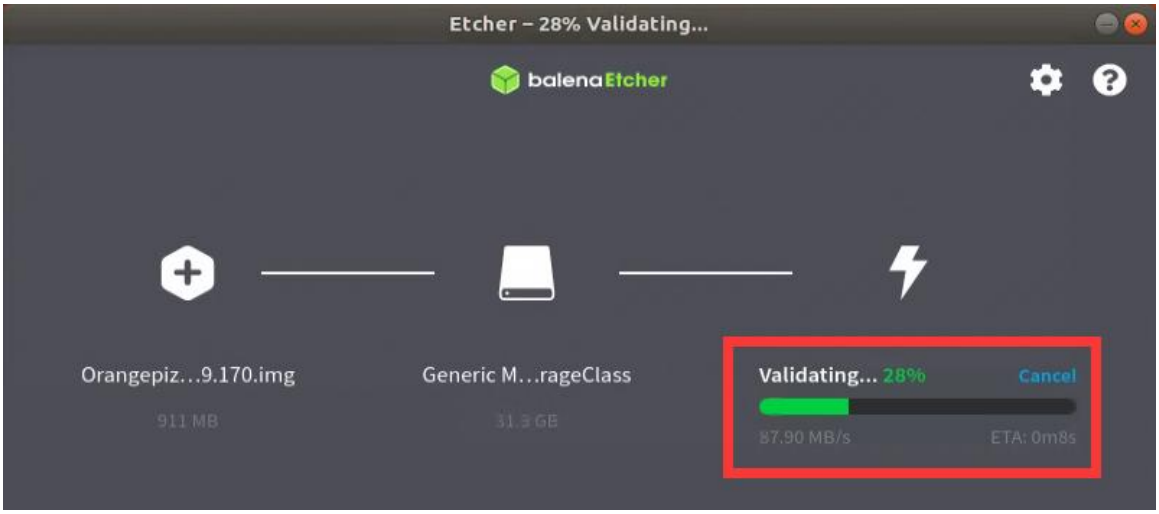
c. Finally, click Flash to start burning the Linux image to the TF card



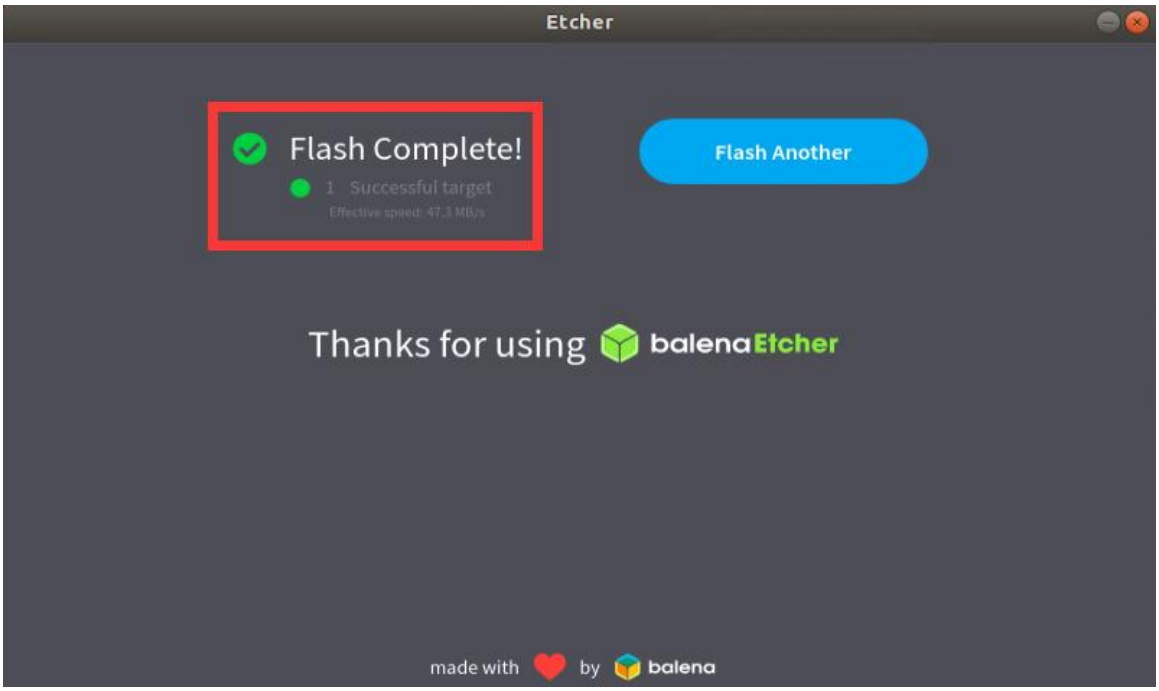
8) The interface displayed during the process of burning a Linux image by balenaEtcher is shown in the following figure. In addition, the progress bar displaying purple indicates that the Linux image is being burned to the TF card



9) After the Linux image is burned, balenaEtcher will also verify the image burned to the TF card by default to ensure that there are no problems during the burning process. As shown in the following figure, a green progress bar indicates that the image has been burned and balenaEtcher is verifying the burned image



10) After successful burning, the display interface of balenaEtcher is shown in the following figure. If a green indicator icon is displayed, it indicates that the image burning is successful. At this time, you can exit balenaEtcher, then unplug the TF card and insert it into the TF card slot of the development board for use



2. 5. Method for burning Linux images to eMMC

See **the method of burning Linux images to EMMC**

2. 6. Method for burning Linux images to SPIFlash+NVMe SSD

See **the method of burning Linux images to SPIFlash+NVMe SSD**

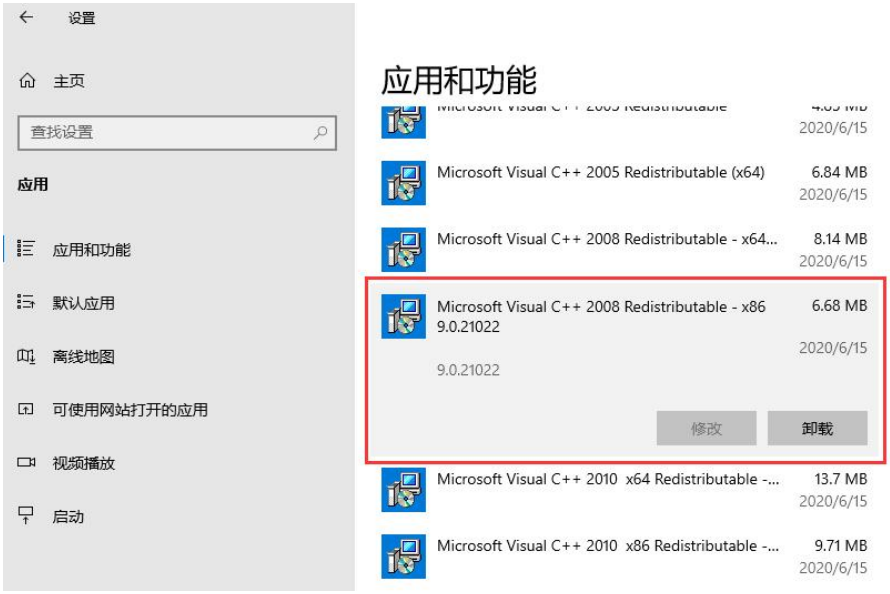
2. 7. Method of burning Android image to TF card

The Android image of the development board can only be burned to a TF card using PhoenixCard software on the Windows platform, and the version of PhoenixCard software must be PhonixCard-4.2.8.

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

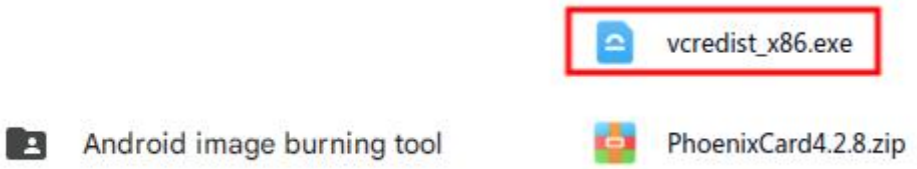
In addition, PhoenixCard software does not have versions for Linux and Mac platforms, so it is not possible to burn Android images to TF cards on Linux and Mac platforms.

1) Firstly, please ensure that the Windows system has installed **Microsoft Visual C++ 2008 Redistrbutable - x86**



2) If **Microsoft Visual C++ 2008 Redistrbutable - x86** is not installed, formatting the TF card with **PhoenixCard** or burning the Android image will prompt the following error

3) The installation package for **Microsoft Visual C++ 2008 Redistrbutable - x86** can be downloaded from the **official tool** of Orange Pi 4A or from the **Microsoft official website**



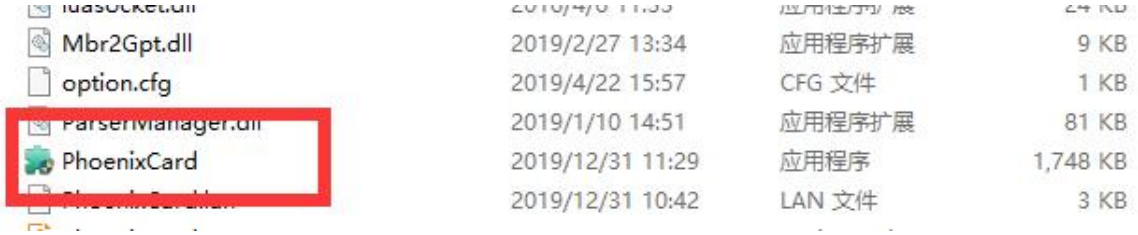
4) Firstly, prepare a TF card with 8GB or larger capacity, and the transfer speed of the TF card must be **class10** or above. It is recommended to use TF cards from brands such as SanDisk

5) Then use a card reader to insert the TF card into the computer

6) Download the Android image and PhoenixCard burning tool from the **Orange Pi's download page**. Please ensure that the version of the PhonenixCrad tool is **PhonexCard-4.2.8. Do not use PhonixCard software below version 4.2.8 to burn the Android image**, as Android images burned by PhonixCard tools below this version may have problems

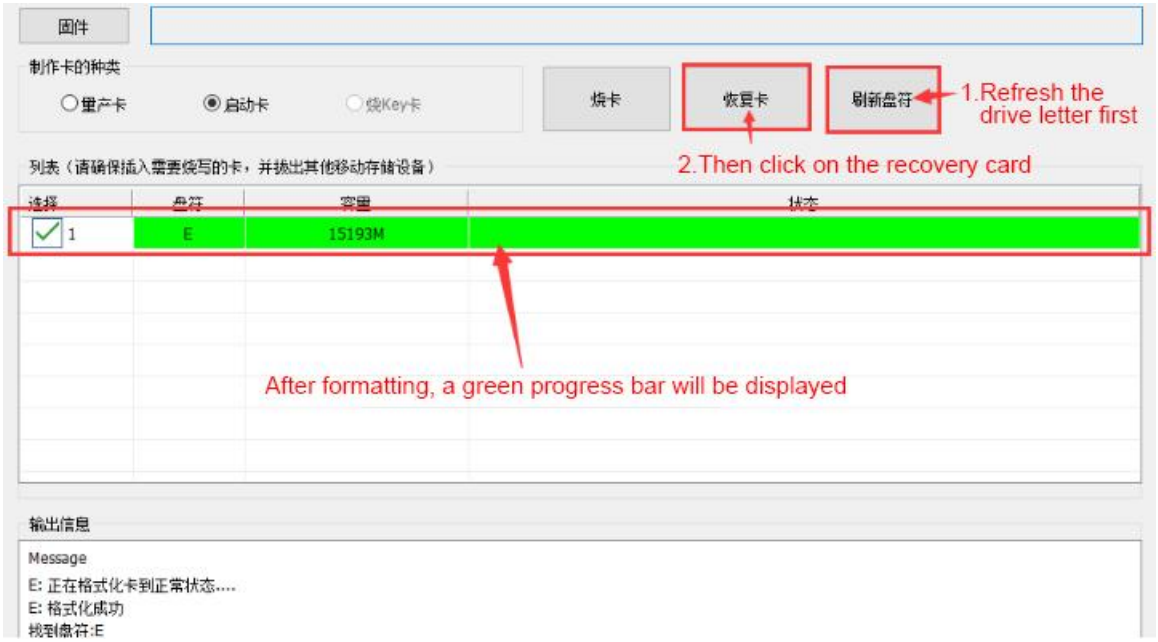
7) Then use decompression software to decompress the downloaded Android image compressed file. In the decompressed file, the file ending with "**.img**" is the Android image file, with a size of 1GB or more. If you don't know how to decompress the compressed file of an Android image, you can install a **360 compression software** to decompress the image.

8) Then use decompression software to extract **PhonixCard4.2.8.zip**. This software does not require installation, just find PhoenixCard in the extracted folder and open it



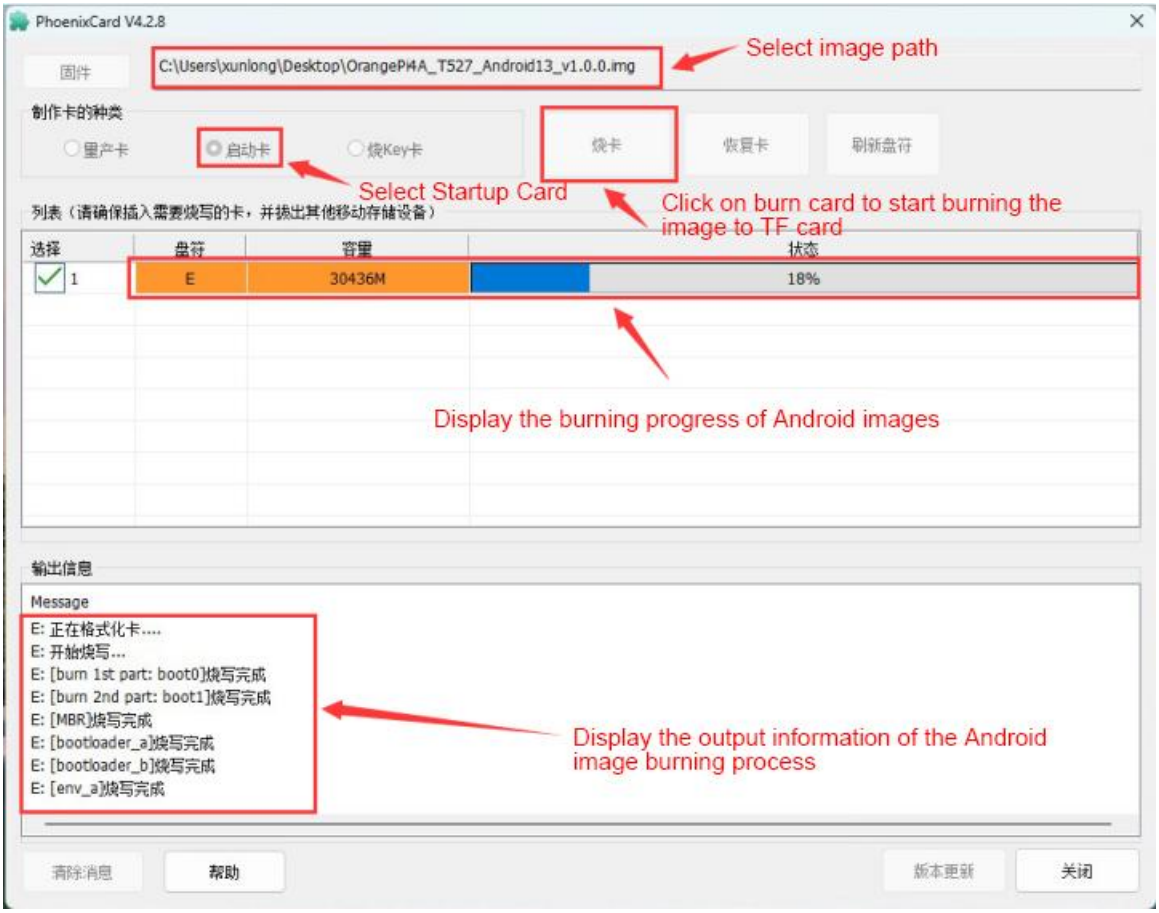
9) After opening PhoenixCard, if the TF card is recognized normally, the drive letter and capacity of the TF card will be displayed in the middle list. **Please make sure that the displayed drive letter is consistent with the drive letter of the TF card you want to burn**. If it is not displayed, you can try unplugging the TF card or clicking the "**Refresh Drive Letter**" button in PhoenixCard

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

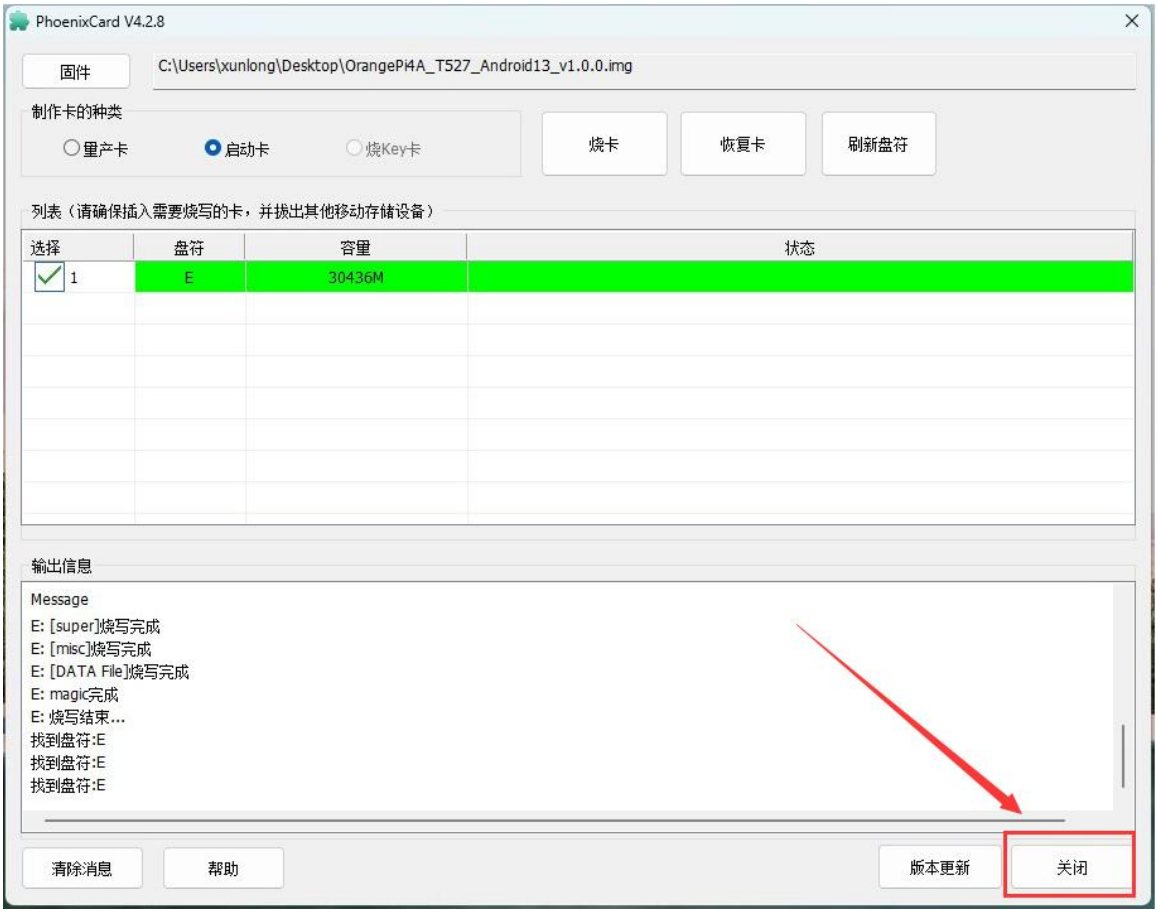


If there is a problem with formatting, please try unplugging and unplugging the TF card before testing again. If the problem persists after unplugging and unplugging the TF card again, you can restart your Windows computer or switch to another computer and try again.

11) Then start writing the Android image to the TF card
    a. Firstly, select the path of the Android image in the "**Firmware**" column
    b. Select "**Startup Card**" in the "**Types of Cards to Make**" section
    c. Then click the "**Burn Card**" button to start burning



12) After burning, the display of PhoenixCard is shown in the following figure. Click the "**Close**" button to exit PhoenixCard, and then you can unplug the TF card from the computer and insert it into the development board to start it



After burning the Android system, only one 128 MB partition can be seen on the TF card in Windows, as shown in the following figure (some computers may pop up more than twenty disk partitions, but can only open the 128 MB partition). Please note that this is normal and do not burn out the TF card. The reason for this is that the Android system has over twenty partitions, but most of them cannot be recognized properly in the Windows system. At this point, please feel free to unplug the TF card and insert it into the development board to start.

After starting the Android system, use the following command to see these twenty partitions in the TF card:

```
console:/ # ls /dev/block/mmcblk0*
/dev/block/mmcblk0      /dev/block/mmcblk0p18  /dev/block/mmcblk0p27
/dev/block/mmcblk0p1    /dev/block/mmcblk0p19  /dev/block/mmcblk0p28
/dev/block/mmcblk0p10   /dev/block/mmcblk0p2   /dev/block/mmcblk0p3
/dev/block/mmcblk0p11   /dev/block/mmcblk0p20  /dev/block/mmcblk0p4
/dev/block/mmcblk0p12   /dev/block/mmcblk0p21  /dev/block/mmcblk0p5
/dev/block/mmcblk0p13   /dev/block/mmcblk0p22  /dev/block/mmcblk0p6
/dev/block/mmcblk0p14   /dev/block/mmcblk0p23  /dev/block/mmcblk0p7
/dev/block/mmcblk0p15   /dev/block/mmcblk0p24  /dev/block/mmcblk0p8
/dev/block/mmcblk0p16   /dev/block/mmcblk0p25  /dev/block/mmcblk0p9
/dev/block/mmcblk0p17   /dev/block/mmcblk0p26
```

Using the df -h command, you can see that after burning the 32GB TF card to the Android system, there is still approximately 24GB of space available for use (not all of the twenty partitions will be mounted to the Android system, so focus on the visible partitions).

```
console:/ # df -h
Filesystem                Size Used Avail Use% Mounted on
tmpfs                     963M 1.2M  961M   1% /dev
tmpfs                     963M    0  963M   0% /mnt
/dev/block/dm-0           803M 803M     0 100% /
/dev/block/dm-4           232K  36K  196K  16% /system_dlkm
/dev/block/dm-1            88M  88M     0 100% /vendor
/dev/block/dm-3            11M  11M     0 100% /vendor_dlkm
/dev/block/dm-2           106M 106M     0 100% /product
tmpfs                     963M 8.0K  963M   1% /apex
tmpfs                     963M 488K  962M   1% /linkerconfig
/dev/block/mmcblk0p21      10M 120K   10M   2% /metadata
/dev/block/mmcblk0p22      80M  54M   26M  67% /treadahead
/dev/block/mmcblk0p26      16M    0   16M   0% /oem
/dev/block/mmcblk0p27      64M 4.0K   64M   1% /Reserve0
/dev/block/dm-5            26G 1.8G   24G   8% /data
tmpfs                     963M    0  963M   0% /data_mirror
/dev/fuse                  26G 1.8G   24G   8% /mnt/user/0/emulated
/dev/block/vold/public:179,1 128M  18M  109M  15% /mnt/media_rw/extsd
/dev/fuse                 128M  18M  109M  15% /mnt/user/0/extsd
```

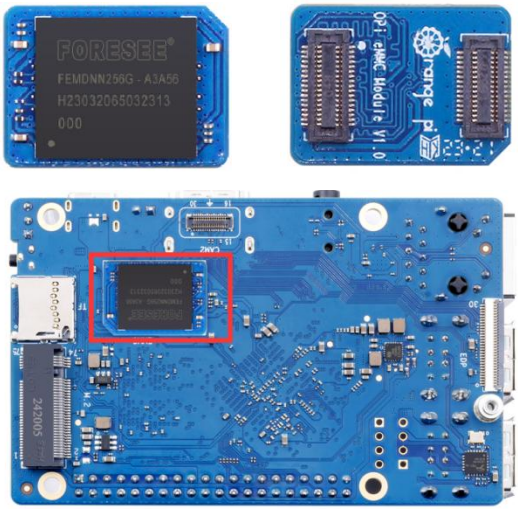
2. 8. Method for burning Android images to eMMC

The Android image of the development board can only be burned to eMMC using PhoenixCard software on the Windows platform, and the version of PhoenixCard software must be PhonixCard-4.2.8.

Please do not use software that burns Linux images, such as Win32Diskimager or BalenaEtcher, to burn Android images.

In addition, PhoenixCard software does not have versions for Linux and Mac platforms, so it is not possible to burn Android images to eMMC on Linux and Mac platforms.

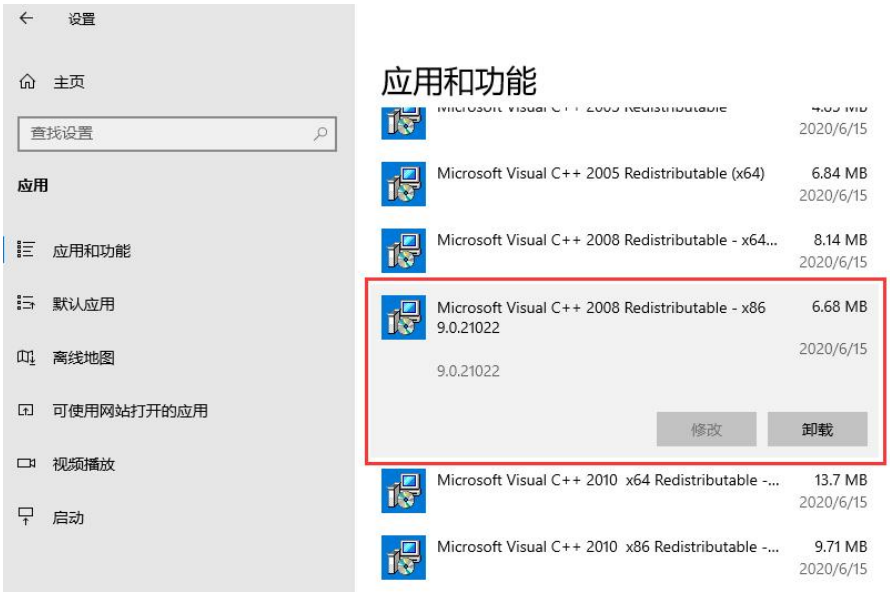
1) The development board has reserved an expansion interface for the eMMC module. Before burning the system to eMMC, it is necessary to purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC module onto the development board. The method of inserting the eMMC module into the development board is as follows:



2) Firstly, please note that this method requires the use of a TF card and is mainly divided into the following two steps

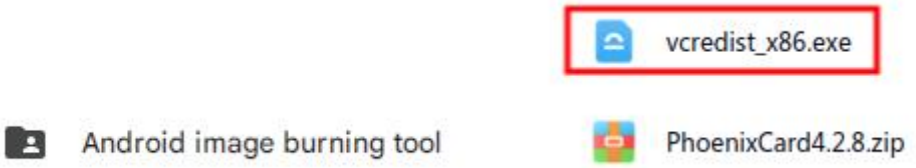
a. First, use PhoenixCard to burn the Android firmware onto the TF card as a production card
b. Then use a TF card to burn the Android firmware into eMMC

3) Please ensure that the Windows system has installed **Microsoft Visual C++ 2008 Redistrbutable - x86**



4) If **Microsoft Visual C++ 2008 Redistrbutable - x86** is not installed, formatting the TF card with PhoenixCard or burning the Android image will prompt the following error

5) The installation package for **Microsoft Visual C++ 2008 Redistrbutable - x86** can be downloaded from the **official tool** of Orange Pi 4A or from the **Microsoft official website**



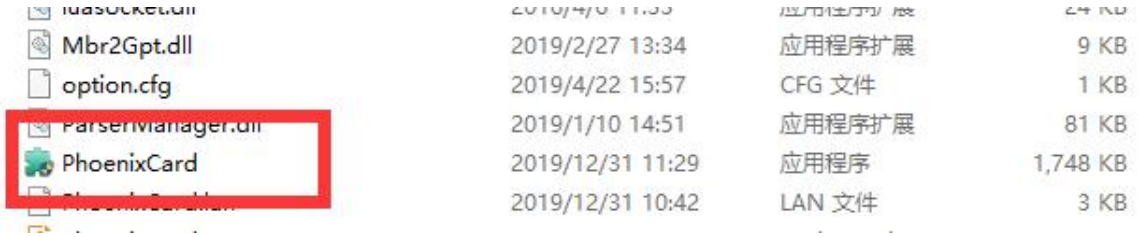
6) Then prepare an 8GB or larger capacity TF card, TF card transmission speed must be **class10** or above, it is recommended to use Sandisk and other brands of TF card

7) Then use the card reader to insert the TF card into the computer

8) Download the Android image and PhoenixCard writing tool from the **Data download page of Orange Pi**. Please ensure that the version of the PhonenixCrad tool is **PhonixCard-4.2.8. Do not use PhonixCard software later than 4.2.8 to burn Android images**. Android images written by PhonixCard tools later than this version may have problems

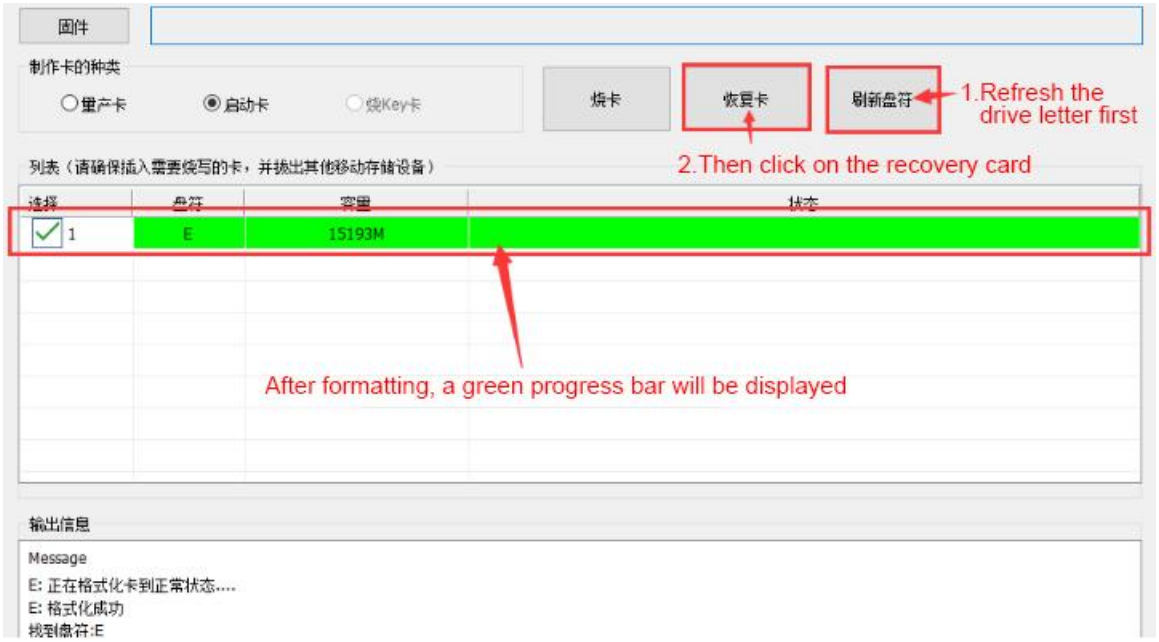
9) Then use the decompression software to decompress the compressed package of the downloaded Android image. In the decompressed file, the file ending with "**.img** "is the Android image file, the size of which is more than 1GB. If you do not know how to decompress the Android image package, you can install a **360 compression software** to decompress the image.

10) Then use the decompression software to decompress **PhonixCard4.2.8.zip**. The software does not need to be installed. Find PhoenixCard in the decompressed folder and open it



11) 打开 PhoenixCard 后，如 After PhoenixCard is opened, if the TF card is recognized normally, the TF card's drive letter and capacity will be displayed in the middle list. **Please make sure that the drive letter displayed is the same as that of the TF card you want to burn**. If it is not displayed, try to remove and insert the TF card, or click "**Refresh Drive Letter**" in PhoenixCard

12) After confirming the drive letter, format the TF card and click the "**Restore Card**" button in PhoenixCard. (If the "**Restore Card**" button is gray and cannot be pressed, click the "**Refresh Drive Letter**" button first.)



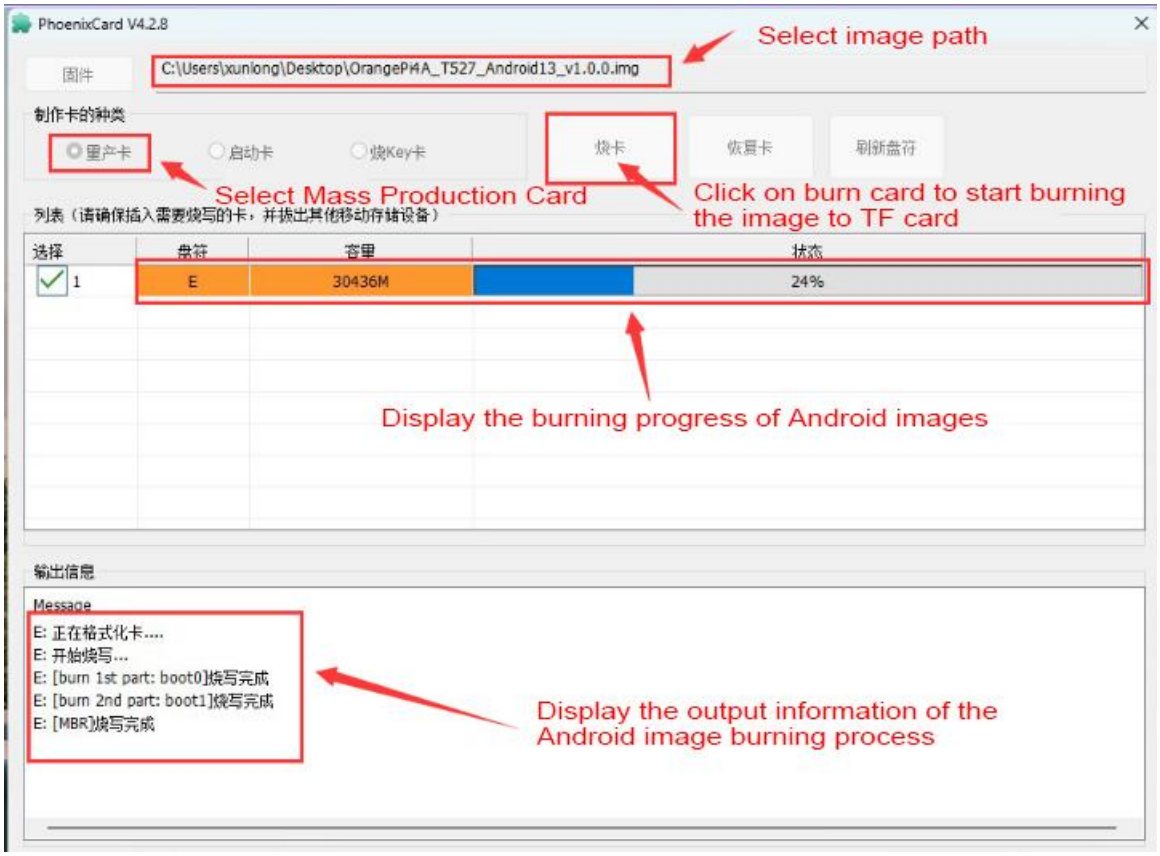
If there is any problem with formatting, please try to remove and insert the TF card and test again. If there is still a problem after removing and inserting the TF card, restart the Window computer or change another computer to try again.

13) Then start writing the Android image to the TF card

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

b. Select "**Mass Production Card**" in the "**Types of Card Production**" section

c. Then click the "**Burn Card**" button to start burning



14) After burning, the display of PhoenixCard is shown in the following figure. Click the "**Close**" button to exit PhoenixCard

15) Then insert the TF card into the development board. After powering on and starting the development board, the Android firmware in the TF card will be automatically burned to the eMMC of the development board. After the burning is completed, it will automatically shut down and the LED light on the development board will turn off

16) At this point, you can unplug the TF card and then power it back on, which will start the Android system in eMMC

2.9. Launch the Orange Pi development board

1) Insert the TF card with the burned image into the TF card slot of the Orange Pie development board

2) The development board has an HDMI interface, which can be connected to a TV or HDMI monitor through an HDMI to HDMI cable. If you purchase an LCD screen, you can also use the LCD screen to display the system interface of the development board.

3) Connect a USB mouse and keyboard to control the Orange Pie development board

4) The development board has an Ethernet port that can be plugged into a network cable for internet access

5) Connect a **high-quality** power adapter with a 5V/5A (5V/4A is also acceptable) USB Type C interface

Remember not to insert a power adapter with a voltage output greater than 5V, as it may burn out the development board.

Many unstable phenomena during the power on startup process of the system are basically caused by power supply problems, so a reliable power adapter is very important. If you notice continuous restarts during the startup process, please replace the power supply or Type C data cable and try again.

6) Then turn on the power adapter switch. If everything is normal, the HDMI monitor will be able to see the system startup screen

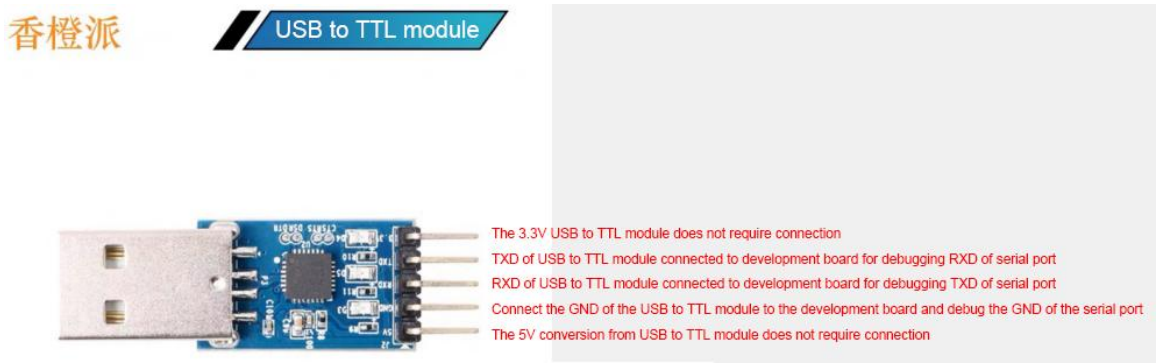
7) If you want to view the system's output information by debugging the serial port,

please connect the development board to the computer using a serial port cable. For the method of connecting the serial port, please refer to the section on **debugging the serial port usage**

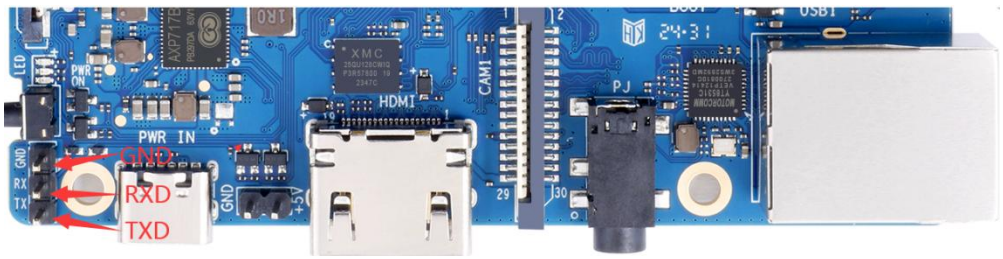
2. 10. Instructions for Debugging Serial Ports

2. 10. 1. Debugging serial port connection instructions

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



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



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 DuPont wires

a. Connect the GND of the USB to TTL module to the GND of the development board
b. **Connect the RX of the USB to TTL module 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 a USB to TTL module to a computer and Orange Pi development board is shown below

Schematic diagram of connecting a USB to TTL module to a 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 randomly connect the TX and RX of the serial port first. If there is no output from the test serial port, then switch the order of TX and RX. This way, there will always be a correct order.

2. 10. 2. Instructions for Debugging Serial Ports on Ubuntu Platform

There are many serial debugging software that can be used under Linux, such as Putty, Minicom, etc. Below is a demonstration of how to use Putty.

1) Firstly, insert the USB to TTL module into the USB interface of the Ubuntu computer. If the USB to TTL module is recognized as connected properly, the corresponding device node name can be seen in the **/dev** section of the Ubuntu PC. Remember this node name, which will be used later when setting up the serial port software

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

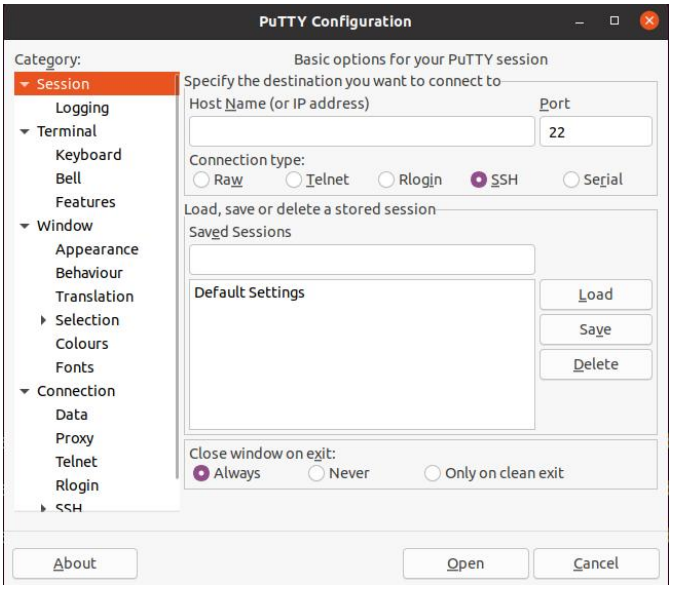
2) Then use the following command to install Putty on Ubuntu PC

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

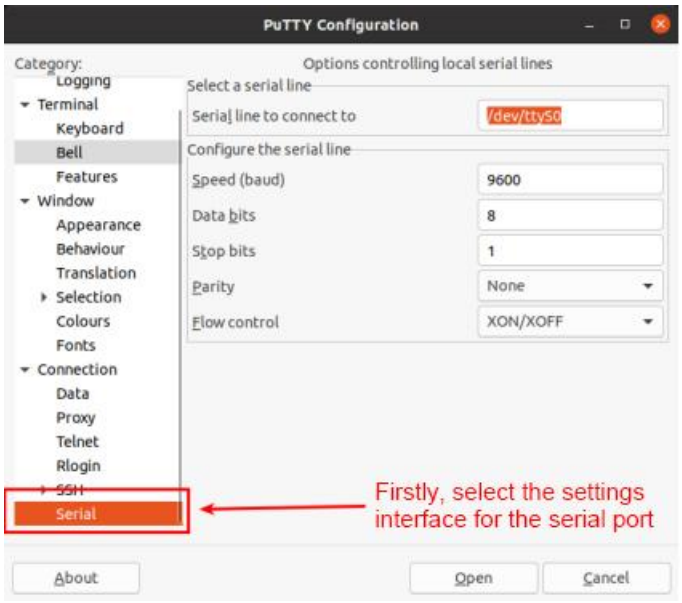
3) Then run putty, remember to **add sudo privileges**

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

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



5) Firstly, select the settings interface for the serial port

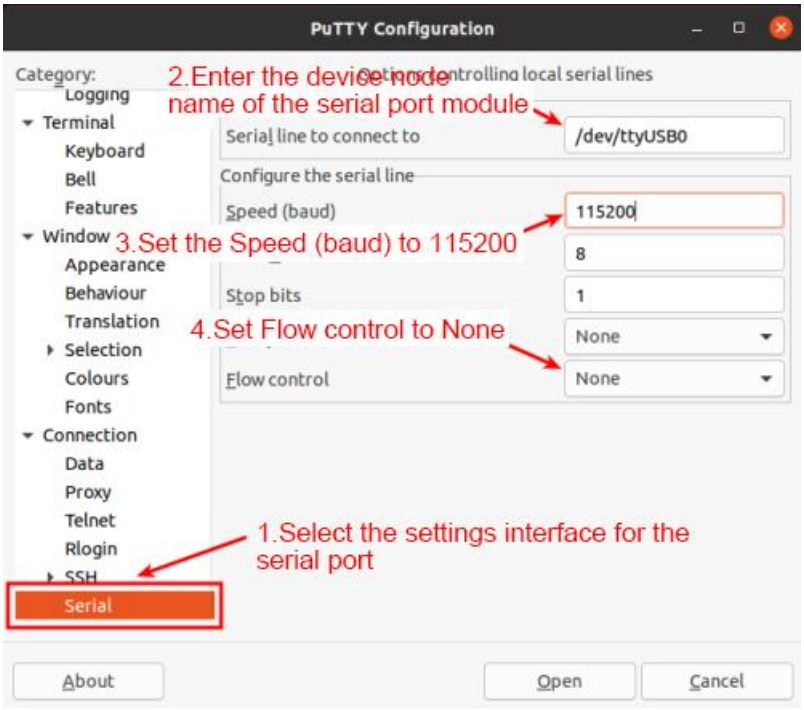


6) Then set the parameters of the serial port

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

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

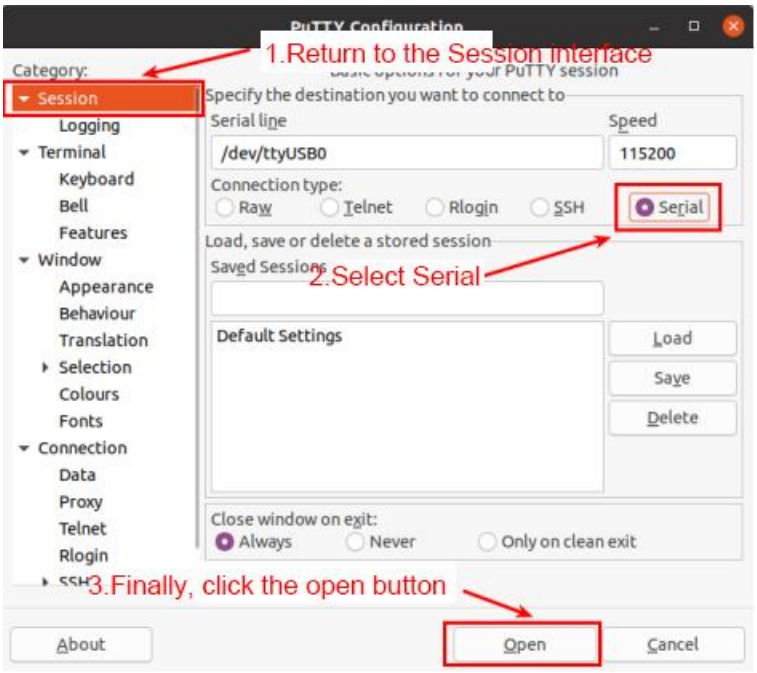
c. Set **Flow control** to **None**



7) After completing the settings on the serial port interface, return to the Session interface

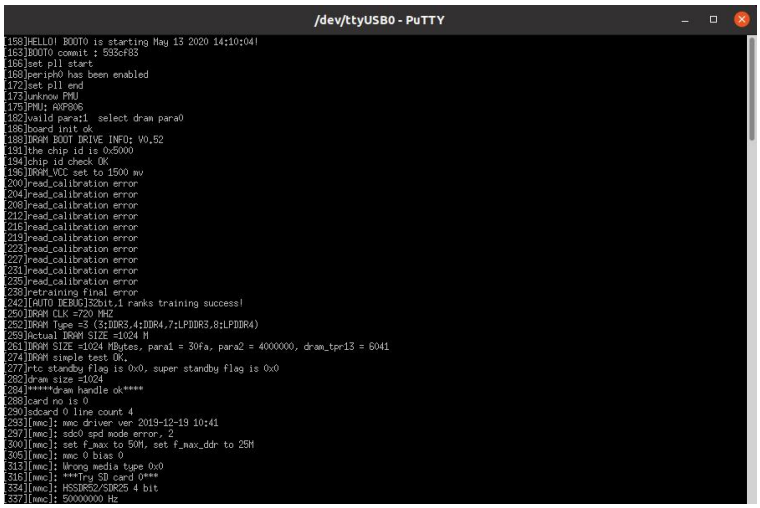
a. First, select **Connection type** as **Serial**

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. 10. 3. Instructions for Debugging Serial Ports on Windows Platform

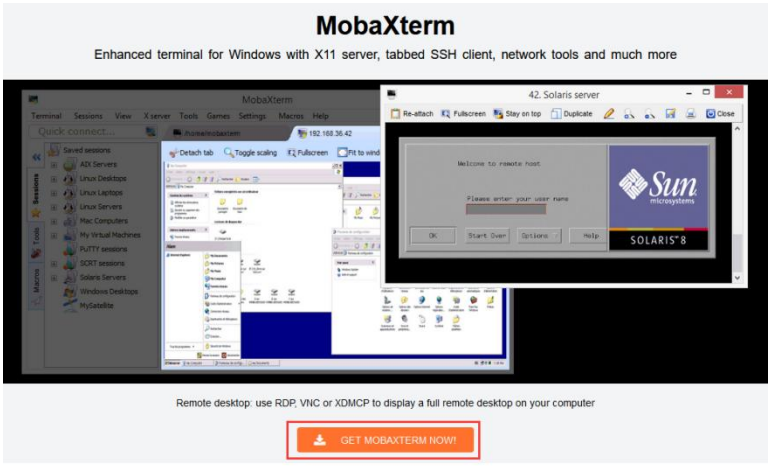
There are many serial debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. Below is a demonstration of how to use MobaXterm. This software has a free version and can be used without purchasing a serial number.

1) Download MobaXterm

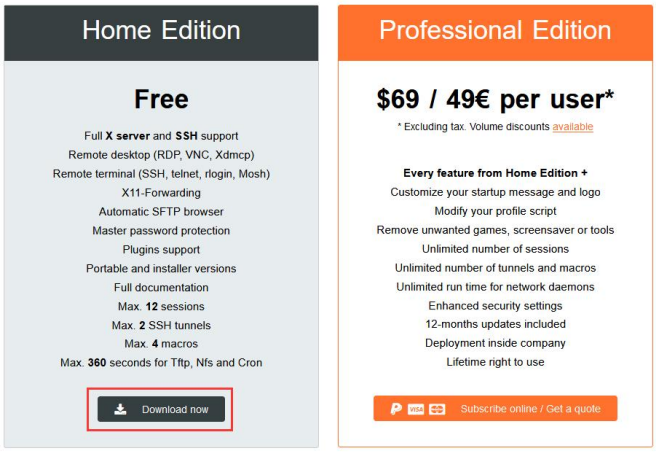
a. The download link for MobaXterm is as follows

https://mobaxterm.mobatek.net/

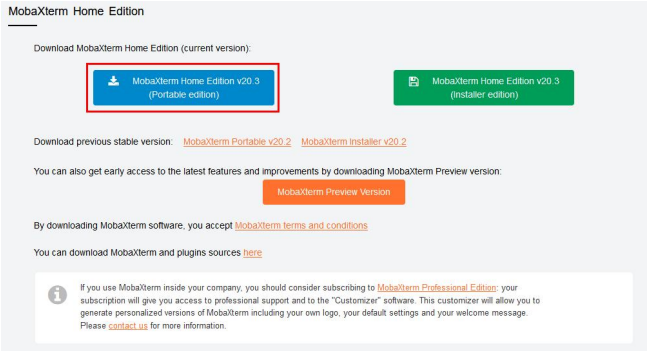
b. After entering the MobaXterm download webpage, click **GET XOBATERM NOW!**



c. Then choose to download the Home version



d. Then select the Portable version, and after downloading, there is no need to install it. Simply open it and you can use it



2) After downloading, use decompression software to extract the downloaded compressed file and obtain the executable software of MobaXterm. Then double-click to open it

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

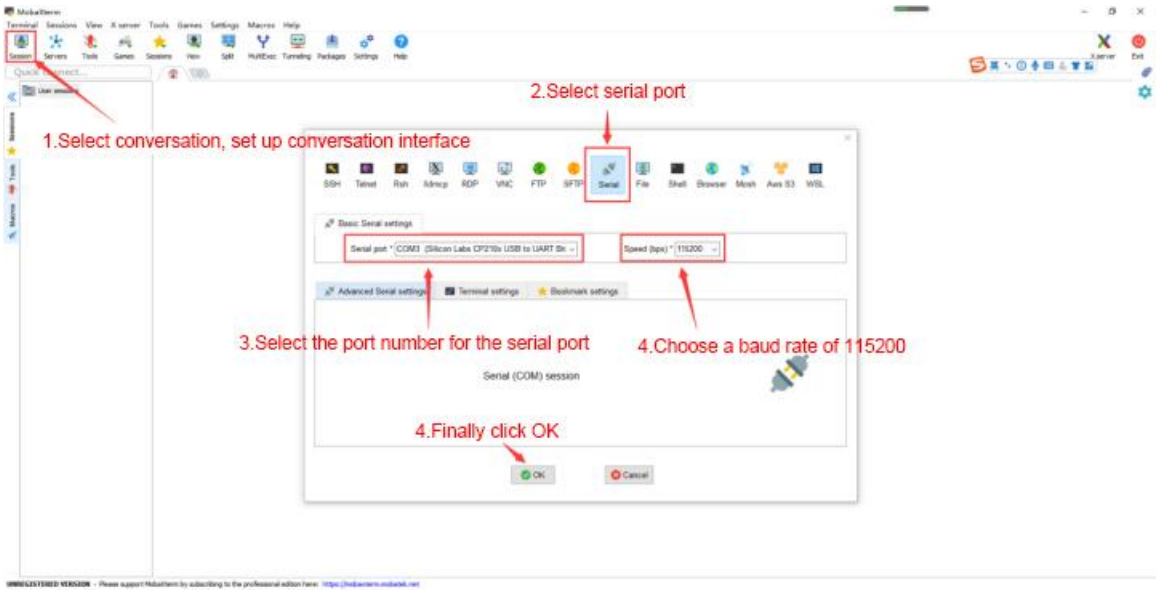
a. Open the session settings interface

b. Select serial port type

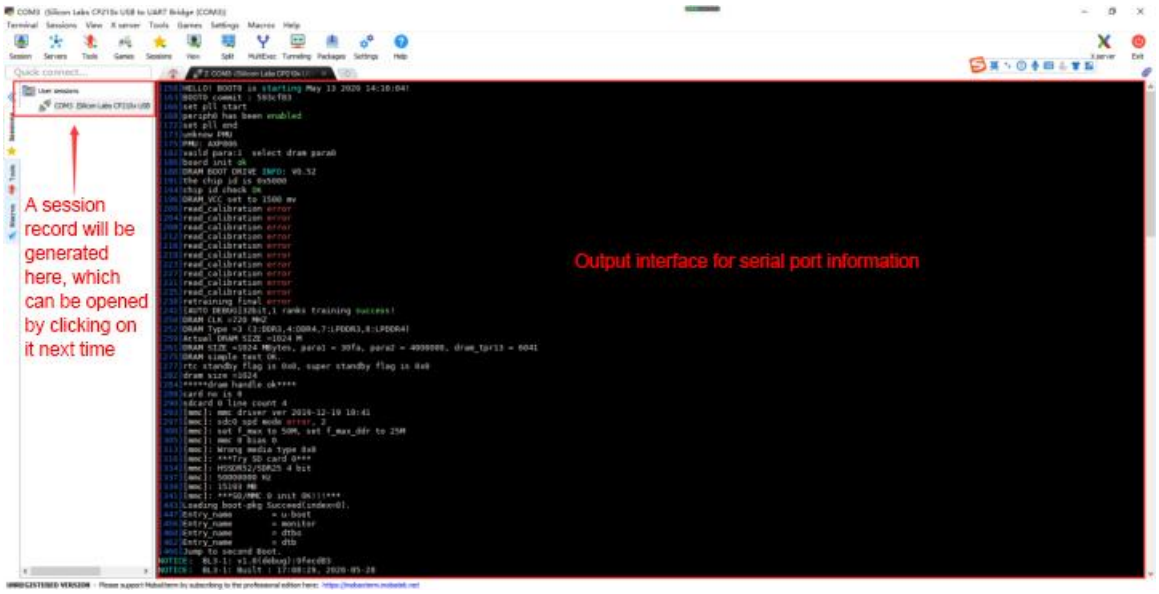
c. Select the port number for the serial port (choose the corresponding port number according to the actual situation). If you cannot see the port number, please use **360 Driver Master** to scan and install the driver for the USB to TTL serial port chip

d. Choose a baud rate of **115200** for the serial port

e. Finally, click the "**OK**" button to complete the setup



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



2.11. Instructions for powering the 5V pin in the 40 pin interface of the development board

The recommended power supply method for the development board is to use a 5V/5A or 5V/4A Type C interface power cord plugged into the Type C power interface of the development board for power supply. If you need to use the 5V pin in the 40 pin interface to power the development board, please ensure that the power cord used can meet the power supply requirements of the development board. If there is unstable usage, please switch back to the Type C power supply.

1) Firstly, it is necessary to prepare a power cord as shown in the following diagram

The power cord shown in the picture can be purchased on Taobao, please search and purchase it yourself.

2) Use the 5V pin in the 40 pin interface to power the development board, and connect the power cord as follows

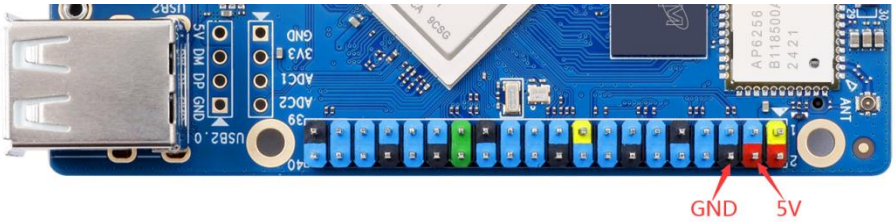
a. The USB A port of the power cord shown in the above figure needs to be plugged into a 5V/5A power adapter connector (**it is not recommended to plug it into the USB port of the computer for power supply, as it may be unstable to use if there are too many peripherals connected to the development board**)

b. The red DuPont wire needs to be plugged into the 5V pin of the 40 pin interface

on the development board

c. The black DuPont wire needs to be plugged into the GND pin of the 40 pin interface

d. The positions of the 5V pin and GND pin of the 40 pin interface in the development board are shown in the following figure. **Remember not to connect them in reverse**

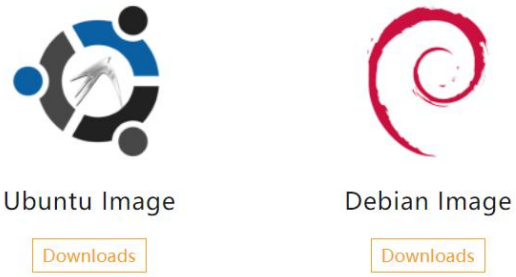


3. Instructions for using Debian/Ubuntu Server and Gnome desktop system

3. 1. Supported Linux image types and kernel versions

Linux image type	Kernel version	Server version	Desktop version
Ubuntu 22.04 - Jammy	Linux5.15	support	support
Debian 12 - Bookworm	Linux5.15	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 following description, **Ubuntu images and Debian images are generally referred to as Linux images**.



The naming convention for Linux images is:

Development board model _ version number _ Linux release type _ release code _ server or desktop _ kernel version

a. **Development board model:** They are all **Orangepi4a**. The model names of different development boards are generally different. Before burning the image, please ensure 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 digit of the version number of the Linux image on the development board is even.

c. **Types of Linux distributions:** Currently supports **Ubuntu** and **Debian**. Due to Ubuntu's origin from Debian, there is generally not much difference in usage between the two systems. But there are still some differences in the default configuration and command usage of some software. In addition, Ubuntu and Debian each maintain their own supported software repositories, and there are also some differences in the supported installable software packages. These require personal experience to gain a deeper understanding. For more details, you can refer to the official documentation provided by Ubuntu and Debian.

d. **Release code:** Used to distinguish between different versions of specific Linux distributions such as Ubuntu or Debian. Among them, **jammy** is the Ubuntu distribution, referring to Ubuntu 22.04. The biggest difference between different versions is that many of the software in the software repository maintained by the new version of Ubuntu system are newer than those in the old version, such as Python and GCC compilation toolchains. **bookworm** is the specific version code for Debian, with **bookworm** representing Debian12.

e. **Server or Desktop:** Used to indicate whether the system has a desktop environment. If it is a **server**, it means that the system does not have a desktop

environment installed. The storage space and resources occupied by the image are relatively small, and the system is mainly operated and controlled using the command line. If it is **desktop_gnome**, it means that the system has the GNOME desktop environment installed by default. The image occupies a relatively large amount of storage space and resources, and can be operated through the interface with a monitor, mouse, and keyboard. Of course, the desktop version of the system can also be operated through the command line like the server version.

f. **Kernel version:** Used to represent the version number of the Linux kernel, currently supporting **linux5.15**.

3.2. linux kernel driver adaptation situation

Function	Linux5.15
HDMI video	OK
HDMI audio	OK
USB2.0 x 4	OK
TF card startup	OK
eMMC	OK
NVME SSD recognition	OK
Gigabit Ethernet	OK
WIFI	OK
Bluetooth	OK
RTC chip	OK
Earphone audio	OK
LCD screen	OK
EDP	OK
CAM1	Kernel driver OK, 3A not tuned
CAM2	Kernel driver OK, 3A not tuned
LED lamp	OK
40 pin GPIO	OK
40 pin I2C	OK

40 pin SPI	**OK**
40 pin UART	**OK**
40 pin PWM	**OK**
Key	**OK**
Temperature sensor	**OK**
hardware watchdog	**OK**
Mali GPU	**NO**
Video Encoding and Decoding	**NO**

3. 3. Explanation of linux Command Format in This Manual

1) All commands that need to be entered in the Linux system in this manual will be enclosed in the boxes below

```
```

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

```
```

2) Description of prompt types before commands

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

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

b. **orangepi@orangepi:~$** The prompt indicates that this command was entered in the **linux system of the development board**, and the last **$** of the prompt indicates that the current user of the system is a regular user. When executing privileged commands, **sudo** needs to be added

c. **root@orangepi:~#** The prompt indicates that this command was 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 and can execute any command you want

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

e. **root@test:~#** The prompt indicates that this command was entered on an Ubuntu PC or Ubuntu virtual machine, not on the Linux system of the development board. The **#** at the end of the prompt indicates that the current system user is the root user and can execute any command they want

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

a. As shown below, **the bold black part represents** the commands that need to be inputted, and the content below the commands is the output (some commands have output, while others may not). This part of the content does not need to be inputted

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

b. As shown below, some commands that cannot be written on one line will be moved to the next line, and any black and bold parts are commands that need to be entered. When these commands are inputted on a single line, the '\' at the end of each line needs to be removed, which is not part of the command. Also, different parts of the command have spaces, please don't miss them

```
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 number	password

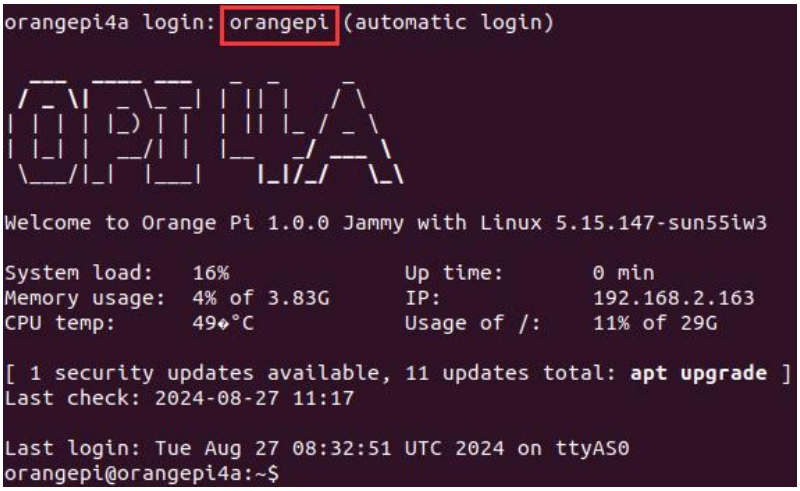
root	**orangepi**
orangepi	**orangepi**

Note that when entering the password, the specific content of the entered password will not be displayed on the screen. Please do not assume that there is any malfunction. After entering, simply press Enter.

When the password prompt is incorrect 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, do not suspect that the password is incorrect, but instead look for other reasons.

3. 4. 2. Method for setting up automatic login for linux system terminals

1) The linux system defaults to automatically logging into the terminal, and the default login username is **orangepi**



2) The following command can be used 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 to the terminal

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

4) The following command can be used to reset the automatic login of Orangepi users to

the terminal

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

3. 4. 3. linux Desktop System Automatic Login Instructions

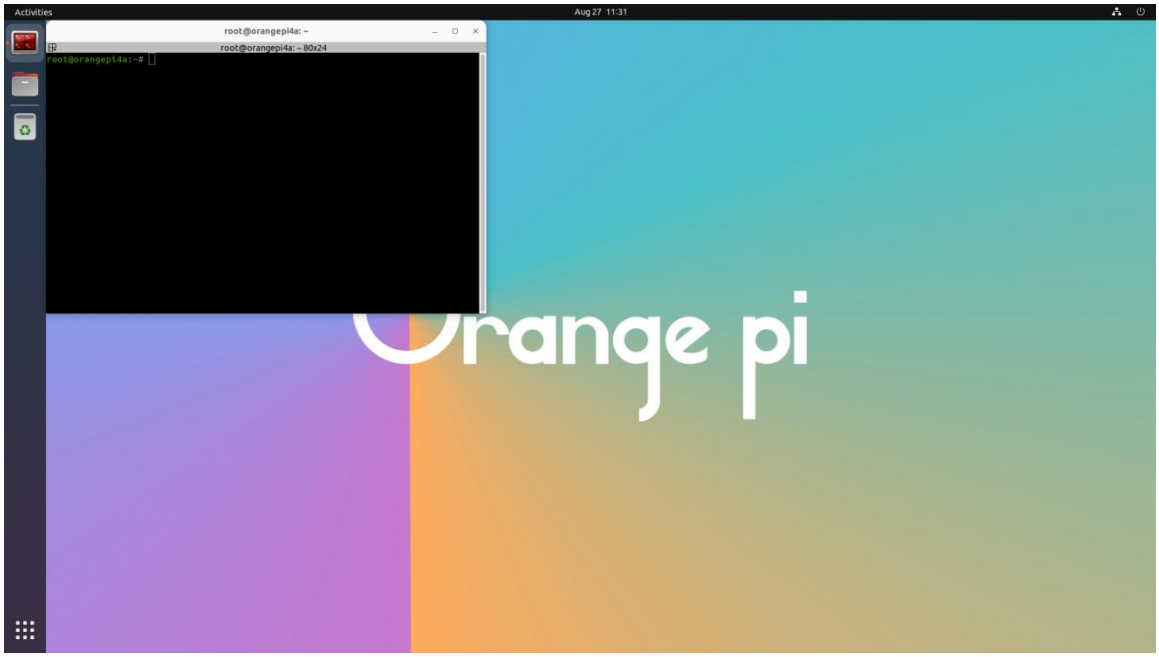
1) After the desktop version system starts, it will automatically log in to the desktop without entering a password

3. 4. 4. Linux Desktop System Root User Automatic Login Setting Method

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 it will automatically log in to the desktop as the root user



3) Execute the following command to reset the desktop system to use Orangepi for automatic user login

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

3. 4. 5. How to disable the desktop in Linux desktop system

1) First enter the following command in the command line, **please remember to add sudo permissions**

```
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 permissions**

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

3. 5. Onboard LED light test instructions

1) There are three LED lights on the development board, one green light, one red light, and one PCIe indicator light. Their locations are shown in the figure below:

2) As long as the development board is powered on, the red LED light will be always on. This is controlled by hardware and cannot be turned off by software.

3) The green LED light will keep flashing after the kernel starts, which is controlled by software.

4) The PCIe indicator will flash when there is data transmission on the PCIe interface.

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

Note: The following operations must be performed as the root user.

a. First enter the green light settings directory

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

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

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

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

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

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

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

3.6. Linux system rootfs partition capacity operation instructions in TF card

3.6.1. The capacity of the rootfs partition in the TF card will be automatically expanded at the first startup

1) After burning the Linux image of the development board to the TF card, you can check the usage of the TF card capacity in the **Ubuntu computer**. The steps are as follows:

Note that not doing this step will not affect the automatic expansion of the Linux system on the development board. Here I just want to explain how to check the capacity of the TF card after burning the Linux image to the TF card.

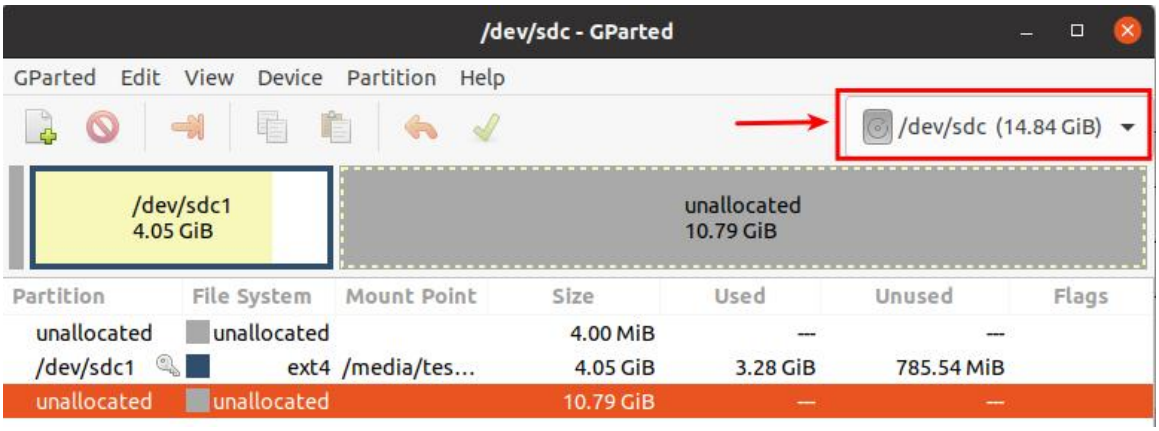
a. First install the gparted software in the Ubuntu computer

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

b. Then open gparted

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

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



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

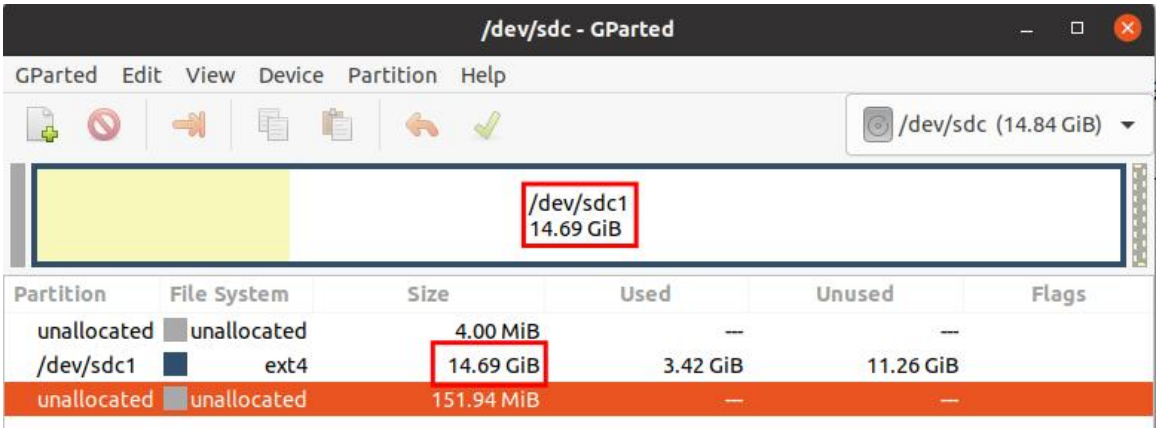
2) Then you can insert the TF card with the Linux system burned into the development board to start it. When the TF card starts the Linux system for the first time, the **orangepi-resize-filesystem.service** systemd service will be used to call the **orangepi-resize-filesystem** script to automatically expand the rootfs partition**, so there is no need to expand it manually.**

3) After logging into 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 TF 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 booting the Linux system for the first time, we can also remove the TF card from the development board and reinsert it into the **Ubuntu computer**, and then use

gparted to check the status of the TF card again. As shown in the figure below, the capacity of the rootfs partition (/dev/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 kernel images and other files, so there is no problem of expanding the BOOT partition.

3.6.2. How to disable automatic expansion of rootfs partition capacity in TF card

1) First, burn the Linux image of the development board to the TF card in the Ubuntu computer (not available on Windows), **then unplug and re-insert the TF card.**

2) Then the Ubuntu computer will generally automatically mount the TF card partition. If the automatic mounting is normal, you can see the following output using the ls command

```
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 in the Ubuntu computer

```
test@test:~$ sudo -i
[sudo] test 的密码：
root@test:~#
```

4) Then enter the root directory of the Linux system in the TF 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 TF card, then pull out the TF card and insert it into the development board to start. When the Linux system starts, if it detects the **.no_rootfs_resize** file in the **/root** directory, it will no longer automatically expand the rootfs.

6) After disabling automatic expansion of rootfs, you can enter the Linux system and 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 TF card, indicating that the automatic expansion of rootfs is successfully disabled.

```
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 expand the capacity of the rootfs partition in the TF card, just execute the following command and restart the Linux system of the development board.

Please execute the following command 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 TF 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. How to manually expand the rootfs partition capacity in the TF card

If the total capacity of the TF card is large, such as 128GB, and you do not want the Linux system rootfs partition to use all the capacity of the TF 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 TF card can be used for other purposes. Then you can use the content introduced in this section to manually expand the capacity of the rootfs partition in the TF.

1) First, burn the Linux image of the development board to the TF card in the Ubuntu computer (not available on Windows), **then unplug and re-insert the TF card.**

2) Then the Ubuntu computer will generally automatically mount the TF card partition. If the automatic mounting is normal, you can see the following output using the ls command

```
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 in the Ubuntu computer

```
test@test:~$ sudo -i
[sudo] test 的密码：
root@test:~#
```

4) Then enter the root directory of the Linux system in the TF 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
```

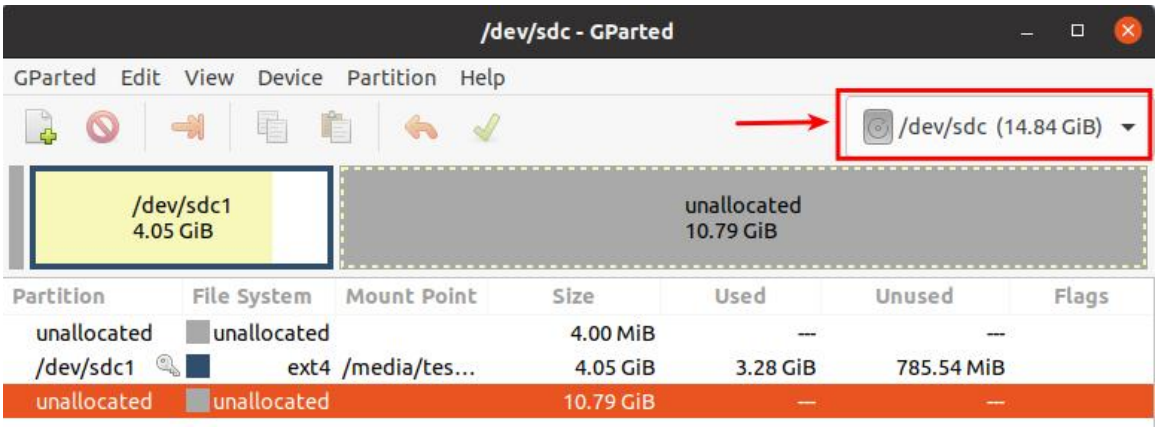
5) Then install the gparted software in the Ubuntu computer

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

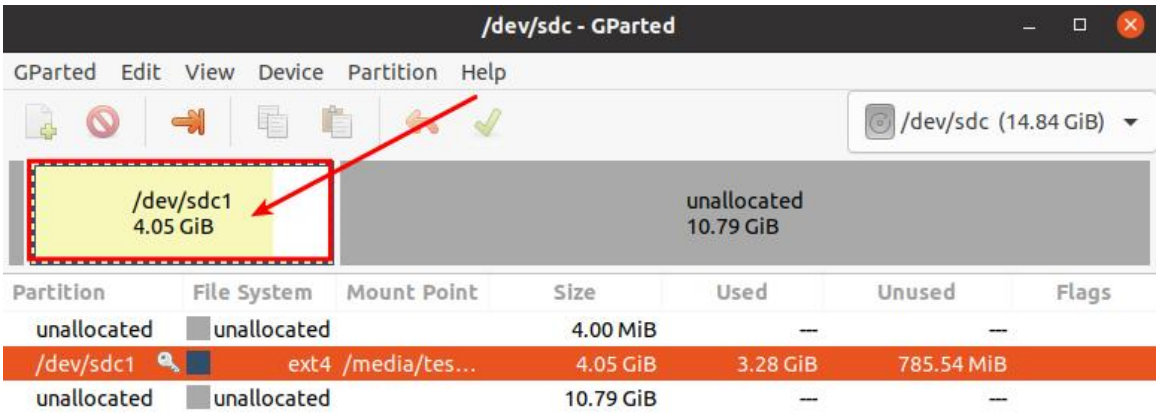
6) Then open gparted

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

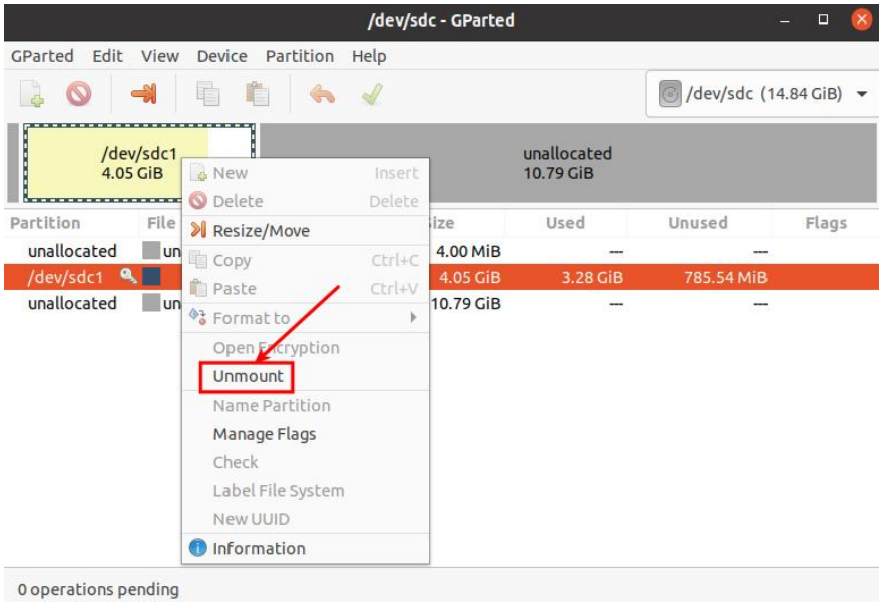
7) After opening gparted, you can select the TF card in the upper right corner, and then you can see the usage of the TF card capacity. The figure below shows the TF card after burning the Linux desktop system. It can be seen that although the total capacity of the TF card is 16GB (displayed as 14.84GiB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



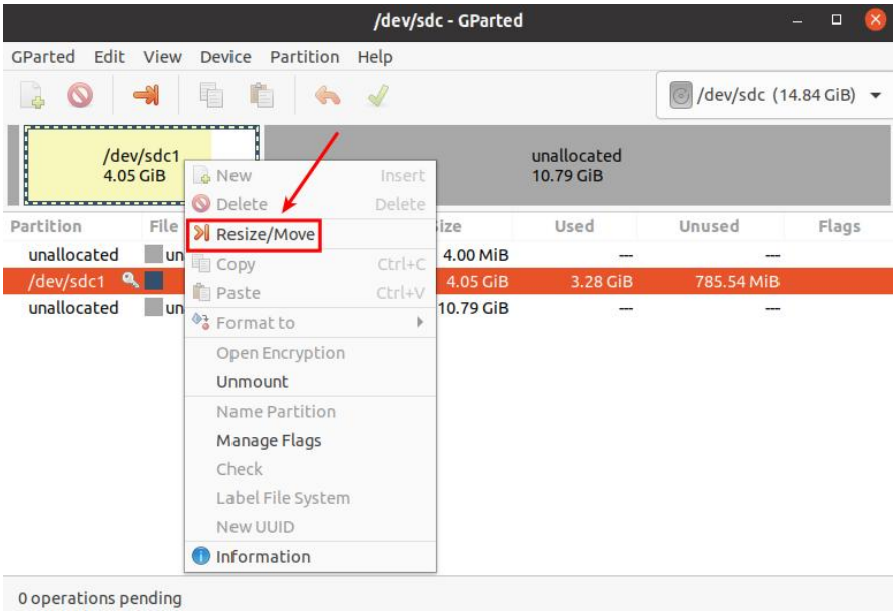
8) Then select the rootfs partition (/dev/sdc1)



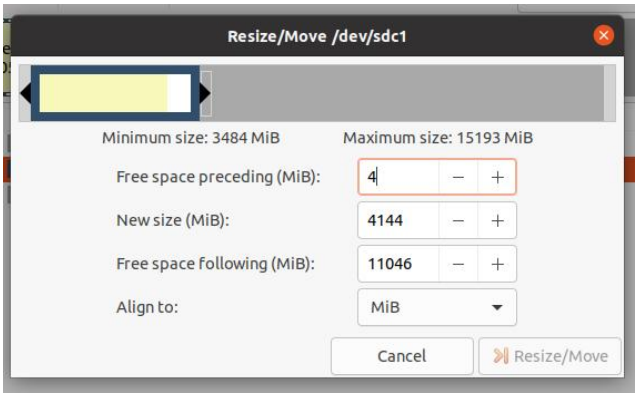
9) Click the right button of the mouse again to see the operation options shown in the figure below. If the TF card has been mounted, you need to Umount the rootfs partition of the TF card first.



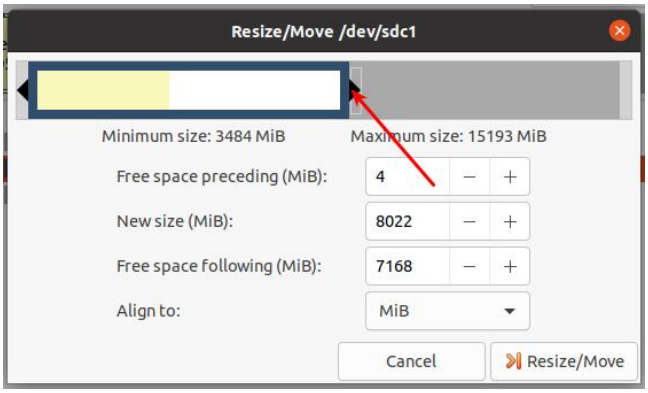
10) Then select the rootfs partition again, right-click, and select **Resize/Move** to start expanding the size of the rootfs partition



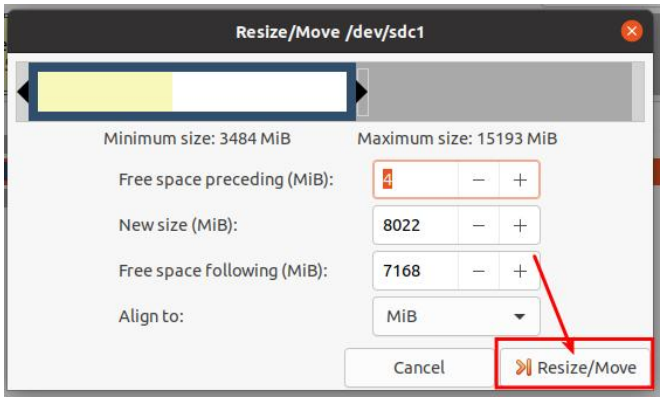
11) After the **Resize/Move** option is turned on, the following setting interface will pop up



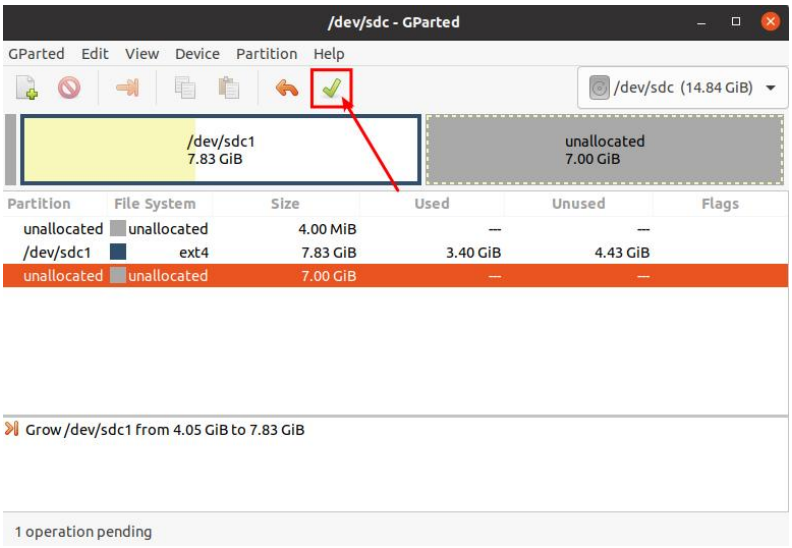
12) Then you can directly drag the position shown in the figure below to set the capacity, or you can set the size of the rootfs partition by setting the number in **New size(MiB)**



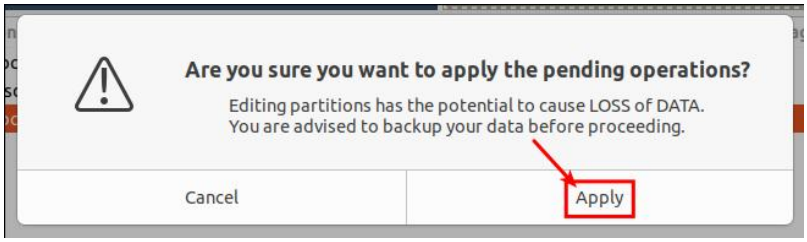
13) After setting the capacity, click **Resize/Move** in the lower right corner.



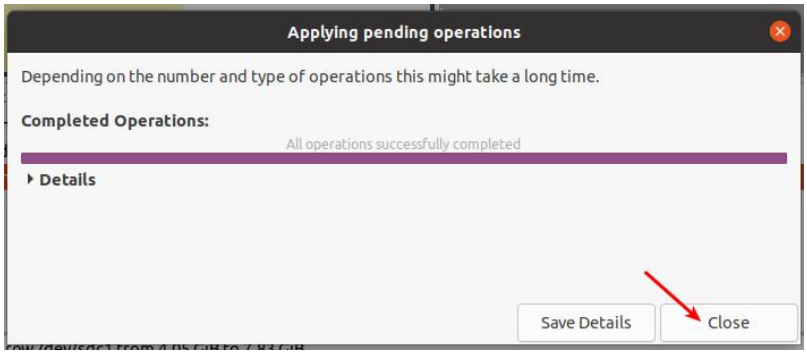
14) After final confirmation, click the green √ as shown below



15) Then select **Apply** to officially start expanding the capacity of the rootfs partition



16) After the expansion is completed, click **Close**.



17) Then you can unplug the TF card and insert it into the development board to start. After entering the Linux system of the development board, if you use the **df -h** command to see that the size of the rootfs partition is consistent with the size set previously, it means that the manual expansion is successful.

```
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 TF card

After configuring the application or other development environment in the Linux system of the TF card, if you want to back up the Linux system in the TF 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 TF card you want to operate into the Ubuntu computer (not available on Windows)

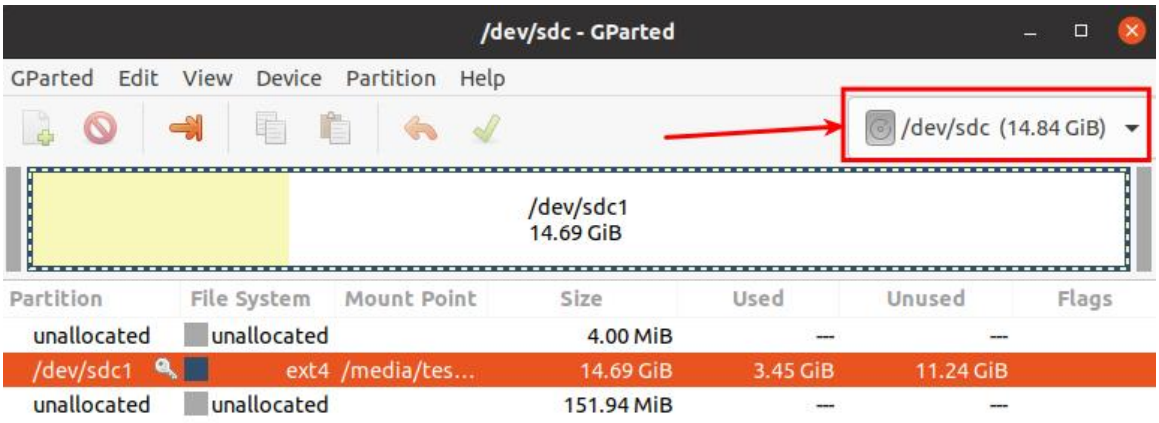
2) Then install the gparted software in the Ubuntu computer

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

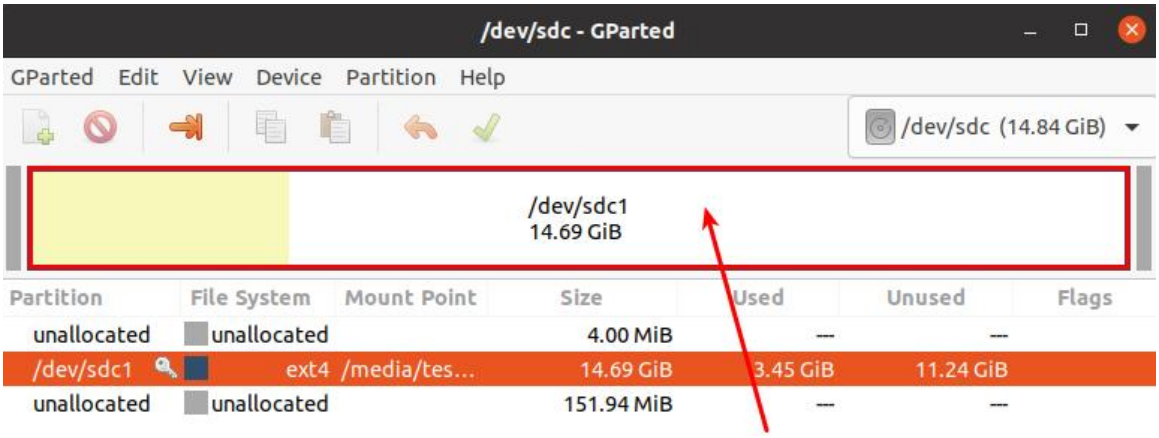
3) Then open gparted

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

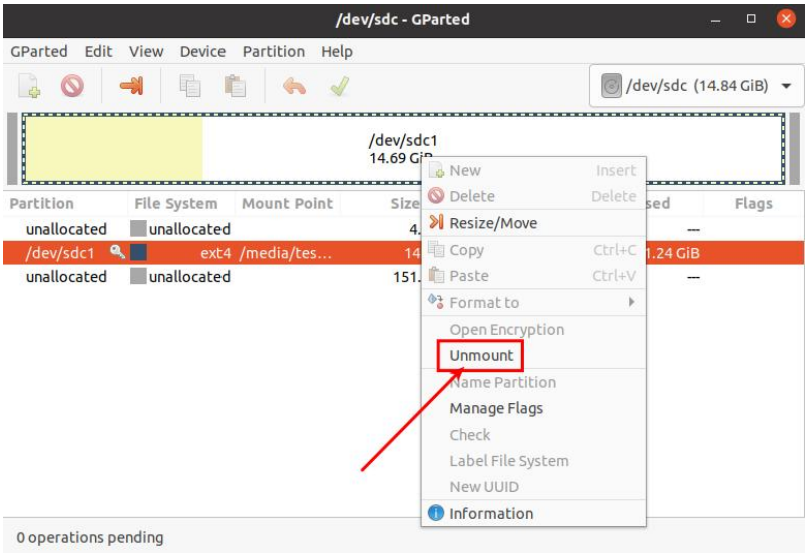
4) After opening gparted, you can select the TF card in the upper right corner, and then you can see the usage of the TF card capacity



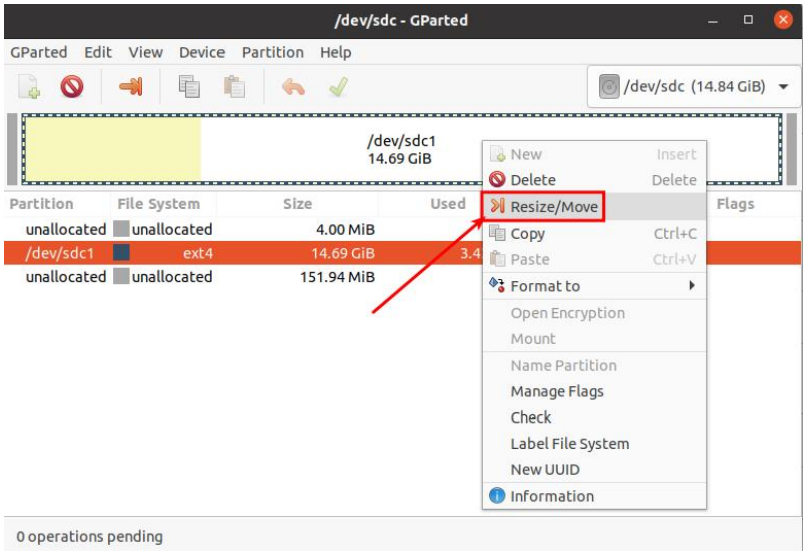
5) Then select the rootfs partition (/dev/sdc1)



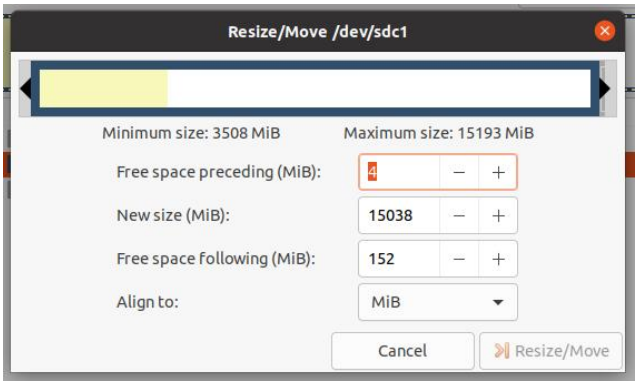
6) Click the right button of the mouse again to see the operation options shown in the figure below. If the TF card has been mounted, you need to Umount the rootfs partition of the TF card first.



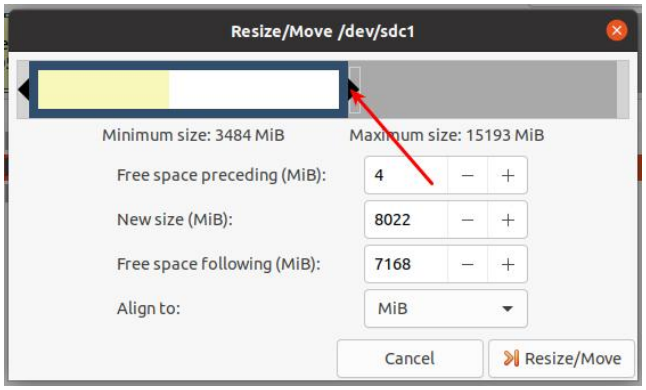
7) Then select the rootfs partition again, right-click, and select **Resize/Move** to start setting the size of the rootfs partition



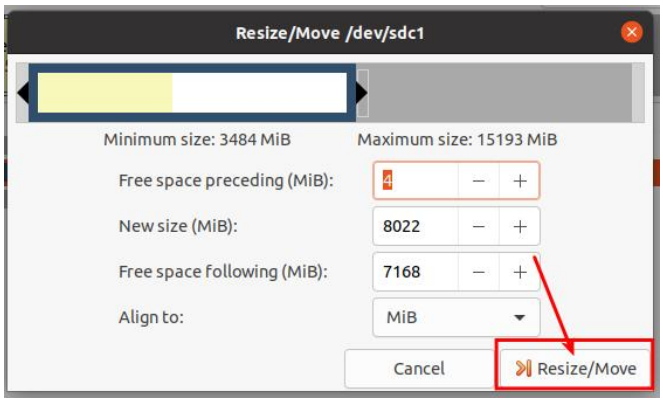
8) After the **Resize/Move** option is turned on, the following setting interface will pop up



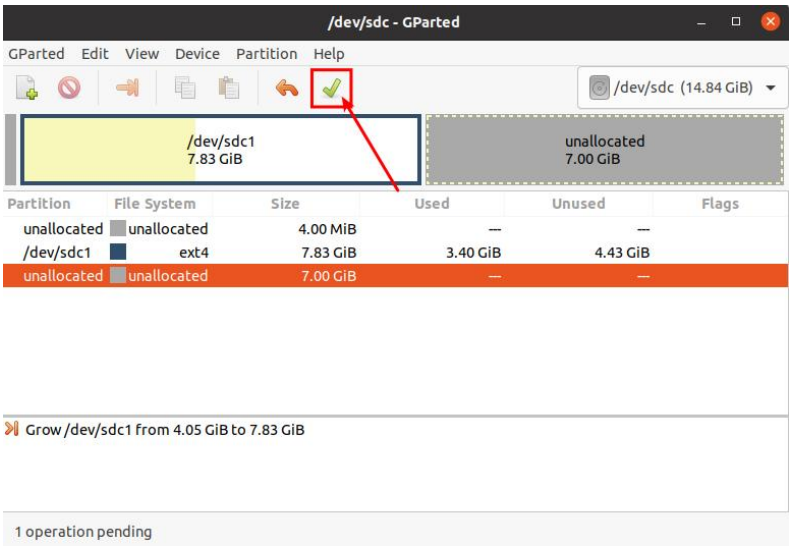
9) Then you can directly drag the position shown in the figure below to set the capacity, or you can set the size of the rootfs partition by setting the number in **New sieze(MiB)**



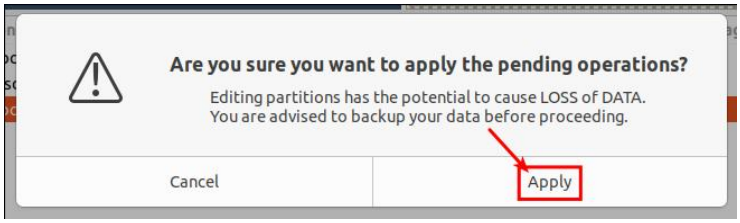
10) After setting the capacity, click **Resize/Move** in the lower right corner.



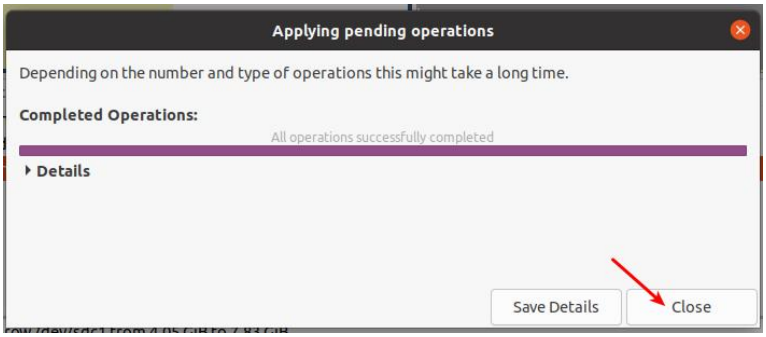
11) After final confirmation, click **the green √** as shown below



12) Then select **Apply** to officially start expanding the capacity of the rootfs partition



13) After the expansion is completed, click **Close**.



14) Then you can unplug the TF card and insert it into the development board to start. After entering the Linux system of the development board, if you use the **df -h** command to see that the size of the rootfs partition is the same as the size set previously, it means that the capacity reduction is successful.

```
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, plug one end of the network cable into the Ethernet port of the development board, and the other end of the network cable into the router, and make sure the network is unobstructed.

2) After the system starts, the IP address will be automatically assigned to the Ethernet card through DHCP, **and no other configuration is required**

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

```
orangepi@orangepi:~$ ip a s eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 3a:3a:57:82:eb:1f brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.163/24 brd 192.168.2.255 scope global dynamic noprefixroute eth0
```

```
        valid_lft 42902sec preferred_lft 42902sec
    inet6 fdcd:e671:36f4::a39/128 scope global dynamic noprefixroute
        valid_lft 42904sec preferred_lft 42904sec
    inet6 fdcd:e671:36f4:0:7b67:e74e:f0e1:849a/64 scope global temporary dynamic
        valid_lft 604504sec preferred_lft 86095sec
    inet6 fdcd:e671:36f4:0:d098:7f17:6cea:4de4/64 scope global mngtmpaddr
noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::cc72:d313:9846:a5e0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

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

1. Connect an HDMI display, 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 debug serial port terminal to view the IP address

3. If there is no debug serial port and no HDMI display, you can also view the IP address of the development board network port through the router's management interface. However, this method often causes some people to be unable to see the IP address of the development board normally. If you can't see it, the debugging method is as follows:

A) First check whether the Linux system has started normally. If the green light on the development board flashes, it is generally started normally. If only the red light is on, or neither the red light nor the green light is 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 (there are many problems with routers, such as the router cannot allocate IP addresses normally, or the IP address has been allocated normally but cannot be seen in the router);

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

It should also be noted that the development board DHCP automatically assigns IP addresses without any settings.

4) The command to test network connectivity is as follows. The **ping** command can be interrupted by pressing **Ctrl+C**

```
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. This method may cause problems when connecting to the WIFI network.

3.7.2.1. Server version image connects to WIFI through command

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

Log in to the Linux system first. There are three ways to do this:

a. If the development board is connected to the network cable, you can log in to the Linux system remotely through SSH

b. If the development board is connected to the debug serial port, you can use the serial terminal to log in to the Linux system

c. If the development board is connected to the HDMI display, you can log in to the Linux system through the HDMI display terminal

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

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

```
root@orangepi:~# nmcli dev wifi
IN-USE  BSSID              SSID                MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
        28:6C:07:6E:87:2E  [redacted]_orangepi  Infra  9     260 Mbit/s  97      ▂▄▆█  WPA1 WPA2
        D8:D8:66:A5:BD:D1  [redacted]           Infra  10    270 Mbit/s  90      ▂▄▆█  WPA1 WPA2
        A0:40:A0:A1:72:20  [redacted]           Infra  4     405 Mbit/s  82      ▂▄▆█  WPA2
        28:6C:07:6E:87:2F  [redacted]_orangepi_5G Infra  149   540 Mbit/s  80      ▂▄▆_  WPA1 WPA2
        CA:50:E9:89:E2:44  [redacted]           Infra  1     130 Mbit/s  79      ▂▄▆_  WPA1 WPA2
        A0:40:A0:A1:72:31  [redacted]           Infra  100   405 Mbit/s  67      ▂▄▆_  WPA2
        D4:EE:07:08:A9:E0  [redacted]           Infra  4     130 Mbit/s  55      ▂▄__  WPA1 WPA2
        88:C3:97:49:25:13  [redacted]           Infra  6     130 Mbit/s  52      ▂▄__  WPA1 WPA2
        00:BD:82:51:53:C2  [redacted]           Infra  12    130 Mbit/s  49      ▂▄__  WPA1 WPA2
        C0:61:18:FA:49:37  [redacted]           Infra  149   270 Mbit/s  47      ▂▄__  WPA1 WPA2
        04:79:70:8D:0C:B8  [redacted]           Infra  153   270 Mbit/s  47      ▂▄__  WPA2
        04:79:70:FD:0C:B8  [redacted]           Infra  153   270 Mbit/s  47      ▂▄__  WPA2
        9C:A6:15:DD:E6:0C  [redacted]           Infra  10    270 Mbit/s  45      ▂▄__  WPA1 WPA2
        B4:0F:3B:45:D1:F5  [redacted]           Infra  48    270 Mbit/s  45      ▂▄__  WPA1 WPA2
        E8:CC:18:4F:7B:44  [redacted]           Infra  157   135 Mbit/s  45      ▂▄__  WPA1 WPA2
        B0:95:8E:D8:2F:ED  [redacted]           Infra  11    405 Mbit/s  39      ▂▄__  WPA1 WPA2
        C0:61:18:FA:49:36  [redacted]           Infra  11    270 Mbit/s  24      ▂___  WPA1 WPA2
root@orangepi:~#
```

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

a. **wifi_name** needs to be replaced with the name of the WIFI hotspot you want to connect to

b. **wifi_passwd** needs to be replaced with 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'.
```

3) Use the **ip addr show wlan0** command to view the IP address of the wifi

```
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
```

4) Use the **ping** command to test the connectivity of the WiFi network. The **ping** command can be interrupted by pressing the **Ctrl+C**shortcut key.

```
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 version image connects to WIFI through a graphical method

1) Log in to the Linux system first. There are three ways to do this:

a. If the development board is connected to the network cable, you can log in to the Linux system remotely through ssh

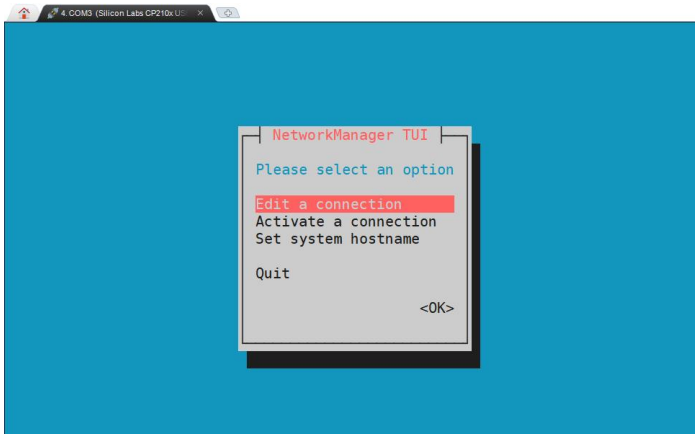
b. b. If the development board is connected to the debug serial port, you can use the serial terminal to log in to the Linux system (use MobaXterm as the serial software, and the graphical interface cannot be displayed using minicom)

c. c. If the development board is connected to an HDMI display, you can log in to the Linux system through the HDMI display terminal

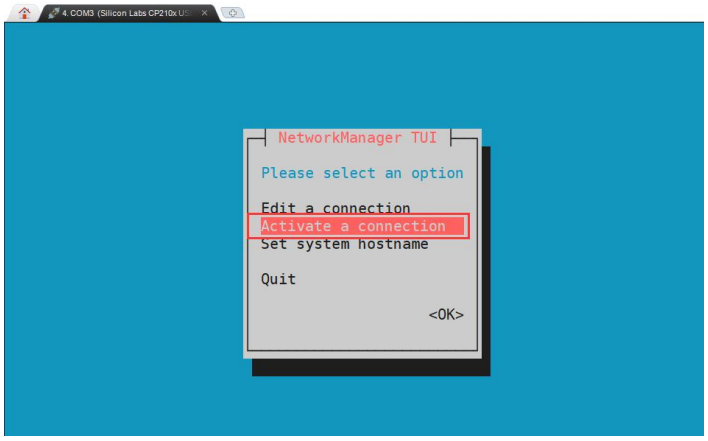
2) Then enter the nmtui command in the command line to open the wifi connection interface

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

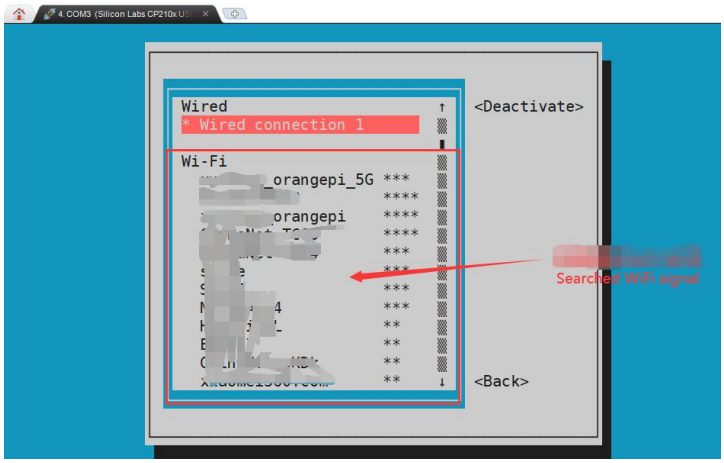
3) Enter the nmtui command to open the interface as shown below



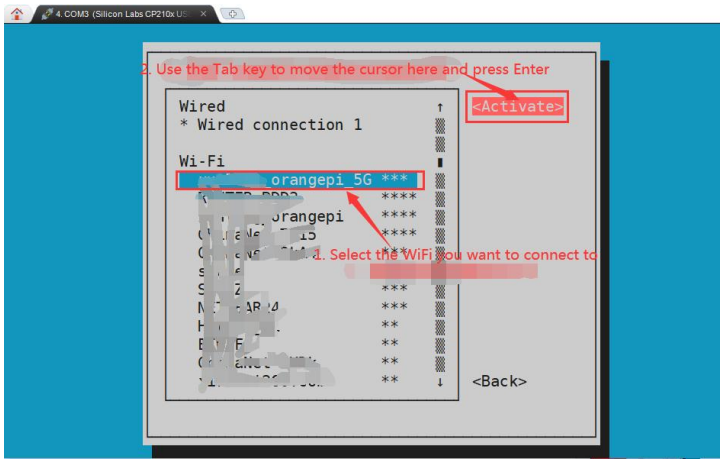
4) Select **Activate a connect** and press Enter



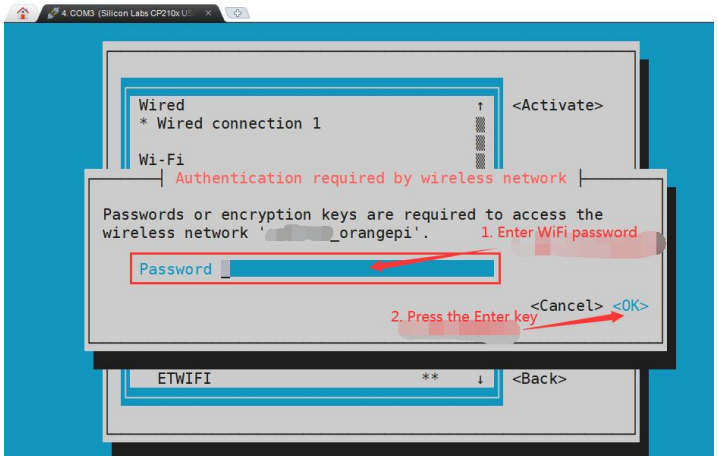
5) Then you can see all the searched WIFI hotspots



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

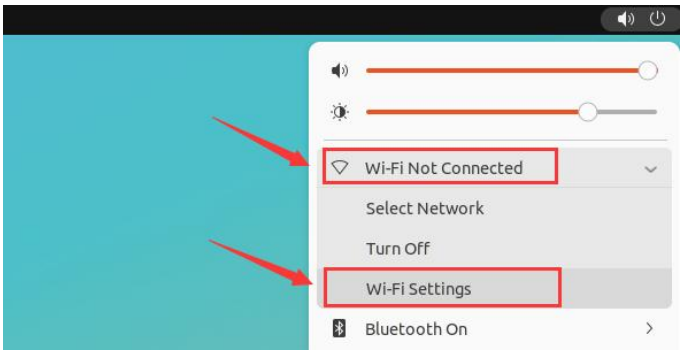
10) Use the **ping** command to test the connectivity of the WiFi network. The **ping** command can be interrupted by pressing the **Ctrl+C** shortcut key.

```
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
```

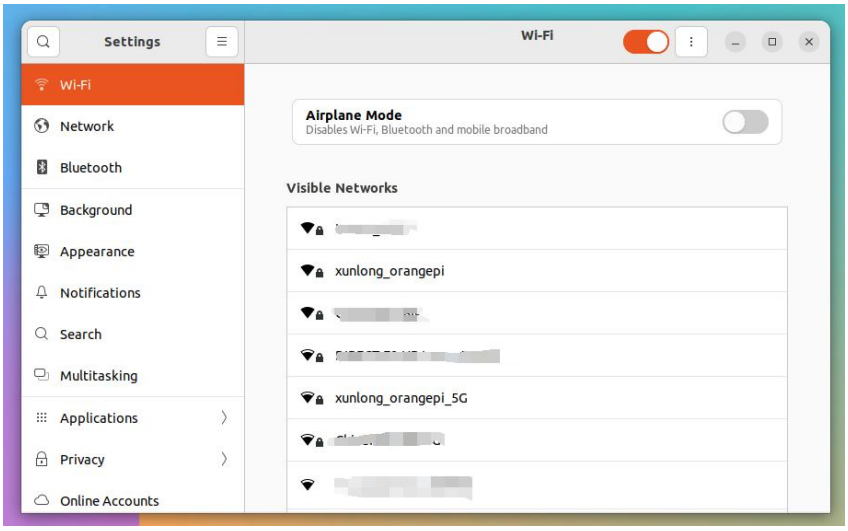
3.7.2.3. Testing methods for desktop images

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

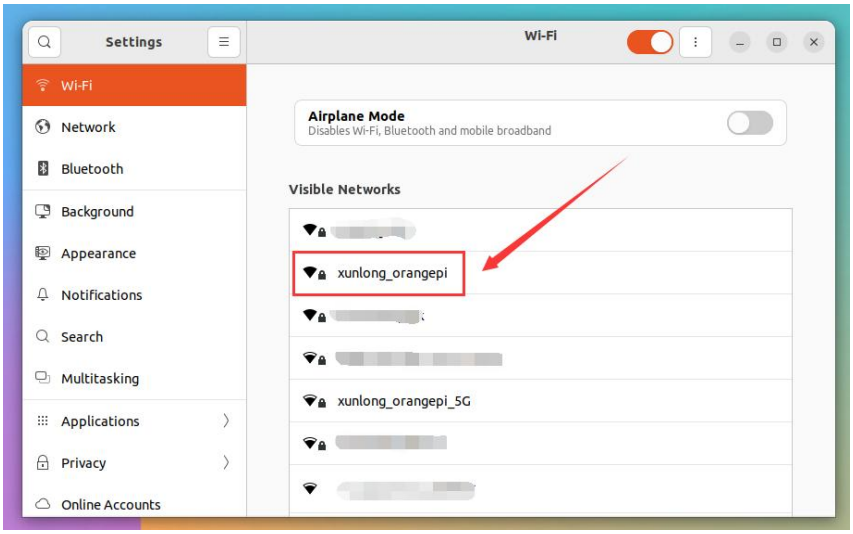
2) Select **Wi-Fi Settings** in the drop-down box that pops up.

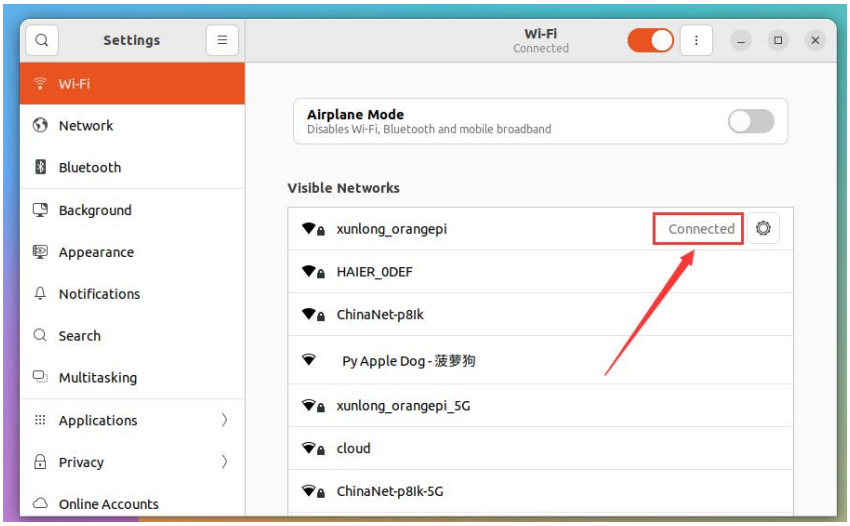


3) Then you can see the searched WIFI hotspots under **Visible Networks**



4) Then click on the WIFI hotspot you want to connect to, and then enter the password to start connecting to WIFI





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

https://github.com/oblique/create_ap

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

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

*** options:** This parameter can be used to specify encryption method, frequency band of WIFI hotspot, bandwidth mode, network sharing method, etc. You can get the specific 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, usually eth0
*** access-point-name:** hotspot name
*** passphrase:** hotspot password

3.7.2.4. create_ap method to create a WIFI hotspot in mode

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

```
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
```

3）Take out your phone and find the WIFI hotspot named **orangepi** created by the development board in the searched WIFI list. Then click **orangepi** to connect to the hotspot. The password is **orangepi** set above.

4）The display after successful connection is as shown below

5）In NAT mode, the IP addresses assigned to the network port and WIFI are in two different network segments. For example, the IP address of the development board's network port here 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 default IP address for the development board WIFI is 192.168.12.1.

```
orangepi@orangepi:~$ ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.12.1  netmask 255.255.255.0  broadcast 192.168.12.255
        inet6 fe80::be17:96ff:fe91:c53c  prefixlen 64  scopeid 0x20<link>
        ether bc:17:96:91:c5:3c  txqueuelen 1000  (Ethernet)
```

```
        RX packets 5973  bytes 1129156 (1.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7213  bytes 6453949 (6.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The DHCP service of the development board defaults to assigning IP addresses in the **192.168.12.0/24** network segment to devices accessing hotspots. Click on the connected **orangepi** hotspot on your phone and you will see that its IP address 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. For example, the command to specify the network segment of the access point AP as 192.168.2.1 through the -g parameter is as follows:

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

At this point, after connecting to the hotspot through your phone, click on the already connected WIFI hotspot **orangepi**, and you will see that your phone's IP address is **192.168.2.X**.

7）Without specifying the **--freq-band** parameter, the default hotspot created is in the 2.4G frequency band. If you want to create a hotspot in the 5G frequency band, you can specify it through the **--freq-band 5** parameter. The specific command is as follows

```
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, and the specific command is as follows

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

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

3.7.2.5. The method of creating a WIFI hotspot in bridge mode by creating a new app

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

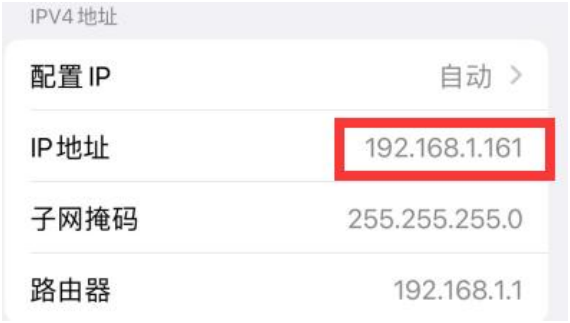
```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --no-virt
```

2）If the following information is output, it indicates that the WIFI hotspot has been successfully created

```
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
```

3）At this point, take out your phone and find the WIFI hotspot named **orangepi** created by the development board in the searched WIFI list. You can then click on **orangepi** to connect to the hotspot, and the password is the **orangepi** set above

4）The display after successful connection is shown in the following figure



5）Without specifying the **--freq-band** parameter, the default hotspot created is in the 2.4G frequency band. If you want to create a hotspot in the 5G frequency band, you can specify it through the **--freq-band 5** parameter. The specific command is as follows

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

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

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

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

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

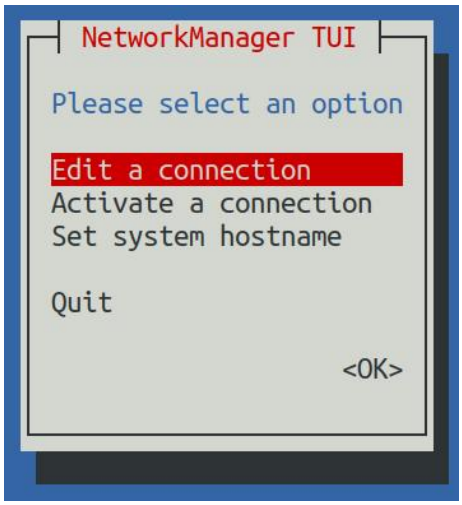
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file.

3. 7. 3. 1. Using nmtui command to set static IP address

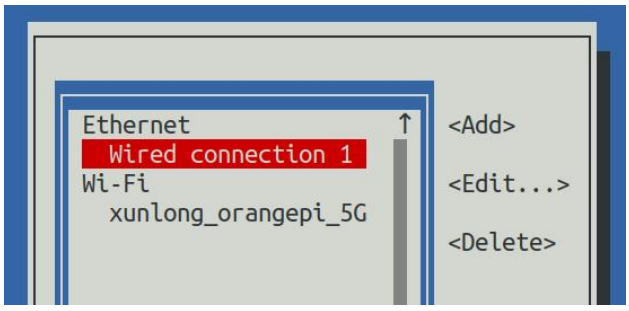
1) First run the **nmtui** command

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

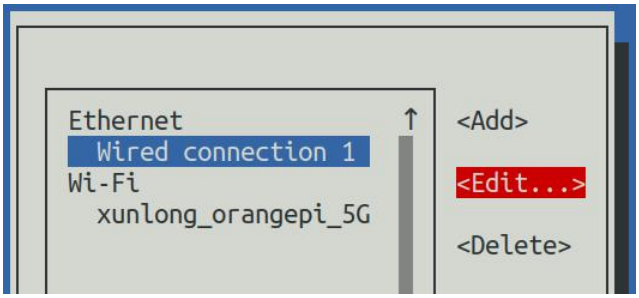
2) Then select **Edit a connection** and press Enter



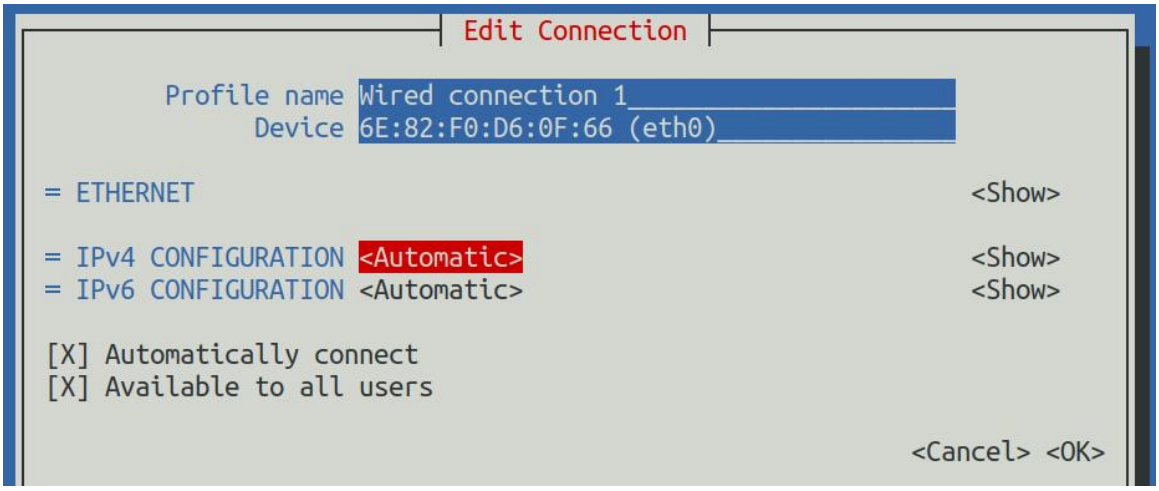
3) Then select the network interface for which you want to set a static IP address. For example, to set a static IP address for an **Ethernet**interface, select **Wired connection 1**.



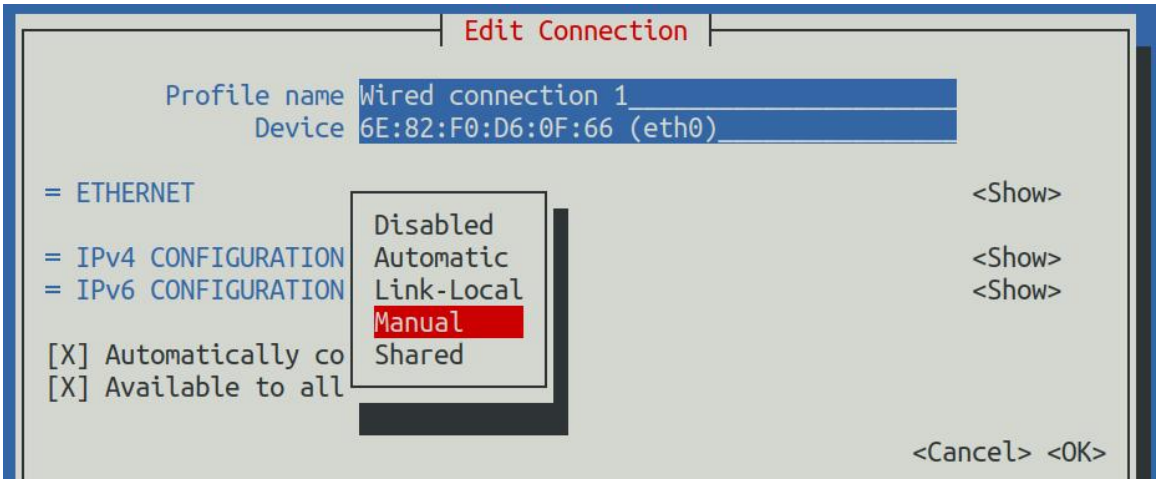
4) Then select **Edit** using the **Tab** key and press Enter



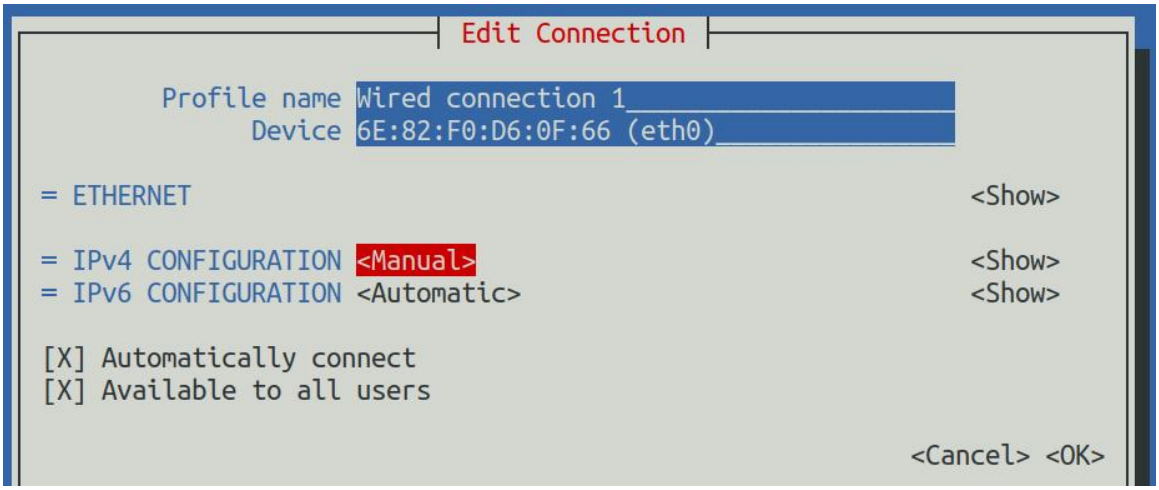
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4



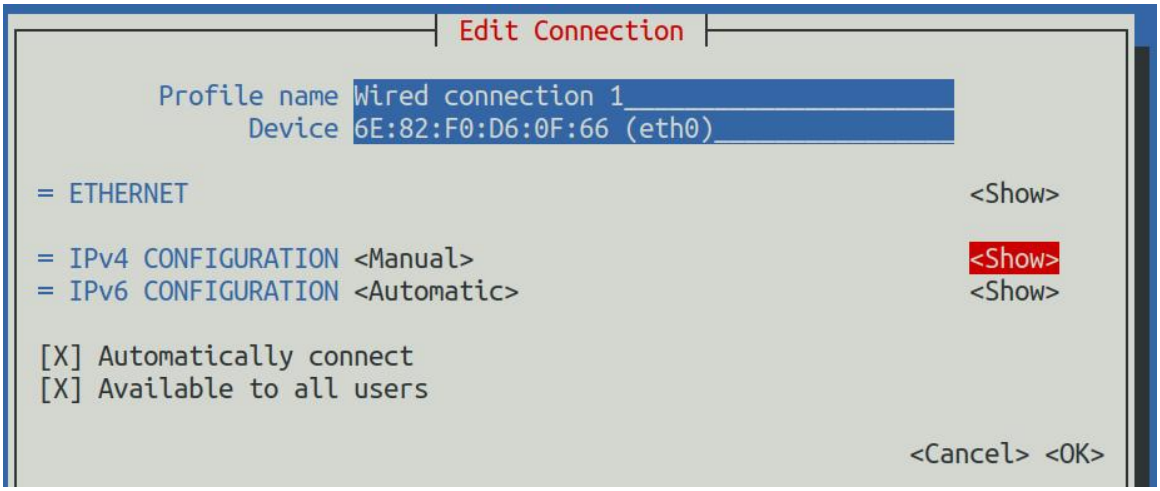
6) Then press Enter, use the up and down arrow keys to select **Manual**, and then press Enter to confirm.



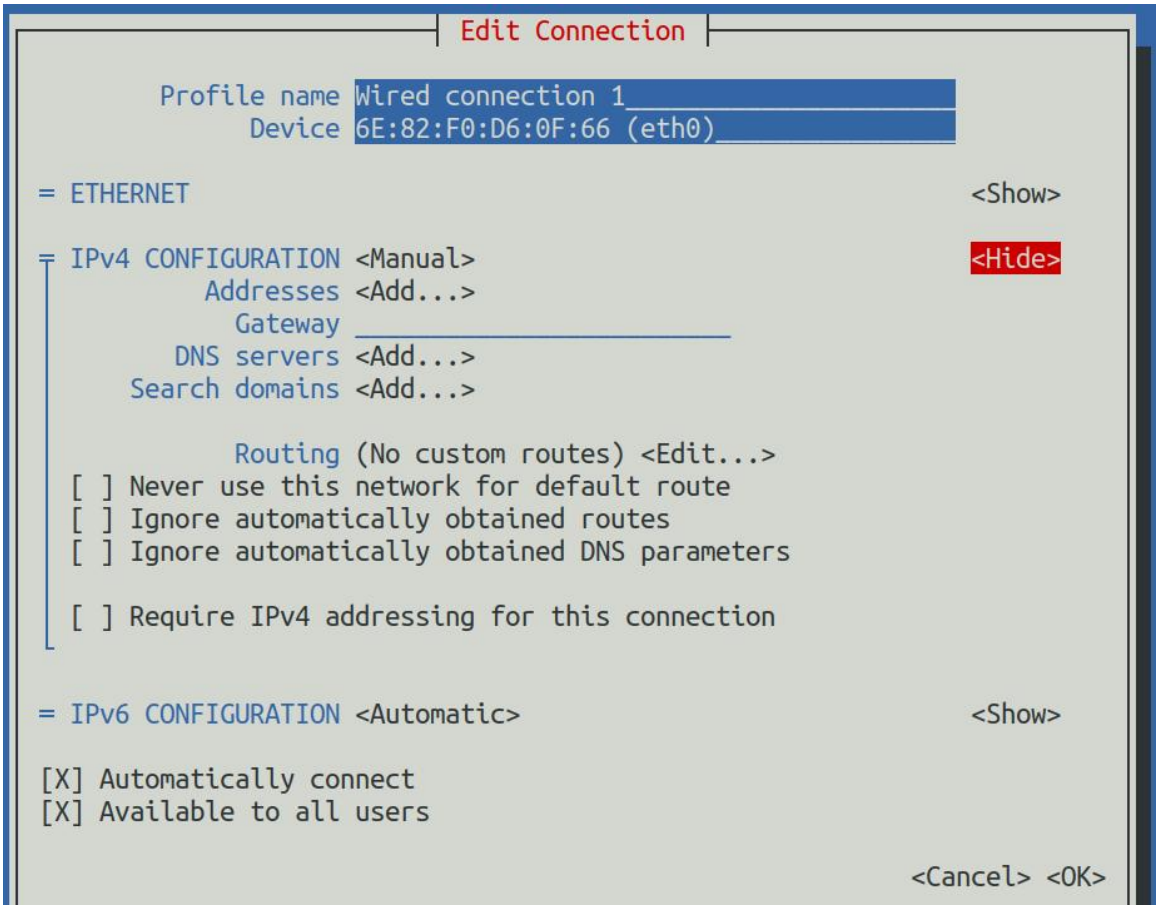
7) The display after selection is as shown below



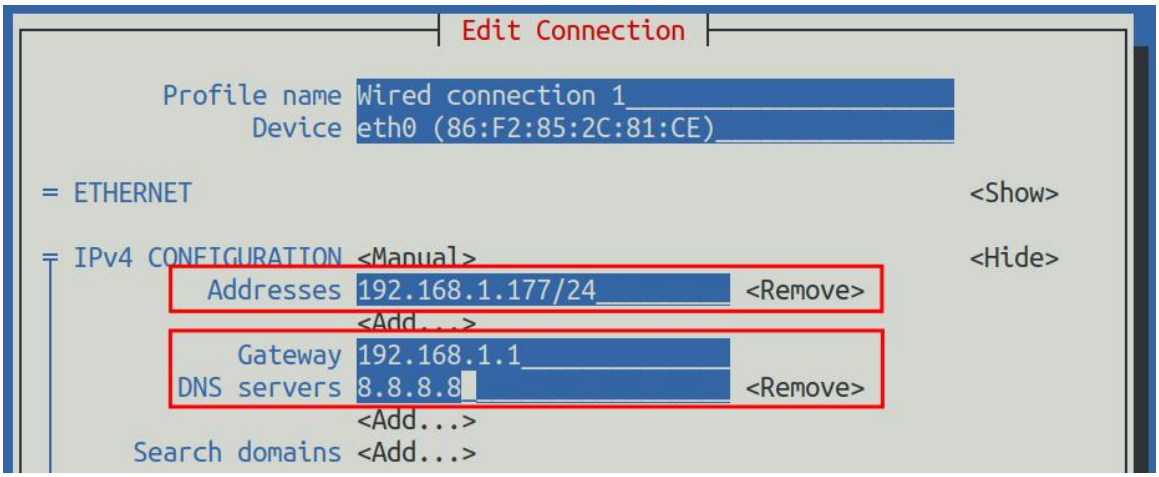
8) Then use the Tab key to move the cursor to **<Show>**



9) Then press Enter, and the following setting interface will pop up.

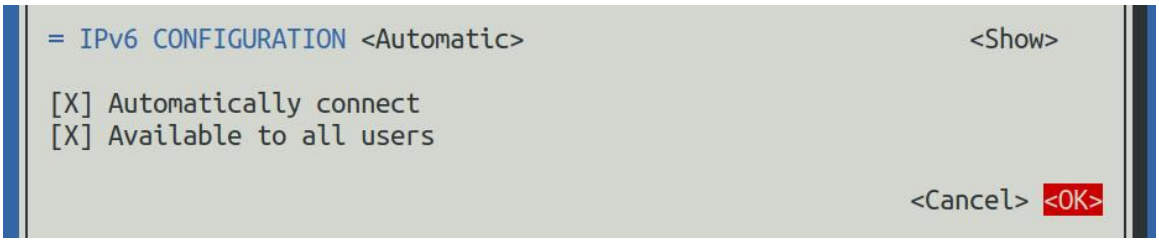


10) Then you can set the IP address, gateway and DNS server address as shown in the figure below (there are many other setting options, please explore them yourself). Please set them according to your specific needs. **The values set in the figure below are just an example.**

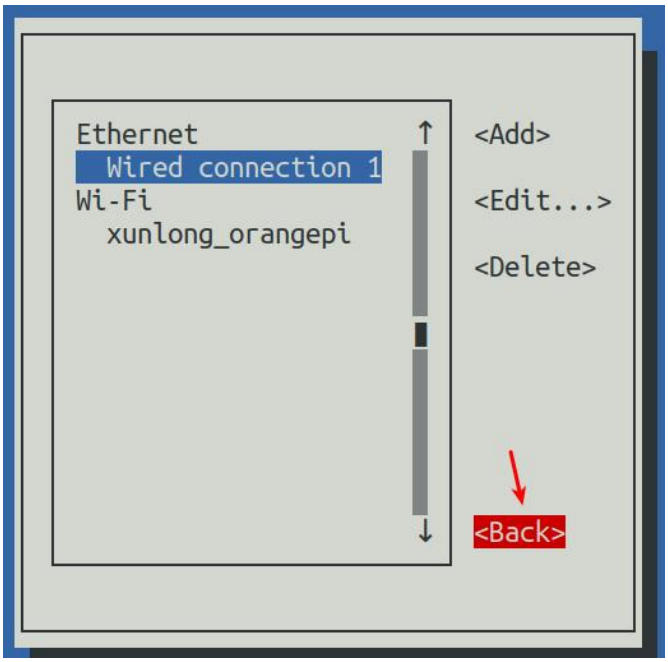


11) After setting, move the cursor to **<OK>** in the lower right corner and press Enter to

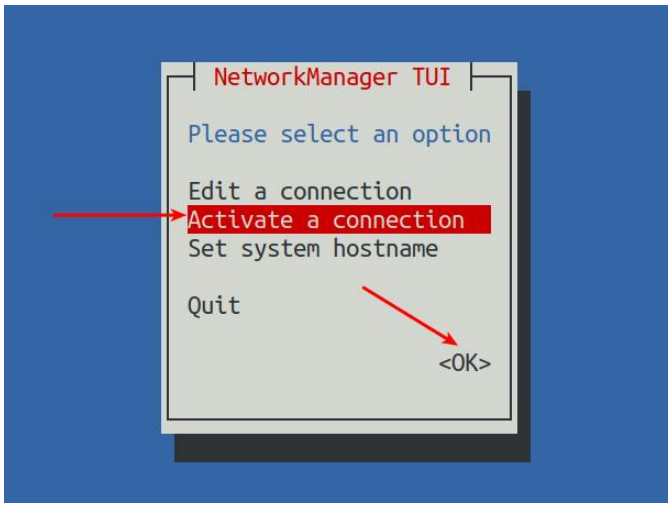
confirm.



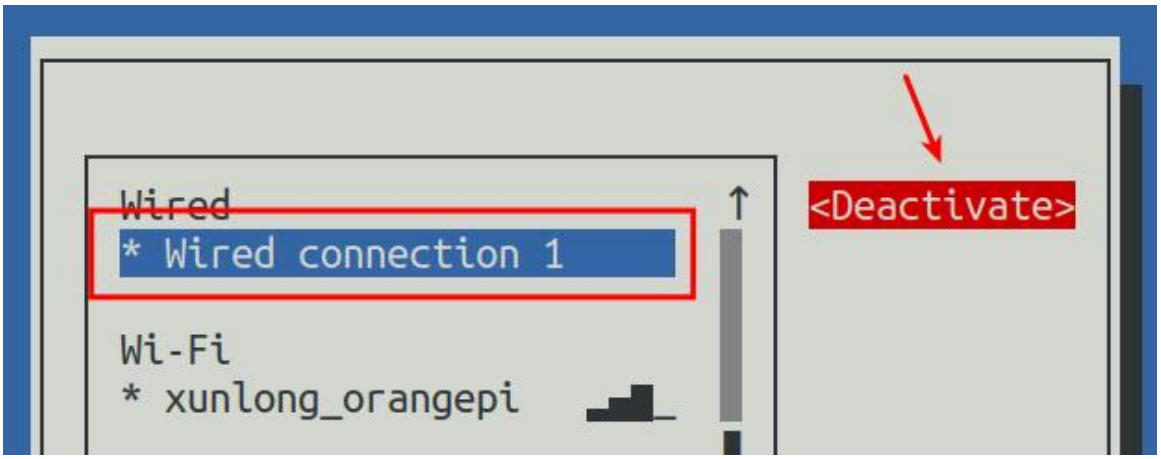
12) Then click **<Back>** to return to the previous selection interface



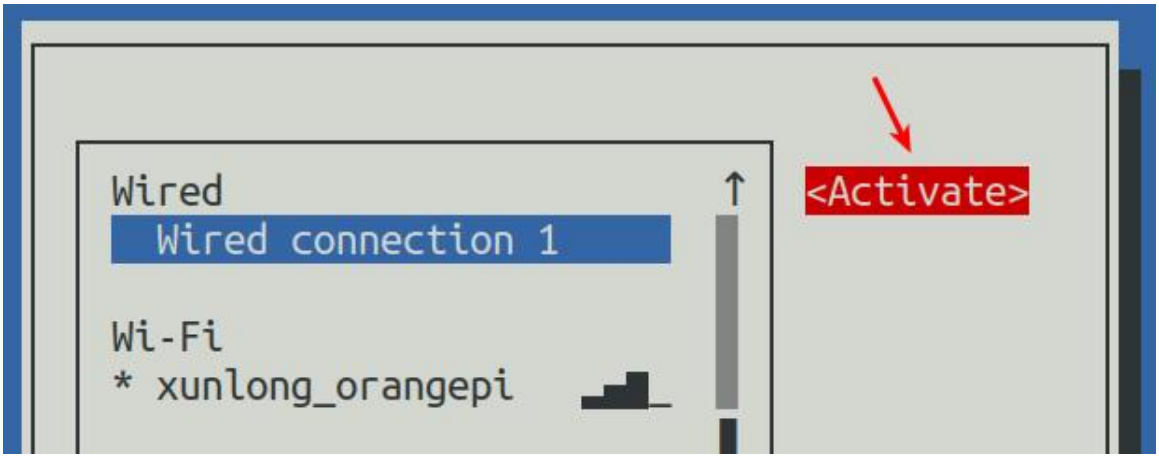
13) Then select **Activate a connection**, move the cursor to **<OK>**, and press Enter



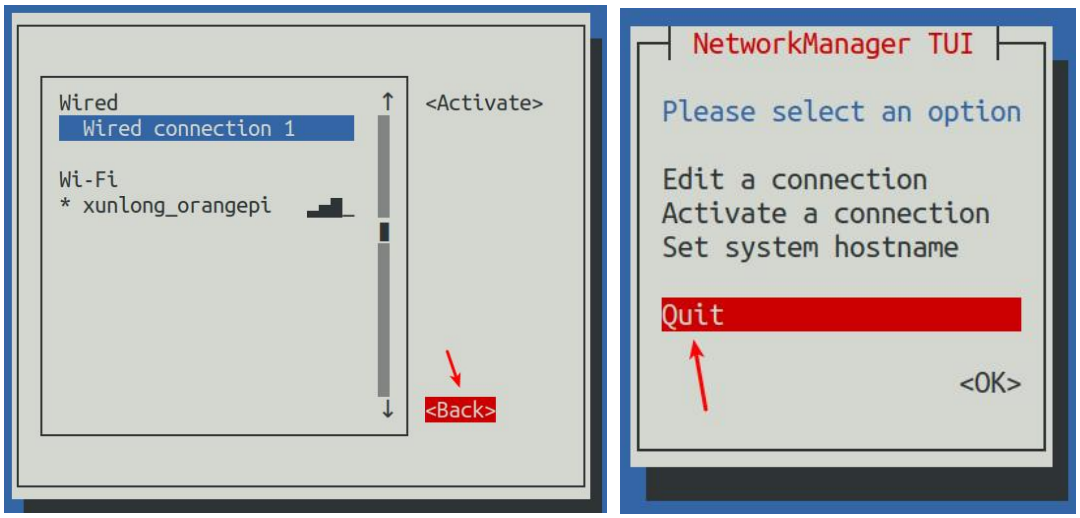
14) Then select the network interface you want to set, such as **Wired connection 1**, then move the cursor to **<Deactivate>**, and press Enter to disable **Wired connection 1**



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



16) Then you can exit nmtui by pressing the **<Back>** and **Quit** buttons.



17) Then use **ip a s eth0** to see that the IP address of the network port has become the static IP address set earlier.

```
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 network connectivity to check if the IP address is configured OK. The **ping** command can be interrupted by pressing **Ctrl+C**.

```
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
```

3.7.3.2. Use nmcli command to set static IP address

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

2) Then use the **nmcli con show** command to view the name of the network device, as shown below

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

b. 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 setting the static IP address of the Ethernet port. If you need to set the static IP address of WIFI, please change it to the name corresponding to the WIFI network interface (which can be obtained through the **nmcli con show** command)

b. The static IP address to be set after **ipv4.addresses** can be changed to the value you want to set

c.**ipv4.gateway** means 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
```

3.7.4. How to set up the Linux system to automatically connect to the network when it starts 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 a network cable that can access the Internet normally into the Ethernet port. After starting the Linux system, an IP address will be automatically assigned to the Ethernet port through DHCP. Then we can obtain the IP address of the Ethernet port through the HDMI screen, serial port or by checking the router background, and then we can remotely log in to the Linux system.

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 through ssh and connect to WIFI through commands, or connect to WIFI through commands in the HDMI screen or serial port.

However, 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 viewed through the router background. Or if there is no HDMI screen, serial port module and network cable, and only WIFI can be connected, you can use the method described in this section to automatically connect to WIFI and 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 virtual machine with Ubuntu system installed.

Why do you need a Linux system machine? Because the root file system of the Linux system of the development board burned in the TF card is in ext4 format. The Linux system machine can mount it normally and then modify the configuration files in it.

If you want to modify it in Windows, you can use 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 using Paragon ExtFS for Windows, please solve them yourself. We will not answer your questions.

1) First, burn the Linux image of the development board you want to use to the TF card, and then use the card reader to insert the TF card with the burned development board Linux image into the machine with the Linux system installed (such as a computer with Ubuntu system installed, the following demonstration will take the Ubuntu computer as an example)

2) When the TF card is inserted into the Ubuntu computer, the Ubuntu computer will generally automatically mount the Linux root file system partition in the TF card. From the following command, we can know that **/media/test/opi_root** is the path where the Linux root file system in the TF card 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 TF 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 development board Linux

system to automatically connect to a WIFI hotspot when it starts for the first time, or you can set a static IP address for the WIFI or Ethernet port.

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

5) Use the following command to open the orangepi_first_run.txt file, 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. The **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 value is 1, which means deletion. If it is set to 0, orangepi_first_run.txt will be renamed to orangepi_first_run.txt.old after the first startup. Generally, keep the default value

b. The **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 a static IP address for the Ethernet port, please set it to 1

d. The **FR_net_wifi_enabled** variable is used to control whether to enable the WIFI configuration. If you need to set the development board to automatically connect to the WIFI hotspot, you must set it to 1. Also, please note that if this variable is set to 1, the Ethernet port setting will be invalid. In other words, the WIFI and Ethernet ports cannot be set at the same time (why, because there is no need...)

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 a static IP address for the WIFI or Ethernet port

h. **FR_net_static_ip** variable is used to set the static IP address. Please set it 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) Here are some 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 WIFI IP address to a specific static IP address (so that when the Linux system starts, you can directly use the set static IP address to remotely log in to the development board through SSH, without having to check the IP address of the development board through the router background), 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 the Linux system of the development board to automatically set the IP address of the Ethernet port to the desired static IP address after the first startup, 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

Linux system of the development board in the TF card, then uninstall the TF card, and then you can insert the TF card into the development board to start it.

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

After the development board's Linux system is started for the first time, orangepi_first_run.txt will be deleted or renamed to orangepi_first_run.txt.old. At this time, even if you reset the orangepi_first_run.txt configuration file and restart the development board's Linux system, the configuration in orangepi_first_run.txt will not take effect again, because this configuration will only take effect at the first startup after burning the Linux system. Please pay special attention to this.

3.8. SSH remote login development board

By default, Linux systems enable SSH remote login and allow the root user to log in. Before logging in through SSH, you must first 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 to the development board under Ubuntu

1) Get the IP address of the development board

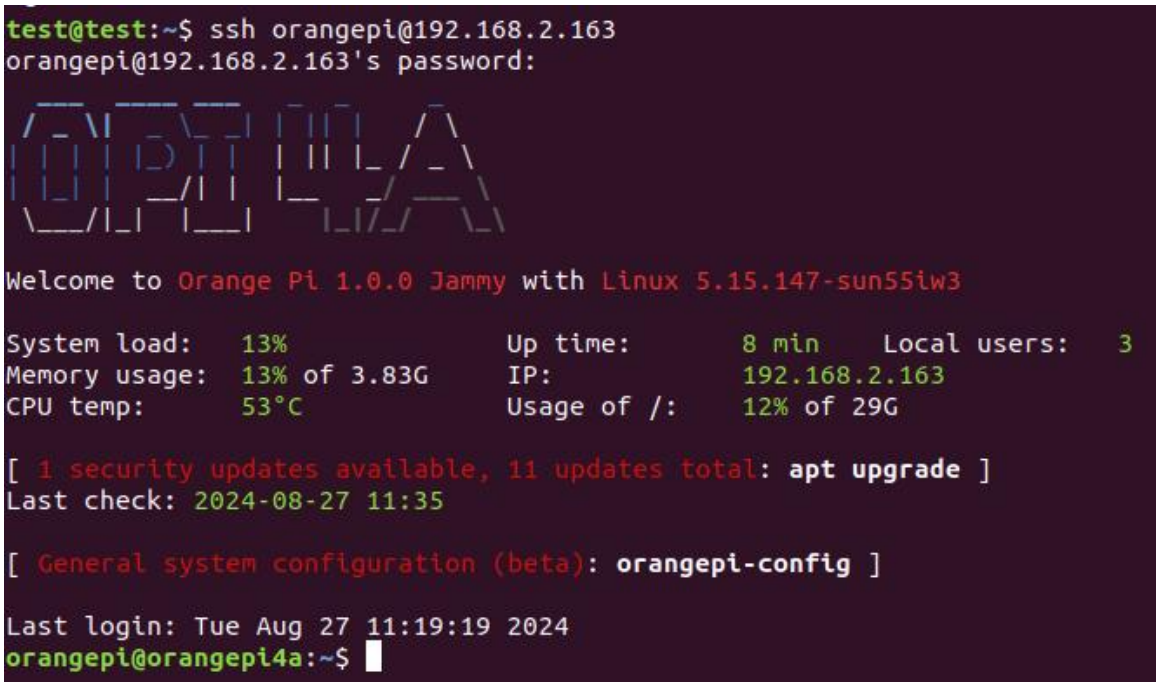
2) Then you can remotely log in to the Linux system through the ssh command

```
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 you enter the password, the screen will not display the specific content of the password you entered. Please do not think that there is any malfunction. Just press Enter after entering it.

If the prompt refuses to connect, as long as you are using the image provided by Orange Pi, please do not doubt whether the password orangepi is wrong, but look for other reasons.

3) After successfully logging into the system, the display is as shown below



If ssh cannot log in to the Linux system normally, first check whether the IP address of the development board can be pinged. If the ping is successful, 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 to try to connect:

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

If it still doesn't work, please re-burn the system and try again.

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

1) First obtain the IP address of the development board

2) You can use MobaXterm to remotely log in to the development board under Windows. First, create a new ssh session

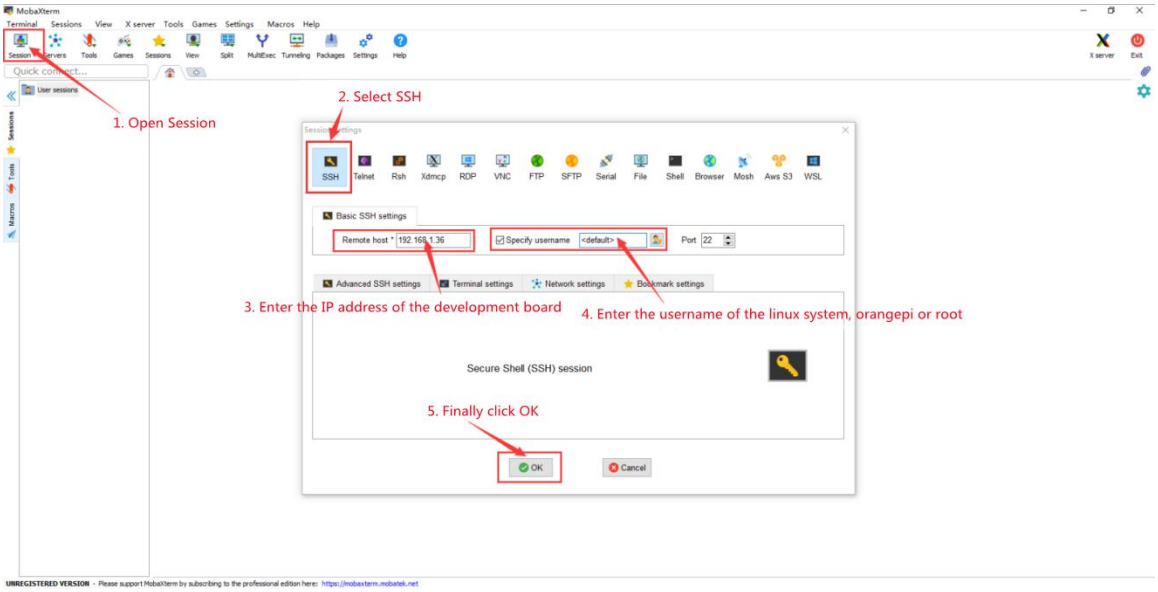
a. Open **Session**

b. Then select **SSH** in **Session Setting**

c. Then enter the IP address of the development board in **Remote host**

d. Then enter the Linux **system username** root or **orangepi** in Specify username

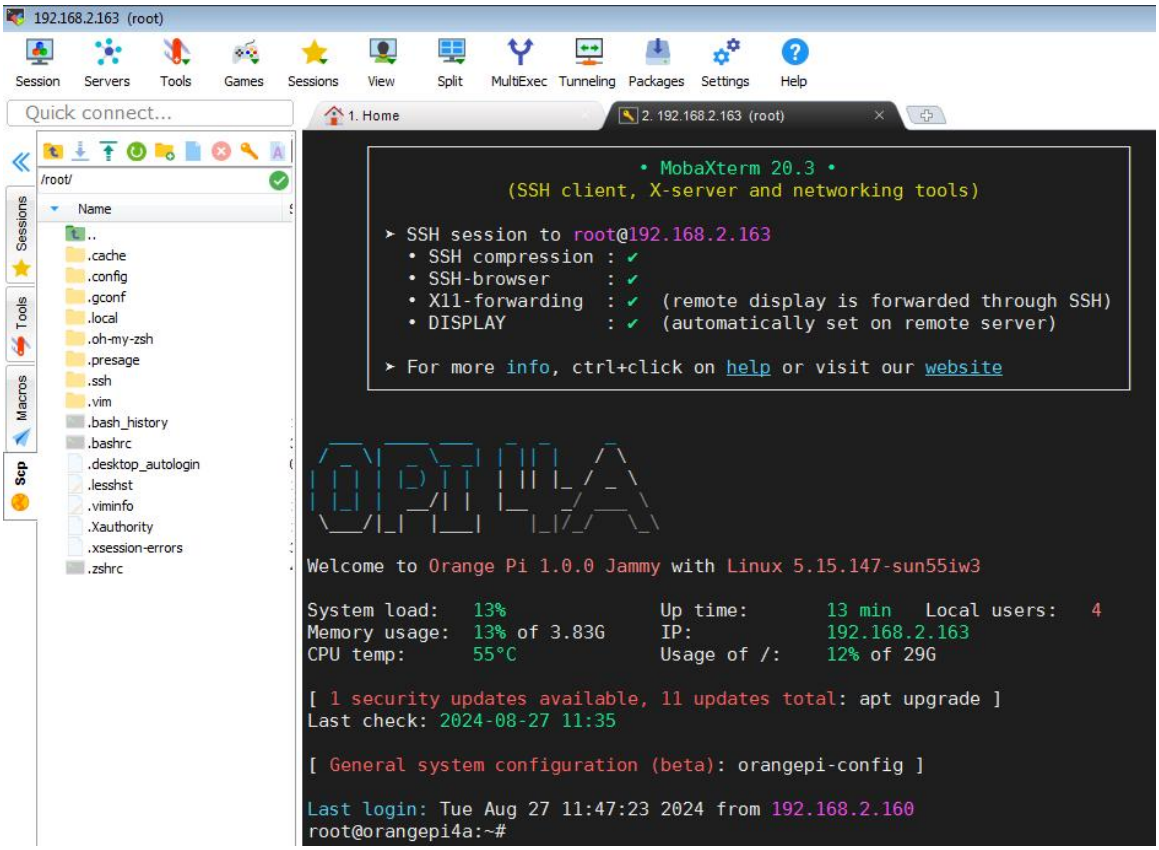
e. Finally, click **OK**



3) You will then be prompted to enter a password. The default password for both root and orangepi users is orangepi

Please note that when you enter the password, the specific content of the password will not be displayed on the screen. Please do not think that there is any malfunction. Just press Enter after entering it.

4) After successfully logging into the system, the display is as shown below



3. 9. HDMI test

3. 9. 1. HDMI display test

1) HDMI to HDMI cable to connect Orange Pi development board and HDMI display

2) After starting the Linux system, if the HDMI display has image output, it means that the HDMI interface is working properly

Note that although many laptops have HDMI interfaces, the HDMI interfaces of laptops generally only have output functions and do not have HDMI in functions,

which means that the HDMI output of other devices cannot be displayed on the laptop screen.

When you want to connect the HDMI of the development board to the HDMI interface of the laptop, please first confirm that your laptop supports the HDMI in function.

When there is no display on HDMI, please first check whether the HDMI cable is plugged in tightly. After confirming that the connection is OK, you can try a different screen to see if there is any display.

3.9.2. HDMI to VGA display test

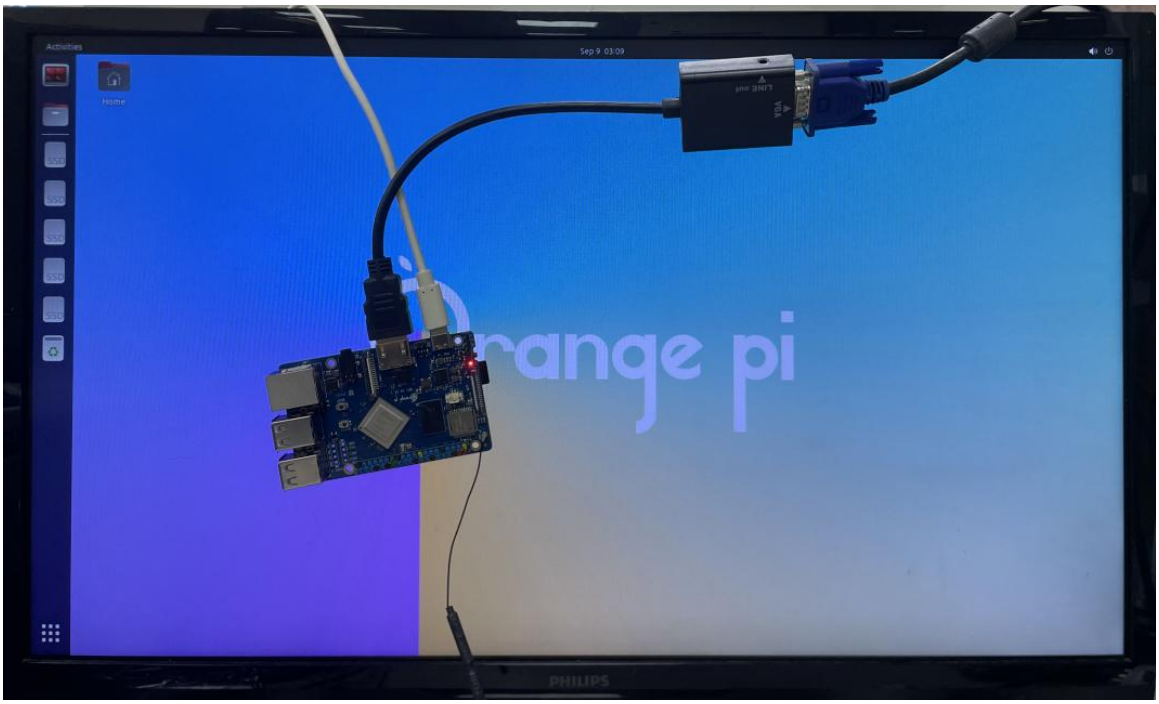
1) First you need to prepare the following accessories

a. HDMI to VGA converter

b. A VGA cable

c. A monitor or TV that supports VGA interface

2) HDMI to VGA display test is as follows



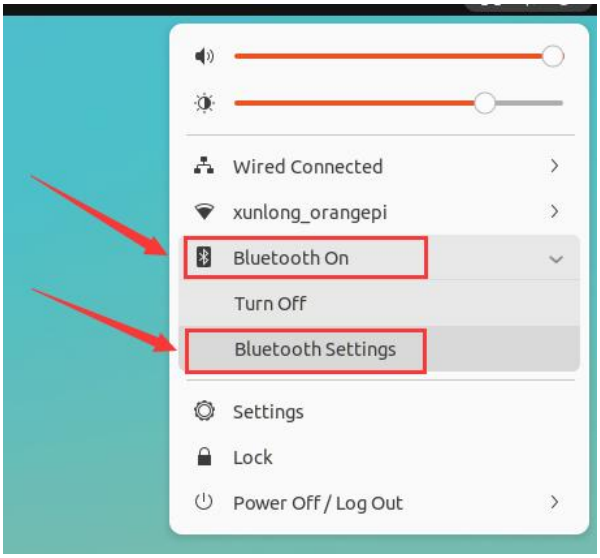
When using HDMI to VGA display, the development board and the Linux system of the development board do not need to do any settings, as long as the HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI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

3. 10. How to use Bluetooth

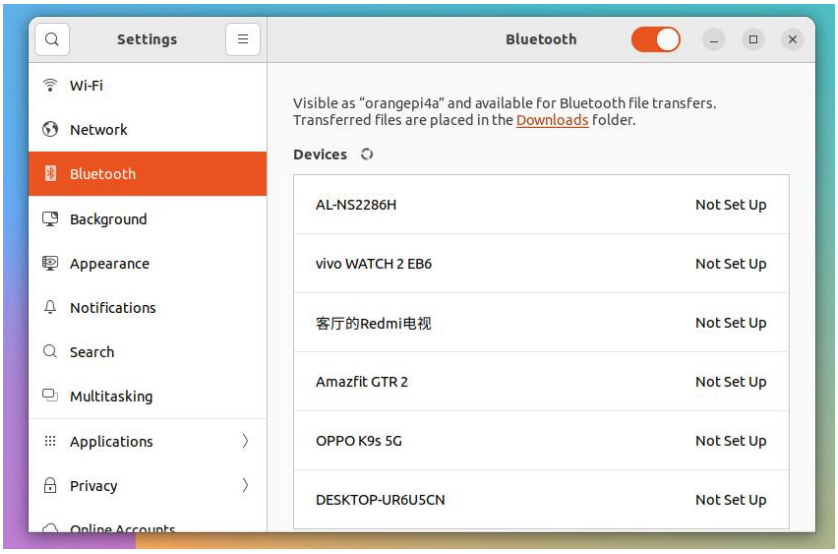
3. 10. 1. Testing methods for desktop images

1) First click on the area in the upper right corner of the desktop

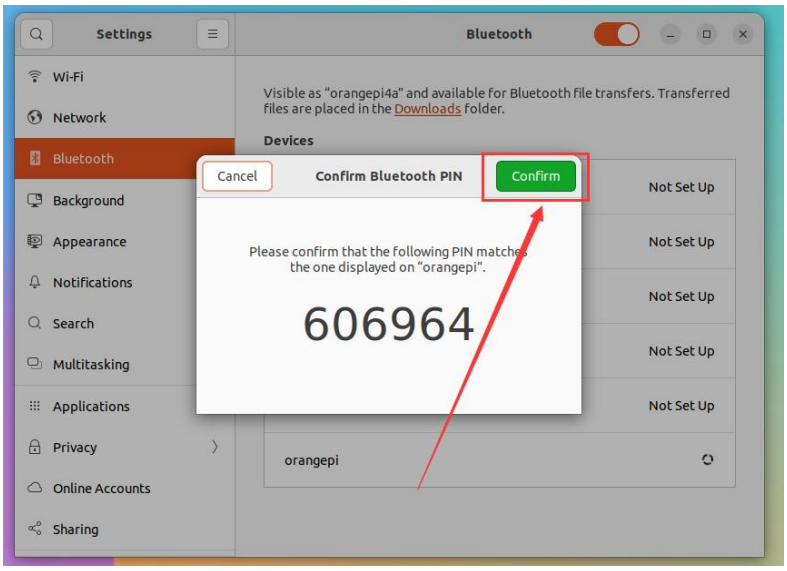
2) Then open **Bluetooth Settings**



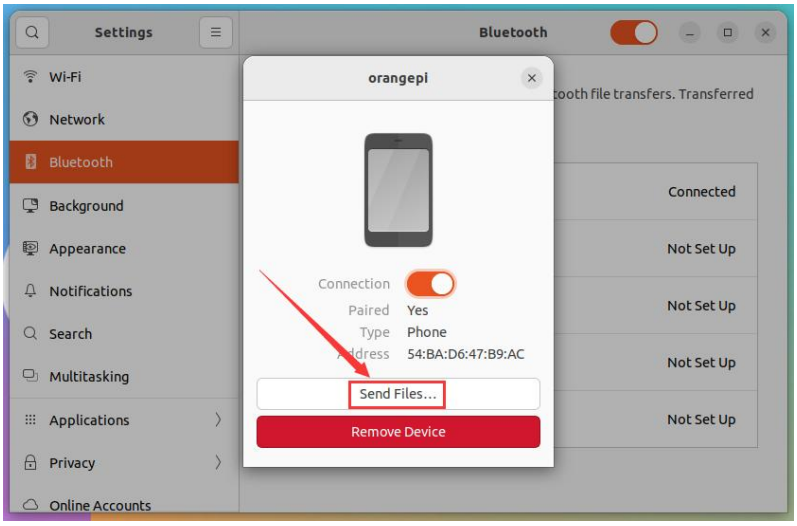
3) Bluetooth is turned on by default, and the Bluetooth devices scanned nearby will be displayed under **Devices**



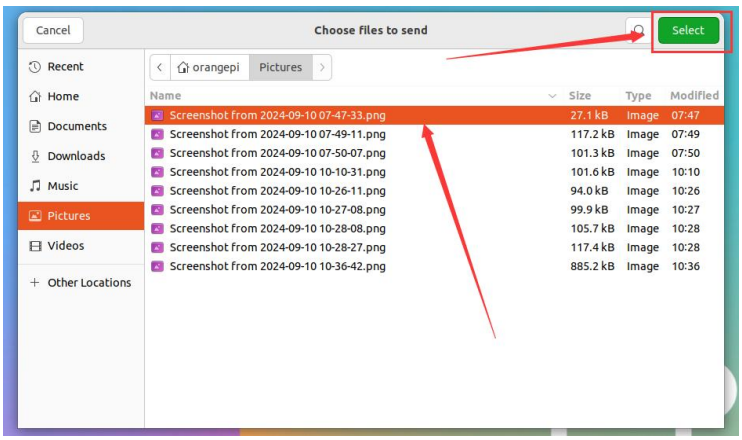
4) Then click on the device you want to connect to start pairing. After pairing starts, a pairing confirmation box will pop up. Select **Confirm**to confirm. At this time, you also need to confirm on the phone.



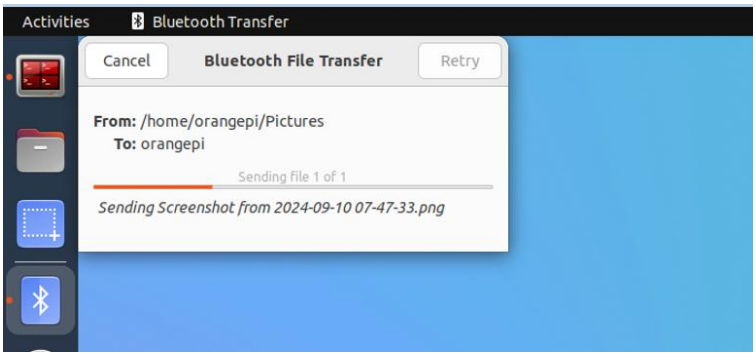
5) After pairing with the phone, you can select the paired Bluetooth device and then select **Send Files** to start sending a file to the phone.



6) Then select the file path to be sent, and click **Select** to start sending.



7) The interface for sending files is shown below



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

1) After entering the system, you can first use the **hciconfig** command to check whether there is a Bluetooth device node. If it exists, it means that 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 for Bluetooth devices

```
orangepi@orangepi:~$ sudo bluetoothctl
[NEW] Controller 10:11:12:13:14:15 orangepi4a [default]
Agent registered
[bluetooth]# power on          # Enable controller
Changing power on succeeded
[bluetooth]# discoverable on      #Set the controller to be discoverable
Changing discoverable on succeeded
[CHG] Controller 10:11:12:13:14:15 Discoverable: yes
[bluetooth]# pairable on     #Set the controller to be pairable
Changing pairable on succeeded
[bluetooth]# scan on          #Start scanning the surrounding 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 Mi phones
[NEW] Device DC:72:9B:4C:F4:CF orangepi
[bluetooth]# scan off          #After scanning the Bluetooth device you want to connect to, you can close the scan and write down the MAC address of the Bluetooth device. The Bluetooth device tested here is an Android phone, the Bluetooth name 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 the device you want to pair, you can pair it. Pairing requires the MAC address of the device

```
[bluetooth]# pair DC:72:9B:4C:F4:CF          #Pair using the MAC address of the scanned Bluetooth device
```

```
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 will also need to confirm on your 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 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 pairing is successful, the Bluetooth interface of the mobile phone is displayed as follows

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

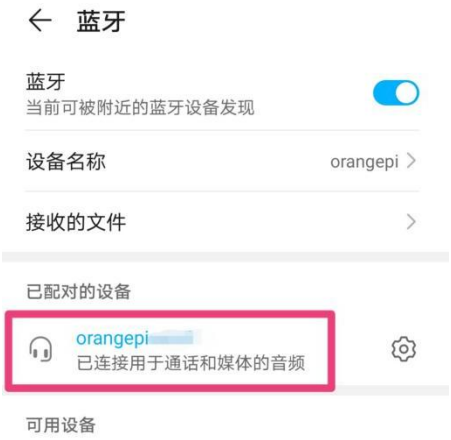
```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt -y install pulseaudio-module-bluetooth
orangepi@orangepi:~$ pulseaudio --start
```

6) How to connect to Bluetooth devices

```
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    #Use MAC address to connect to Bluetooth device
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 indicates that the connection is successful
```

7) After connecting to the Bluetooth device, you can see the prompt that the audio for calls and media has been connected on the Bluetooth configuration interface of the Android phone.



3. 11. USB interface test

The USB port can be connected to a USB hub to expand the number of USB ports.

3. 11. 1. Test by connecting USB mouse or keyboard

1) Insert the USB keyboard into the USB port 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 the system normally, it means that the USB interface is working properly (the mouse can only be used in the desktop version of the system)

3.11.2. Test by connecting USB storage device

1) First, insert the USB flash drive or USB mobile hard disk into the USB port of the Orange Pi development board.

2) Execute the following command. If you can see the output of sdX, it means that the USB disk has been successfully recognized.

```
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 USB drive to **/mnt,** and then you can view the files in the USB drive.

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

4) After mounting, you can use the **df -h** command to view the capacity usage and mount point of the USB drive.

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

3.11.3. USB Ethernet Card Test

1) The USB Ethernet cards that **have been tested** and can be used are as follows. The RTL8153 USB Gigabit Ethernet card can be used normally when inserted into the USB 2.0 Host interface of the development board, but the speed cannot reach Gigabit. Please note this.

Serial number	model
1	RTL8152B USB 100M LAN
2	RTL8153 USB Gigabit LAN

2) First, insert the USB network card into the USB port 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 the following log information can be seen 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 use the ifconfig command to see the device node of the USB network card and the automatically assigned IP address

```
orangepi@orangepi:~$ sudo ifconfig
eth1: 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 eth1
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 you can see through the lsmod command that the kernel automatically loads the following modules

```
orangepi@orangepi:~$ lsmod
Module                  Size  Used by
uvcvideo              106496  0
```

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 video number is not always video0, please refer to the actual one you 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) The -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) The -r option is used to specify the resolution of the photo

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

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 Linux, after taking a photo, you can use the scp command to transfer the photo to the Ubuntu PC for mirror viewing.

```
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 Linux system, you can directly view the captured pictures through the HDMI display

5) Test USB camera using mjpg-streamer

a. download mjpg-streamer

a) Github download address：

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

b) The mirror 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) DebianSystem

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

c. Compile and install mjpg-streamer

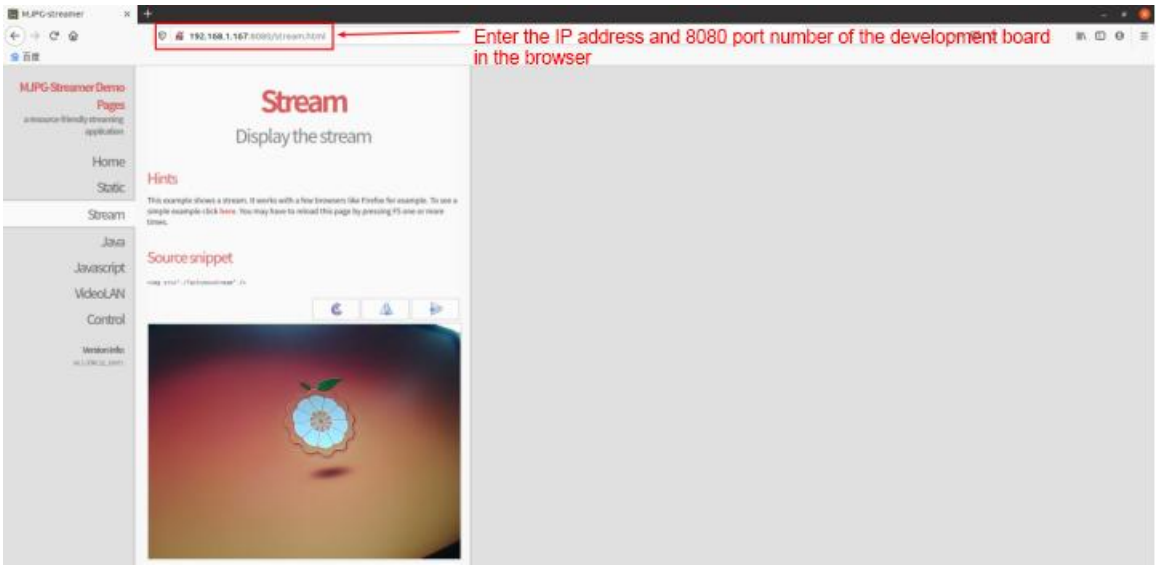
```
orangepi@orangepi:~$ cd mjpg-streamer/mjpg-streamer-experimental
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ make -j4
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ sudo make install
```

d. Then enter the following command to start mjpg_streamer

Note that the video number is not always video0. Please refer to the actual video number.

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

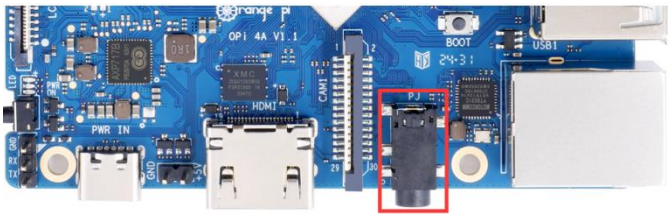


3. 12. Audio Test

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

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

1) First, plug the earphone into the earphone jack of the development board.



2) Use the **aplay -l** command to view the sound card devices supported by the Linux system, where **audiocodec** is the sound card device required for headphone playback.

```
root@orangepi:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: audiocodec [audiocodec], device 0: sunxi-snd-plat-aaudio-sunxi-snd-codec 7110000.codec-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: sndi2s1 [sndi2s1], device 0: sunxi-snd-plat-i2s-snd-soc-dummy-dai snd-soc-dummy-dai-0 []
```

```
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: sndhdmi [sndhdmi], device 0: sunxi-snd-plat-i2s-soc@3000000:hdmi_codec soc@3000000:hdmi_code []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio, and the headphones will be able to 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 during the headphone test, please pull the headphones out a little instead of plugging them all the way in.

3.12.1.2. HDMI audio playback test

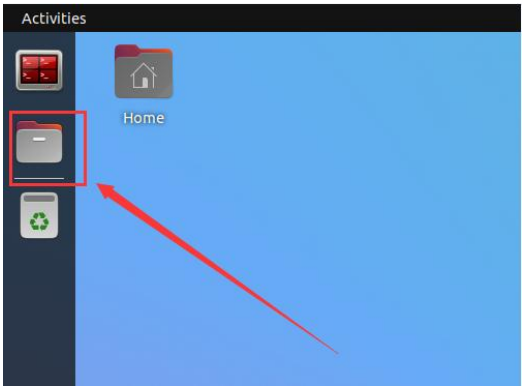
1) First, use an HDMI to HDMI cable to connect the Orange Pi development board to the TV (other HDMI displays need to ensure that they can play audio)

2) No additional settings are required for HDMI audio playback. Just use the **aplay** command to play it.

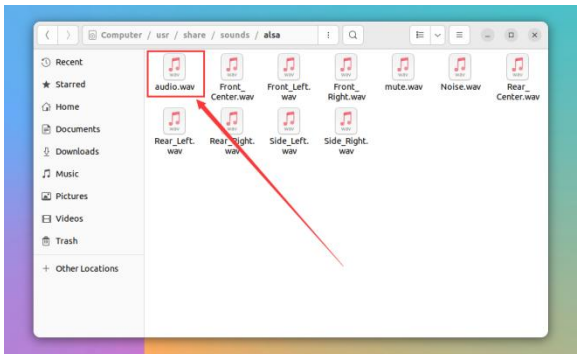
```
root@orangepi:~# aplay -D hw:2,0 /usr/share/sounds/alsa/audio.wav
```

3.12.2. Testing Audio Methods on Desktop Systems

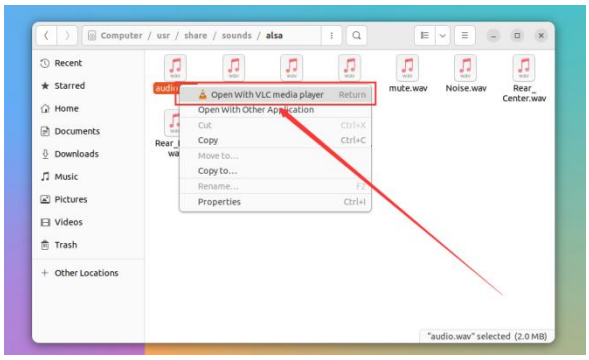
1) First open the file manager



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



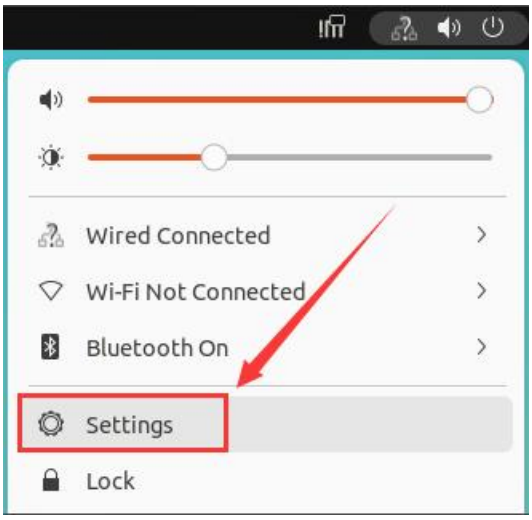
3) Then select the audio.wav file, right-click and choose to open it with vlc to start playing



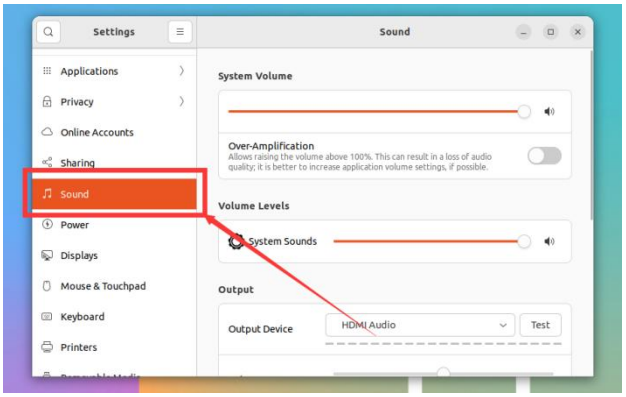
4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First click on the area in the upper right corner

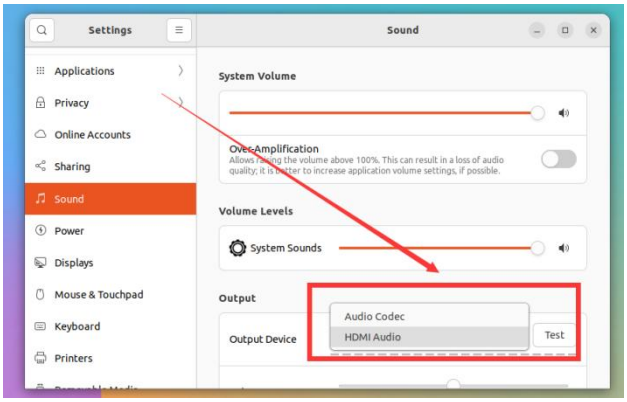
b. Then select **Settings**



c. Then find **Sound**



d. Then select the audio device you want to play in the drop-down selection box of Output Device (**select Audio Codec to output the sound from the headphones, select HDMI Audio to output the sound from HDMI**)



3. 12. 3. How to test recording using commands

1) The Orange Pi 4A development board does not have an onboard MIC, so you can only record audio through headphones with a MIC function. After plugging a headphone with

a MIC function into the development board, run the following command to record an audio clip through the headphone:

```
orangepi@orangepi:~$ arecord -D hw:0,0 -d 5 -f S16_LE -t wav /tmp/test.wav
```

3. 13. Temperature sensor

T527 has a total of 6 temperature sensors. The command to check the temperature is as follows:

The displayed temperature value needs to be divided by 1000 to get the unit in Celsius.

a. sensor0：CPUL 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_zone0/type
cpul_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone0/temp
54925
```

b. sensor1：CPUB 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_zone1/type
cpub_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone1/temp
54990
```

c. sensor2：The temperature sensor of the GPU. 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_zone2/type
gpu_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone2/temp
55056
```

d. sensor3：NPU 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_zone3/type
```

```
npu_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone3/temp
54686
```

e. sensor4：DDR 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

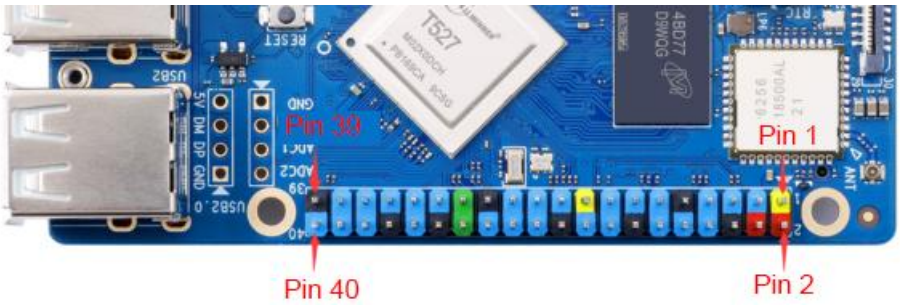
```
orangepi@orangepi:~# cat /sys/class/thermal/thermal_zone4/type
ddr_thermal_zone
orangepi@orangepi:~# cat /sys/class/thermal/thermal_zone4/temp
54925
```

f. sensor5：The temperature sensor of axp2202, 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_zone5/type
axp2202-usb
orangepi@orangepi:~# cat /sys/class/thermal/thermal_zone5/temp
45600
```

3. 14. 40 Pin Interface Pin Description

1) For the order of the 40-pin interface pins on the Orange Pi 4A development board, please refer to the silkscreen diagram on the development board.



2) The functions of the 40 pin interface pins of the development board are shown in the following table

a. Below is the complete pin diagram of 40pin

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

b. The table below is a picture of the left half of the complete table above, which can be seen more clearly

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
			3.3V		1
	UART4-TX	TWI4_SDA	PI1	257	3
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5
	TWI1-SCK	PWM0-8	PB4	36	7
			GND		9
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13
SPI2-MOSI			PB2	34	15
			3.3V		17
		SPI1_MOSI	PI4	261	19
		SPI1_MISO	PI5	260	21
UART5-RX		SPI1_CLK	PI3	259	23
			GND		25
		TWI5-SDA	PI9	265	27
SPI2-MISO			PB3	35	29
	PWM0-2		PB11	43	31
	UART3-RX	PWM0-13	PI12	268	33
		PWM0-10	PB6	38	35
			PB12	44	37
			GND		39

c. The table below is the right half of the complete table above, which can be seen more clearly

引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
2		5V			
4		5V			
6		GND			
8	45	PB13	UART7-TX	PWM0-4	
10	46	PB14	UART7-RX	PWM0-5	
12	37	PB5	PWM0-9	TWI1-SDA	
14		GND			
16	269	PI13			PWM0-14
18	270	PI14			PWM0-15
20		GND			
22	258	PI6		UART6-TX	
24	262	PI2	SPI1_CS0		UART5-TX
26	263	PI7		UART6-RX	
28	264	PI8	TWI5-SCK		
30		GND			
32	267	PI11	PWM0-12	UART3-TX	
34		GND			
36	266	PI10			
38	39	PB7	PWM0-11		
40	40	PB8			

3) There are **28** GPIO ports in the 40-pin interface, and the voltage of all GPIO ports is **3.3v**

3. 15. How to install wiringOP

Note that wiringOP is pre-installed in the Linux image released by Orange Pi. Unless the wiringOP code is updated, you do not need to download, compile and install it again. You can 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 wiringOP has been pre-installed and can be used normally.

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |   SDA.4  |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |   SCL.4  |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |    PWM8  |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.7    | 4   | 46   |
 |   32 |   5 |   TXD.2  |    OFF | 0 | 11 || 12 | 0 | OFF    | PB05     | 6   | 37   |
 |   33 |   7 |   RXD.2  |    OFF | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 |    PB02  |    OFF | 0 | 15 || 16 | 0 | OFF    | PI13     | 9   | 269  |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PI14     | 10  | 270  |
 |  260 |  11 |  MOSI.1  |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  261 |  12 |  MISO.1  |    OFF | 0 | 21 || 22 | 0 | OFF    | TXD.6    | 13  | 262  |
 |  259 |  14 |  SCLK.1  |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 258  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | RXD.6    | 16  | 263  |
 |  265 |  17 |   SDA.5  |    OFF | 0 | 27 || 28 | 0 | OFF    | SCL.5    | 18  | 264  |
 |   35 |  19 |    PB03  |    OFF | 0 | 29 || 30 |   |        | GND      |     |      |
 |   43 |  20 |    PB11  |    OFF | 0 | 31 || 32 | 0 | OFF    | PWM12    | 21  | 267  |
 |  268 |  22 |   PWM13  |    OFF | 0 | 33 || 34 |   |        | GND      |     |      |
 |   38 |  23 |    PB06  |    OFF | 0 | 35 || 36 | 0 | OFF    | PI10     | 24  | 266  |
 |   44 |  25 |    PB12  |    OFF | 0 | 37 || 38 | 0 | OFF    | PB07     | 26  | 39   |
 |      |     |      GND |        |   | 39 || 40 | 0 | OFF    | PB08     | 27  | 40   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
```

1) Download the wiringOP code

```
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 be downloaded from the wiringOP next branch, so 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, which is stored in /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

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |   SDA.4  |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |   SCL.4  |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |    PWM8  |    OFF | 0 |  7 || 8  | 0 |  OFF   | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 |  OFF   | RXD.7    | 4   | 46   |
 |   32 |   5 |   TXD.2  |    OFF | 0 | 11 || 12 | 0 |  OFF   | PB05     | 6   | 37   |
 |   33 |   7 |   RXD.2  |    OFF | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 |    PB02  |    OFF | 0 | 15 || 16 | 0 |  OFF   | PI13     | 9   | 269  |
 |      |     |     3.3V |        |   | 17 || 18 | 0 |  OFF   | PI14     | 10  | 270  |
 |  260 |  11 |  MOSI.1  |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  261 |  12 |  MISO.1  |    OFF | 0 | 21 || 22 | 0 |  OFF   | TXD.6    | 13  | 262  |
 |  259 |  14 |  SCLK.1  |    OFF | 0 | 23 || 24 | 0 |  OFF   | CE.1     | 15  | 258  |
 |      |     |      GND |        |   | 25 || 26 | 0 |  OFF   | RXD.6    | 16  | 263  |
 |  265 |  17 |   SDA.5  |    OFF | 0 | 27 || 28 | 0 |  OFF   | SCL.5    | 18  | 264  |
 |   35 |  19 |    PB03  |    OFF | 0 | 29 || 30 |   |        | GND      |     |      |
 |   43 |  20 |    PB11  |    OFF | 0 | 31 || 32 | 0 |  OFF   | PWM12    | 21  | 267  |
 |  268 |  22 |   PWM13  |    OFF | 0 | 33 || 34 |   |        | GND      |     |      |
 |   38 |  23 |    PB06  |    OFF | 0 | 35 || 36 | 0 |  OFF   | PI10     | 24  | 266  |
 |   44 |  25 |    PB12  |    OFF | 0 | 37 || 38 | 0 |  OFF   | PB07     | 26  | 39   |
 |      |     |      GND |        |   | 39 || 40 | 0 |  OFF   | PB08     | 27  | 40   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
```

3. 16. 40pin interface GPIO, I2C, UART, SPI and PWM test

3. 16. 1. 40pin GPIO port test

1) Below, we take pin 7, which corresponds to GPIO PB4, and wPi number 2 as an example to demonstrate how to set the high and low levels of the GPIO port.

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |   SDA.4  |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |   SCL.4  |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |    PWM8  |    OFF | 0 |  7 || 8  | 0 |  OFF   | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 |  OFF   | RXD.7    | 4   | 46   |
```

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

```
orangepi@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

```
orangepi@orangepi:~/wiringOP$ gpio write 2 0
```

Using gpio readall, you can see that the value of pin 7 (V) has changed to 0

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |   SDA.4  |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |   SCL.4  |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |     PWM8 |    OUT | 0 |  7 || 8  | 0 | OFF    | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.7    | 4   | 46   |
```

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.

```
orangepi@orangepi:~/wiringOP$ gpio write 2 1
```

Using gpio readall, you can see that the value of pin 7 (V) has changed to 1

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |   SDA.4  |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |   SCL.4  |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |     PWM8 |    OUT | 1 |  7 || 8  | 0 | OFF    | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.7    | 4   | 46   |
```

5) The setting method of other pins is similar. Just change the serial number of wPi to the serial number corresponding to the pin.

3. 16. 2. How to set pull-up and pull-down resistors on pin GPIO

1) Below, we take pin 7, which corresponds to GPIO PB4 and wPi number 2, as an example to demonstrate how to set the pull-up and pull-down resistors of the GPIO port.

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |   SDA.4  |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |   SCL.4  |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |     PWM8 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.7    | 4   | 46   |
```

2) First, you need to set the GPIO port to input mode. The third parameter needs to enter the wPi number corresponding to the pin.

```
orangepi@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

```
orangepi@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.

```
orangepi@orangepi:~/wiringOP$ gpio read 2
1
```

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

```
orangepi@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, it means that the pull-down mode is set successfully.

```
orangepi@orangepi:~/wiringOP$ gpio read 2
0
```

3.16.3. 40 Pin SPI Test

1) As shown in the figure below, the available SPIs of Orange Pi 4A are SPI1 and SPI2.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

2) SPI1 和 SPI2 在 40pin 中对应的引脚如下表所示。

	SPI1 corresponds to 40 pins	SPI2 corresponds to 40 pins
MOSI	**Pin 19**	**Pin 15**
MISO	**Pin 21**	**Pin 29**
CLK	**Pin 23**	**Pin 13**
CS0	**Pin 24**	**Pin 11**

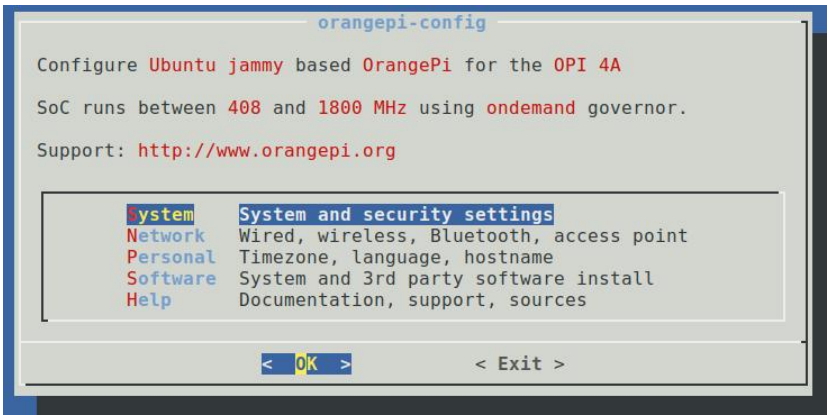
3) In Linux systems, SPI in 40 pins is turned off by default and needs to be manually

turned on to use. The detailed steps are as follows:

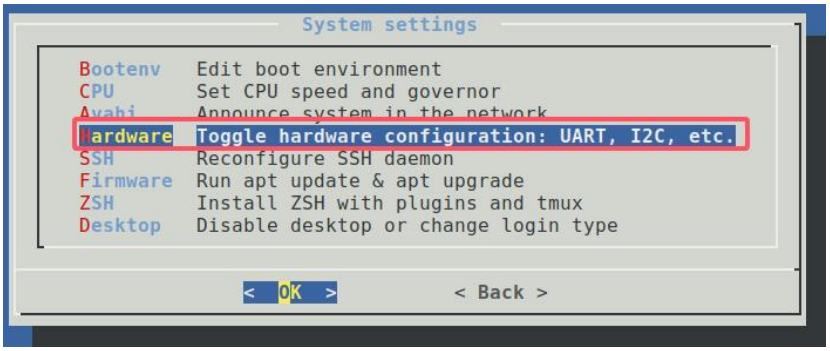
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

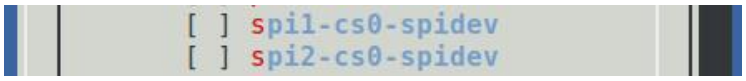
b. Then select **System**



c. Then select **Hardware**



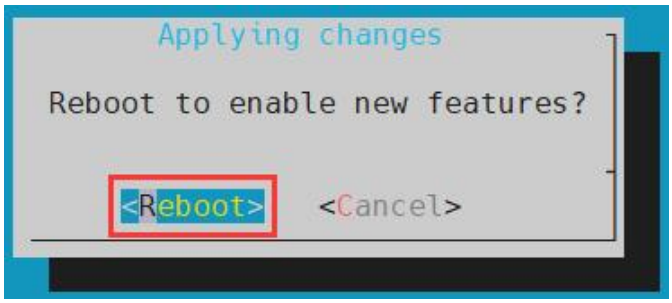
d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the SPI configuration that is open



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

f. Then select**<Back>**

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



4) Then check whether the device node of **spidevx.x** exists in the Linux system. If it exists, it means that the SPI configuration has taken effect.

```
orangepi@orangepi:~$ ls /dev/spidev*
/dev/spidev1.0 /dev/spidev2.0
```

5) Do not short the mosi and miso pins of SPI1 or SPI2. The output 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/spidev1.0
or
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev2.0
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    | ................................
```

6) Then short the mosi and miso pins of SPI1 or SPI2 and run spidev_test. The output is as follows. You can see that the data sent and received are the same.

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.0
or
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev2.0
```

```
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. 16. 4. 40 pin I2C test

1) As can be seen from the figure below, the available i2c buses for Orange Pi 4A are i2c1, i2c4 and i2c5, a total of three i2c buses.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

2) The corresponding pins of the three I2C buses in the 40 pin configuration are shown in the table below.

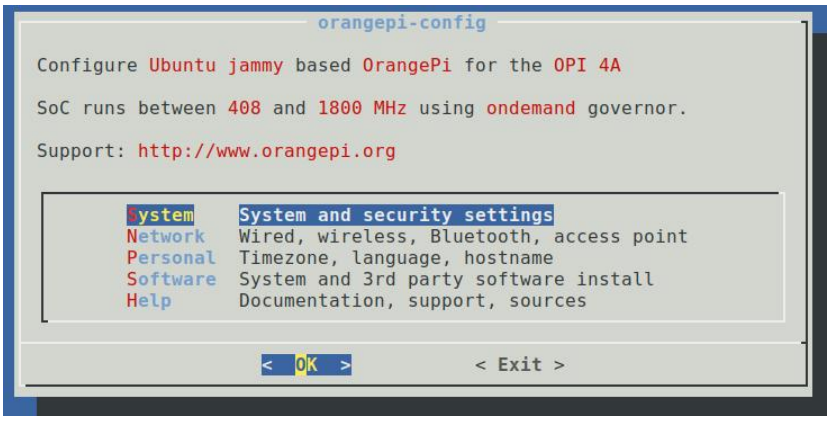
I2C bus	SDA corresponds to 40 pins	SCL corresponds to 40 pins	Dtbo corresponding configuration
I2C1	**Pin 12**	**Pin 7**	**i2c1**
I2C4	**Pin 3**	**Pin 5**	**i2c4**
I2C5	**Pin 27**	**Pin 28**	**i2c5**

3) In linux systems, I2C in 40 pins is turned off by default and needs to be manually turned on to use. The detailed steps are as follows:

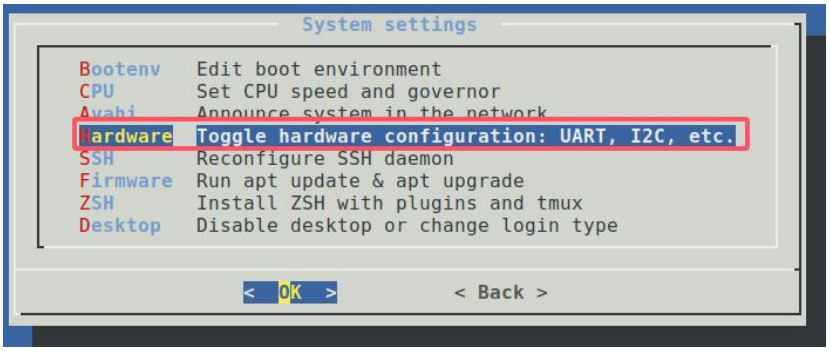
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

b. Then select **System**



c. Then select **Hardware**

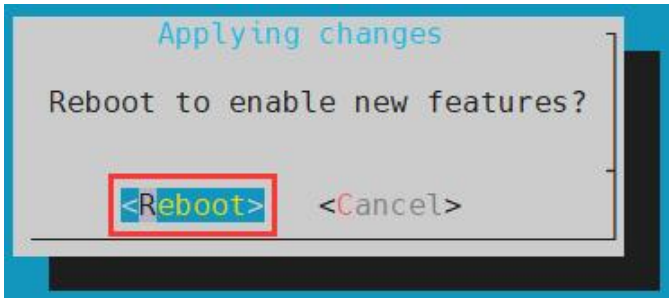


d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the I2C configuration that is open

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

f. Then select**<Back>**

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



4) After starting the Linux system, first confirm that the i2c device node exists under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-*
/dev/i2c-0  /dev/i2c-1  /dev/i2c-3  /dev/i2c-31  /dev/i2c-4  /dev/i2c-5  /dev/i2c-6
```

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

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

6) Then connect an i2c device to the i2c pin of the 40 pin connector

Please select the 5V and 3.3V pins according to the specific i2c device. Different i2c devices may require different voltage values.

7) Then use the **i2cdetect -y** command. If the address of the connected i2c device can be detected, it means that i2c can be used normally.

```
orangepi@orangepi:~$ sudo i2cdetect -y 1      #i2c1 command
orangepi@orangepi:~$ sudo i2cdetect -y 4      #i2c4 command
orangepi@orangepi:~$ sudo i2cdetect -y 5      #i2c5 command
```

```
root@orangepi4a:~# i2cdetect -y -r 1
     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: -- -- -- -- -- -- -- --
```

3. 16. 5. 40 pinUART test

1) As can be seen from the table below, Orange Pi 4A has six uart buses: uart2, uart3, uart4, uart5, uart6 and uart7.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

2) The corresponding pins of the six UART bus groups in the 40 pins are shown in the following table.

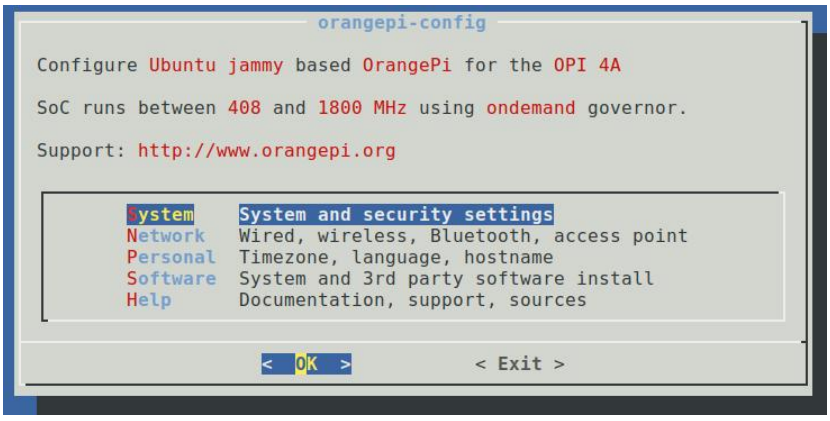
UART Bus	RX corresponds to 40pin	TX corresponds to 40pin	dtbo corresponding configuration
UART2	**Pin 13**	**Pin 11**	**uart2**
UART3	**Pin 33**	**Pin 32**	**uart3**
UART4	**Pin 5**	**Pin 3**	**uart4**
UART5	**Pin 23**	**Pin 24**	**uart5**
UART6	**Pin 26**	**Pin 22**	**uart6**
UART7	**Pin 10**	**Pin 8**	**uart7**

3) In linux systems, UART in 40 pins is turned off by default and needs to be manually turned on to use. The detailed steps are as follows:

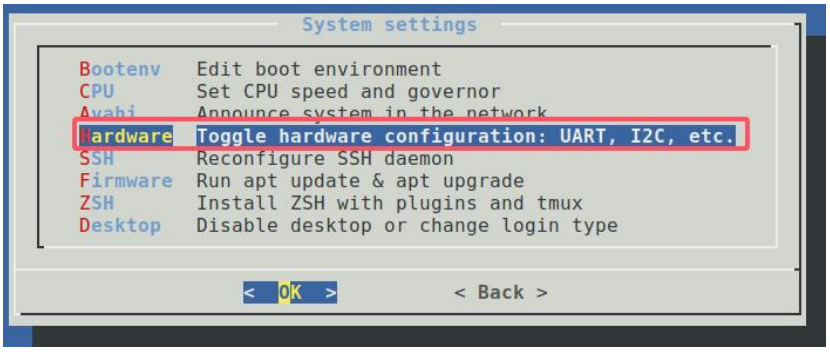
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

b. Then select **System**



c. Then select **Hardware**

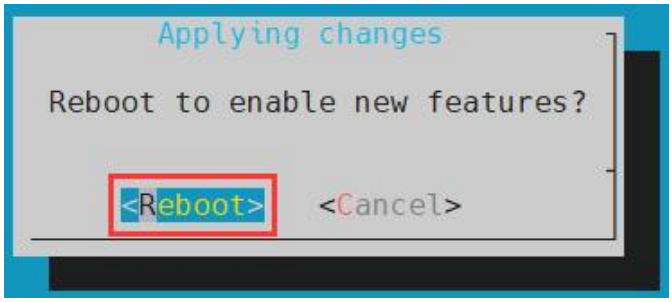


d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the UART configuration that is turned on

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

f. Then select**<Back>**

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



4) After entering the Linux system, first confirm whether there is a uart device node under **/dev**

```
orangepi@orangepi:~$ ls /dev/ttyAS*
```

```
/dev/ttyAS0  /dev/ttyAS1  /dev/ttyAS2  /dev/ttyAS3  /dev/ttyAS4  /dev/ttyAS5
/dev/ttyAS6  /dev/ttyAS7
```

5) Then start testing the UART interface. First use the Dupont line to short-circuit the rx and tx of the UART interface to be tested.

6) Use the **gpio serial** command 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 (ttyASX needs to be replaced with the corresponding uart node name, please do not copy it)

```
orangepi@orangepi:~$ gpio serial /dev/ttyASX

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

3. 16. 6. How to test PWM using /sys/class/pwm/

1) According to the table below, the Orange Pi 4A has 14 PWM channels available, including pwm1, pwm2, pwm4, pwm5, pwm6, pwm7, pwm8, pwm9, pwm10, pwm11, pwm12, pwm13, pwm14, and pwm15

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

2) The corresponding pins of PWM in 40pin are shown in the following table.

PWM Bus	Corresponding to 40pin	dtbo corresponding configuration
PWM0-1	**Pin 5**	**pwm1**
PWM0-2	**Pin 31**	**pwm2**
PWM0-4	**Pin 8**	**pwm4**

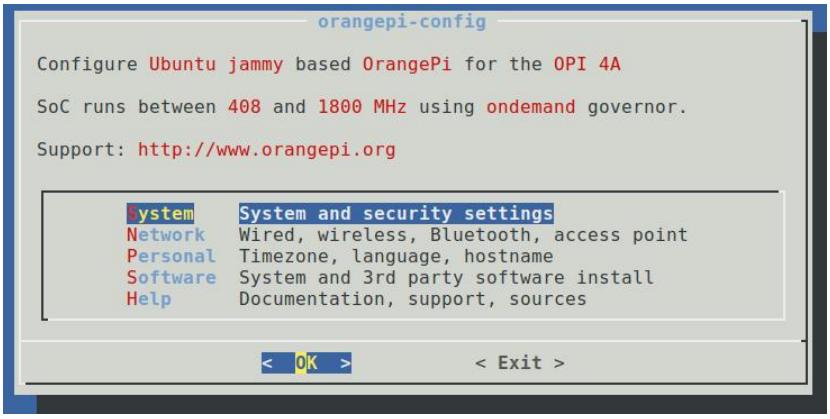
PWM0-5	**Pin 10**	**pwm5**
PWM0-6	**Pin 11**	**pwm6**
PWM0-7	**Pin 13**	**pwm7**
PWM0-8	**Pin 7**	**pwm8**
PWM0-9	**Pin 12**	**pwm9**
PWM0-10	**Pin 35**	**pwm10**
PWM0-11	**Pin 38**	**pwm11**
PWM0-12	**Pin 32**	**pwm12**
PWM0-13	**Pin 33**	**pwm13**
PWM0-14	**Pin 16**	**pwm14**
PWM0-15	**Pin 18**	**pwm15**

3) In linux systems, UART in 40 pins is turned off by default and needs to be manually turned on to use. The detailed steps are as follows:

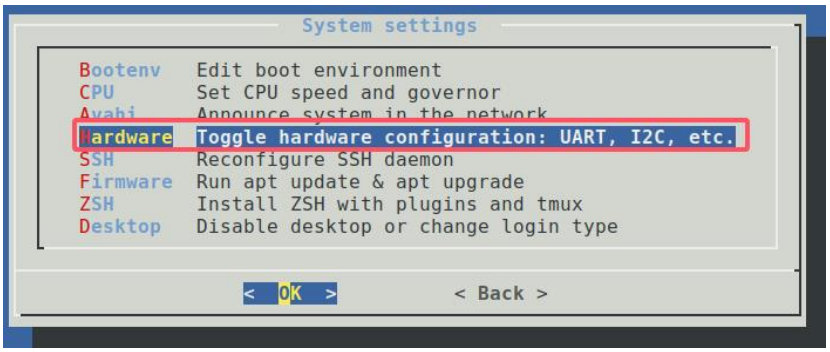
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

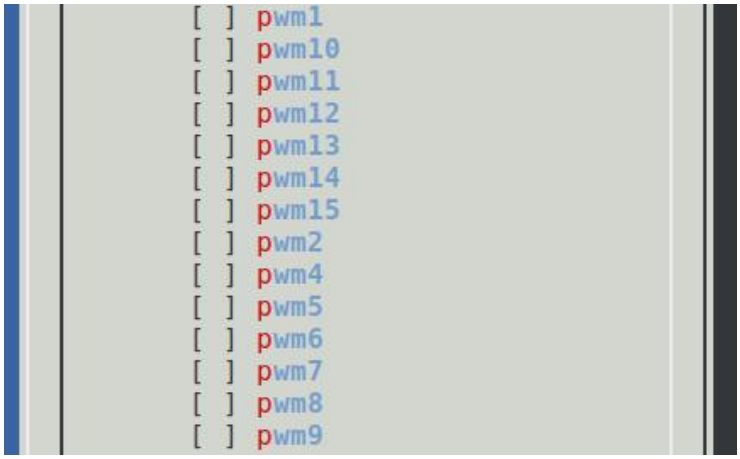
b. Then select **System**



c. Then select **Hardware**



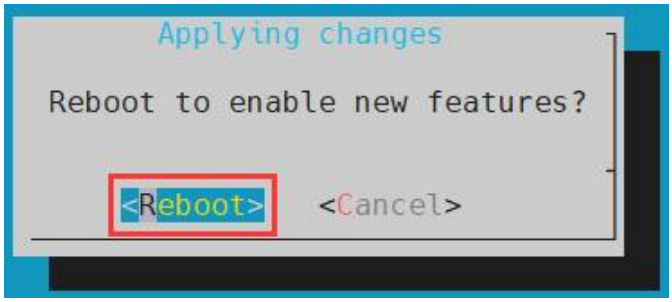
d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the PWM configuration that is turned on



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

f. Then select**<Back>**

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



4) After restarting, you can start the PWM test

Please execute the following commands as the root user.

a. Enter the following command in 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
```

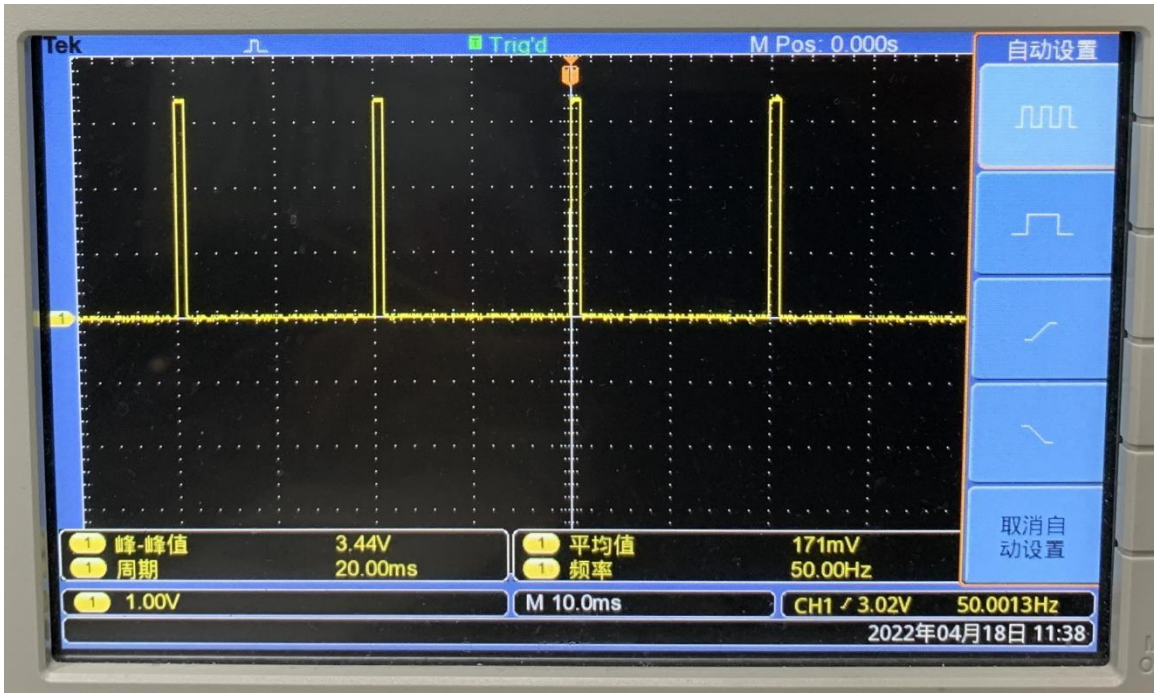
b. Enter the following command in the command line to make pwm2 output a 50Hz square wave

```
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 in 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
```

d. Other PWM test methods are similar and will not be described here.



3. 17. Installation and use of wiringOP-Python

wiringOP-Python is the Python version of wiringOP, which is used to operate the GPIO, I2C, SPI, UART and other hardware resources of the development board in Python programs.

Also note that all the commands below are performed under the root user.

3. 17. 1. Installation of 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 there are no errors during the download process 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, which is stored in **/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 help information is output, it means wiringOP-Python has been successfully installed. 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 in the python command line are as follows:

a. First use the python3 command to enter the python3 command line mode

```
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. 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. 17. 2. 40 pin GPIO port test

WiringOP-Python is the same as wiringOP. It can also determine which GPIO pin to operate by specifying the wPi number. Because there is no command to view the wPi number in wiringOP-Python, the correspondence between the board's wPi number and the physical pin can only be viewed through the gpio command in wiringOP.

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |    SDA.4 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |    SCL.4 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |     PWM8 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.7    | 4   | 46   |
 |   32 |   5 |    TXD.2 |    OFF | 0 | 11 || 12 | 0 | OFF    | PB05     | 6   | 37   |
 |   33 |   7 |    RXD.2 |    OFF | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 |     PB02 |    OFF | 0 | 15 || 16 | 0 | OFF    | PI13     | 9   | 269  |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PI14     | 10  | 270  |
 |  260 |  11 |   MOSI.1 |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  261 |  12 |   MISO.1 |    OFF | 0 | 21 || 22 | 0 | OFF    | TXD.6    | 13  | 262  |
 |  259 |  14 |   SCLK.1 |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 258  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | RXD.6    | 16  | 263  |
 |  265 |  17 |    SDA.5 |    OFF | 0 | 27 || 28 | 0 | OFF    | SCL.5    | 18  | 264  |
 |   35 |  19 |     PB03 |    OFF | 0 | 29 || 30 |   |        | GND      |     |      |
 |   43 |  20 |     PB11 |    OFF | 0 | 31 || 32 | 0 | OFF    | PWM12    | 21  | 267  |
 |  268 |  22 |    PWM13 |    OFF | 0 | 33 || 34 |   |        | GND      |     |      |
 |   38 |  23 |     PB06 |    OFF | 0 | 35 || 36 | 0 | OFF    | PI10     | 24  | 266  |
 |   44 |  25 |     PB12 |    OFF | 0 | 37 || 38 | 0 | OFF    | PB07     | 26  | 39   |
 |      |     |      GND |        |   | 39 || 40 | 0 | OFF    | PB08     | 27  | 40   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
```

1) Below, we take pin 7, which corresponds to GPIO PB4 and wPi number 2, as an example to demonstrate how to set the high and low levels of the GPIO port.

```
orangepi@orangepi4a:~$ gpio readall
 +------+-----+----------+--------+---+  OPI 4A  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  257 |   0 |    SDA.4 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  256 |   1 |    SCL.4 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   36 |   2 |     PWM8 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.7    | 3   | 45   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.7    | 4   | 46   |
```

2) The steps for testing directly using commands are as follows:

a. First, set the GPIO port to output mode. The first parameter of the **pinMode** function is the wPi number 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 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-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 for testing in the python3 command line are as follows:

a. First use the python3 command to enter the python3 command line mode

```
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 wPi number 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 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.

```
>>> 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) wiringOP-Python For setting the GPIO high and low levels in Python code, please refer to the **blink.py** test program in the examples. The **blink.py** test program will set the voltage of all GPIO ports in the 40 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. 17. 3. 40 pin SPI test

1) As shown in the figure below, the available SPIs of Orange Pi 4A are SPI1 and SPI2.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

2) The corresponding pins of SPI1 and SPI2 on 40pin are shown in the following table

	SPI1 corresponds to 40pin	SPI2 corresponds to 40pin
MOSI	**Pin 19**	**Pin 15**
MISO	**Pin 21**	**Pin 29**
CLK	**Pin 23**	**Pin 13**

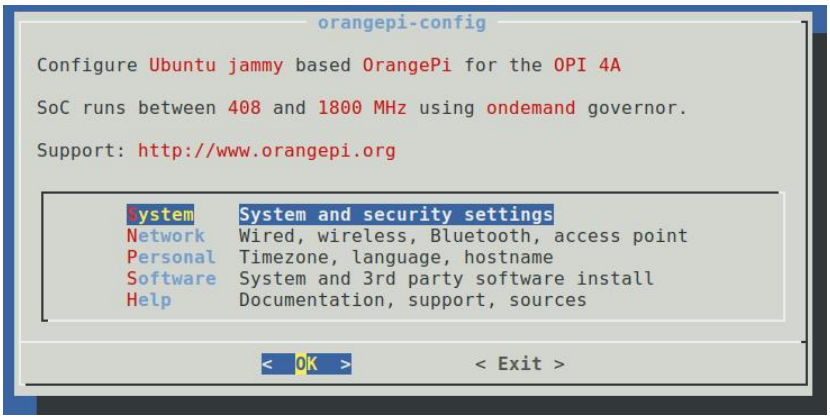
CS0	Pin 24	Pin 11

3) In linux systems, SPI in 40 pins is turned off by default and needs to be manually turned on to use. The detailed steps are as follows:

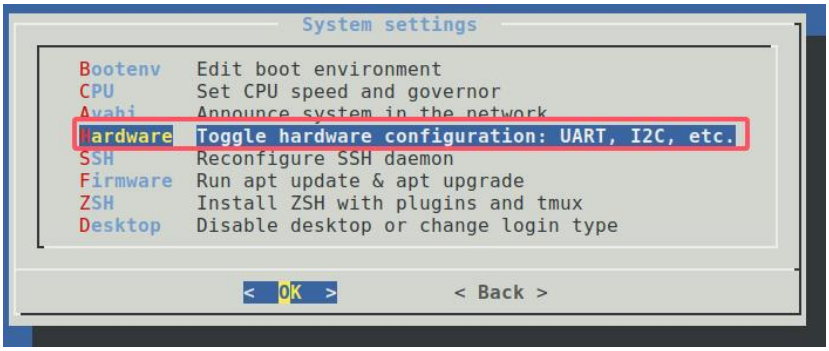
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

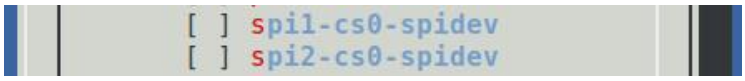
b. Then select **System**



c. Then select **Hardware**



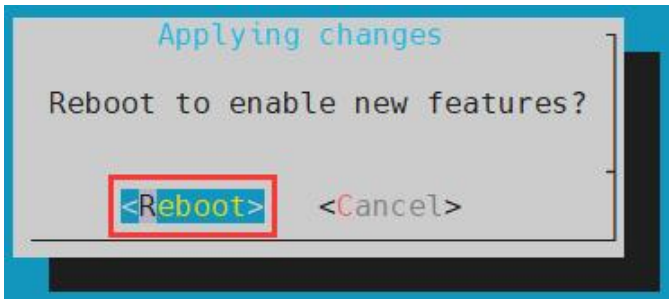
d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the SPI configuration that is open



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

f. Then select**<Back>**

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



4) Then check whether the device node of **spidevx.x** exists in the Linux system. If it exists, it means that the SPI configuration has taken effect.

```
orangepi@orangepi:~$ ls /dev/spidev*
/dev/spidev1.0 /dev/spidev2.0
```

5) Then you can use the **spidev_test.py** program in the examples to test the SPI loopback function. The **spidev_test.py** program needs to specify the following two parameters:

a. **--channel：** Specify the SPI channel number

b. **--port：** Specify the SPI port number

6) Do not short the mosi and miso pins of SPI. The output of running spidev_test.py is as follows. You can see that the data of TX and RX are inconsistent.

The x after the --channel and --port parameters needs to be replaced with the specific SPI channel number and SPI port number.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel x --port x
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    |................|
```

7) Then use the Dupont line to short the SPI's txd and rxd pins and run spidev_test.py. The output is as follows. You can see that the sent and received data are the same, indicating that the SPI loopback test is normal.

The x after the --channel and --port parameters needs to be replaced with the specific SPI channel number and SPI port number.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel x --port x
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    |......@..........|
```

3.17.4. 40 pin I2C test

1) As can be seen from the table below, the available i2c buses for Orange Pi 4A are i2c1, i2c4 and i2c5, a total of three i2c buses.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

2) The corresponding pins of the three groups of I2C buses in 40 pins are shown in the following table.

I2CBus	SDA corresponds to 40pin	SCL corresponds to 40pin	dtbo corresponding configuration
I2C1	Pin 12	Pin 7	i2c1
I2C4	Pin 3	Pin 5	i2c4

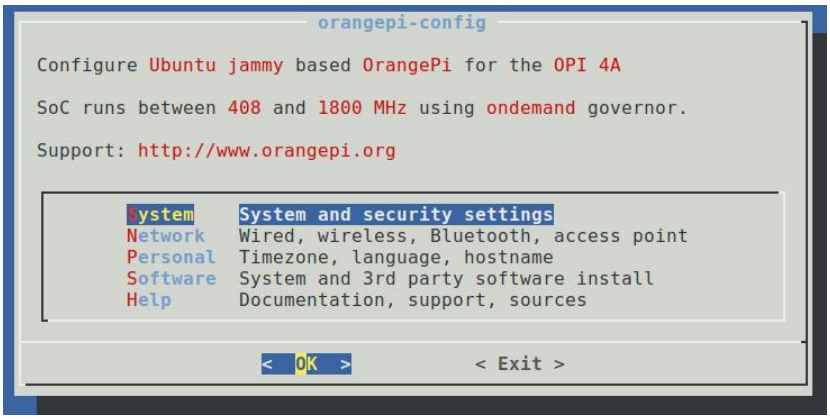
I2C5	Pin 27	Pin 28	i2c5

3) In linux systems, I2C in 40 pins is turned off by default and needs to be manually turned on to use. The detailed steps are as follows:

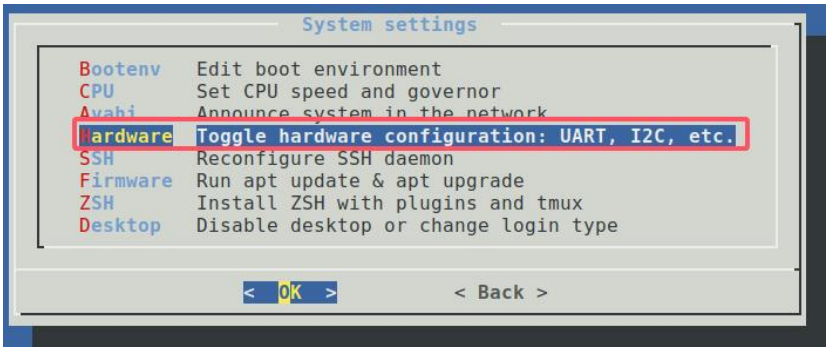
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

b. Then select **System**



c. Then select **Hardware**

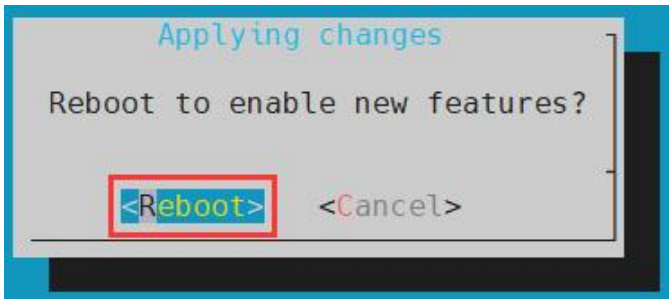


d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the I2C configuration that is open

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

f. Then select**<Back>**

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



4) After starting the Linux system, first confirm that the i2c device node exists under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-*
/dev/i2c-0  /dev/i2c-1  /dev/i2c-3  /dev/i2c-31  /dev/i2c-4  /dev/i2c-5  /dev/i2c-6
```

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

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

6) Then connect an i2c device to the i2c pin of the 40 pin connector. Here we take the DS1307 RTC module as an example.

7) Then use the **i2cdetect -y** 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 1      #i2c1 Command
orangepi@orangepi:~$ sudo i2cdetect -y 4      #i2c4 Command
```

orangepi@orangepi:~$ **sudo i2cdetect -y 5** #i2c5 Command

```
root@orangepi4a:~# i2cdetect -y -r 1
     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: -- -- -- -- -- -- -- --
```

8) 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-1"
Thu 2022-06-16 04:35:46
Thu 2022-06-16 04:35:47
Thu 2022-06-16 04:35:48
^C
exit

3.17.5. 40 pin 的 UART test

1) As can be seen from the table below, Orange Pi 4A has six uart buses: uart2, uart3, uart4, uart5, uart6 and uart7

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4-TX	TWI4_SDA	PI1	257	3	4		5V			
PWM0-1	UART4-RX	TWI4_SCK	PI0	256	5	6		GND			
	TWI1-SCK	PWM0-8	PB4	36	7	8	45	PB13	UART7-TX	PWM0-4	
			GND		9	10	46	PB14	UART7-RX	PWM0-5	
SPI2-CS0	PWM0-6	UART2-TX	PB0	32	11	12	37	PB5	PWM0-9	TWI1-SDA	
SPI2-CLK	PWM0-7	UART2-RX	PB1	33	13	14		GND			
SPI2-MOSI			PB2	34	15	16	269	PI13			PWM0-14
			3.3V		17	18	270	PI14			PWM0-15
		SPI1_MOSI	PI4	261	19	20		GND			
		SPI1_MISO	PI5	260	21	22	258	PI6		UART6-TX	
UART5-RX		SPI1_CLK	PI3	259	23	24	262	PI2	SPI1_CS0		UART5-TX
			GND		25	26	263	PI7		UART6-RX	
		TWI5-SDA	PI9	265	27	28	264	PI8	TWI5-SCK		
SPI2-MISO			PB3	35	29	30		GND			
	PWM0-2		PB11	43	31	32	267	PI11	PWM0-12	UART3-TX	
	UART3-RX	PWM0-13	PI12	268	33	34		GND			
		PWM0-10	PB6	38	35	36	266	PI10			
			PB12	44	37	38	39	PB7	PWM0-11		
			GND		39	40	40	PB8			

2) The corresponding pins of the six UART bus groups in the 40 pins are shown in the following table.

UART Bus	RX corresponds to 40pin	TX corresponds to 40pin	dtbo corresponding configuration

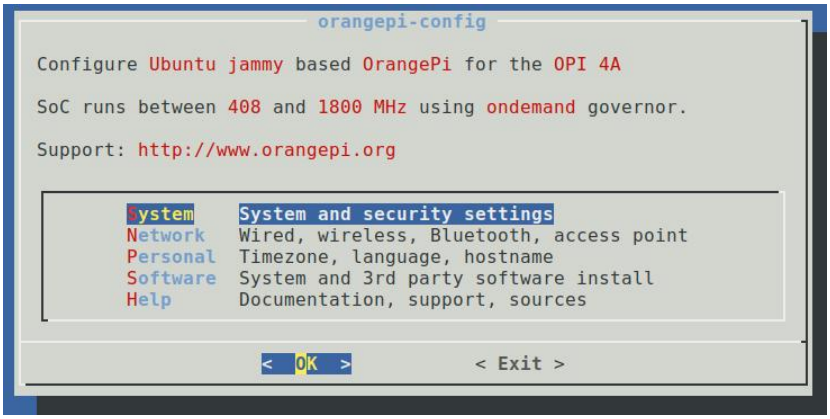
UART2	Pin 13	Pin 11	uart2
UART3	Pin 33	Pin 32	uart3
UART4	Pin 5	Pin 3	uart4
UART5	Pin 23	Pin 24	uart5
UART6	Pin 26	Pin 22	uart6
UART7	Pin 10	Pin 8	uart7

3) In linux systems, UART in 40 pins is turned off by default and needs to be manually turned on to use. The detailed steps are as follows:

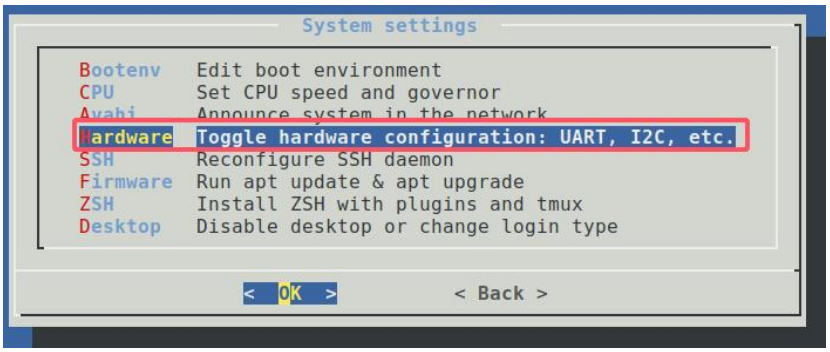
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

b. Then select **System**



c. Then select **Hardware**

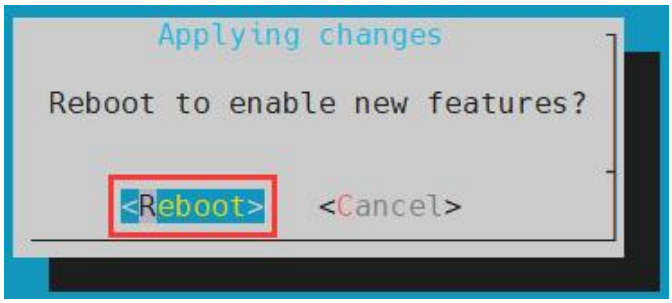


d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the UART configuration that is turned on

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

f. Then select**<Back>**

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



4) After entering the Linux system, first confirm whether there is a uart device node under **/dev**

```
orangepi@orangepi:~$ ls /dev/ttyAS*
/dev/ttyAS0  /dev/ttyAS1  /dev/ttyAS2  /dev/ttyAS3  /dev/ttyAS4  /dev/ttyAS5
/dev/ttyAS6  /dev/ttyAS7
```

5) Then start testing the UART interface. First use the Dupont line to short-circuit the rx and tx of the UART interface to be tested

6) Finally, you can run the **serialTest.py** program in examples to test the loopback function of the serial port. If you can see the following print, it means that the serial port loopback test is normal (ttyASX needs to be replaced with the corresponding uart node name, please do not copy it)

```
root@orangepi:~/wiringOP-Python# cd examples
```

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device /dev/ttyASX
Out:    0: ->    0
Out:    1: ->    1
Out:    2: ->    2
Out:    3: ->    3
Out:    4:^C
exit
```

3.18. Hardware watchdog test

The Linux system released by Orange Pi has the watchdog_test program pre-installed, which can be used for direct testing.

The method to run the watchdog_test program is as follows:

a. The second parameter 10 represents the watchdog count time. If the watchdog is not fed within this time, the system will restart.
b. We can feed the dog by pressing any key on the keyboard (except ESC). After feeding the dog, the program will print a line of keep alive to indicate that the dog was successfully fed.

```
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
```

3.19. Check the chipid of T527 chip

The command to view the chipid of the T527 chip is as follows. The chipid of each

chip is different, so the chipid can be used to distinguish multiple development boards.

```
orangepi@orangepi:~# cat /sys/class/sunxi_info/sys_info |grep sunxi_serial
sunxi_serial            : 208d211475779d0c0000019000000000
```

3. 20. Python related instructions

3. 20. 1. How to compile and install Python source code

If the Python version in the Ubuntu or Debian system software repository does not meet the development requirements and you want to use the latest version of Python, you can use the following method to download the Python source package to compile and install the latest version of Python.

The following demonstrates compiling and installing the latest version of Python 3.9. If you want to compile and install other versions of Python, the method is the same (you need to download the source code of the Python you want to install).

1) First install the dependency packages 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 Python 3.9 source code and unzip 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:~/Python-3.9.10$ ./configure --enable-optimizations
```

4) Then compile and install Python 3.9. The compilation time takes about half an hour.

```
orangepi@orangepi:~/Python-3.9.10$ make -j4
orangepi@orangepi:~/Python-3.9.10$ sudo make altinstall
```

5) After installation, you can use the following command to view the version number of

Python just installed

```
orangepi@orangepi:~/Python-3.9.10$ python3.9 --version
Python 3.9.10
```

6) Then update pip

```
orangepi@orangepi:~$ /usr/local/bin/python3.9 -m pip install --upgrade pip
```

3. 20. 2. How to change pip source in Python

The default source used by pip in Linux system is the official source of Python, but the speed of accessing the official source of Python in China is very slow, and the installation of Python packages often fails due to network problems. So when using pip to install Python libraries, please remember to change the pip source.

1) First install **python3-pip**

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3-pip
```

2) How to permanently change pip source under Linux

a. First create a new **~/.pip** directory, then add the **pip.conf** configuration file and set the source of pip to 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 use pip3 to install the Python library quickly.

3) How to temporarily change the pip source under Linux, where **<packagename>** needs to be replaced with the specific package name

```
orangepi@orangepi:~$ pip3 install <packagename> -i  \
https://pypi.tuna.tsinghua.edu.cn/simple --trusted-host pypi.tuna.tsinghua.edu.cn
```

3. 21. How to install Docker

The Linux image provided by Orange Pi has Docker pre-installed, but the Docker service is not enabled by default. Use the **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 started the next time you start the system.

```
orangepi@orangepi:~$ enable_docker.sh
```

You can use the following command to test docker. If you can run **hello-world**, it means 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 **permission denied** is prompted, 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 and log in again to take effect, or restart the system.

3. 22. How to install Home Assistant

Note that this article will only provide methods for installing Home Assistant in Ubuntu or Debian systems. For detailed usage of Home Assistant, please refer to the official documentation or corresponding books.

3. 22. 1. Install via Docker

1) First, please install Docker and make sure it can run normally. For the installation steps of Docker, please refer to the instructions in the section **How to install Docker**.

2) Then you can search for Home Assistant's docker image

```
orangepi@orangepi:~$ docker search homeassistant
```

3) Then use the following command to download the Home Assistant docker image to your local computer. The image size is about 1GB and the download time will be longer. 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 that you 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
```

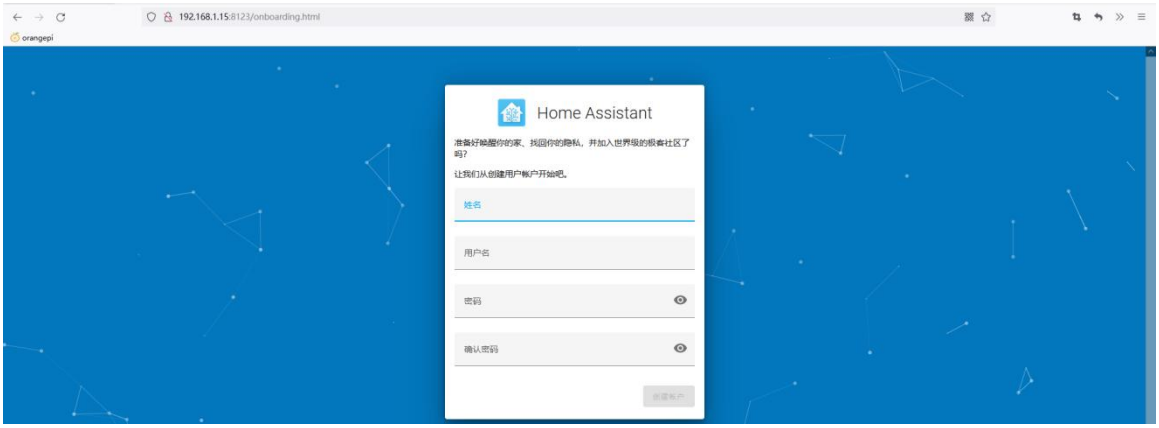
5) Now you can run the Home Assistant docker container

```
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
```

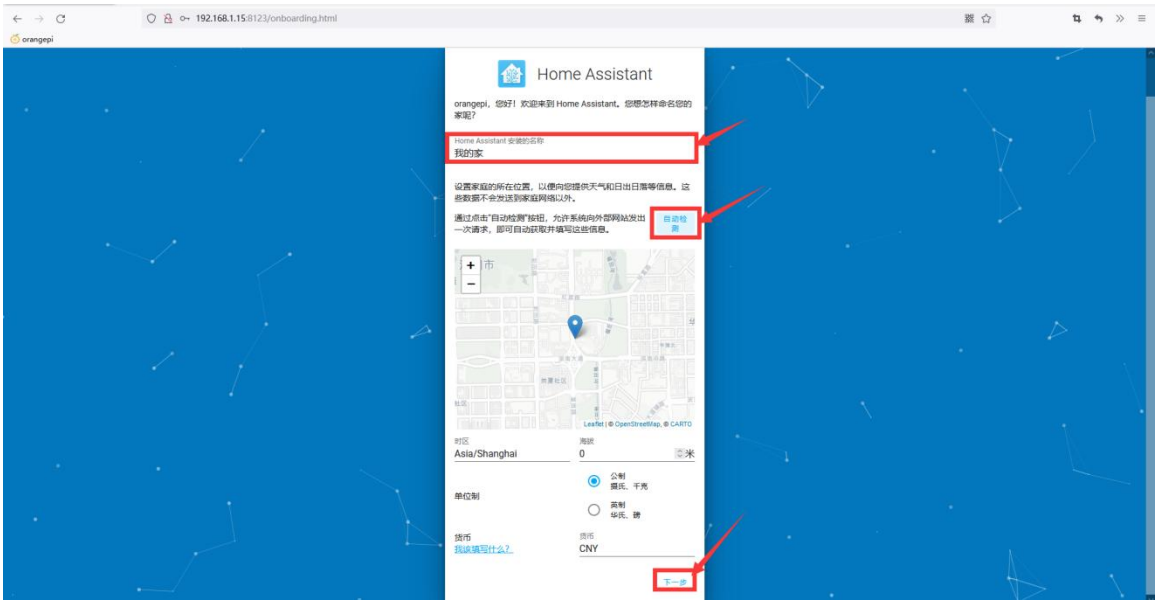
6) Then enter [IP address of the development board: 8123] in the browser to see the Home Assistant interface

It takes a while for the Home Assistant container to start. If the following interface does not display normally, please wait a few seconds and refresh it. If the following interface does not display normally after waiting for more than a minute, it means that there is a problem with the Home Assistant installation. At this time, you need to check whether there is a problem with the previous installation and setup process.

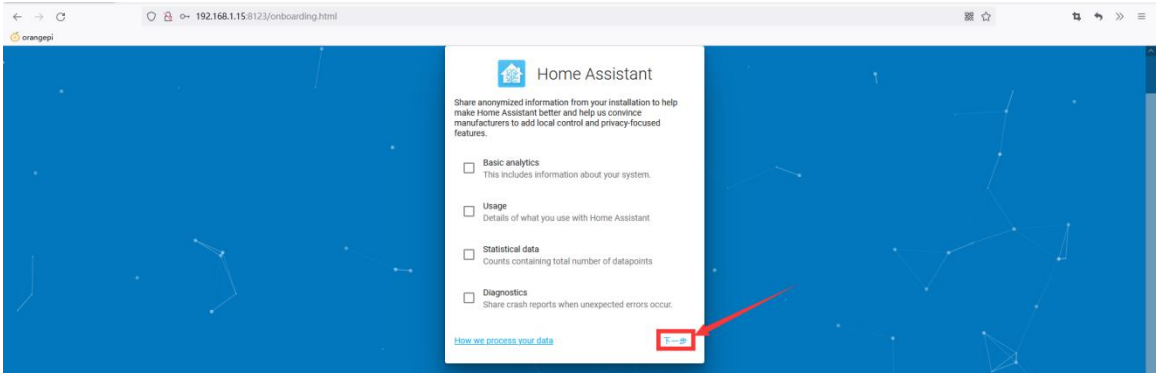


7) Then enter your **name**, **username** and **password** and click **Create Account**

8) Then follow the interface prompts to set according to your preferences, and then click Next

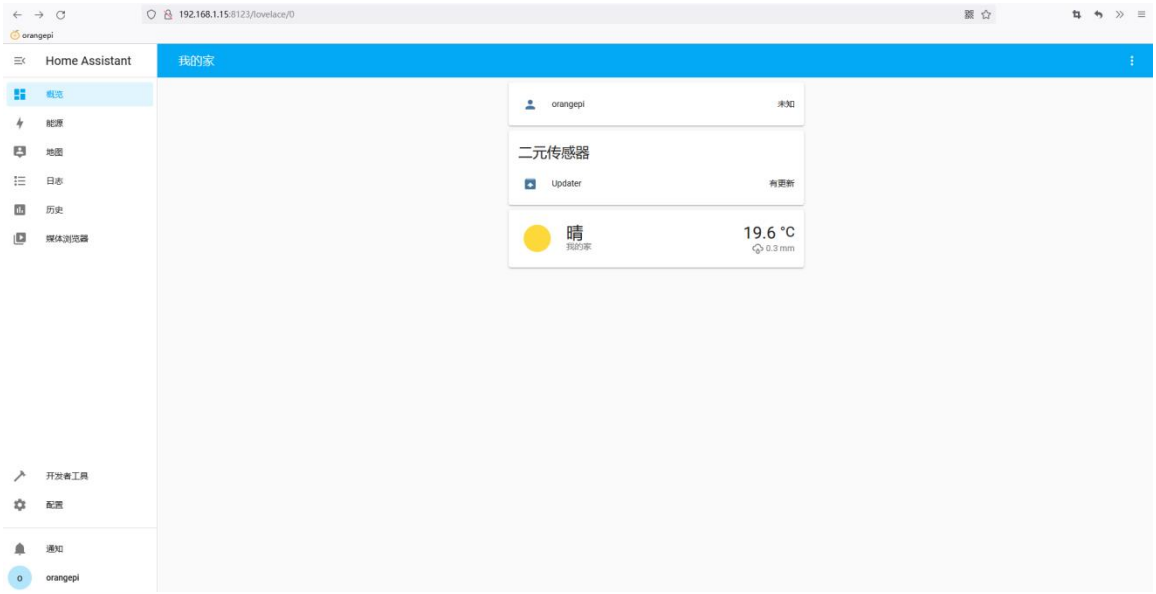


9) Then click Next



10) Then click Finish

11) The main interface of Home Assistant is shown 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
```

3.22.2. Installation via Python

Before installation, please change the source of pip to the domestic source to speed up the installation of the Python package. For the configuration method, see the section How to change the pip source in Python.

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 Python 3.9. For more information, please refer to the section on **compiling and installing Python source code**.

The default Python version of Ubuntu Jammy is Python 3.10, so there is no need to compile and install it.

The default Python version of Debian Bookworm is Python 3.11, so there is no need to compile and install it.

3) Then create a Python virtual environment

In Debian Bookworm, it is python3.11, so please remember to replace the corresponding commands.

```
orangepi@orangepi:~$ sudo mkdir /srv/homeassistant
orangepi@orangepi:~$ sudo chown orangepi:orangepi /srv/homeassistant
orangepi@orangepi:~$ cd /srv/homeassistant
orangepi@orangepi:/srv/homeassistant$ python3.9 -m venv .
orangepi@orangepi:/srv/homeassistant$ source bin/activate
(homeassistant) orangepi@orangepi:/srv/homeassistant$
```

4) Then install the required Python packages

```
(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
```

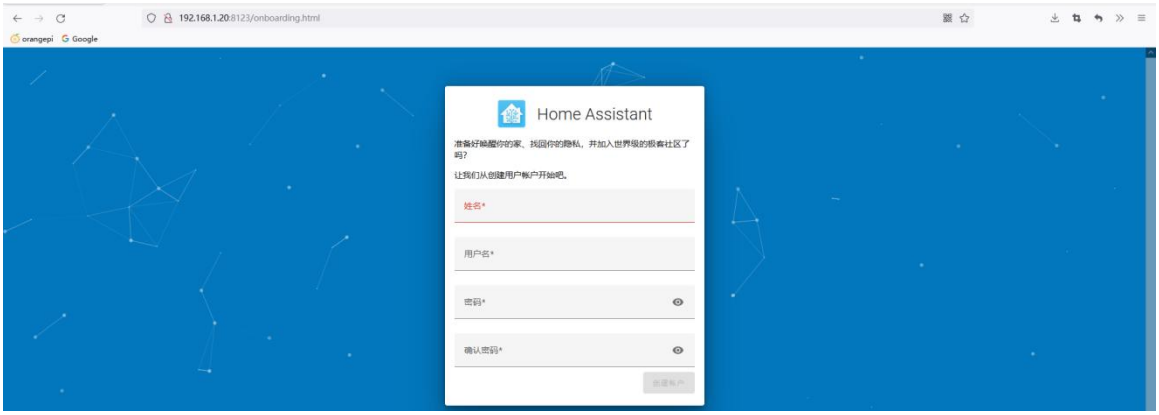
6) Then enter the following command to run Home Assistant Core

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ hass
```

7) Then enter **[IP address of the development board: 8123]** in the browser to see the Home Assistant interface

When you run the hass command for the first time, it will download, install, and cache some necessary libraries and dependencies. This process may take several

minutes. Note that you will not be able to see the Home Assistant interface in your browser at this time. Please wait for a while before refreshing.



3. 23. OpenCV installation method

3. 23. 1. Install OpenCV using apt

1) The installation command is as follows

```
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 version number of OpenCV. The output is normal, indicating that OpenCV is installed successfully.

a. The version of OpenCV in Ubuntu 22.04 is as follows:

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.5.4
```

b. The version of OpenCV in Debian 12 is as follows:

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.6.0
```

3. 24. How to install Baota Linux Panel

Baota 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, you need to expand the size of the **/tmp** space. After setting, you need to **restart the Linux system of the development board**. The command is as follows：

```
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 baota

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

4) Then the Baota installation program will prompt whether to install **Bt-Panel** to the **/www** folder, just enter y

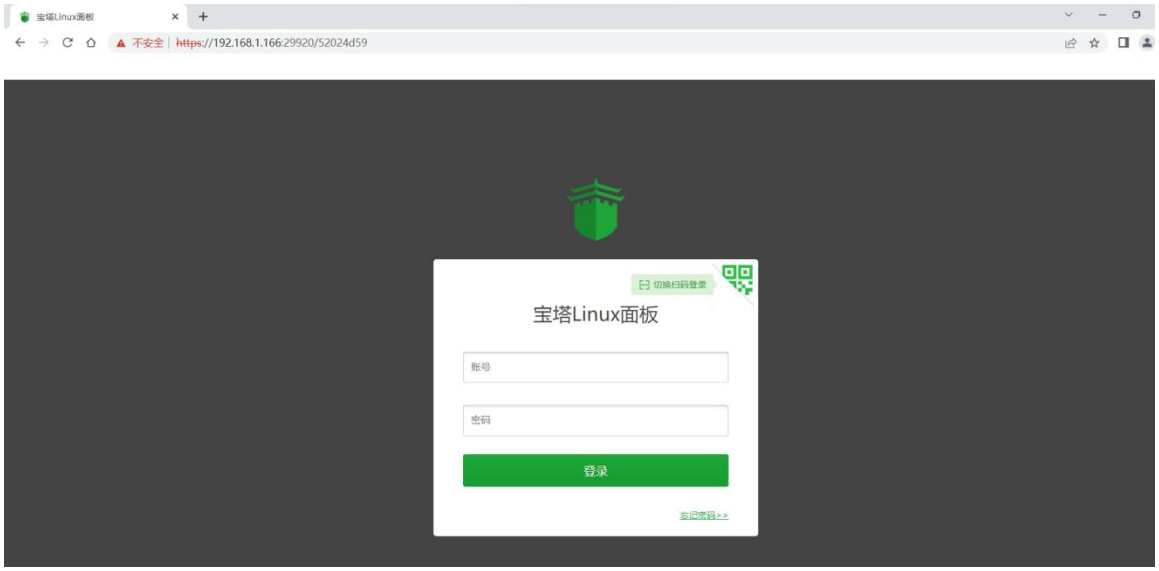
```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| 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
```

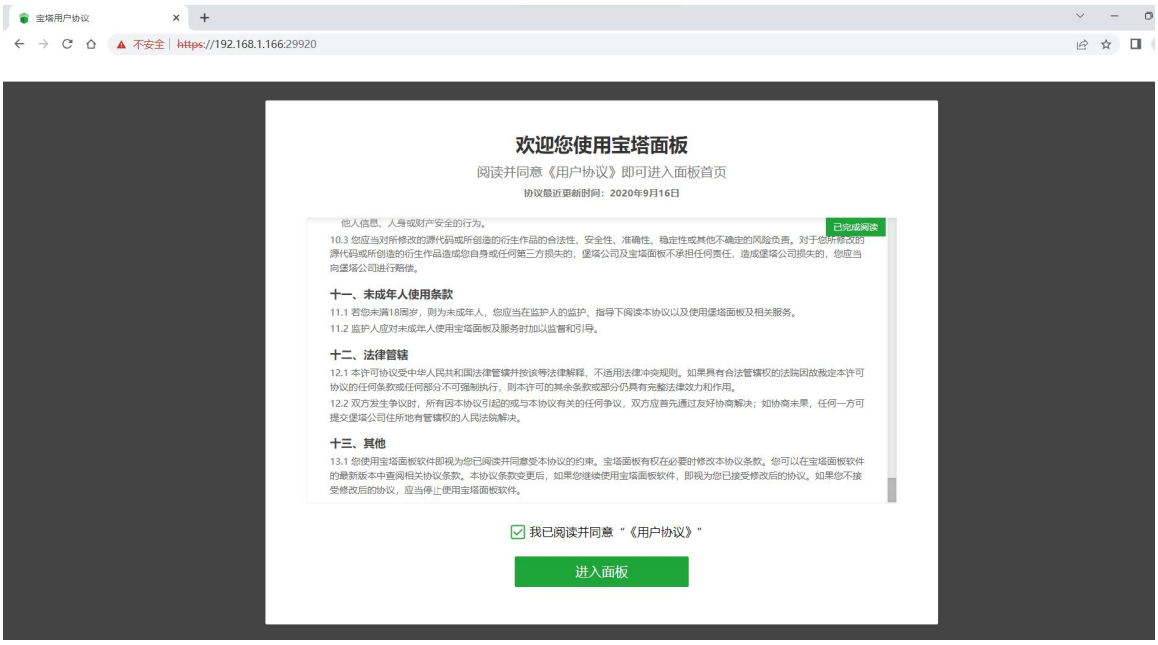
5) Then all you have to do is wait patiently. When you see the following print information output by the terminal, it means that the pagoda has been installed. The entire installation process takes about 34 minutes, which may vary depending on the network speed.

```
==========================================================
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:~$
```

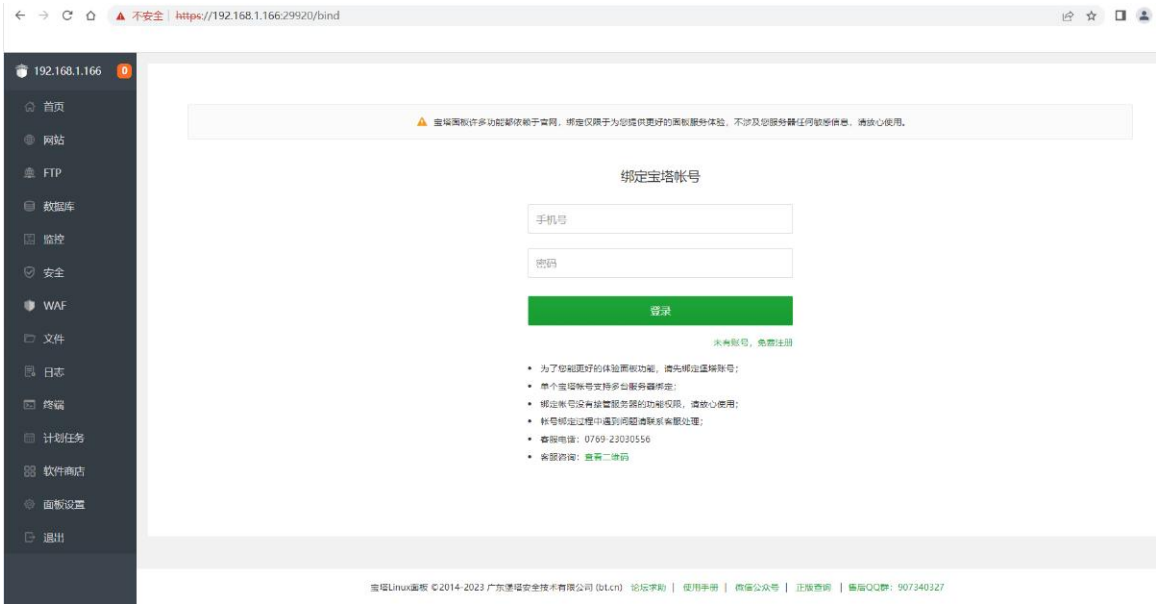
6) At this time, enter the **panel address** shown above in the browser to open the login interface of the Baota Linux panel, and then enter the **username** and **password** shown in the above figure in the corresponding position to log in to Baota



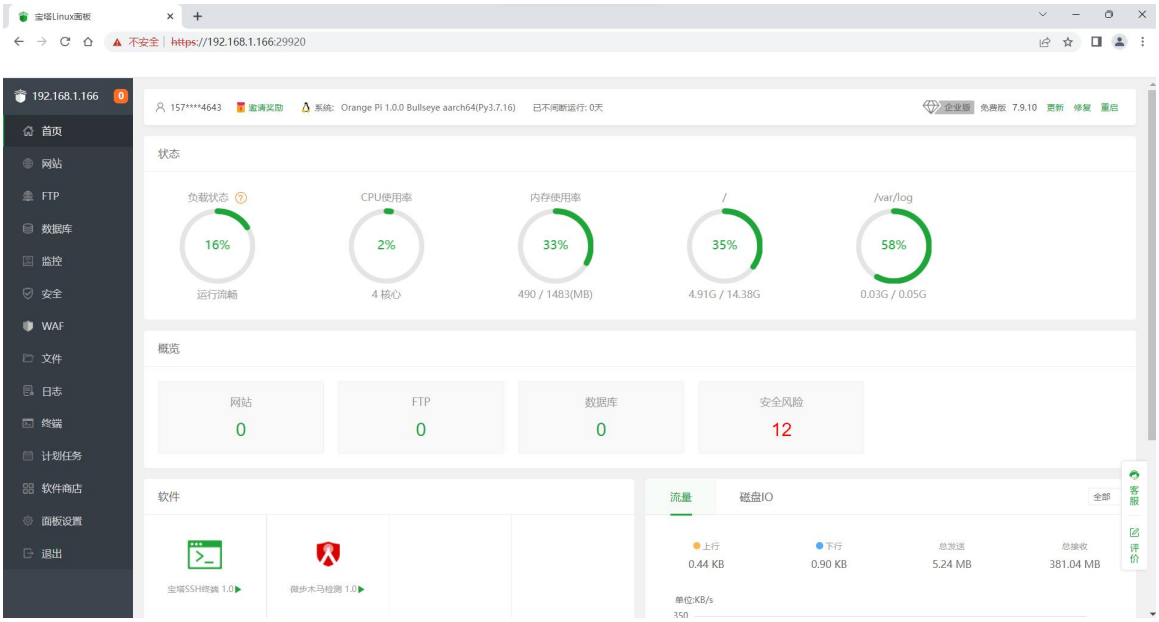
7) After successfully logging into the pagoda, the following welcome interface will pop up. First, please read the user instructions in the middle and drag them to the bottom. Then you can select "I have agreed and read the User Agreement", and then click "Enter the Panel" to enter the baota.



8) After entering the pagoda, you will be prompted to bind an account on the pagoda official website. If you do not have an account, you can go to the pagoda official website (**https://www.bt.cn**) to register one.



9) The final interface is shown in the figure below. You can intuitively see some status information of the development board Linux system, such as load status, CPU usage, memory usage, and storage space usage



10) For more functions of the pagoda, please refer to the following information to explore it yourself

User Manual：**http://docs.bt.cn** Forum Address：**https://www.bt.cn/bbs** GitHub Link：**https://github.com/aaPanel/BaoTa**

3. 25. QT installation method

1) Use the following script to install QT5 and QT Creator

```
orangepi@orangepi:~$ install_qt.sh
```

2) After installation, the QT version number will be automatically printed

a. The QT version that comes with Ubuntu 22.04 is **5.15.3**

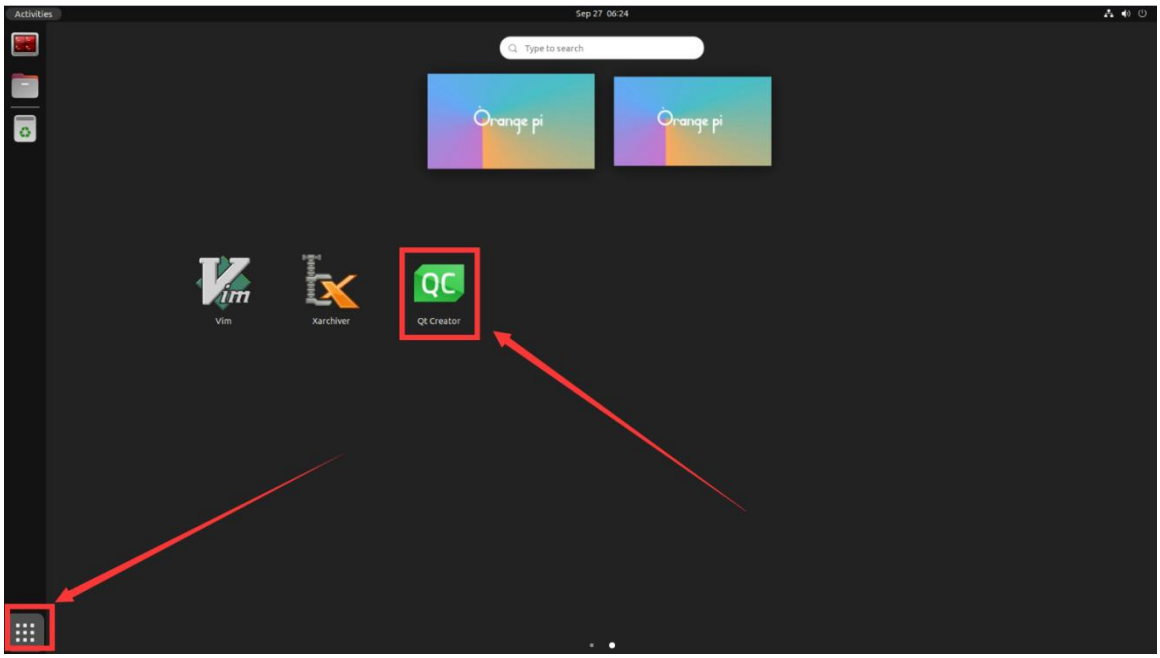
```
orangepi@orangepi:~$ install_qt.sh
.....
QMake version 3.1
Using Qt version 5.15.3 in /usr/lib/aarch64-linux-gnu
```

b. 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
```

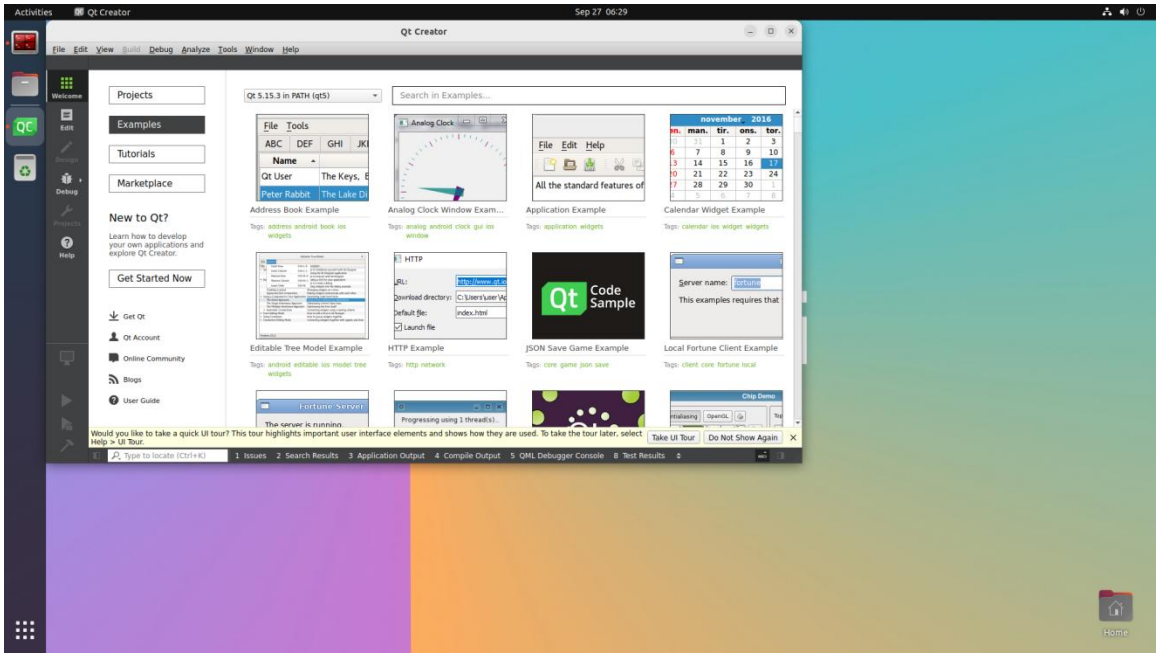
3) Then you can see the QT Creator startup icon in the **Applications** list



You can also use the following command to open QT Creator

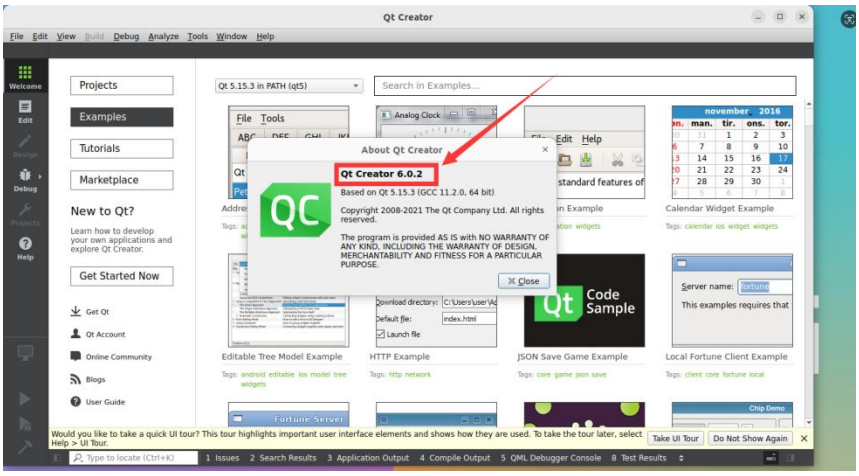
```
orangepi@orangepi:~$ qtcreator
```

4) The interface after QT Creator is opened is as follows

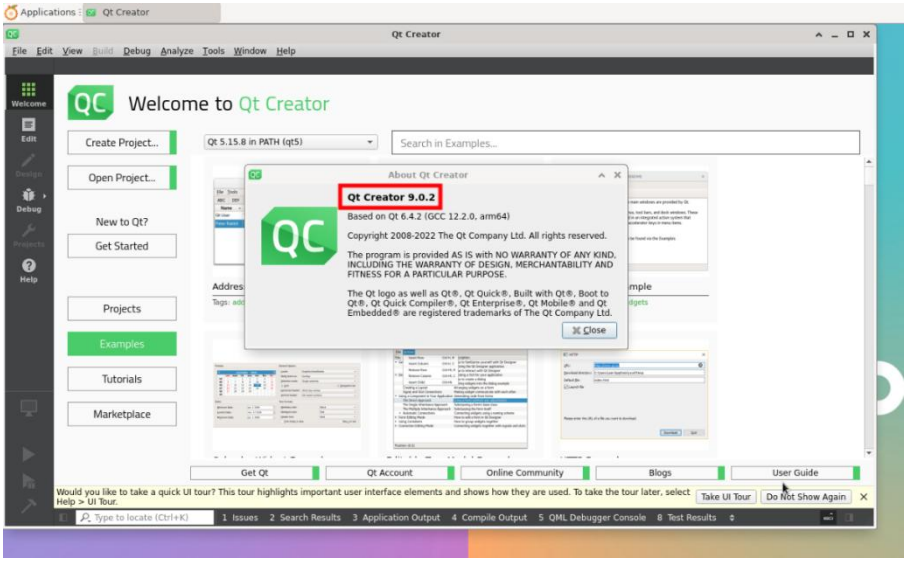


5) The version of QT Creator is as follows

a. The default version of QT Creator in **Ubuntu22.04** is as follows

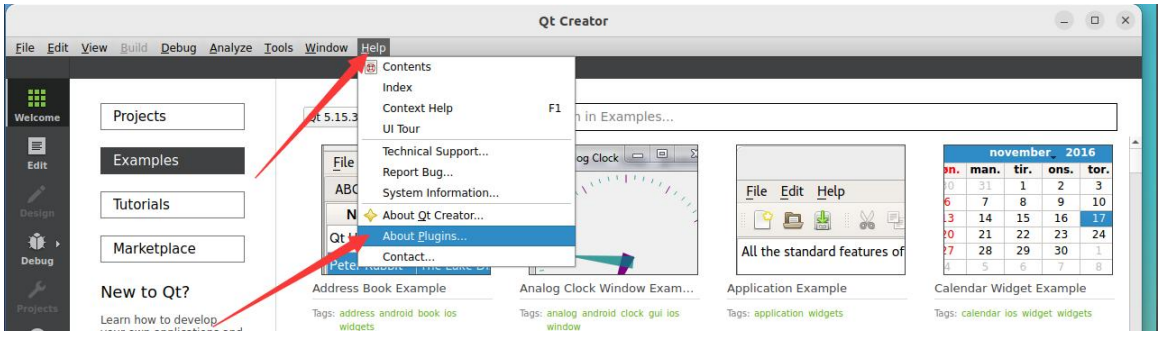


b. The default version of QT Creator in **Debian12** is as follows

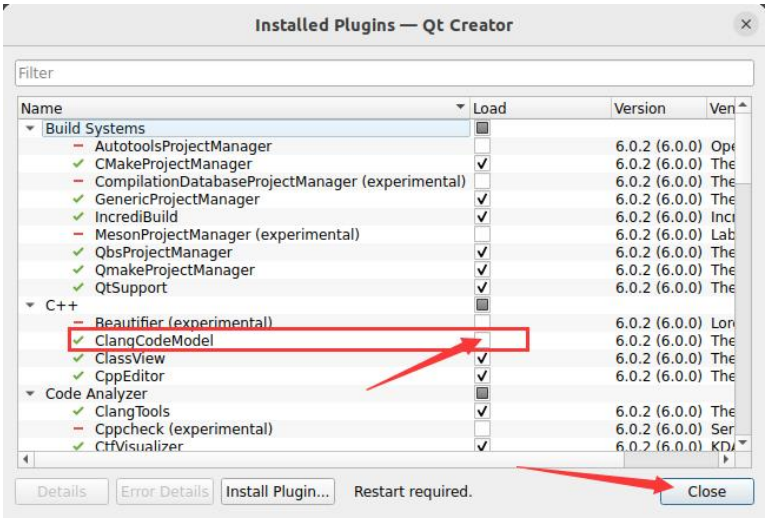


6) Then set up QT

a. First open **Help->About Plugins...**



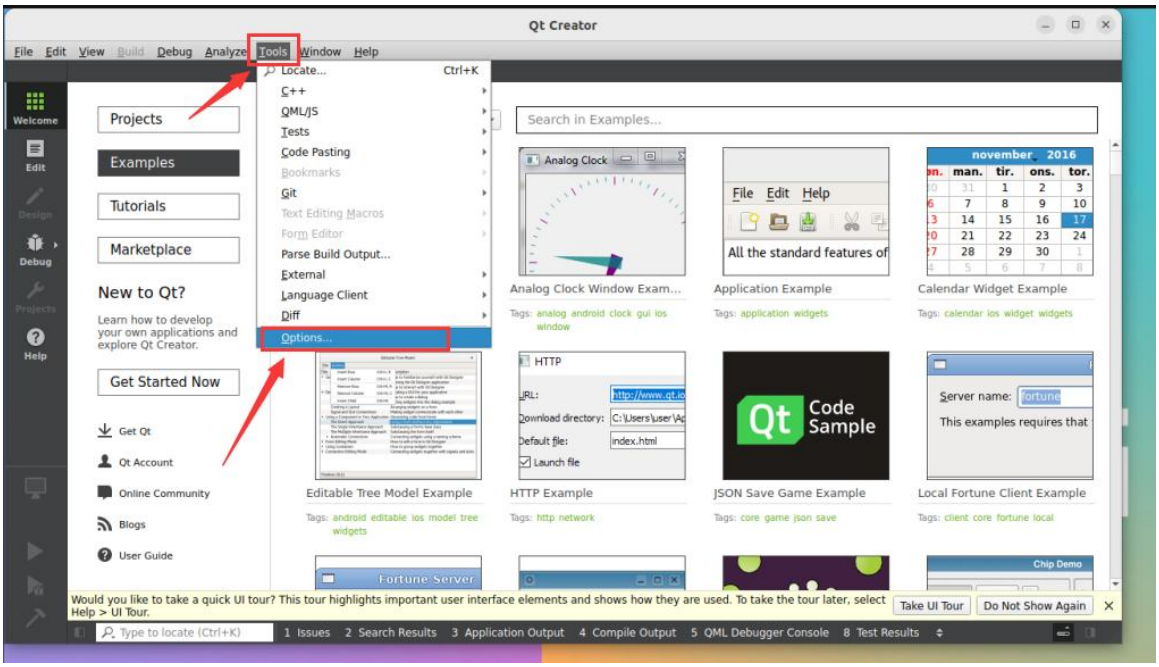
b. Then remove the check mark of **ClangCodeModel**

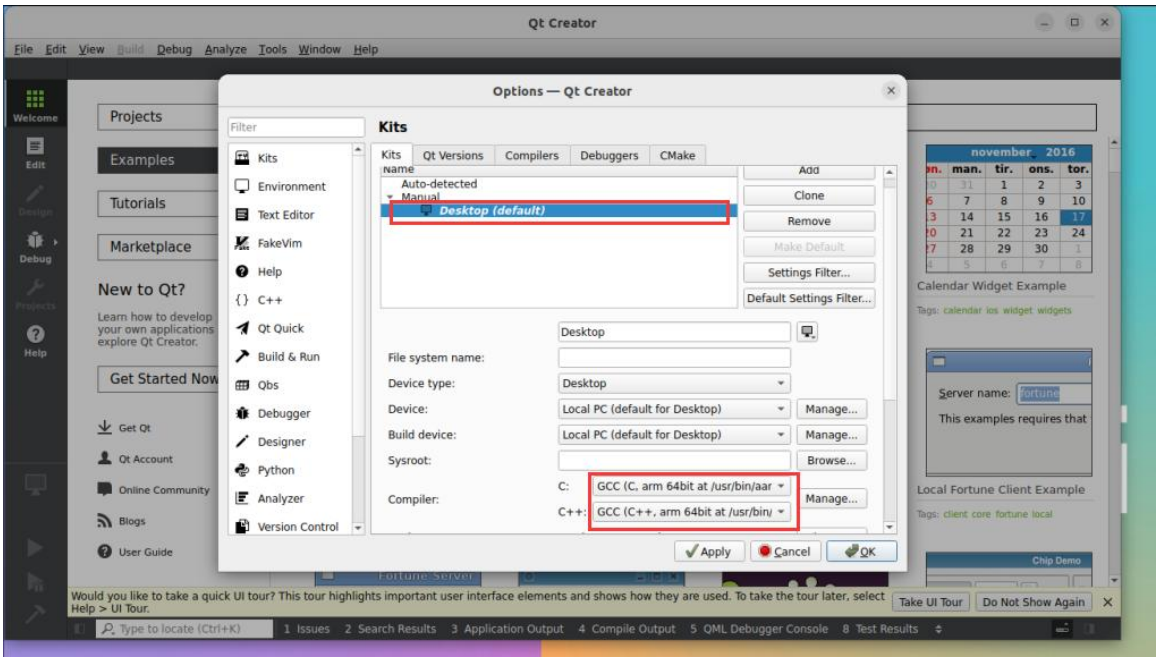


c. **After setting, you need to restart QT Creator**

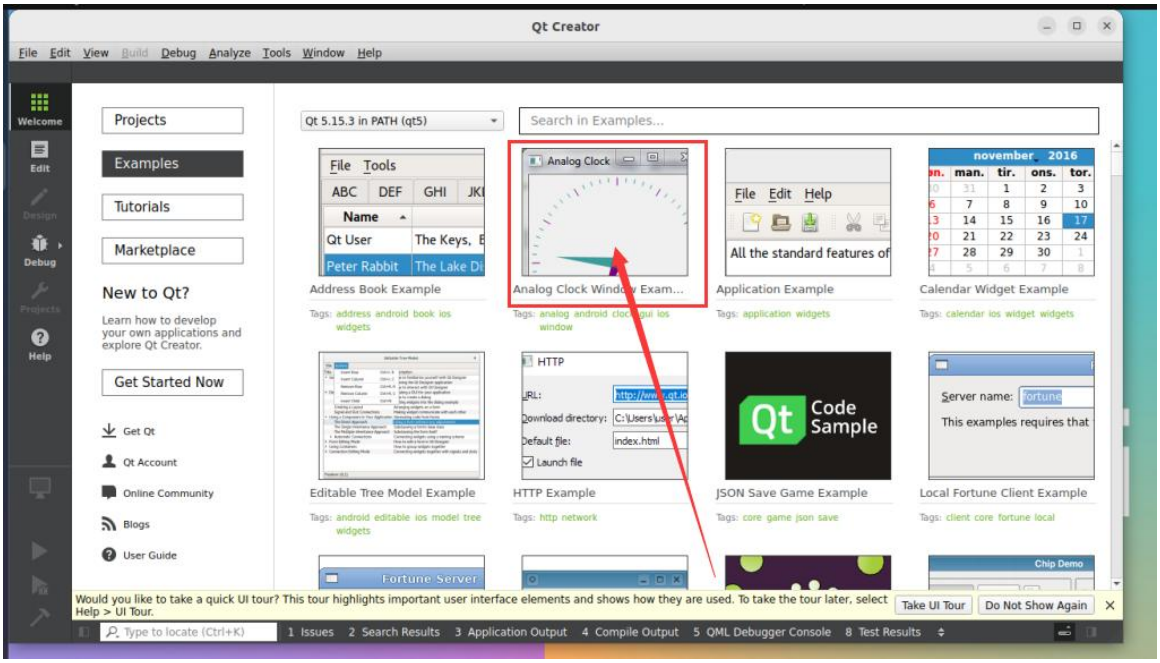
d. Then make sure that QT Creator uses the GCC compiler. If it defaults to Clang, change it to GCC

For Debian 12, please skip this step.

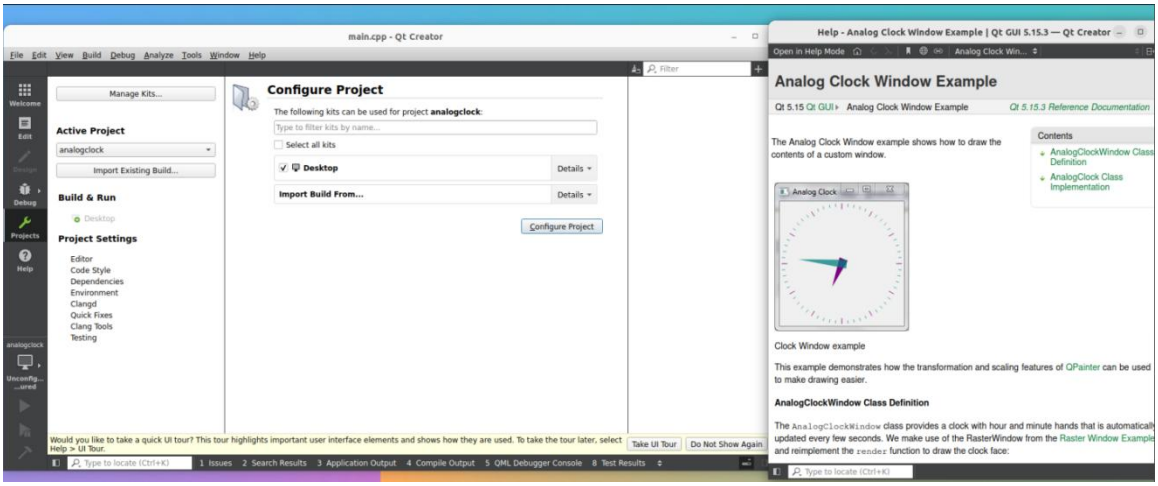




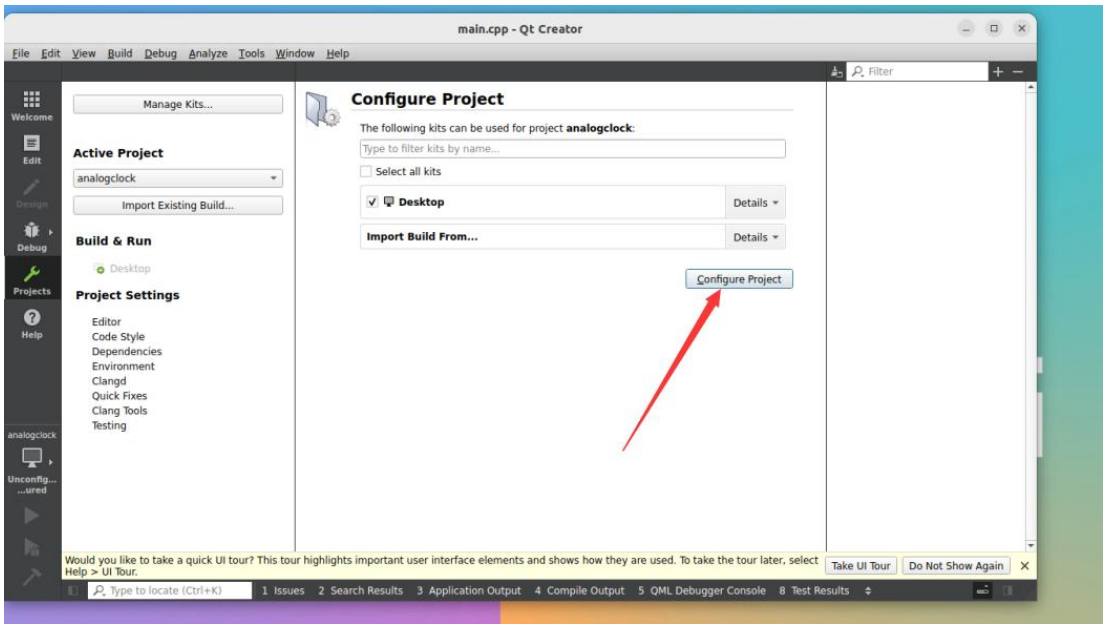
7) Then you can open a sample code



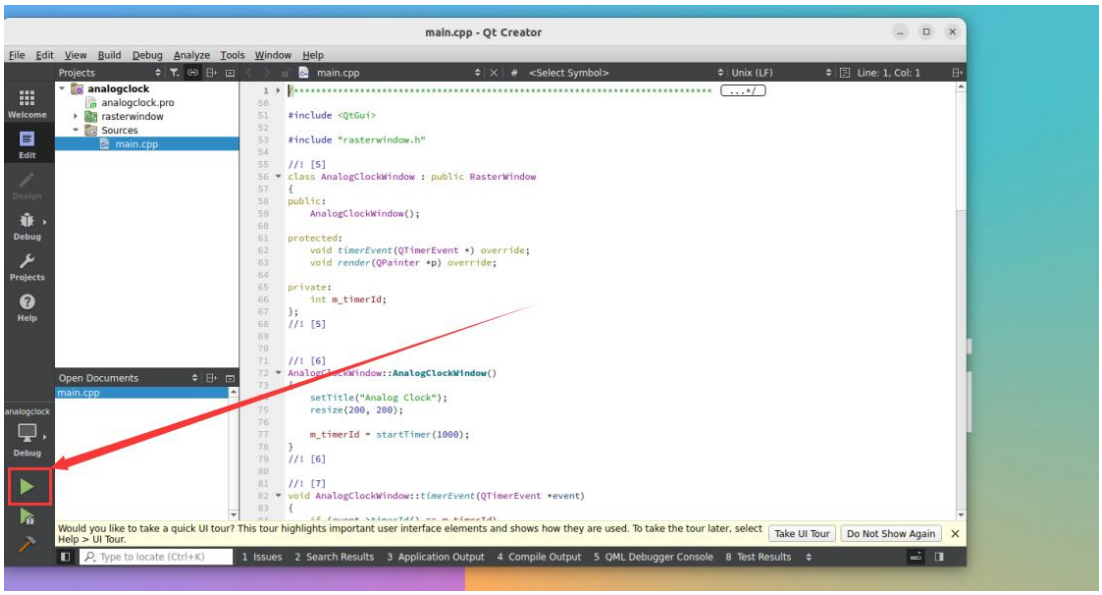
8) Clicking on the sample code will automatically open the corresponding documentation. Please read the instructions carefully.



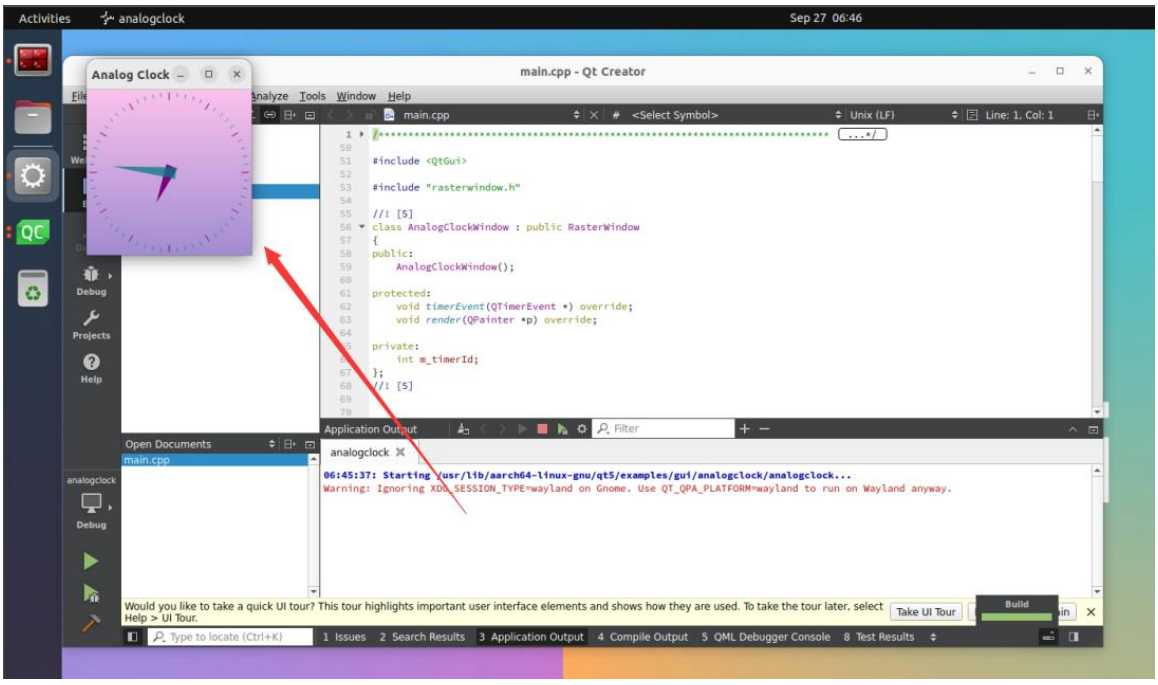
9) Then click Next **Configure Project**



10) Then click the green triangle in the lower left corner to compile and run the sample code



11) After waiting for a while, 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. 26. ROS installation method

3. 26. 1. How to install ROS 2 Humble on Ubuntu 22.04

1) Use the **install_ros.sh** script to install ros2

```
orangepi@orangepi:~$ install_ros.sh ros2
```

2) After the**install_ros.sh** script installs ros2, it will automatically run the**ros2 -h** command. If you can see the following print, it means that ros2 is installed successfully.

```
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`
```

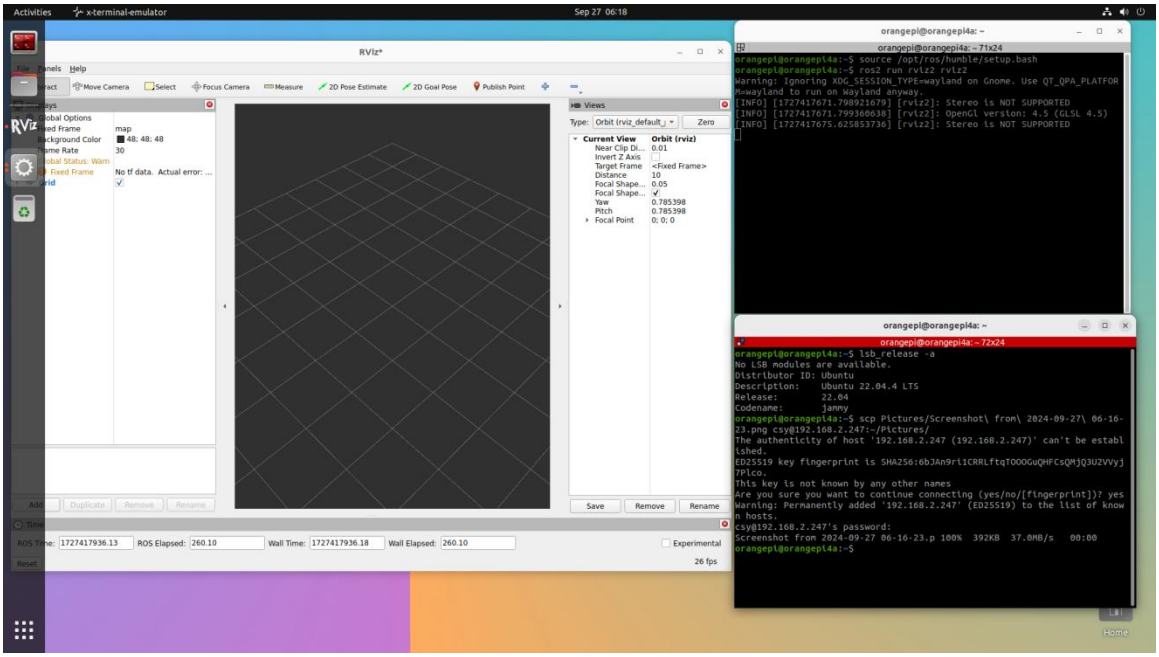
```
Call `ros2 <command> -h` for more detailed usage.
```

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. 27. How to install kernel header files

1) The Linux image released by OPi comes with a deb package of kernel header files by default, which is stored in **/opt/**

```
orangepi@orangepi:~$ ls /opt/linux-headers*
/opt/linux-headers-xxx-sun55iw3_x.x.x_arm64.deb
```

2) Use the following command to install the kernel header file deb package

```
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 it.

```
orangepi@orangepi:~$ cd /usr/src/hello/
orangepi@orangepi:/usr/src/hello$ sudo make
make -C /lib/modules/5.15.147-sun55iw3/build M=/usr/src/hello modules
make[1]: Entering directory '/usr/src/linux-headers-5.15.147-sun55iw3'
  CC [M]  /usr/src/hello/hello.o
  MODPOST /usr/src/hello/Module.symvers
  CC [M]  /usr/src/hello/hello.mod.o
  LD [M]  /usr/src/hello/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.15.147-sun55iw3'
```

5) After compilation, the **hello.ko** kernel module will be generated

```
orangepi@orangepi:/usr/src/hello$ ls *.ko
hello.ko
```

6) Use the **insmod** command to 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 following output, it means that 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. 28. How to use the 10.1 inch MIPI LCD screen

3. 28. 1. 10.1 inch MIPI screen assembly method

1) First prepare the necessary accessories

a. 10.1 inch MIPI LCD display + touch screen

b. Screen adapter board + 31pin to 40pin cable

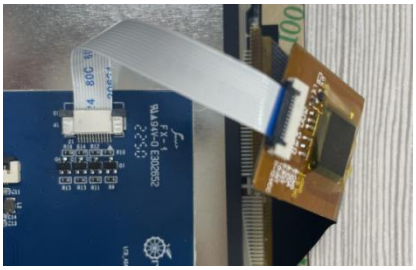
c. 30pin MIPI cable

d. 12pin touch screen cable

2) Connect the 12-pin touch screen cable, 31-pin to 40-pin cable, and 30-pin MIPI cable to the screen adapter board as shown below. Note that the blue insulation side of the touch screen cable should face down, and the insulation sides of the other two cables should face up. If connected incorrectly, it will cause no display or inability to touch.

3) Place the adapter board with the connected cable on the MIPI LCD screen as shown below, and connect the MIPI LCD screen and the adapter board via a 31pin to 40pin cable.

4) Then connect the touch screen and the adapter board through the 12-pin touch screen cable, paying attention to the direction of the insulating surface

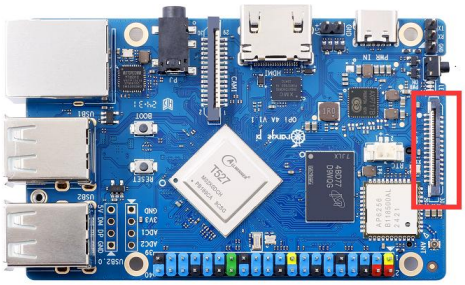


5) Finally, connect it to the LCD interface of the development board through the 30pin MIPI cable

3. 28. 2. How to open the 10.1-inch MIPI LCD screen configuration

1) The Linux image does not have the mipi lcd screen configuration turned on by default. If you need to use the mipi lcd screen, you need to turn it on manually.

2) The interface of the mipi lcd screen on the development board is shown in the figure below

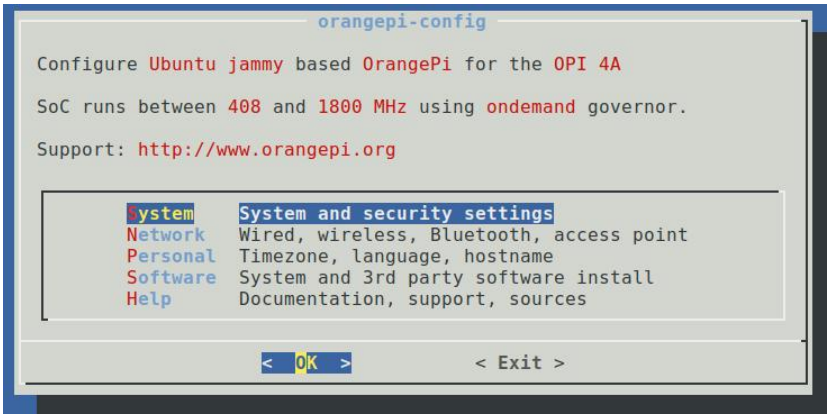


3) The method to open mipi lcd configuration is as follows:

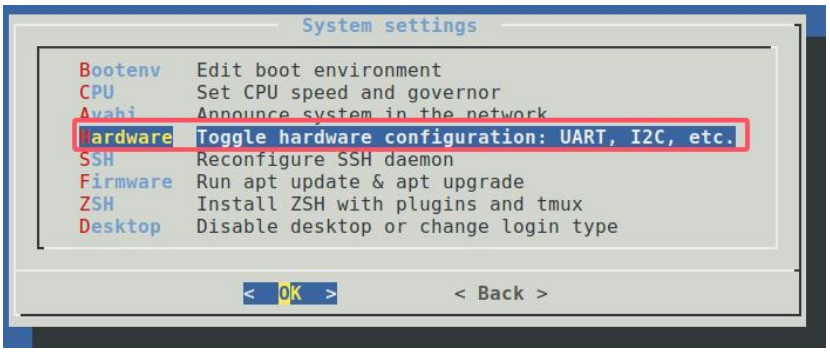
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

b. Then select **System**



c. Then select **Hardware**

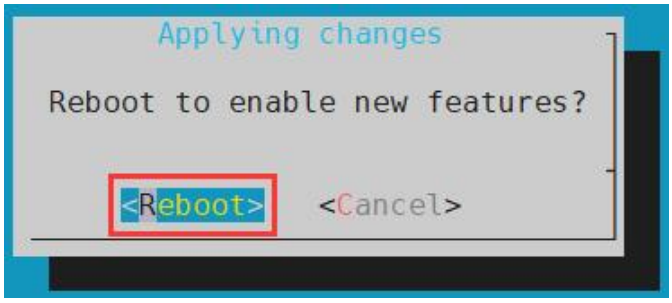


d. Then use the directional keys on the keyboard to locate the **lcd**, and use a **space** to select it

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

f. Then select**<Back>**

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



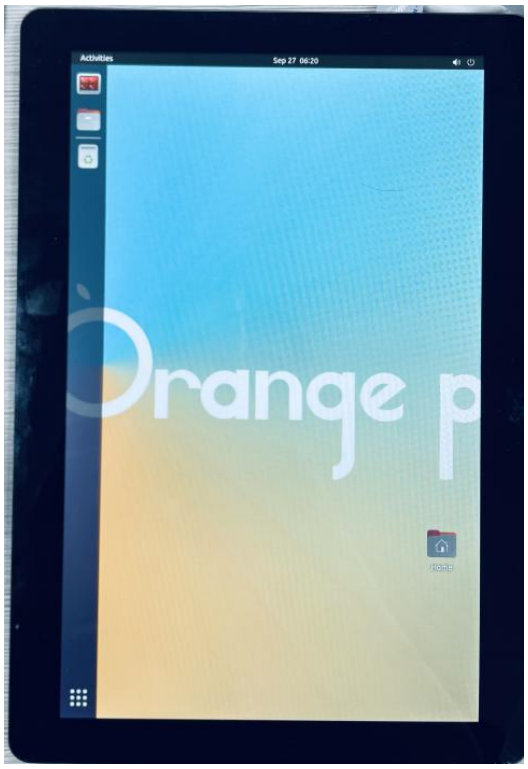
The above settings will eventually add overlays=lcd in /boot/orangepiEnv.txt. After setting it up, you can check it first. If this line of configuration does not exist, then there is a problem with the settings.

If you find using orangepi-config troublesome, you can also use the Vim editor to open /boot/orangepiEnv.txt and add the "overlays=lcd" configuration line.

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "lcd"**
overlays=lcd #Example configuration

4) **Then restart the Linux system**

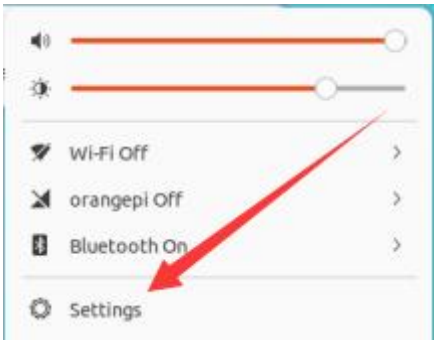
5) After restarting, you can see the display of the LCD screen as shown below (the default is vertical screen)：



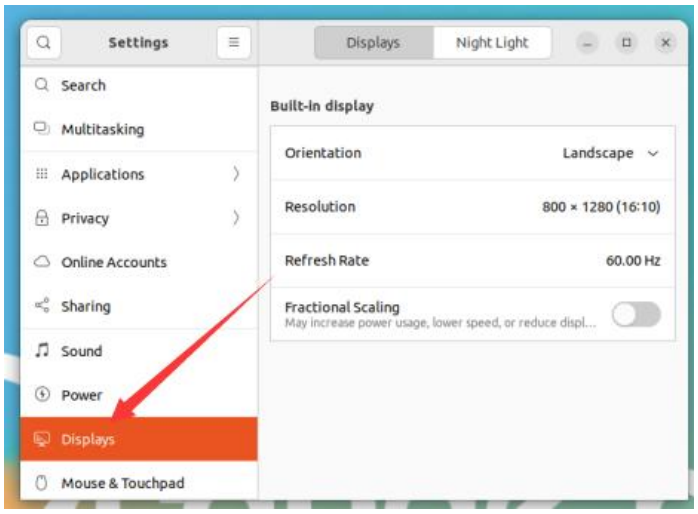
3. 28. 3. Methods for rotating display and touch directions

1) First click on the area in the upper right corner of the desktop

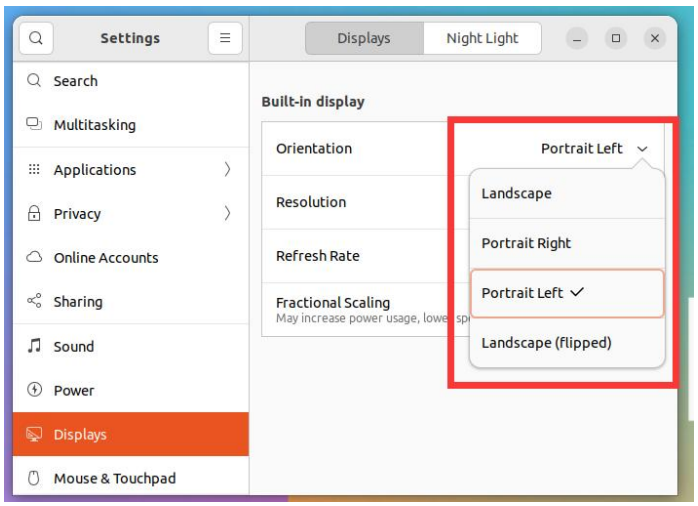
2) Then open Settings



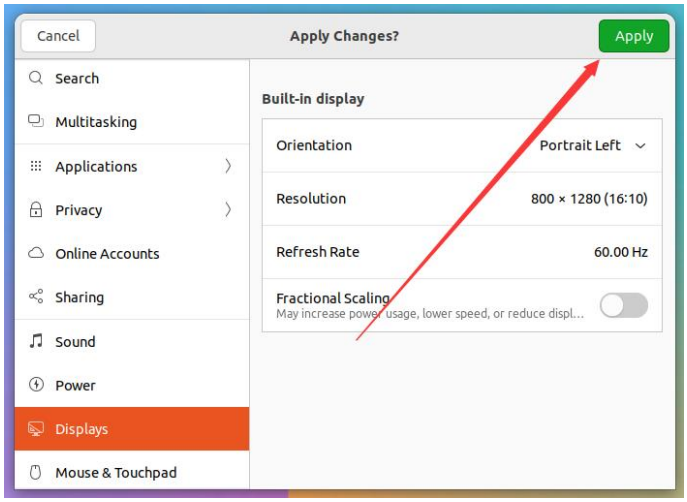
3) Then select**Displays**



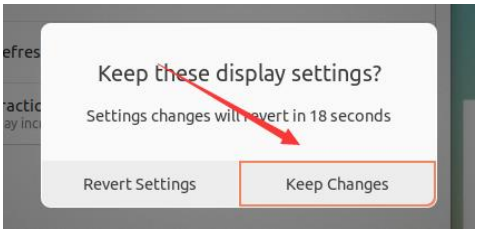
4) Then select the direction you want to rotate in **Orientation** of **Displays**



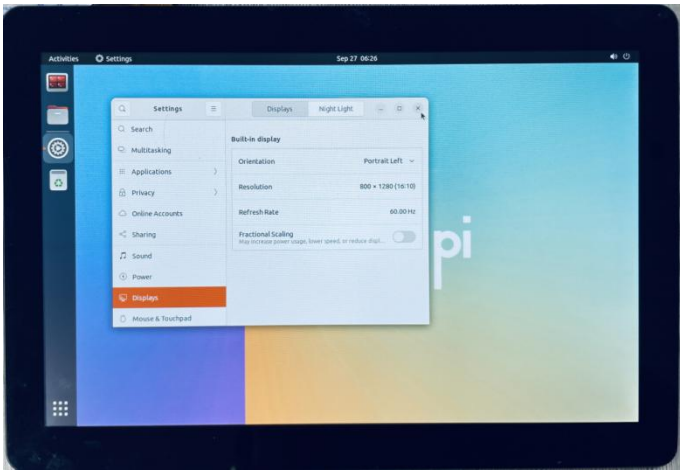
5) Then select **Apply**



6) Then you can see that the screen has been rotated. At this time, you need to select **Keep Changes** to finalize the rotation.



7) The LCD screen will display the following after rotating 90 degrees:



8) **The touch function of the Linux system LCD screen will rotate with the rotation of the display direction, without any other settings**

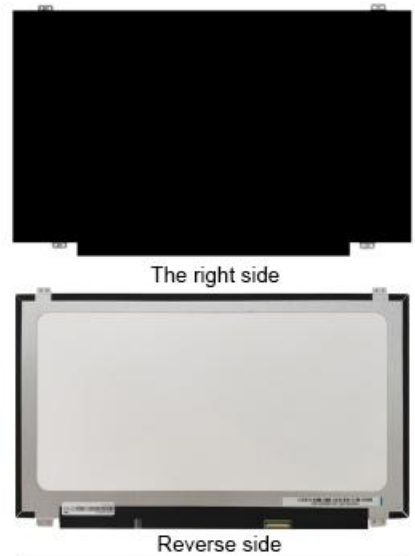
3. 29. How to use the eDP screen

3. 29. 1. Assembly method of eDP screen

1) Currently only a 15.6-inch eDP screen is compatible, and the accessories included are as follows：

a. 0.5 pitch 30pin single head same direction cable

b. 15.6-inch eDP display, resolution 1920x1080.



2) Connect the FPC end of the 30-pin single-head unidirectional cable to the eDP interface of the development board, and the other end to the eDP interface of the screen.

3. 29. 2. How to open eDP screen configuration

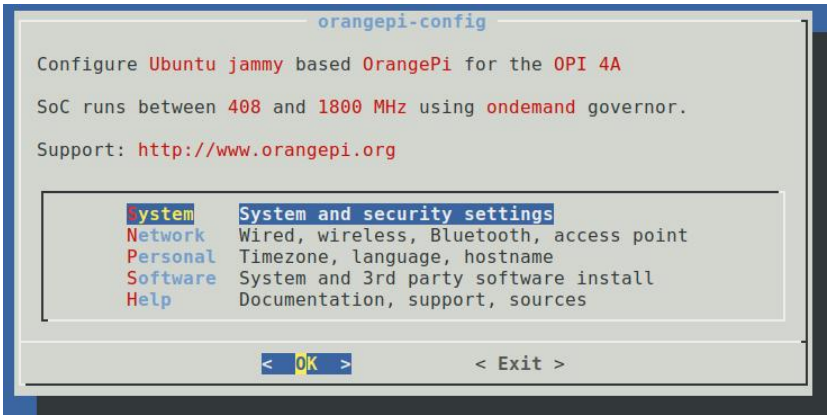
Please note that the method described below is only applicable to adapted eDP screens. If the customer is using an unadapted screen, it will not light up according to the method below.

1) Linux images are not configured with eDP screens enabled by default. If you need to use eDP screens, you need to manually enable them.

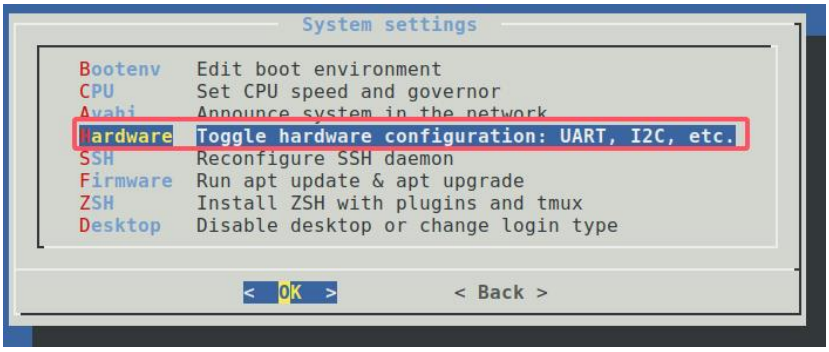
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

b. Then select **System**



c. Then select **Hardware**

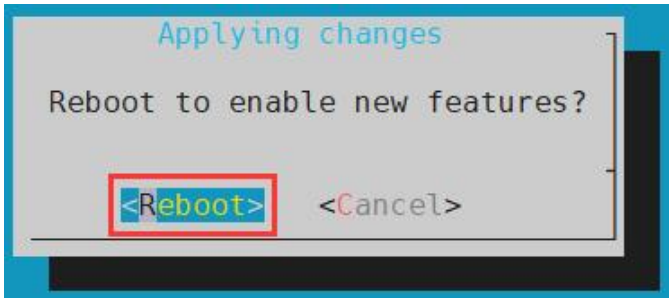


d. Then use the directional keys on the keyboard to locate **edp**, and use a **space** to select it

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

f. Then select**<Back>**

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



The above settings will eventually add overlays=edp in /boot/orangepiEnv.txt. After setting it up, you can check it first. If this line of configuration does not exist, then there is a problem with the settings.

If you find using orangepi-config troublesome, you can also use the Vim editor to open /boot/orangepiEnv.txt and add the "overlays=edp" configuration line.

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "edp"**
overlays=edp #Example configuration

2) After startup, you can see the eDP screen display as shown below：

3. 30. Test of some programming languages supported by Linux system

3. 30. 1. Debian Bookworm System

1) Debian Bookworm is installed with the gcc compilation toolchain by default, which can compile C language programs directly in the Linux system of the development board

a. gcc version 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 Python 3 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 key to exit Python's interactive mode.

b. Write the **hello_world.py** program in Python

```
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 runtime 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 Java version

```
orangepi@orangepi:~$ java --version
```

c. Write a 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. 30. 2. Ubuntu Jammy System

1) Ubuntu Jammy is installed with the gcc compilation tool chain by default, which can compile C language programs directly in the Linux system of the development board

a. gcc version is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
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 Python 3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.12 (main, Jul 29 2024, 16:56:48) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Use the Ctrl+D shortcut key to exit Python's interactive mode.

b. Write the **hello_world.py** program in Python

```
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 runtime 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 Java version

```
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 a **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!
```

3.31. How to upload files to the Linux system of the development board

3.31.1. How to upload files from Ubuntu PC to the Linux system of the development board

3.31.1.1. 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：** The user name of the development board's Linux system can also be replaced with other names, 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 uses for scp. Please use the following command to view the man page

```
test@test:~$ man scp
```

3.31.1.2. How to upload files using FileZilla

1) First install filezilla in your Ubuntu PC

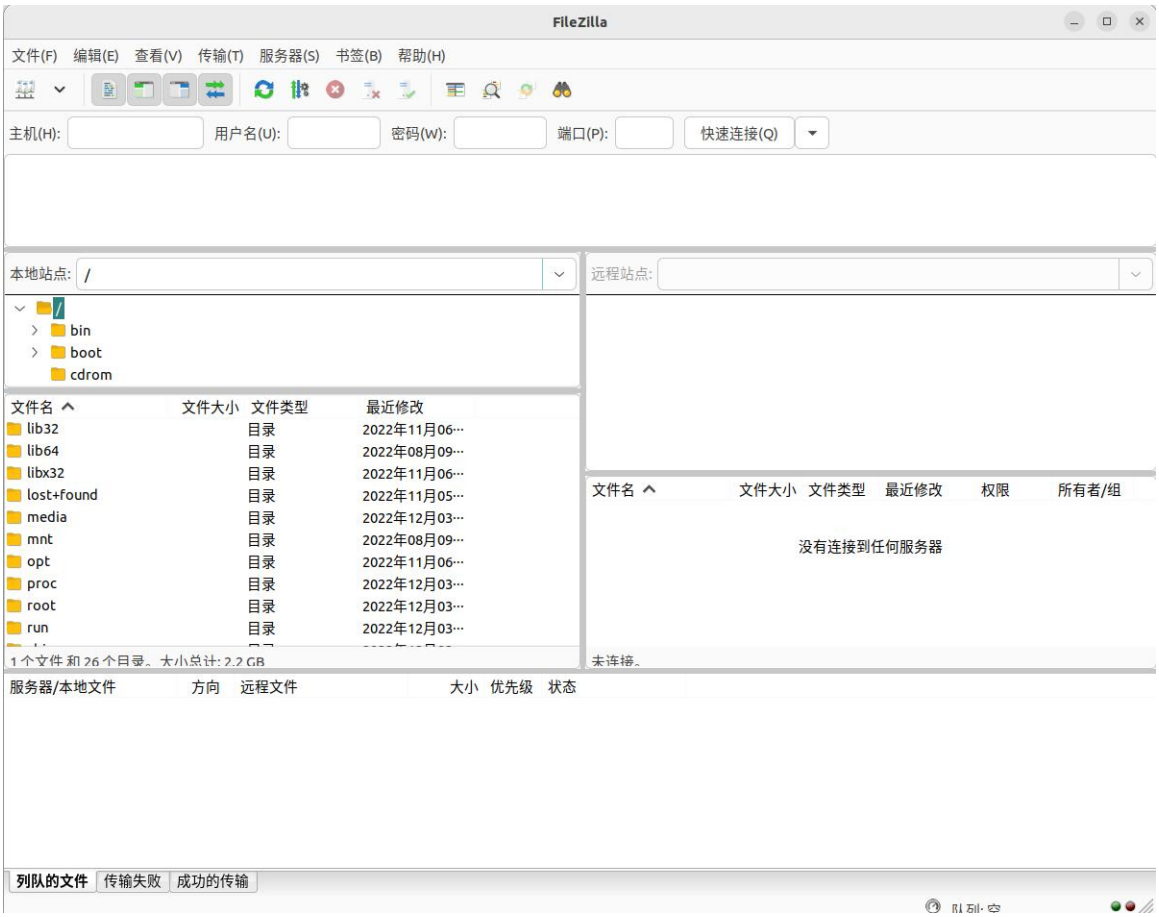
```
test@test:~$ sudo apt install -y filezilla
```

2) Then open filezilla using the following command

```
test@test:~$ filezilla
```

3) The interface after opening filezilla is as follows. At this time, the remote site on the

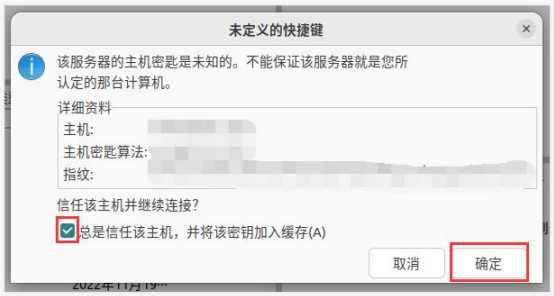
right is empty.



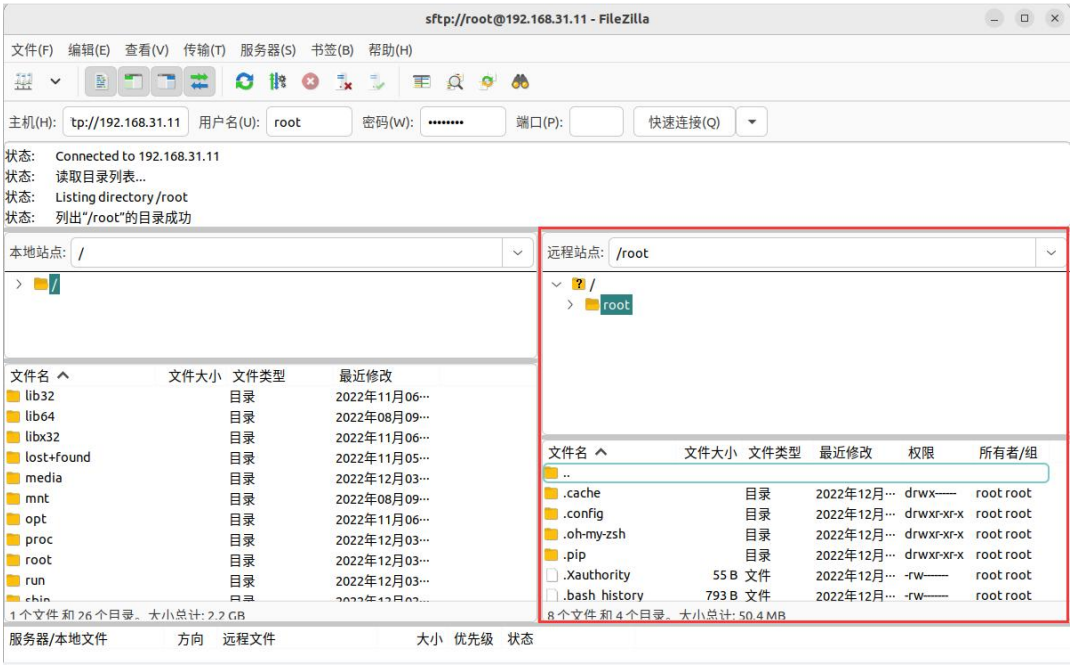
4) The method of connecting the development board is shown in the figure below

5) Then select **Save Password** and click **OK**

6) Then select Always **trust this host** and click **OK**

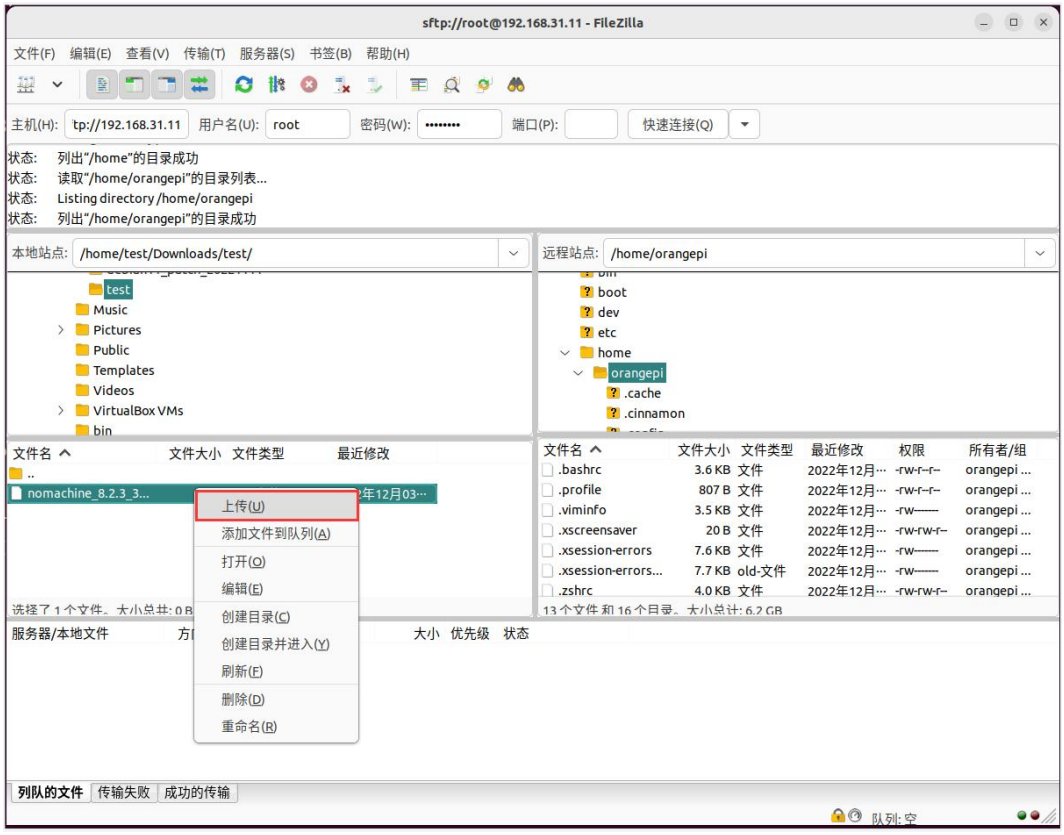


7) After the connection is successful, you can see the directory structure of the development board's Linux file system on the right side of the filezilla software



8) Then select the path to upload to the development board on the right side of the

filezilla software, select the file to be uploaded in the Ubuntu PC on the left side of the filezilla software, right-click the mouse, and then click the upload option to start uploading the file to the development board.



9) After uploading, you can check the uploaded files in the corresponding path of the development board Linux system.

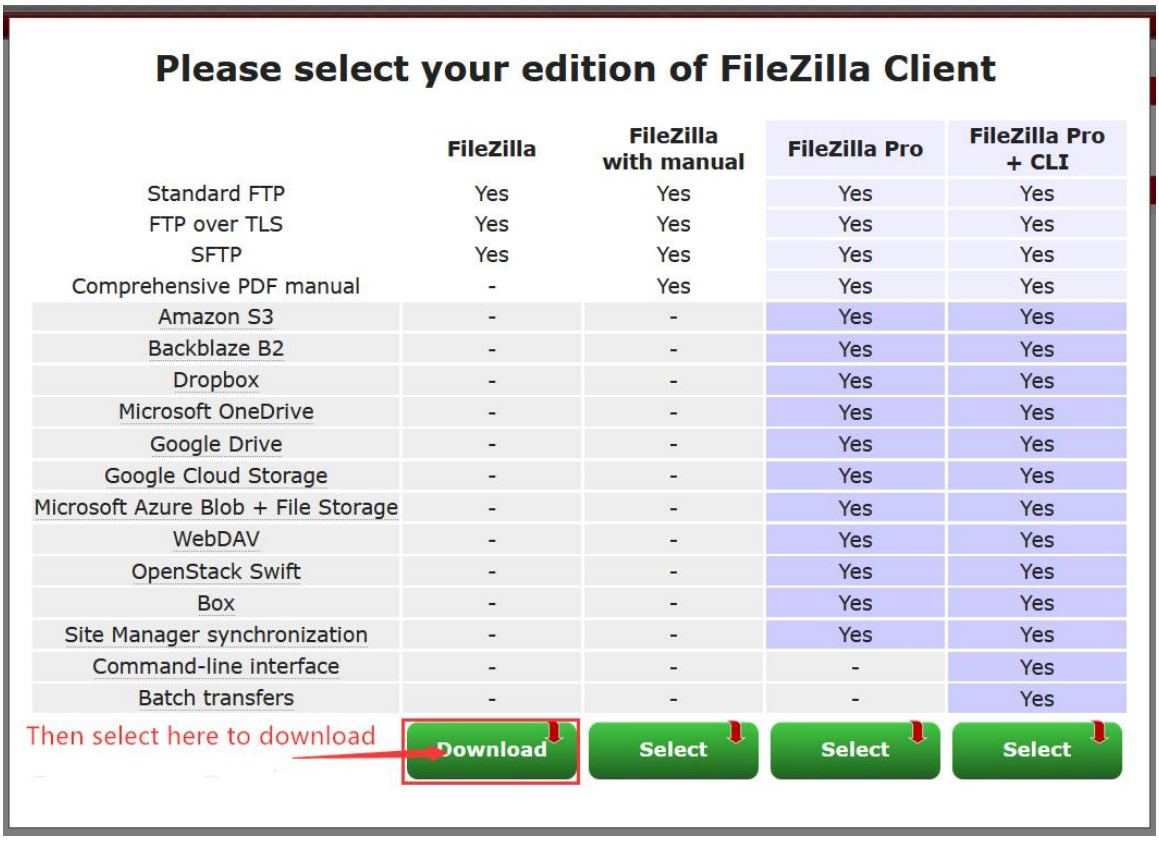
10) The method of uploading a folder is the same as the method of uploading a file, so I will not go into details here.

3.31.2. How to upload files from Windows PC to the Linux system of the development board

3.31.2.1. 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

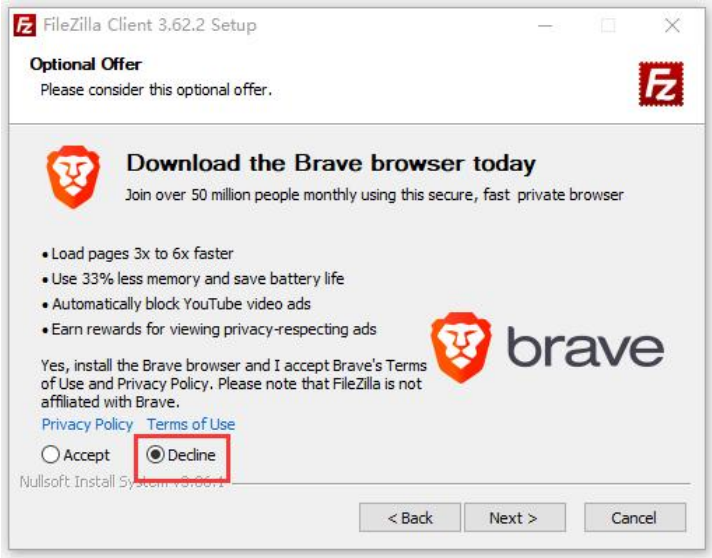
https://filezilla-project.org/download.php?type=client



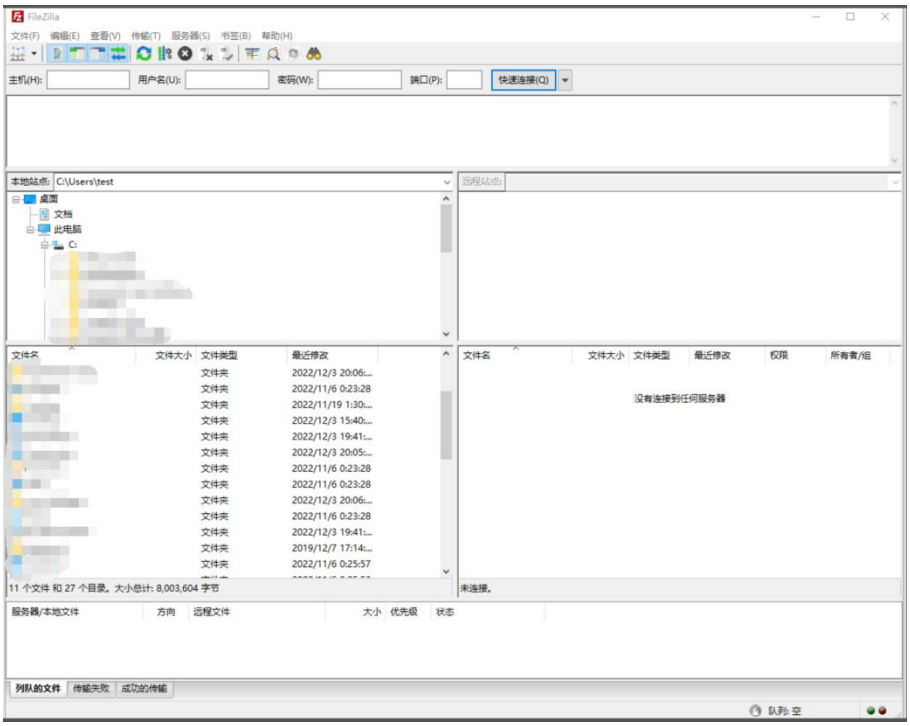
2) The downloaded installation package is as follows, then double-click to install directly

FileZilla_Server_1.5.1_win64-setup.exe

During the installation process, select **Decline** on the following installation interface, then select **Next>**

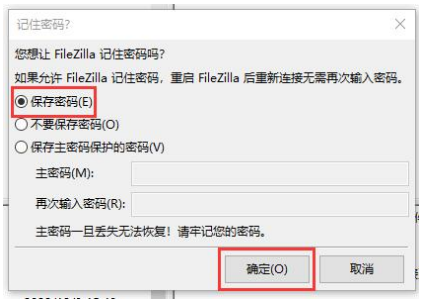


3) The interface after opening filezilla is as follows. At this time, the remote site on the right is empty.

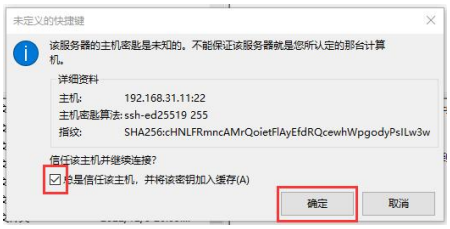


4) The method of connecting the development board is shown in the figure below:

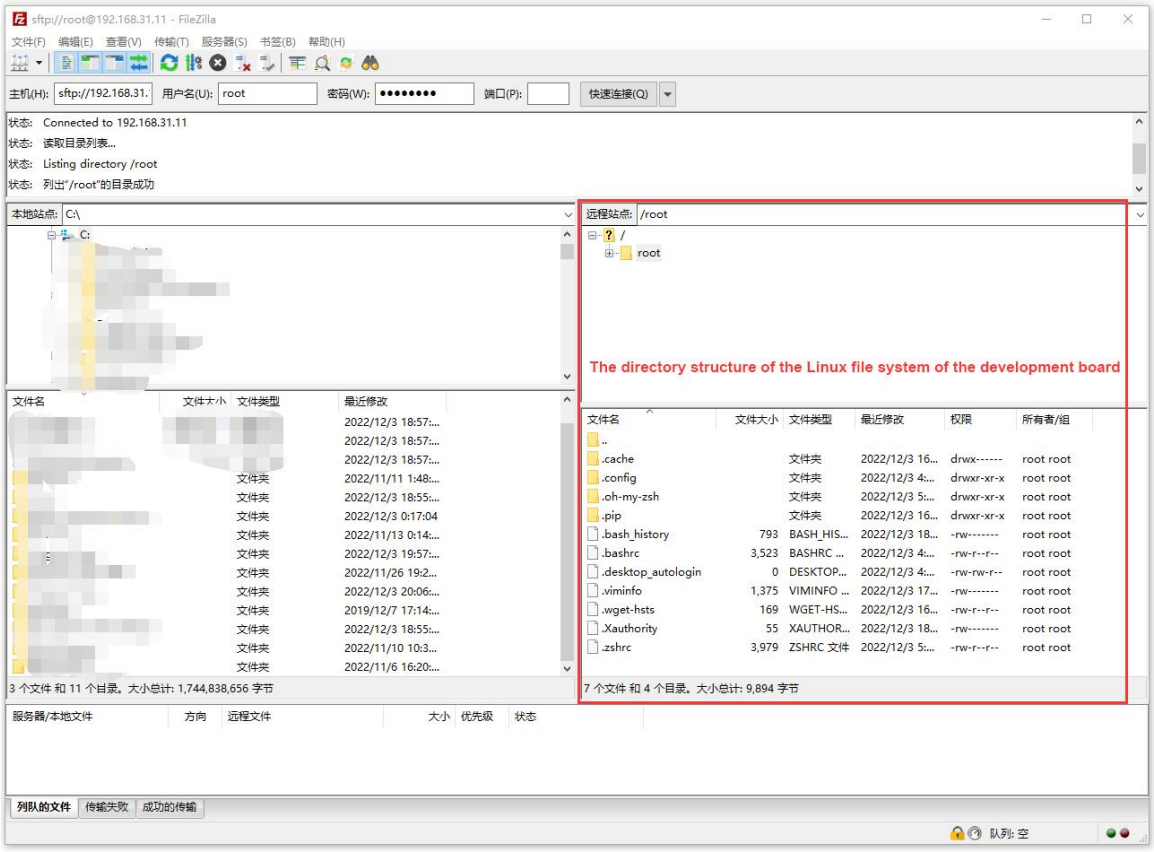
5) Then select **Save Password** and click **OK**



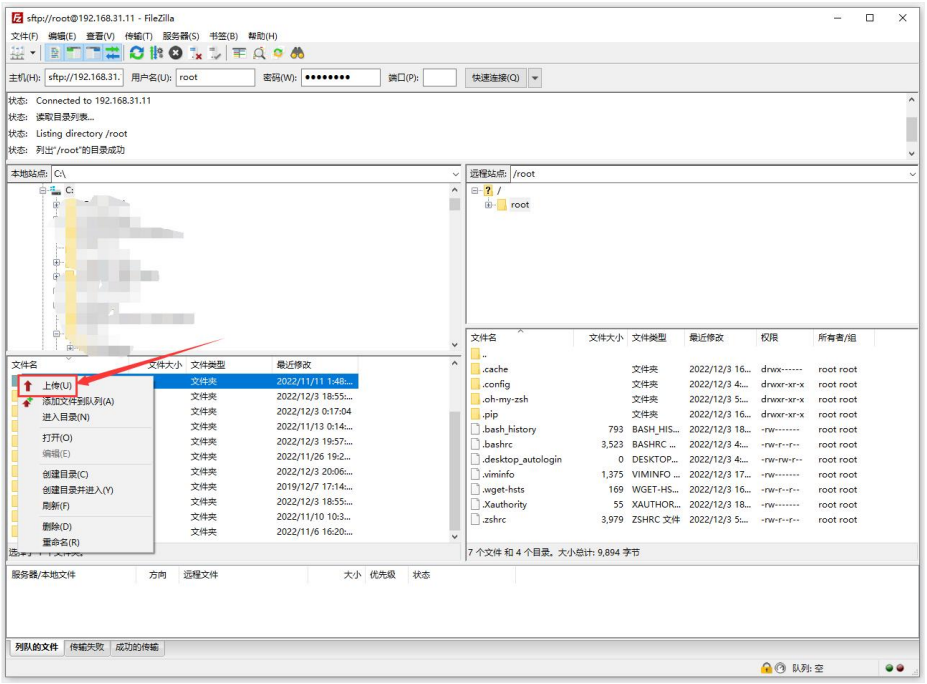
6) Then select **Always trust this host** and click **OK**



7) After the connection is successful, you can see the directory structure of the development board's 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, then select the file to be uploaded in the Windows PC on the left side of the filezilla software, then right-click the mouse, and then click the upload option to start uploading the file to the development board



9) After uploading, you can check the uploaded files in the corresponding path of the development board Linux system.

10) The method of uploading a folder is the same as the method of uploading a file, so I will not go into details here.

3. 32. Instructions for use of NPU

3. 32. 1. Board environment preparation

1) First, you need to install opencv and cmake on the development board.

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install libopencv-dev cmake
```

2) Then download the compressed package of the NPU sample program from the official Baidu Cloud.

3) Then upload the NPU sample program compressed package to the development board and decompress it.

```
orangepi@orangepi:~$ tar -xvf board-demo.tar.gz
```

4) After the compressed package is decompressed, the directory structure is as follows.

```
orangepi@orangepi:~$ cd board-demo/
orangepi@orangepi:~/board-demo$ ls
chineseocr  common  head_pose  lenet  libawnn_viplite  libawutils
mobilenet_v2_ssd_demo  struct2depth  struct2depth  yolov5
```

5) Then you need to copy the NPU library to the system's /usr/lib directory, which is needed to run the sample program below.

```
orangepi@orangepi:~/board-demo$ sudo cp ./common/lib_linux_aarch64/T527/*.so /usr/lib
```

3.32.2. Board Example Run

3.32.2.1. Run the mobilenet_v2_ssd target detection example

1）First compile the mobilenet_v2_ssd target detection example

```
orangepi@orangepi:~/board-demo$ cd mobilenet_v2_ssd_demo
orangepi@orangepi:~/board-demo/mobilenet_v2_ssd_demo$ mkdir build
orangepi@orangepi:~/board-demo/mobilenet_v2_ssd_demo$ cd build
orangepi@orangepi:~/board-demo/mobilenet_v2_ssd_demo/build$ cmake ..
orangepi@orangepi:~/board-demo/mobilenet_v2_ssd_demo/build$ make
```

2）Then execute the following command to run the example.

```
orangepi@orangepi:/root/board-demo/mobilenet_v2_ssd_demo/build$ ./mbv2-ssd-demo -b ../model/mbv2_ssd_x527.nb -i ../000012.jpg
```

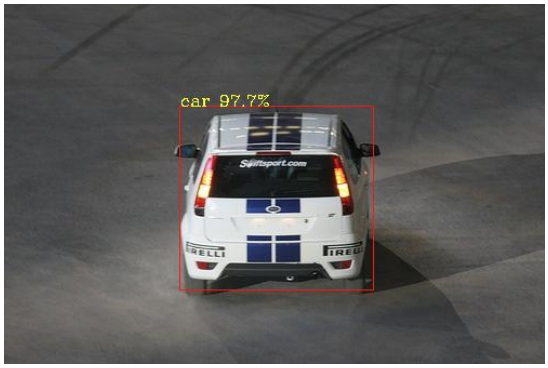
3）The output information of normal program operation is as follows.

```
orangepi@orangepi:/root/board-demo/mobilenet_v2_ssd_demo/build$ ./mbv2-ssd-demo -b ../model/mbv2_ssd_x527.nb -i ../000012.jpg
.....................
get output finished.
mbv2_ssd_postprocess.cpp run.
car = 0.97705 at 162 94 179 x 170
```

```
postprocess time : 0.025 Sec
destory npu finished.
~NpuUint.
```

4）The output will be saved as ssd_out.png in the current directory

```
orangepi@orangepi4a:~/board-demo/mobilenet_v2_ssd_demo/build$ ls ssd_out.png
ssd_out.png
```



3. 32. 2. 2. Run the yolov5 object detection example

1）First compile the yolov5 example.

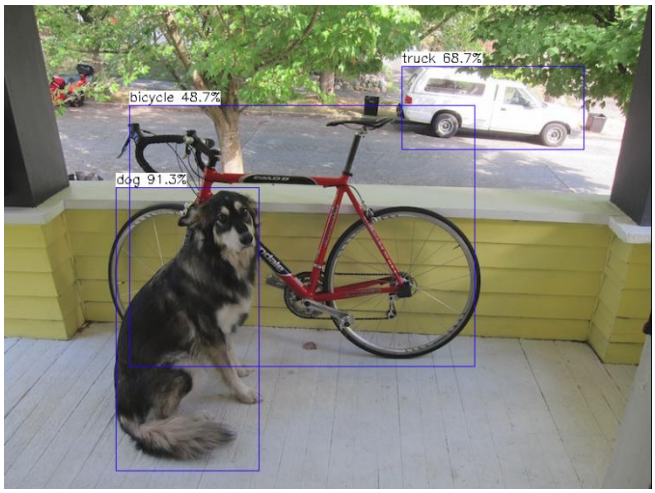
```
orangepi@orangepi:~/board-demo$ cd yolov5
orangepi@orangepi:~/board-demo/yolov5$ mkdir build
orangepi@orangepi:~/board-demo/yolov5$ cd build
orangepi@orangepi:~/board-demo/yolov5/build$ cmake ..
orangepi@orangepi:~/board-demo/yolov5/build$ make
```

2）Then execute the following command to run the example.

```
orangepi@orangepi:~/board-demo/yolov5/build$ ./yolov5 ../model/v2/yolov5.nb ../input_data/dog.jpg
./yolov5 nbg input
VIPLite driver software version 1.13.0.0-AW-2023-10-19
yolov5_preprocess.cpp run.
yolov5_postprocess.cpp run.
detection num: 3
16:  91%, [ 135,  218,  305,  553], dog
 7:  69%, [ 473,   75,  689,  174], truck
 1:  49%, [ 151,  121,  561,  431], bicycle
```

3）The output will be saved as result.png in the current directory

```
orangepi@orangepi4a:~/board-demo/yolov5/build$ ls result.png
result.png
```



3. 32. 2. 3. Run the head_pose human posture recognition example

1）First compile the head_pose example.

```
orangepi@orangepi:~/board-demo$ cd head_pose
orangepi@orangepi:~/board-demo/head_pose$ mkdir build
orangepi@orangepi:~/board-demo/head_pose$ cd build
orangepi@orangepi:~/board-demo/head_pose/build$ cmake ..
orangepi@orangepi:~/board-demo/head_pose/build$ make
```

2）Then execute the following command to run the example.

```
orangepi@orangepi:~/board-demo/head_pose/build$ ./head_pose -b1 ../model/rfb_landm_face_320_320_sim_x527.nb
-b2 ../model/head_pose_x527.nb -i ../input_data/000438.jpg
```

3）The output result will be saved to head_pose_result.jpg in the current directory.

```
orangepi@orangepi:~/board-demo/head_pose/build$ ls head_pose_result.jpg
head_pose_result.jpg
```

3. 32. 2. 4. Run the resnet50 image classification example

1）First compile the resnet50 example.

```
orangepi@orangepi:~/board-demo$ cd resnet50
orangepi@orangepi:~/board-demo/resnet50$ mkdir build
orangepi@orangepi:~/board-demo/resnet50$ cd build
orangepi@orangepi:~/board-demo/resnet50/build$ cmake ..
orangepi@orangepi:~/board-demo/resnet50/build$ make
```

2）Then execute the following command to run the example.

```
orangepi@orangepi:~/board-demo/resnet50/build$ ./resnet50 ../model/v2/resnet50.nb ../input_data/dog_224_224.jpg
```

3）The output result information is as follows, which outputs the top 5 predicted by the model, among which the most likely category is collie

```
orangepi@orangepi:~/board-demo/resnet50/build$ ./resnet50 ../model/v2/resnet50.nb ../input_data/dog_224_224.jpg
...
========== top5 ==========
class id: 231, prob: 15.432617, label: collie
class id: 230, prob: 13.103271, label: Shetland sheepdog, Shetland sheep dog, Shetland
class id: 169, prob: 12.617920, label: borzoi, Russian wolfhound
class id: 224, prob: 12.423828, label: groenendael
class id: 160, prob: 10.191406, label: Afghan hound, Afghan
class_postprocess success.
```

3. 32. 2. 5. Run struct2depth depth detection example

1）First compile the struct2depth example.

```
orangepi@orangepi:~/board-demo$ cd struct2depth
orangepi@orangepi:~/board-demo/struct2depth$ mkdir build
orangepi@orangepi:~/board-demo/struct2depth$ cd build
orangepi@orangepi:~/board-demo/struct2depth/build$ cmake ..
orangepi@orangepi:~/board-demo/struct2depth/build$ make
```

2）Then execute the following command to run the example

```
orangepi@orangepi:~/board-demo/struct2depth/build$ ./struct2depth
-b ../model/v2/struct2depth.nb -i ../input_data/0015.jpg
```

3）The depth information of the model inference will be saved in jpg and txt files.

```
orangepi@orangepi:~/board-demo/struct2depth/build$ ls disp_* output_*
disp_color.jpg    disp_show.jpg    output_1.txt    output_3.txt
disp_gray.jpg     output_0.txt     output_2.txt
```

3. 32. 2. 6. Run ChineseOCR text recognition example

1）First compile the ChineseOCR example.

```
orangepi@orangepi:~/board-demo$ cd chineseocr
orangepi@orangepi:~/board-demo/chineseocr$ mkdir build
orangepi@orangepi:~/board-demo/chineseocr$ cd build
orangepi@orangepi:~/board-demo/chineseocr/build$ cmake ..
orangepi@orangepi:~/board-demo/chineseocr/build$ make
```

2）Then execute the following command to run the example.

```
orangepi@orangepi:~/board-demo/chineseocr/build$ ./chineseocr -d ../model/v2/ -1 dbnet_\
1024 -2 angle_net -3 crnn_lite_lstm_256 -4 keys.txt -i ../input_data/1.jpg
```

3）The output result is as follows, and you can see that the text in the image has been recognized.

```
orangepi@orangepi:~/board-demo/chineseocr/build$ ./chineseocr -d ../model/v2/ -1 dbnet_\
```

```
1024 -2 angle_net -3 crnn_lite_lstm_256 -4 keys.txt -i ../input_data/1.jpg
...
=====End detect=====
FullDetectTime(903.447417ms)
We at Allwinner Technology
AI
chip
take off12345666666 !

run finished.
~CrnnNet.
~AngleNet.
~DbNet.
~NpuUint.
```

3. 33. How to burn Linux image to eMMC

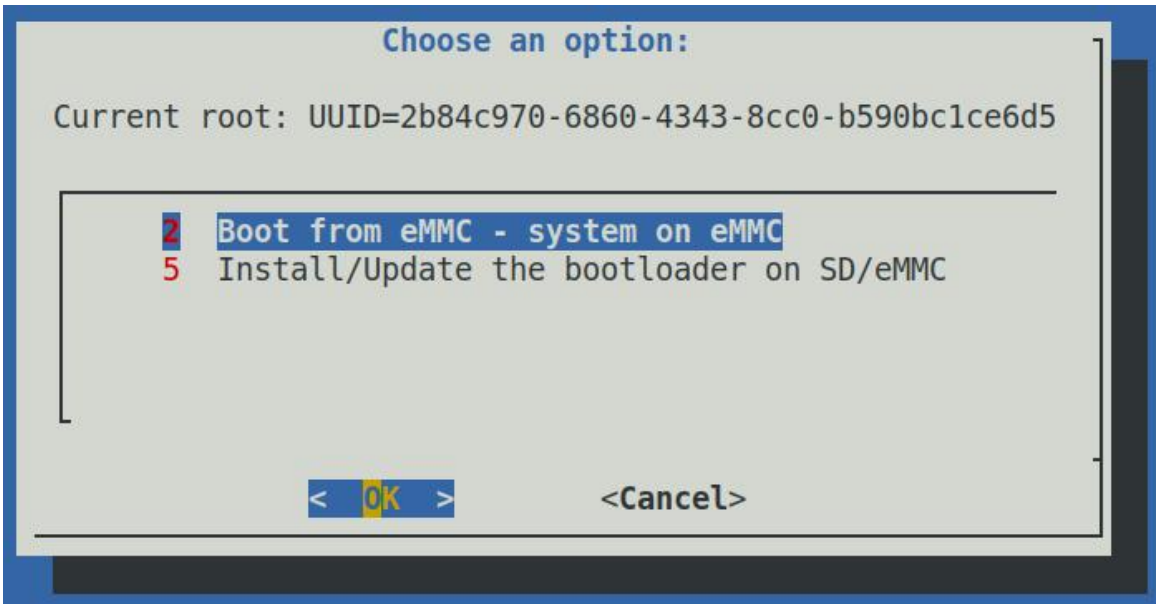
Note that the development board can be started through a TF card or eMMC, and the priority of the TF card is higher than that of the eMMC. In other words, if a TF card is inserted into the development board and there is a system in the TF card, the system in the TF card will be started by default, and the system in the eMMC will not be started.

1) Burning the Linux image to the eMMC requires the use of a TF card. First, burn the Linux image to the TF card, then start the development board and enter the Linux system

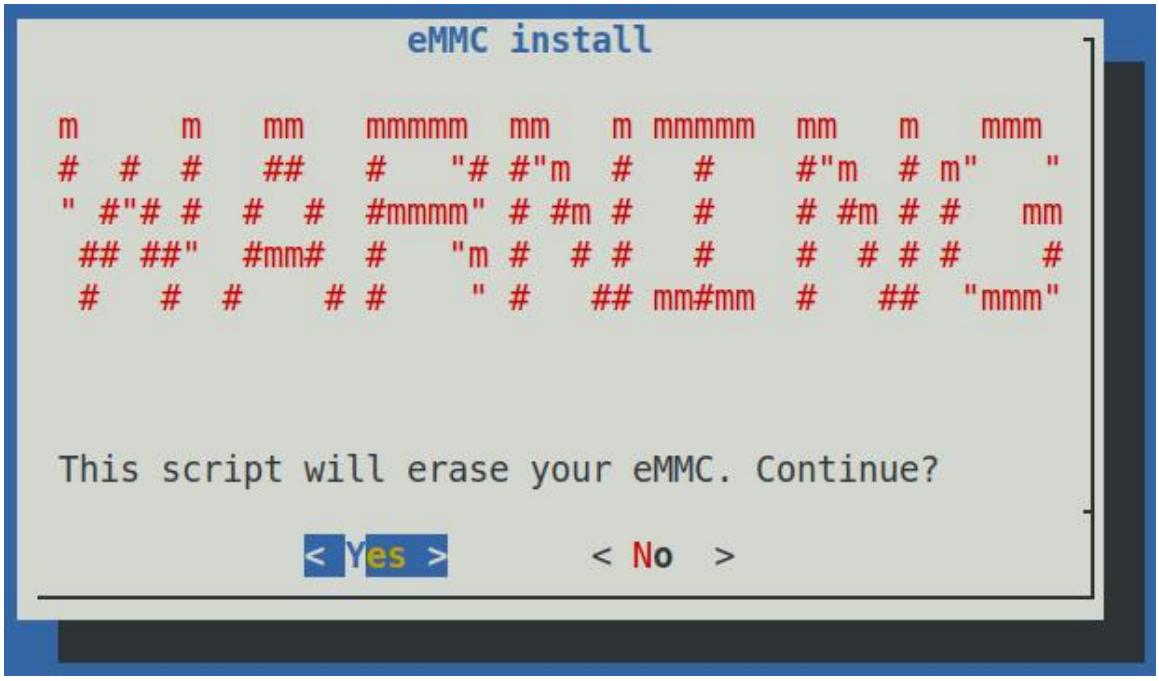
2) Then run the **nand-sata-install** script, remember to **add sudo permissions**.

```
orangepi@orangepi:~$ sudo nand-sata-install
```

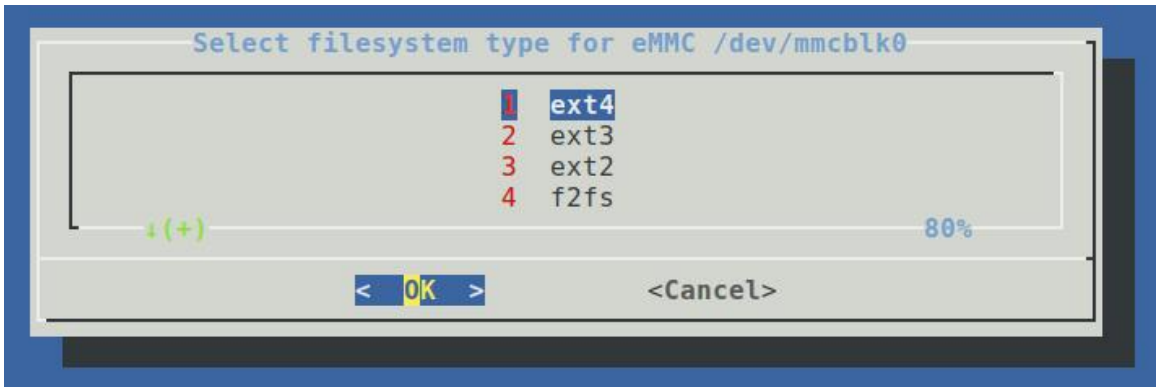
3) Then select **2 Boot from eMMC - system on eMMC**



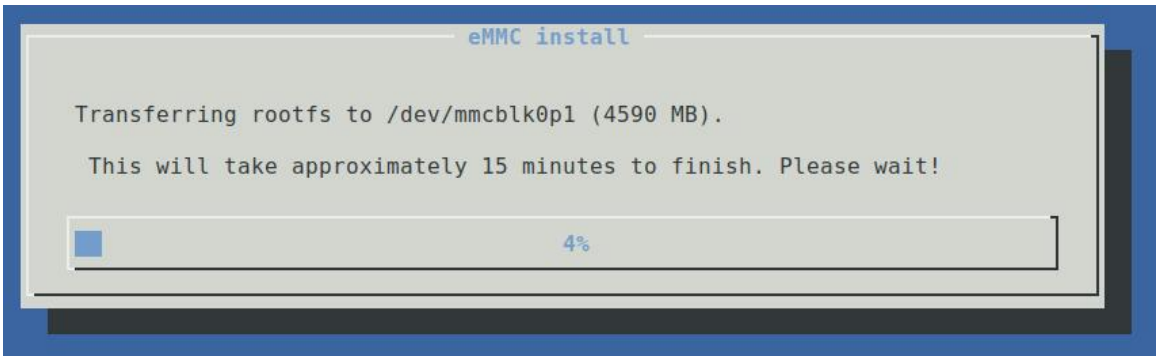
4) Then a warning will pop up, the script will erase all data on the eMMC, select **<Yes>** to continue



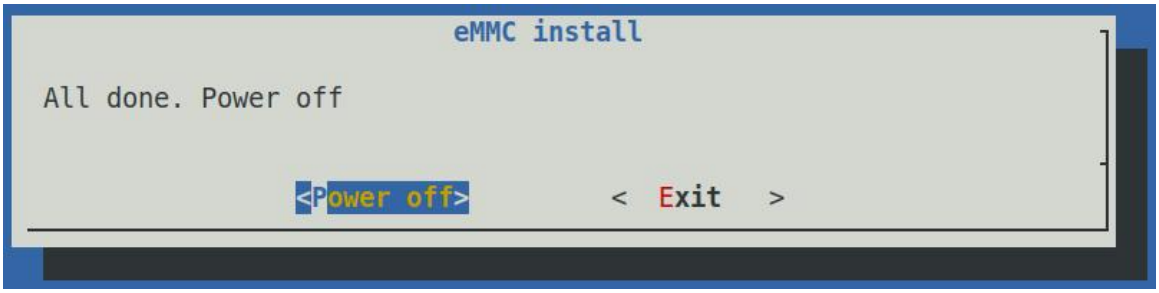
5) You will then be prompted to select the type of file system. Five file systems are supported: ext2/3/4, f2fs, and btrfs.



6) Then it will start to format the eMMC. After formatting the eMMC, it will start to burn the Linux image to the eMMC.



7) After burning, the following options will be prompted. You can select **<Power off>** to shut down directly



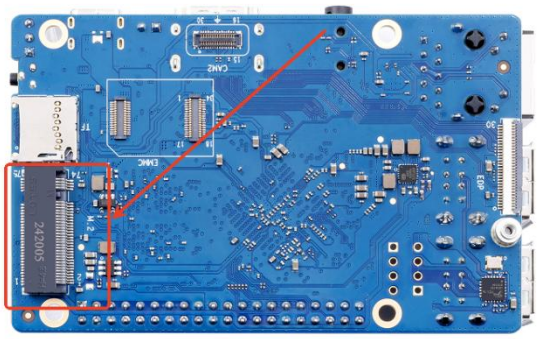
8) Then remove the TF card and power on again, the Linux system in the eMMC will start.

3. 34. Method for burning Linux images to SPIFlash+NVMe SSD

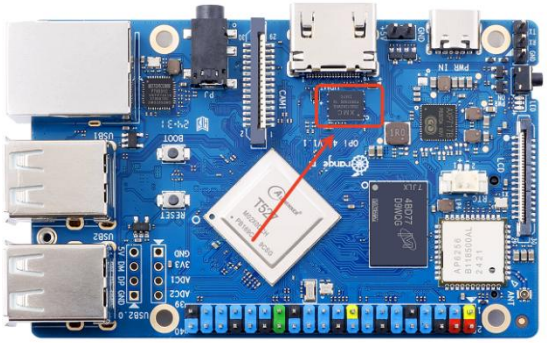
1) Firstly, it is necessary to prepare an NVMe SSD solid state drive with a PCIe interface

specification of PCIe2.0x2 for the M.2 slot on the development board.

2) Then insert the NVMe SSD into the M.2 PCIe interface of the development board and secure it in place.



3) The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting to burn.



4) Burning the image to SPIFlash+NVMe SSD requires the use of a TF card, so the first step is to burn the Linux image onto the TF card, and then use the TF card to boot the development board into the Linux system. The method of burning a Linux image to a TF card can be found in the two sections: **the method of burning a Linux image to a TF card based on Windows PC** and **the method of burning a Linux image to a TF card based on Ubuntu PC**.

5) After starting the linux system with a TF card, you can burn the image to SPI

Flash+NVMe SSD.

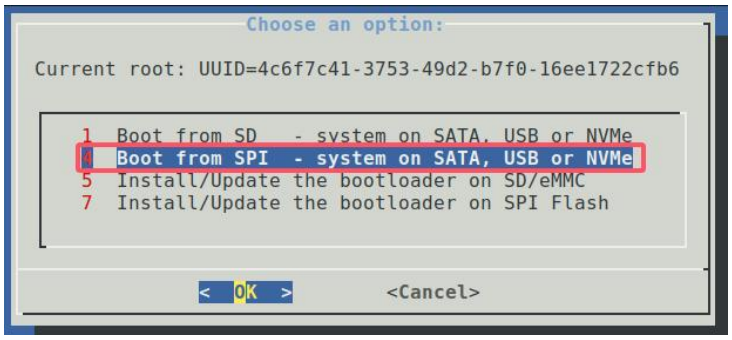
a. First, create a partition for NVMe SSD.

```
orangepi@orangepi:~$ sudo parted --script /dev/nvme0n1 mklabel gpt mkpart primary ext4 8192s 100%
```

b. Then run **nand-sata-install**, **remember to add sudo privileges for regular users**.

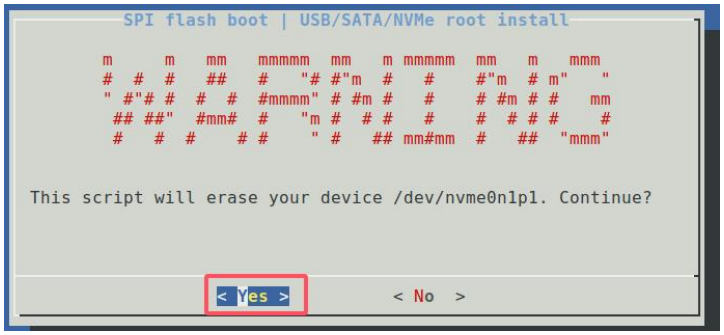
```
orangepi@orangepi:~$ sudo nand-sata-install
```

c. Then select **4 Boot from SPI - system on SATA, USB or NVMe**。

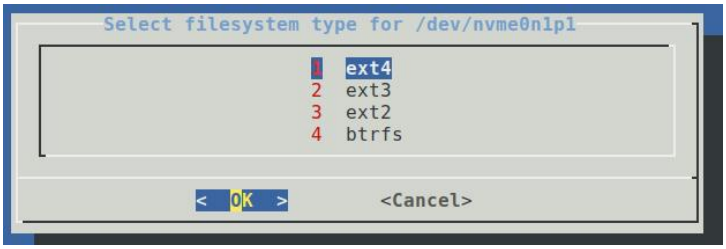


d. Then press enter to confirm

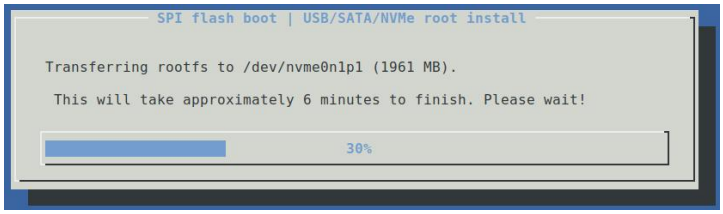
e. Then select **<Yes>**。



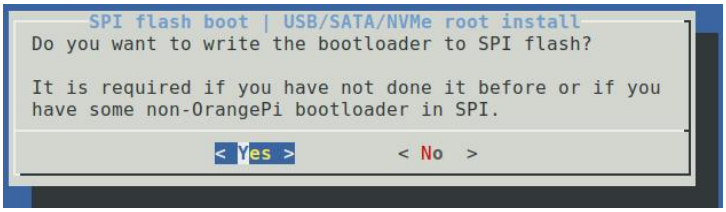
f. Then it will prompt to select the type of file system.

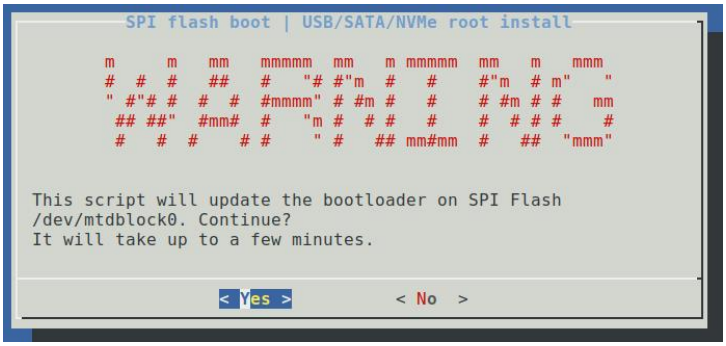


g. Then it will start formatting the NVMe SSD, and after formatting is complete, it will start burning the system into the NVMe SSD.



h. Then please be patient and wait for the burning to complete. After burning, you will be prompted whether to burn the bootloader to SPI Flash, and then select**<Yes>**.





i. After burning, the following options will be prompted, you can choose**<Power off>**to shut down directly

3. 35. Instructions for using the system backup script opi-bkimg

opi-bkimg is a system backup script designed for the Orange Pi series development board, which can help users fully backup their currently running Linux system to an image file for subsequent recovery or migration to other devices.

Before starting the backup, please ensure that there is enough space left in the

system, otherwise the backup may fail.

1) Run the **opi-bkimg** script to start backing up the system.

```
orangepi@orangepi:~$ sudo opi-bkimg
```

2) If there is no **opi-bkimg** script in the system, you can download it from the following repository by yourself.

https://gitee.com/orangepi-xunlong/opi-bkimg

3) The script will check the root file system size and create a properly sized image file, then use the **rsync** command to copy system data to the image file. The specific process is as follows.

```
orangepi@orangepi:~$ sudo opi-bkimg
[ o.k. ] Starting backup to image [ /mnt/backup.img ]
[ o.k. ] Current rootfs size [ 2561 MiB ]
[ o.k. ] Creating blank image for rootfs [ 3500 MiB ]
[ .... ] dd: 3.42GiB [ 126MiB/s] [=========================================>] 100%
[ o.k. ] Creating partitions [ root: ext4 ]
[ .... ] Creating rootfs [ ext4 on /dev/loop1p1 ]
[ .... ] Copying file to [ / ] [################################] 100%
[ .... ] Re-enabling [ orangepi-resize-filesystem ]
[ o.k. ] Writing U-boot bootloader [ /dev/loop1 ]
[ o.k. ] Unmounting [ /mnt/opi-bkimg.iaioMi/rootfs ]
[ o.k. ] Backup completed [ /mnt/backup.img ]
orangepi@orangepi:~$
```

4) When the script displays the message '**Backup completed**', it indicates that the backup has ended. The image generation path is shown below,

```
orangepi@orangepi:~$ ls /mnt/
backup.img
```

5) **backup.img** is the image file that can be burned.

3. 36. Usage of Linux Overlayoot

In some user scenarios, the system often shuts down by directly cutting off power instead of executing the poweroff command, which may cause file system damage and prevent normal startup. To prevent such issues, it is recommended to enable the overlay root feature and mount the root file system as read-only. This ensures that even in the event of an abnormal power outage, the original file system

is not affected, thereby improving the stability and reliability of the system.

This feature has been tested on Ubuntu Jammy, Ubuntu Focal, Ubuntu Noble, and Debian Bookworm, but it is currently not supported on Debian Bullseye.

Overlayboot is a technology based on Overlayfs that can mount the Linux root file system (/) as read-only and provide writable storage through an overlay. All user modifications (such as file writes, configuration changes, etc.) will be redirected to the writable layer, while the original root file system remains unchanged, making it suitable for scenarios such as embedded devices and servers that require high system stability.

Overlayboot consists of two layers: a read-only layer (lowerdir) and a writable layer (upperdir). The read-only layer is the original root file system (/), mounted in read-only mode; The writable layer is used to store user modified data, which is merged with the read-only layer to form the final root file system view, making users unaware of the hierarchical structure of the underlying layer during operation.

The writable layer can be stored in tmpfs or block devices. When using tmpfs, all modifications are stored in memory and lost after restart, and the system returns to its original state; When using block devices such as USB drives and TF card partitions, data can be stored persistently and will not be lost after restarting.

3.36.1. Configure Overlayboot to tmpfs mode

In this mode, all modifications made by the system will be stored in memory (tmpfs). After restarting, all changes will be lost and the system will return to its initial state.

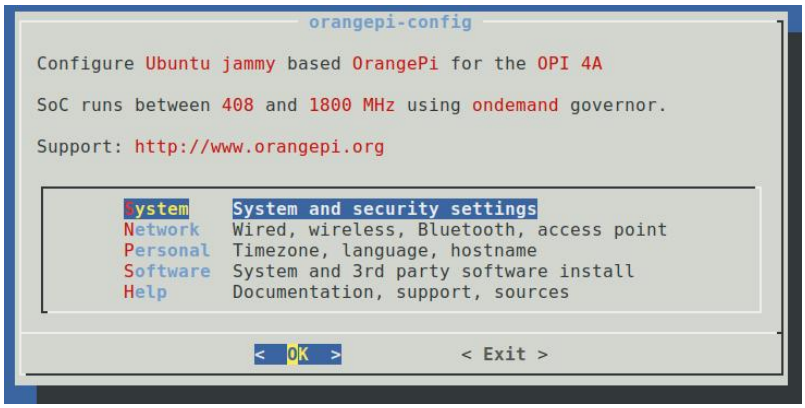
1) Overlayboot is turned off by default in Linux systems and needs to be manually turned on to use. The opening steps are as follows:

a. Firstly, please ensure that the development board is connected to a wired or wireless network. For specific methods, please refer to the **Network Connection** Testing section

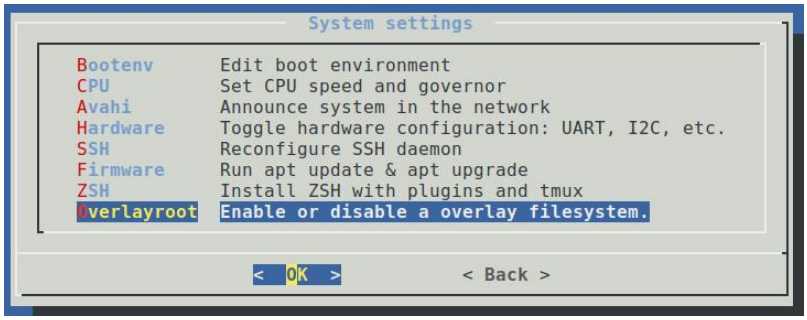
b. Then run **orangepi-config**, remember to add **sudo** privileges for regular users

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

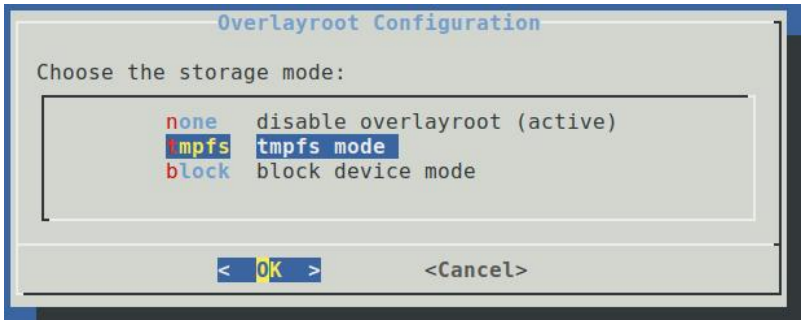
c. Then select **System**



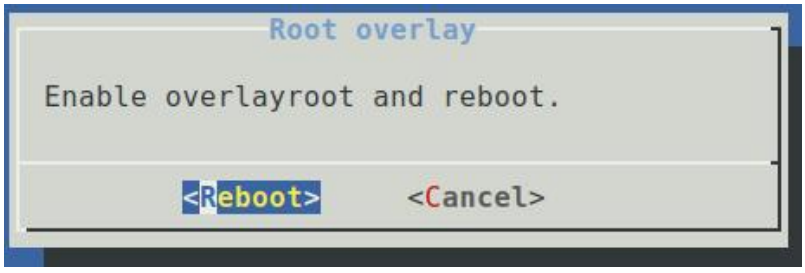
d. Then select **Overlayroot**



e. Then select **tmpfs** and press enter



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



2) Then enter the **mount | grep root** command to see the mounting information of the file system.

```
orangepi@orangepi:~$ mount |grep root
/dev/mmcblk0p1 on /media/root-ro type ext4 (ro,relatime)
tmpfs-root on /media/root-rw type tmpfs (rw,relatime)
overlayroot on / type overlay (rw,relatime,lowerdir=/media/root-ro,upperdir=/media/root-rw/overlay,workdir=/media/root-rw/overlay-workdir/_)
overlayroot on /var/log.hdd type overlay (rw,relatime,lowerdir=/media/root-ro,upperdir=/media/root-rw/overlay,workdir=/media/root-rw/overlay-workdir/_)
orangepi@orangepi:~$
```

From the mounting information, it can be seen that the file system partition /dev/mmcblk0p1 is mounted in read-only mode to /media/root-ro，tmpfs is mounted as a writable layer to /media/root-rw, indicating that the overlay root configuration has taken effect.

Due to tmpfs being a memory file system, all modifications will be lost after restarting.

3. 36. 2. Configuring Overlayboot to block device storage mode

In this mode, the writable layer uses specific storage devices (such as TF card partition /dev/mmcblk0p2), and data will not be lost after restarting.

Please note that after opening Overlayboot, modifications to the/boot partition (such as updating the kernel or modifying /boot/orangepiEnv.txt) will not take effect. The reason is that in block device mode, modifications are actually made to /boot in the overlay layer, while Uboot reads files from the real partition at startup, so the modifications will not take effect. To make the modifications effective, it is necessary to disable Overlayboot before performing these operations.

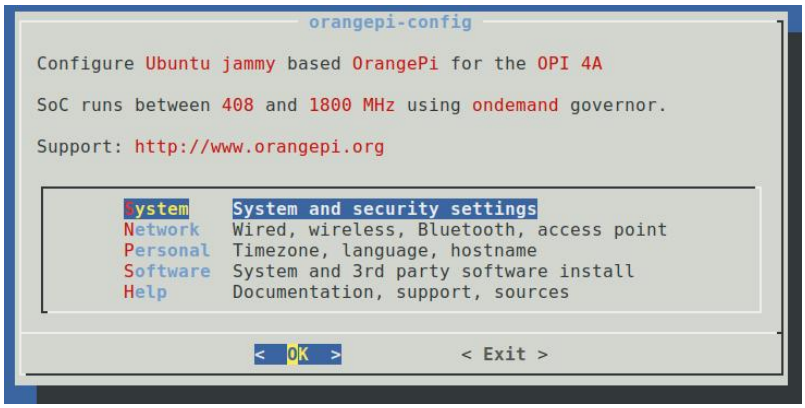
1) Overlayboot is turned off by default in Linux systems and needs to be manually turned on to use. The opening steps are as follows:

a. Firstly, please ensure that the development board is connected to a wired or wireless network. For specific methods, please refer to the **Network Connection** Testing section

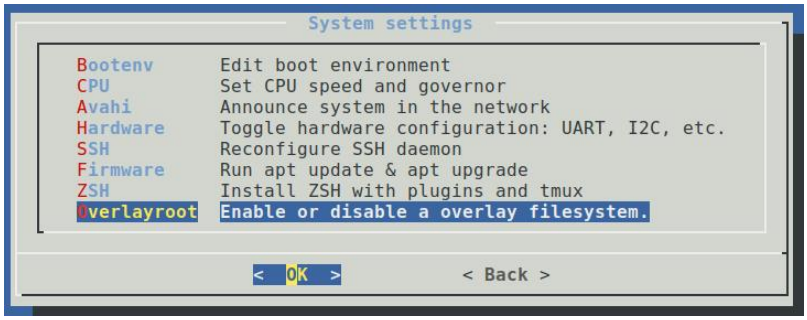
a. Then run **orangepi-config**, remember to add **sudo** privileges for regular users

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

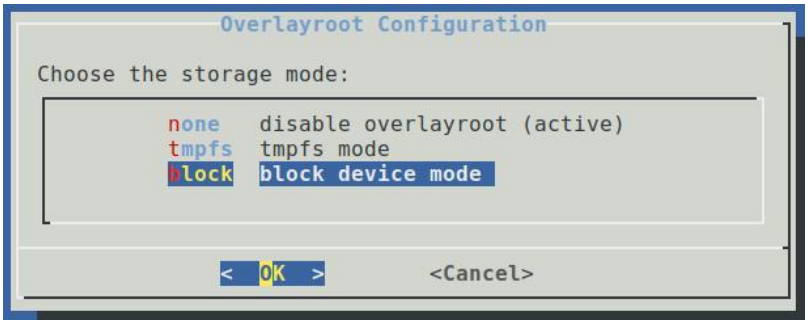
b. Then select **System**



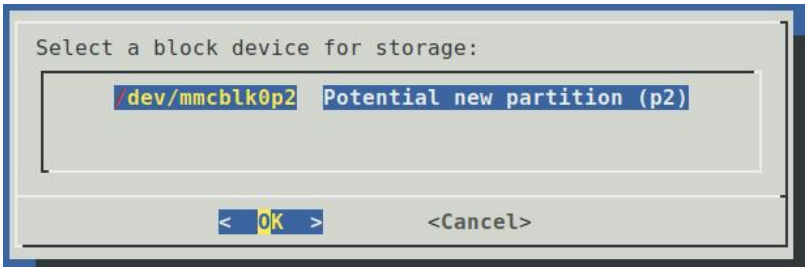
c. Then select **Overlayroot**



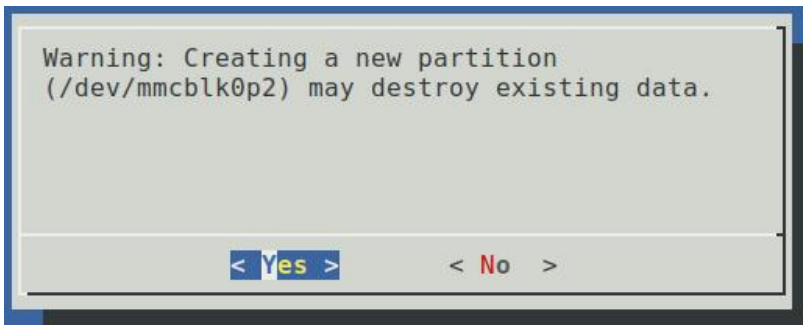
d. Then select **block** and press enter



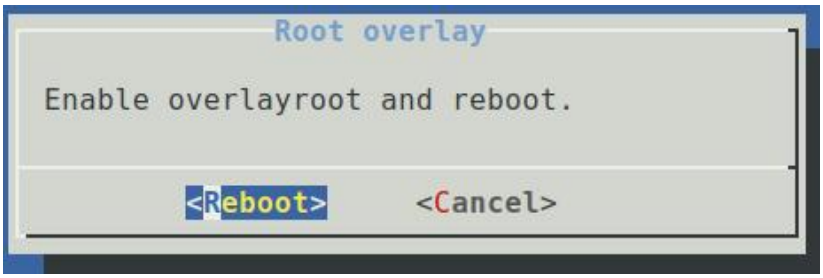
e. Then an Overlay writable layer partition will be created. You need to select the storage device where the **Overlay writable layer** partition is located. If no other USB devices are inserted into the development board, there is only one option by default, which means creating a partition in the TF card. The size of the partition is: the total capacity of TF minus twice the capacity occupied by rootfs.



f. Then select**<Yes>**to automatically create and format partitions



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



2) Then enter the **mount | grep root** command to see the mounting information of the file system.

```
orangepi@orangepi:~$ mount |grep root
/dev/mmcblk0p1 on /media/root-ro type ext4 (ro,relatime)
/dev/mmcblk0p2 on /media/root-rw type ext4 (rw,relatime)
overlayroot on / type overlay (rw,relatime,lowerdir=/media/root-ro,upperdir=/media/root-rw/overlay,workdir=/media/root-rw/overlay-workdir/_)
overlayroot on /var/log.hdd type overlay (rw,relatime,lowerdir=/media/root-ro,upperdir=/media/root-rw/overlay,workdir=/media/root-rw/overlay-workdir/_)
orangepi@orangepi:~$
```

From the mounting information, it can be seen that the file system partition /dev/mmcblk0p1 is mounted in read-only mode to /media/root-ro，/dev/mmcblk0p2 is mounted as a writable layer to /media/root-rw, indicating that the overlay root configuration has taken effect.

Since /dev/mmcblk0p2 is a partition on the TF card, data will not be lost after restarting.

3) Because all user modifications are redirected to the **/media/root-rw** directory, it is possible to restore factory settings by deleting all content under **/media/root-rw/**(**which requires a restart to take effect**).

```
orangepi@orangepi:~$ rm -rf /media/root-rw/*
orangepi@orangepi:~$ reboot
```

3. 36. 3. Method for disabling Overlayboot

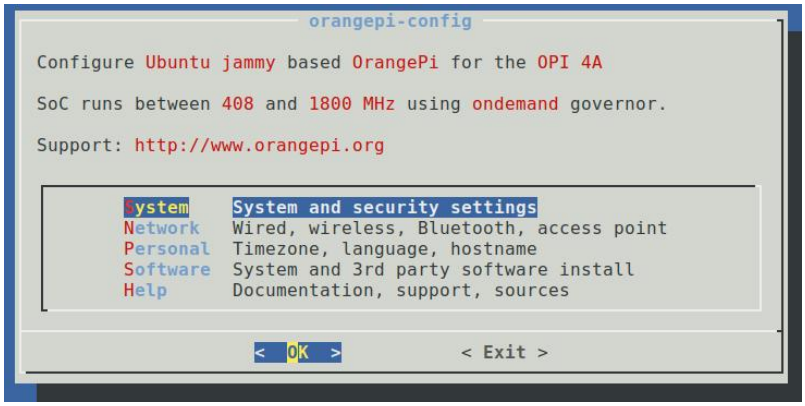
1) The steps to disable Overlayboot are as follows:

a. First, run **orangepi-config**. Regular users should remember to add **sudo**

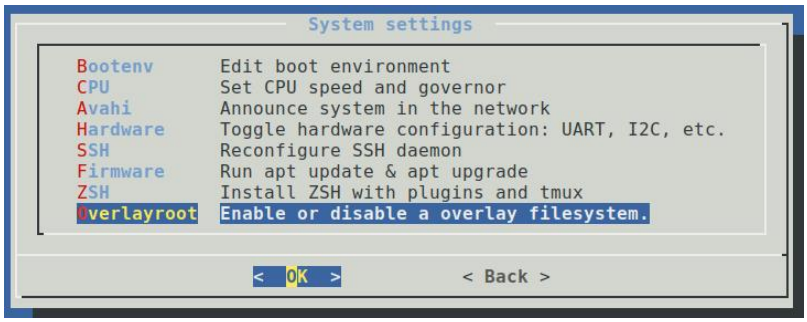
privileges

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

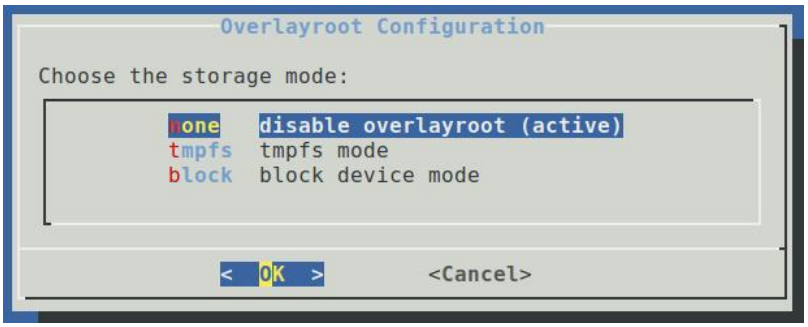
b. Then select **System**



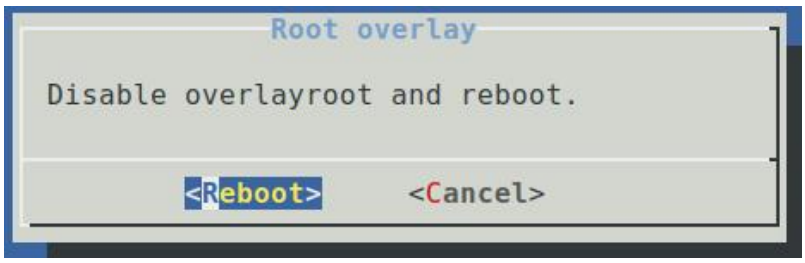
c. Then select **Overlayroot**



d. Then select **none**，and press enter



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



2) If you find using orangepi-config cumbersome, you can also follow the steps below.

a. Firstly, it is necessary to re mount the **root-ro** directory in writable mode.

```
orangepi@orangepi:~$ sudo mount -o remount,rw /media/root-ro
```

b. Then modify the **/etc/overlayroot.conf** file under the read-only partition, set the overlayroot variable to empty, and restart the system.

```
orangepi@orangepi:~$ sudo vim /media/root-ro/etc/overlayroot.conf
overlayroot=""
```

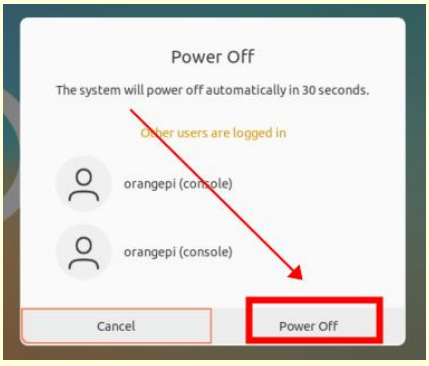
3. 37. How to shut down and restart the development board

1) If you unplug the power supply directly while the Linux system is running, some data may be lost in the file system. It is recommended to use the **poweroff** command to shut down the Linux system of the development board before unplugging the power supply.

```
orangepi@orangepi:~$ sudo poweroff
```

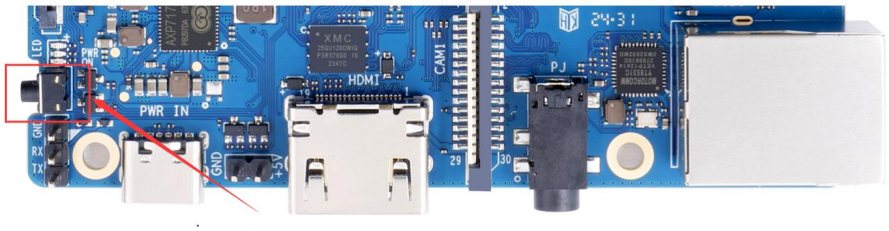
2) In addition, the development board is equipped with a power button, and you can also **short press** the power button on the development board to shut down.

Note that when you press the power button on the Linux desktop system, a confirmation box as shown in the figure below will pop up. You need to click the Power Off option before the system will shut down.



3) After shutting down, long press the power button on the development board to turn it

on.



4) Use the reboot command to restart the Linux system in the development board

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

4. Linux SDK——orangepi-build usage instructions

4. 1. Compilation system requirements

Linux SDK, **orangepi-build**, only supports running on X64 computers with Ubuntu 22.04 installed, so before downloading orangepi-build, please first 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 change 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 your computer is running Windows and does not have Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in Windows. But please note that you should not compile orangepi-build on a WSL virtual machine, because orangepi-build has not been tested in a WSL virtual machine, so you cannot ensure that orangepi-build can be used normally in WSL. In addition, please do not use orangepi-build in the Linux system of the development board. The installation image download address of Ubuntu 22.04 **amd64** version is:

https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso

After installing Ubuntu 22.04 on your computer or in a virtual machine, please first set the software source of Ubuntu 22.04 to Tsinghua source (or other domestic sources that you think are fast), otherwise it is easy to make mistakes when installing software later due to network reasons. The steps to replace Tsinghua source are as follows：

a. To replace Tsinghua source, please refer to the instructions on this page

https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/

b. Note that the Ubuntu version needs to be switched to 22.04.

c. The content of the **/etc/apt/sources.list** file that needs to be replaced is：

test@test:~$ **sudo mv /etc/apt/sources.list cat /etc/apt/sources.list.bak**

test@test:~$ **sudo vim /etc/apt/sources.list**

#The source mirror is commented out by default to increase the speed of apt update. You can uncomment it if necessary.

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse

deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse


# Pre-release software source, not recommended to enable
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

d. After the replacement, you need to update the package information and ensure that there is no error

```
test@test:~$ sudo apt-get update
```

e. In addition, since the source code of the kernel and U-boot are stored on GitHub, it is very important to make sure that the computer can download the code from GitHub normally when compiling the image.

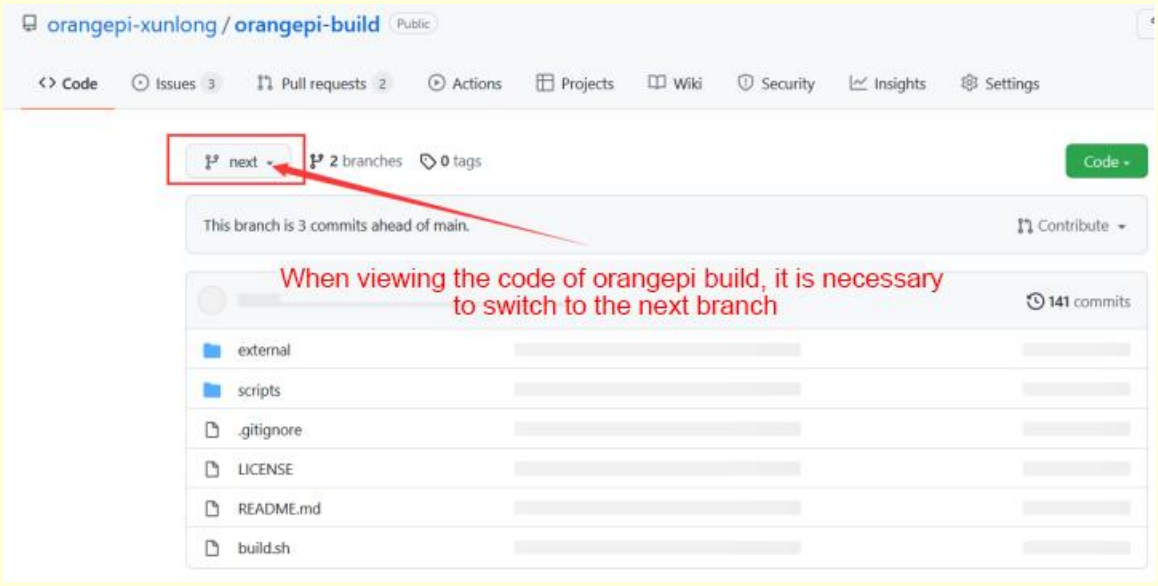
4. 2. Get the source code of Linux SDK

4. 2. 1. Download orangepi-build from github

Linux SDK refers to the orangepi-build code. Orangepi-build is modified based on the armbian build compilation system. Orangepi-build can be used to compile multiple versions of Linux images. 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 to use the T527 Soc development board, you need to download the next branch source code of orangepi-build. The above git clone command needs to specify the branch of orangepi-build source code as next.



You do not need to enter the username and password of the GitHub account when downloading the orangepi-build code through the git clone command (the same applies to downloading other codes in this manual). If the Ubuntu PC prompts you to enter the username and password of the GitHub account after entering the git clone command, it is usually because the address of the orangepi-build warehouse after git clone is entered incorrectly. Please check the command spelling carefully for any errors, instead of thinking that we forgot to provide the username and password of the GitHub account here.

The u-boot and linux kernel versions currently used by the T527 series development board are as follows：

Branches	u-boot Version	Linux Kernel version
current	u-boot v2018.05	linux5.15

The branch mentioned here is not the same as the branch of orangepi-build source code, please do not confuse them. This branch is mainly used to distinguish different kernel source code versions.

The Linux 5.15 BSP kernel currently provided by Allwinner is defined as the current branch.

orangepi-build After downloading, the following files and folders will be included：

a. **build.sh**: Compile the startup script
b. **external**: Contains configuration files, specific scripts, and source code of some programs needed to compile the image.
c. **LICENSE**: GPL 2 License File
d. README.md: orangepi-build Documentation
e. **scripts**: Generic script for compiling linux images

```
test@test:~/orangepi-build$ ls
build.sh  external  LICENSE  README.md  scripts
```

If you download the orangepi-build code from github, you may find that orangepi-build does not contain the source code of u-boot and linux kernel, nor the cross-compilation toolchain required to compile u-boot and linux kernel. This is normal because these things are stored in other separate github repositories or some servers (the addresses will be detailed below). orangepi-build will specify the addresses of u-boot, linux kernel and cross-compilation toolchain in the script and configuration file. When running orangepi-build, if it finds that these things are not available locally, it will automatically download them from the corresponding places.

4.2.2. Download the cross-compilation toolchain

When orangepi-build is run for the first time, it will automatically download the cross-compilation toolchain and put it in the **toolchains** folder. Each time you run the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists. If not, it will restart the download. If it exists, it will be used directly without repeated download.

```
[ o.k. ] Checking for external GCC compilers
[ .... ] downloading using http(s) network [ gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz ]
 #8d7029 16MiB/24MiB(65%) CN:1 DL:7.9MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz: 24.9MiB [14.4MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz ]
 #e30eec 17MiB/33MiB(50%) CN:1 DL:10MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz: 33.9MiB [9.66MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux.tar.xz ]
 #041c24 48MiB/48MiB(99%) CN:1 DL:2.7MiB]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux.tar.xz: 48.8MiB [13.0MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz ]
 #3dee3e 72MiB/76MiB(93%) CN:1 DL:3.7MiB ETA:1s]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz: 77.0MiB [14.2MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz ]
 #42e728 104MiB/104MiB(99%) CN:1 DL:2.8MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz:  104MiB [13.9MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz ]
 #2c065e 108MiB/111MiB(97%) CN:1 DL:3.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz:  111MiB [13.4MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf.tar.xz ]
 #d232ee 250MiB/251MiB(99%) CN:1 DL:2.0MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf.tar.xz:  251MiB [13.7MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu.tar.xz ]
 #88b441 268MiB/269MiB(99%) CN:1 DL:0.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
```

The mirror website of the cross-compilation tool chain in China is the open source software mirror website of Tsinghua University：

https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/

Toolchains After downloading, it will contain multiple versions of cross-compilation tool chains：

```
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 tool chain used to compile the T527 Linux kernel source code is：

a. linux5.15

```
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
```

The cross-compilation toolchain used to compile the T527 u-boot source code is：

a. v2018.05

```
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
```

4.2.3. Explanation of the complete directory structure of orangepi build

1) After downloading the orangepi build repository, it does not include the Linux kernel, U-boot source code, or cross compilation toolchain. The Linux kernel and U-boot source code are stored in separate Git repositories

a. The git repository where the Linux kernel source code is stored is as follows. Please note to switch the branch of the Linux orangepi repository to

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.15-sun55iw3
```

b. The git repository where the u-boot source code is stored is as follows. Please note to switch the branch of the u-boot orangepi repository to

```
https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2018.05-t527
```

2) When Orangepi build is first run, it will download the cross compilation toolchain, u-boot, and Linux kernel source code. After successfully compiling the Linux image once, the files and folders that can be seen in Orangepi build are:

a. **build.sh**: Compile startup script
b. **external**: Contains configuration files required for compiling the image, scripts for specific functions, and source code for some programs. The rootfs compressed file cached during the image compilation process is also stored in the external file
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 deb packages such as u-boot and Linux, compilation logs, and compiled images
g. **scripts**: General script for compiling Linux images
h. **toolchains**: Store cross compilation toolchain
i. **u-boot**: Store the source code of u-boot
j. **userpatches**: Store the configuration files required for compiling scripts

```
test@test:~/orangepi-build$ ls
```

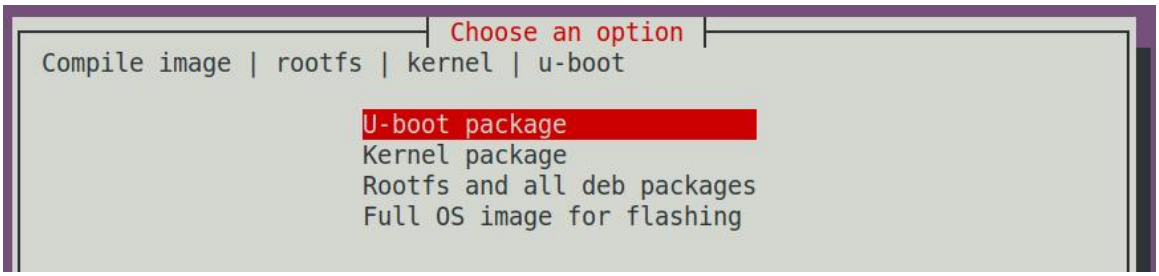
```
build.sh external kernel LICENSE output README.md scripts toolchains u-boot userpatches
```

4. 3. Compiling u-boot

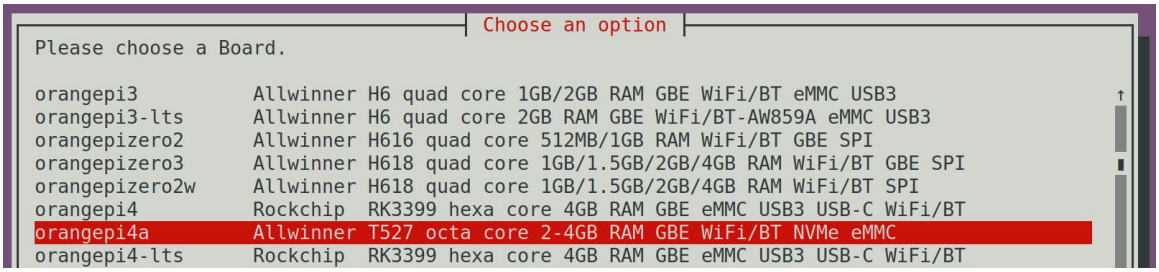
1) Run the build.sh script, remember to grant sudo privileges

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package** and press Enter



3) Next, select the model of the development board



4) Then it will start compiling u-boot, and some of the information prompted when compiling the current branch is as follows:

a. Version of u-boot source code

```
[ o.k. ] Compiling u-boot [ v2018.05 ]
```

b. Version of cross compilation toolchain

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 11 ]
```

c. The path of the compiled u-boot deb package

```
[ o.k. ] Target directory [ orangepi-build/output/debs/u-boot ]
```

d. The package name of the compiled u-boot deb package

```
[ o.k. ] File name [ linux-u-boot-current-orangepi4a_x.x.x_arm64.deb ]
```

e. Compilation time used

```
[ o.k. ] Runtime [ 1 min ]
```

f. Repeat the command to compile u-boot, and 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=orangepi4a
BRANCH=current BUILD_OPT=u-boot ]
```

5) View the compiled u-boot deb package

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-current-orangepi4a_x.x.x_arm64.deb
```

6) When the orangepi build compilation system compiles the U-boot source code, it first synchronizes the U-boot source code with the GitHub server's U-boot source code. Therefore, 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 **(you need to compile the U-boot completely before turning off this function, otherwise you will prompt that the U-boot source code cannot be found)**. Otherwise, the modifications 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"
.....
```

7) When debugging u-boot code, you can use the following method to update u-boot in the Linux image for testing

a. Firstly, upload the compiled deb package of u-boot to the Linux system on the development board

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-current-orangepi4a_x.x.x_arm64.deb root@192.168.1.xxx:/root
```

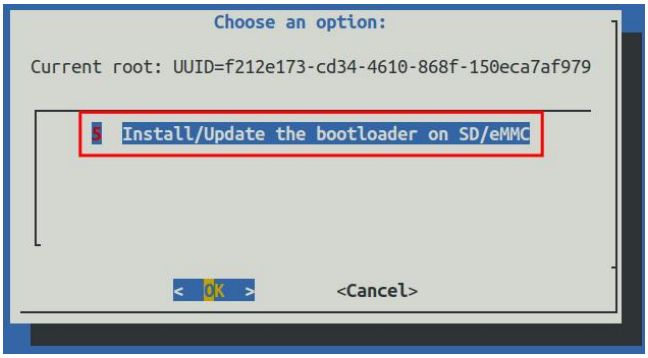
b. Reinstall the newly uploaded deb package for u-boot

```
orangepi@orangepi:~$ sudo dpkg -i linux-u-boot-current-orangepi4a_x.x.x_arm64.deb
```

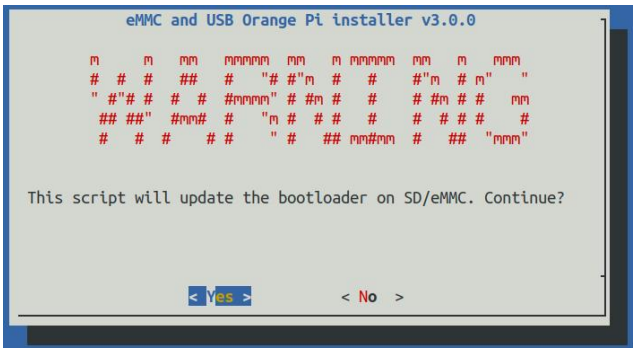
c. Then run the nand sata install script

```
orangepi@orangepi:~$ sudo nand-sata-install
```

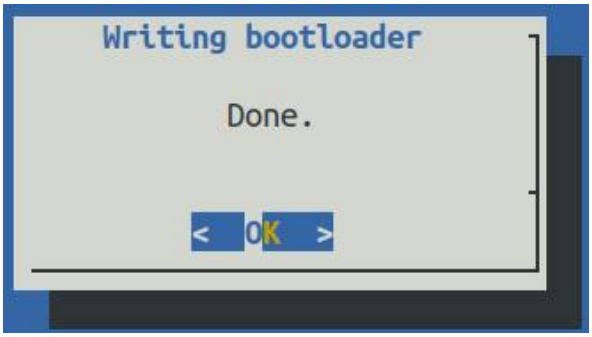
d. Then select **5 Install/Update the bootloader on SD/eMMC**



e. After pressing the enter key, a warning will first pop up



f. Pressing the enter key again will start updating u-boot, and after the update is complete, the following information will be displayed



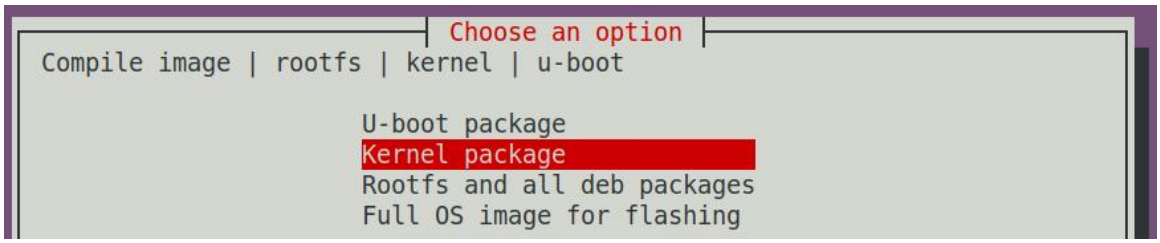
g. Then you can restart the development board to test whether the u-boot modifications have taken effect

4. 4. Compiling Linux Kernel

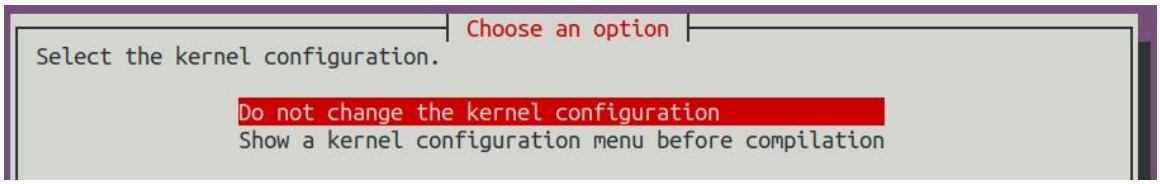
1) Run the **build.sh** script, remember to grant sudo privileges

```
test@test:~/orangepi-build$ sudo ./build.sh
```

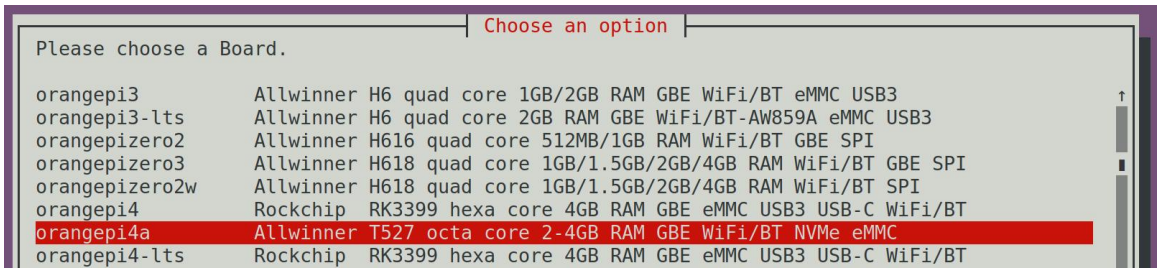
2) Select **Kernel package** and press Enter



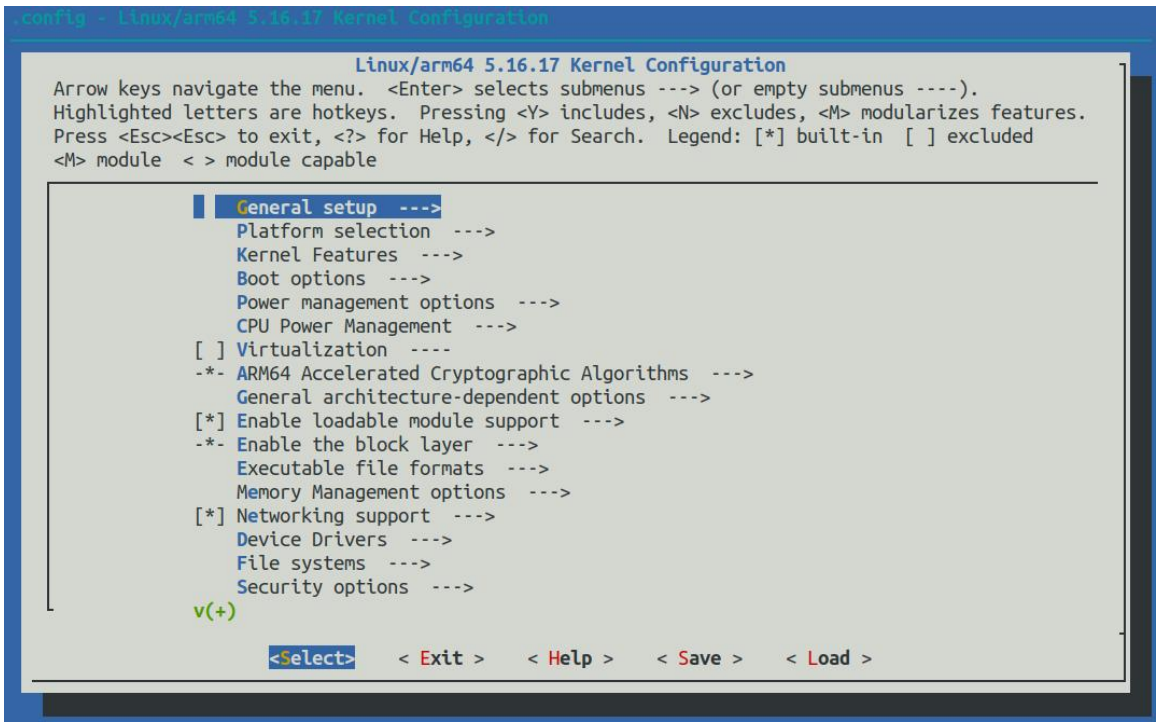
3) Then it will prompt whether the kernel configuration interface needs to be displayed. If the kernel configuration does not need to be modified, select the first one. If the kernel configuration needs to be modified, select the second one



4) Next, select the model of the development board



5) If step 3) selects the option to display the kernel configuration menu (second option), a kernel configuration interface opened through **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration, save and exit after modification, and the kernel source code will be compiled after exit.



a. If there is no need to modify the configuration options of the kernel, passing **KERNEL_CONFIGURE=no** when running the build.sh script can 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 default.exe configuration file to permanently disable this feature

c. If the following error appears when compiling the kernel, it is due to the small terminal interface of Ubuntu PC, which causes the make menuconfig interface to not display. Please set the terminal of Ubuntu PC to its maximum size 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
```

6) The following is a partial explanation of the information prompted when compiling the current branch kernel source code:

a. Version of Linux kernel source code

```
[ o.k. ] Compiling current kernel [ 5.15.147 ]
```

b. The version of the cross compilation toolchain used

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 11 ]
```

c. The default configuration file used by the kernel and its storage path are as follows

```
[ o.k. ] Using kernel config file
[ orangepi-build/external/config/kernel/linux-5.15-sun55iw3-current.config ]
```

d. The path of the compiled kernel related deb package

```
[ o.k. ] Target directory [ output/debs/ ]
```

e. The package name of the compiled kernel image deb package

```
[ o.k. ] File name [ linux-image-current-sun55iw3_x.x.x_arm64.deb ]
```

f. Compilation time used

```
[ o.k. ] Runtime [ 10 min ]
```

g. Finally, the compilation command for the kernel selected last time will be displayed. The following command can be used to start compiling the kernel source code without selecting it through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepi4a
BRANCH=current BUILD_OPT=kernel KERNEL_CONFIGURE=no ]
```

7) View the compiled kernel related deb packages

a. **linux-dtb-current-sun55iw3_x.x.x_arm64.deb** Contains dtb files used by the kernel

b. **linux-headers-current-sun55iw3_x.x.x_arm64.deb** Contains kernel header files

c. **linux-image-current-sun55iw3_x.x.x_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-current-sun55iw3_x.x.x_arm64.deb
output/debs/linux-headers-current-sun55iw3_x.x.x_arm64.deb
output/debs/linux-image-current-sun55iw3_x.x.x_arm64.deb
```

8) When the Orangepi build compilation system compiles the Linux kernel source code, it first synchronizes the Linux kernel source code with the GitHub server's Linux kernel source code. Therefore, if you want to modify the Linux kernel source code, you first need to turn off the source code update function **(which requires a complete compilation of the Linux kernel source code before it can be turned off, otherwise it will prompt that the Linux kernel source code cannot be found)**. Otherwise, the modifications 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"
```

9) If the kernel has been modified, you can use the following methods to update the kernel and kernel modules of the development board linux system

a. Upload the compiled deb package of the linux kernel to the linux system on the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-current-sun55iw3_x.x.x_arm64.deb root@192.168.1.xxx:/root
```

b. Install the new linux kernel deb package that you just uploaded

```
orangepi@orangepi:~$ sudo dpkg -i linux-image-current-sun55iw3_x.x.x_arm64.deb
```

c. Restart the development board and check whether the kernel changes have taken effect

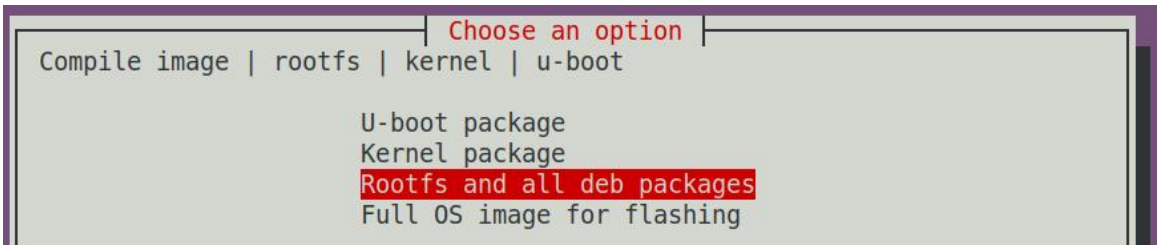
```
orangepi@orangepi:~$ sudo reboot
```

4. 5. Compile rootfs

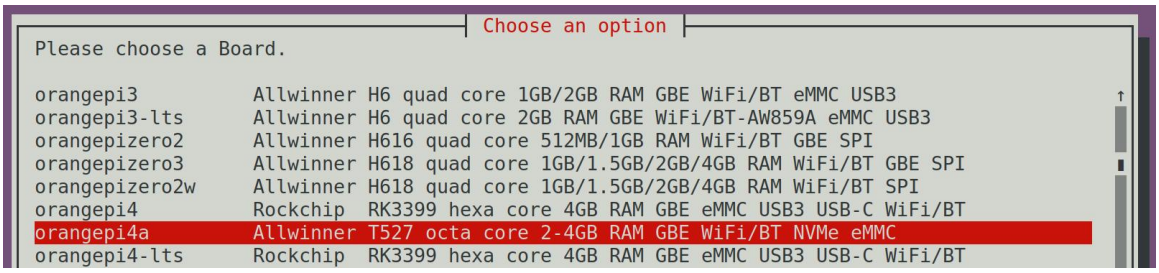
1) Run the build.sh script and remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

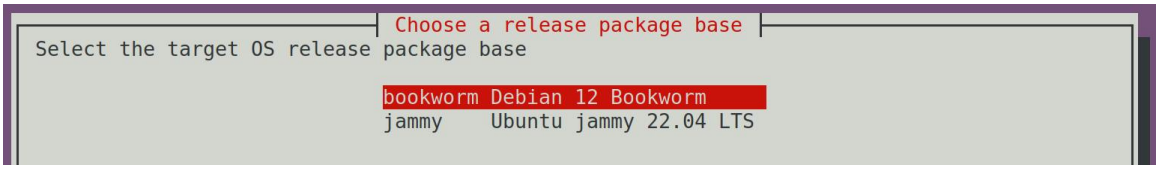
2) Select **Rootfs and all deb packages**，then press enter



3) Then select the model of the development board



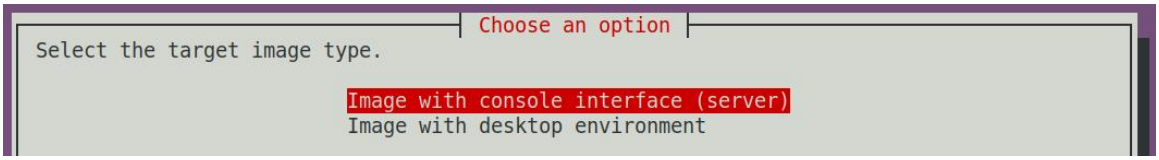
4) Then select the type of rootfs



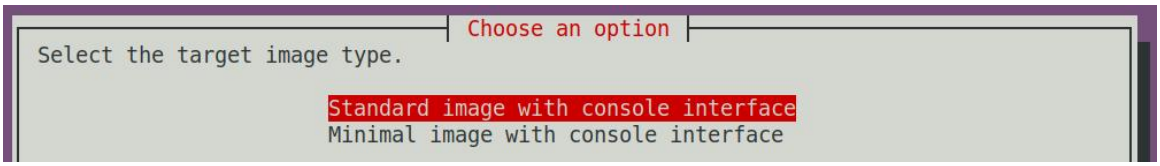
5) Then select the type of image

a. **Image with console interface (server)** Indicates the server version of the image, the volume is relatively small

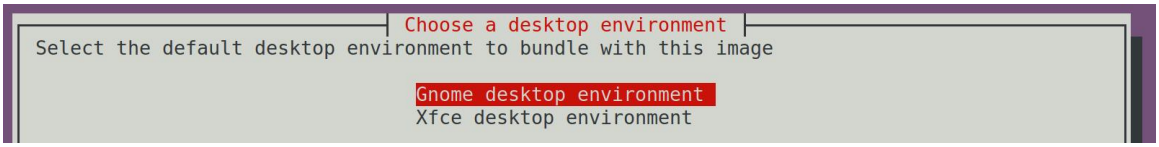
b. **Image with desktop environment** It is a large image with a desktop

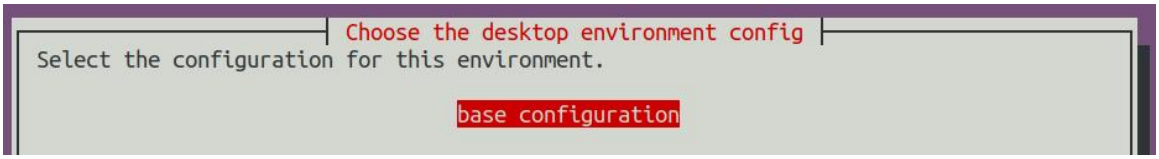


6) If it is the image of the compile server version, you can also choose to compile the Standard version or Minimal version, the Minimal version of the pre-installed software will be much less than the Standard version **(no special requirements, please do not choose the Minimal version, because many things are not pre-installed by default, some functions may not be used)**

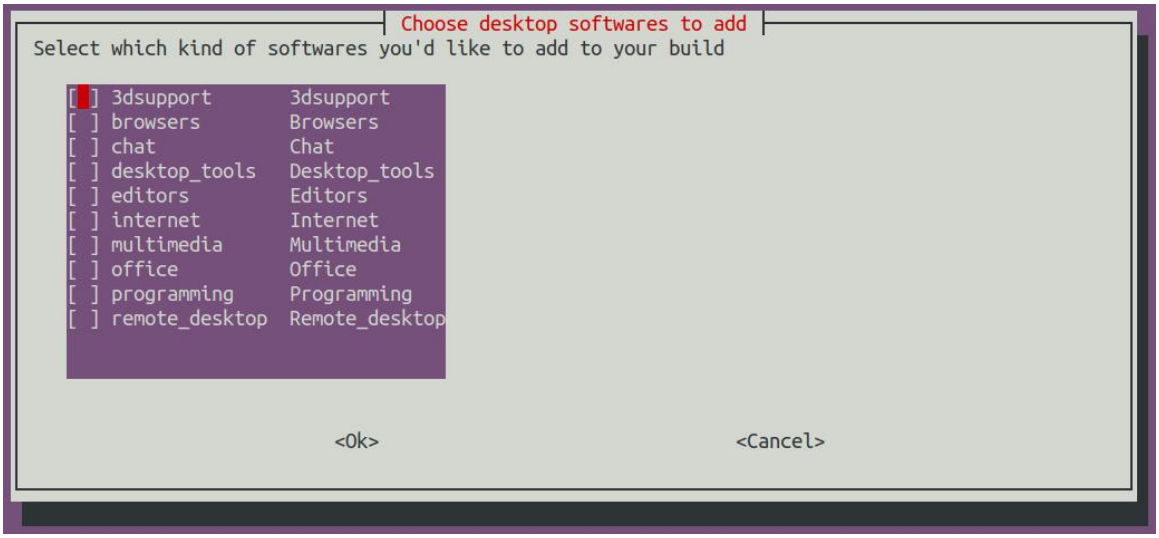


7) If you compile the desktop version of the image also need to select the type of desktop environment, currently only maintain GNOME, so please choose GNOME type desktop





You can then select additional packages that need to be installed. Please press Enter to skip this.



8) You will then start compiling the rootfs. Some of the information prompted during compilation is described below

a. The type of rootfs

[o.k.] local not found [Creating new rootfs cache for **jammy**]

b. Directory for storing the generated rootfs package

```
[ o.k. ] Target directory [ orangepi-build/external/cache/rootfs ]
```

c. Name of the generated rootfs package

```
[ o.k. ] File name [ jammy-gnome-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4 ]
```

9) View the compiled rootfs compressed package

a. **jammy-gnome-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4** is rootfs zip, the meaning of each field name is

a) **jammy** indicates the type of linux distribution for rootfs

b) **gnome** indicates that rootfs is the type of the desktop version. If it is **cli**, it indicates the type of the server version

c) **arm64** indicates the schema type of rootfs

d) **25250ec7002de9e81a41de169f1f89721** is generated by the package name rootfs to install all of the packages of the MD5 hash value, as long as no modification rootfs installation package list, then this value will not change, The compilation script uses this MD5 hash to determine if the rootfs needs to be recompiled

b. **jammy-gnome-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
jammy-gnome-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4
jammy-gnome-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.current
jammy-gnome-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list
```

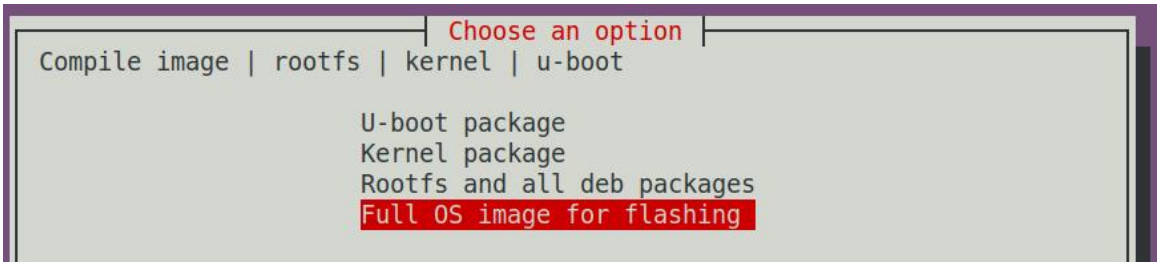
10) If the required rootfs already exists under **external/cache/rootfs**, compiling the rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to the **external/cache/rootfs** to check whether there is a cached rootfs. If there is one, it will be used directly, which can save a lot of download and compilation time

4. 6. Compile the linux image

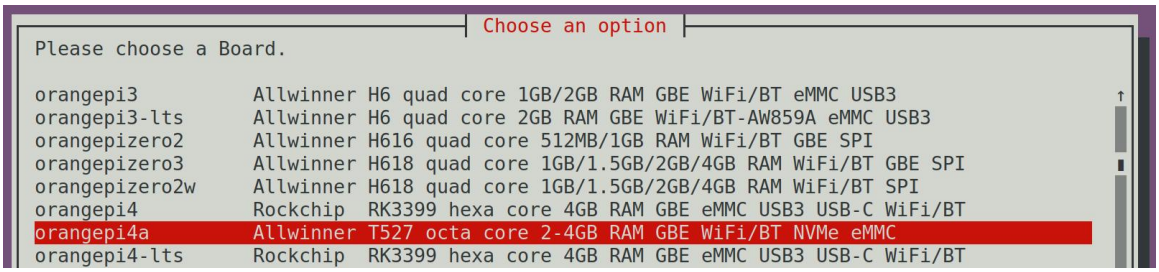
1) Run the **build.sh** script and remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

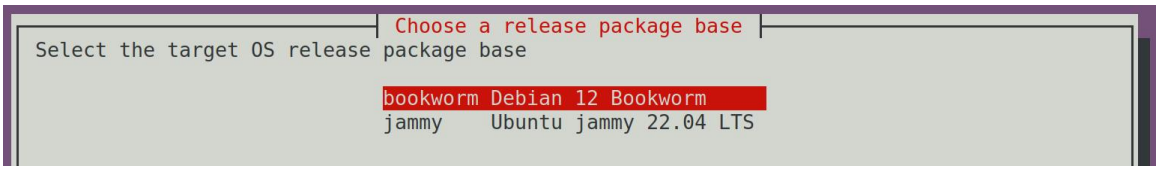
2) Select **Full OS image for flashing** and press Enter



3) Then select the model of the development board



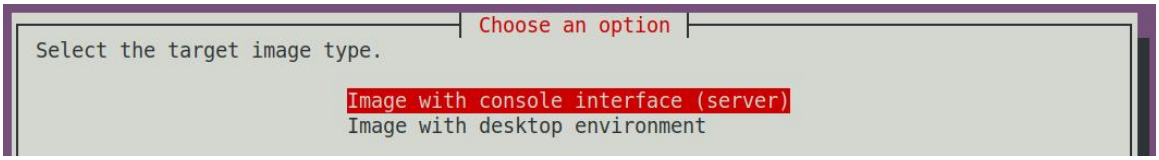
4) Then select the type of rootfs



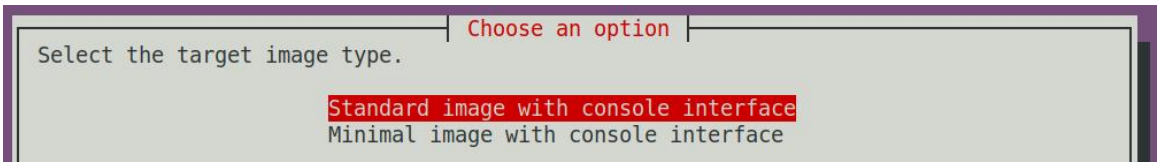
5) Then select the type of image

a. **Image with console interface (server)** Indicates the server version of the image, the volume is relatively small

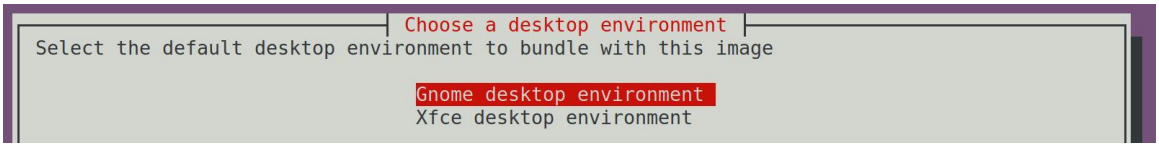
b. **Image with desktop environment** It is a large image with a desktop

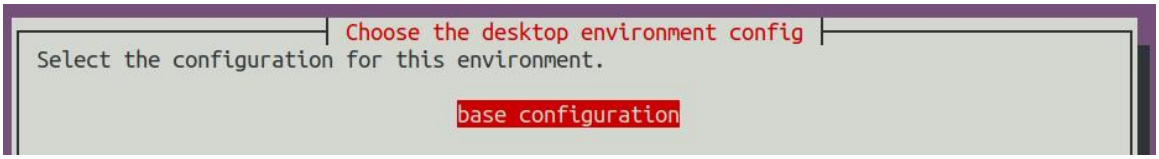


6) If it is the image of the compile server version, you can also choose to compile the Standard version or Minimal version, the Minimal version of the pre-installed software will be much less than the Standard version **(no special requirements, please do not choose the Minimal version, because many things are not pre-installed by default, some functions may not be used)**

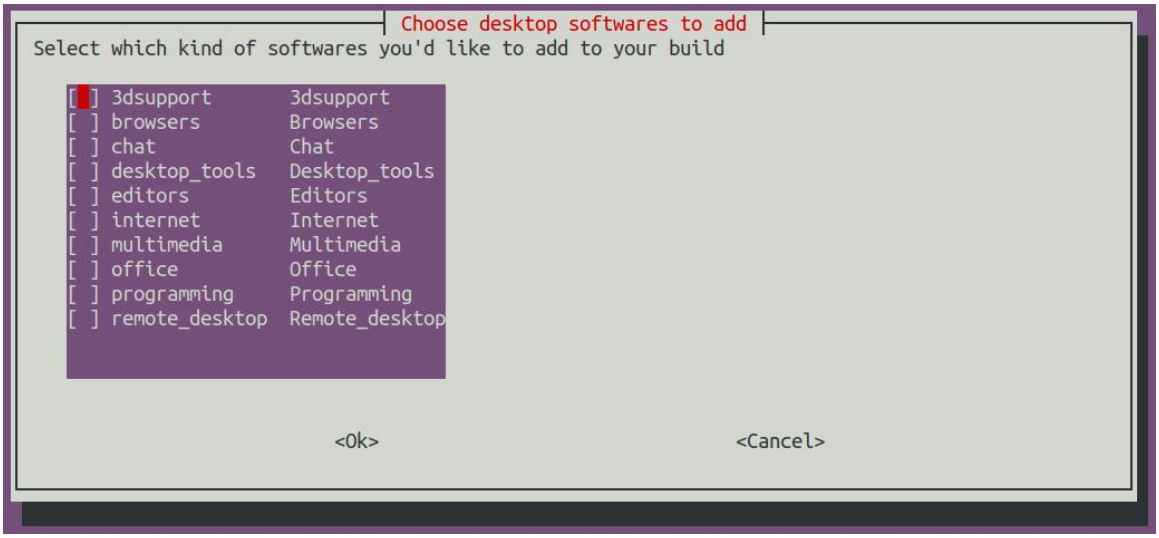


7) If you compile the desktop version of the image also need to select the type of desktop environment, currently only maintain GNOME, so please choose GNOME type desktop





You can then select additional packages that need to be installed. Please press Enter to skip this.



8) The linux image will then be compiled, the general process of compilation is as follows

a. Initialize the Ubuntu PC compilation environment and install the software packages required for the compilation process
b. Download the u-boot and linux kernel source code (if cached, update the code only)

c. Compile u-boot source code to generate the u-boot deb package
d. Compile linux source code and generate Linux-related deb packages
e. Create the deb package of the linux firmware
f. Create the deb package of the orangepi-config tool
g. Create a board-level deb package
h. If you are compiling the desktop version image, you will also make desktop related deb packages
i. Check whether the rootfs is cached. If no, create a new rootfs. If the rootFS is cached, decompress it and use it
j. Install the previously generated deb package into rootfs
k. Perform some specific Settings for different development boards and different types of images, such as preinstalling additional software packages and modifying system configurations
l. Then create an image file and format the partition. The default type is ext4
m. Copy the configured rootfs to the image partition
n. Then update initramfs
o. Run the dd command to write the bin file of the u-boot to the image

9) The following information is displayed after the image is compiled

a. Path for storing the generated image

```
[ o.k. ] Done building
[ output/images/orangepi4a_x.x.x_debian_jammy_linux5.15.xx_gnome_desktop/orangepi4a_x.x.x_debian_jammy_linux5.15.xx_gnome_desktop.img ]
```

b. Compilation time

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the image compilation command. Run the following command to compile the image without using the GUI

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepi4a BRANCH=current BUILD_OPT=image RELEASE=jammy BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

5. Android 13 operating system instructions

5. 1. Supported Android versions

Android version	Kernel version
Android 13	**linux5.15**

5. 2. Android 13 Function Adaptation

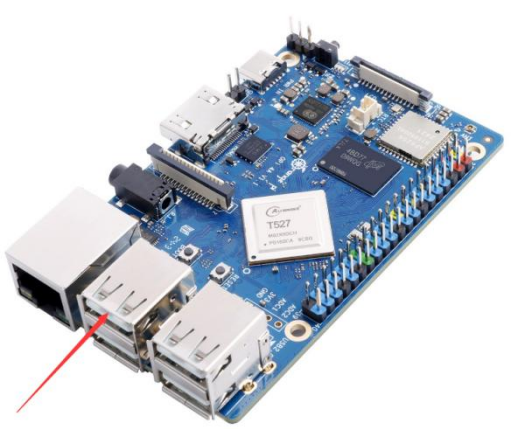
Function	Android 13
HDMI video	**OK**
HDMI Audio	**OK**
USB2.0 x 4	**OK**
TF card start	**OK**
eMMC	**OK**
Identify NVME SSDS	**OK**
Gigabit network card	**OK**
WIFI	**OK**
Bluetooth	**OK**
RTC chip	**OK**
Headphone audio	**OK**
LCD screen	**OK**
EDP	**OK**
CAM1	**NO**
CAM2	**NO**
LED light	**OK**
40 pin GPIO	**OK**
40 pin I2C	**OK**
40 pin SPI	**OK**
40 pin UART	**OK**

40 pin PWM	**OK**
Temperature sensor	**OK**
Mali GPU	**OK**
Video codec	**OK**

5. 3. Usage of ADB

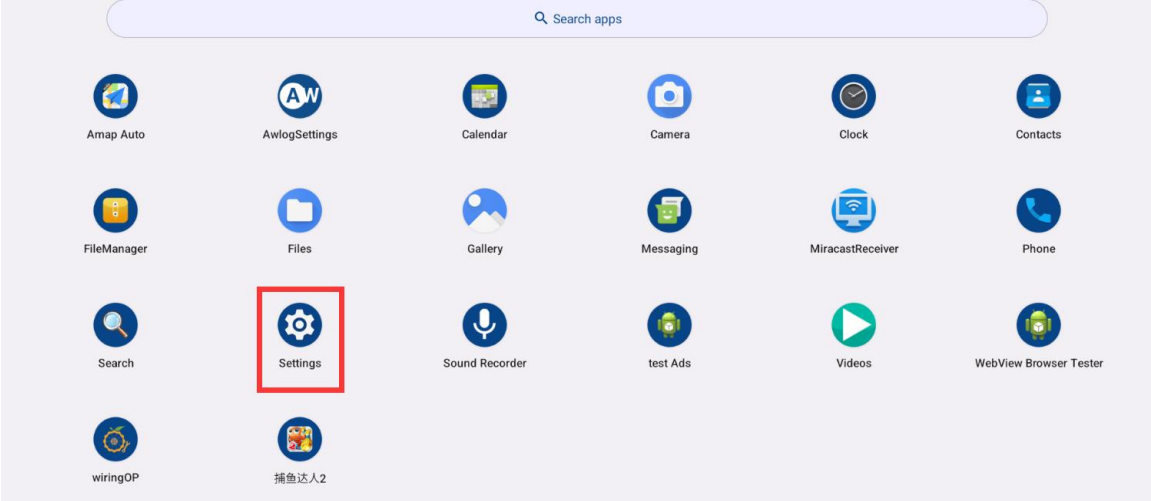
5. 3. 1. USB OTG mode switching method

The development board has four USB ports, of which the USB port marked in the red box below can support both Host mode and Device mode, and the other three USB ports only support Host mode.

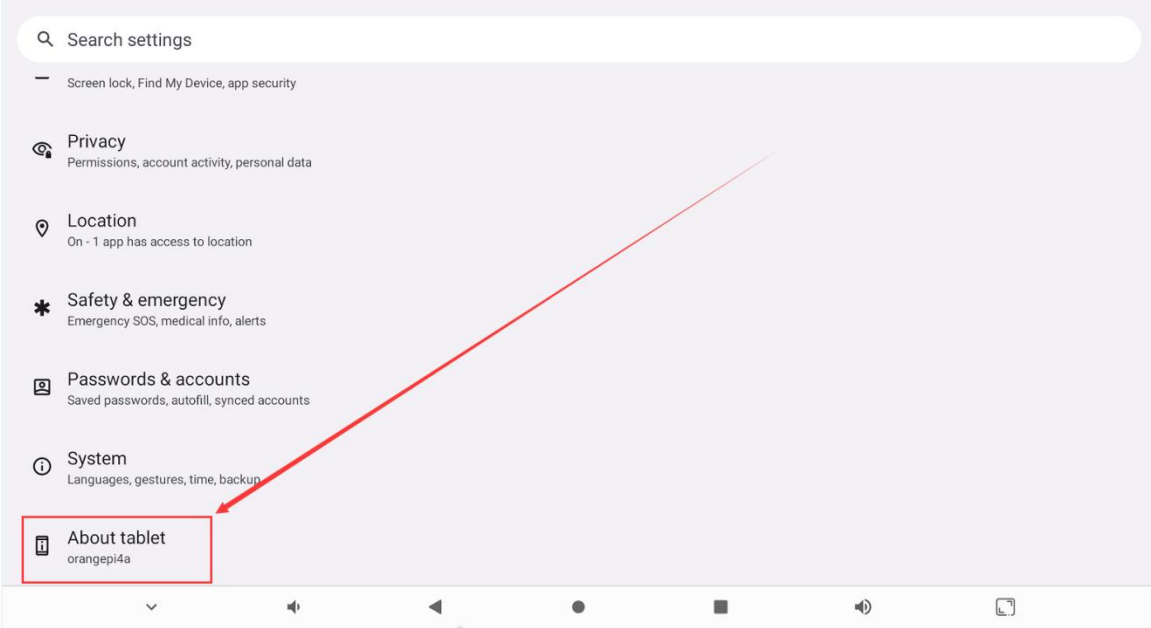


The USB OTG interface is in Host mode by default and can be used to connect USB devices such as mouse and keyboard. If you want to use ADB, you need to manually switch to Device mode.

1) First open Settings

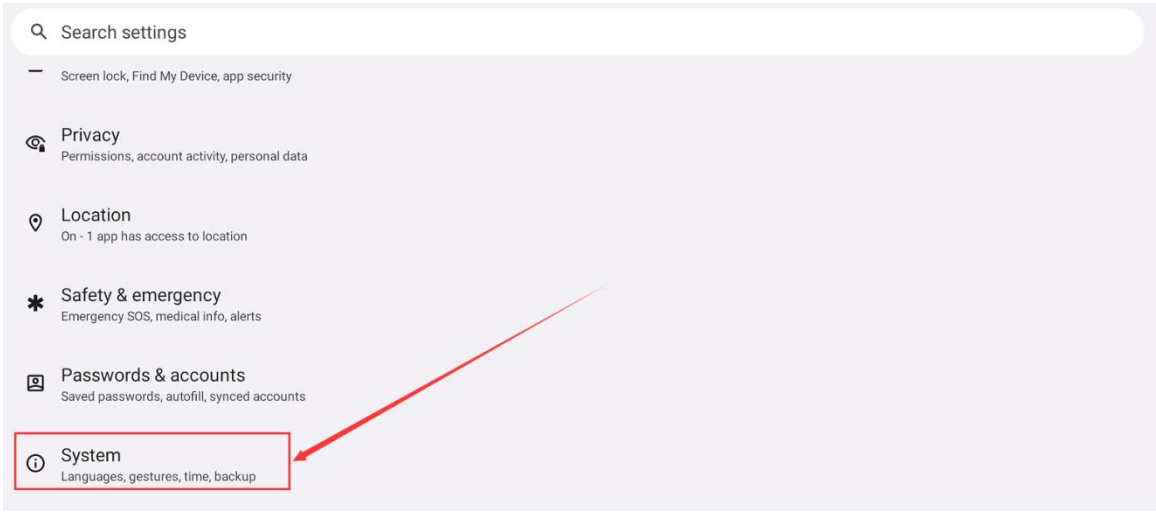


2) Then find **About tablet**



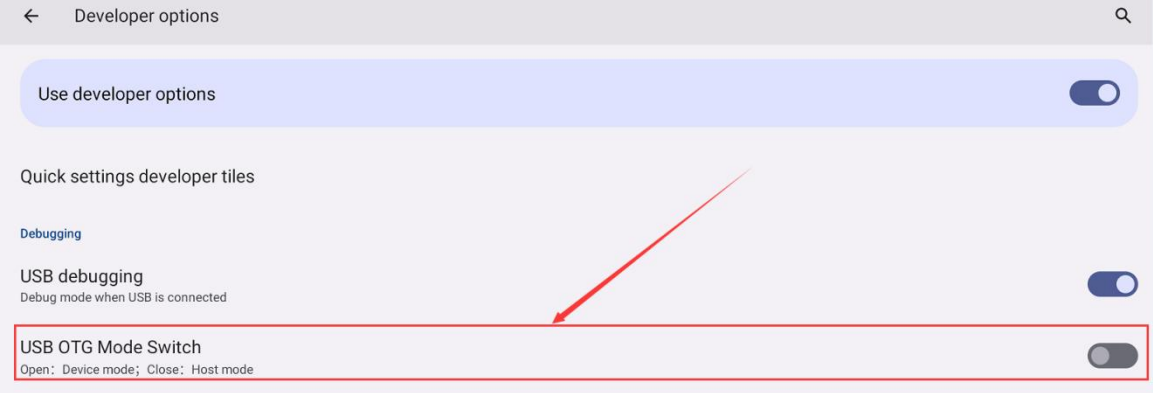
3) Then click the **Build number** option several times with your mouse until it appears. **You are now a developer!** Tips for

4) Then return to the upper-level menu and select **System**



5) Then select **Developer options**

6) Finally, find the **USB OTG Mode Switch**, **turn on the switch to switch to Device mode, turn off the switch to switch to Host mode**



5. 3. 2. Use a data cable to connect adb to debug

1) First prepare a good quality USB 2.0 public to public data cable

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 adb shell on the Ubuntu PC

```
test@test:~$ adb shell
t527-demo:/ #
```

5. 3. 3. adb debugging using a network connection

Using network adb does not require a USB2.0 peer-to-peer data cable to connect the computer to the development board, but communicates over the network, so first make sure that the wired or wireless network of the development board is connected, and then obtain the IP address of the development board, which will be used later.

1) Ensure that the **service.adb.tcp.port** of the Android operating system is set to 5555

```
console:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command in the serial port to set the port number of network adb

```
console:/ # setprop service.adb.tcp.port 5555

console:/ # stop adbd
console:/ # 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      (Change it 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 adb shell on the Ubuntu PC

```
test@test:~$ adb shell
t527-demo:/ #
```

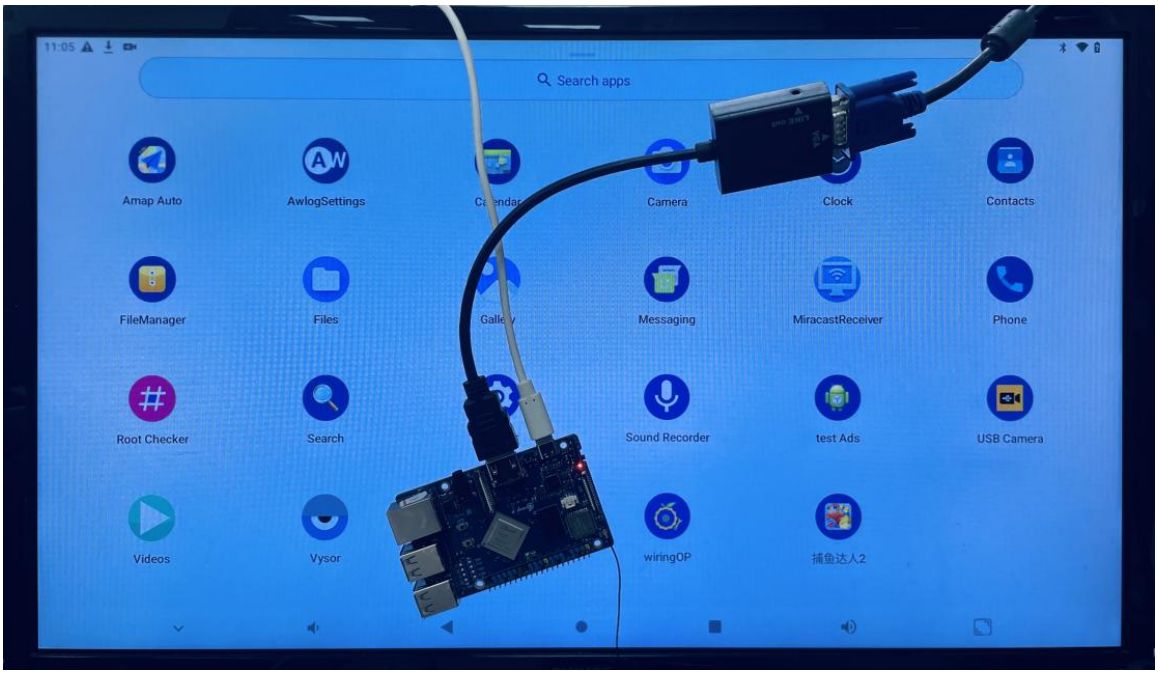
5. 4. HDMI to VGA display test

1) The following accessories need to be prepared first
   a. HDMI to VGA converter

   b. One VGA cable

   c. A monitor or TV that supports VGA port

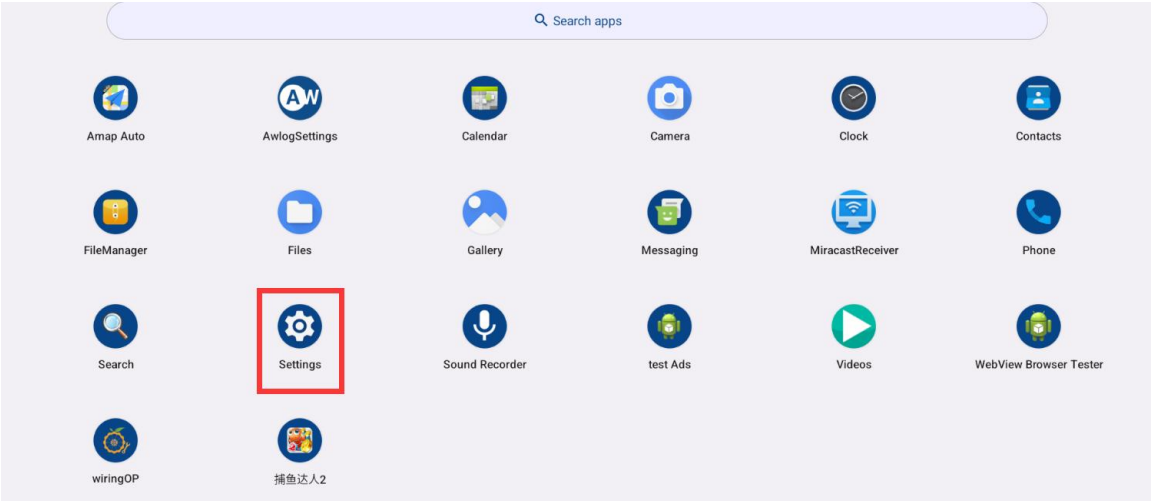
2) HDMI to VGA display test is shown below



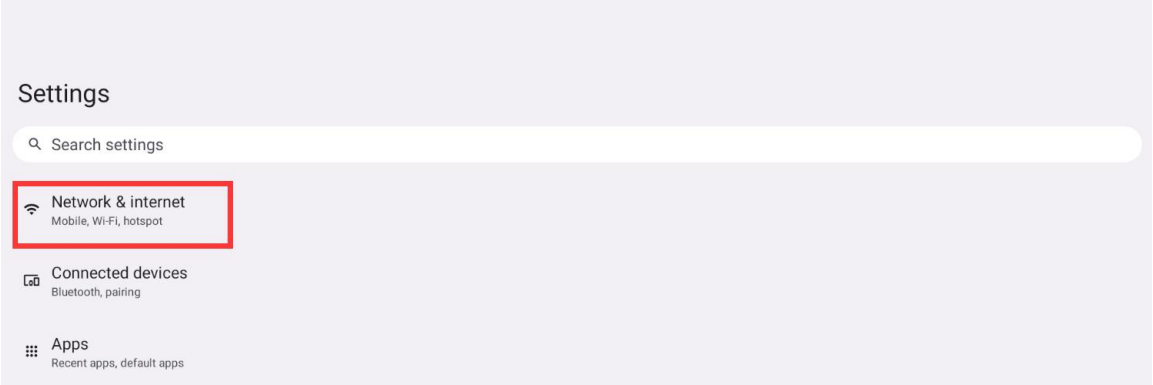
When using HDMI to VGA display, the development board and the Android system of the development board do not need to do any Settings, only the HDMI interface of the development board can be displayed normally. So if the test has problems, check the HDMI to VGA converter, VGA cable, and monitor for problems.

5. 5. WI-FI connection method

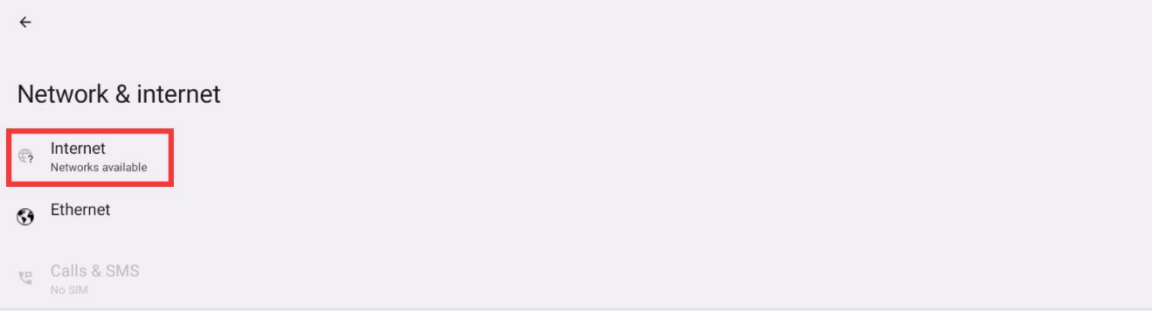
1) First select **Settings**



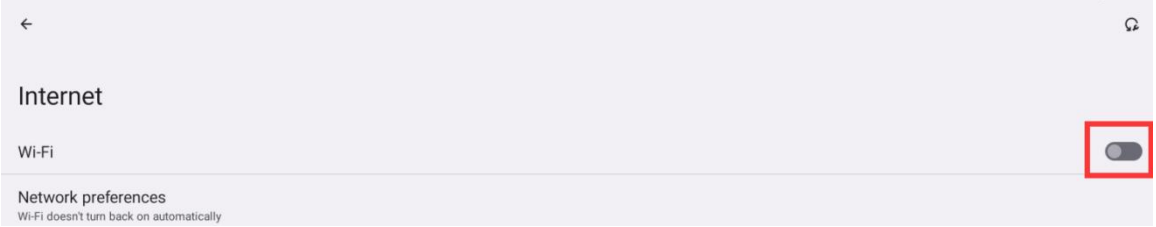
2) Then select **Network & Internet**



3) Then select **Internet**



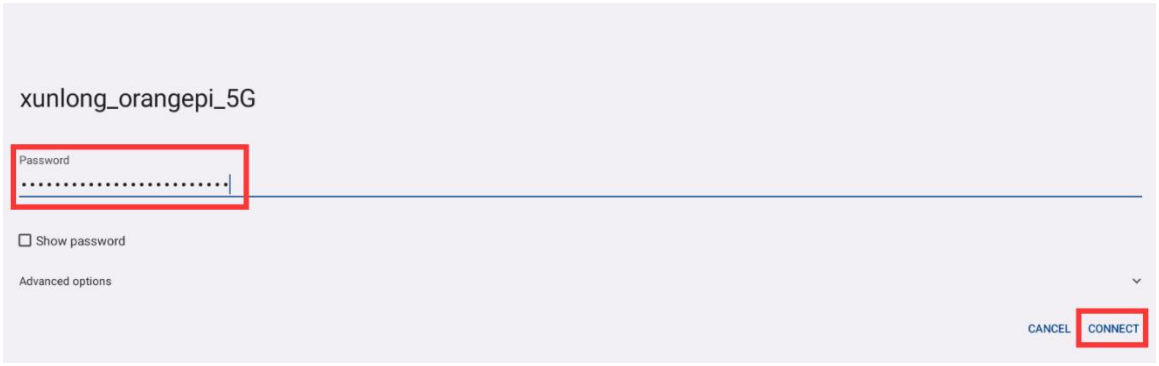
4) Then open WI-FI



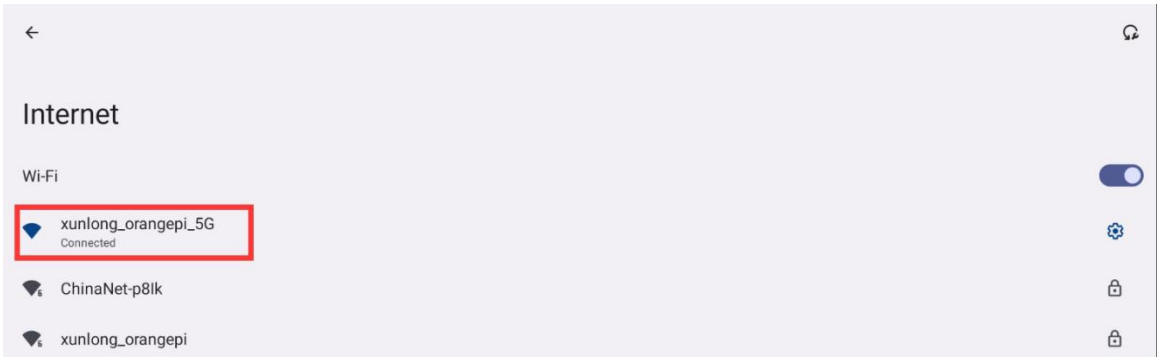
5) After turning on WI-FI, you can see the search signal under **Available networks**

6) Select the WI-FI you want to connect to, and the password input interface will pop up as shown in the following figure

7) Then use the keyboard to enter the WI-FI password, and then use the **mouse** to click the Enter button in the virtual keyboard to start the WI-FI connection



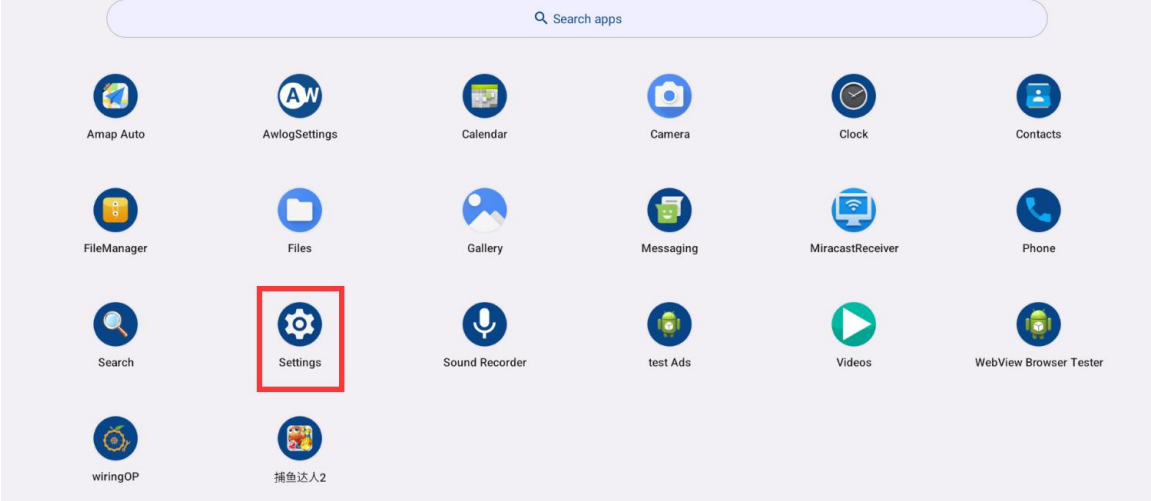
8) The display after the WI-FI connection is successful is as shown in the following figure



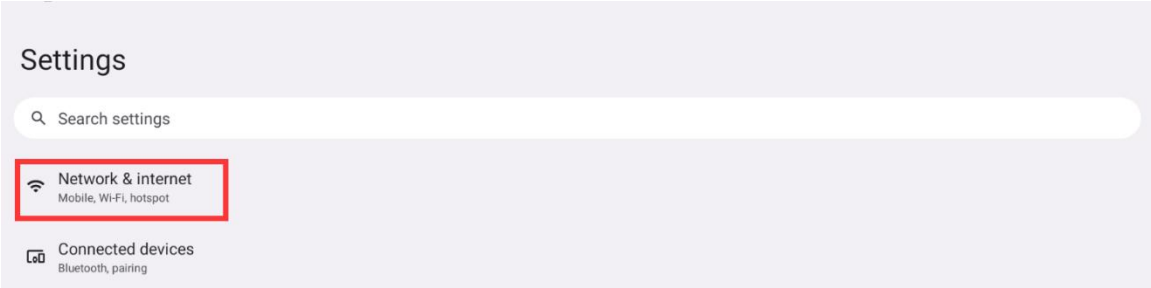
5. 6. How to use WI-FI hotspot

1) Ensure that the Ethernet port is connected to a network cable and can access the Internet properly

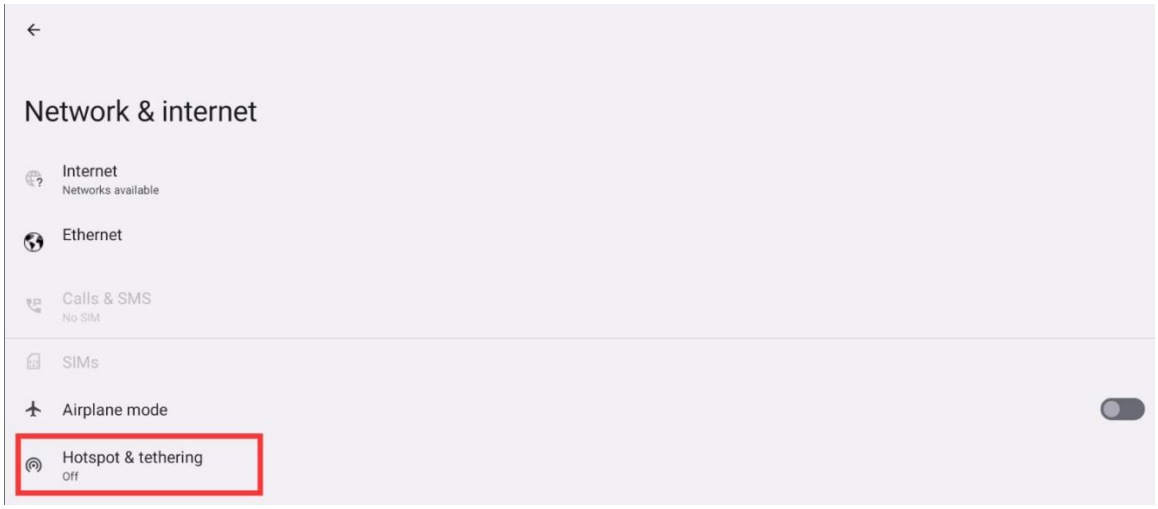
2) Then select **Settings**



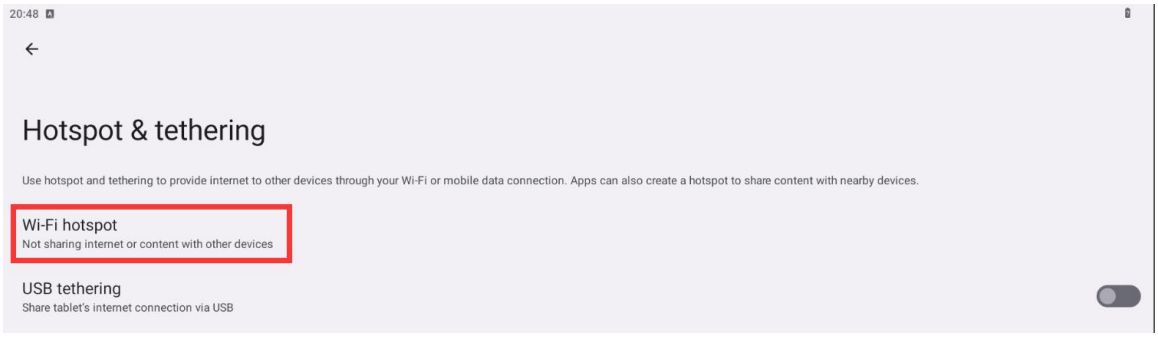
3) Then select **Network & Internet**



4) Then select **Hotspot & tethering**



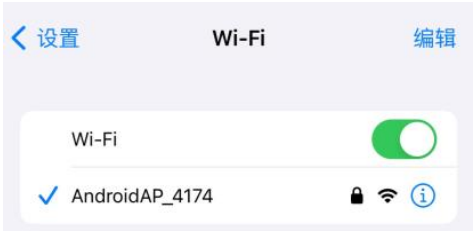
5) Then select **Wi-Fi hotspot**



6) Then open the **Wi-Fi Hotspot**, you can also see the name and password of the generated Hotspot in the following figure, remember them, and need to use them when connecting to the hotspot (if you need to change the name and password of the Hotspot, you need to close the **Wi-Fi Hotspot** first, and then modify it)

7) At this time, you can take out your mobile phone, if everything is normal, you can find the WIFI Hotspot with the same name (**AndroidAP_4174**) shown below **Hotspot name** in the above picture in the WI-FI search list. Then you can tap AndroidAP_4174 to connect to the Hotspot, the password can be seen under **Hotspot password** in the image above

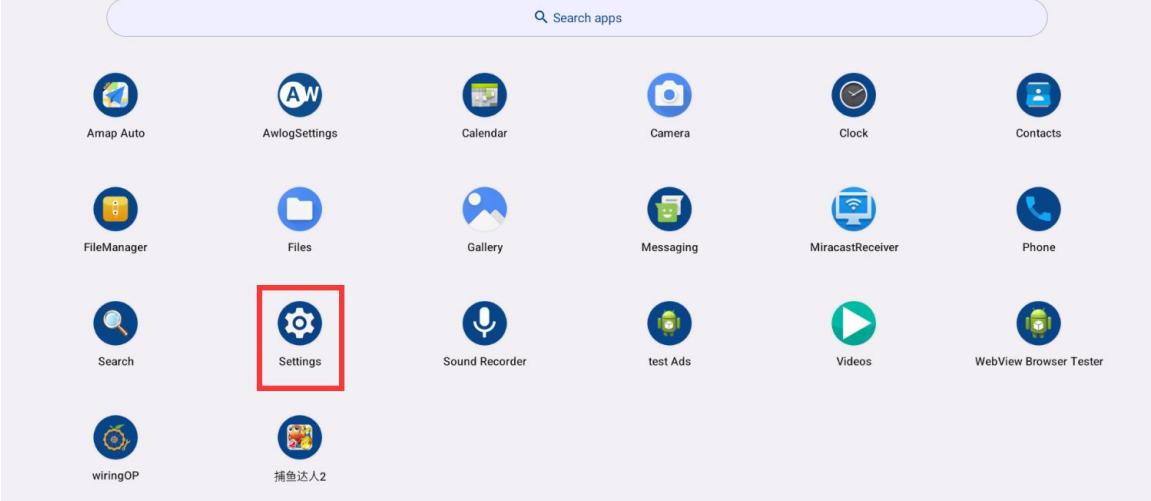
8) After successful connection, it will be displayed as shown in the following figure (different mobile phone interfaces will be different, the specific interface is subject to the display of your mobile phone). At this time, you can open a web page on the phone to see whether the Internet can be accessed. If the web page can be opened normally, it indicates that the **WI-FI Hotspot** of the development board can be used normally



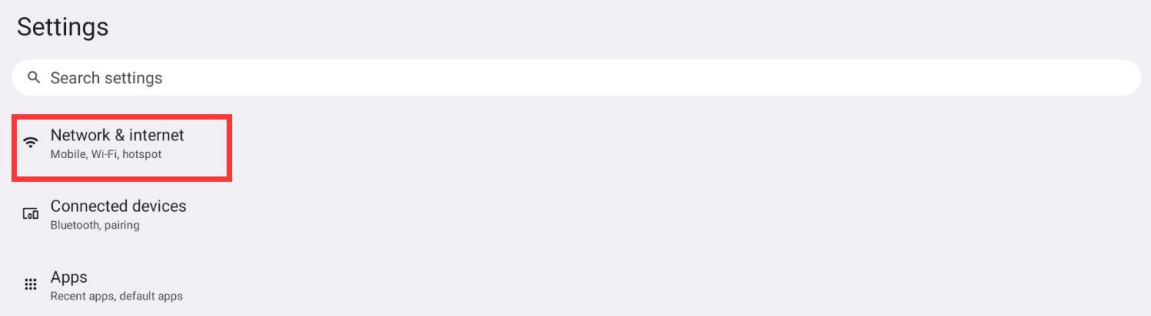
5. 7. Method to view Ethernet port IP address

1) First, ensure that the gigabit network port of the development board is connected to a router or switch

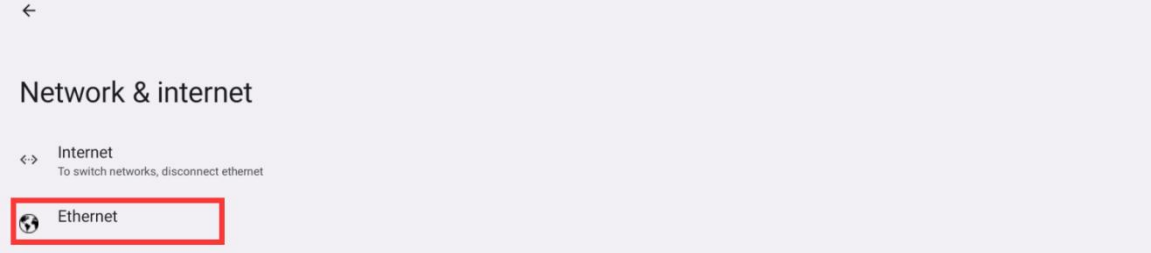
2) First open **Settings**



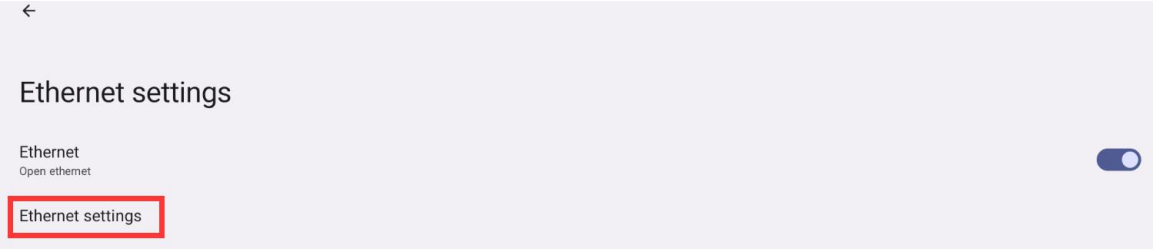
3) Then select **Network & Internet**



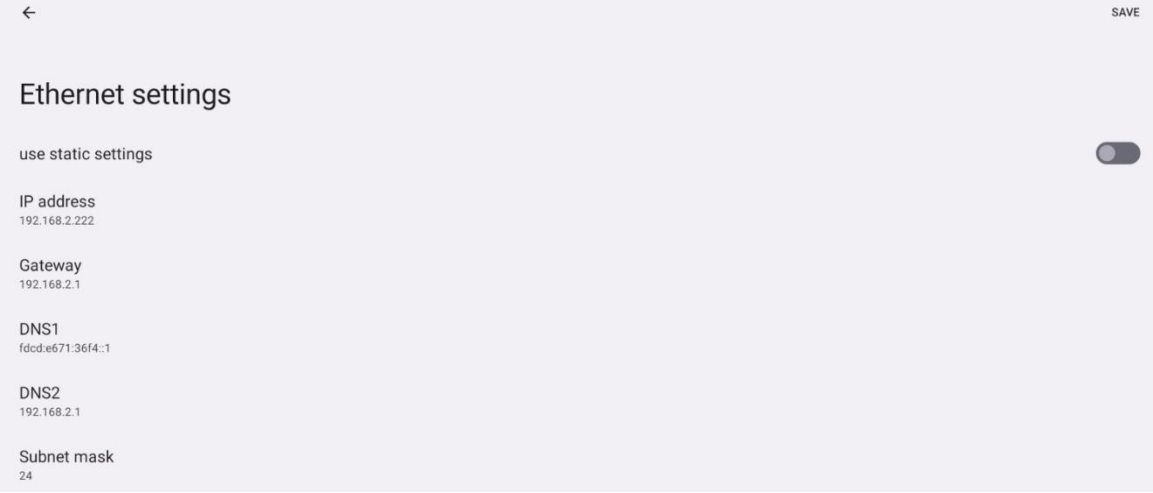
4) Then select **Ethernet**



5) Then select **Ethernet settings**

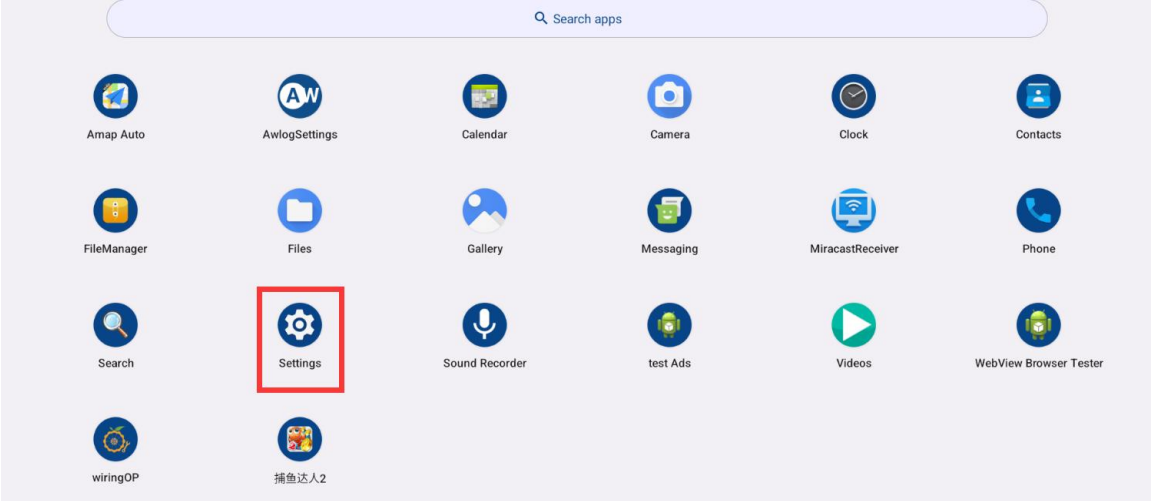


6) Then you can see the IP address information of the wired network port of the development board

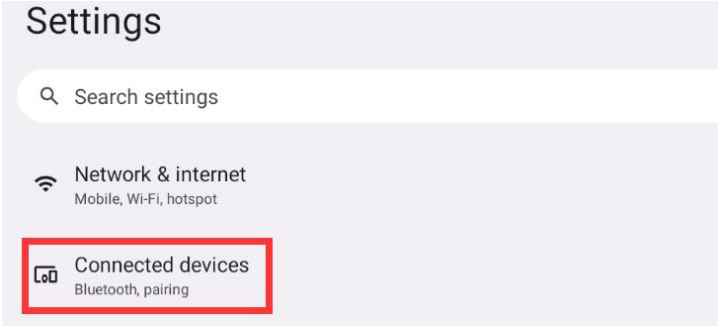


5. 8. Bluetooth connection method

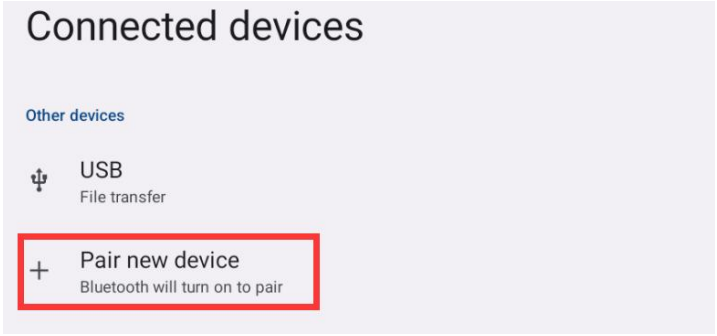
1) First select **Settings**



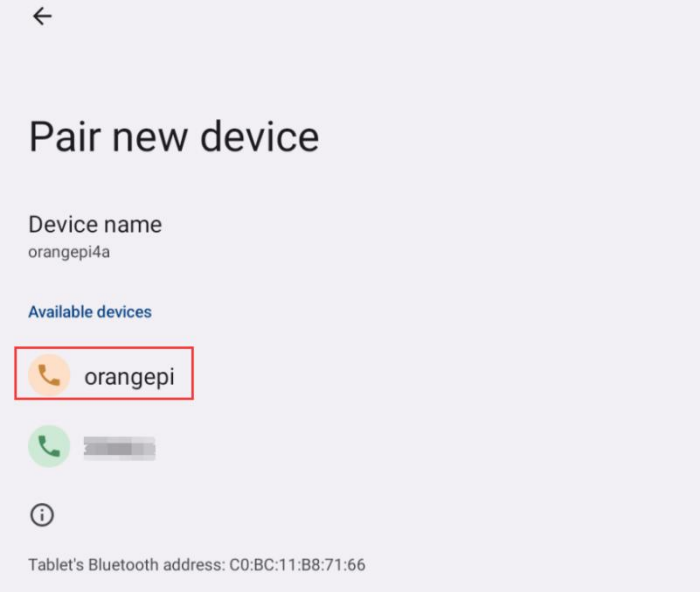
2) Then select **Connected devices**



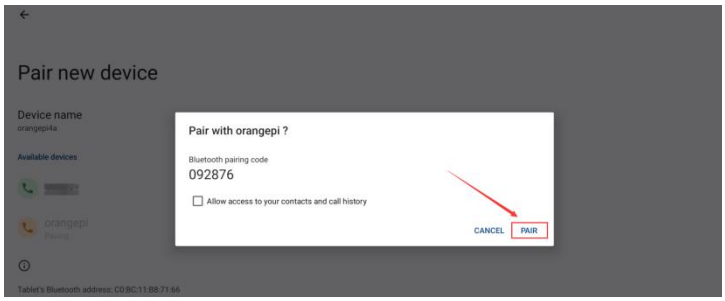
3) Then select **Pair new device** to start scanning the surrounding Bluetooth devices



4) The searched Bluetooth device will be displayed under **Available devices**

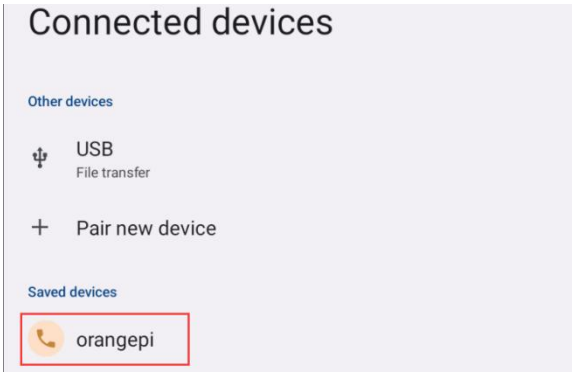


5) Then click on the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option

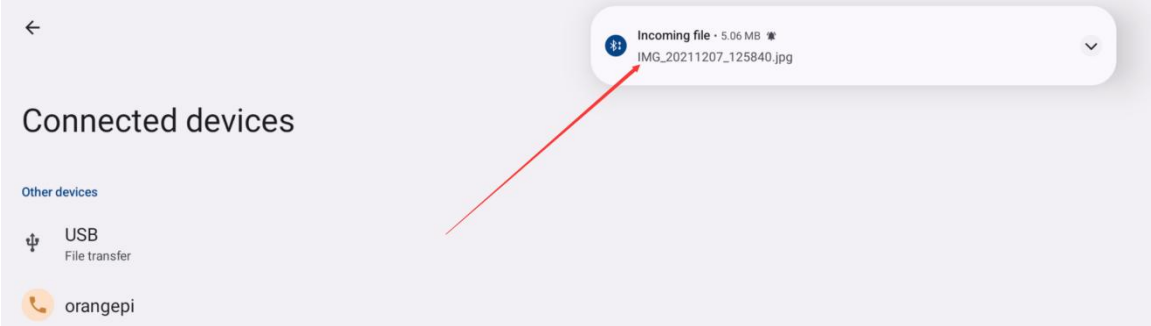


6) The test here is the configuration process of the development board and Bluetooth of **Android phones**. At this time, the confirmation interface will pop up on the phone, and the pairing process will start after clicking the pairing button on the phone

7) After pairing, open **Paired devices** and you will see the paired Bluetooth devices

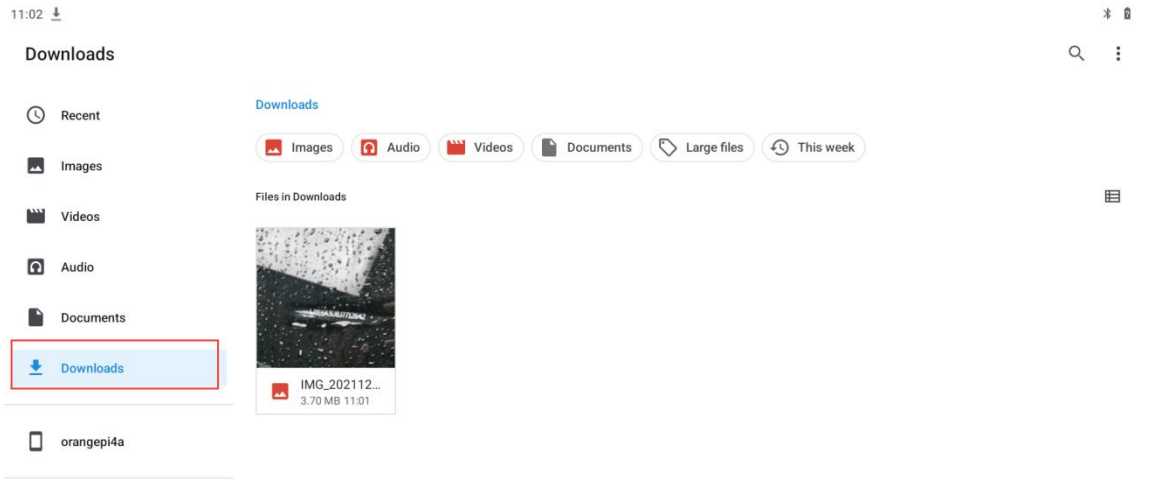


8) At this time, you can send a picture to the development board using the Bluetooth of your phone. After sending, you can see the following prompt in the Android system of the development board, and then click **Incoming file**



9) Then click **Accept** in the pop-up window to start receiving pictures sent by the phone

10) The picture received by the Bluetooth of the Android system can be viewed by opening the **Download** directory of the file manager

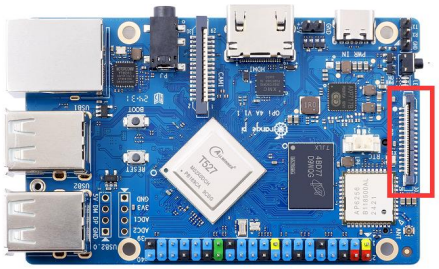


5. 9. 10.1 inch MIPI screen usage

> Make sure that the Android image you use is one of the following versions:
> OrangePi4A_T527_Android13_lcd_v1.x.x.img

1) The screen needs to be assembled first, please refer to the assembly method of 10.1 inch MIPI screen

2) The position of the interface of the mipi lcd screen on the development board is shown as follows:



3) Connect the assembled screen to the LCD interface of the development board, pay attention to unplug the HDMI interface, connect the Type-C power supply to the board, and power on, after the system starts, you can see the screen displayed as follows (**default is portrait screen**)

5. 10. How to use eDP screen

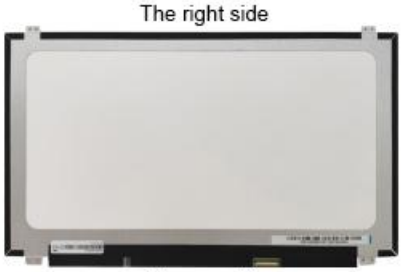
> **eDP screens are touch-free.**
> Make sure that the Android image you use is one of the following versions:
> OrangePi4A_T527_Android13_v1.x.x.img

3) Currently only one 15.6-inch eDP screen is available, including the following accessories:

c. 0.5 Spaced 30-pin single-head coaxial cables

d. 15.6-inch eDP display, resolution is 1920x1080.

The right side

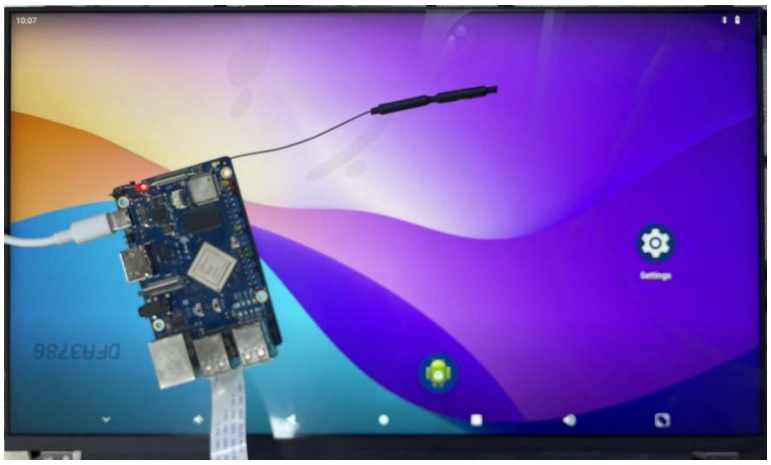


Reverse side

30pins EDP interface

4) Connect the FPC end of the 30pin single-head codirectional cable to the eDP interface of the development board, and connect the other end to the eDP interface of the screen

5) Then connect the Type-C power supply to the board and power it on. After the system is started, you can see the screen displayed as shown in the following figure



5. 11. How to use USB camera

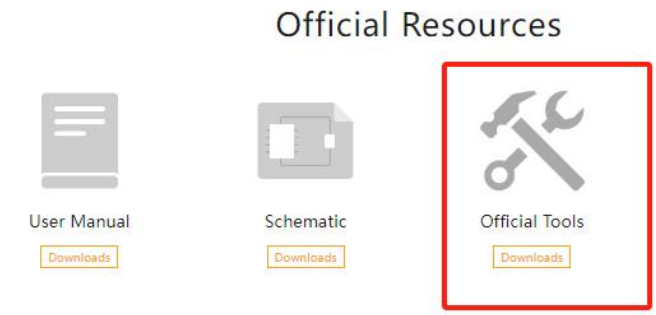
1) First, insert a USB (UVC protocol) camera into the USB interface of the development board

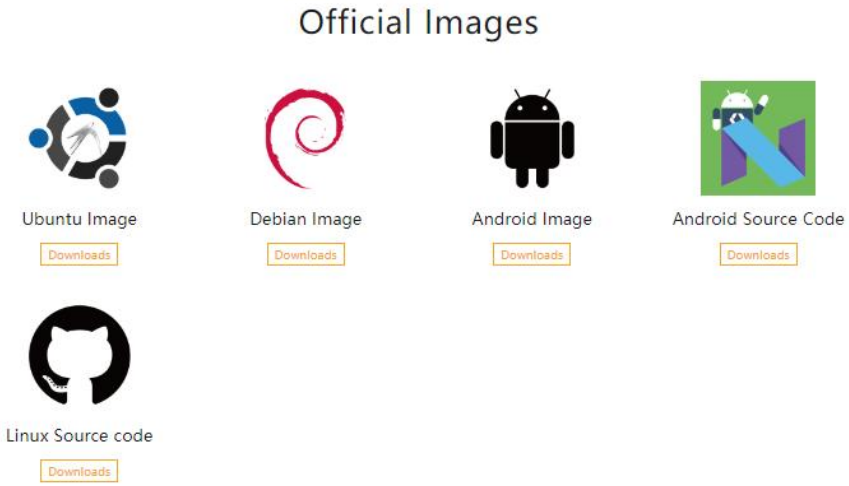
2) If the USB camera is properly identified, a video device node is generated under /dev

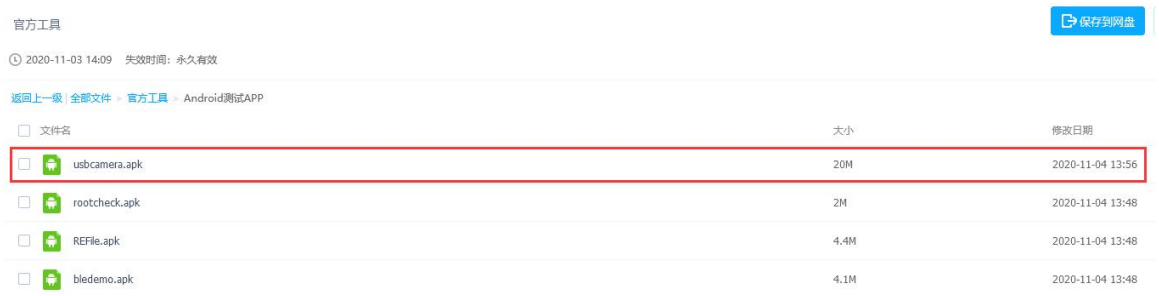
```
console:/ # ls /dev/video0
/dev/video0
```

3) Then ensure that the ADB connection between the Ubuntu PC and the development board is normal. Please refer to the instructions in the section on **adb Usage**

4) Download the USB camera test APP from the **official tool** on the development board data download page



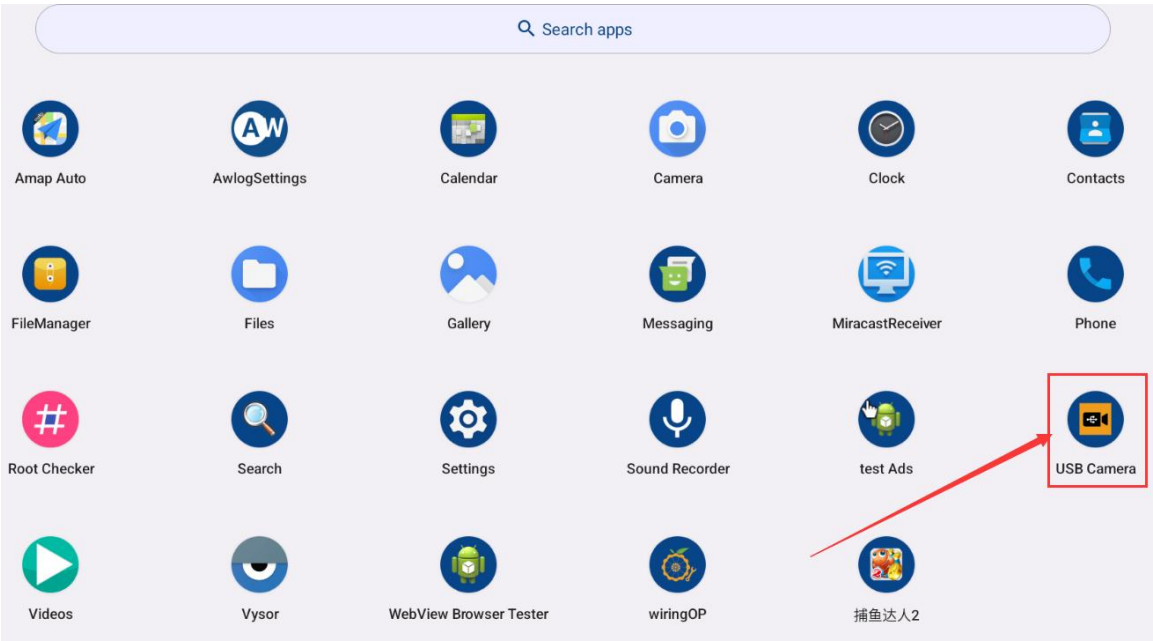




5) Then use adb command to install the USB camera test APP into the Android system, of course, you can also use the U disk copy for installation

```
test@test:~$ adb install usbcamera.apk
```

6) After installation, you can see the startup icon of USB camera on the Android APP interface



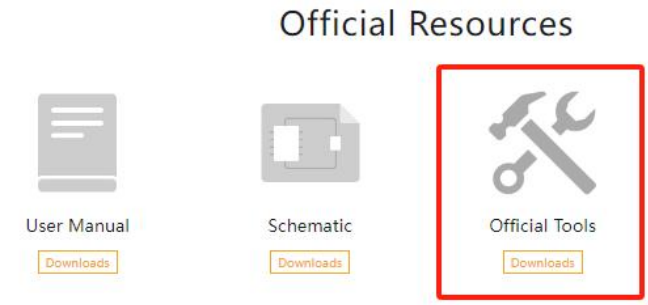
7) Then double-click on the USB camera APP and you can see the output video of the USB camera

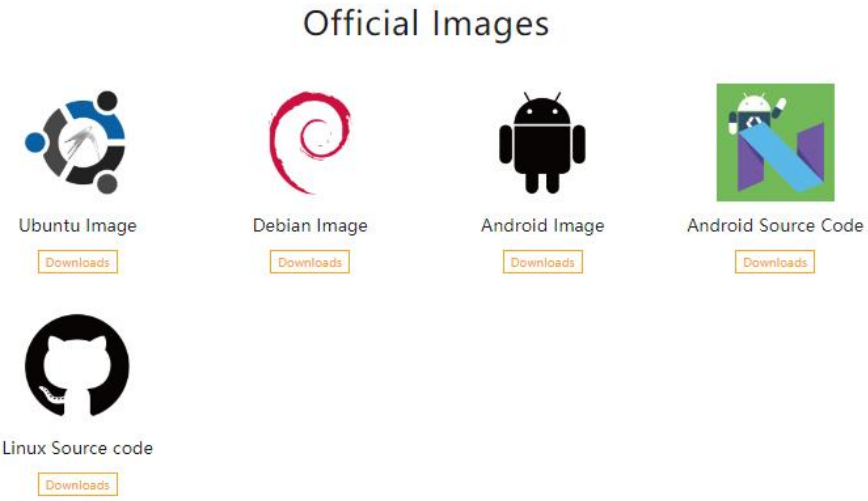
5. 12. Android system ROOT Description

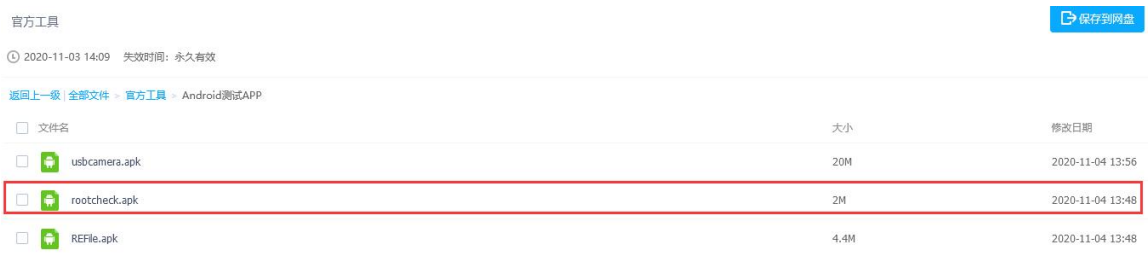
The Android system released by Orange Pi has been ROOT, you can use the

following method to test.

1) Download **rootcheck.apk** from the **official tool** on the development board data download page





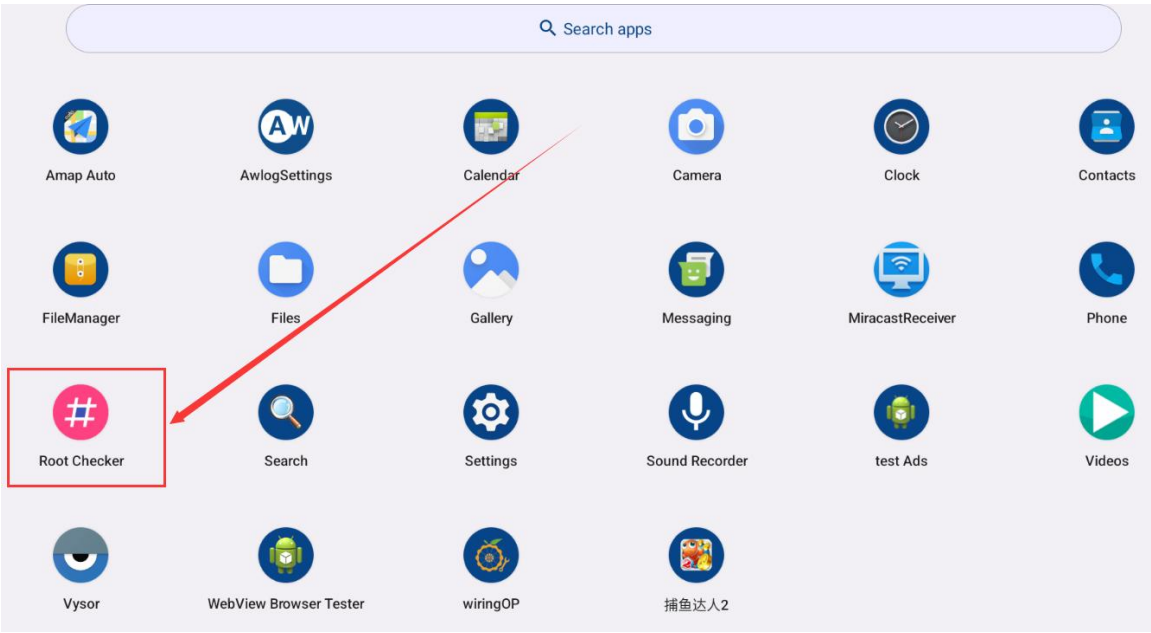


2) Then ensure that the ADB connection between the Ubuntu PC and the development board is normal. Please refer to the instructions in the section on **adb Usage**

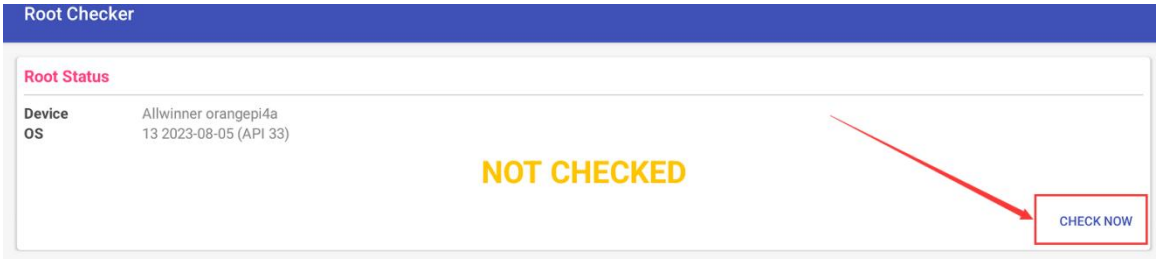
3) Then use adb command to install rootcheck.apk into the Android system, of course, you can also use the U disk copy for installation

```
test@test:~$ adb install rootcheck.apk
```

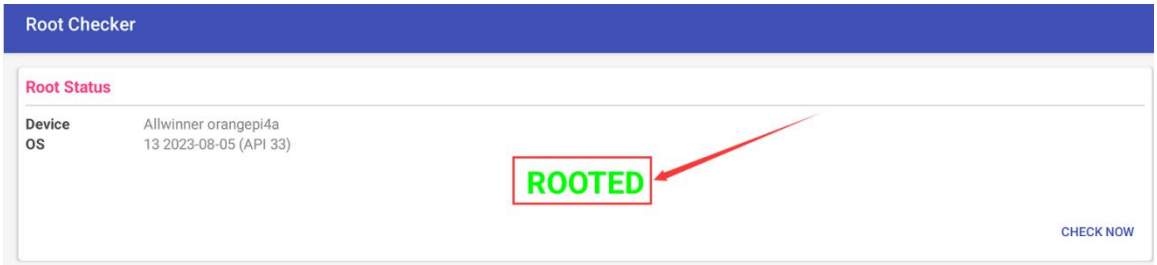
4) After installation, you can see the startup icon of the ROOT test tool on the Android APP interface



5) After opening the **ROOT test tool** for the first time, the display interface is as shown in the following figure



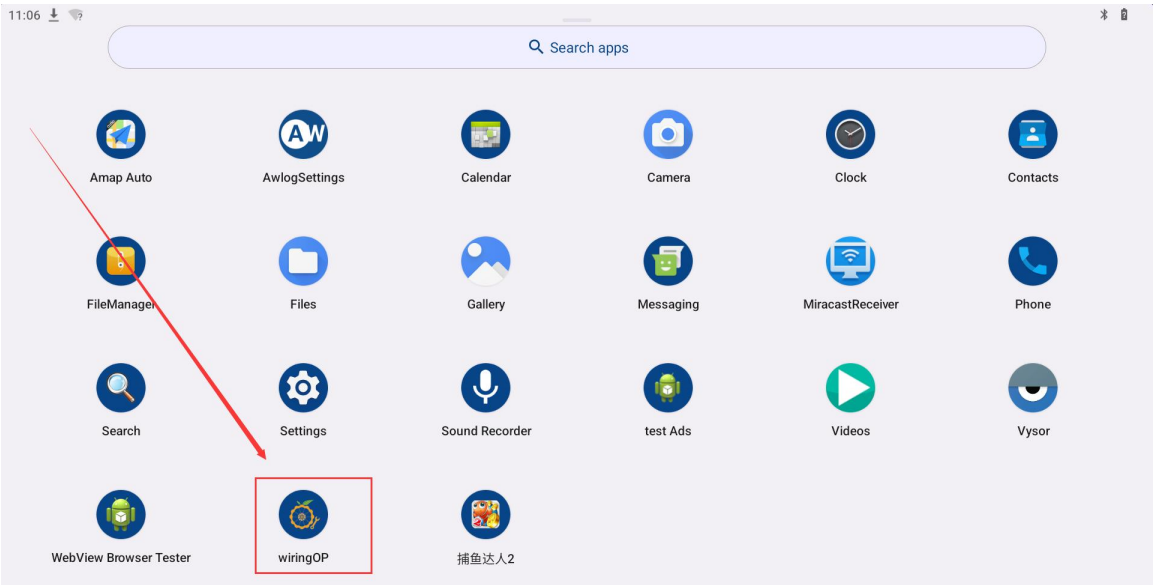
6) Then you can click **CHECK NOW** to start the ROOT status check of the Android system. After the check, the display is as follows, you can see that the Android system has obtained the ROOT permission



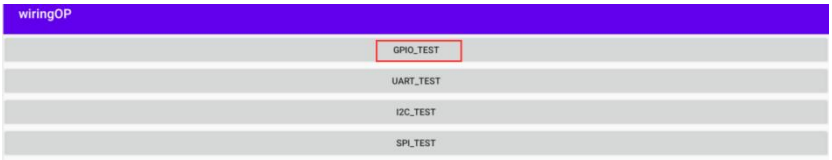
5. 13. 40 pin interface GPIO, UART, SPI test

5. 13. 1. 40 pin GPIO port test method

1) First open the wiringOP APP on your desktop



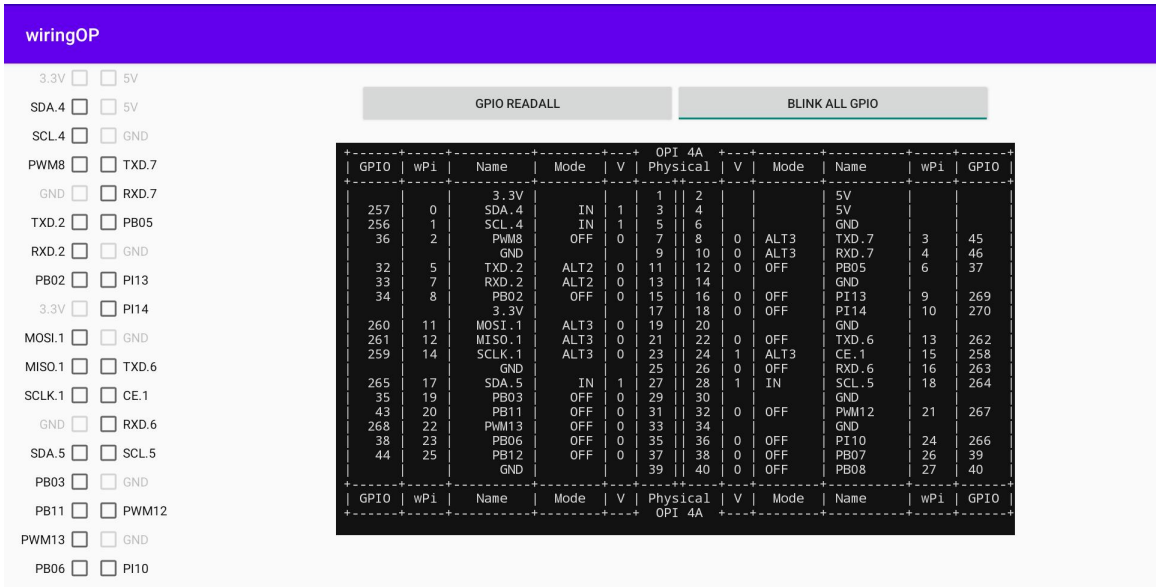
2) Then click **GPIO_TEST** button to open the GPIO test interface



3) The GPIO test interface is shown in the figure below. The two rows of **CheckBox** buttons on the left are one-to-one corresponding to the 40-pin pin. When **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT**mode and the pin level will be set to high. When unchecked, the GPIO pin level is set to low; When you click the

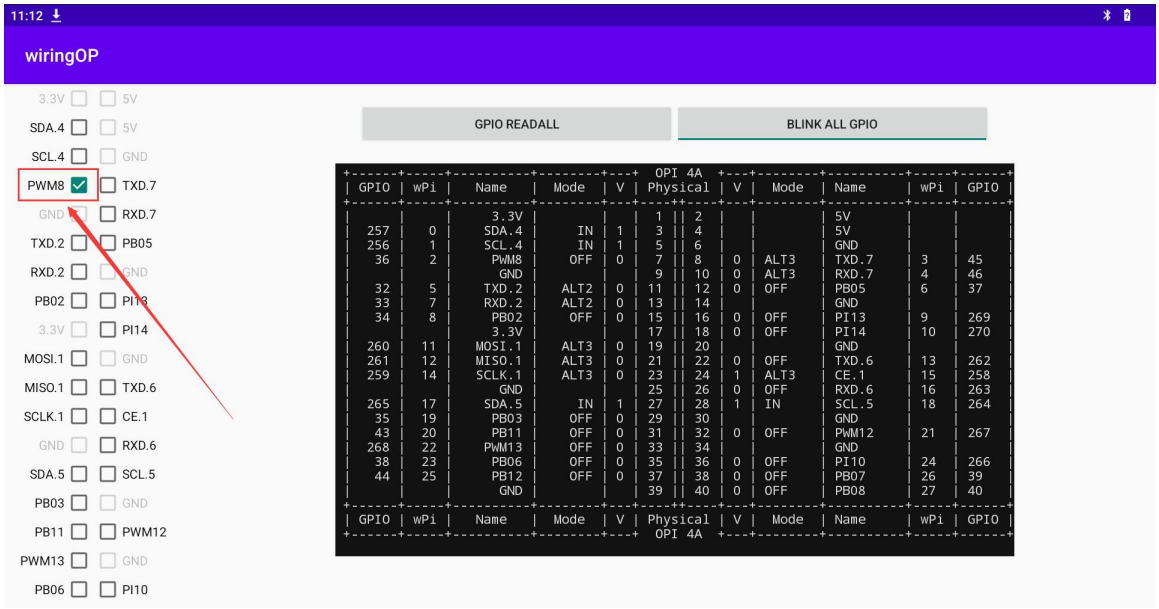
GPIO READALL button on the right, you can get the wPi number, GPIO mode, and pin parity information. When the **BLINK ALL GPIO** button is clicked, all pins continuously switch between high and low levels

4) Then click the **GPIO READALL** button, and the output information is as shown below:

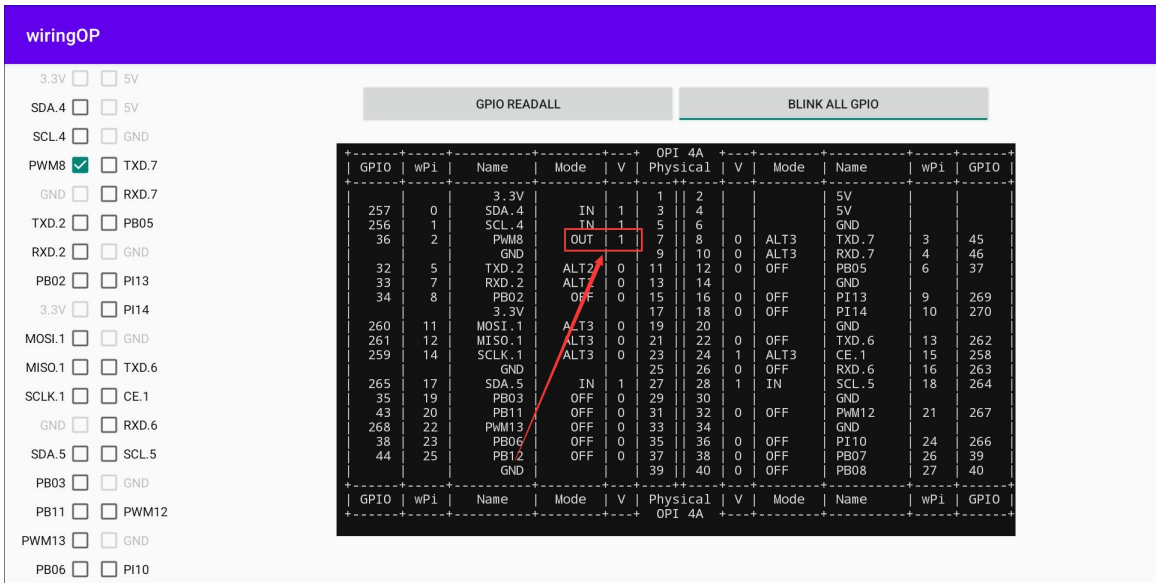


5) There are a total of 28 GPIO ports available in the 40 pin development board. Taking pin 7- corresponding to GPIO PB4- corresponding to wPi serial number 2- as an example, we will demonstrate how to set the high and low levels of GPIO ports. Firstly, click on

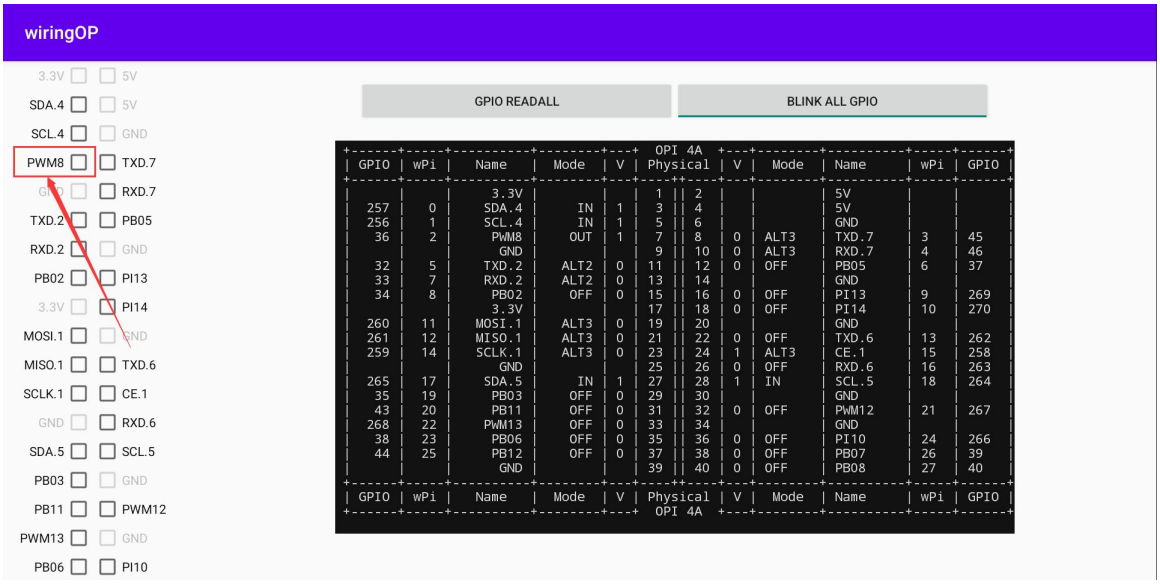
the **CheckBox** button corresponding to pin 7. When the button is selected, pin 7 will be set to a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **3.3v**, it means that the high level has been successfully set



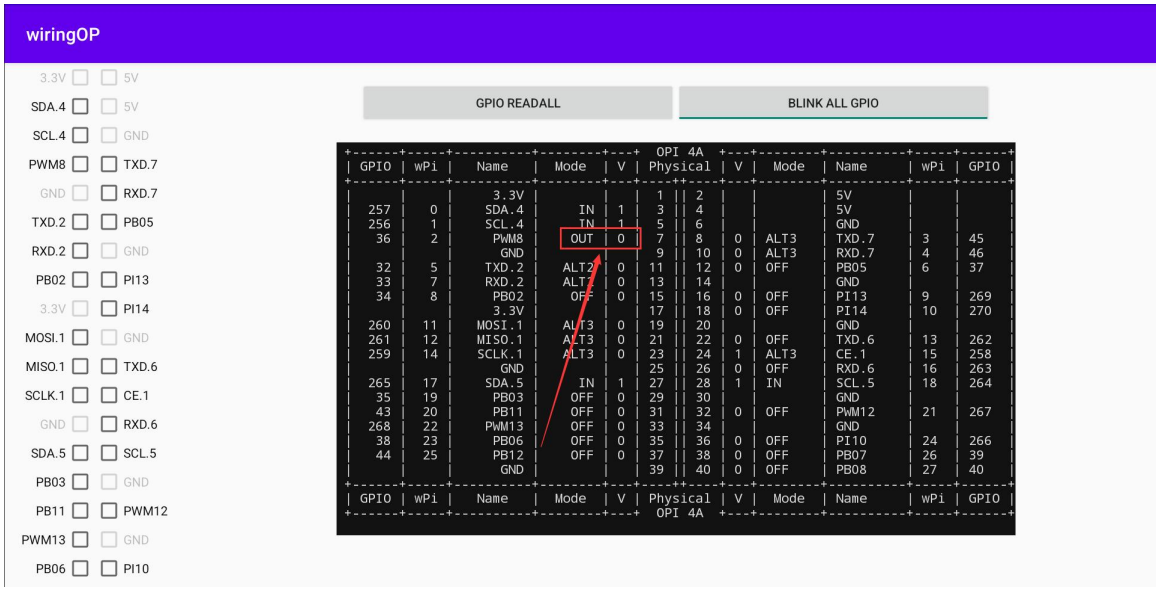
6) Then click the **GPIO READALL** button, and you can see that the current mode of pin 7 is **OUT** and the pin level is high



7) Click the **CheckBox** button in the following image again to uncheck the status. Pin 7 will be set to 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 has been successfully set



8) Then click the **GPIO READALL** button, and you can see that the current mode of pin 7 is OUT and the pin level is low

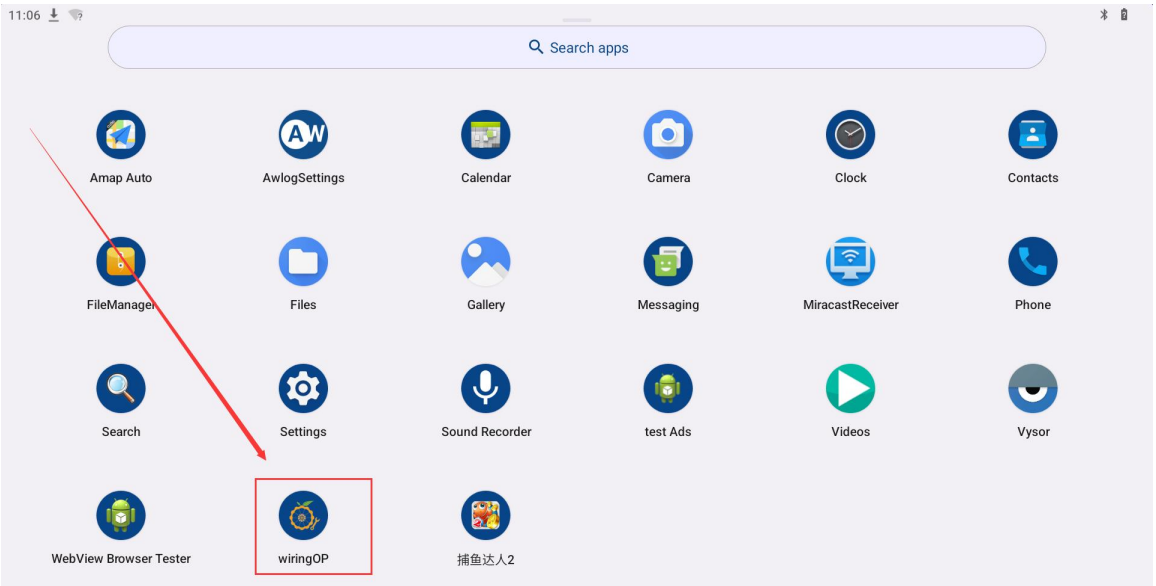


5. 13. 2. 40 pin UART testing method

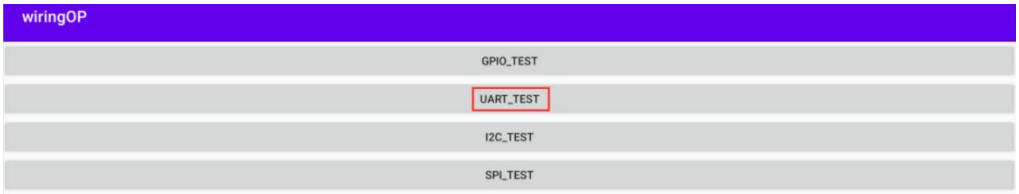
1) By default, **UART2** and **UART7** are enabled in Android, corresponding to device nodes**/dev/ttyAS2** and**/dev/ttyAS7**

```
t527-demo:/ $ ls /dev/ttyAS*
ttyAS0  ttyAS1  ttyAS2  ttyAS7
```

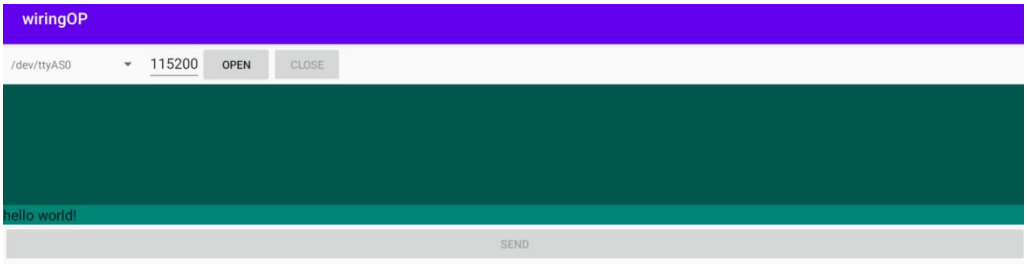
2) First, open the WiringoP app on the desktop



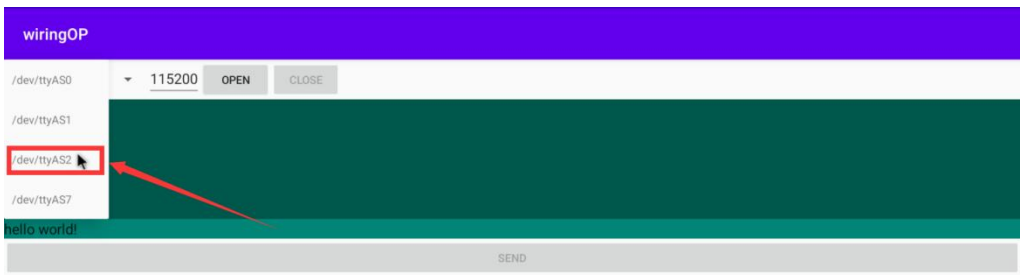
3) Then click the **UART_TEST** button to open the UART test interface



4) The serial port test interface of WiringOP is shown in the following figure



5) Taking testing UART2 as an example, select the**/dev/ttyAS2** node in the selection box

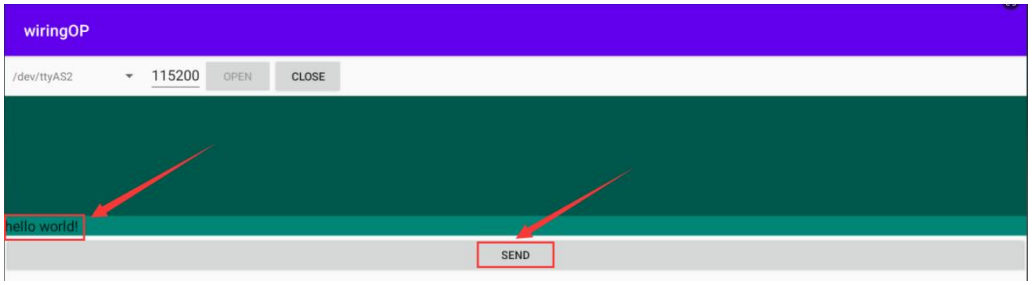


6) Enter the desired baud rate in the editing box, then click the **OPEN** button to open the **/dev/ttyAS2** node. After successful opening, the **OPEN** button becomes unselectable, and the **CLOSE** and **SEND** buttons become selectable

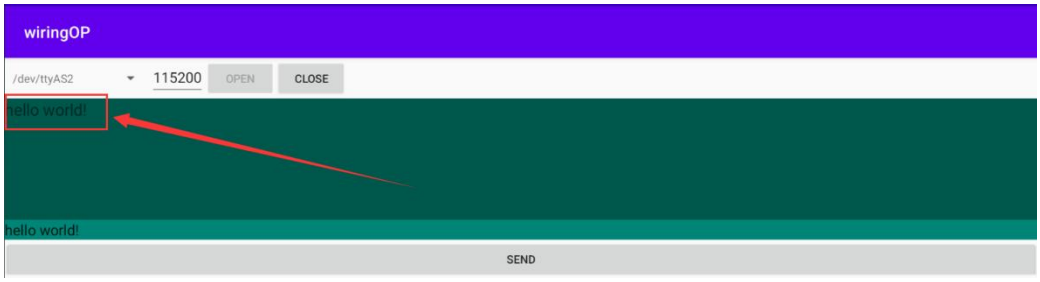
7) Then use DuPont wire to short-circuit the rx and tx pins of UART 2

	uart2
tx pin	Corresponding to pin 11 of pin 40
rx pin	Corresponding to pin 13 of pin 40

8) Then you can enter a character in the send edit box below and click the **SEND** button to start sending

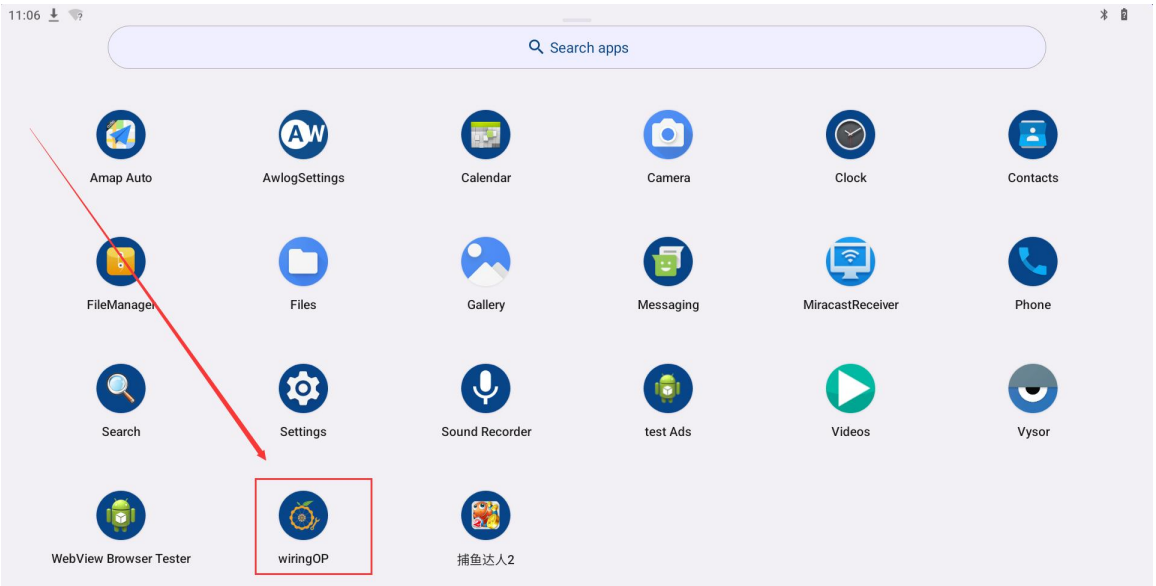


9) If everything is normal, the received string will be displayed in the receiving box



5. 13. 3. 40 pin SPI testing method

1) The SPI that can be used in 40 pins is SPI1, and the corresponding device node is**/dev/spidev1.0**

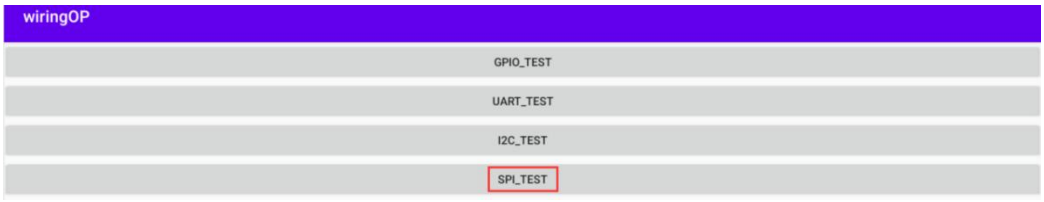


2) Here is a demonstration of testing the SPI1 interface using the **w25q64** module. First, connect the w25q64 module to the SPI1 interface

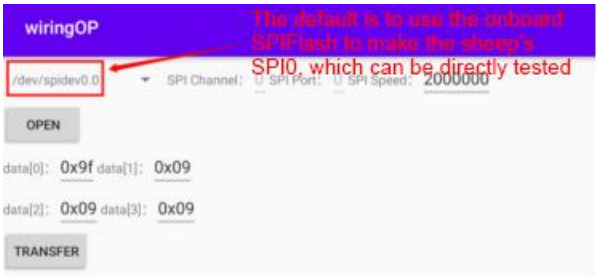
If there is no w25q64 module, it doesn't matter, because there is a SPIFlash connected to SPI0 on the development board, and the SPI0 configuration is also enabled by default in Android, so we can directly use the onboard SPIFlash for testing.

3) Then open the WiringoP app on the desktop

4) Then click the **SPI_TEST** button to open the SPI testing interface



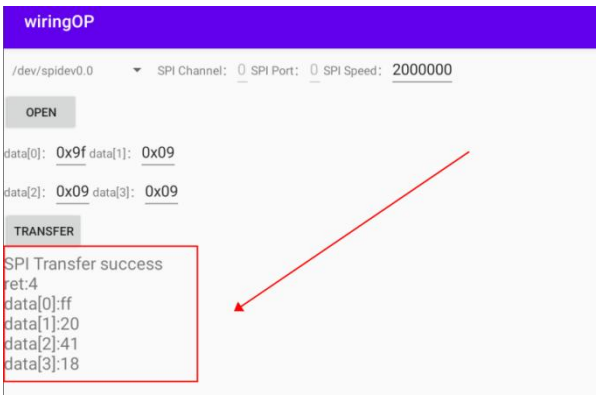
5) Then select the device node of SPI in the upper left corner. If testing the onboard SPIFlash directly, keep the default**/dev/spidev0.0**. If the **w25q64** module is connected to the 40 pin SPI1, select**/dev/spidev1.0**



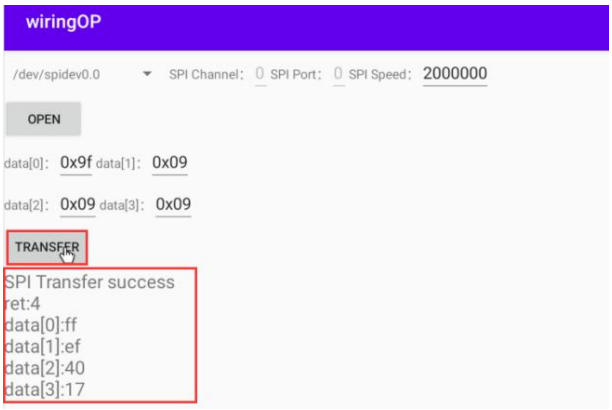
6) Then click the **OPEN** button to initialize SPI

7) Then fill in the bytes that need to be sent, such as reading the ID information of the onboard SPIFlash, filling in the address 0x9f in data [0], and then clicking the **TRANSFER** button

8) Finally, the APP will display the ID information of the read onboard SPI Flash



9) If reading the w25q64 module connected to the 40 pin SPI1, the read ID information is shown in the following figure



10) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, which corresponds to the values read above (h represents hexadecimal)

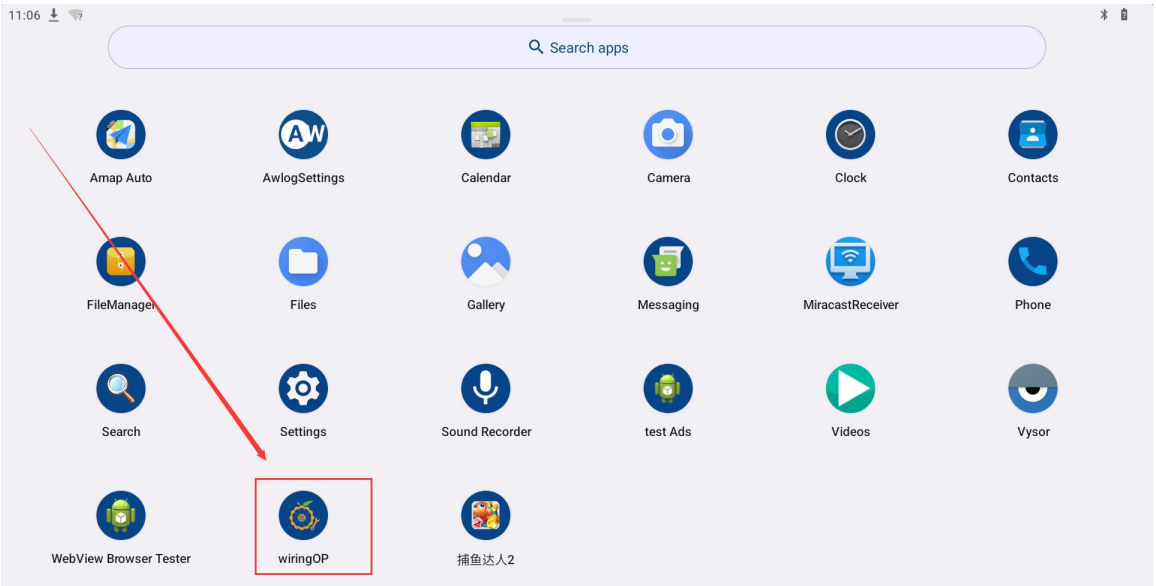
MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

5. 13. 4. 40 pin I2C testing method

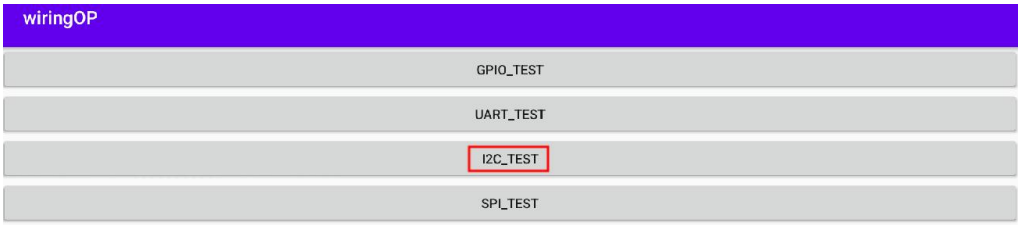
1) In Android, i2c4 and i2c5 in pin 40 are enabled by default, corresponding to device nodes**/dev/i2c-4** and**/dev/i2c-5**, respectively

```
console:/ # ls /dev/i2c-4 /dev/i2c-5
/dev/i2c-4    /dev/i2c-5
```

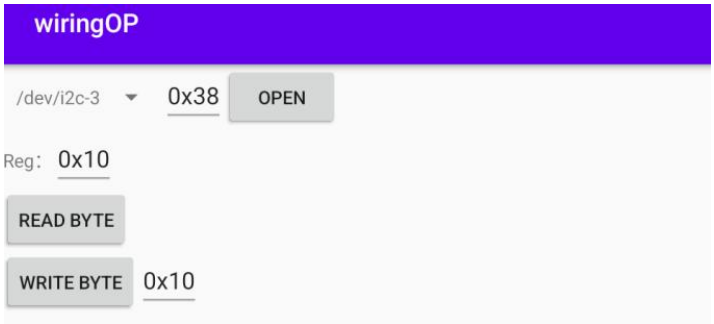
2) First, open the WiringoP app on the desktop



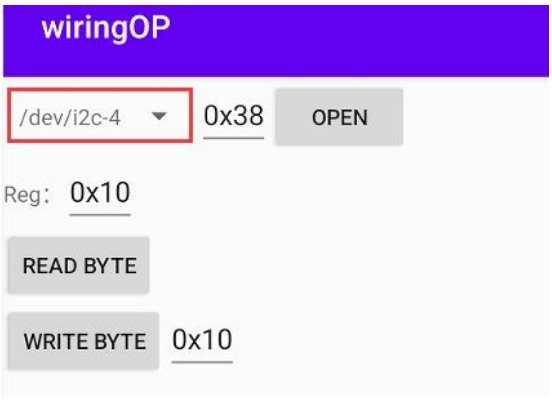
3) Then click the **I2C_TEST** button to open the testing interface of i2c



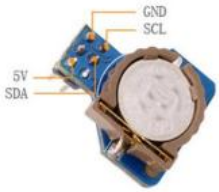
4) The i2c testing interface of WiringOP is shown in the following figure



5) Taking testing i2C4 as an example, select the**/dev/i2c-4 node** in the selection box



6) Then connect an I2C device to the 40 pin I2C4 pin, using the DS1307 RTC module as an example



Pin of RTC module	Development board 40 pin corresponding pins
5V	**Pin 2**
GND	**Pin 6**
SDA	**Pin 3**
SCL	**Pin 5**

7) The i2c address of the ds1307 RTC module is 0x68. After connecting the cable, we can use the **i2cdetect -y -r 4** command in the serial port command line to check if we can scan the i2c address of the ds1307 RTC module. As shown in the figure below, if the address 0x68 can be seen, it indicates that the wiring of the ds1307 RTC module is correct.

```
console:/ # i2cdetect -y 4
```

```
t527-demo:/ # i2cdetect -y -r 4
     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: -- -- -- -- -- -- -- --
t527-demo:/ #
```

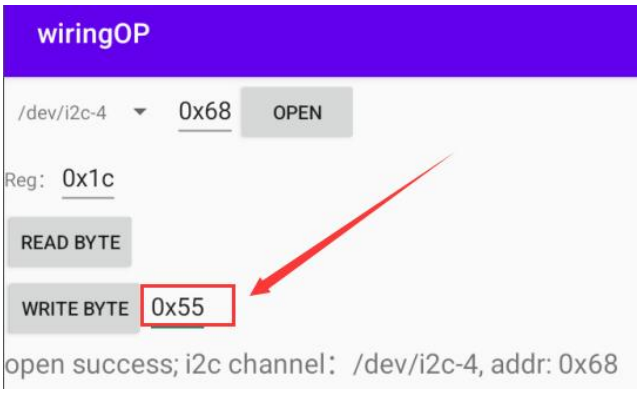
8) Then set the address of i2c to 0x68 in WirgOP, and click the **OPEN** button to open i2c4

9) After clicking the **OPEN** button to open i2C4, the display is as follows:

10) Then we will test writing a value to the register of the RTC module, such as writing 0x55 to address 0x1c

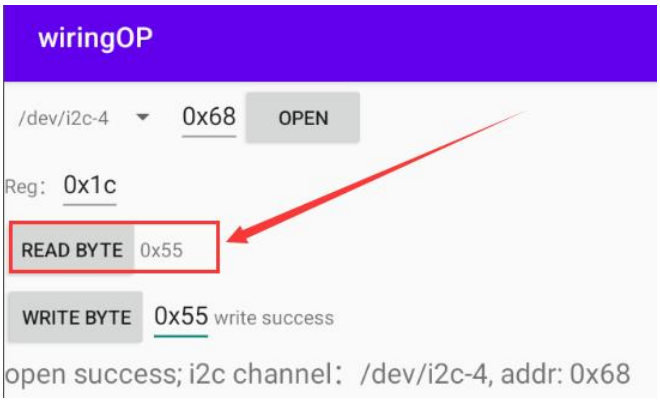
a. We first set the address of the register that needs to be written to 0x1c

b. Then set the value to be written as 0x55



c. Then click the **WRITE BYTE** button to perform the write action

11) Then click the **READ BYTE** button to read the value of the 0x1c register. If it displays 0x55, it means that the i2c read-write test has passed



6. Compilation method of Android 13 source code

6. 1. Download the source code of Android 13

1) Firstly, download the compressed Android 13 source code from Baidu or Google Drive

2) After downloading the compressed Android 13 source code, please check if the MD5 checksum is correct. If it is not correct, please download the source code again. The method for checking MD5 checksum is as follows:

```
test@test:~$ md5sum -c md5sum
t527_android13.tar.gz00: OK
t527_android13.tar.gz01: OK
.....
```

3) Execute the following command to extract the Android source code

```
test@test:~$ cat t527_android13.tar.gz0* | tar -xvzf -
```

6. 2. Compile the source code for Android 13

The compilation of Android 13 is performed on an x86_64 computer with Ubuntu 22.04 installed. There may be some differences in the dependencies of other versions of Ubuntu system packages. The download address for the Ubuntu 22.04 amd64 version image is as follows:
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.2-desktop-amd64.iso

The recommended hardware configuration for compiling Android 13 source code on an x86_64 computer is 16GB or more of memory, 200GB or more of hard disk space, and as many CPU cores as possible.

1) First, install and compile the software packages required for Android 13 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** sets compilation options

```
test@test:~$ cd t527_android13/longan
test@test:~/t527_android13/longan$ ./build.sh config
========ACTION List: mk_config ;========
options :
All available platform:
    0. android
    1. linux
Choice [android]: 0
All available ic:
    0. a523
    1. a527
```

```
    2. t527
Choice [t527]: 2
All available board:
    0. demo
    1. demo_car
    2. demo_fastboot
    3. demo_linux_aiot
    4. demo_linux_car
Choice [demo]: 0
All available flash:
    0. default
    1. nor
Choice [default]: 0
Setup BSP files
INFO: Prepare toolchain ...
INFO: longanforandroid project, clang_top_path: /home/test/a527_android13
INFO: kernel defconfig: generate /home/test/a527_android13/longan/out/t527/kernel/build/.config by
/home/test/a527_android13/longan/device/config/chips/t527/configs/default/linux-5.15/android13_arm64_defconfig
INFO: Prepare toolchain ...
make: Entering directory '/home/test/a527_android13/longan/kernel/linux-5.15'
make[1]: 进入目录“/home/test/a527_android13/longan/out/t527/kernel/build”
  GEN     Makefile
  HOSTCC  scripts/basic/fixdep
  HOSTLD  scripts/kconfig/conf
*** Default configuration is based on
'../../../../../device/config/chips/t527/configs/default/linux-5.15/android13_arm64_defconfig'
#
# configuration written to .config
#
make[1]: 离开目录“/home/test/a527_android13/longan/out/t527/kernel/build”
make: Leaving directory '/home/test/a527_android13/longan/kernel/linux-5.15'
INFO: clean buildserver
INFO: prepare_buildserver
```

b. Then run **./build.sh** script can start compiling now

```
test@test:~/t527_android13/longan$ ./build.sh
```

c. After compilation, you will see the following output

```
sun55iw3p1 compile all(Kernel+modules+boot.img) successful



INFO: build dts ...
INFO: Prepare toolchain ...
Setup BSP files
'/home/test/a527_android13/longan/out/t527/kernel/staging/sunxi.dtb' ->
'/home/test/a527_android13/longan/out/t527/demo/android/sunxi.dtb'
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: ----------------------------------------
INFO: build Tina OK.
INFO: ----------------------------------------
```

3) Then use the following command to compile the Android source code and generate the final Android image

```
test@test:~$ cd t527_android13
test@test:~/t527_android13$ source build/envsetup.sh
test@test:~/t527_android13$ lunch t527_demo_arm_go-userdebug
test@test:~/t527_android13$ make -j8
test@test:~/t527_android13$ pack
```

4) The storage path of the compiled Android image is:

```
longan/out/t527_android13_demo_uart0.img
```

7. Appendix

7.1. User Manual Update History

edition	Date	Update Explanation
v1.0	2024-11-14	Initial version
v1.1	2025-03-26	1. Compilation method of Android 13 source code 2. Method for burning Linux images to SPIFlash+NVMe SSD 3. Instructions for using the system backup script opi-bkimg 4. How to use Linux Overlayboot

7.2. Image update history

Date	Update Explanation
2024-11-14	OrangePi4A_T527_Android13_v1.0.0.tar.gz OrangePi4A_T527_Android13_lcd_v1.0.0.tar.gz Orangepi4a_1.0.0_ubuntu_jammy_server_linux5.15.147.7z Orangepi4a_1.0.0_debian_bookworm_server_linux5.15.147.7z Orangepi4a_1.0.0_ubuntu_jammy_desktop_gnome_linux5.15.147.7z Orangepi4a_1.0.0_debian_bookworm_desktop_gnome_linux5.15.147.7z *Initial version
2024-11-26	Orangepi4a_1.0.2_ubuntu_jammy_desktop_xfce_linux5.15.147.7z Orangepi4a_1.0.2_ubuntu_jammy_desktop_gnome_linux5.15.147.7z Orangepi4a_1.0.2_debian_bookworm_desktop_gnome_linux5.15.147.7z * Supports GPU acceleration and Gstreamer hard decoding, and can use the gst-play-1.0 command to hard decode and play videos

2025-03-26	Orangepi4a_1.0.4_ubuntu_jammy_server_linux5.15.147.7z Orangepi4a_1.0.4_debian_bookworm_server_linux5.15.147.7z Orangepi4a_1.0.4_ubuntu_jammy_desktop_xfce_linux5.15.147.7z Orangepi4a_1.0.4_ubuntu_jammy_desktop_gnome_linux5.15.147.7z Orangepi4a_1.0.4_debian_bookworm_desktop_gnome_linux5.15.147.7z * Kernel updated to Allwinner bsp v1.4 version * Switch boot mode from extLinux to boot.scr * Support SPIFlash+NVMe SSD startup * Add system backup script opi-bkimg * Support configuring overlay read-only file system through orangepi config