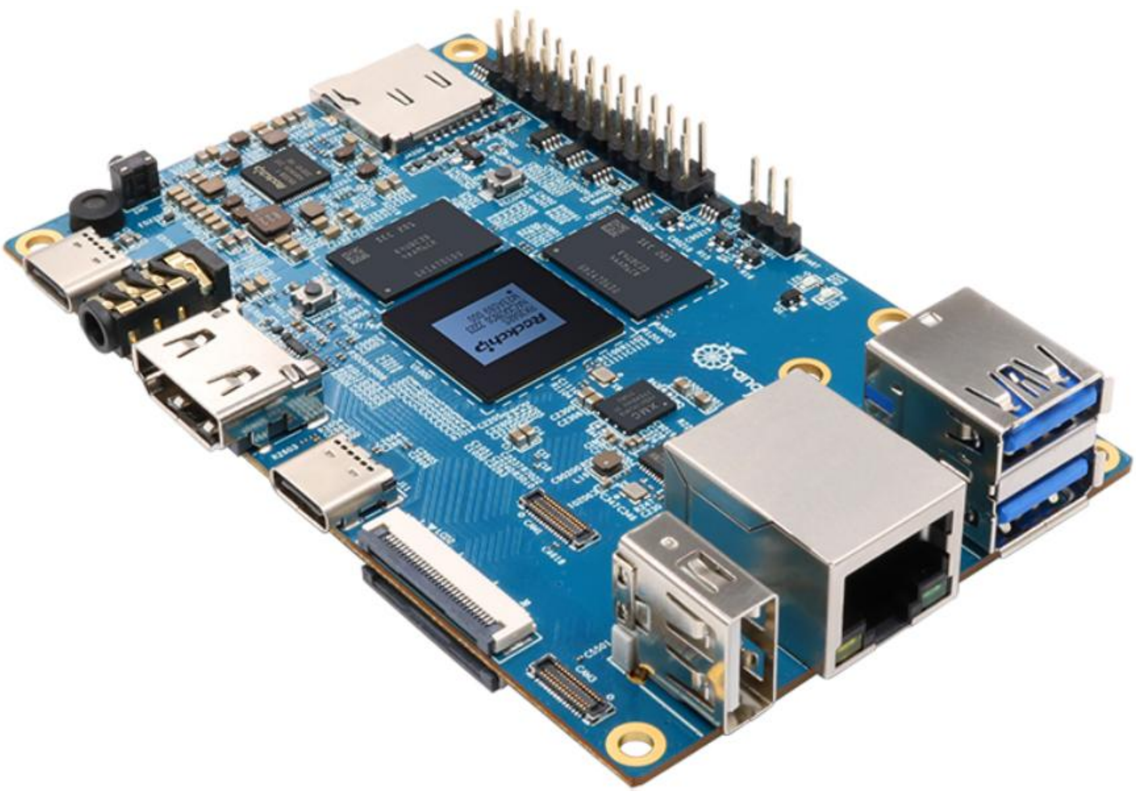
Orange Pi 5
User Manual



Contents

1. Basic features of Orange Pi 5

1. 1. What is Orange Pi 5

The Orange Pi 5 uses the new-generation Rockchip RK3588S ARM processor, which consists of quad-core A76 and quad-core A55. It features Samsung's 8nm LP process technology, a large-core main frequency of up to 2.4GHz, and an integrated ARM Mali-G610 MP4 GPU for high-performance 3D and 2D image acceleration. In addition, it comes embedded with an AI accelerator NPU that can handle up to 6 Tops of computing power. The device also has 4GB/8GB/16GB/32GB (LPDDR4/4x) memory and supports up to 8K display processing capabilities.

Orange Pi 5 offers a wide range of interfaces, such as HDMI output, Type-C, M.2 pcie 2.0, Gigabit Ethernet port, USB 2.0, USB 3.0 interface, and 26 pin expansion pin header. it can be used extensively in high-end tablets, edge computing, artificial intelligence, cloud computing, ar/vr, smart security, smart home, and other fields, covering various aiot industries.

Orange Pi 5 is compatible with several operating systems, including the official Orange Pi OS. additionally, it supports Android 12.1, Debian 11, Ubuntu 20.04, Ubuntu 22.04, and other systems.

1. 2. Uses of Orange Pi 5

We can use it to achieve：

- A linux desktop computer
- A linux network server
- An android tablet
- An android game console, etc.

Of course, there are many more functions available because orange pi 5 development board can install Linux systems such as Debian and ubuntu, as well as android. this means that within the range of hardware and software support provided by the development board, we can implement various types of functions.

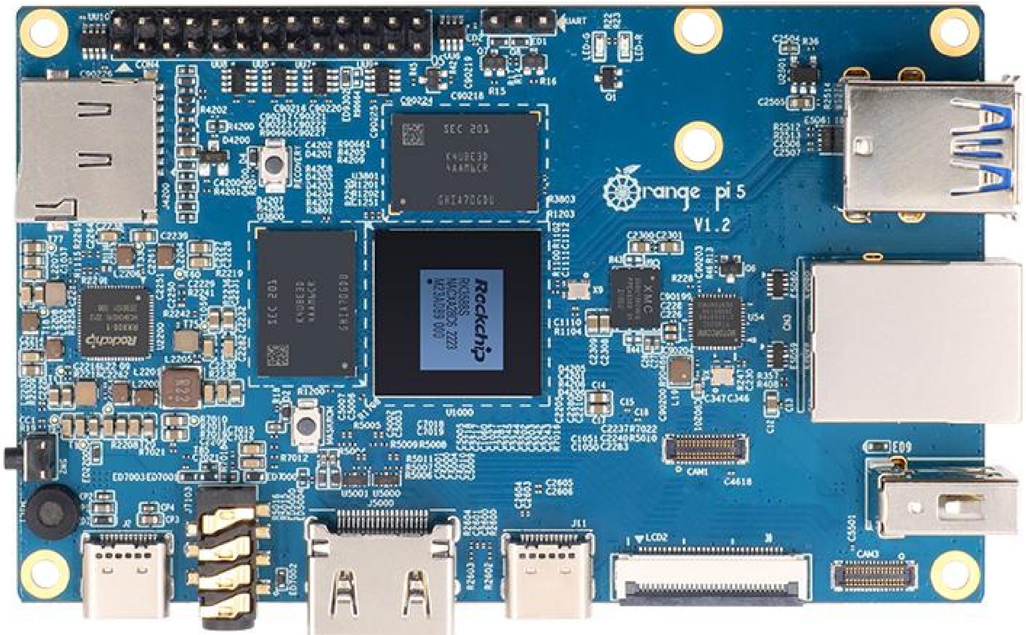
1. 3. Hardware features of Orange Pi 5

Introduction to hardware features	
CPU	• Rockchip RK3588S (8nm LP processor) • 8-core 64-bit processor • 4-core Cortex-A76 and 4-core Cortex-A55 core architecture • The main frequency of the large core is up to 2.4GHz, and the main frequency of the small core is up to 1.8GHz
GPU	•Integrated ARM Mali-G610 • OpenGL ES1.1/2.0/3.2, OpenCL 2.2 and Vulkan 1.2
NPU	• Built-in AI accelerator NPU with a computing power of up to 6 Tops • Support INT4/INT8/INT16 mixed operation
Video Output	• HDMI 2.1, up to 8K @60Hz • DP1.4（DisplayPort） • 2*MIPI D-PHY TX 4Lan
Memorry	4GB/8GB/16GB（LPDDR4/4x）
Camera	• 1 * MIPI CSI 4Lane • 2 * MIPI D-PHY RX 4Lane
PMU	RK806-1
Onboard Storage	• 16MB QSPI Nor FLASH • MicroSD (TF) Card slot • PCIe2.0x1 M.2 M-KEY (SSD) slot
Ethernet	10/100/1000Mbps ethernet (YT8531C)
Audio	• 3.5mm headphone jack audio in/out • Onboard MIC input • HDMI output
PCIe M.2 M-KEY	• Support PCIe WIFI6+BT5.0+BLE • Support SSD
USB Interface	1 * USB3.0 Interface 2 * USB2.0 Interface （ One of them is shared with the

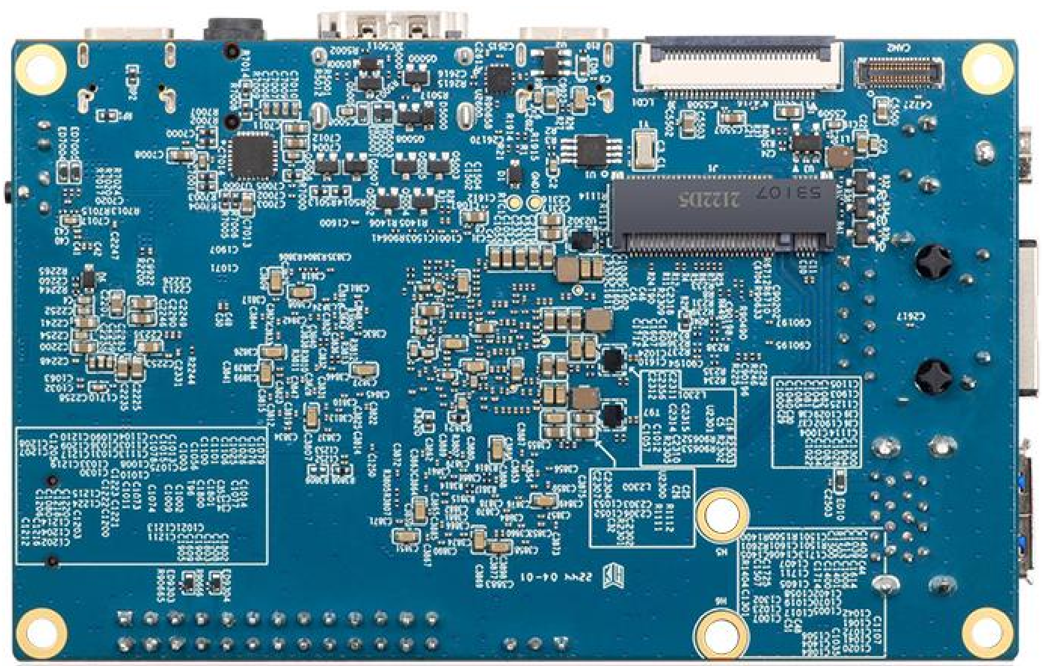
	Type-C interface） 1 * USB3.0 Type-C Interface
26pin Extension Header	Used to expand UART, PWM, I2C, SPI, CAN and GPIO interfaces
Debug Serial Port	3pin debug serial port
LED Light	Power light and status light
Button	1 * Mask ROM key, 1 * RECOVERY, 1 * switch key
Power Supply	5V/4A Type-C power supply
Supported OS	Orange Pi OS（Droid）、Orange Pi OS（Arch）、Android12.1、Debian11、Ubuntu20.04 and Ubuntu22.04 operating systems
Introduction of Appearance Specifications	
Product Size	100mm*62mm
Weight	46g
range Pi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd	

1.4. Top view and bottom view of Orange Pi 5

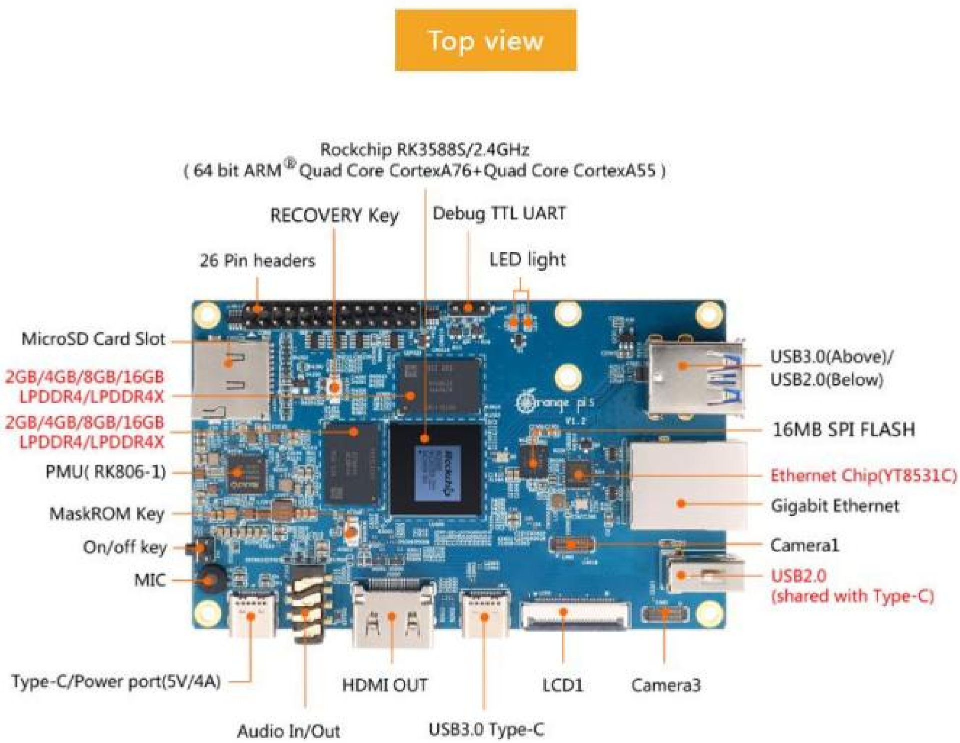
Top view:

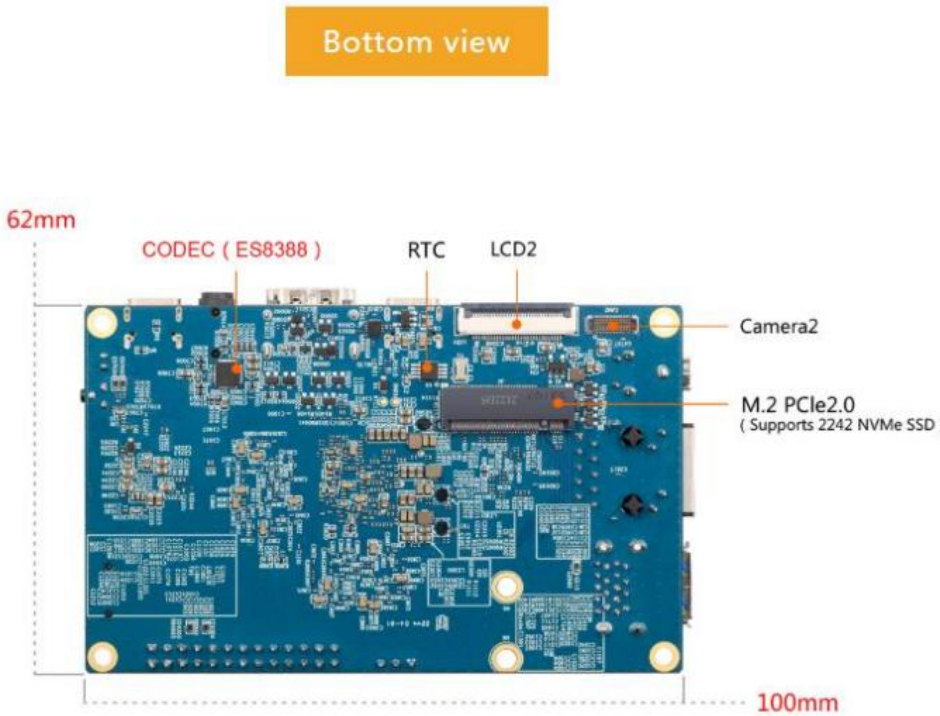


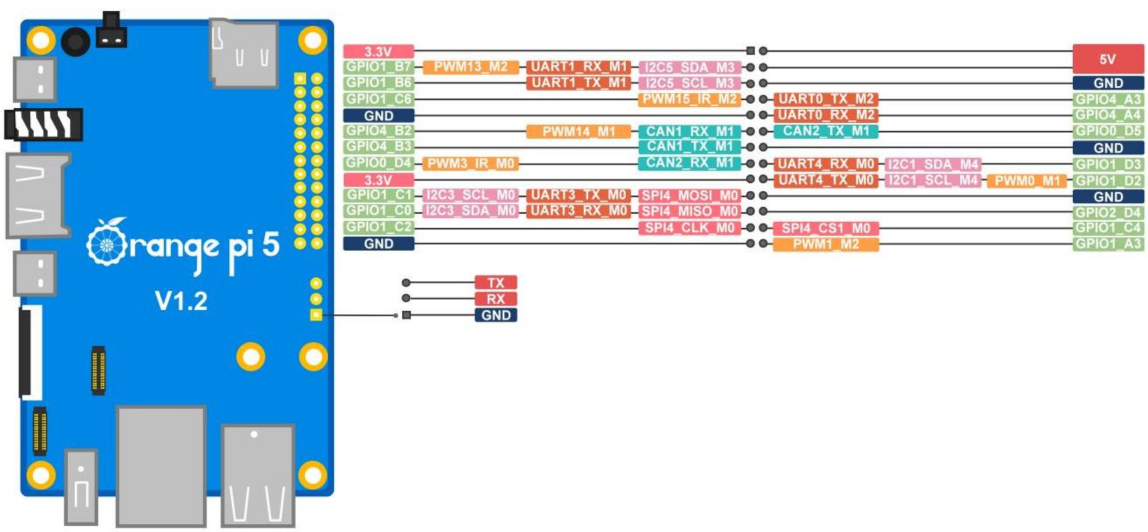
Bottom view:



1.5. Interface details of Orange Pi 5







The diameter of the four positioning holes is 3.0mm, and the diameter of the two M.2 PCIE device fixing holes is 3.5mm.

2. How to use the development board

2. 1. Prepare the required accessories

1) TF card, a class 10 or above high-speed SanDisk card with a minimum capacity of 8GB (32GB or above is recommended)

2) TF card reader, used to burn the image into the TF card

3) Display with HDMI interface

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

Note, if you want to connect a 4K or 8K display, please make sure that the HDMI cable supports 4K or 8K video output.

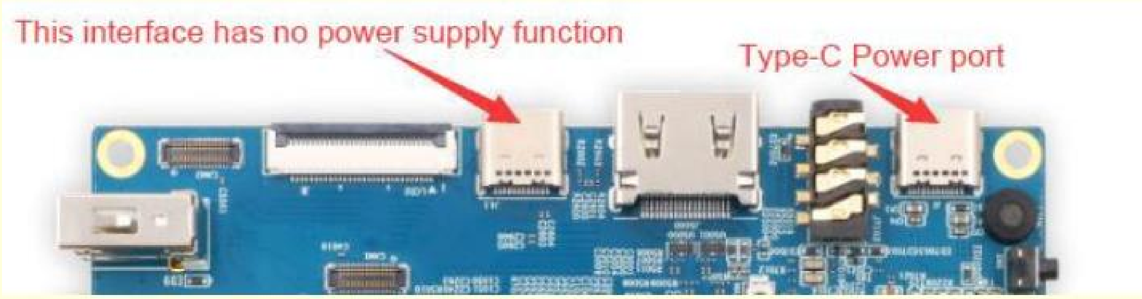
5) Type-C to HDMI cable, connect the development board to an HDMI monitor or TV for display through the Type-C interface

6) Type-C to USB adapter, used to connect USB storage devices or USB devices such as mouse and keyboard

7) 10.1-inch MIPI screen, used to display the system interface of the development board

8) Power adapter, Orange Pi 5 is recommended to use 5V/4A Type-C power supply for power supply

There are two Type-C ports that look the same on the development board. The one on the right is the power port, and the one in the middle has no power supply function. Please don't connect it wrong.



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

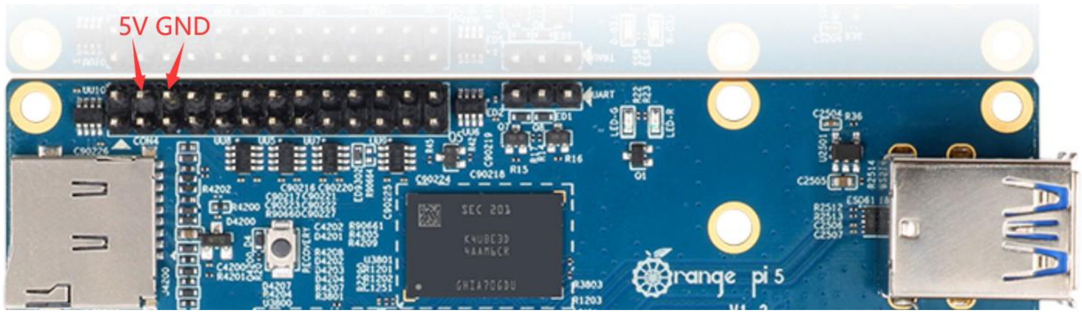
9) The mouse and keyboard of the USB interface, as long as the mouse and keyboard of the standard USB interface are acceptable, the mouse and keyboard can be used to control Orange Pi development board

10) USB Camera

11) 5V cooling fan. As shown in the figure below, the 5V and GND pins on the 26pin interface of the development board can be connected to the cooling fan. The spacing between the 26pin headers is **2.54mm**. The power interface of the cooling fan can be purchased according to this specification

Note that the 5V pin on the 26pin pin header can be used directly after the development board is plugged into the power supply of the Type-C interface. No other settings are required. In addition, the output voltage of the 5V pin on the

26pin pin header cannot be adjusted and turned off by software. (no PWM function).



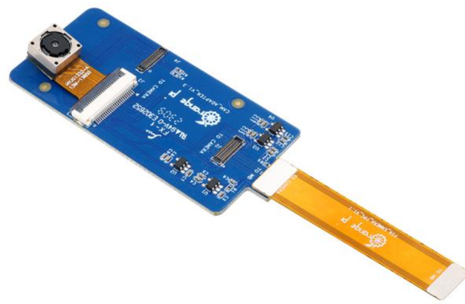
12) 100M or 1000M network cable, used to connect the development board to the Internet

13) The data cable of the Type-C interface, used to burn the image to NVMe SSD, use ADB and other functions

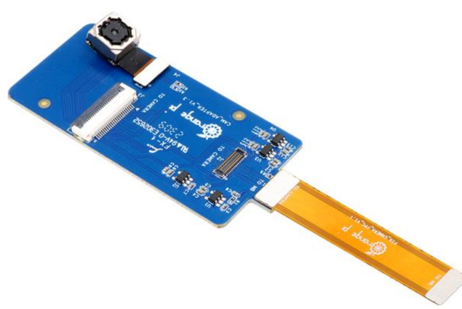
14) AP6275P PCIe WIFI6+Blustooth 5.0 2 in 1 module

Orange Pi 5

15) OV13850 camera with 13 million MIPI interface



16) OV13855 camera with 13 million MIPI interface



17) Matching shell (pictures and assembly methods to be added)

18) **3.3V** USB to TTL module and DuPont line, when using the serial port debugging function, need USB to TTL module and DuPont line to connect the development board and compute

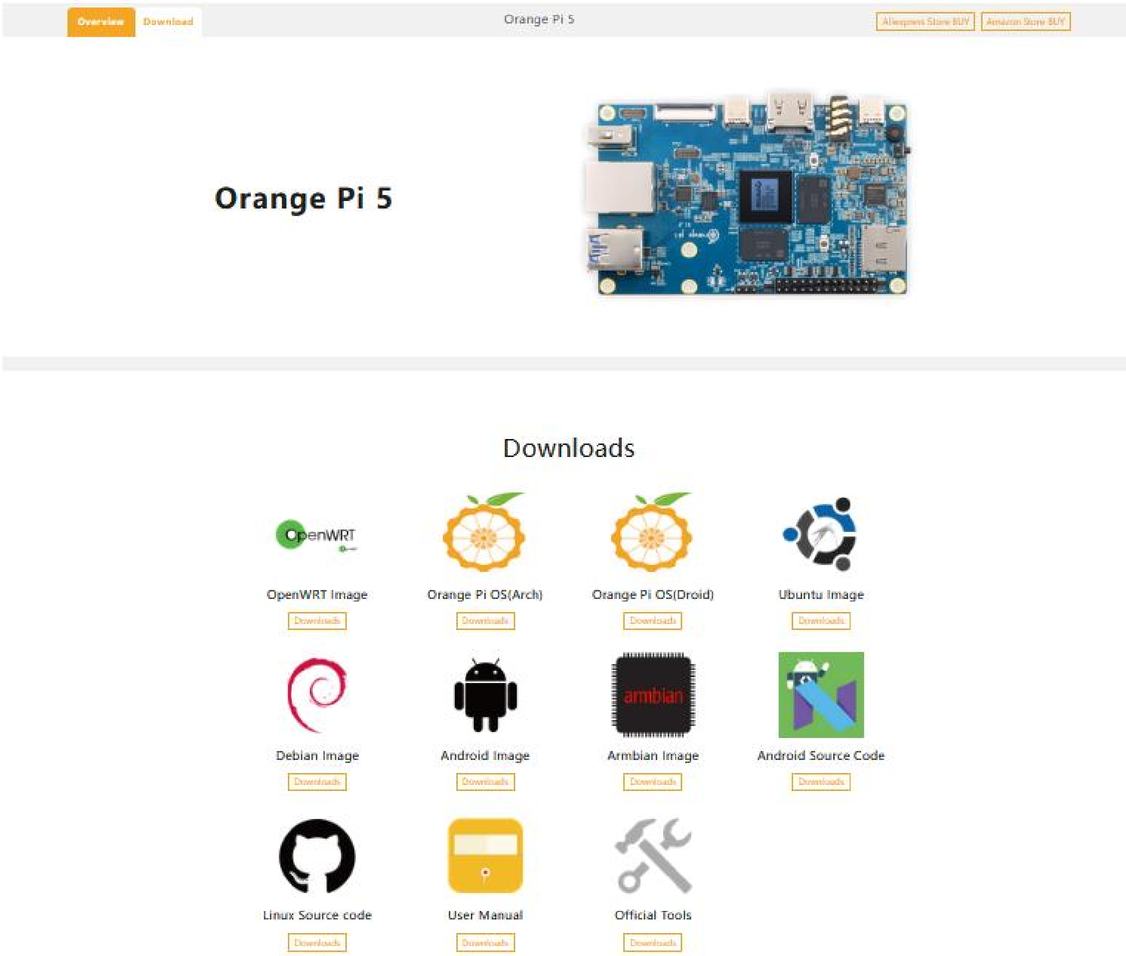
19) Personal computer with Ubuntu and Windows operating systems

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

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

1) The website for downloading the English version：

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-pi-5.html



2) The information mainly includes
   a. Android source code: saved on Google Drive
   b. Linux source code: saved on Github
   c. User manual and schematic diagram: saved on Google Drive
   d. Official tools: mainly include the software that needs to be used during the use of the development board
   e. Android image: saved on Google Drive
   f. **Ubuntu** image: saved on Google Drive
   g. **Debian** image: saved on Google Drive
   h. **Orange Pi OS** image: saved on Google Drive
   i. **OpenWRT** image: saved on Google Drive

2. 3. How to burn Linux image to TF card based on Windows PC

Note that the Linux image mentioned here specifically refers to the image of Linux distributions such as Debian, Ubuntu, OpenWRT or OPi OS Arch downloaded from the Orange Pi data download page.

Note that the OpenWRT image currently only supports TF card booting, and does not support SPIFlash+SSD booting.

Before the external TF card starts the OpenWRT image, if the U-boot is burned in the SPI Flash, please erase it first, otherwise the OpenWRT image in the TF card cannot be started. The command to erase U-boot in SPI Flash is:

orangepi@orangepi:~$ **sudo dd if=/dev/zero of=/dev/mtdblock0**

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

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

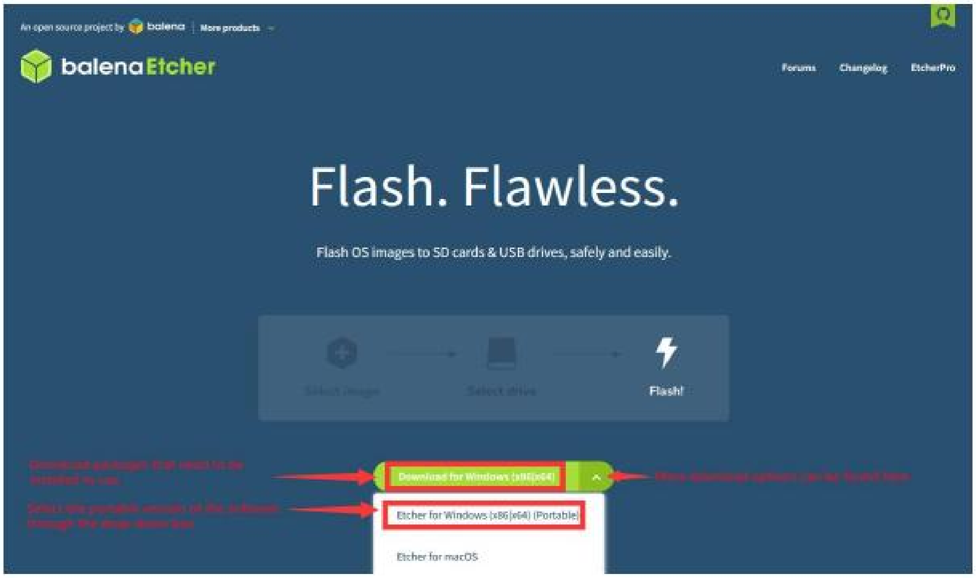
2) Then use the card reader to insert the TF card into the computer

3) Download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with ".img" is the image file of the operating system. The size is generally more than 2G

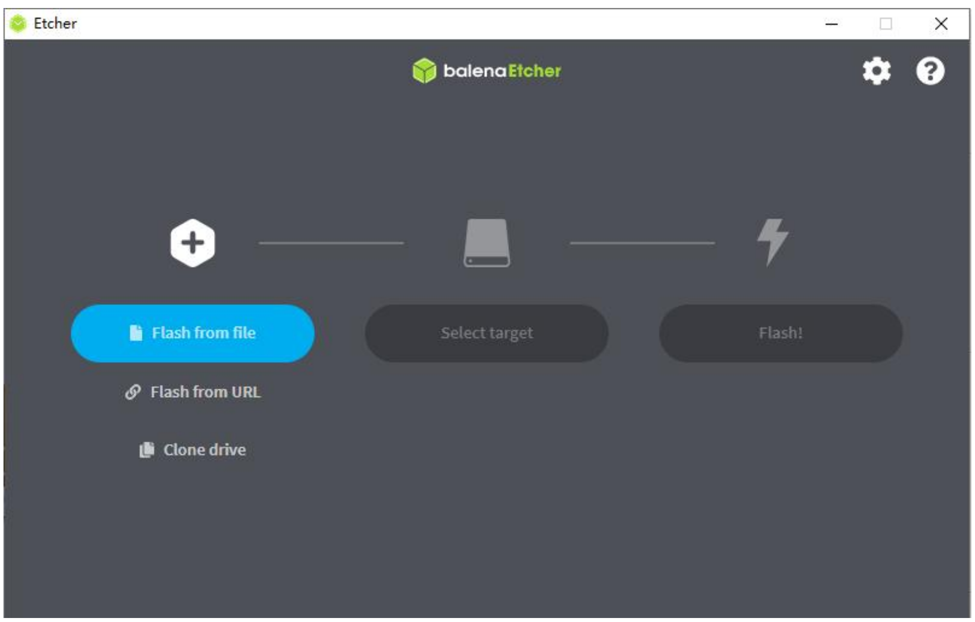
4) Then download the burning software of Linux image——**balenaEtcher**, the download address:

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

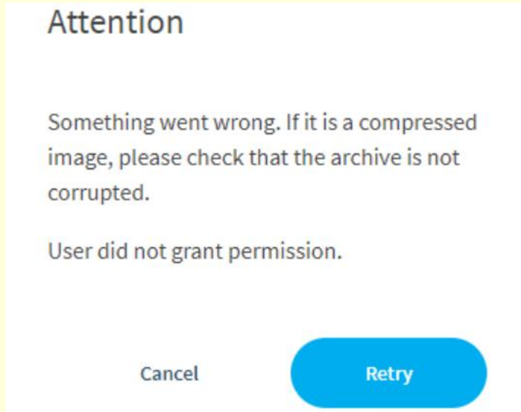
5) After entering the balenaEtcher download page, click the green download button to download the installation package of balenaEtcher. You can also select the Portable version of the balenaEtcher software through the drop-down box. The Portable version does not need to be installed, and it can be used by double-clicking to open it



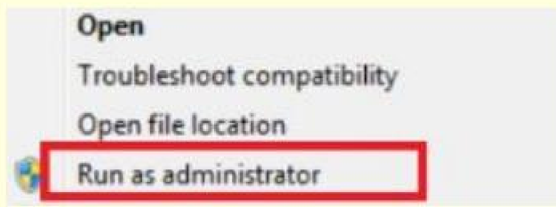
6) If the downloaded version of balenaEtcher needs to be installed, please install it before using it. If you downloaded the Portable version of balenaEtcher, just double-click to open it. The opened balenaEtcher interface is shown in the figure below



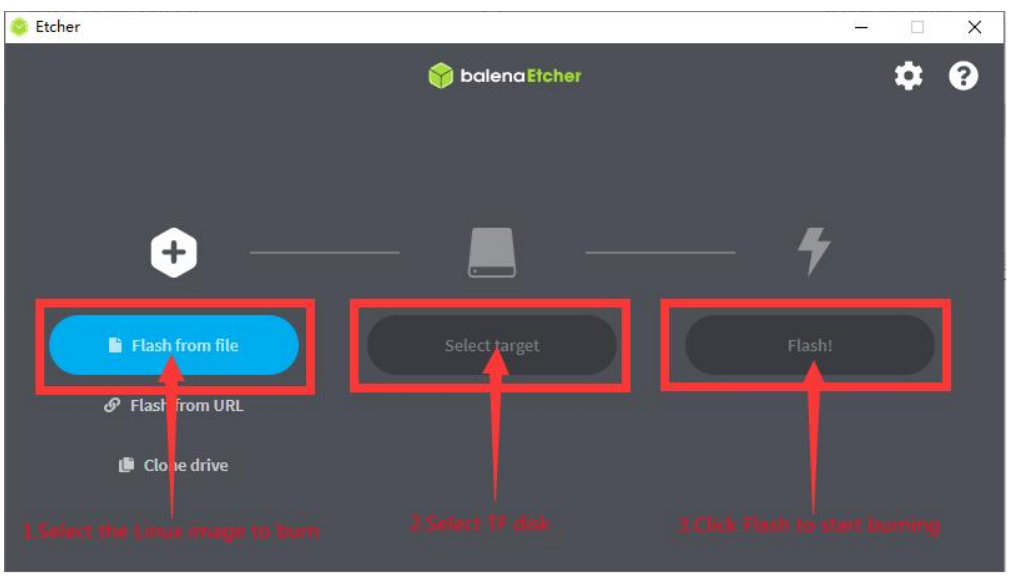
When opening balenaEtcher, if the following error is prompted:



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

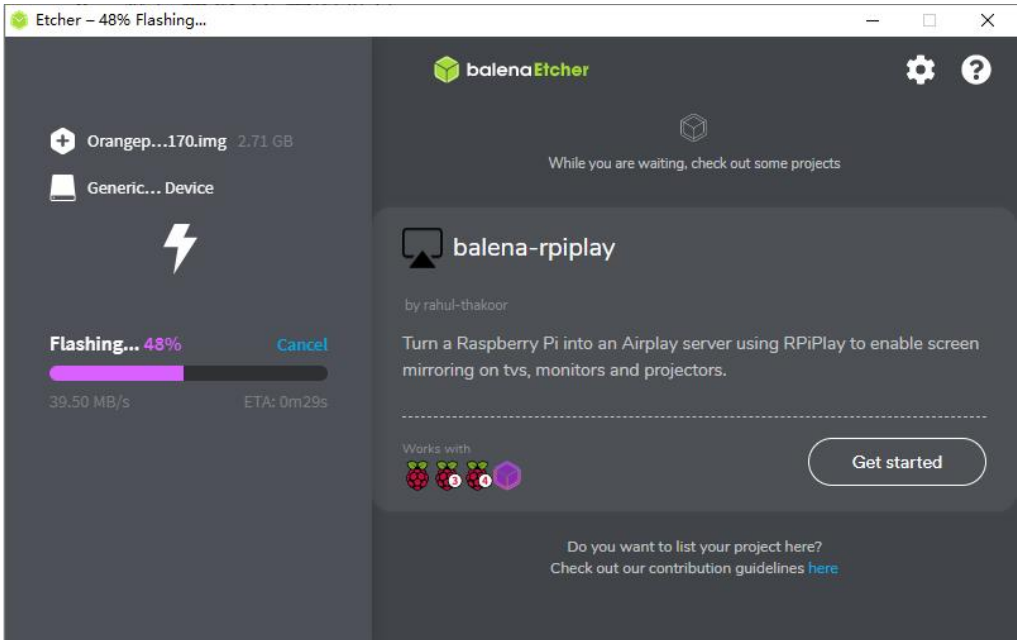


7) The specific steps to use balenaEtcher to burn the Linux image are aa follow
   a. First select the path of the Linux image file to burned
   b. Then select the drive letter of the TF card
   c. Finally, click Flash to start burning the Linux image to the TF card

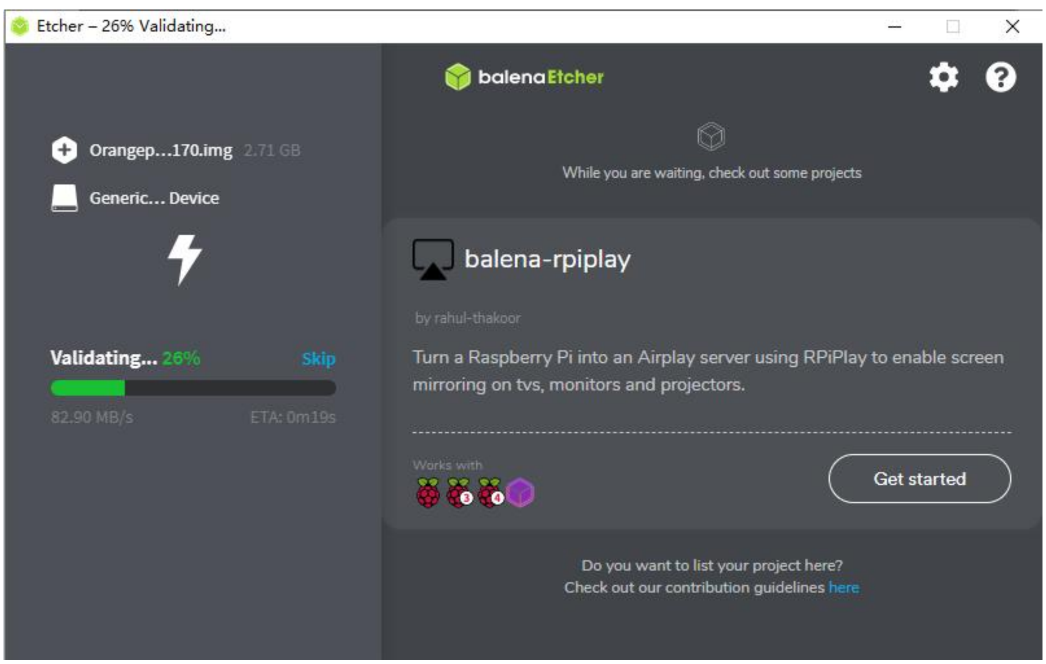


8) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux

image is being burned into the TF card

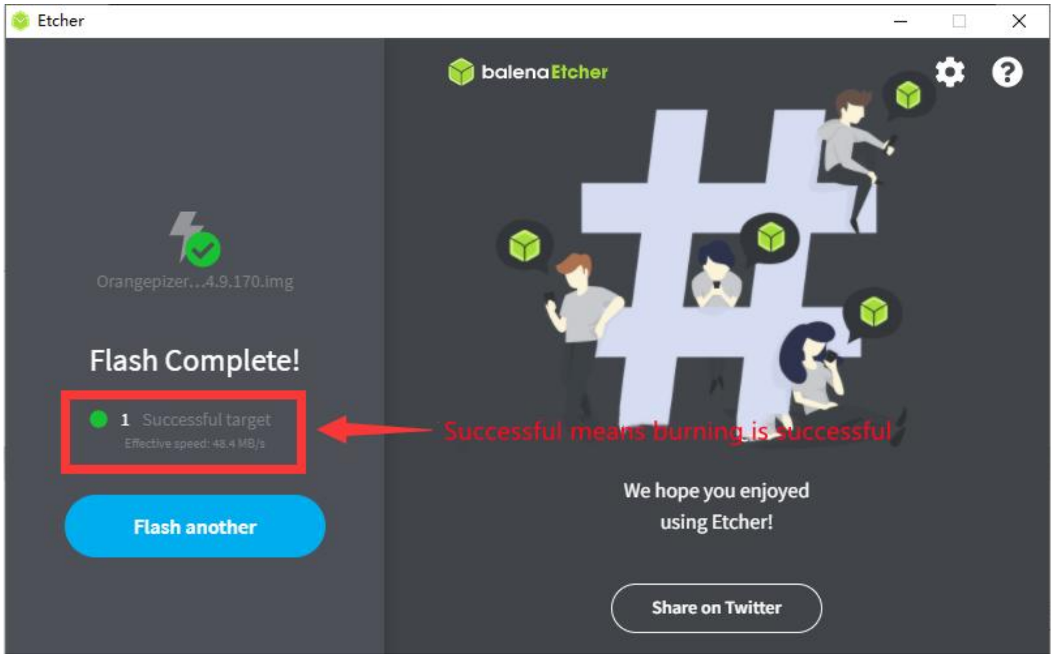


9) After burning the Linux image, balenaEtcher will also verify the image burned into the TF card by default to ensure that there is no problem in the burning process. As shown in the figure below, a green progress bar indicates that the image has been burnt, and balenaEtcher is verifying the burnt image



10) After successful burning, the display interface of balenaEtcher is shown in the figure

below. If a green indicator icon is displayed, it means that the image burning is successful. At this time, you can exit balenaEtcher, and then pull out the TF card and insert it into the TF card slot of the development board for use up

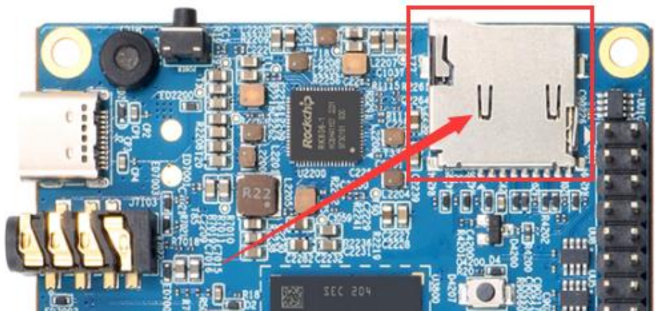


2. 3. 2. How to use RKDevTool to burn Linux image to TF card

1) First, you need to prepare a data cable with a good quality Type-C interface

2) You also need to prepare a 16GB or larger TF card. 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

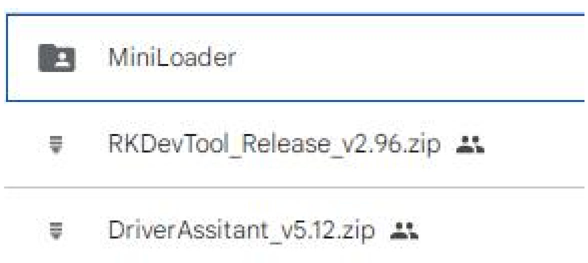
3) Then insert the TF card into the card slot of the development board



4) Then download Rockchip **DriverAssitant_v5.12.zip** and **MiniLoader** and the burning tool **RKDevTool_Release_v2.96.zip** from the **Orange Pi data download page, please make sure that the version of the downloaded RKDevTool tool is v2.96**

a. On the Orange Pi data download page, first select the official tool, and then enter the following folder

b. Then download all the files below



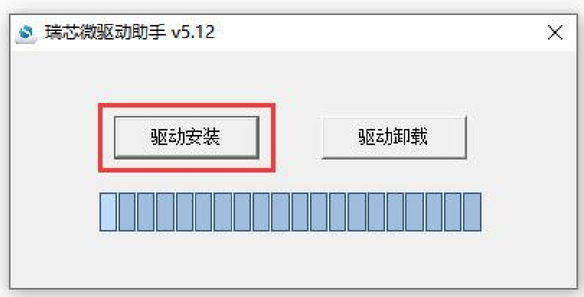
Note that the folder of MiniLoader - the things needed to burn the Linux image is hereinafter referred to as the MiniLoader folder.

5) Then download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system , the size is generally above 2GB

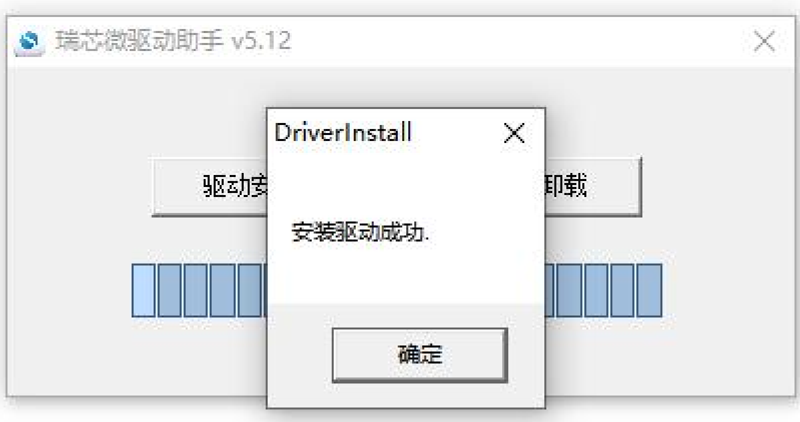
6) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

7) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

a. Click the "**Driver Installation**" button



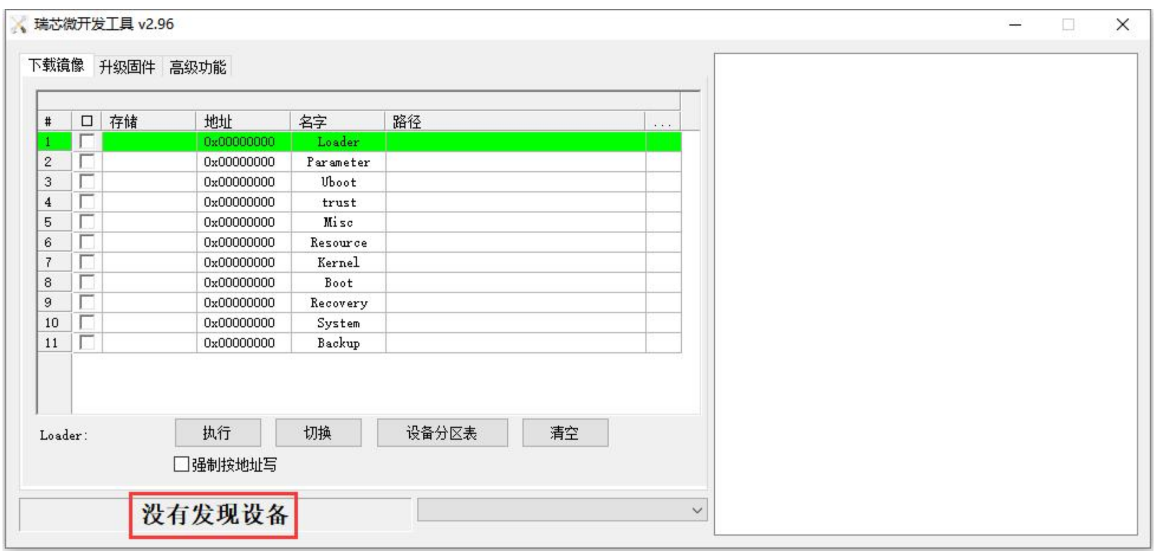
b. After waiting for a while, a pop-up window will prompt "**driver installed successfully**", and then click the "**OK**" button.



8) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be

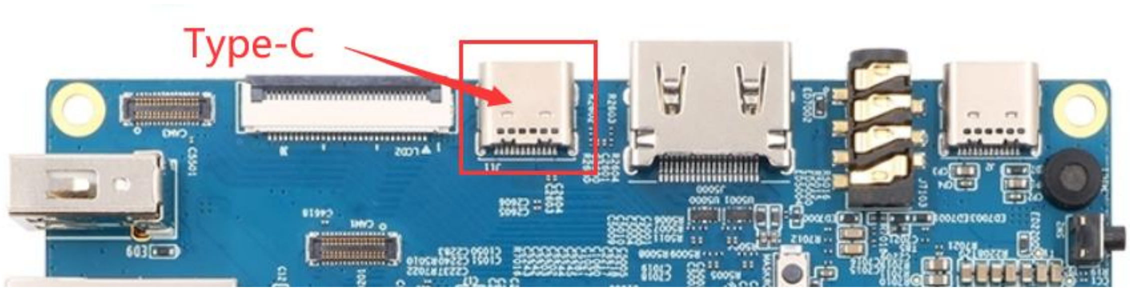
installed, just find **RKDevTool** in the decompressed folder and open it

9) After opening the **RKDevTool** burning tool, because the computer has not been connected to the development board through the Type-C cable at this time, the lower left corner will prompt "**No device found**"



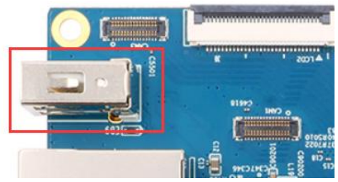
10) Then start burning the Linux image into eMMC

   a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below

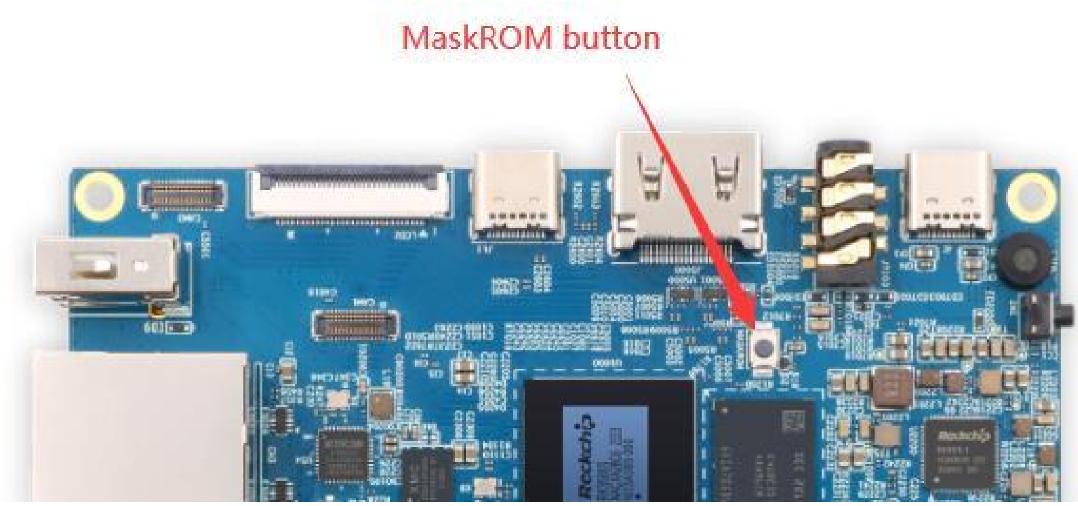


b. Make sure the board is not connected to power

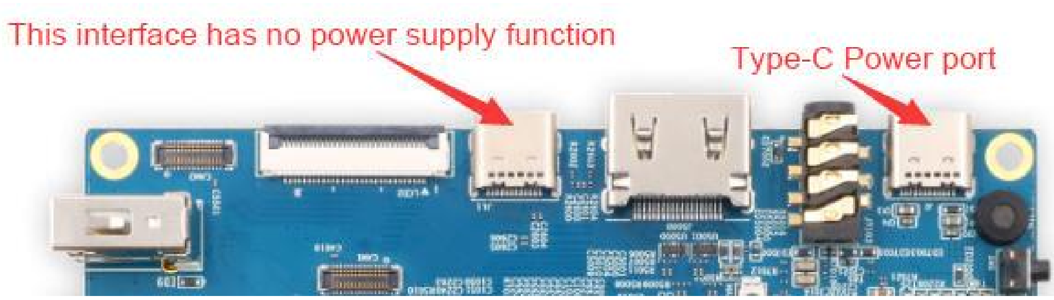
c. Also make sure that the white USB2.0 interface in the position shown below is not plugged into a USB device



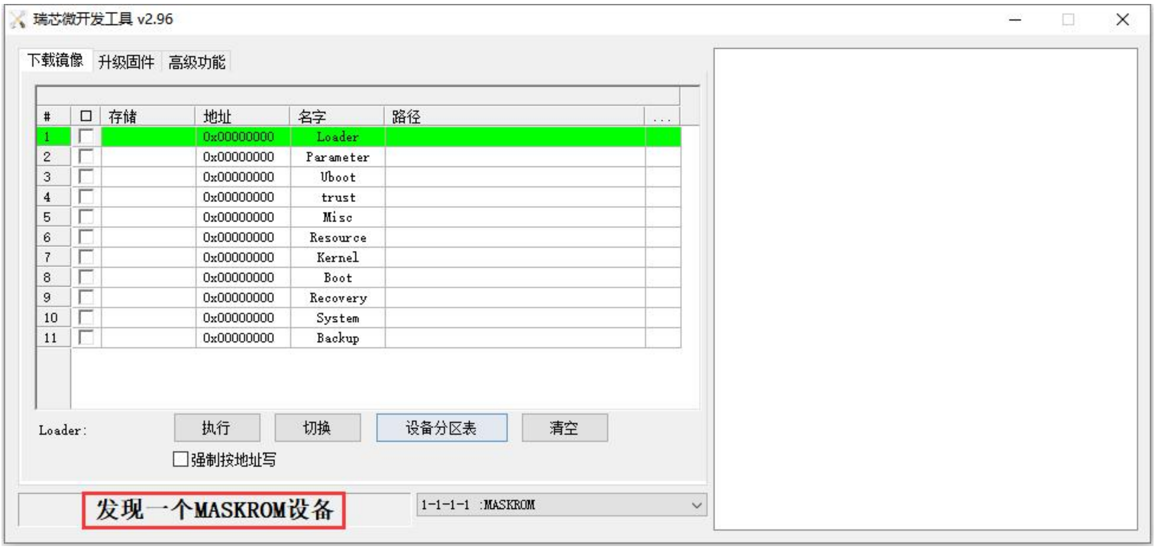
d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:



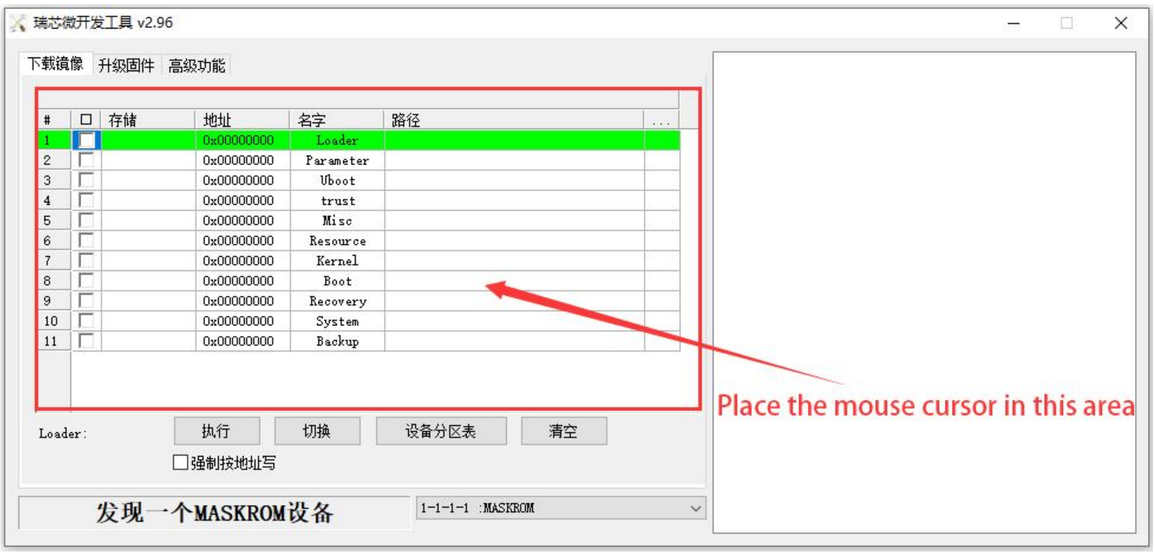
e. Then connect the power supply of the Type-C interface to the development board, and power on, and then release the MaskROM
button



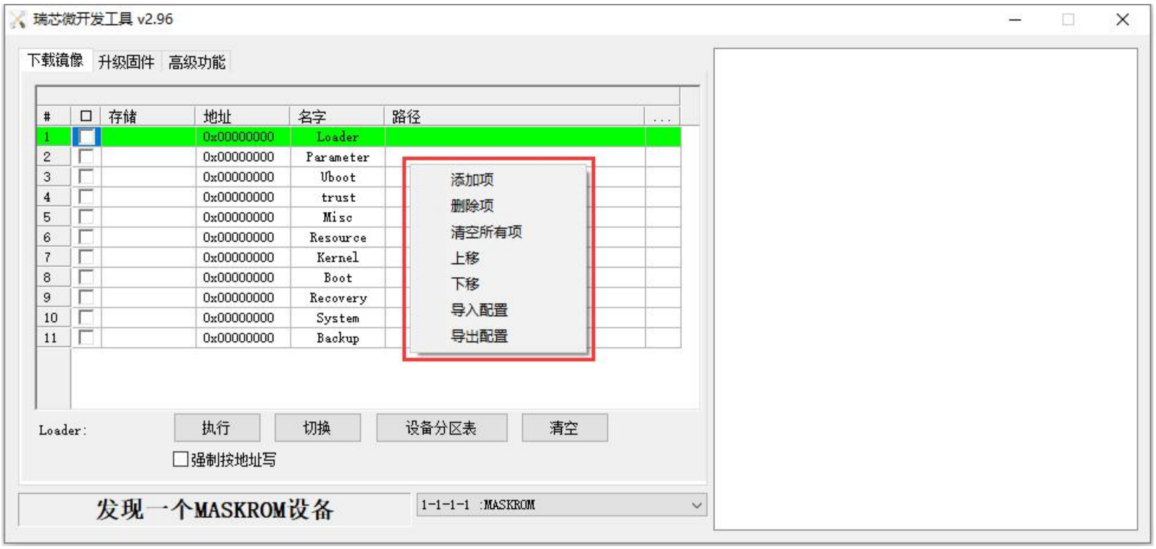
f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



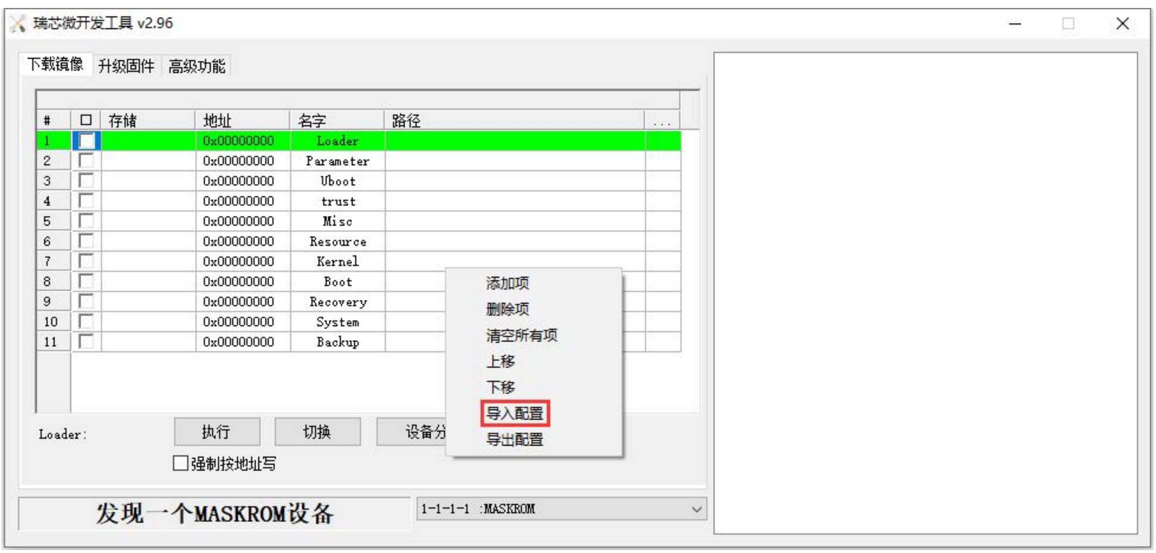
g. Then place the mouse cursor in the area below



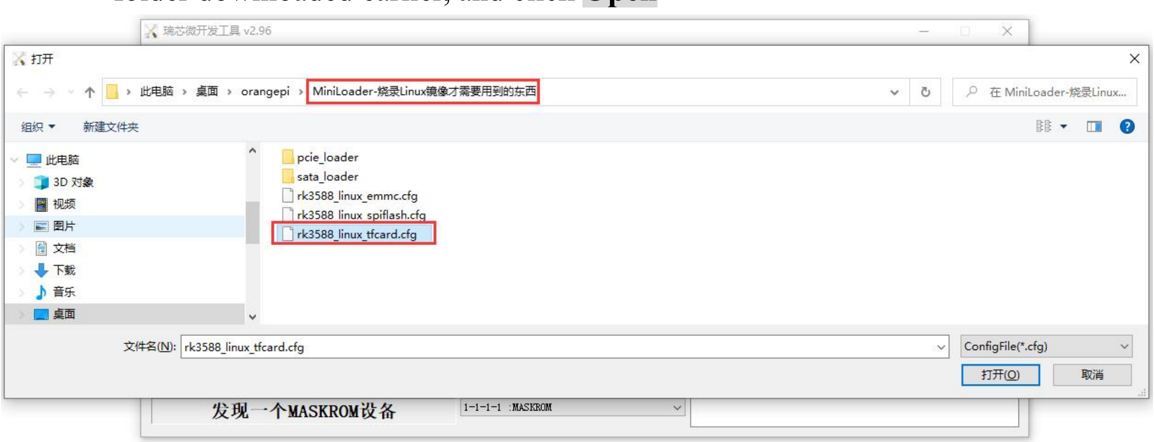
h. Then click the right mouse button and the selection interface shown in the figure below will pop up



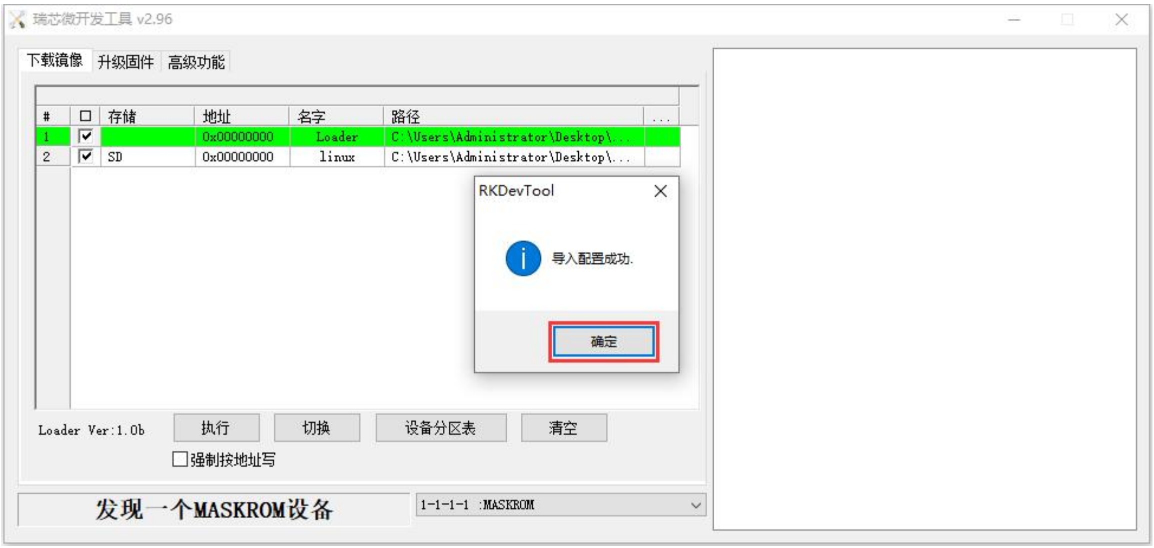
i. Then select the **import configuration** option



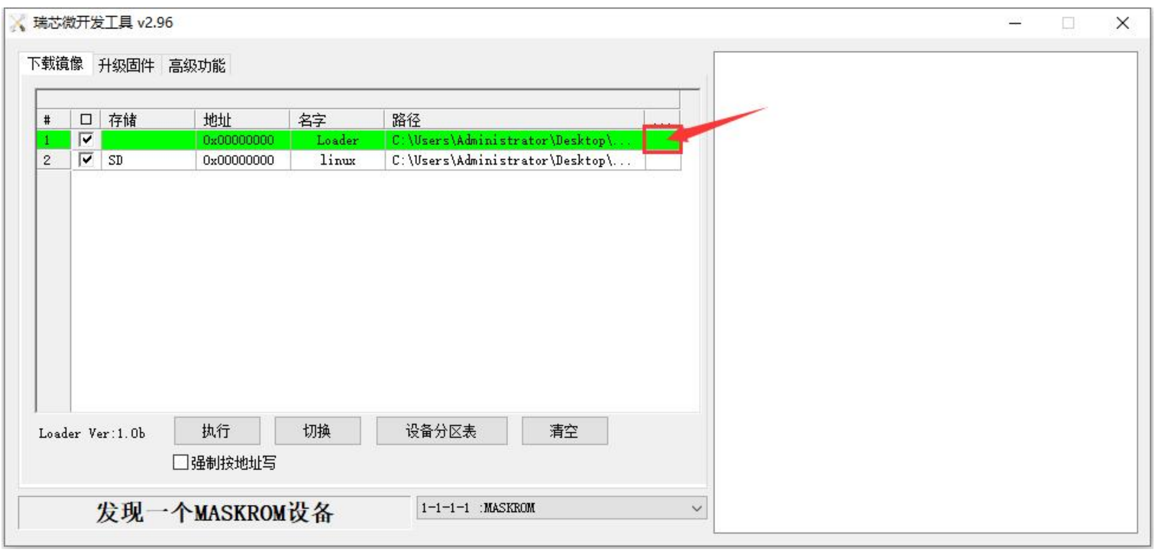
j. Then select the **rk3588_linux_tfcard.cfg** configuration file in the MiniLoader folder downloaded earlier, and click **Open**



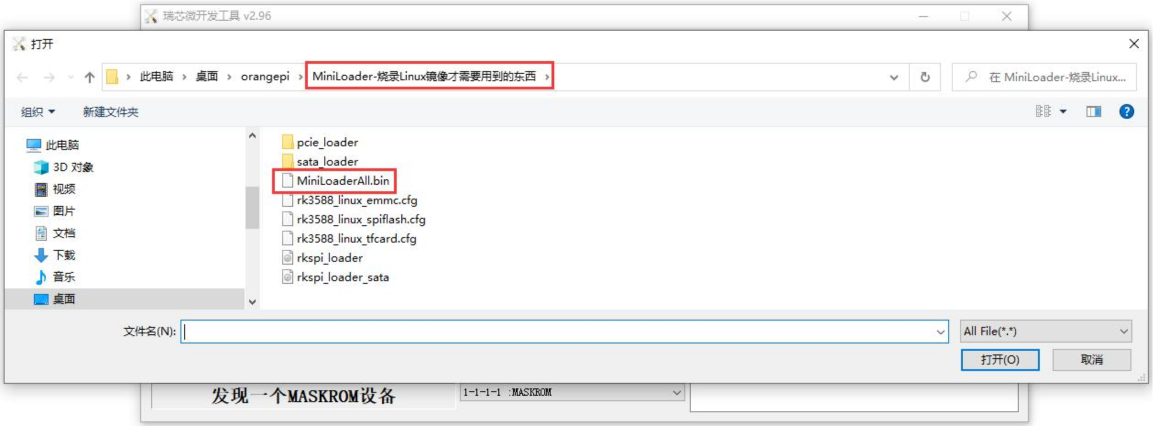
k. Then click **OK**



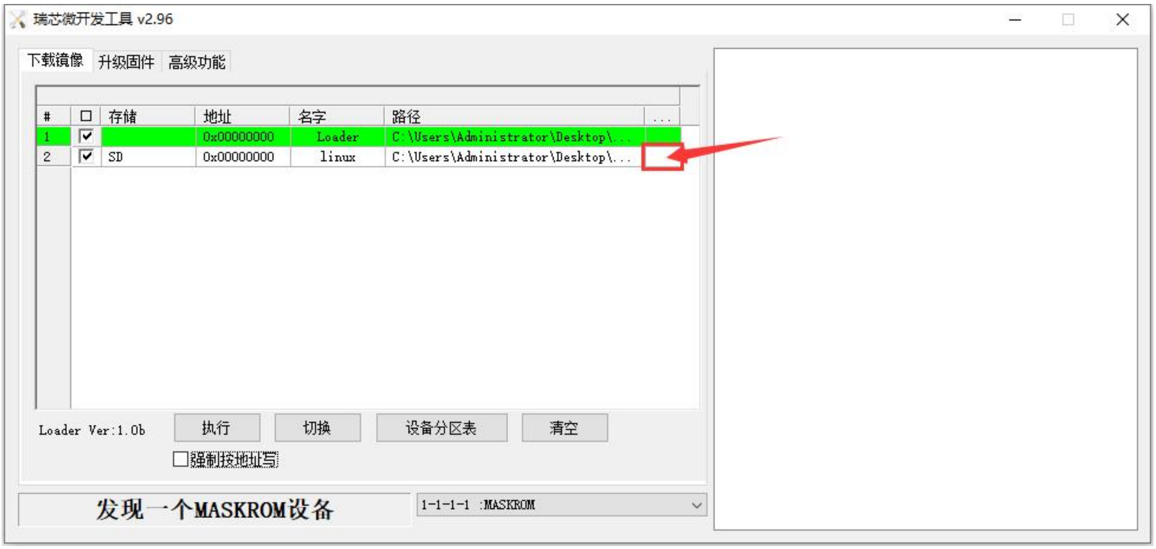
l. Then click the position shown in the figure below



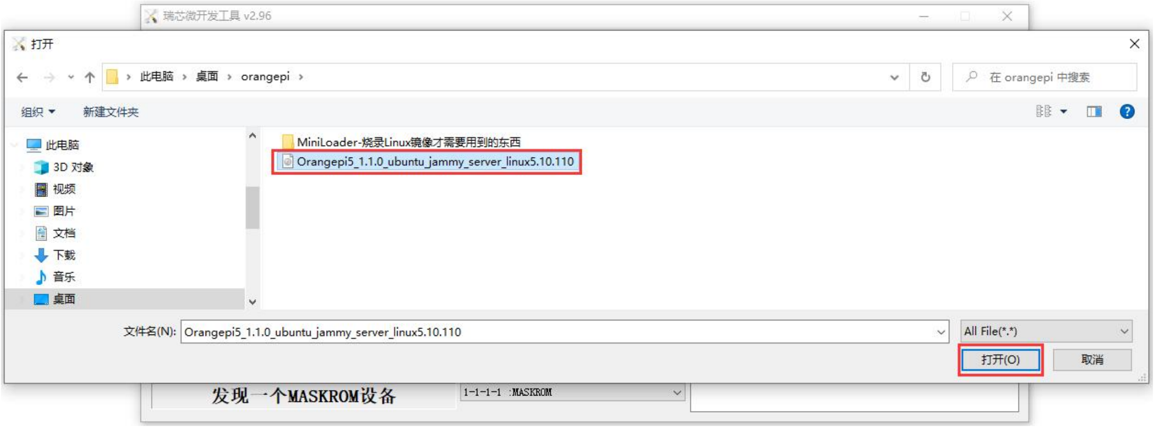
m. Select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click to **open**



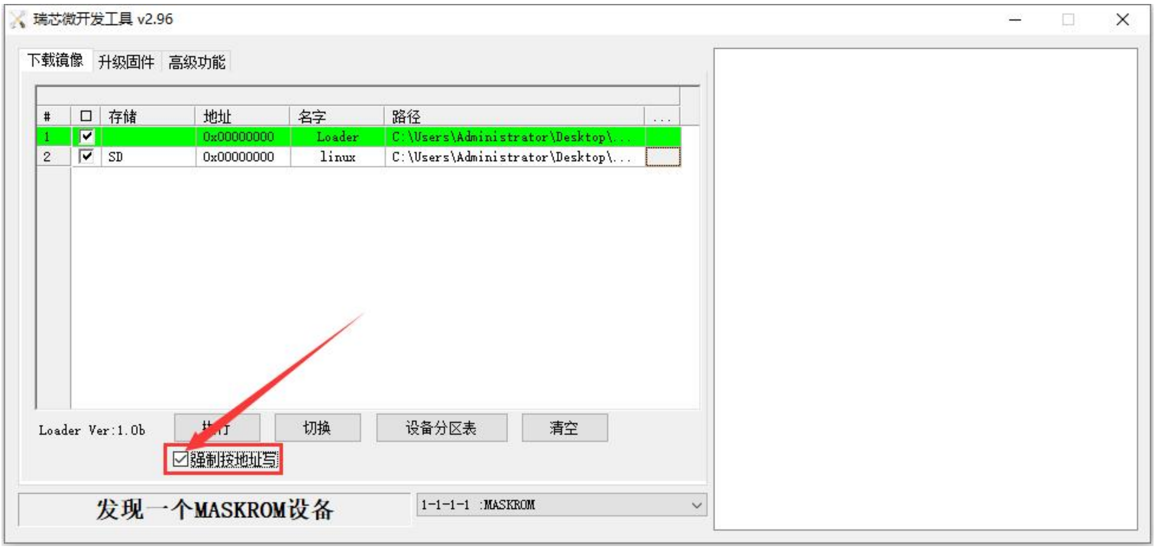
n. Then click the position shown in the figure below



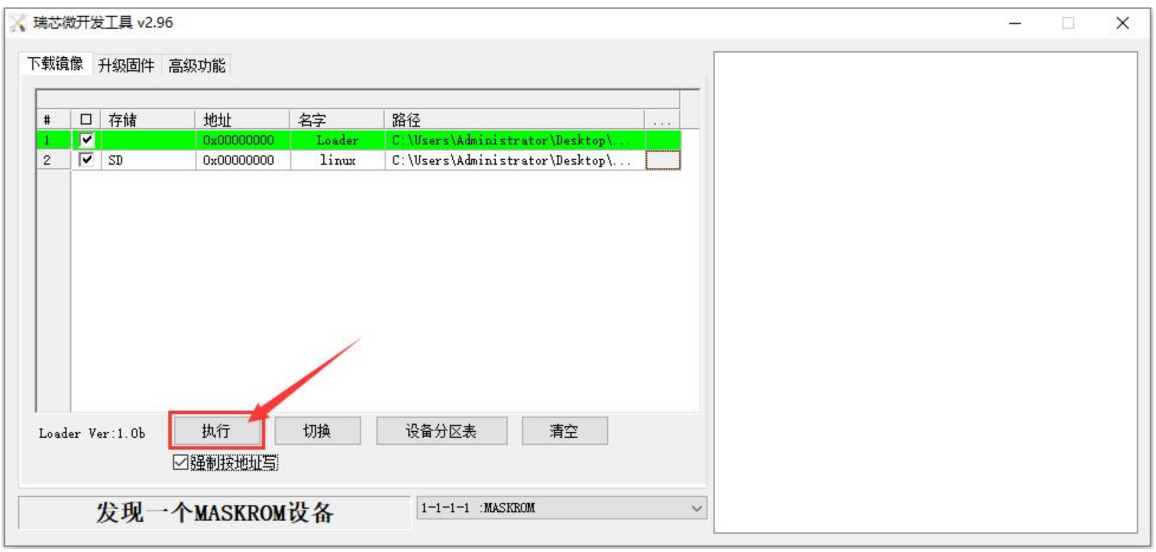
o. Then select the path of the linux image you want to burn, and then click **Open**



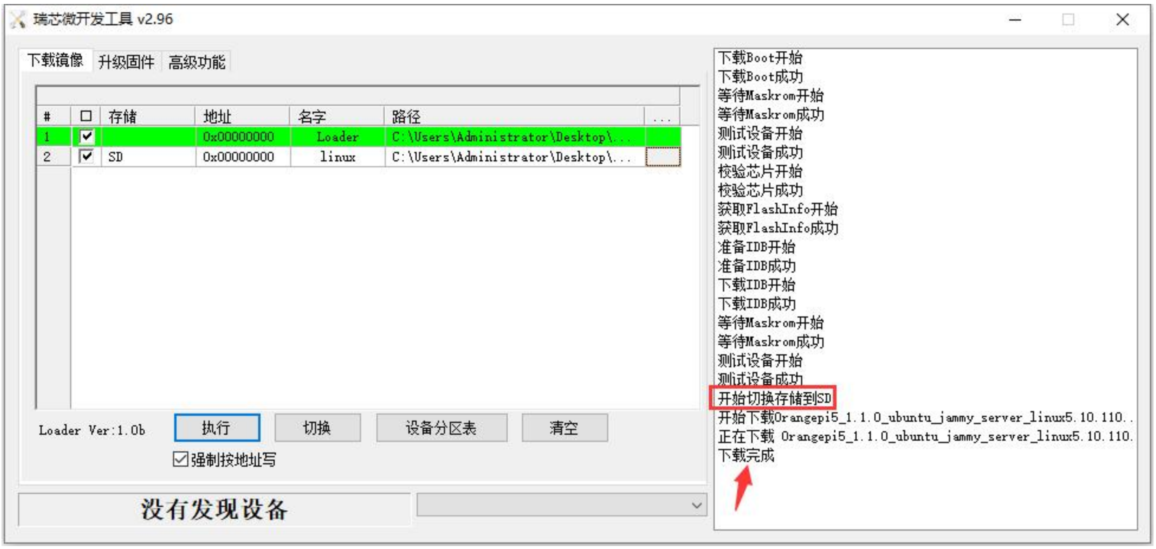
p. Then please check the option to **force write by address**



q. Click the execute button again to start burning the Linux image to the TF card of the development board



r. The display log after burning the Linux image is shown in the figure below 显示 log



s. After burning the linux image to the TF card, the linux system will start automatically.

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

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

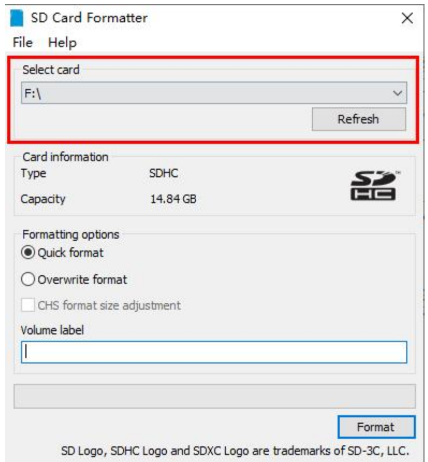
3) Then format the TF

a. **SD Card Formatter** can be used to format the TF card. The download address is:

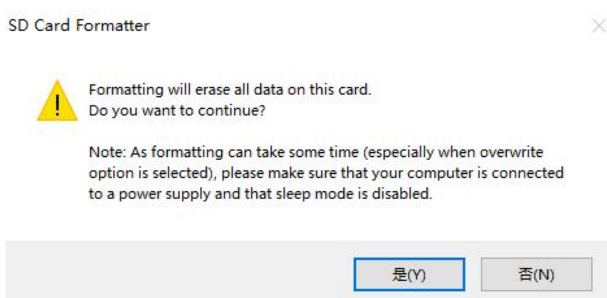
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

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

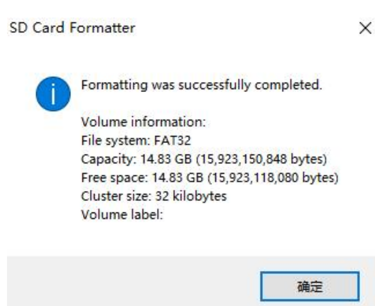
c. If only a TF card is inserted into the computer, the drive letter of the TF card will be displayed in the "**Select card**" column. If multiple USB storage devices are inserted into the computer, you can select the corresponding drive letter of the TF card through the drop-down box.



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

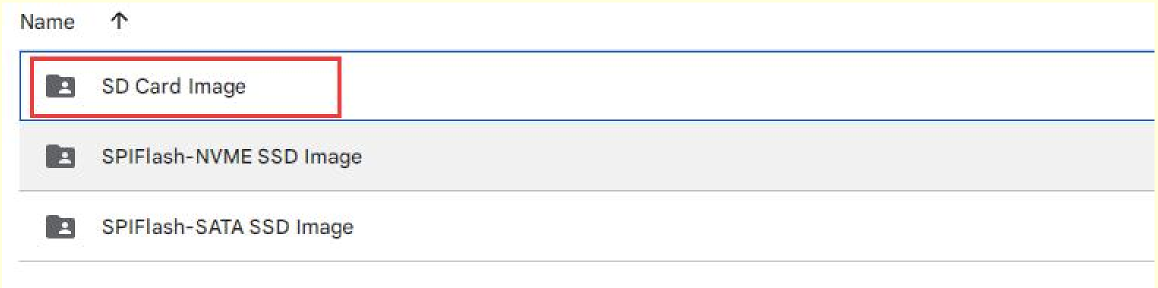


e. After formatting the TF card, the information shown in the figure below will pop up, click OK

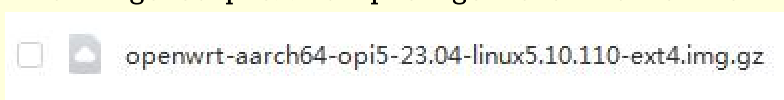


4) Then download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system , the size is generally above 2GB

Note that if you download an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the "TF card boot image" folder.



Then select the image compression package below to download



5) Use **Win32Diskimager** to burn the Linux image to the TF Card

a. The download page of Win32Diskimager is

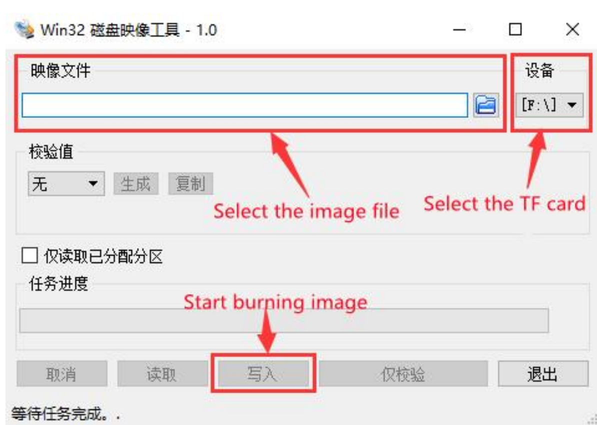
http://sourceforge.net/projects/win32diskimager/files/Archive/

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

a) First select the path of the image

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. How to burn Linux image to TF card based on Ubuntu

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

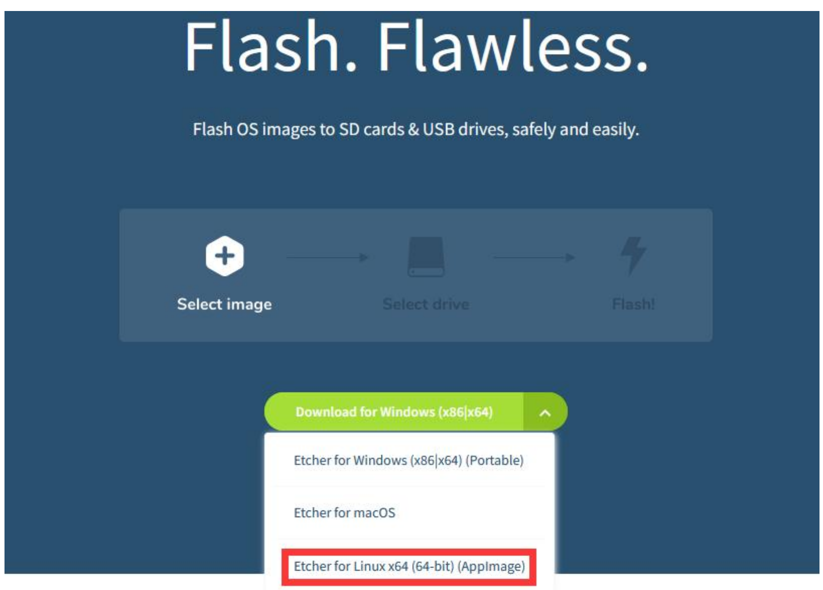
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brand

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

3) Download the balenaEtcher software, the download address is:

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

4) After entering the balenaEtcher download page, please select the Linux version of the software from the drop-down box to download



5) Download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression

software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system. The size is generally more than 22GB

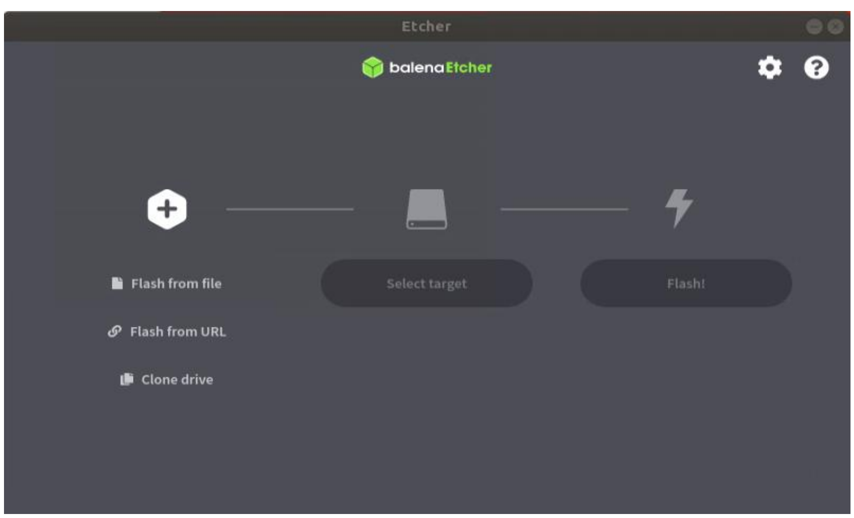
The decompression command for the compressed package ending in 7z is as follow

```
test@test:~$ 7z x Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
test@test:~$ ls Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.*
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.sha     #checksum file
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img     #Image file
```

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

```
test@test:~$ sha256sum -c *.sha
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img: OK
```

7) Then double-click **balenaEtcher-1.5.109-x64.AppImage** on the graphical interface of Ubuntu PC to open balenaEtcher (**no installation required**), and the interface after balenaEtcher is opened is shown in the figure belobalenaEtcher

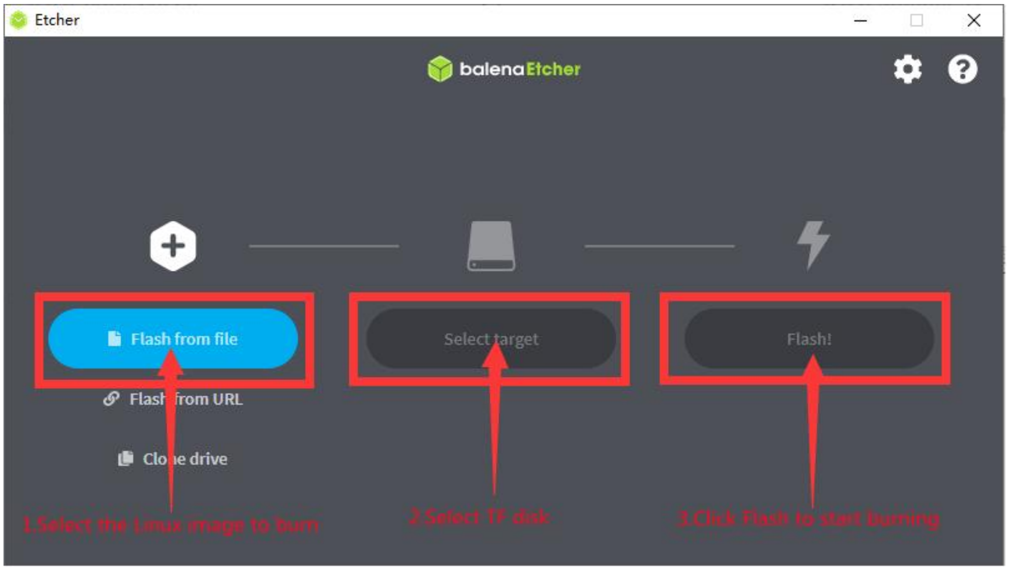


8) The specific steps to use balenaEtcher to burn the Linux image are as follows

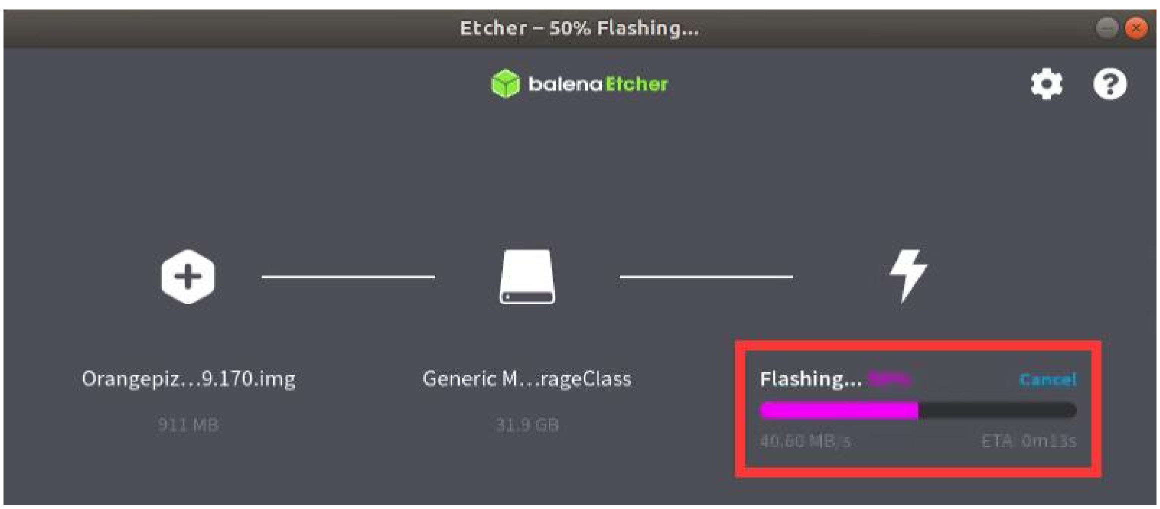
a. First select the path of the Linux image file to be burned

b. Then select the drive letter of the TF Card

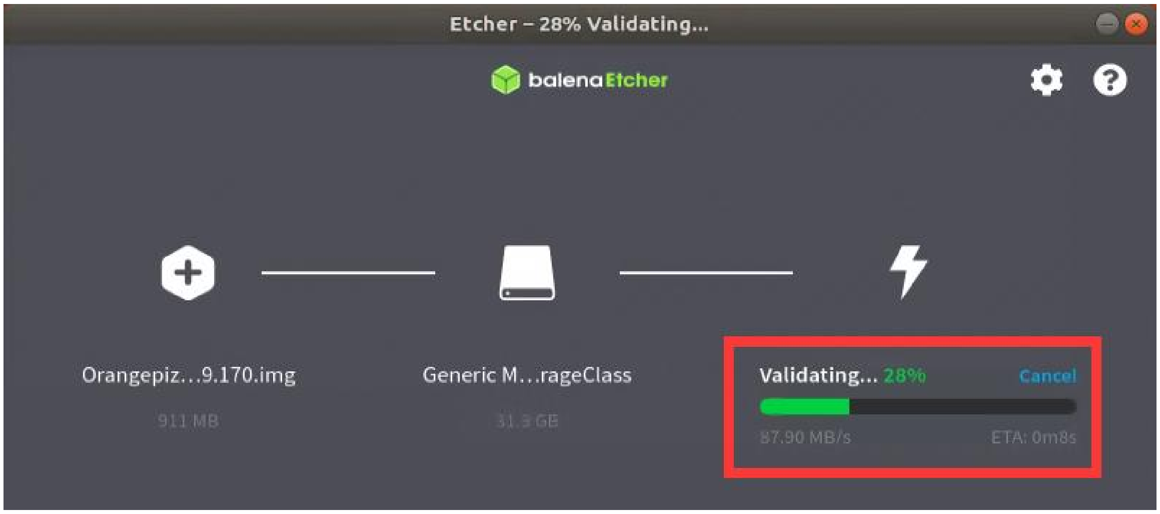
c. Finally, click Flash to start burning the Linux image to the TF Card



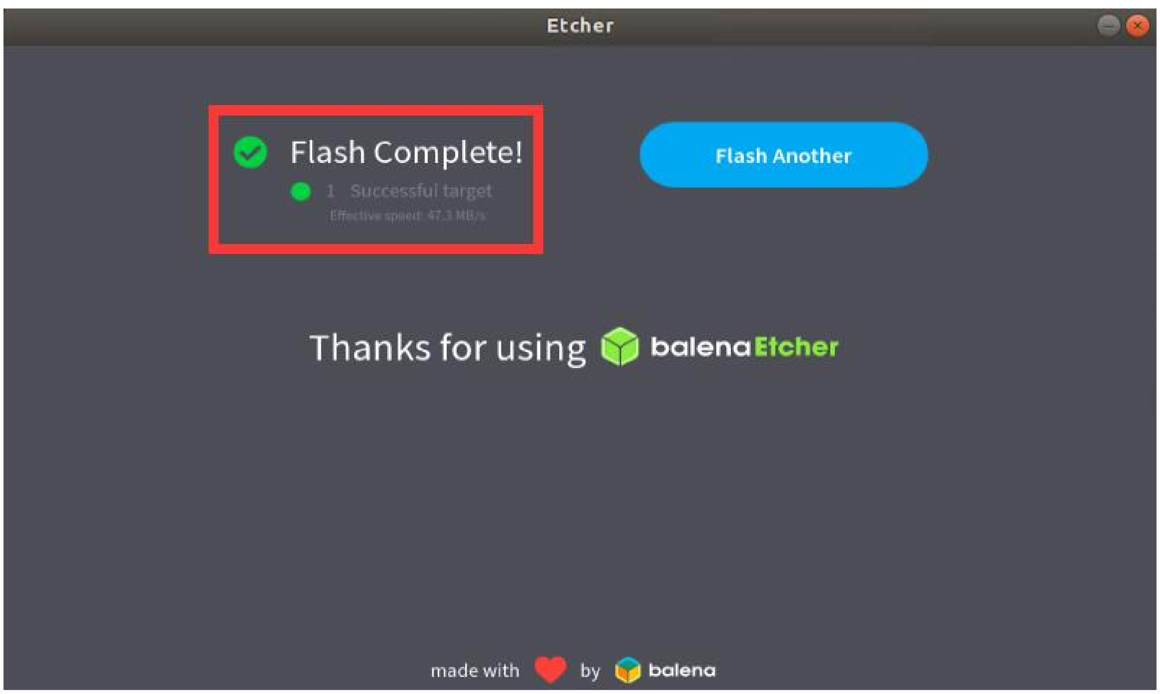
9) The interface displayed during the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF
Card



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



12) The display interface of Balenaetcher after the successful record is completed. If the green indicator icon is displayed in the figure below, the image burning is successful, then you can exit Balenaetcher, then unplug the TF card into the TF card slot in the development board and use it.



2. 5. How to write Linux image to SPI Flash+NVMe SSD

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

Note that all the following operations are performed on a Windows computer.

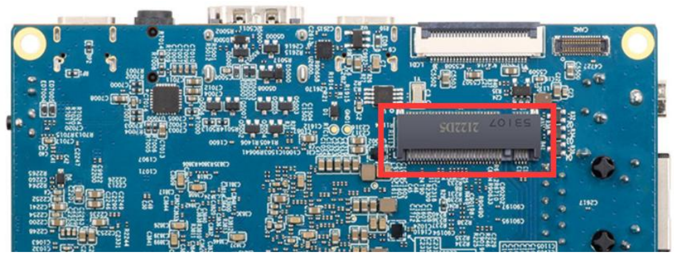
2. 5. 1. How to use RKDevTool to burn

1) First, you need to prepare an NVMe SSD. The PCIe supported by the M.2 slot of the development board is PCIe2.0x1, and the theoretical maximum speed is 500MB/s. PCIe3.0 and PCIe4.0 NVMe SSDs are also available, but the highest speed is only PCIe2.0.

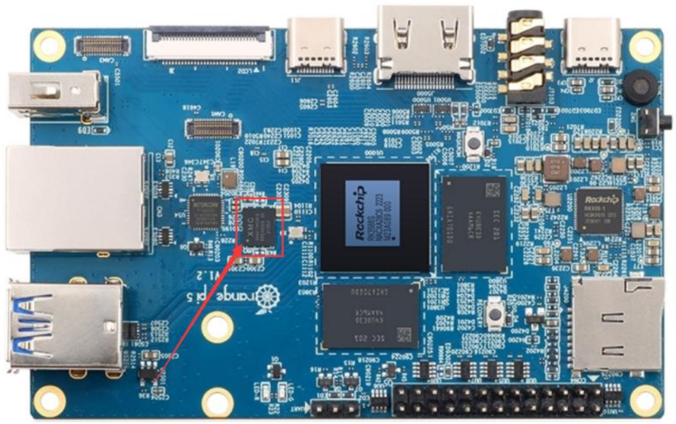
a. The M.2 2230 SSD is as follows

b. The M.2 2242 SSD is as follows

2) Then insert the NVMe SSD into the M.2 PCIe interface of the development board and fix it



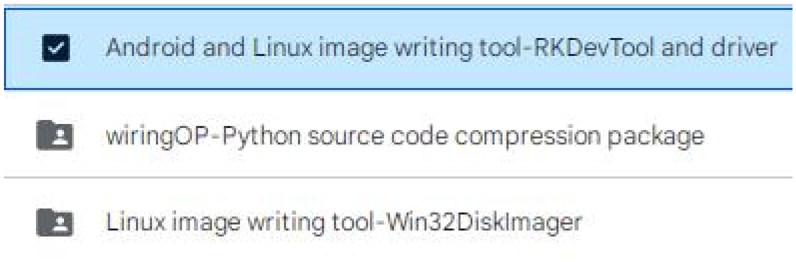
3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming



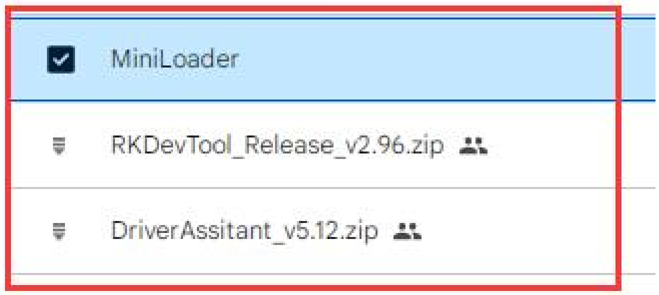
4) Then you need to prepare a data cable with good quality Type-C interface

5) Then download Rockchip **DriverAssitant_v5.12.zip** and **MiniLoader** and the burning tool **RKDevTool_Release_v2.96.zip** from the **Orange Pi data download page,** please make sure that the version of the downloaded **RKDevTool** tool is **v2.96**

a. On the Orange Pi data download page, first select the official tool, and then enter the following folder



b. Then download all the files below



Note that the folder of MiniLoader - the things needed to burn the Linux image is hereinafter referred to as the MiniLoader folder.

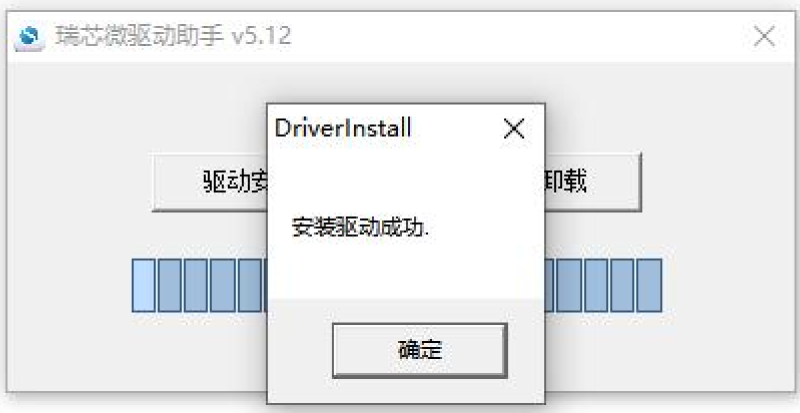
5) Then download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with ".img" is the image file of the operating system , the size is generally above 2GB

6) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

7) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

a. Click the "**Driver Installation**" button

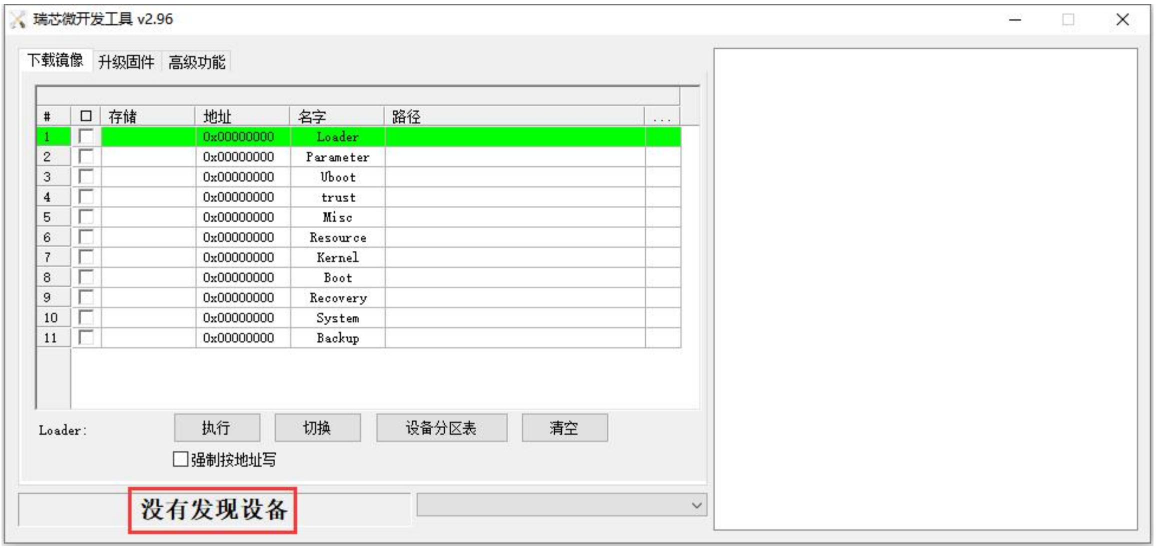
b. After waiting for a while, a pop-up window will prompt "**driver installed successfully**", and then click the "**OK**" button.



8) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

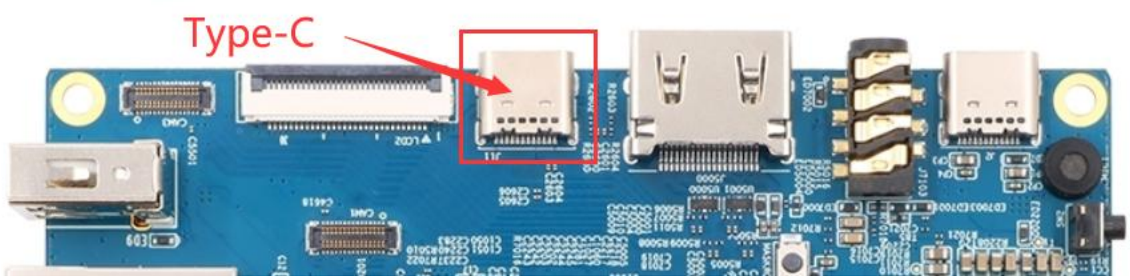
9) After opening the **RKDevTool** burning tool, because the computer has not been connected to the development board through the Type-C cable at this time, the lower left

corner will prompt "**No device found**"



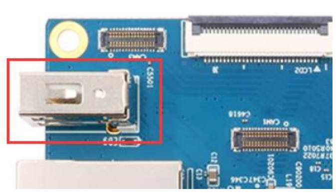
10) Then start burning the Linux image to the SSD

a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below.



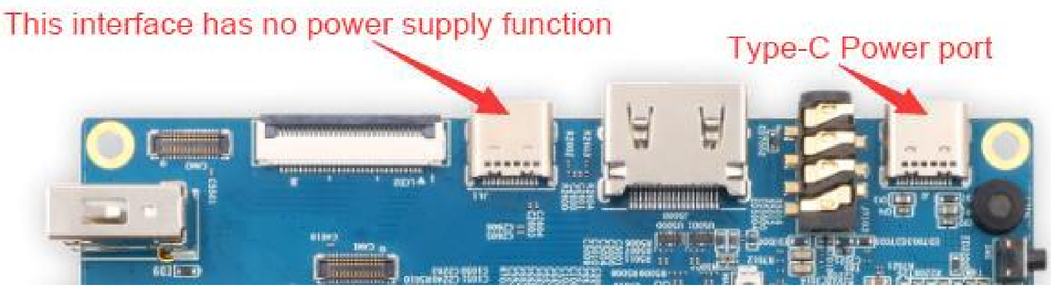
b. Make sure that the development board is not connected to the power supply and inserted into the TF card.

c. Also make sure that the white USB2.0 interface in the position shown below is not plugged into a USB deviceUSB2.0.

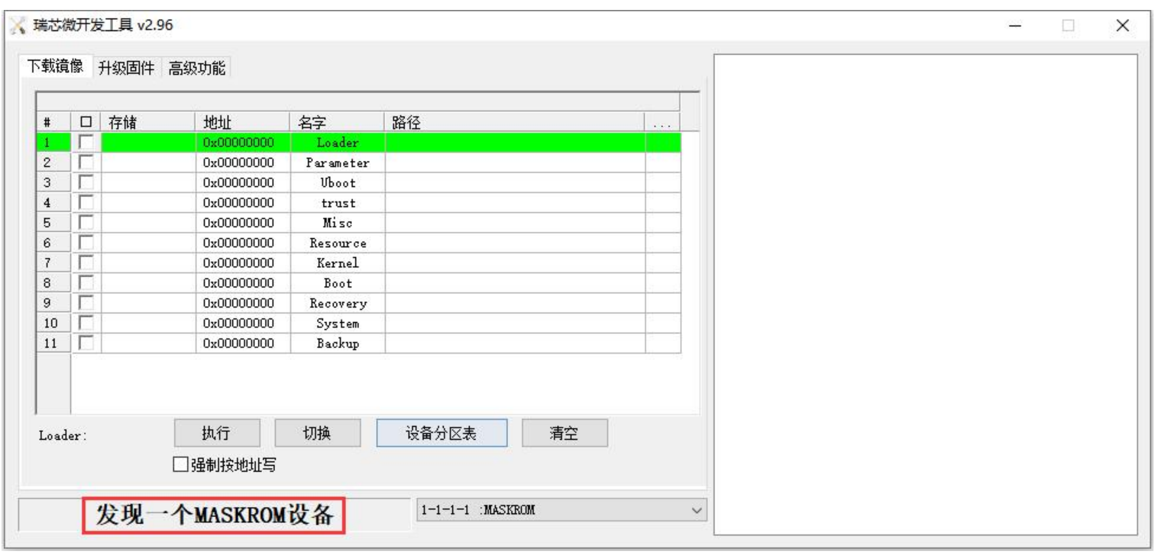


d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:

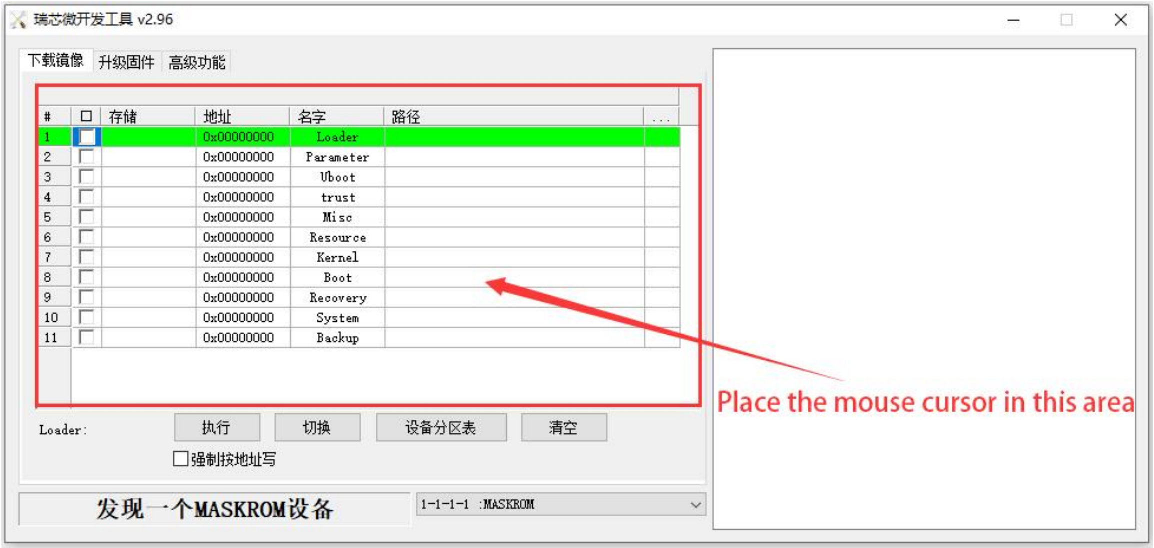
e. Then connect the power supply of the Type-C interface to the development board, and power on, and then release the MaskROM button



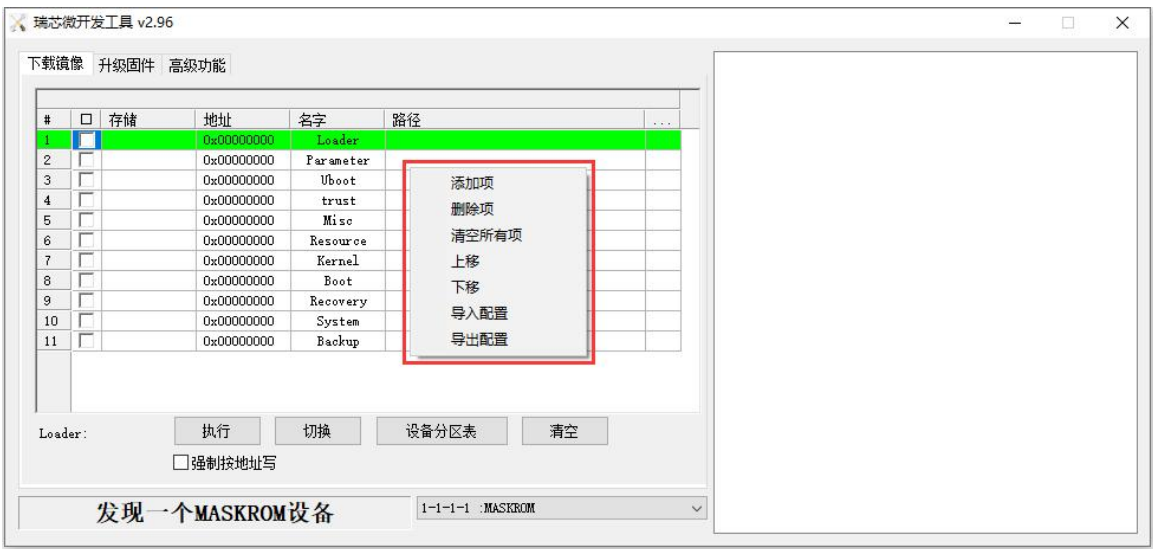
f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



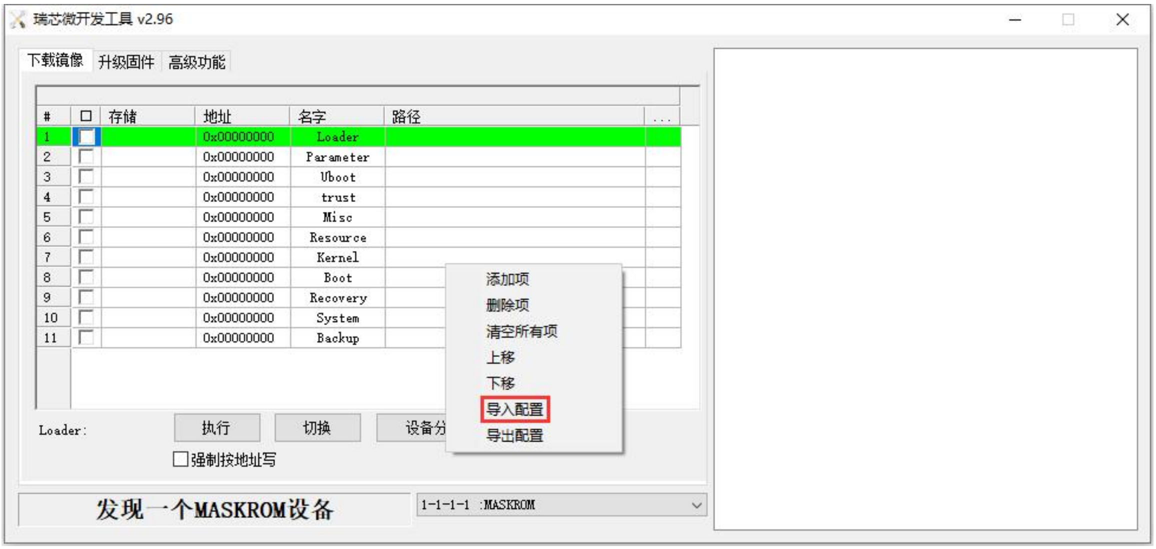
g. Then place the mouse cursor in the area below



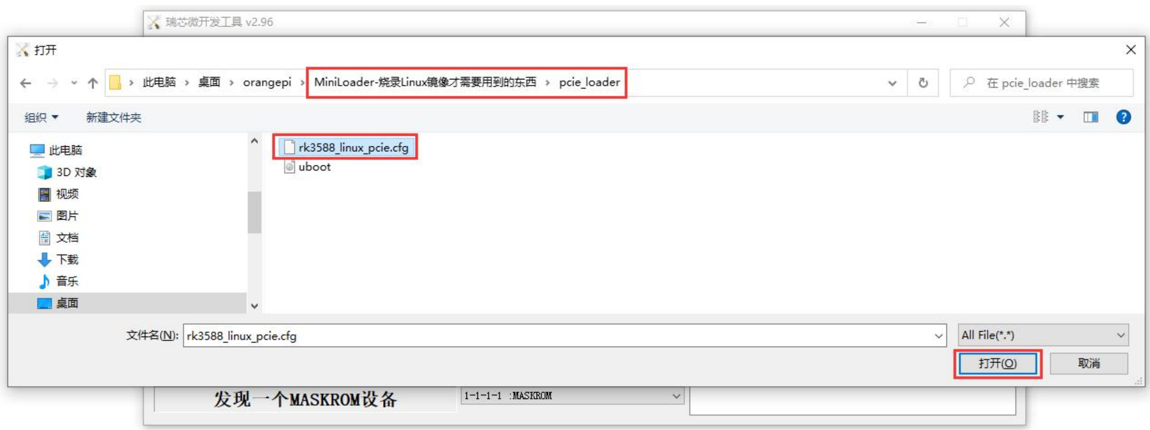
h. Then click the right mouse button and the selection interface shown in the figure below will pop up



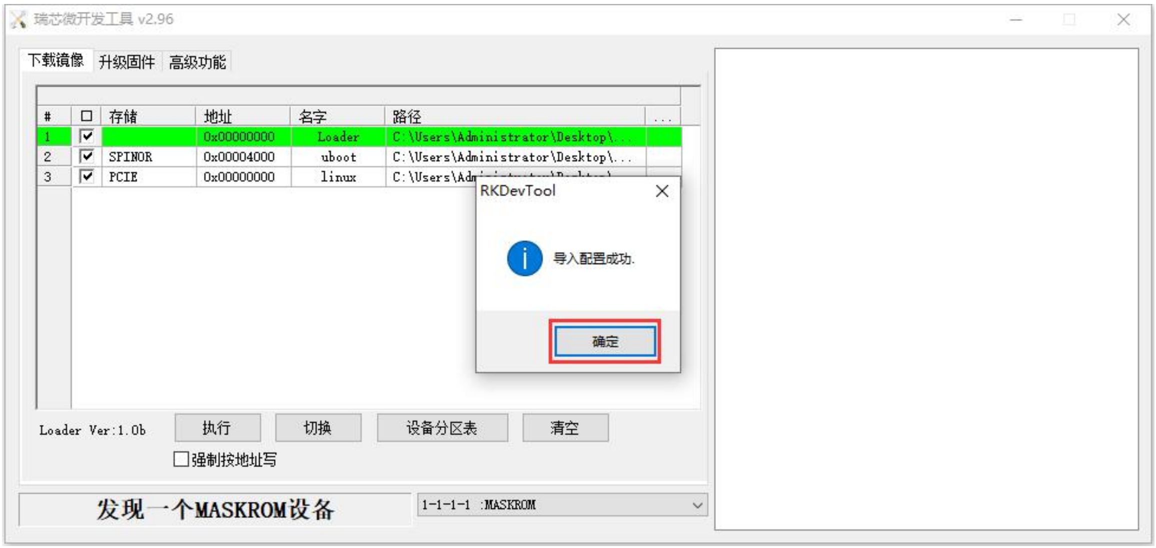
i. Then select the **import configuration** option



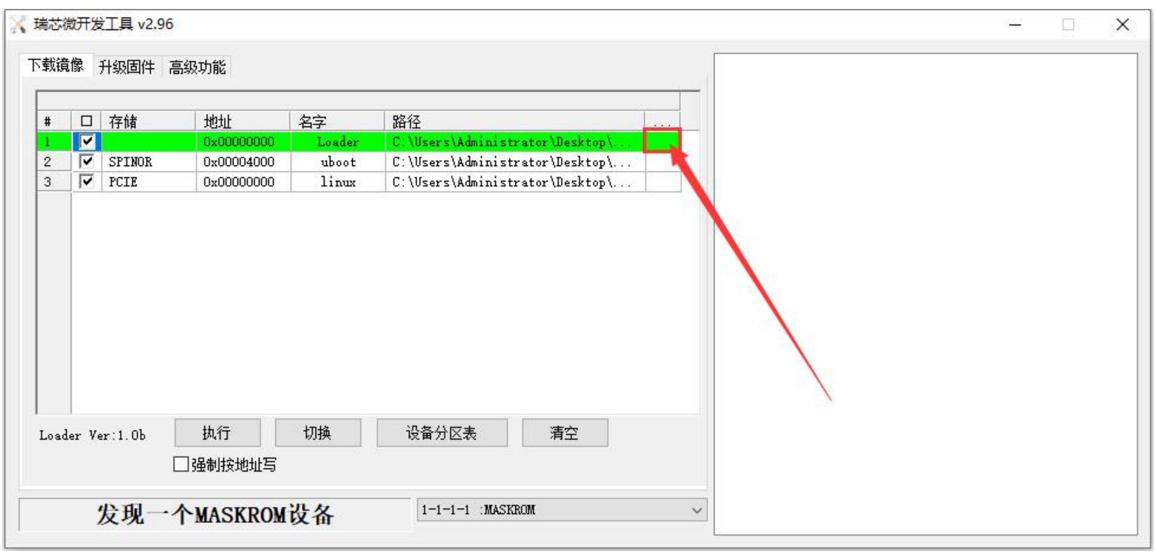
j. Then enter the **MiniLoader** folder downloaded earlier, and then enter the pcie_loader folder, then select the **rk3588_linux_pcie.cfg** configuration file, and click **Open**.



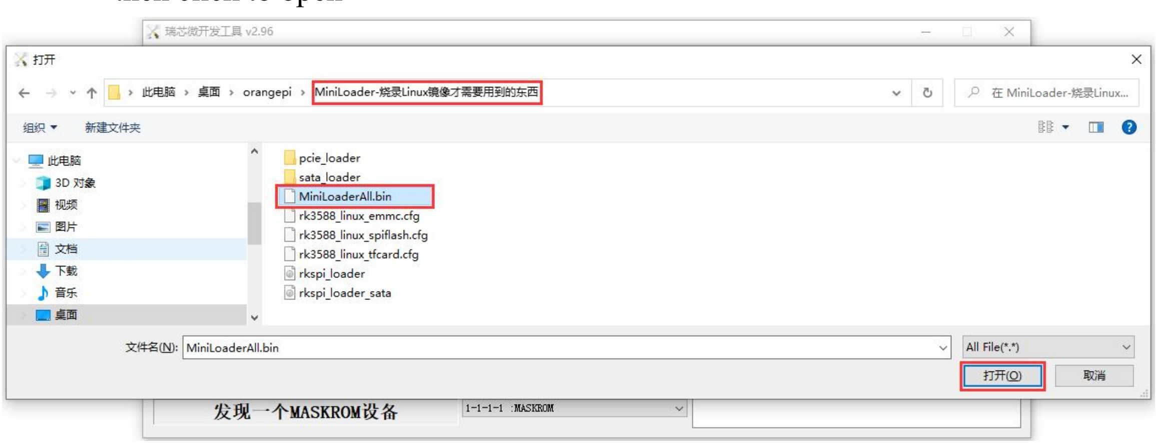
k. Then click **OK**



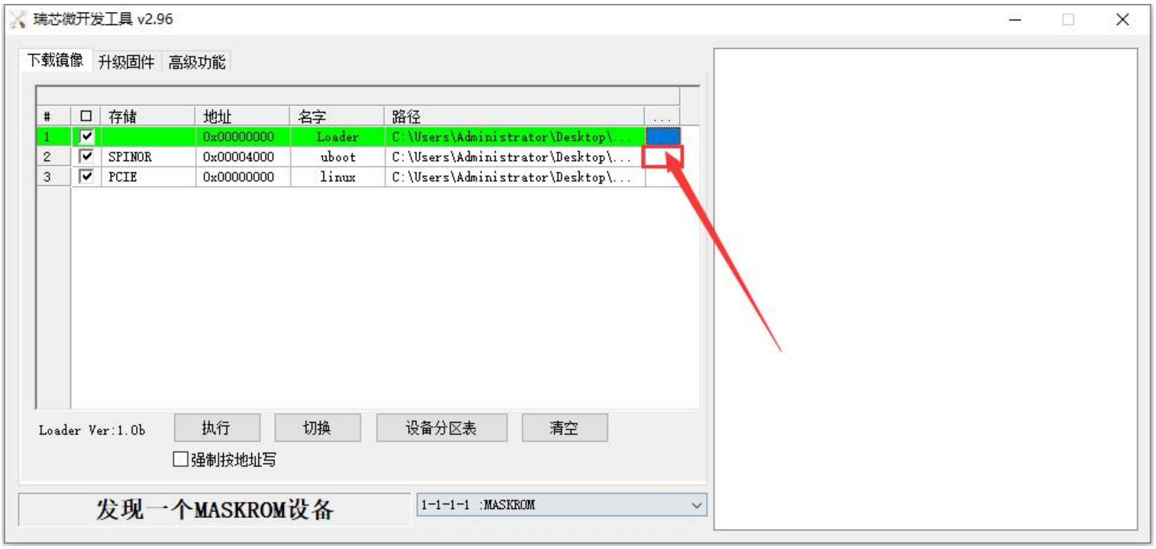
l. Then click the position shown in the figure below



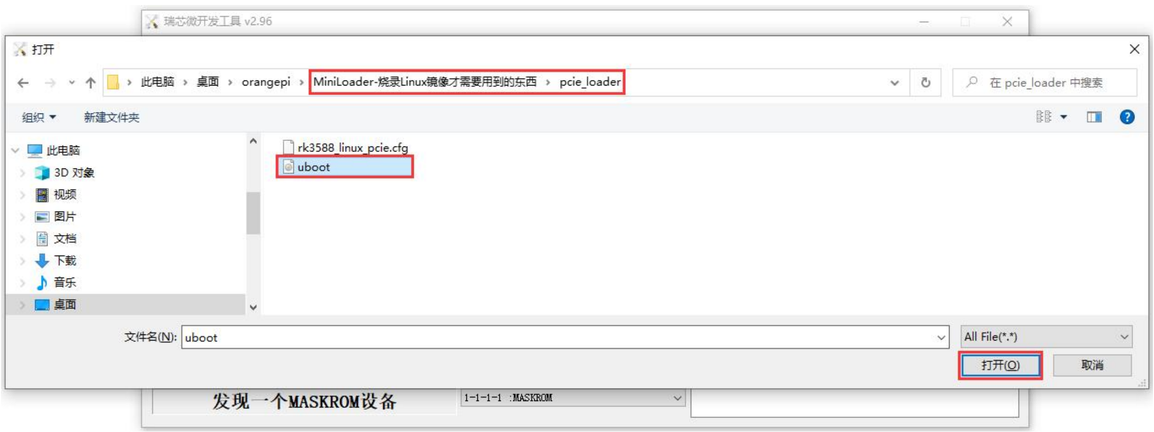
m. Select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click to open



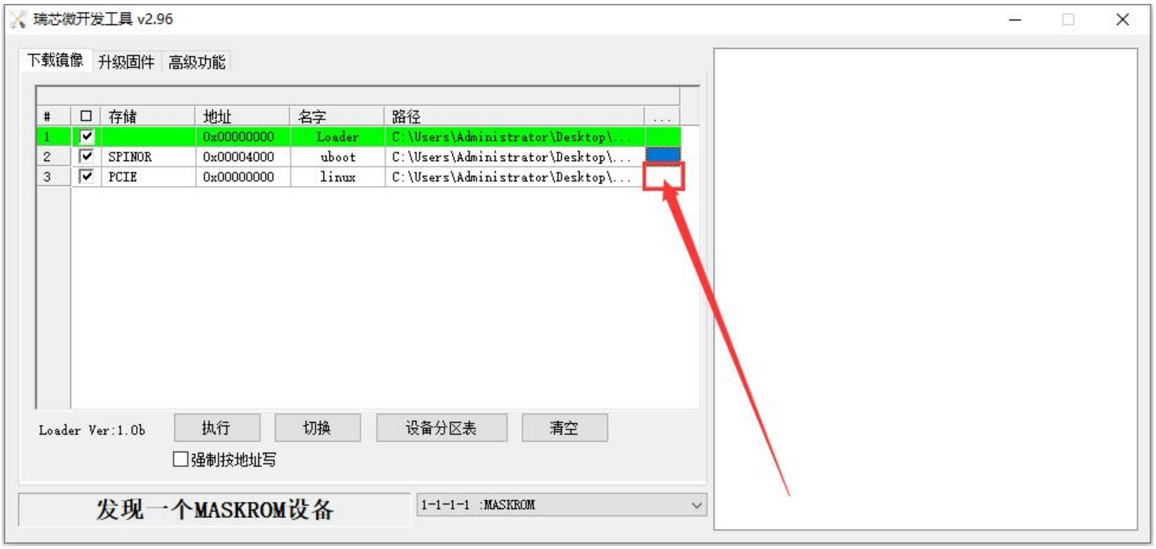
n. Then click the position shown in the figure below



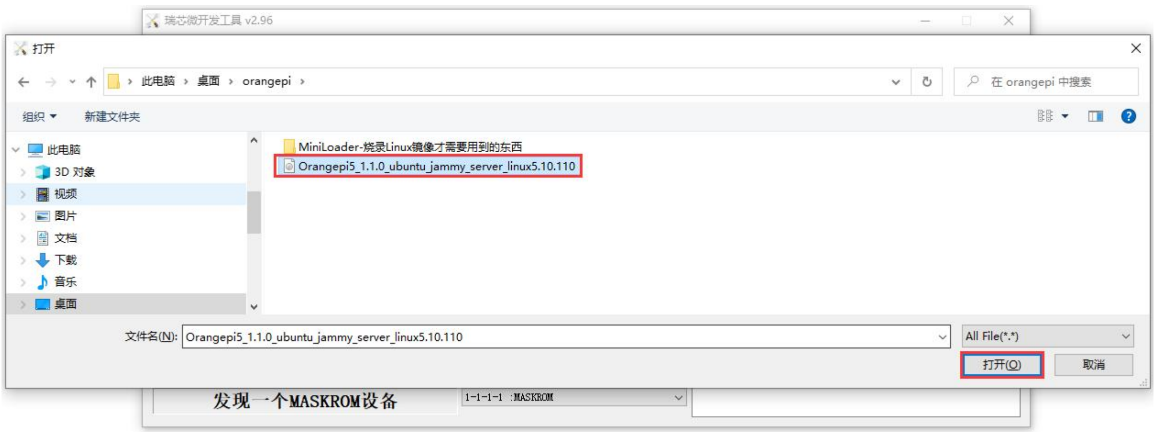
o. Then enter the **MiniLoader** folder downloaded earlier, then enter the pcie_loader folder, then select the **uboot** file, and click **Open.**



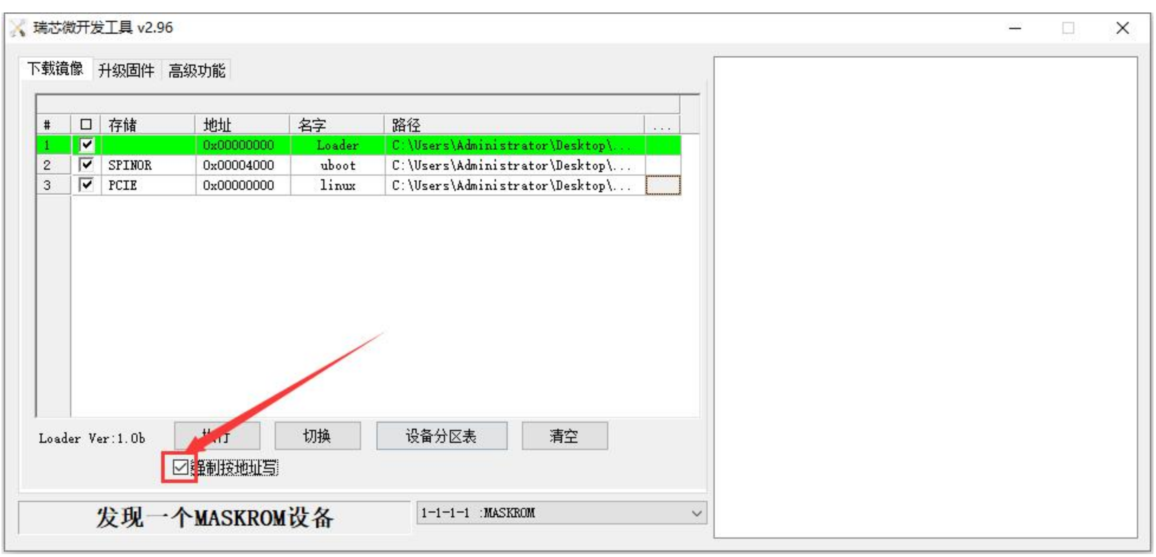
p. Then click the position shown in the figure below



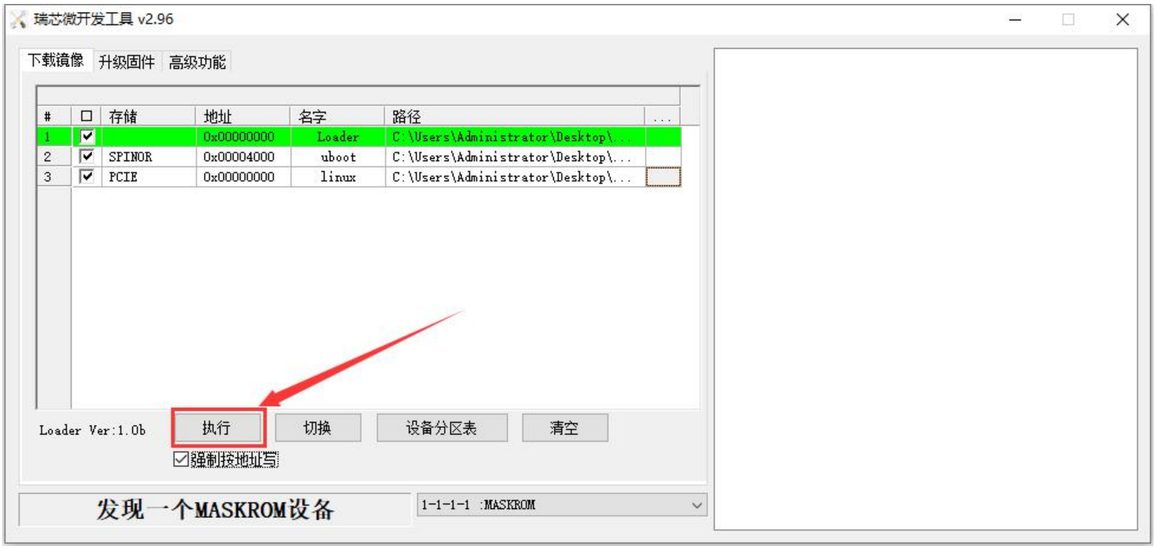
q. Then select the path of the linux image you want to burn, and click **Open**



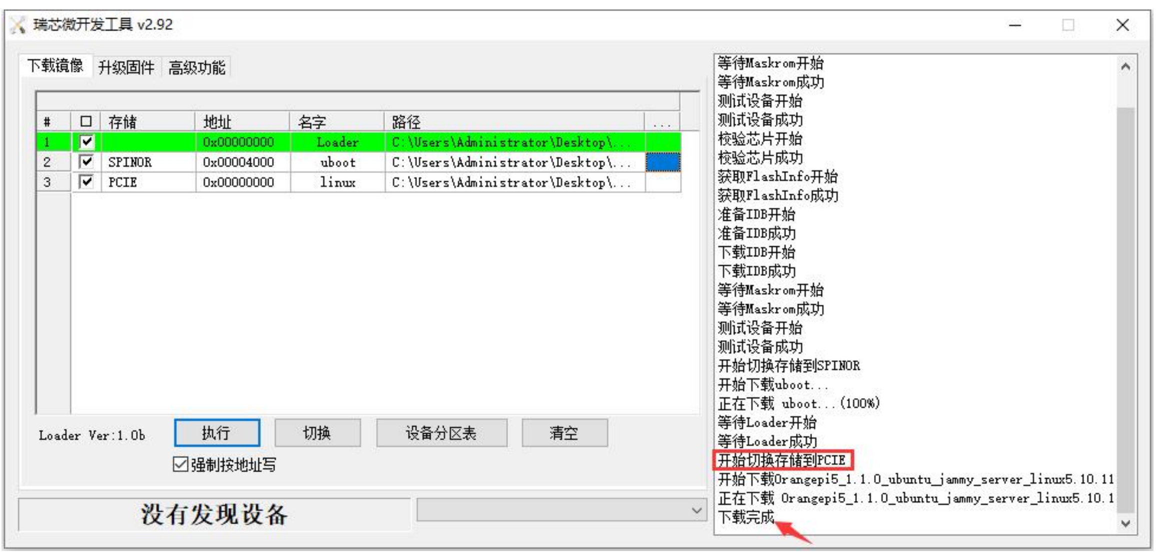
r. Then please check the option to **force writing by address**



s. Click the Execute button again to start burning the linux image to the SSD

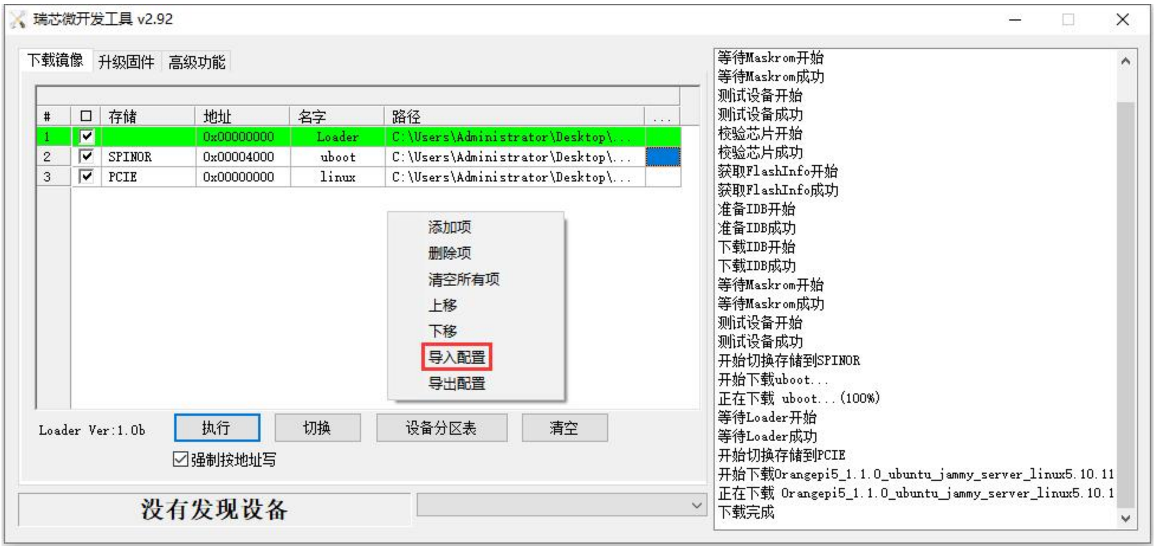


t. The log displayed after burning the linux image is shown in the figure below

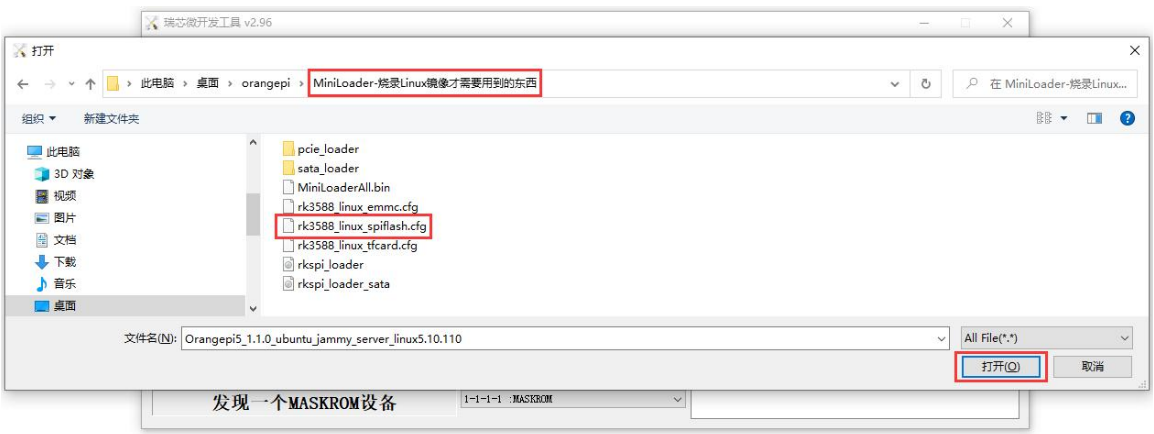


u. Then burn the uboot image required by the linux system in the SSD to SPIFlash, and **first enter the MaskROM mode again**.

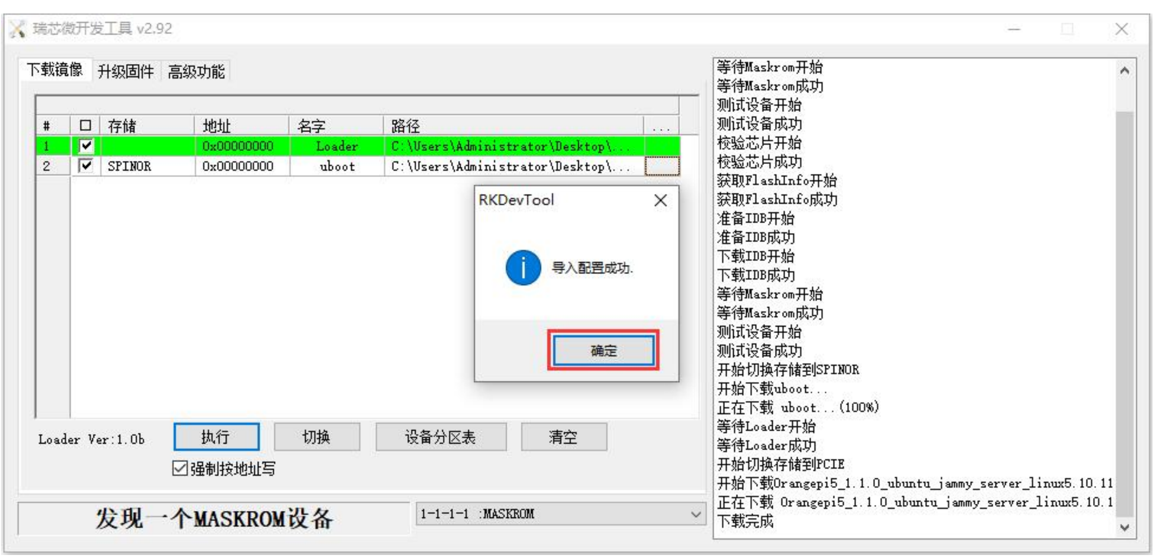
v. Then select the **import configuration** option



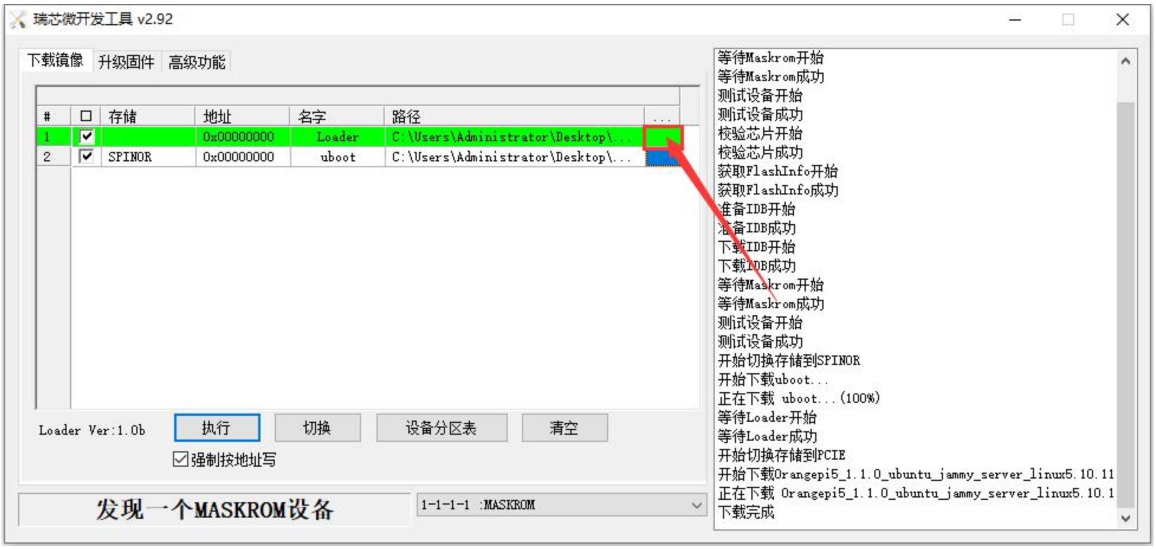
w. Then select the **rk3588_linux_spiflash.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and click **Open**.



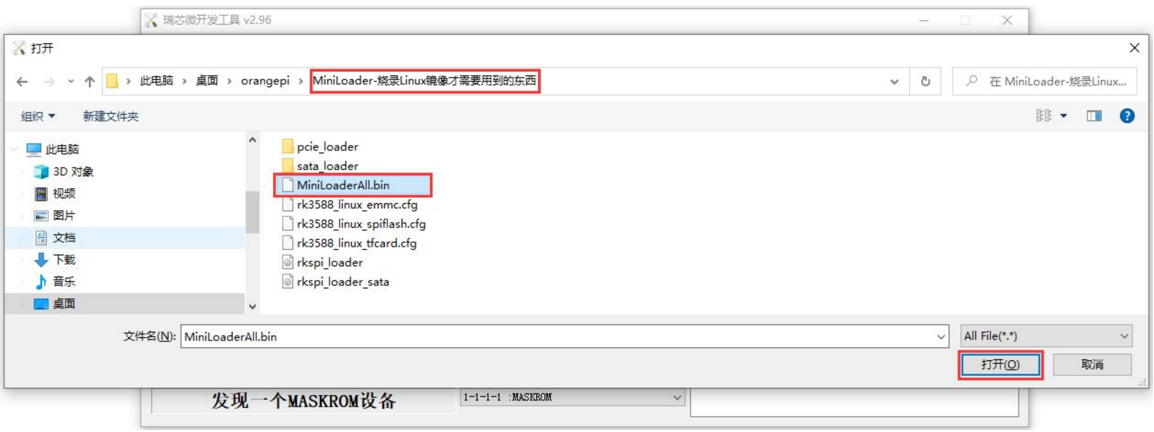
x. Then click **OK**



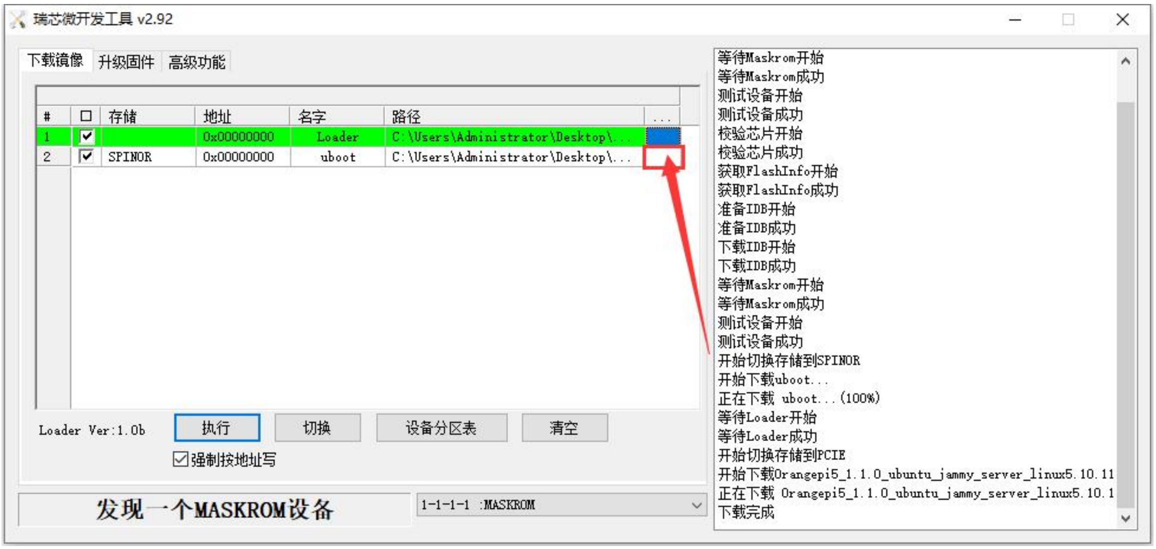
y. Then click the position shown in the figure below



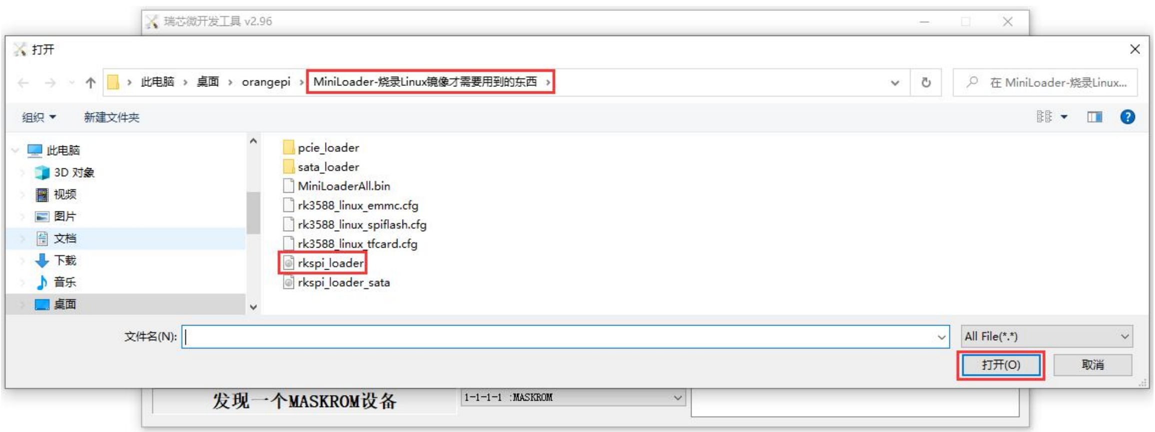
z. Select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click to open



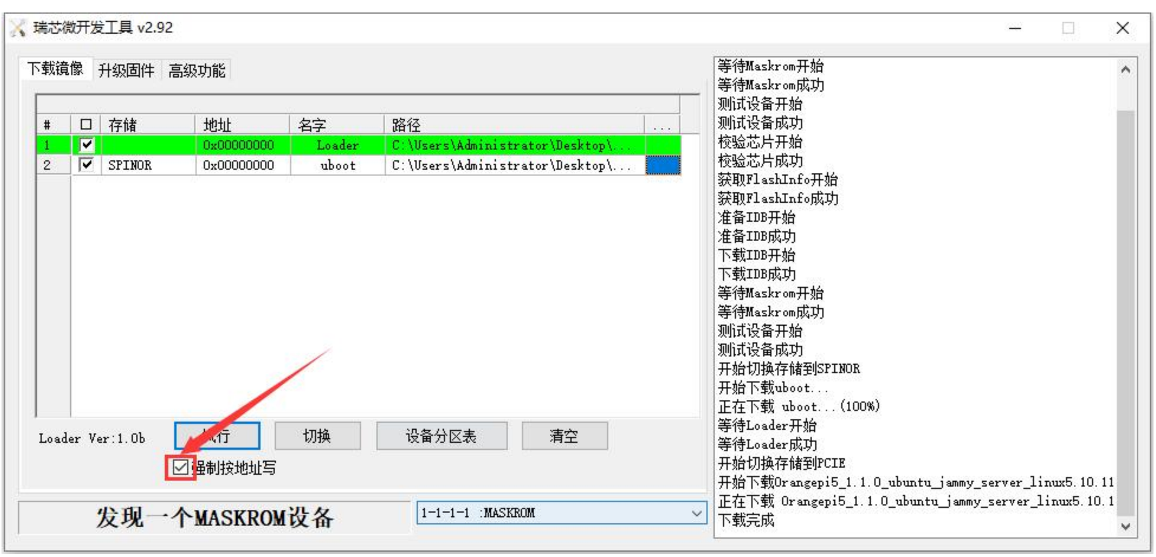
aa. Then click the location shown in the figure below



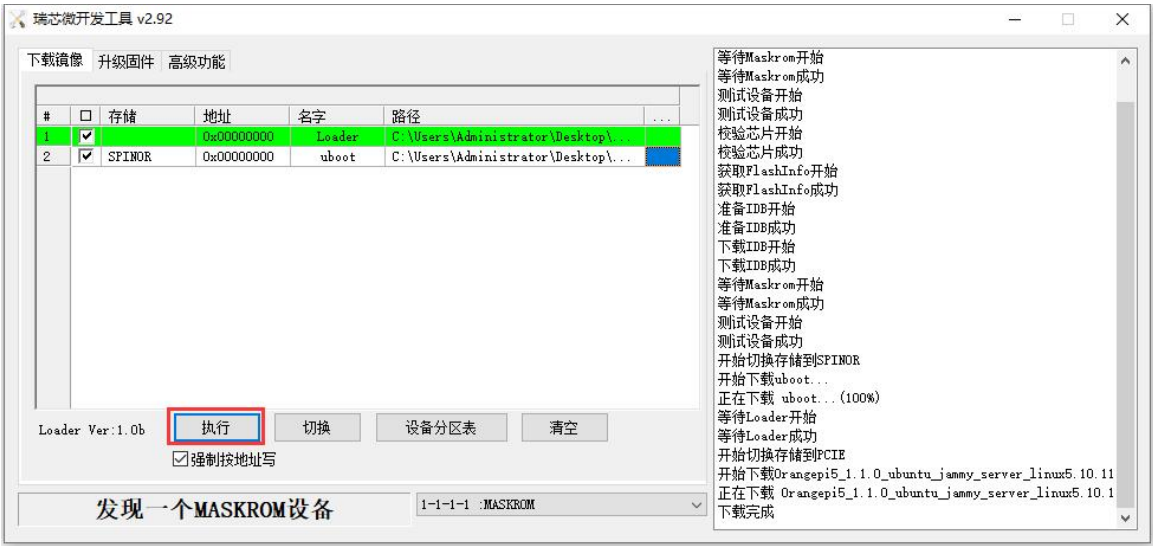
ab. Then select **rkspi_loader** in the **MiniLoader** folder downloaded earlier, and then click to **open**



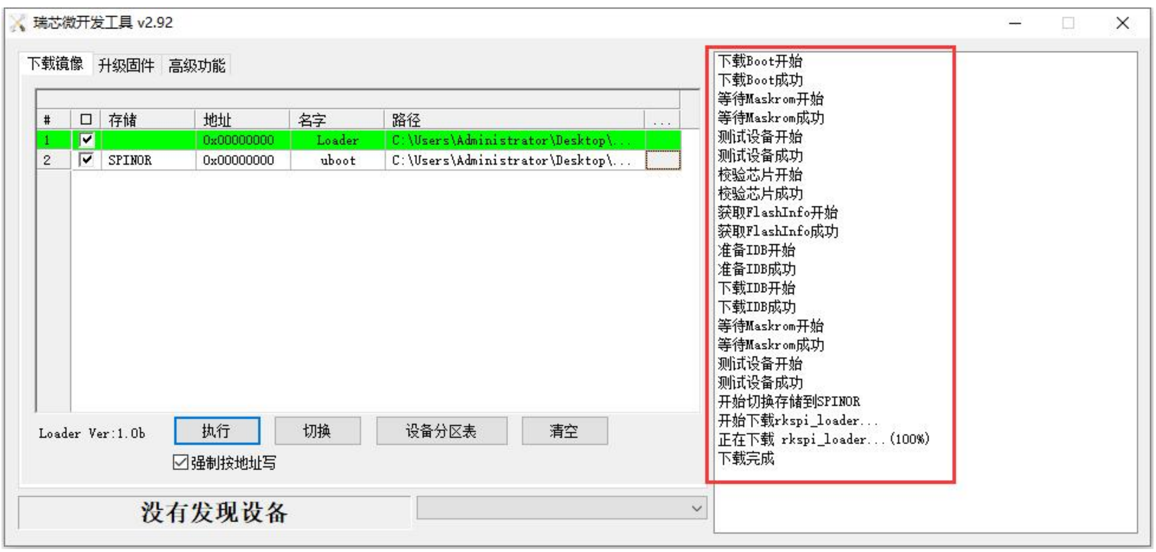
ac. Then please make sure that the option to **force writing by address** is ticked



ad. Click the Execute button again to start burning the u-boot image into SPIFlash



ae. The display log after burning the u-boot image is shown in the figure below.



af. After the u-boot image is burnt, it will automatically start the linux system in the SPIFlash+PCIe SSD. If it does not start normally, please power on and try again.

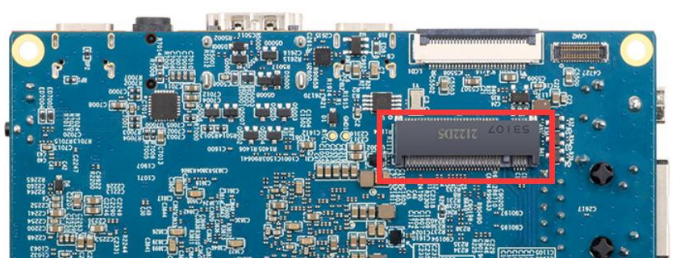
2. 5. 2. How to use the dd command to burn

1) First, you need to prepare an NVMe SSD. The PCIe supported by the M.2 slot of the development board is PCIe2.0x1, and the theoretical maximum speed is 500MB/s. PCIe3.0 and PCIe4.0 NVMe SSDs are also available, but the highest speed is only PCIe2.0x1 PCIe2.0x1.

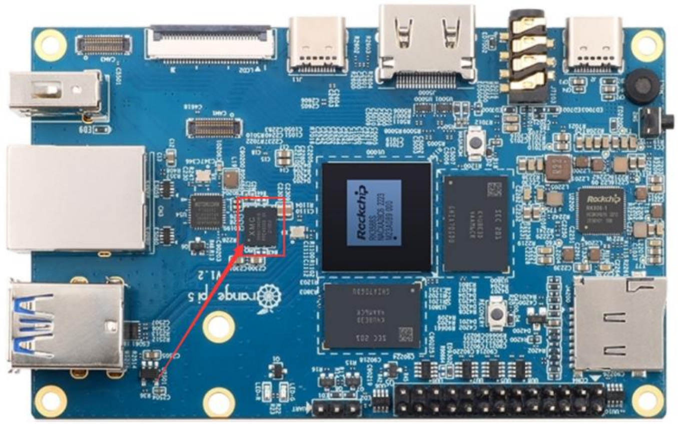
a. The M.2 2230 SSD is as follow

b. The M.2 2242 SSD is as follow

2) Then insert the NVMe SSD into the M.2 PCIe interface of the development board and fix it



3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting



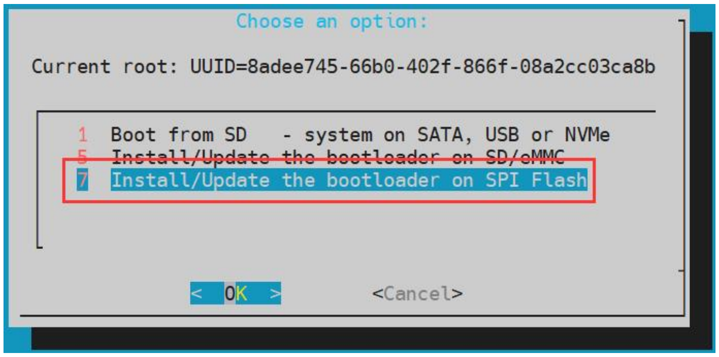
4) Burning the linux image to SPIFlash+NVMe SDD needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please refer to the instructions in the two sections of **the method of burning the Linux image to the TF card based on the Windows PC** and **the method of burning the Linux image to the TF card based on the Ubuntu PC**

5) After using the TF card to start the Linux system, we first burn the u-boot image into the SPI Flash

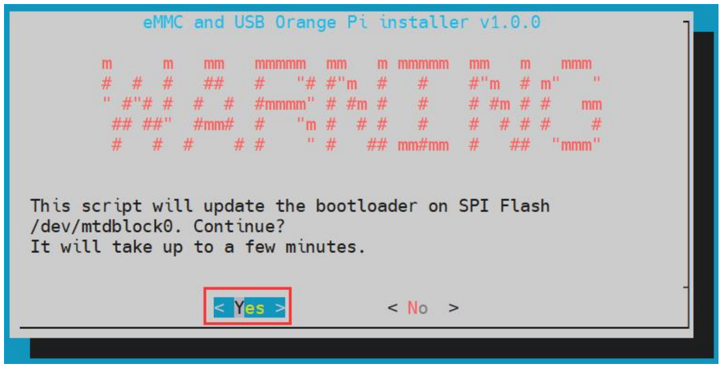
a. Run **nand-sata-install** first, **ordinary users remember to add sudo permission**

```
orangepi@orangepi:~$ sudo nand-sata-install
```

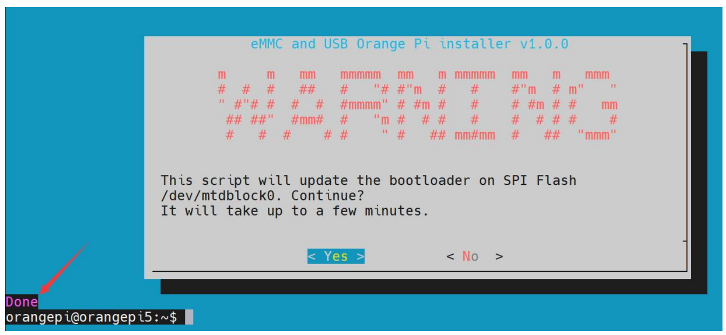
b. Then select **7 Install/Update ther bootloader on SPI Flash**



c. Then select**<Yes>**



d. Then please wait patiently for the burning to complete. After the burning is completed, the display will be as follows (a **Done** will be displayed in the lower left corner)

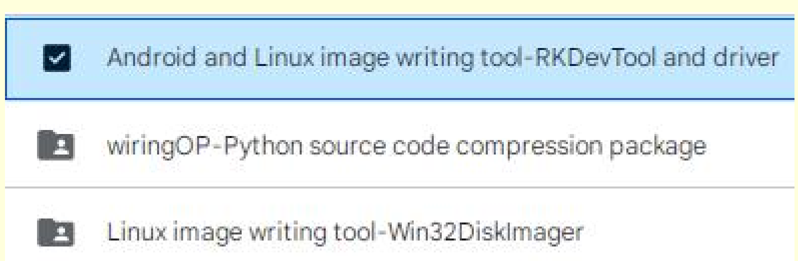


There is no nand-sata-install script in OPi OS Arch system, please use the following command to image u-boot to SPI Flash:

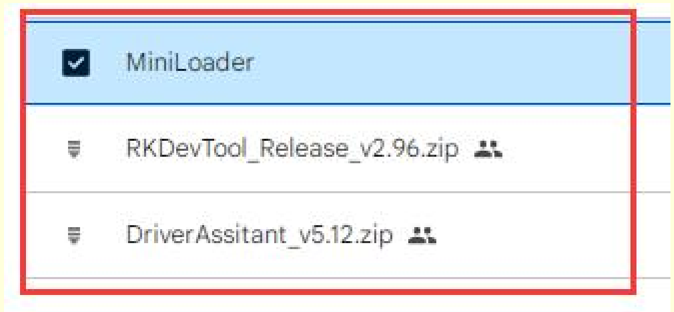
[orangepi@orangepi ~]$ **sudo dd if=/boot/rkspi_loader.img of=/dev/mtdblock0**

If you need to start the OpenWRT image, you need to download the latest version of u-boot image from the official website, and then burn it into SPI Flash. The download steps are as follows:

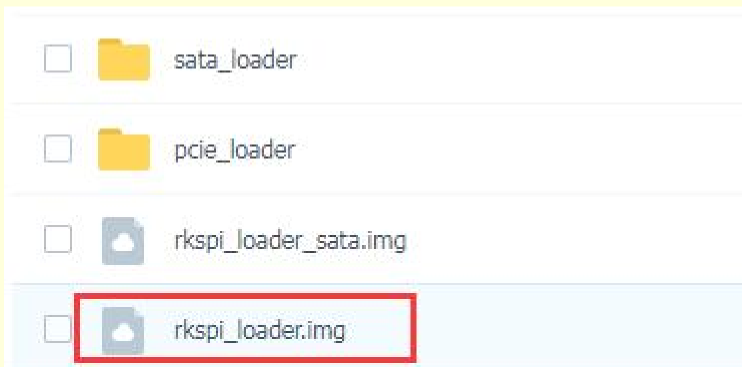
a. First enter the data download page of the development board, then select the official tool on the data download page, and then enter the folder below



b. Then choose to enter the following directory



c. Then download rkspi_loader.img



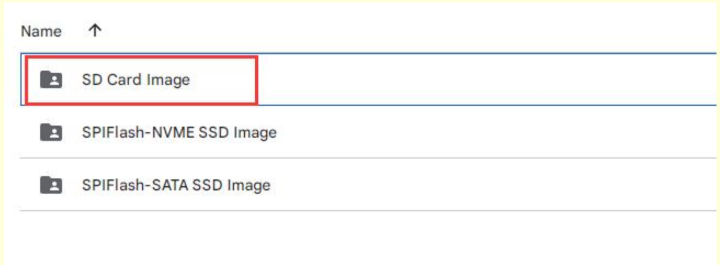
d. Then upload rkspi_loader.img to the Ubuntu or Debian or OPi OS Arch system of the development board. For the upload method, please refer to the instructions in the method of uploading files to the Linux system of the development board.

Finally, execute the following command to burn the u-boot image into SPI Flash (note that this command is executed in Ubuntu, Debian, or OPi OS Arch):

orangepi@orangepi:~$ **sudo dd if=rkspi_loader.img of=/dev/mtdblock0**

6) Then upload the linux image file (Debian or Ubuntu image downloaded from the official website) to the TF card. For the method of uploading the linux image file to the development board, please refer to the description in the section of **the method of uploading files to the development board Linux system**

Note that if you download an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the "TF card boot image" folder.



7) After uploading the image to the linux system of the development board, we enter the storage path of the image file in the command line of the linux system of the development board. For example, I store the linux image of the development board in the **/home/orangepi/Desktop** directory Download it, and then enter the **/home/orangepi/Desktop** directory to see the uploaded image file

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

How to enter the command line of the development board linux system?

1. For the method of using the serial port to log in to the terminal, please refer to the instructions in the section on how to use the debugging serial port

2. Use ssh to remotely log in to the Linux system, please refer to the instructions in the section of SSH remote login to the development board.

3. If HDMI, LCD and other display screens are connected, you can open a command line terminal on the desktop**.**

8) Next, let's confirm that the NVMe SSD has been recognized by the development board's linux. If the NVMe SSD is recognized normally, use the **sudo fdisk -l** command to see **nvme** related information.

```
orangepi@orangepi:~/Desktop$ sudo fdisk -l | grep "nvme0n1"
Disk /dev/nvme0n1: 1.86 TiB, 2048408248320 bytes, 4000797360 sectors
```

Use the **lspci** command to see an NVMe-related PCI device

```
orangepi@orangepi:~/Desktop$ lspci
```

```
0004:40:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3588 (rev 01)
0004:41:00.0 Non-Volatile memory controller: MAXIO Technology (Hangzhou) Ltd. NVMe SSD Controller MAP1202 (rev 01)
```

9) Then we can use the dd command to clear the NVMe SSD(Optional)

```
orangepi@orangepi5:~/Desktop$ sudo dd bs=1M if=/dev/zero of=/dev/nvme0n1 count=2000 status=progress
orangepi@orangepi5:~/Desktop$ sudo sync
```

10) Then you can use the dd command to burn the linux image of the development board to the NVMe SSD.

a. 下面 In the following command, the **if**= parameter is followed by the full path where the linux image is stored + the name of the Linux image (such as **the name of /home/orangepi/Desktop/Linux image**). Because we have entered the path of the linux image above, we only need to fill in the name of the Linux image.
b. Please do not copy the linux image name in the following command, but replace it with the actual image name (because the version number of the image may be updated)

```
sudo dd bs=1M if=Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/nvme0n1 status=progress

sudo sync
```

Note, if you upload a .7z or .xz linux image compressed file, please remember to decompress it before using the dd command to burn.

The detailed description of all parameters of the dd command and more usage can be viewed by executing the man dd command in the linux system.

11) After successfully burning the linux image of the development board to the NVMe SSD, you can use the poweroff command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in SPIFlash+NVMe SSD will be started.

12) After starting the system in the NVMe SSD, use the **df -h** command to see the actual hard disk capacityNVMe SSD.

a. 128GB NVMe SSD

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G  8.0K  3.8G   1% /dev
tmpfs           769M  1.4M  768M   1% /run
/dev/nvme0n1p2  118G  5.8G  111G   5% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.8G   16K  3.8G   1% /tmp
/dev/nvme0n1p1  256M   90M  166M  36% /boot
/dev/zram1      194M  9.9M  170M   6% /var/log
tmpfs           769M   60K  769M   1% /run/user/1000
tmpfs           769M   48K  769M   1% /run/user/0
```

b. 2TB NVMe SSD

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G  8.0K  3.8G   1% /dev
tmpfs           769M  1.4M  768M   1% /run
/dev/nvme0n1p2  1.9T  4.1G  1.8T   1% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/zram2      3.7G   76K  3.5G   1% /tmp
/dev/nvme0n1p1  256M   90M  166M  36% /boot
/dev/zram1      194M   15M  165M   9% /var/log
tmpfs           769M   60K  769M   1% /run/user/1000
tmpfs           769M   48K  769M   1% /run/user/0
```

13) When the TF card and NVMe SSD are programmed with exactly the same system, **if both the TF card and NVMe SSD are inserted into the development board, power on the development board at this time, and u-boot will give priority to starting the system in the TF card**. However, since the systems in the TF card and NVMe SSD are exactly the same, the UUIDs of the **/boot** partition and the **rootfs** partition in the two storage devices are also the same, which may cause the partition in the NVMe SSD to be loaded when the TF card starts. Running the script below resolves this issue

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

Exactly the same system means that the image name is exactly the same. Even if they are all Debian11 systems, the versions are different

There is no fix_mmc_ssd.sh script in OPi OS Arch system.

2. 5. 3. How to use the balenaEtcher to burn

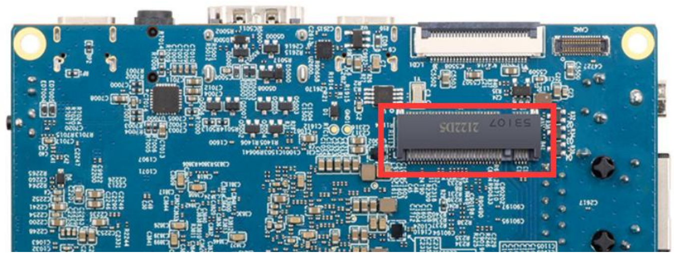
Please do not use this method for OPi OS Arch system.

1) First, you need to prepare an NVMe SSD. The PCIe supported by the M.2 slot of the development board is PCIe2.0x1, and the theoretical maximum speed is 500MB/s. PCIe3.0 and PCIe4.0 NVMe SSDs are also available, but the highest speed is only PCIe2.0x1.

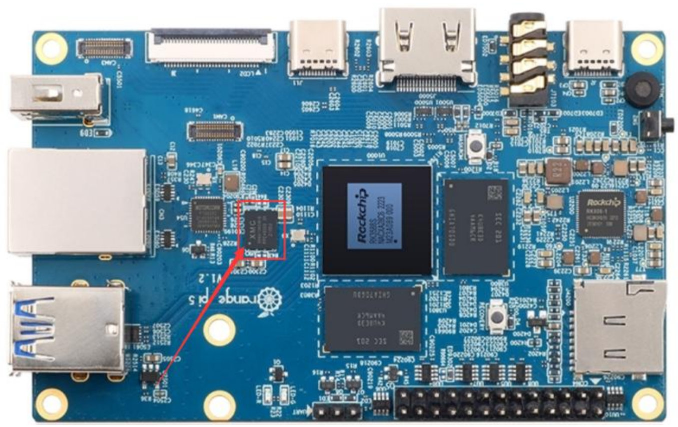
a. The M.2 2230 SSD is as follows

b. The M.2 2242 SSD is as follows

2) Then insert the NVMe SSD into the M.2 PCIe interface of the development board and fix it



3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming



4) Burning the linux image to SPIFlash+NVMe SDD needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please refer to the instructions in the two sections of the method of burning **the Linux image to the TF card based on the Windows PC** and **the method of burning the Linux image to the TF card based on the Ubuntu PC**.

5) After booting into the linux system in the TF card, please confirm that the NVMe SSD has been properly recognized by the linux of the development board. If the NVMe SSD is recognized normally, use the **sudo fdisk -l** command to see **nvme**-related information.

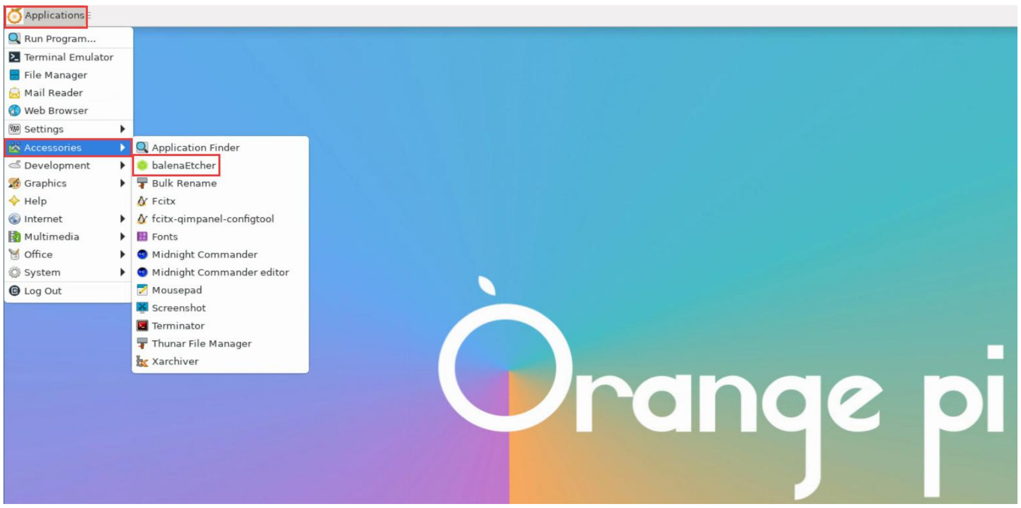
```
orangepi@orangepi:~/Desktop$ sudo fdisk -l | grep "nvme0n1"
Disk /dev/nvme0n1: 1.86 TiB, 2048408248320 bytes, 4000797360 sectors
```

Use the **lspci** command to see an NVMe-related PCI device

```
orangepi@orangepi:~/Desktop$ lspci
0004:40:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3588 (rev 01)
```

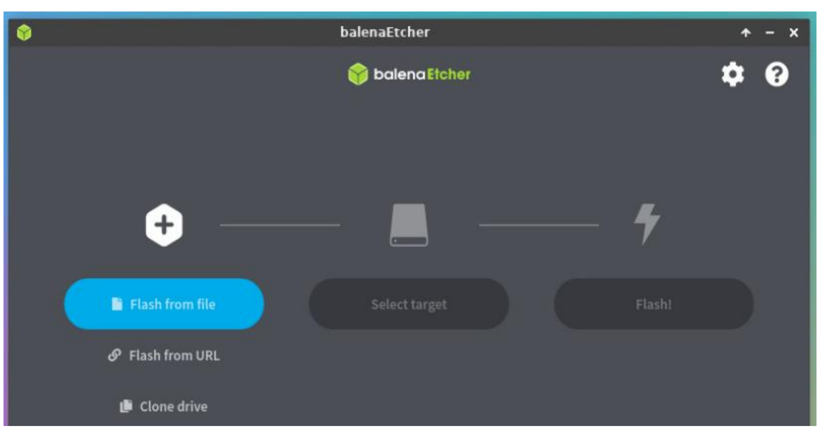
```
0004:41:00.0 Non-Volatile memory controller: MAXIO Technology (Hangzhou) Ltd.
NVMe SSD Controller MAP1202 (rev 01)
```

6) The balenaEtcher has been pre-installed in the linux image, and the opening method is as follows:



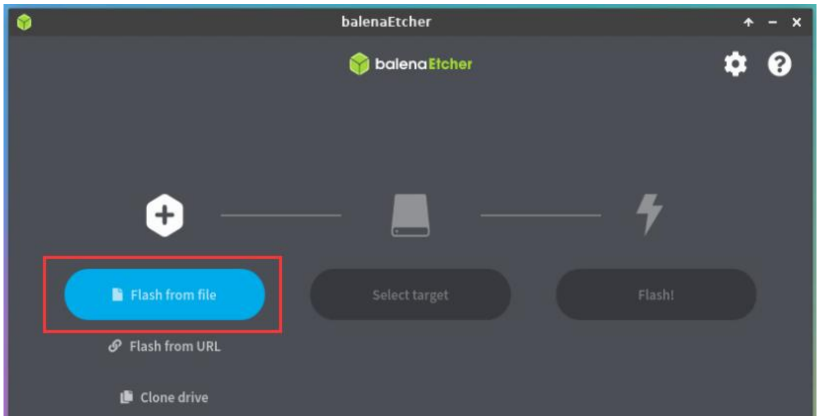
If it is not pre-installed, for how to download and install the arm64 version of balenaEtcher, please refer to **the instructions in the section on how to download and install the arm64 version of balenaEtcher.**

7) The interface after balenaEtcher is opened is as follows：

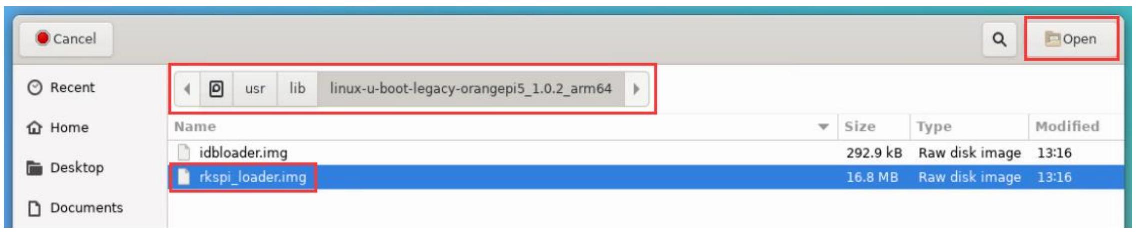


8) The method of using balenaEtcher to burn u-boot to the SPI Flash of the development board is as follows:

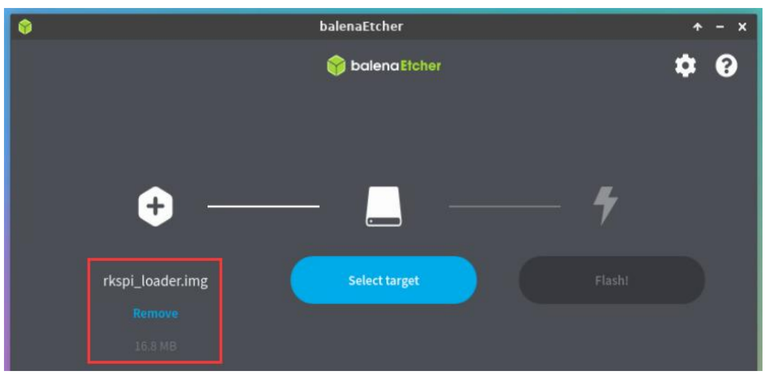
a. First click **Flash from file**



b. Then enter the **/usr/lib/linux-u-boot-legacy-orangepi5_1.x.x_arm64** directory, select**rkspi_loader.img**, and click Open to open

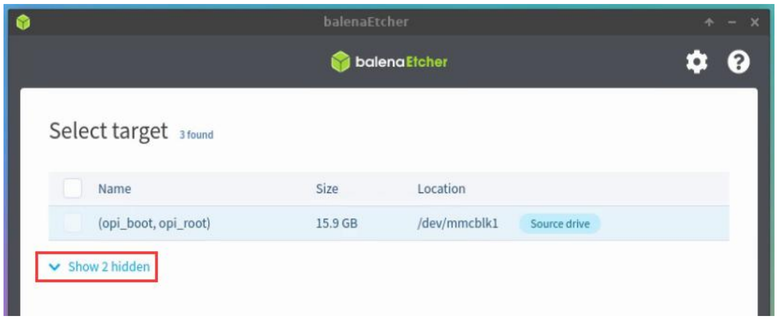


c. The interface after opening **rkspi_loader.img** is as follow：

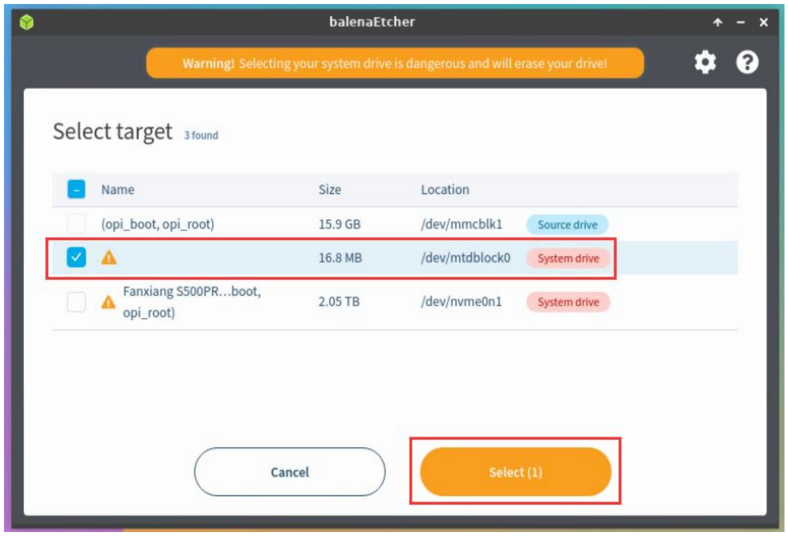


d. Then click **Select target**

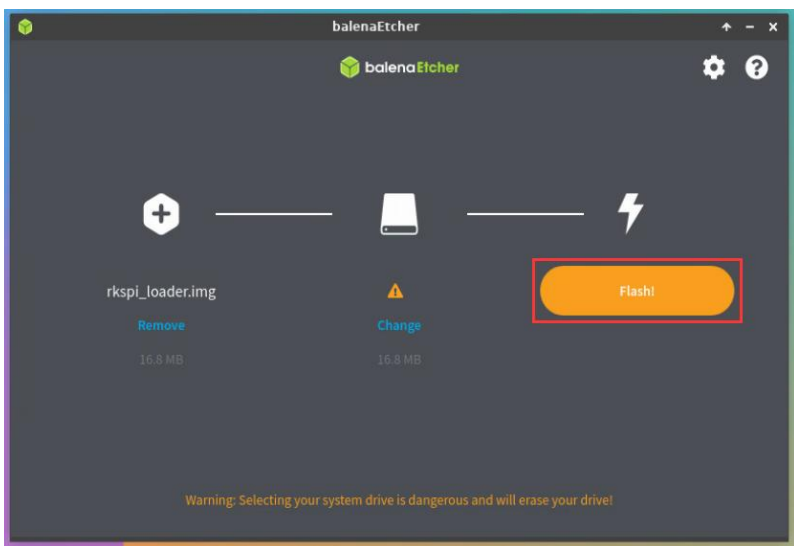
e. Then click **Show 2 hidden**to open more options for storage devices.



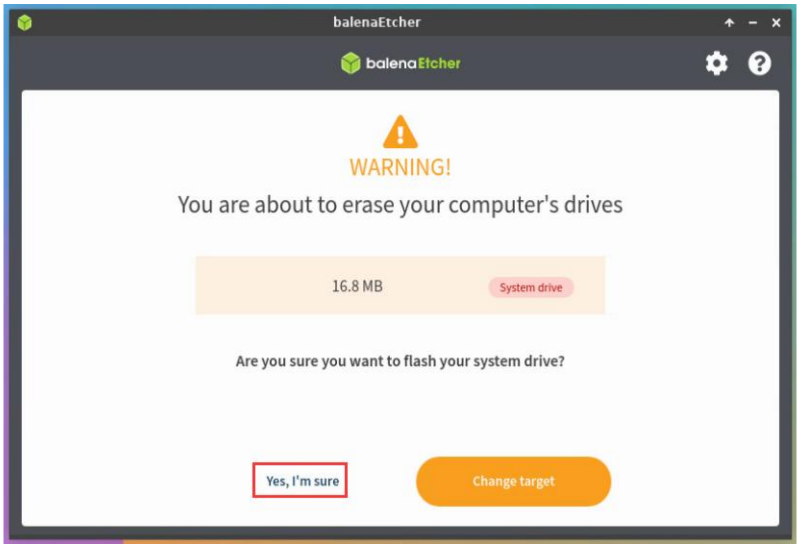
f. Then select the device name of SPI Flash **/dev/mtdblock0**, and click **Select**



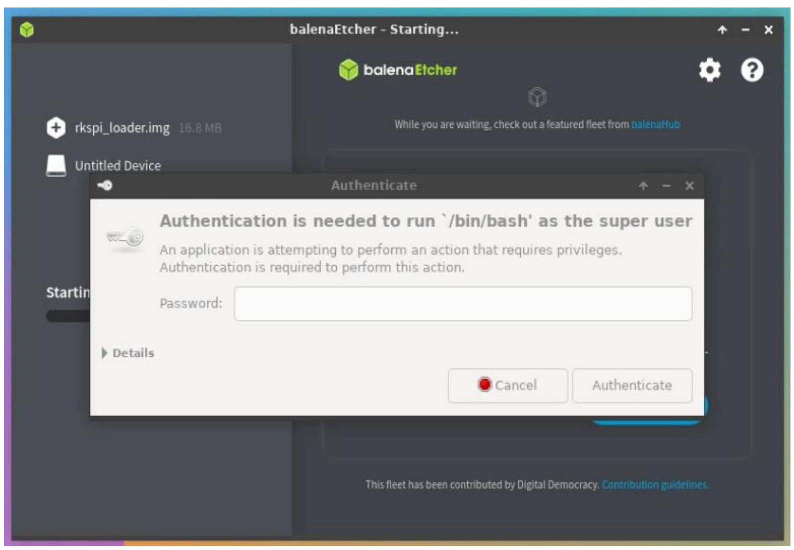
g. Then click **Flash**



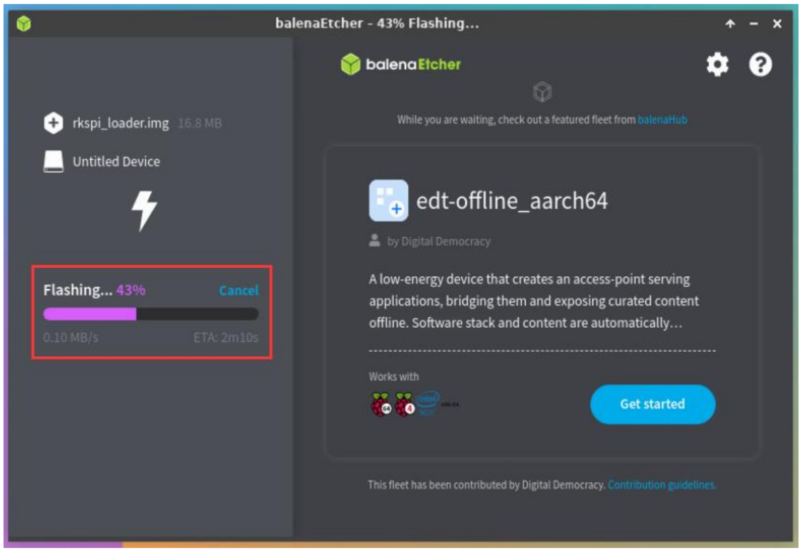
h. And then click **Yes, I'm sure**



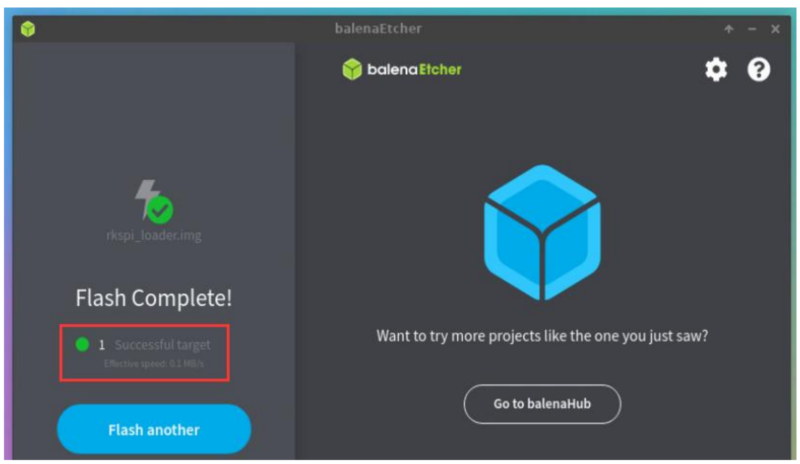
i. Then enter the password **orangepi** of the development board linux system, and it will start burning the u-boot image into the SPI F.



j. The display of the burning process is as follow：

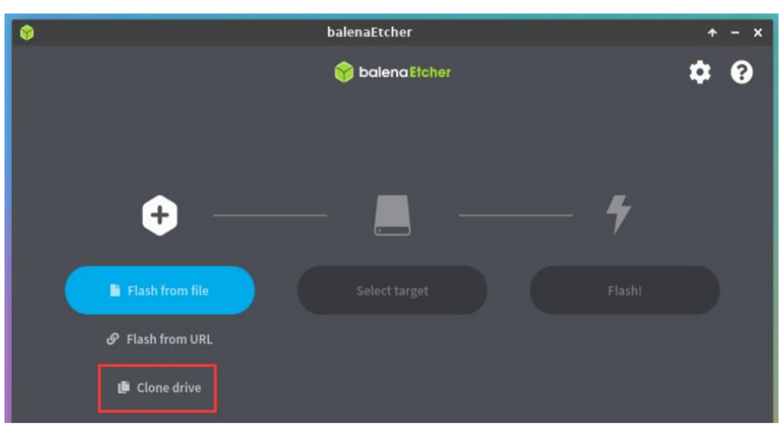


k. The display after burning is as follow:

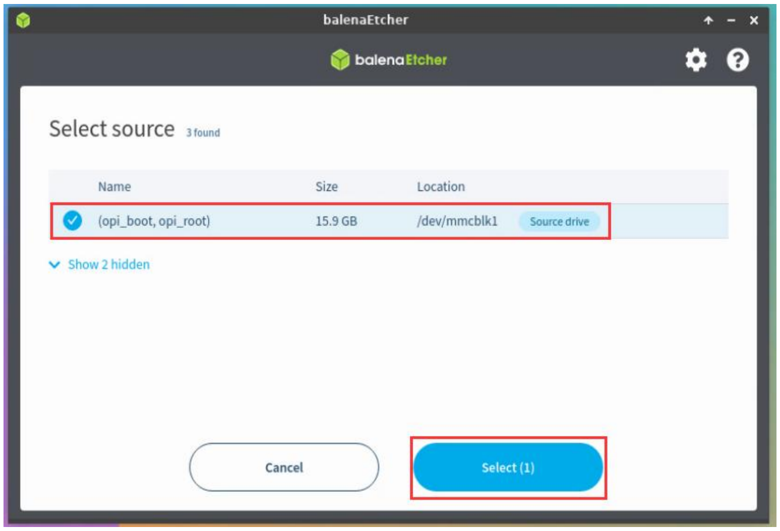


9) The method of burning the linux system in the TF card to the NVMe SSD (this method is equivalent to cloning the system in the TF card to the NVMe SSD)

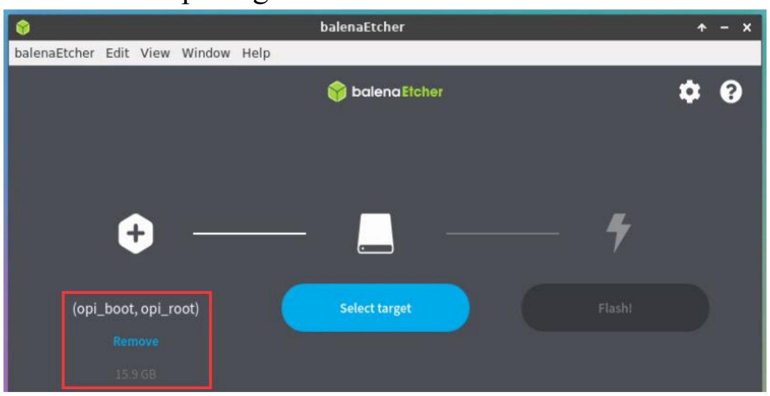
a. First click **Clone drive**



b. Then select the device name of the TF card **/dev/mmcblk1**



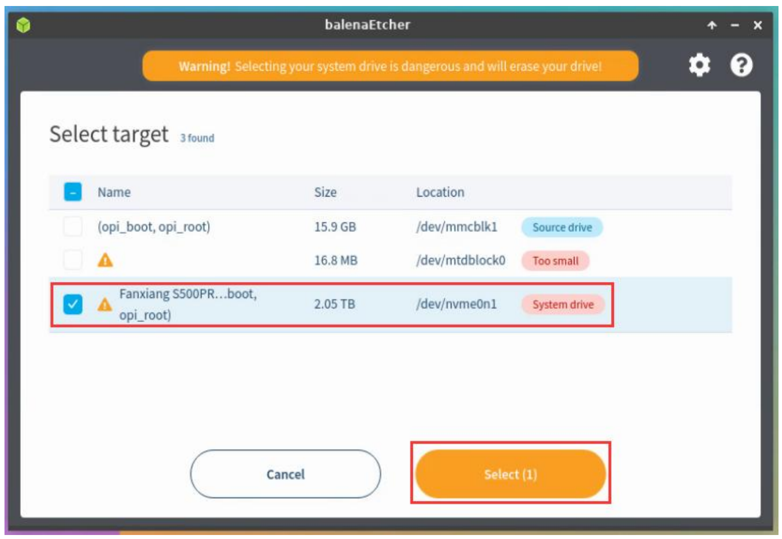
c. The interface after opening the TF card is as follows：



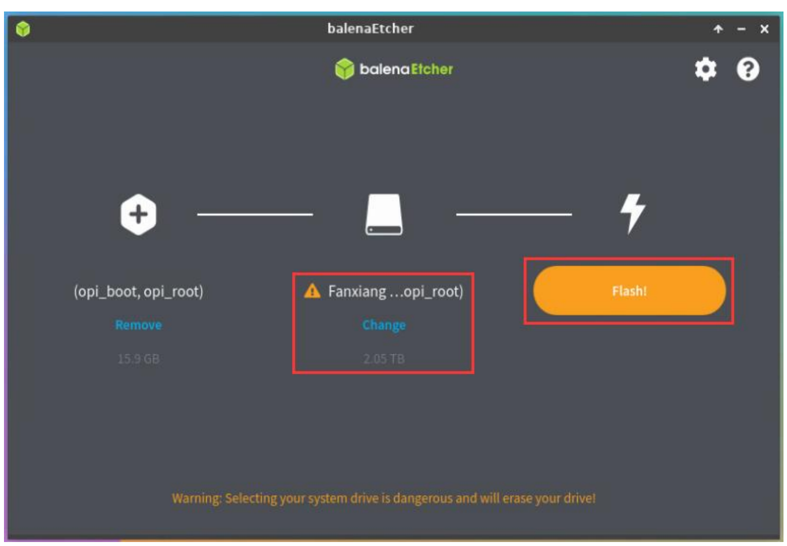
d. Then click **Select target**

e. Then click **Show 2 hidden**to open more options for storage devices

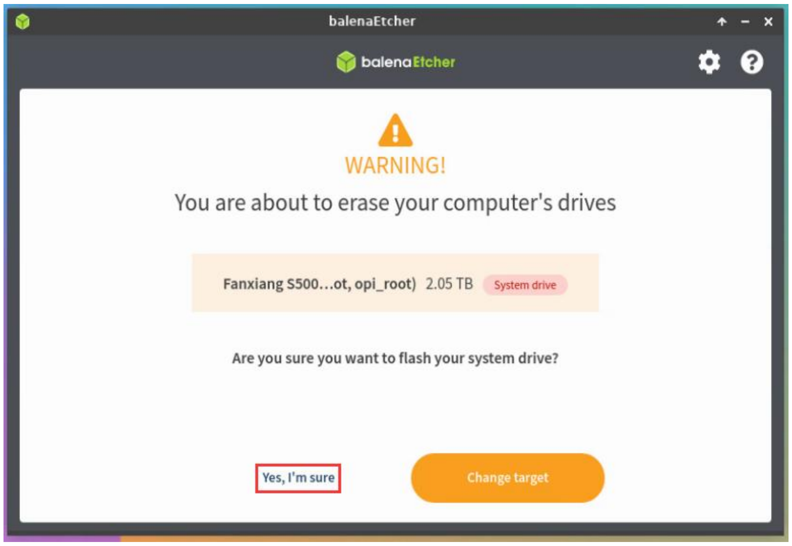
f. Then select the device name of the NVMe SSD **/dev/nvme0n1,** and click **Select**



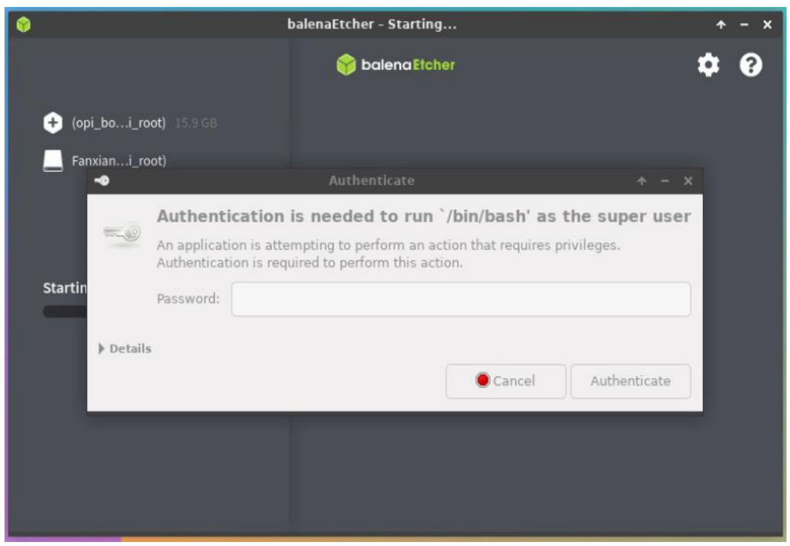
g. Then click **Flash**



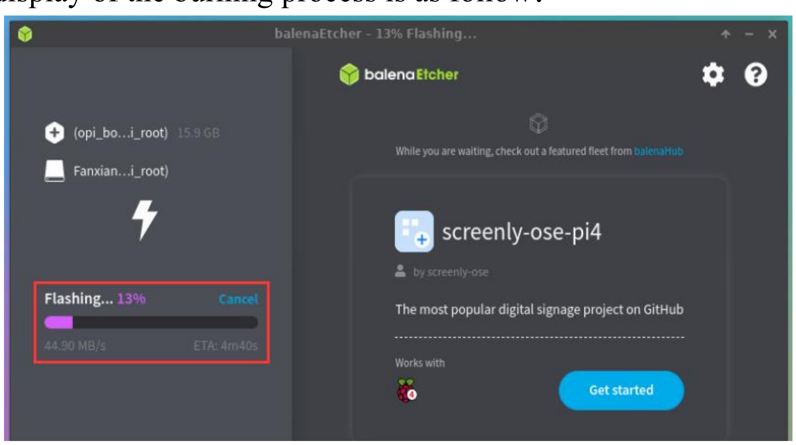
h. Then click **Yes, I'm sure**

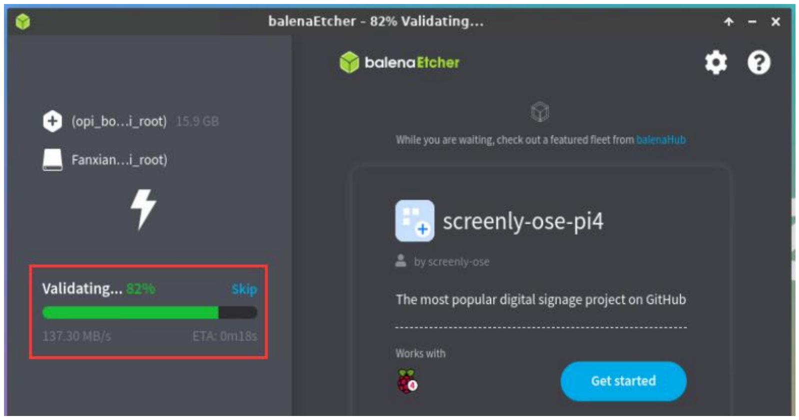


i. Then enter the password orangepi of the linux system on the development board, and it will start burning the linux image to the SSD

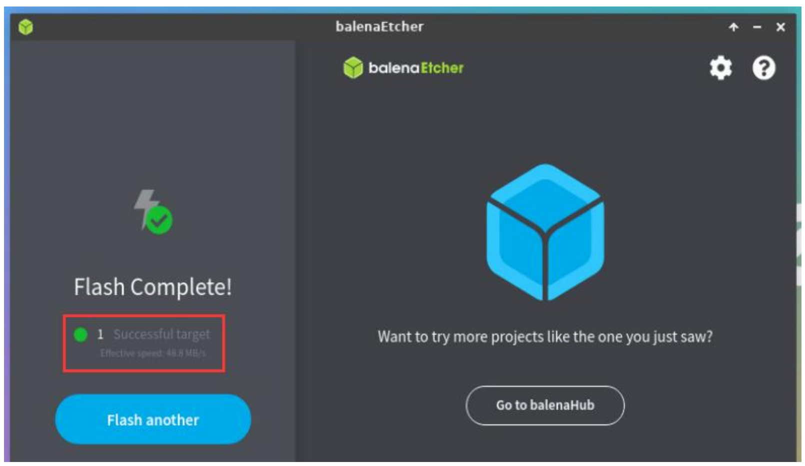


j. The display of the burning process is as follow：



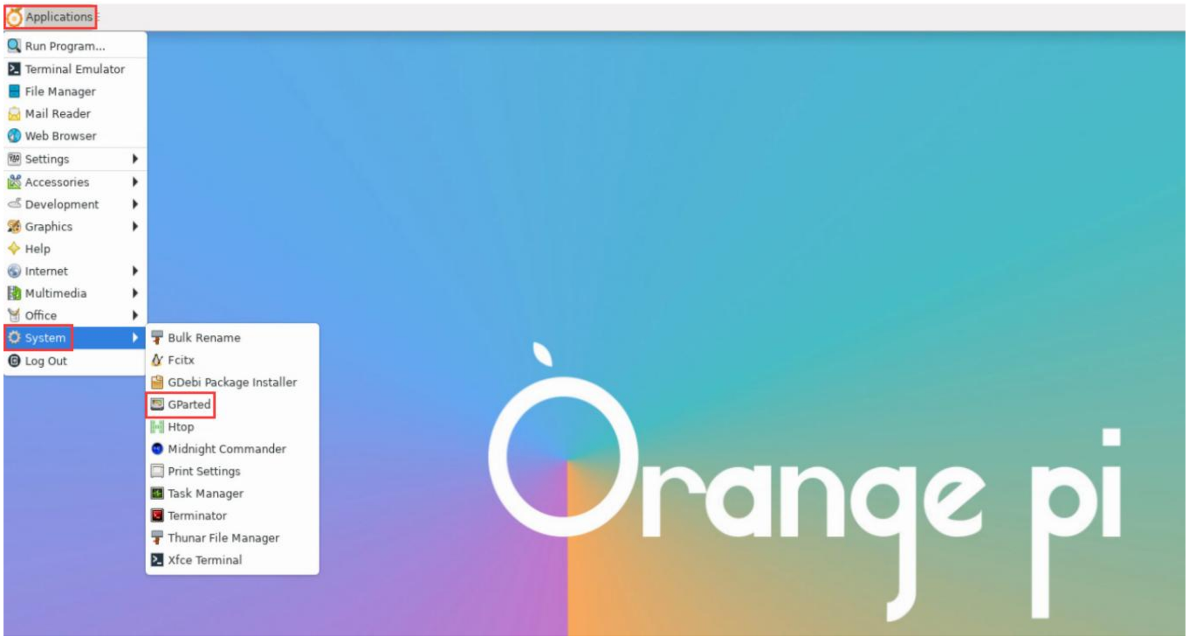


k. The display after burning is as follow:



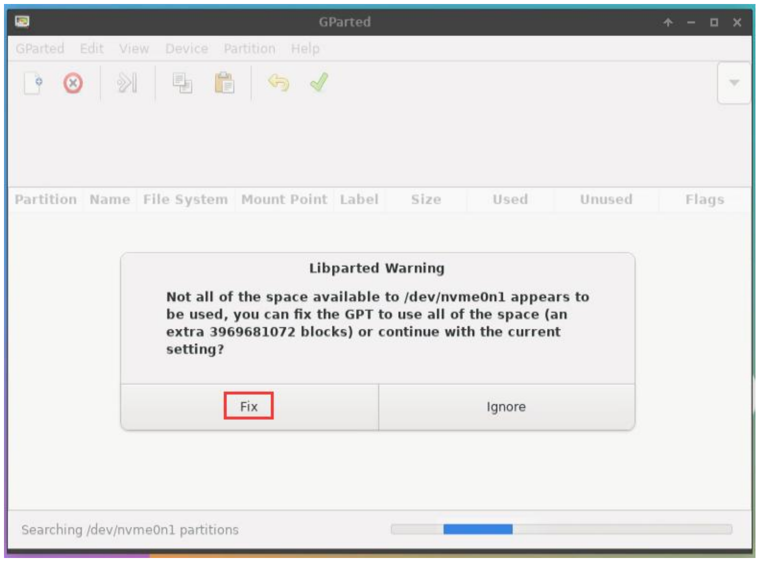
l. Then you need to expand the capacity of the rootfs partition in the NVMe SSD. The steps are as follows:

a) First open **GParted**

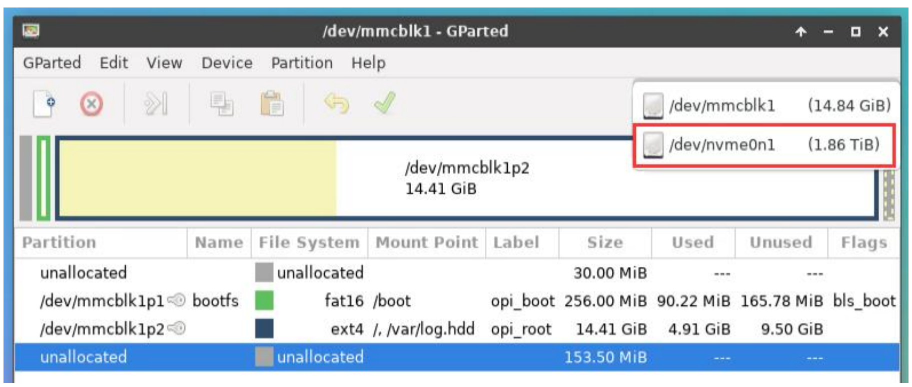


b) Then enter the password orangepi of the linux system, and click **Authenticate**

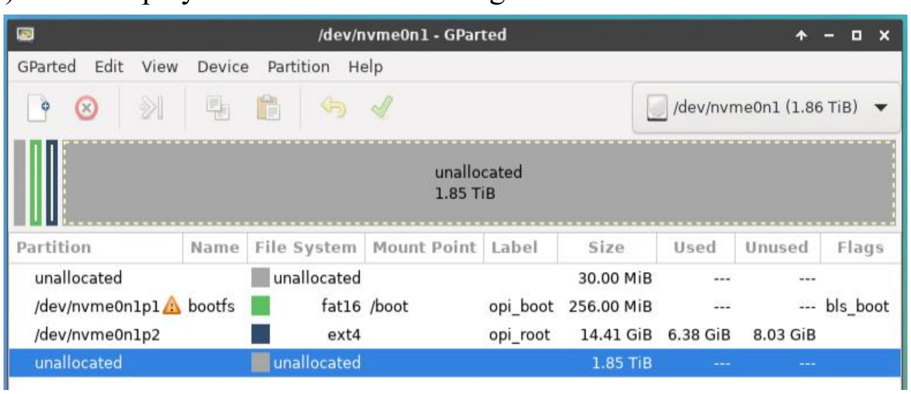
c) Then click **Fix**



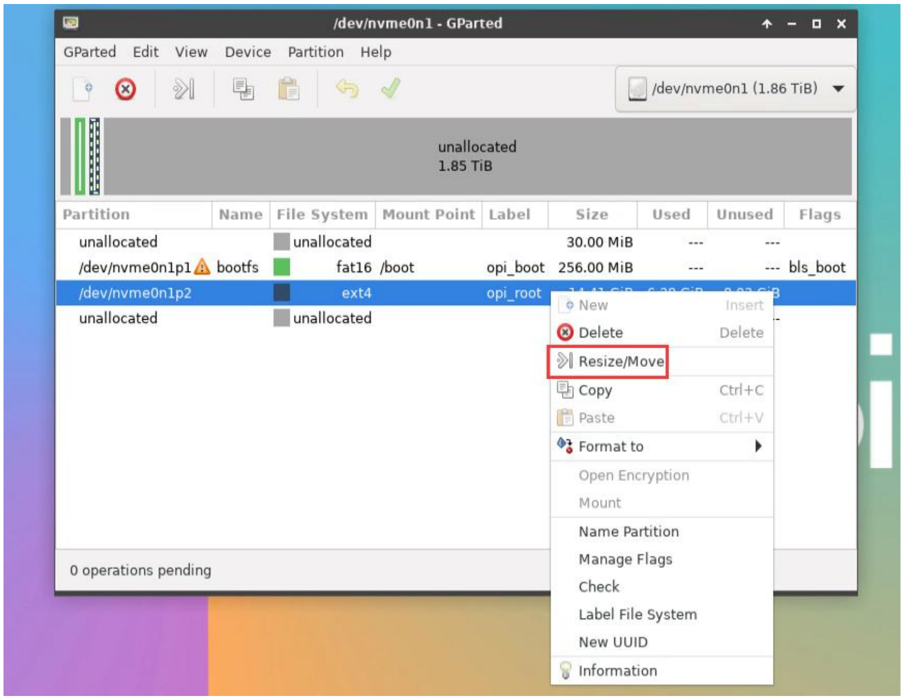
d) Then select NVMe SSD



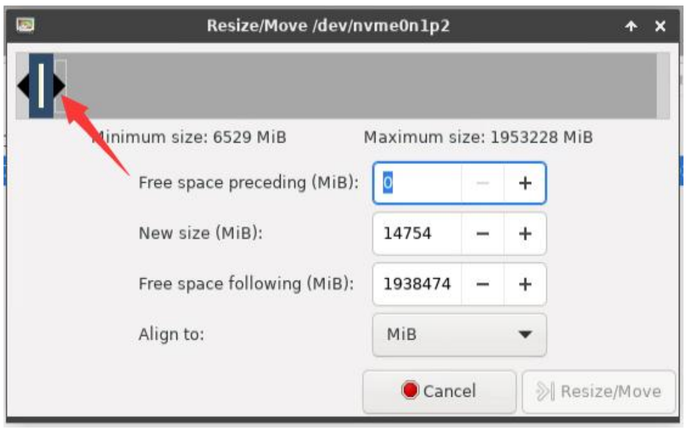
e) The display interface after selecting NVMe SSD is as follow：



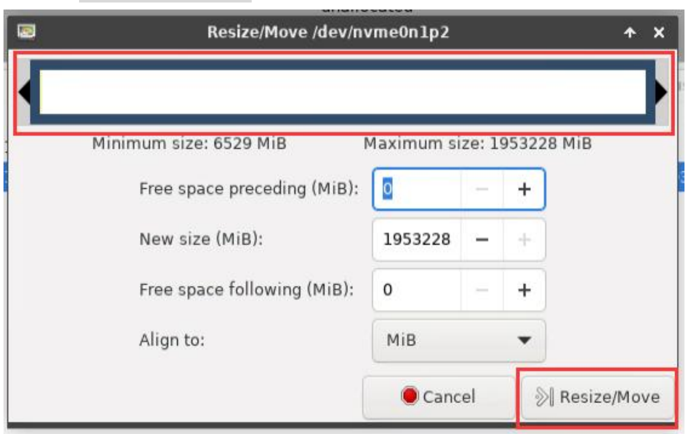
f) Then select the **/dev/nvme0n1p2** partition, click the right button again, and then select **Resize/Move**



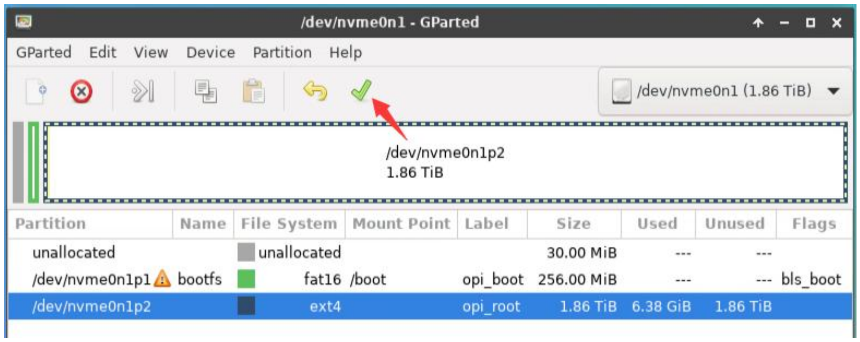
g) Then drag the capacity to the maximum at the position shown in the figure below



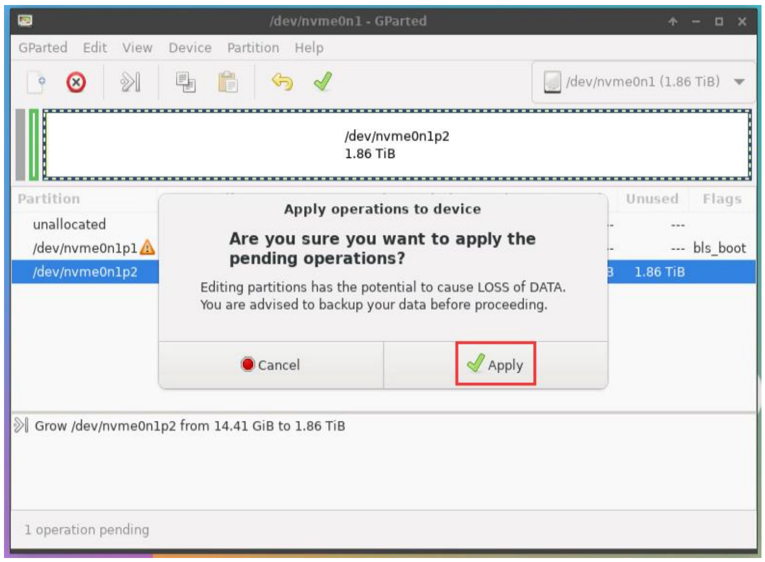
h) Then click **Resize/Move**



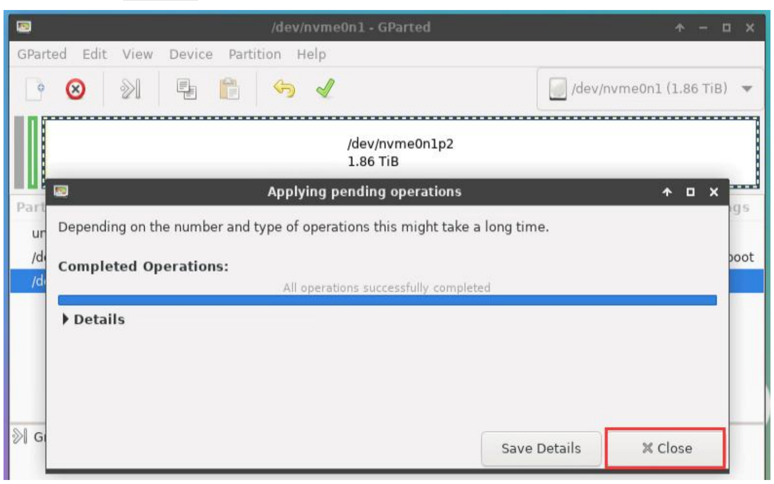
i) Then click the green√ in the picture



j) Then Click **Apply**



k) Then click **Close** to close

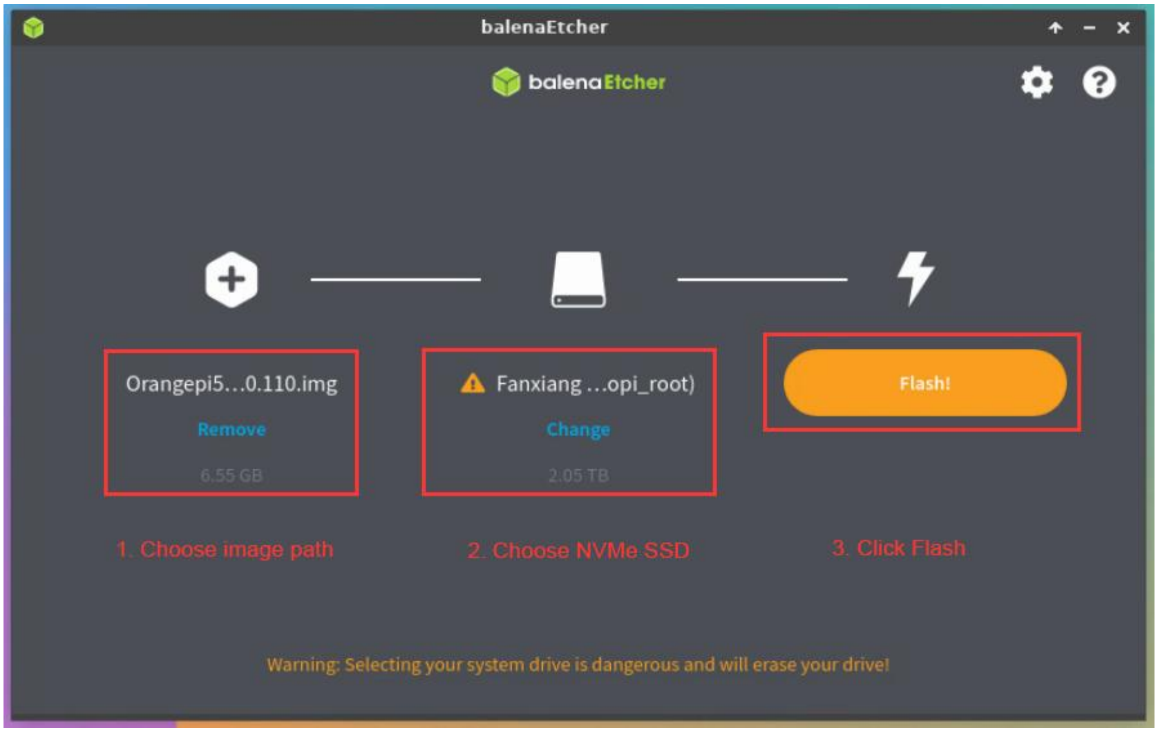


m. At this point, you can use the **sudo poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in SPIFlash+NVMe SSD will be started.

10) Step **9)** is to clone the system in the TF card to the NMVe SSD. We can also directly burn the linux image file to the NVMe SSD. Here are the step

a. Upload the linux image file to the linux system of the development board

b. Then use balenaEtcher to burn.



c. After using this method to burn the image, there is no need to manually expand the capacity, and it will automatically expand the capacity at the first startup**.**

2. 6. How to write Linux image to SPIFlash+SATA SSD

Note that the Linux image mentioned here specifically refers to the image of Linux distributions such as Debian or Ubuntu downloaded from the Orange Pi data download page**.**

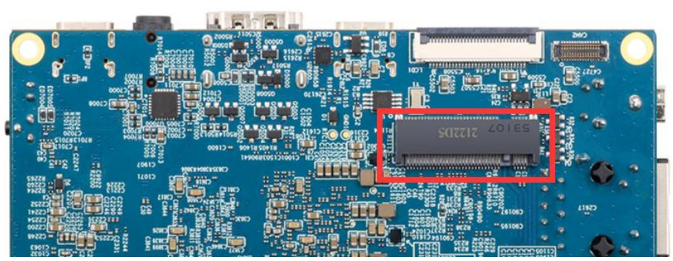
2. 6. 1. How to use the dd command to burn

1) First, you need to prepare a SATA SSD solid state drive

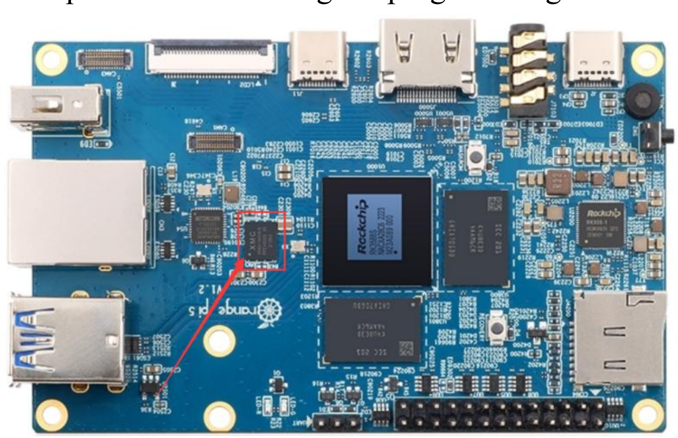
a. The M.2 2242 SSD is as folloews

b. M.2 The 2280 specification SSD is as follows (the 2280 specification SATA SSD can also be used, but the SSD will exceed the development board after being inserted into the development board

2) Then insert the SSD into the M.2 interface of the development board and fix it



3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming



4) Burning the linux image to SPIFlash+SDD needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF

card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please refer to the instructions in the two sections of t**he method of burning the Linux image to the TF card based on the Windows PC** and **the method of burning the Linux image to the TF card based on the on the Ubuntu PC**.

5) After using the TF card to start the Linux system, we first burn the u-boot image dedicated to the sata ssd into the SPI Flash

a. sata ssd startup dedicated u-boot image storage path is:

```
/usr/share/orangepi5/rkspi_loader_sata.img
```

b. Make sure that **rkspi_loader_sata.img** exists in the Linux system, and then use the following command to burn it into the SPIFlash of the development board

```
orangepi@orangepi:~$ cd /usr/share/orangepi5/
orangepi@orangepi:~$ sudo dd if=rkspi_loader_sata.img of=/dev/mtdblock0
orangepi@orangepi:~$ sudo sync
```

The storage path of the u-boot image of the OPi OS Arch system is somewhat different, as shown below:

e. OPi OS Arch system sata ssd boot dedicated u-boot image storage path is:

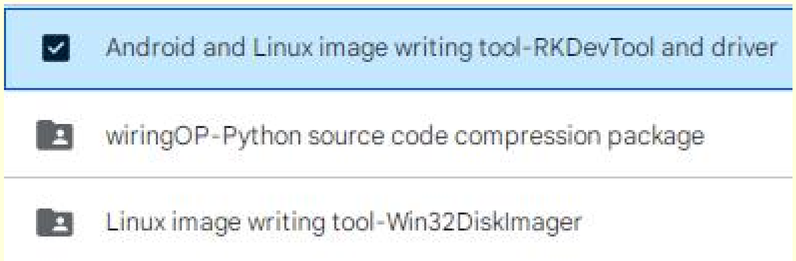
/boot/rkspi_loader_sata.img

f. Make sure**rkspi_loader_sata.img** exists in the OPi OS Arch system, and then use the following command to burn it into the SPIFlash of the development board:

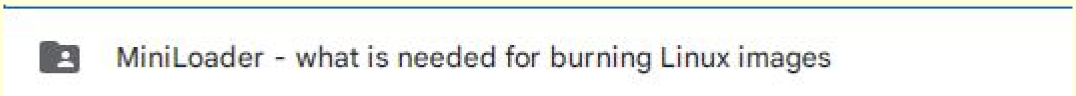
```
[orangepi@orangepi boot]$ cd /boot/
[orangepi@orangepi boot]$ sudo dd if=rkspi_loader_sata.img of=/dev/mtdblock0
[orangepi@orangepi boot]$ sudo sync
```

If you need to start the OpenWRT image, you need to download the latest version of u-boot image from the official website, and then burn it into SPI Flash. The download steps are as follows:

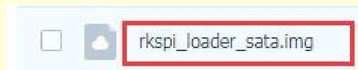
a. First enter the data download page of the development board, then select the official tool on the data download page, and then enter the folder below



b. Then choose to enter the following directory



c. Then download rkspi_loader_sata.img



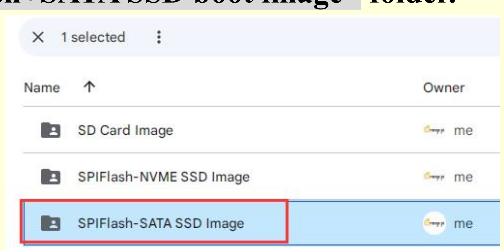
d. Then upload rkspi_loader_sata.img to the Ubuntu, Debian or OPi OS Arch system of the development board. For the upload method, please refer to the instructions in the method of uploading files to the Linux system of the development board.

Finally, execute the following command to burn the u-boot image into SPI Flash (note that this command is executed in Ubuntu, Debian, or OPi OS Arch):

orangepi@orangepi:~$ **sudo dd if=rkspi_loader_sata.img of=/dev/mtdblock0**

6) Then upload the linux image file (Debian or Ubuntu image downloaded from the official website) to the TF card. For the method of uploading the linux image file to the development board, please refer to the description in the section of **the method of uploading files to the development board Linux system**

Note that if you download an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the image file in the "SPIFlash+SATA SSD boot image" folder.



7) After uploading the image to the linux system of the development board, we enter the storage path of the image file in the command line of the linux system of the development board. For example, I store the linux image of the development board in the **/home/orangepi/Desktop** directory Download it, and then enter the **/home/orangepi/Desktop** directory to see the uploaded image file.

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

How to enter the command line of the development board linux system?

1. For the method of using the serial port to log in to the terminal, please refer to t**he instructions in the section on how to use the debugging serial port**.

2. Use ssh to remotely log in to the Linux system, please refer to the instructions in t**he section of SSH remote login to the development board**.

3. If HDMI, LCD and other display screens are connected, you can open a command line terminal on the desktop.

8) Then please refer to the instructions in the section of **the method of using SATA SSD to open the sata ssd** configuration to ensure that the system can recognize the ssd normally.

OPi OS Arch For the method of opening the sata ssd configuration in the OPi OS Arch system, please refer to the instructions in the section of the method of using SATA SSD in the OPi OS Arch system.

9) Then we can use the dd command to empty the SSD (Optional)

```
sudo dd bs=1M if=/dev/zero of=/dev/sda count=2000 status=progress

sudo sync
```

10) Then you can use the dd command to burn the linux image of the development board into the SSD

a. In the following command, the if= parameter is followed by the full path where the linux image is stored + the name of the Linux image (such as **the name of /home/orangepi/Desktop/Linux image**). Because we have entered the path of the linux image above, we only need to fill in the name of the Linux image.

b. Please do not copy the linux image name in the following command, but replace it with the actual image name (because the version number of the image may be updated)

```
sudo dd bs=1M if=Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/sda status=progress

sudo sync
```

> **Note, if you upload a linux image compressed file ending in .7z or xz, please remember to decompress it before using the dd command to burn**
>
> **The detailed description of all parameters of the dd command and more usage can be viewed by executing the man dd command in the linux system。**

11) After successfully burning the linux image of the development board to the SATA SSD, it cannot be used directly at this time. Because the default setting of the linux image is to only recognize NVMe SSDs, but not SATA SSDs, the following settings need to be done：

a. First mount the boot partition of the SATA SSD to the **/mnt** directory of the TF card card Linux system.

```
orangepi@orangepi:~/Desktop$ sudo mount /dev/sda1 /mnt/
```

b. Then open the SATA SSD configuration in the **orangepiEnv.txt** file in the boot partition of the SATA SSD (**please note that it is not /boot/orangepiEnv.txt in the TF card)**

```
orangepi@orangepi:~/Desktop$ sudo vim /mnt/orangepiEnv.txt
overlays=ssd-sata
```

> **Note that this step is somewhat different for the OPi OS Arch system, please add the following configuration in /boot/extlinux/extlinux.conf：**
>
> **[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf**
> **FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-ssd-sata.dtbo**

c. Then uninstall the boot partition of the SATA SSD

```
orangepi@orangepi:~/Desktop$ sudo umount /mnt/
```

12) At this point, you can use the **poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in

SPIFlash+SATA SSD will be started

13) After starting the system in the SATA SSD, use the **df -h** command to see the actual hard disk capacity

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G  8.0K  3.8G   1% /dev
tmpfs           769M  1.4M  768M   1% /run
/dev/sda2       233G  4.3G  226G   2% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/zram2      3.7G   76K  3.5G   1% /tmp
/dev/sda1       256M   90M  166M  36% /boot
/dev/zram1      194M   10M  170M   6% /var/log
tmpfs           769M   60K  769M   1% /run/user/1000
```

14) When the same system is burned in the TF card and SSD, **if both the TF card and SSD are inserted into the development board, and the development board is powered on at this time, u-boot will give priority to starting the system in the TF card**. However, since the systems in the TF card and the SSD are exactly the same, the UUIDs of the **/boot** partition and the **rootfs** partition in the two storage devices are also the same, which may cause the partition in the SSD to be loaded when the TF card starts. Running the script below resolves this issue

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

Exactly the same system means that the image name is exactly the same. Even if they are all Debian11 systems, different versions are different.

There is no fix_mmc_ssd.sh script in OPi OS Arch system.

2. 6. 2. How to use balenaEtcher software to burn

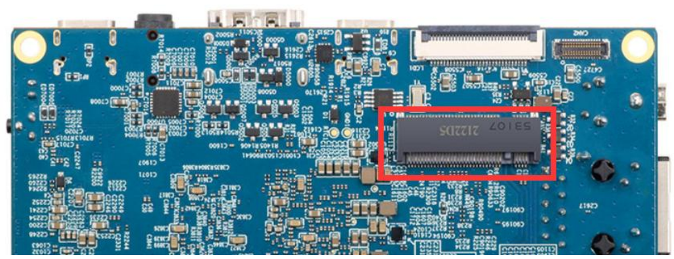
Please do not use this method for OPi OS Arch system.

1) First, you need to prepare a SATA SSD solid state drive

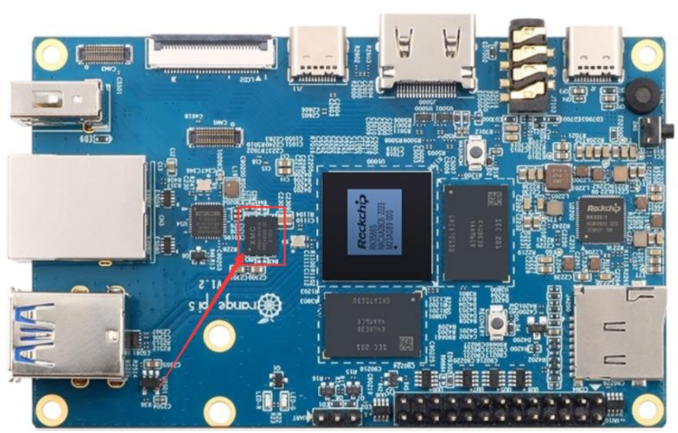
a. The M.2 2242 SSD is as follow

b. The 2280 specification SSD is as follows (the 2280 specification SATA SSD can also be used, but the SSD will exceed the development board after being inserted into the development board）

2) Then insert the SSD into the M.2 interface of the development board and fix it.



3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming

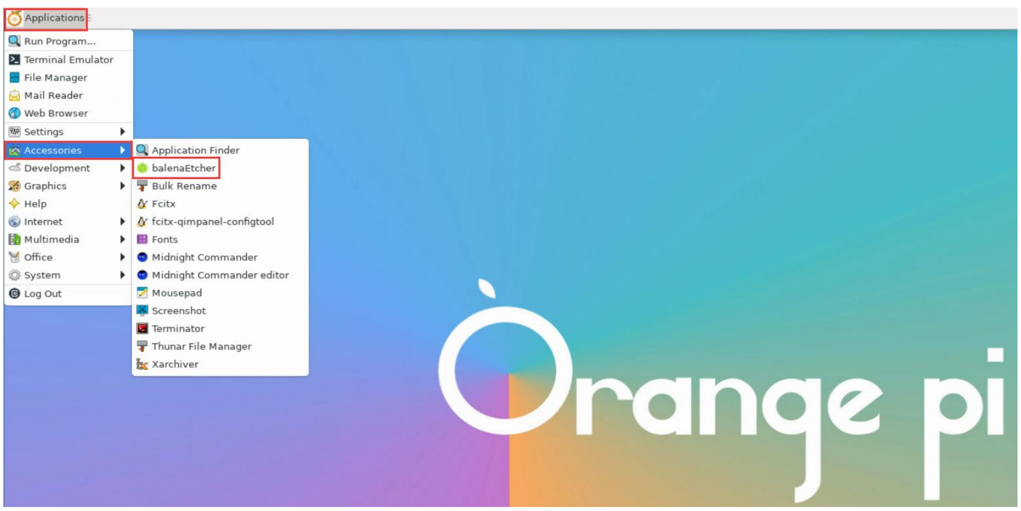


4) Burning the linux image to SPIFlash+SDD needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF

card to start the development board to enter the linux system. For the method of burning the linux image to the TF card, please refer to the instructions in the two sections of **the method of burning the Linux image to the TF card based on the Windows PC** and t**he method of burning the Linux image to the TF card based on the Ubuntu PC**.

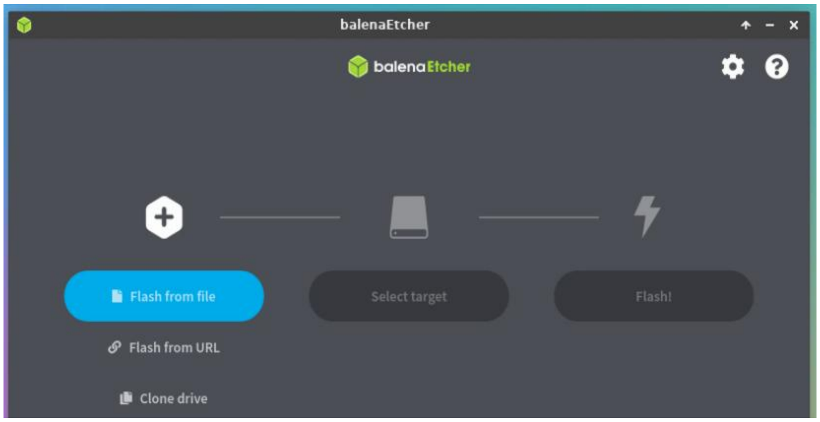
5) Then please refer to the instructions in the section of t**he method of using SATA SSD** to open the sata ssd configuration to ensure that the system can recognize the ssd normally.

6) The balenaEtcher has been pre-installed in the linux image, and the opening method is as follows:



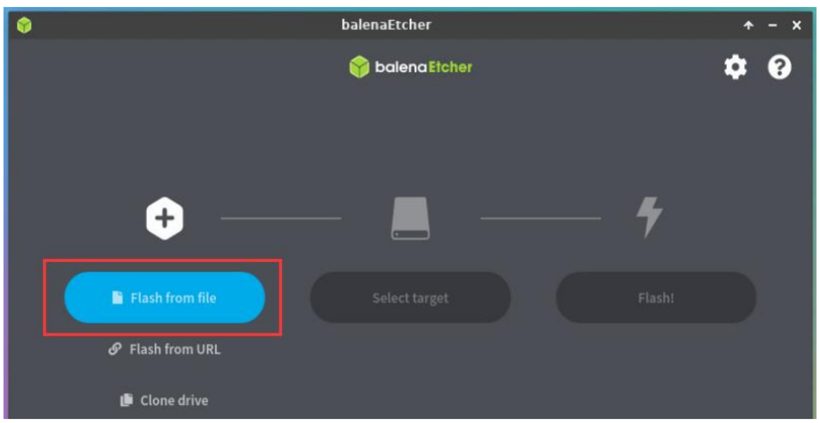
> If it is not pre-installed, for how to download and install the arm64 version of balenaEtcher, please refer to the instructions in the section on **how to download and install the arm64 version of balenaEtcher.**

7) The interface after balenaEtcher is opened is as follow：

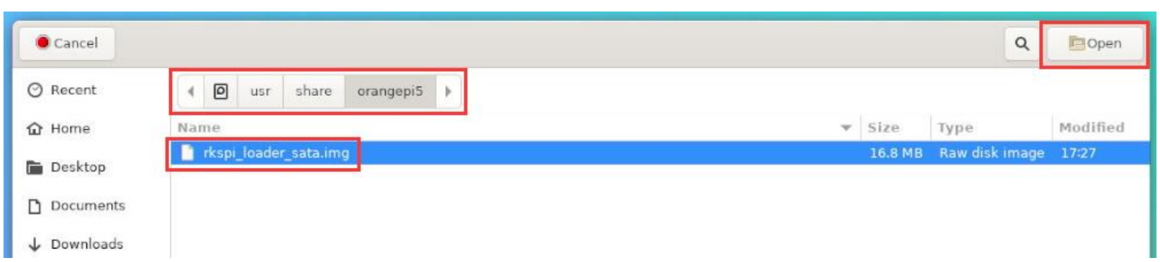


8) The method of using balenaEtcher to burn u-boot to the SPI Flash of the development board is as follows：

a. First click **Flash from file**



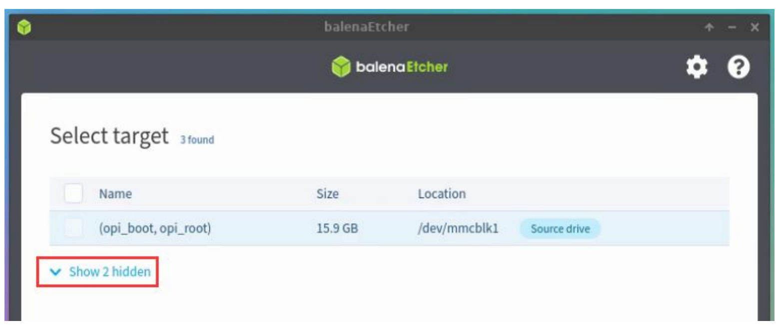
b. Then enter the **/usr/share/orangepi5/** directory, select rkspi_loader_sata.img, and click **Open** to open.



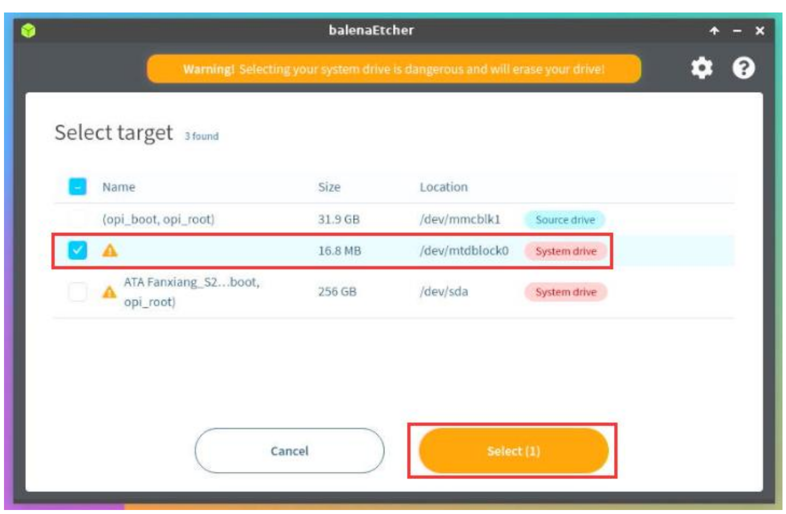
c. The interface after opening **rkspi_loader.img** is as follows：

d. Then click **Select target**

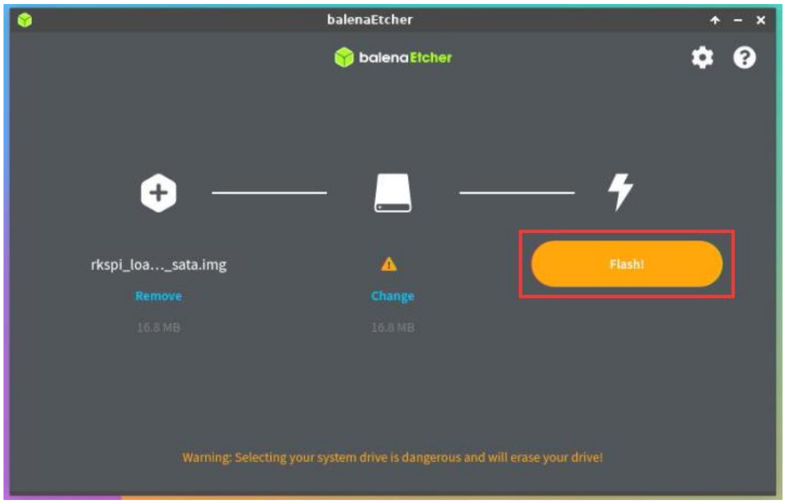
e. Then click **Show 2 hidden** to open more options for storage devices



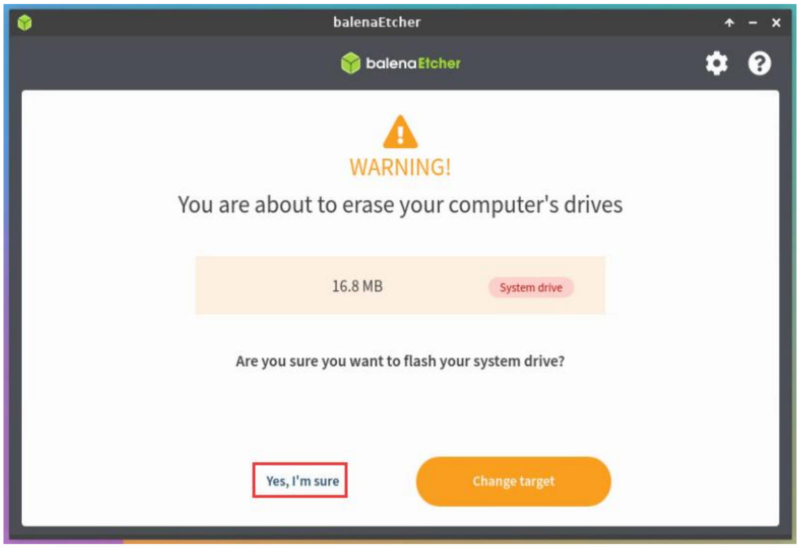
f. Then select the device name of SPI Flash **/dev/mtdblock0,** and click **Select**



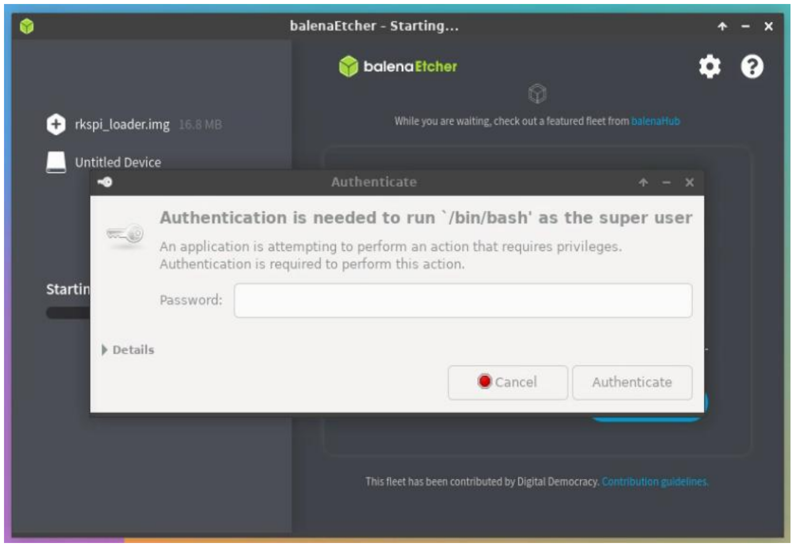
g. Then click **Flash**



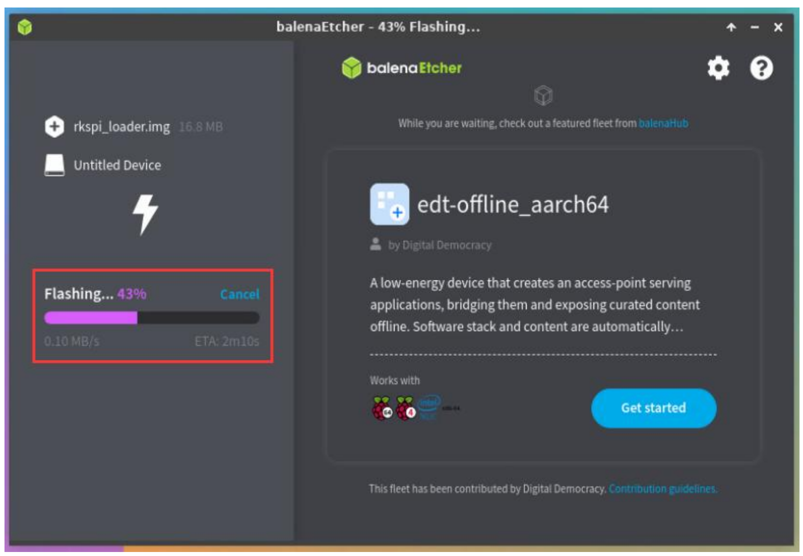
h. Then click **Yes, I'm sure**



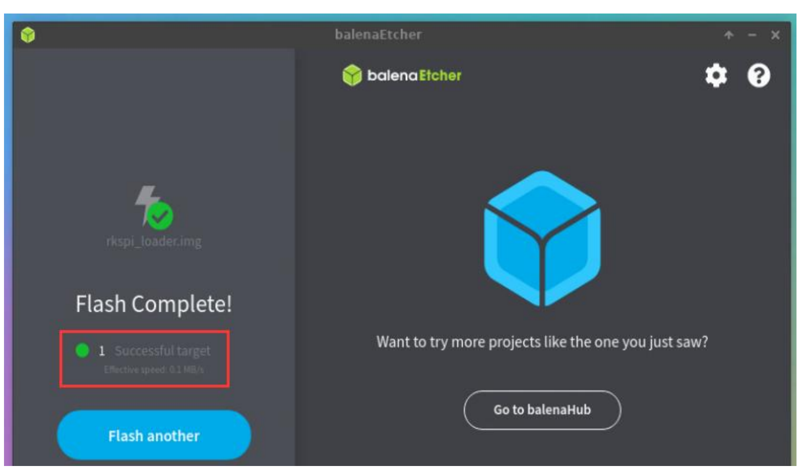
i. Then enter the password orangepi of the development board linux system, and it will start burning the u-boot image into the SPI Flash



j. The display of the burning process is as follow：

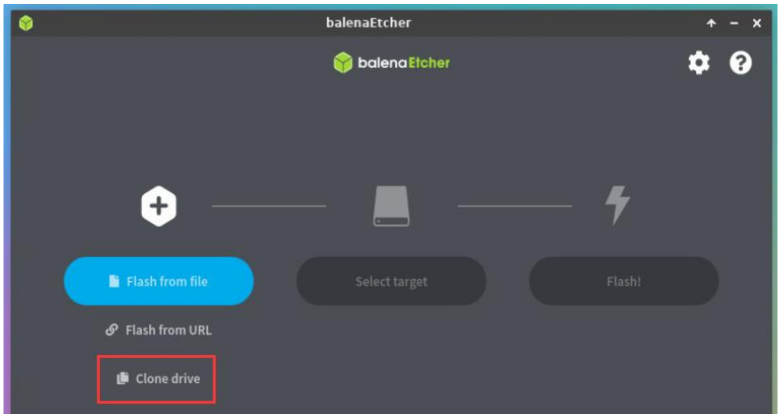


k. The display after burning is as follows:

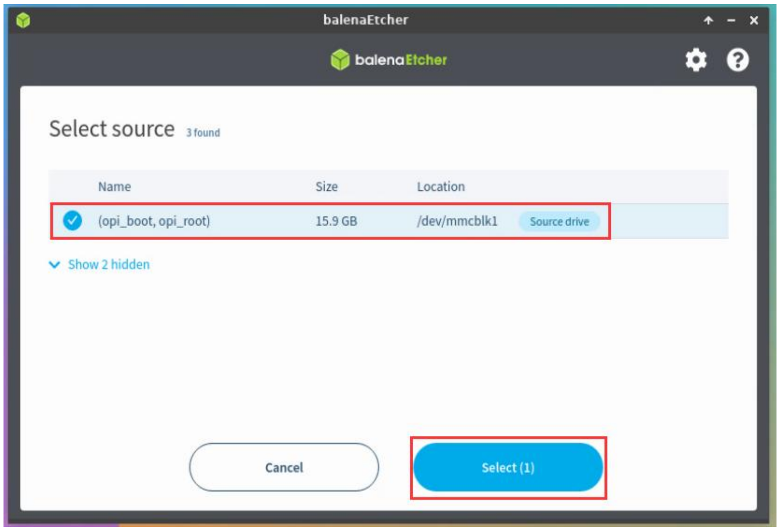


9) The method of burning the linux system in the TF card to the SSD (this method is equivalent to cloning the system in the TF card to the SSD）

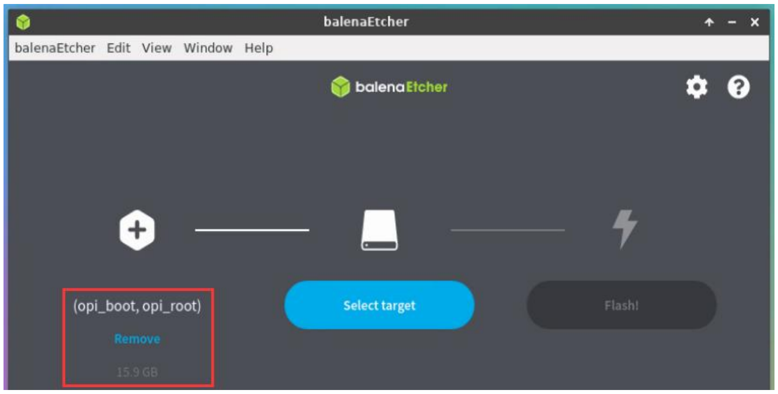
a. First click **Clone drive**



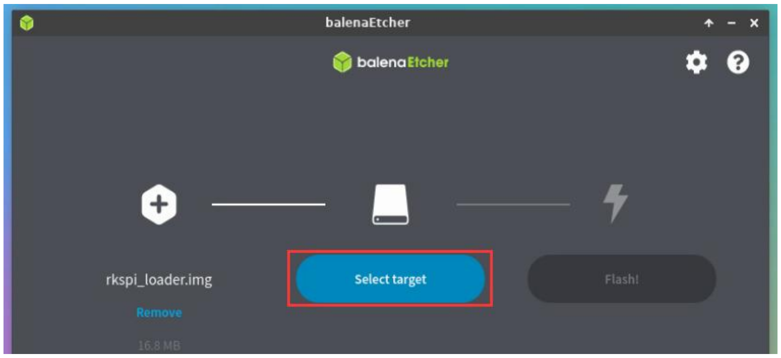
b. Then select the device name of TF card **/dev/mmcblk1**



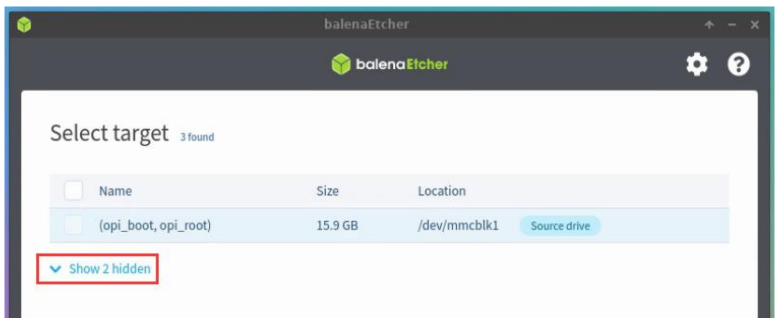
c. The interface after opening the TF card is as follow：



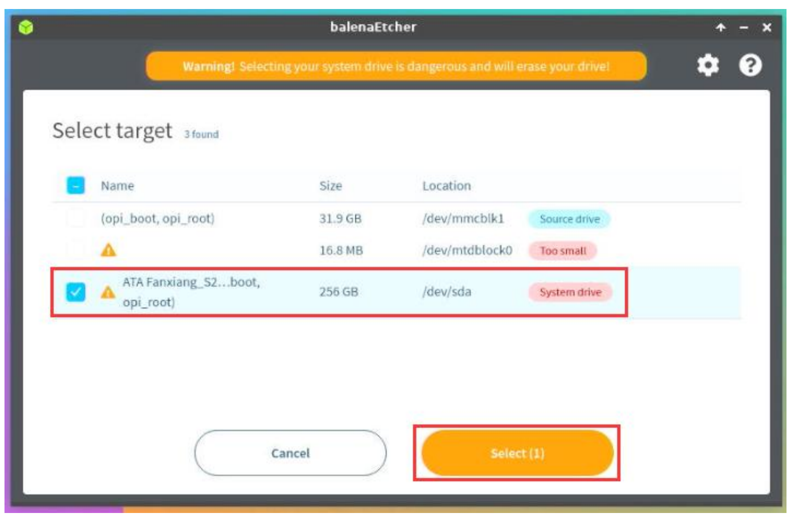
d. Then click **Select target**



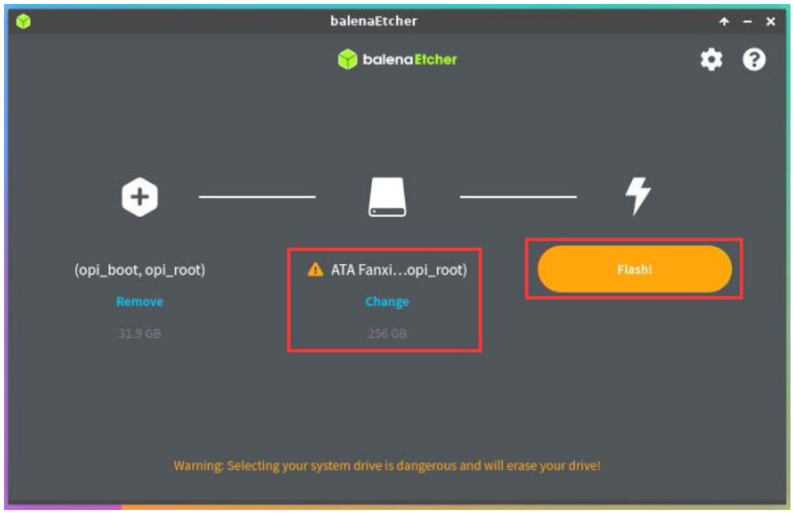
e. Then click **Show 2 hidden** to open more options for storage device



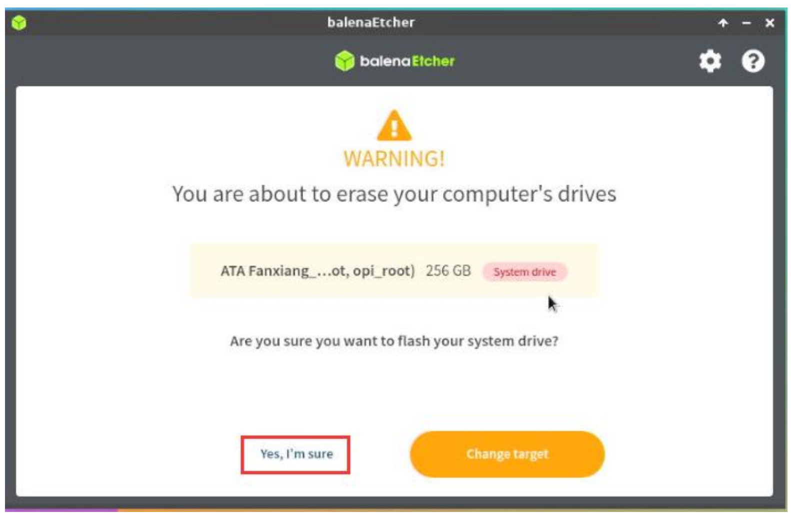
f. Then select the SSD device name **/dev/sda,** and click **Select**



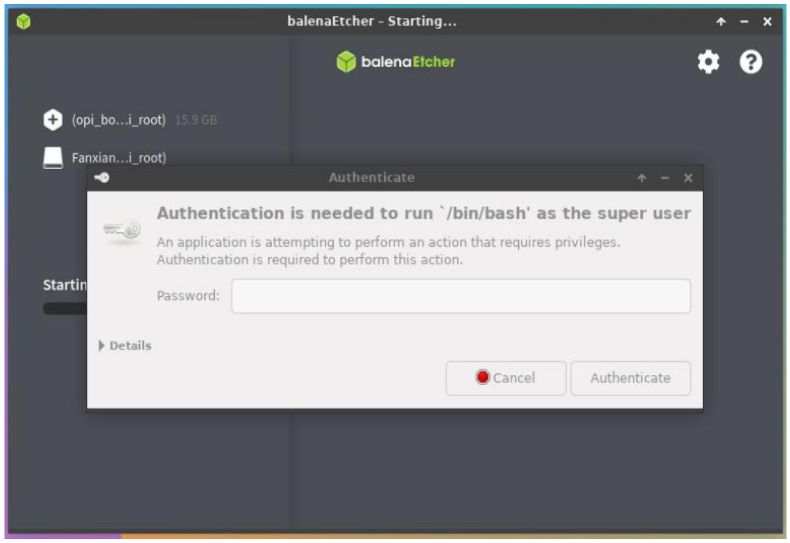
g. Then click **Flash**



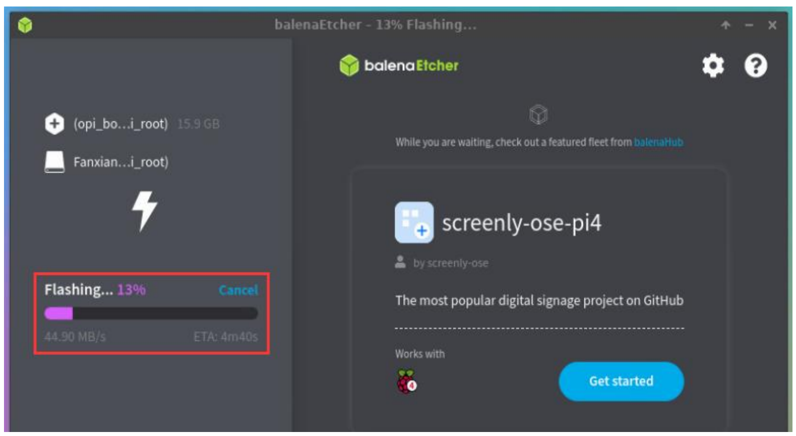
h. Then click **Yes, I'm sure**

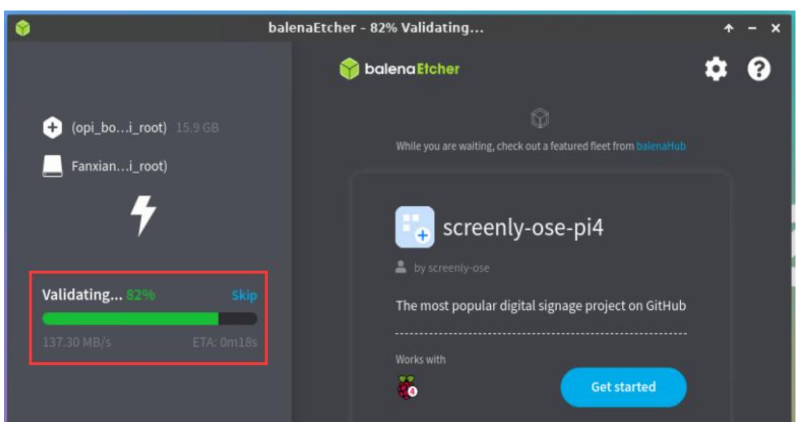


i. Then enter the password **orangepi** of the linux system on the development board, and it will start burning the linux image to the SSD.

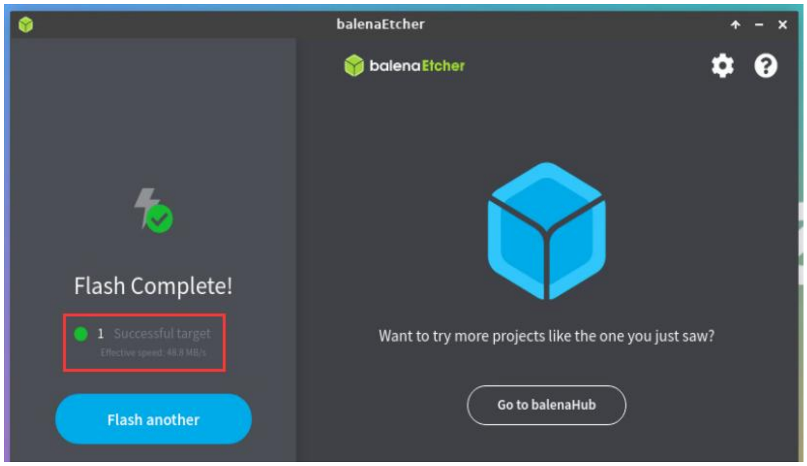


j. The display of the burning process is as follows:



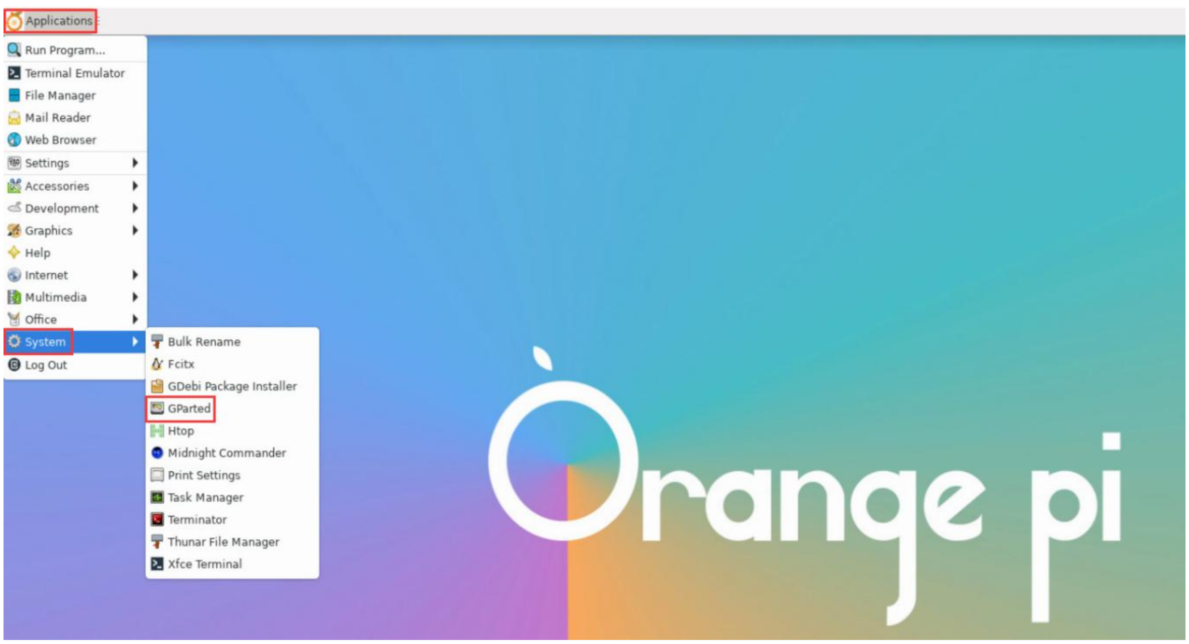


k. The display after burning is as follow:



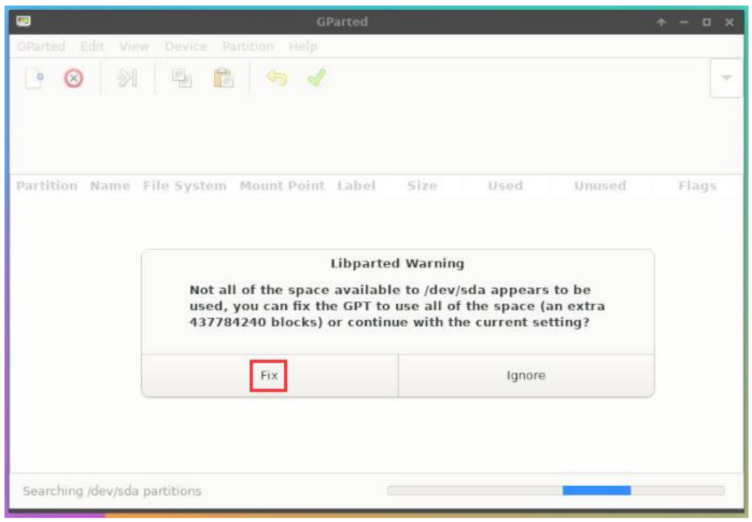
l. Then you need to expand the capacity of the rootfs partition in the SSD. The steps are as follow:

a) First open **GParted**

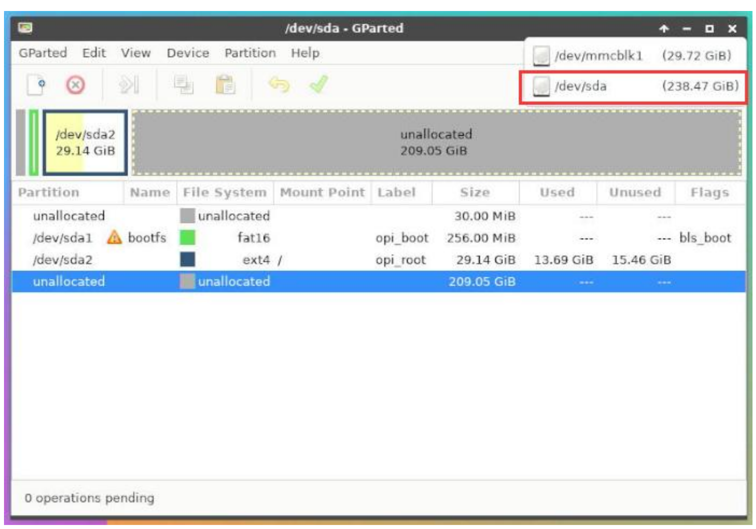


b) Then enter the password **orangepi** of the linux system, and click **Authenticate**

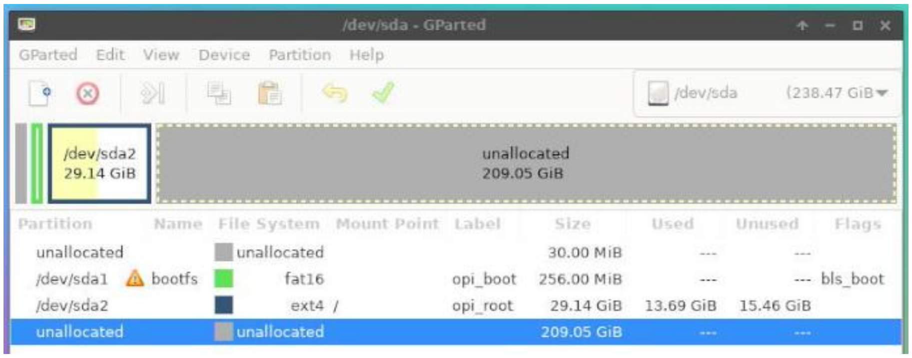
c) Then click **Fix**



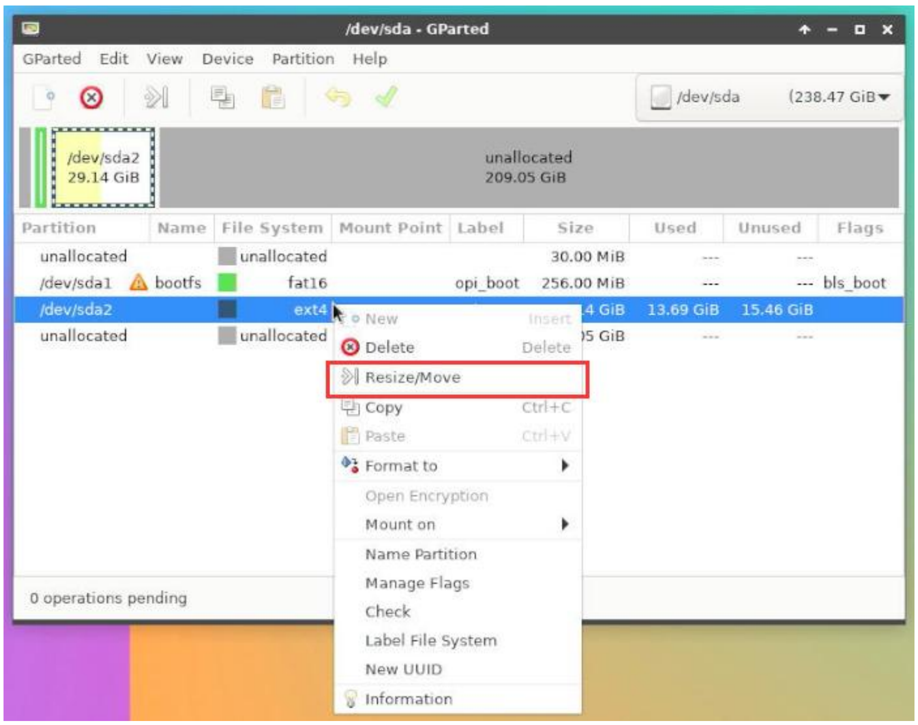
d) Then choose SSD



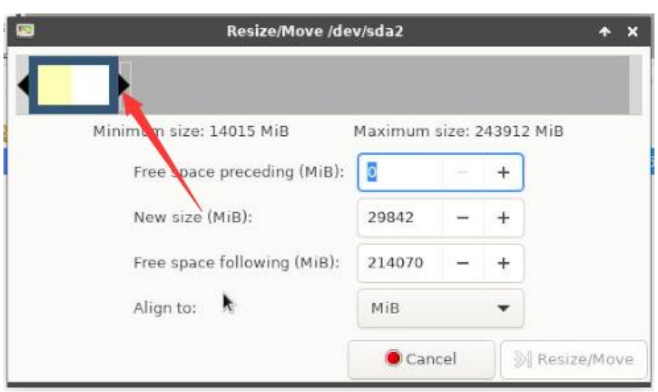
e) The display interface after selecting SSD is as follow：



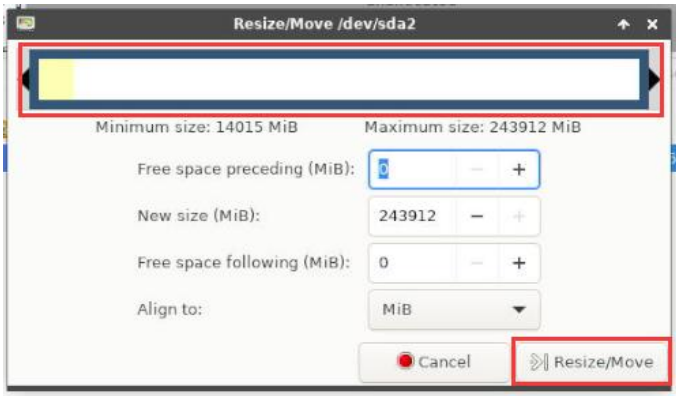
f) Then select the **/dev/sda2** partition, then right-click, and then select **Resize/Move**



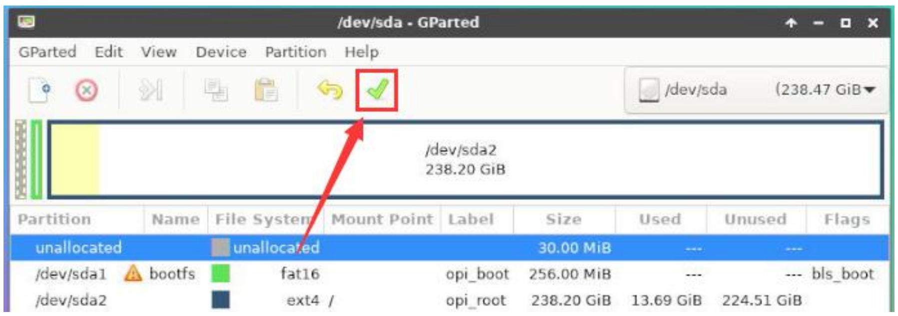
g) Then drag the capacity to the maximum at the position shown in the figure below



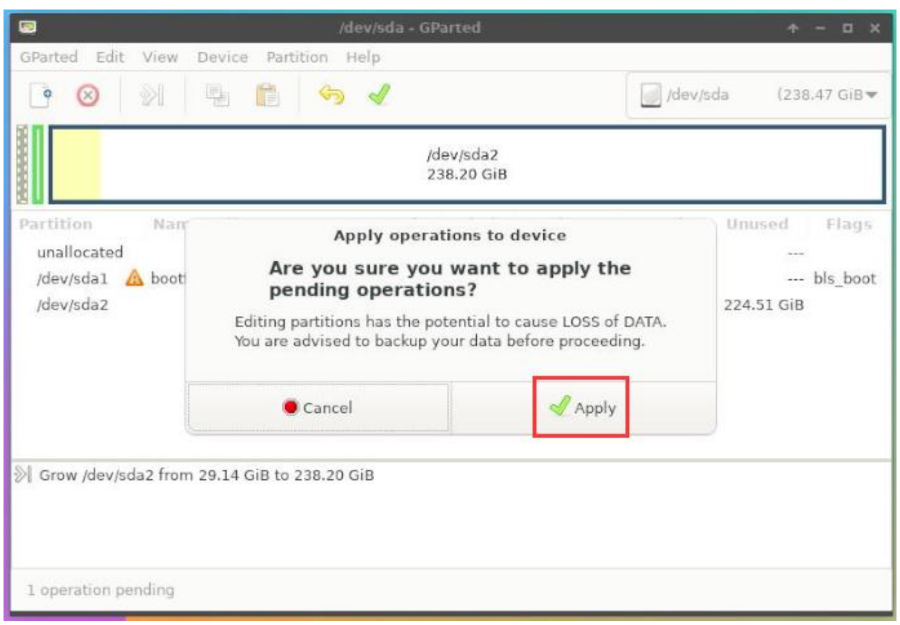
h) Then click **Resize/Move**



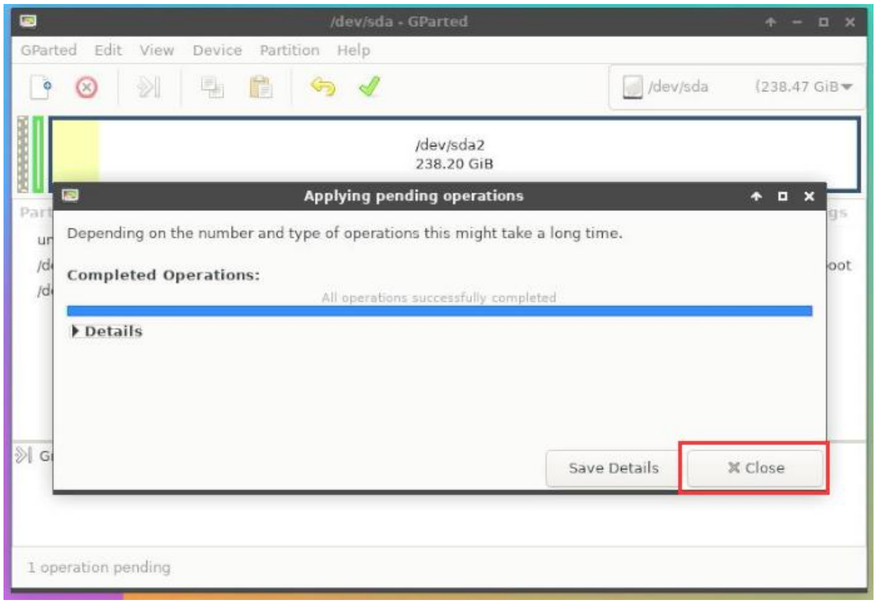
i) Then click the green√



j) Then click **Apply**



k) Then click **Close**

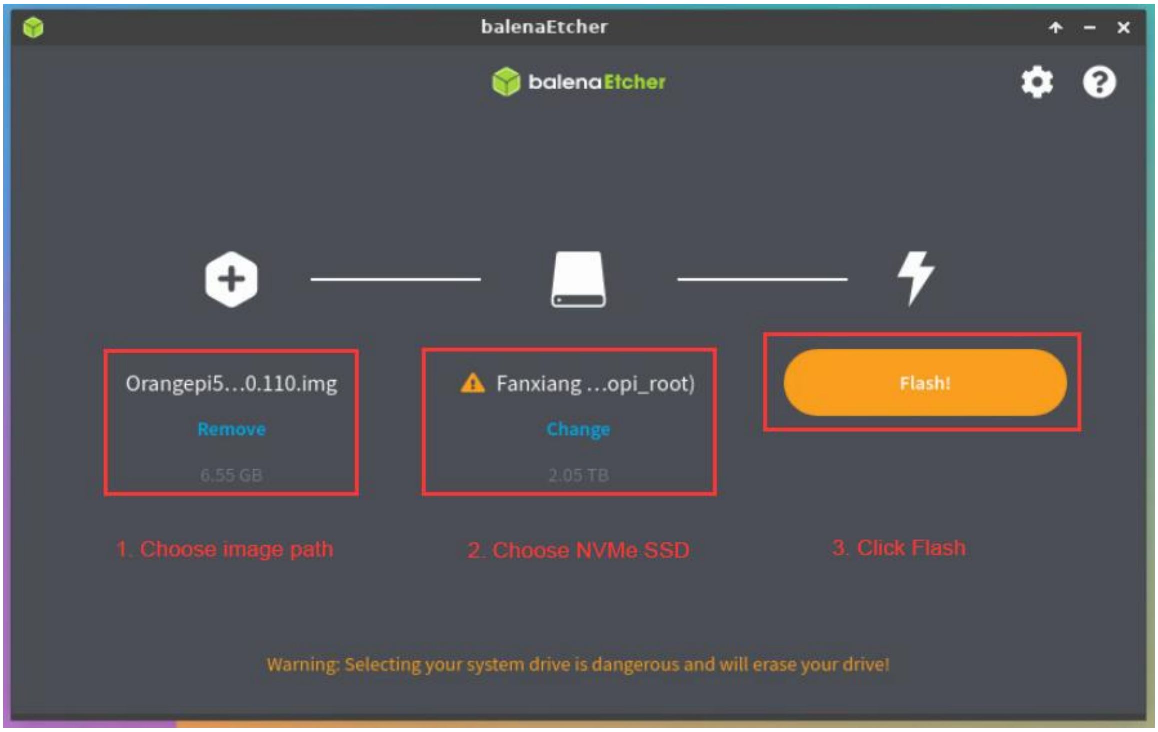


m. At this point, you can use the **sudo poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in SPIFlash+STAT SSD will be started.

10) Step **9)** is to clone the system in the TF card to the SSD. We can also directly burn the linux image file to the SSD. Here are the steps:

a. Upload the linux image file to the linux system of the development board

b. Then use balenaEtcher to burn



c. **After using this method to burn the image, there is no need to manually expand the capacity, and it will automatically expand the capacity at the first startup.**

d. After successfully burning the linux image of the development board to the SATA SSD, it cannot be used directly at this time. Because the default setting of the linux image is to only recognize NVMe SSDs, but not SATA SSDs, the following settings need to be done:

   a) First mount the boot partition of the SATA SSD to the **/mnt** directory of the TF card Linux system

```
orangepi@orangepi:~/Desktop$ sudo mount /dev/sda1 /mnt/
```

   b) Then open the SATA SSD configuration in the **orangepiEnv.txt** file in the boot partition of the SATA SSD (note that it is not **/boot/orangepiEnv.txt** in the TF card)

```
orangepi@orangepi:~/Desktop$ sudo vim /mnt/orangepiEnv.txt
overlays=ssd-sata
```

   c) Then unmount the boot partition of the SATA SSD

```
orangepi@orangepi:~/Desktop$ sudo umount /mnt/
```

e. At this point, you can use the **sudo poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in SPIFlash+STAT SSD will be started.

2. 7. How to write Linux image to SPIFlash+USB storage devices

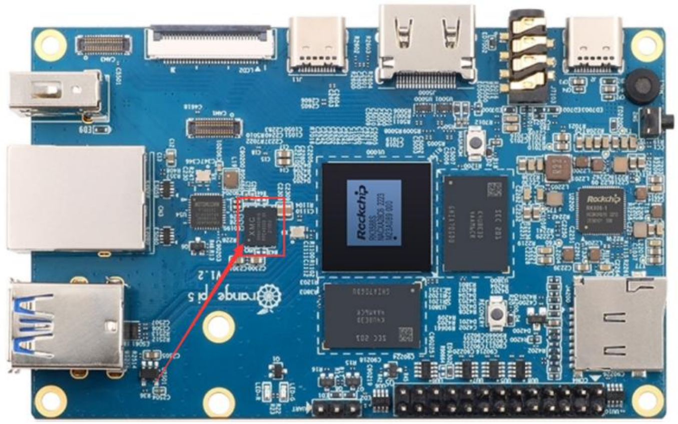
Note that the Linux image mentioned here specifically refers to the image of Linux distributions such as Debian or Ubuntu downloaded from the Orange Pi data download pag**e.**

1) First, you need to prepare a USB storage device, such as a U Disk

2) Then please refer to the instructions in the two sections of the method of burning **the Linux image to the TF card based on the Windows PC** and t**he method of burning the Linux image to the TF card based on the Ubuntu PC** to burn the Linux image to the USB storage device. There is no difference between burning the Linux image to the USB storage device and burning the Linux image to the TF card (when the TF card is inserted into the card reader, the card reader at this time is actually equivalent to a U disk）

3) Then insert the USB storage device with the programmed Linux system into the USB3.0 interface of the development board, **Note, please do not insert the USB storagedevice with the programmed system into other USB interfaces of the development board. Only the USB3.0 interface shown supports booting the Linux system**.

4) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming



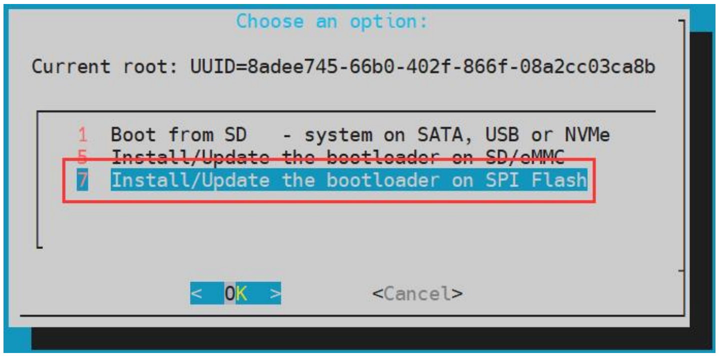
5) Burning the u-boot image to SPIFlash needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please refer to the instructions in the two sections of t**the method of burning the Linux image to the TF card based on the Windows PC** and **the method of burning the Linux image to the TF card based on the Ubuntu PC**

6) After using the TF card to start the Linux system, you can burn the u-boot image into the SPI Flash

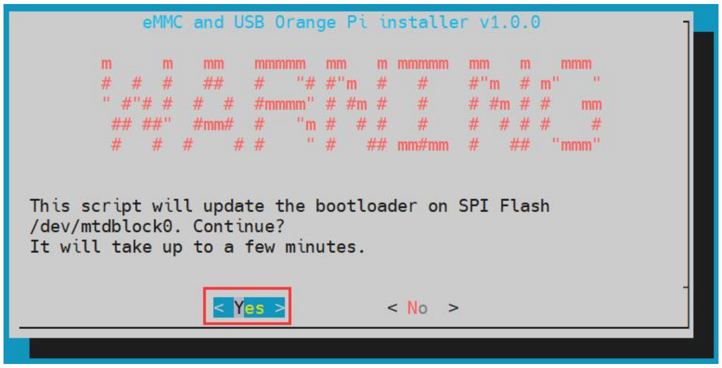
a. Run **nand-sata-install** first, o**rdinary users remember to add sudo permission**

```
orangepi@orangepi:~$ sudo nand-sata-install
```

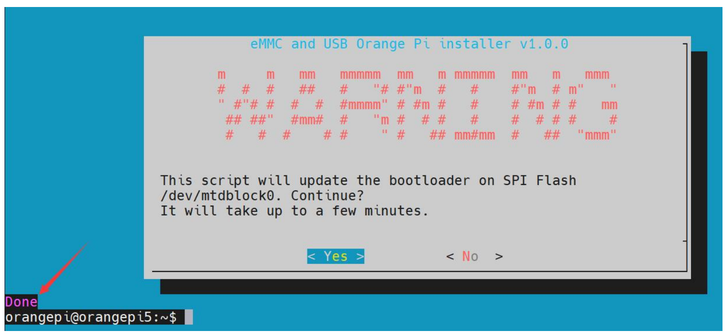
b. Then select **7 Install/Update ther bootloader on SPI Flash**



c. Then select **<Yes>**



d. Then please wait patiently for the burning to complete. After the burning is completed, the display will be as follows (a **Done** will be displayed in the lower left corner):

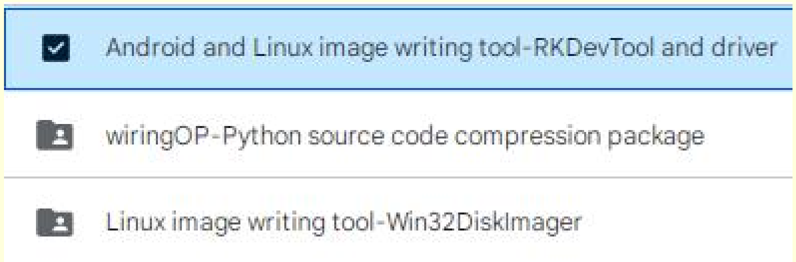


There is no nand-sata-install script in OPi OS Arch system, please use the following command to image u-boot to SPI Flash:

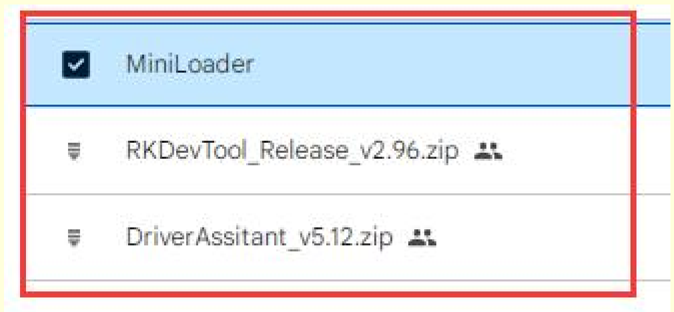
[orangepi@orangepi ~]$ **sudo dd if=/boot/rkspi_loader.img of=/dev/mtdblock0**

If you need to start the OpenWRT image, you need to download the latest version of u-boot image from the official website, and then burn it into SPI Flash. The download steps are as follows:

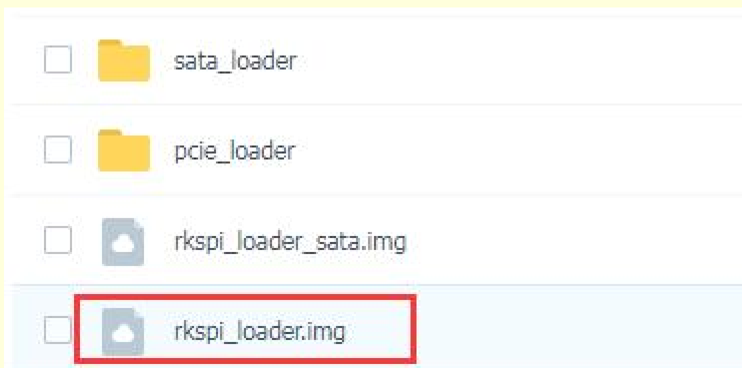
a. First enter the data download page of the development board, then select the official tool on the data download page, and then enter the folder below



b. Then choose to enter the following directory



c. Then download rkspi_loader.img



d. Then upload rkspi_loader.img to the Ubuntu or Debian or OPi OS Arch system of the development board. For the upload method, please refer to the instructions in the method of uploading files to the Linux system of the development board.

Finally, execute the following command to burn the u-boot image into SPI Flash (note that this command is executed in Ubuntu, Debian, or OPi OS Arch):

orangepi@orangepi:~$ **sudo dd if=rkspi_loader.img of=/dev/mtdblock0**

7) At this point, you can use the **poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in the SPIFlash+USB storage device will be started

8) After starting the system in the USB storage device, use the **df -h** command to see the actual capacity of the USB storage device

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G  8.0K  3.8G   1% /dev
tmpfs           769M  588K  769M   1% /run
/dev/sda2        15G  1.6G   13G  11% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/zram2      3.7G   60K  3.5G   1% /tmp
/dev/sda1       256M  111M  146M  44% /boot
/dev/zram1      194M  9.0M  171M   5% /var/log
tmpfs           769M     0  769M   0% /run/user/1000
```

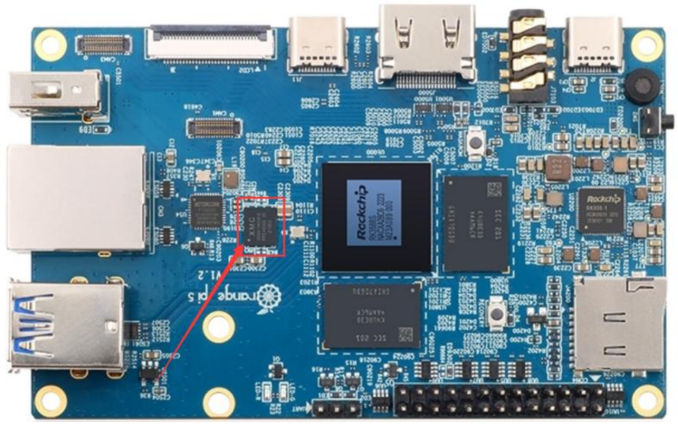
2. 8. Method of burning OpenWRT image into SPI FLASH

The method introduced in this section is to burn the entire OpenWRT image into SPI Flash. No SSD or USB disk is required. In other words, u-boot, kernel and rootfs are all stored in SPI Flash.

Since the SPI Flash on the development board is only 16MB, this system cannot install much software and can only implement some basic functions at present.

2. 8. 1. How to burn using RKDevTool

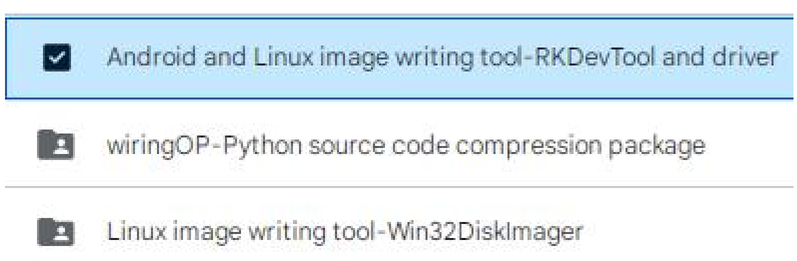
1) The location of SPI Flash on the development board is as shown in the picture below. No other settings are required before starting burning.



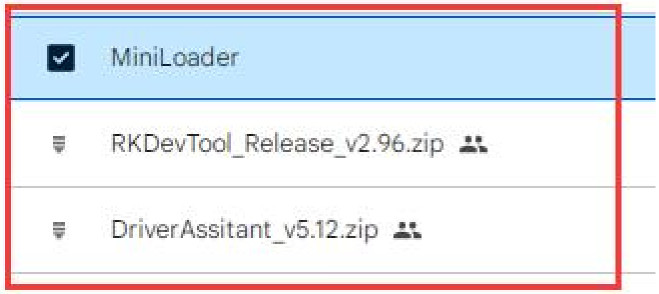
2) Then you need to prepare a good quality Type-C interface data cable

3) Then download the Rockchip micro-driver **DriverAssitant_v5.12.zip** and **MiniLoader** and the burning tool **RKDevTool_Release_v2.96.zip** from the Orange Pi data download page. Please ensure that the version of the downloaded **RKDevTool** tool is **v2.96**

a. On the Orange Pi data download page, first select the **official tool**, and then enter the folder below



b. Then download all the files below



Note that the "MiniLoader-things needed to burn Linux images" folder will be referred to as the MiniLoader folder below.

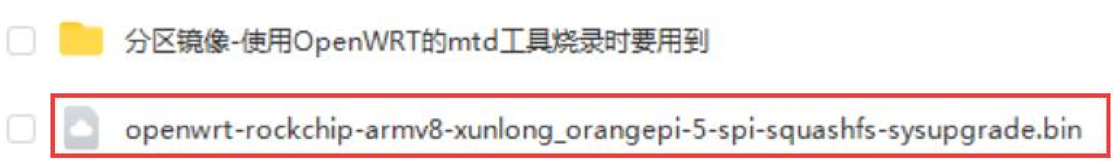
4) Then download the OpenWRT image that can be booted from SPIFlash from the Orange Pi download page. Due to the capacity of SPIFlash, the image is less than 16MB.

a. After opening the download link, you can see the following three types of OpenWRT images. Please select the image in the SPIFlash startup image folder.

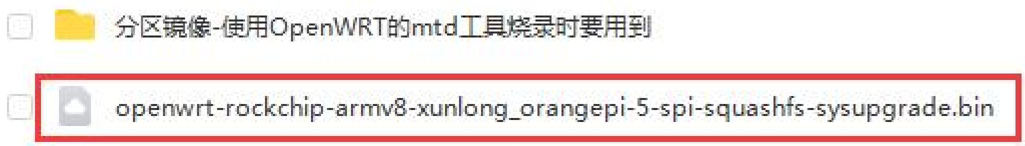
b. Then you can see a "Partition Image" folder and a separate OpenWRT image file. The difference between them is:

a) The three files u-boot, dtb, and firmware (including kernel and rootfs) files contained in the "Partition Image" folder, together form an OpenWRT image.

b) The image file shown in the figure below is a complete image file generated by packaging u-boot, dtb, and firmware in the "Partition Image" folder



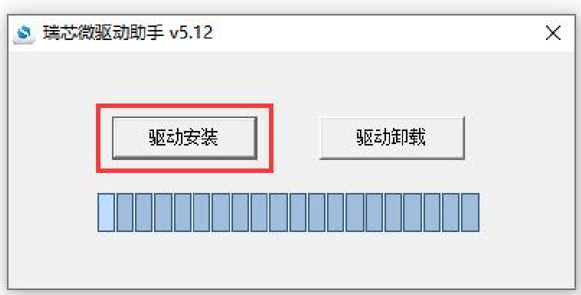
c. Here we need to select the packaged complete image



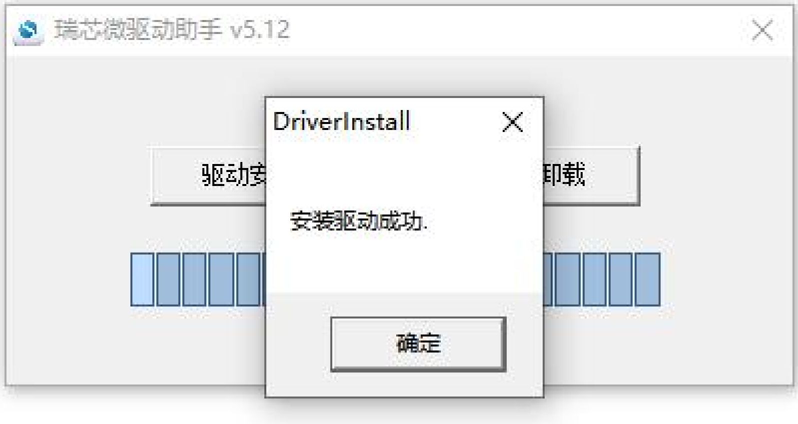
5) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

6) Open **DriverInstall.exe** and install the Rockchip microdriver as follows:

a. Click the "**Driver Installation**" button

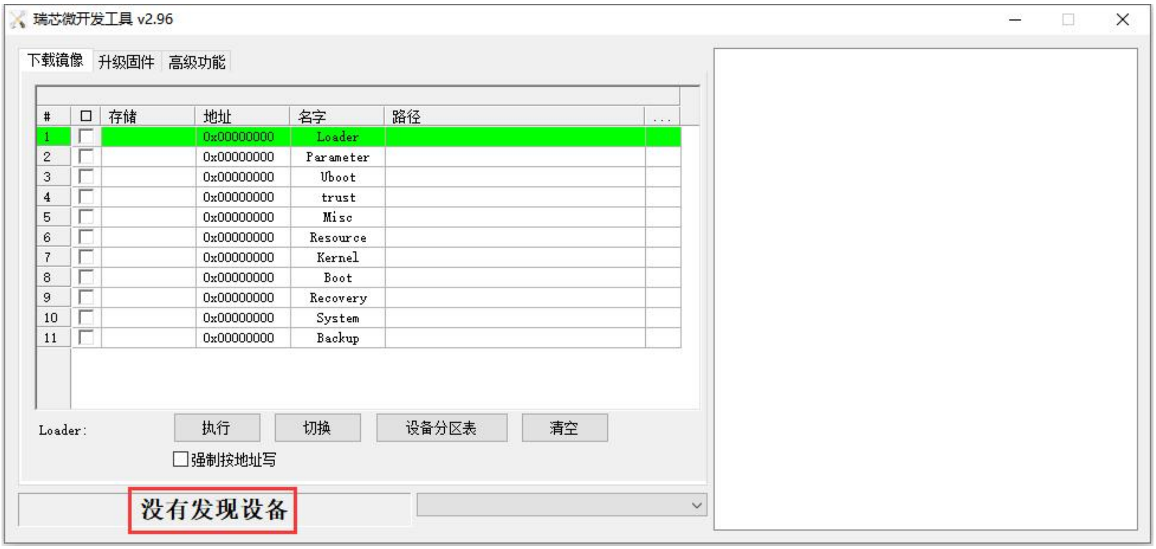


b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



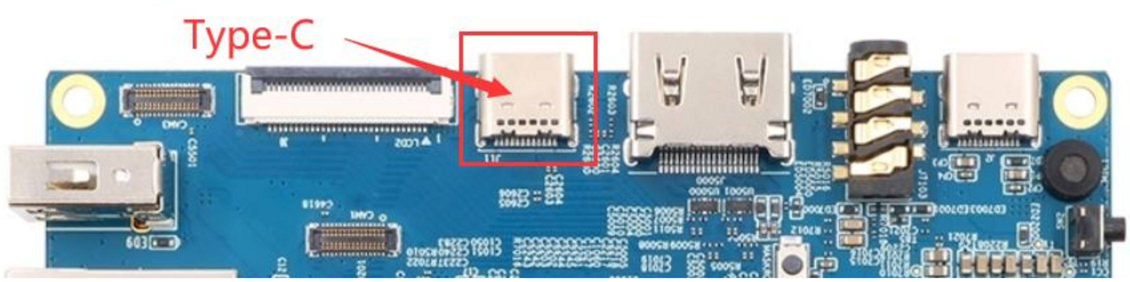
7) Then unzip **RKDevTool_Release_v2.96.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

8) After opening the **RKDevTool** burning tool, because the computer has not been connected to the development board through the Type-C cable at this time, "**No device found**" will be prompted in the lower left corner.



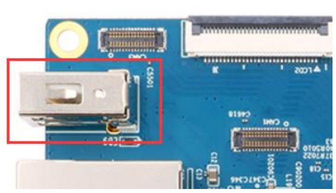
9) Then start burning the OpenWRT image into SPI FLASH

a. First, connect the development board and Windows computer through the Type-C data cable. The location of the Type-C interface of the development board is as shown in the figure below.



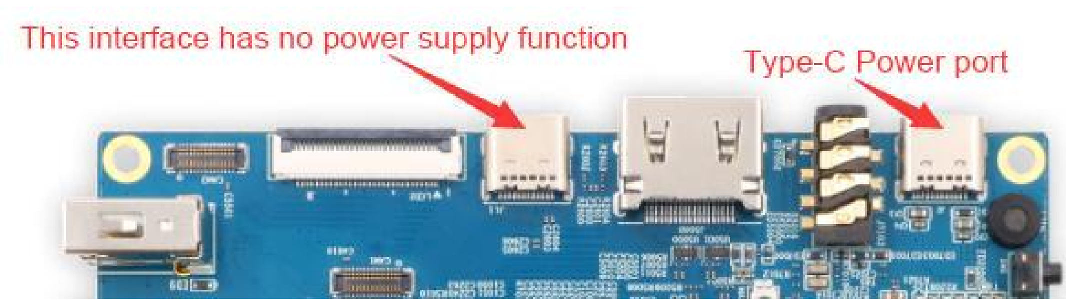
b. Make sure the development board is not connected to the power supply and the TF card is not inserted.

c. Also make sure that no USB device is plugged into the white USB2.0 interface in the picture below

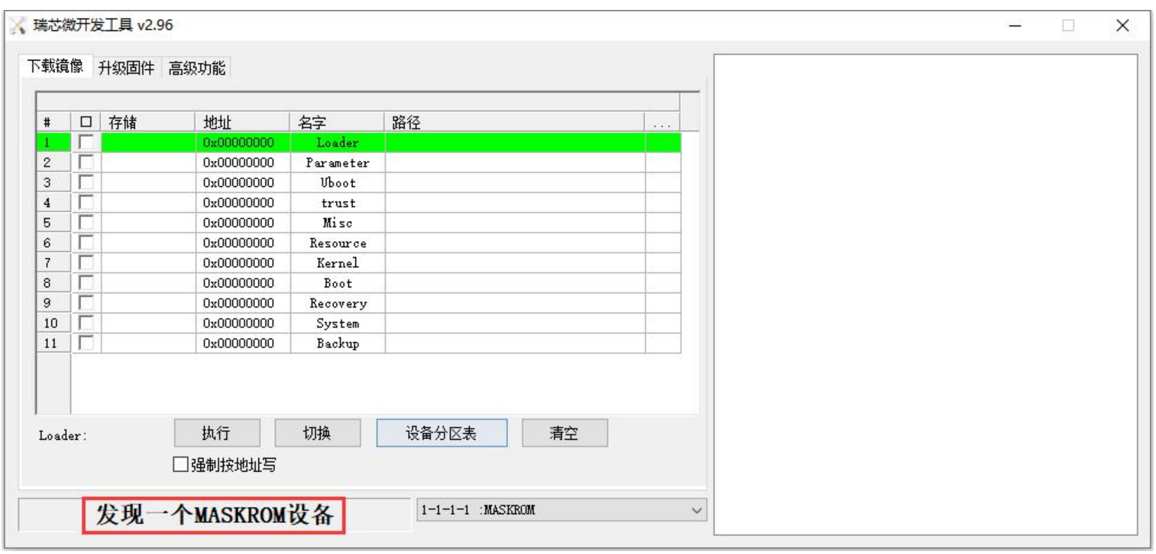


d. Then press and hold the MaskROM button of the development board. The position of the MaskROM button on the development board is as shown in the figure below:

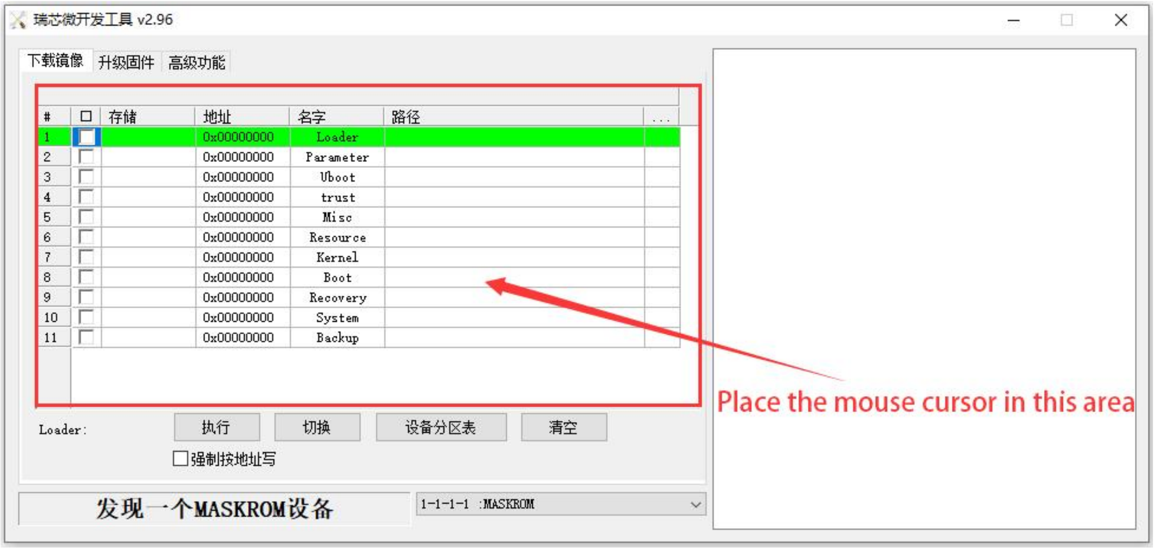
e. Then connect the power supply of the Type-C interface to the development board, power it on, and then release the MaskROM button.



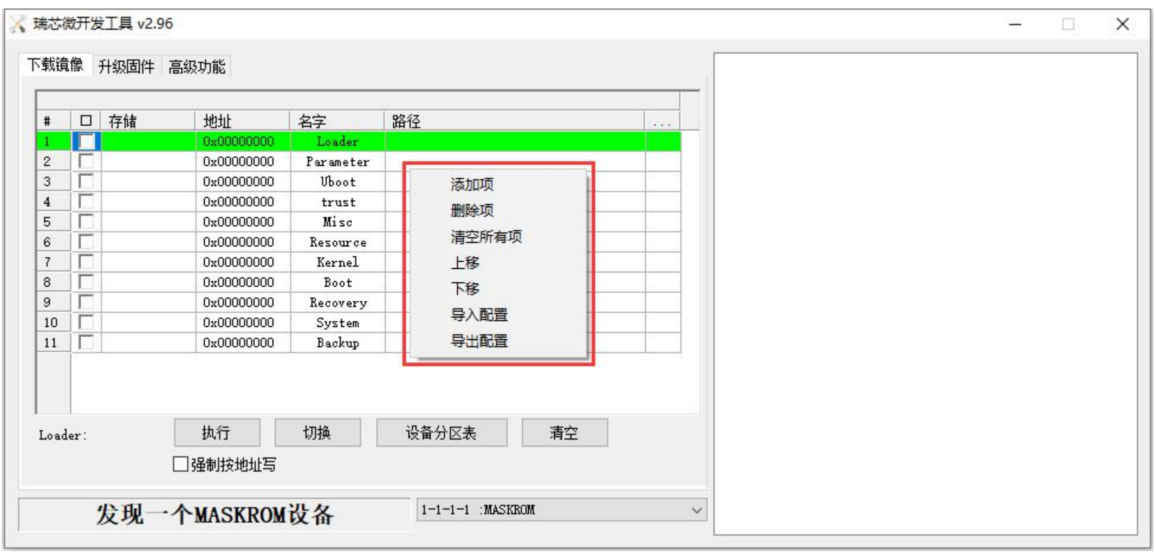
f. If the previous steps go well, the development board will enter **MASKROM** mode at this time, and the interface of the burning tool will prompt "**A MASKROM device was found**"



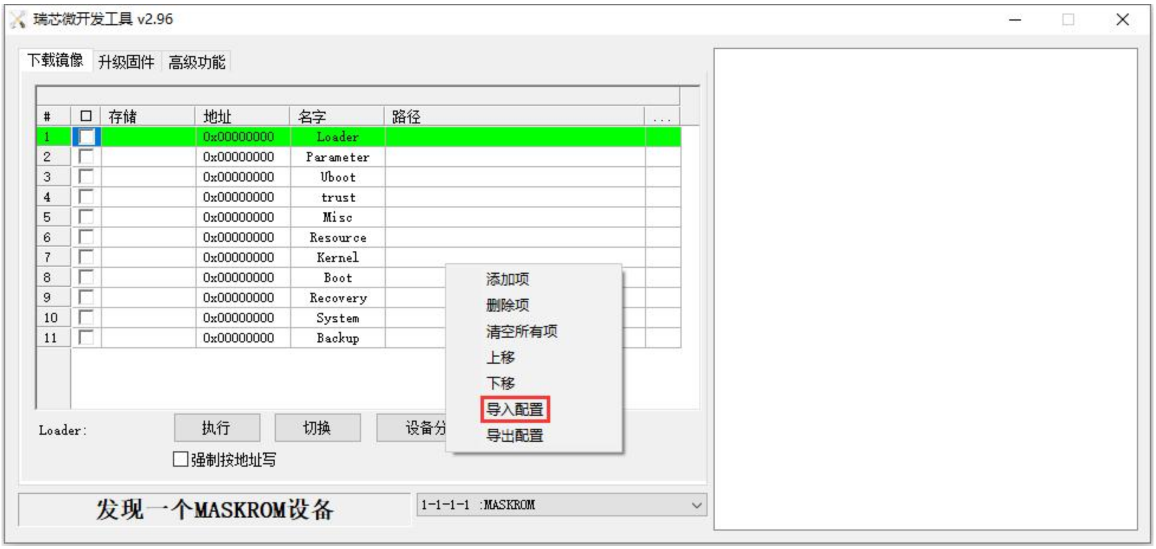
g. Then place the mouse cursor in the area below



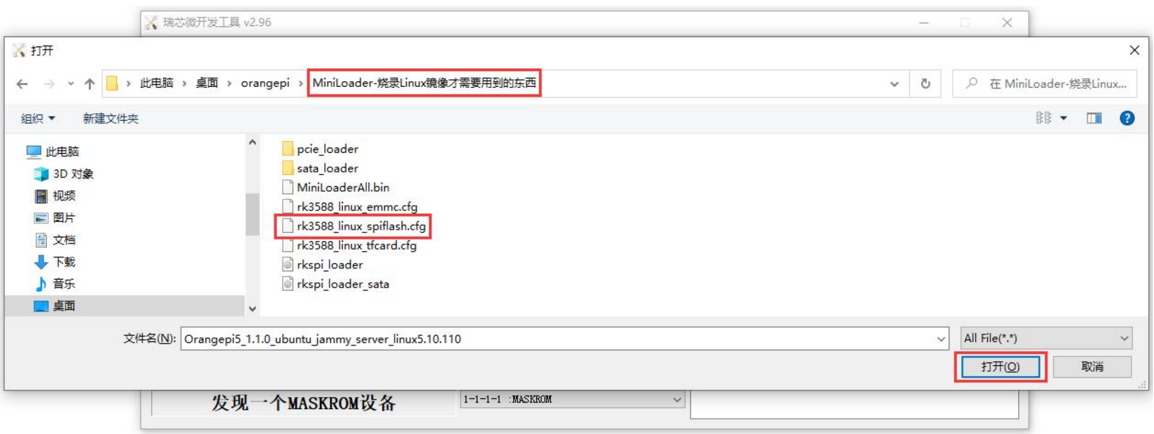
h. Then click the right button of the mouse and the selection interface shown below will pop up.



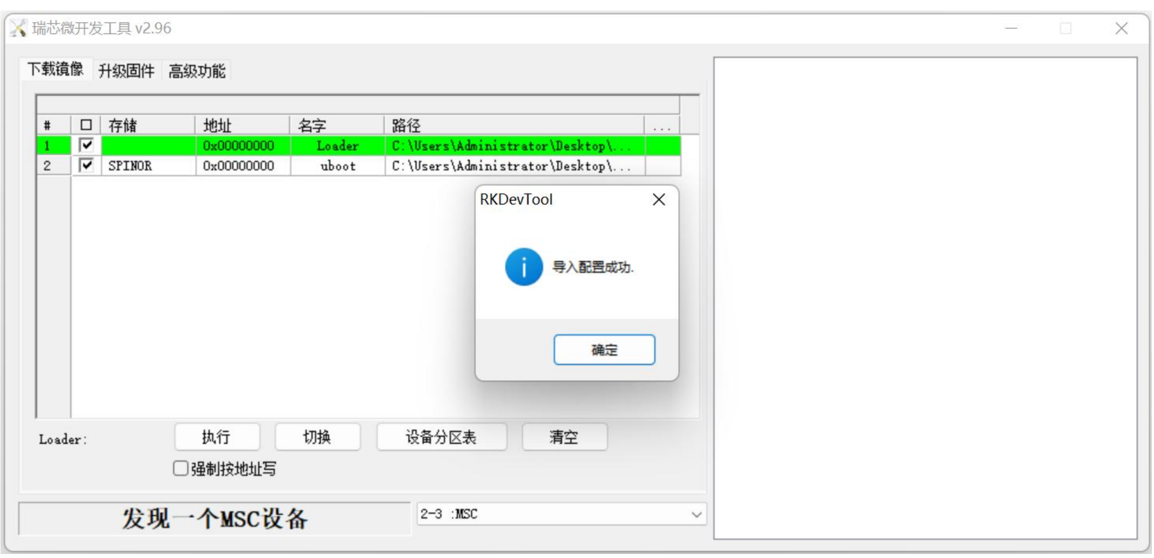
i. Then select the **Import Configuration** option



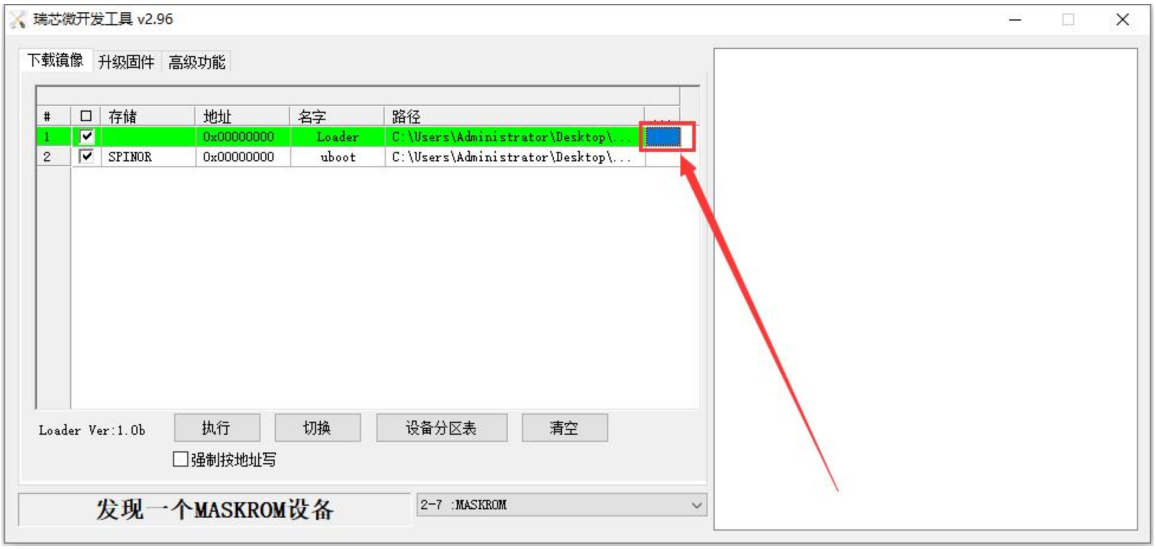
j. Then select the **rk3588_linux_spiflash.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and then click to **open**



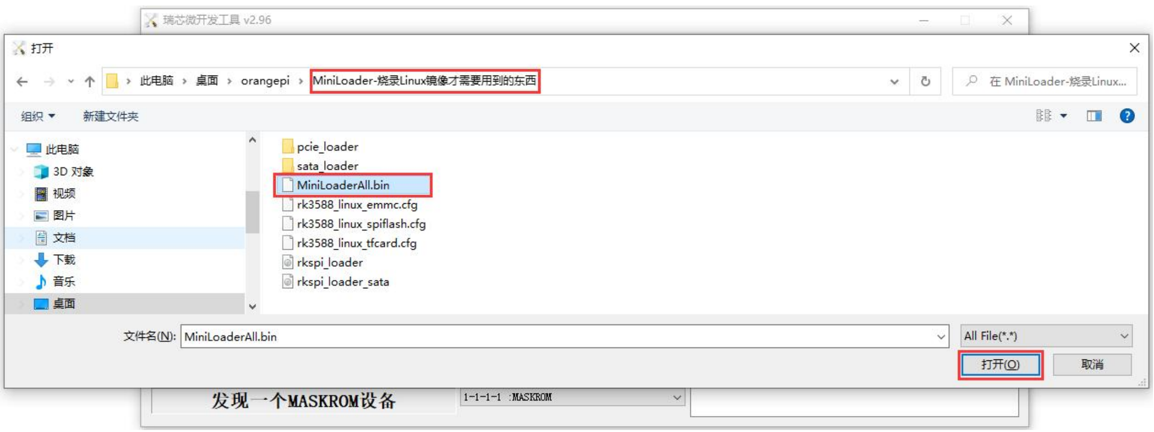
k. Then click **OK**



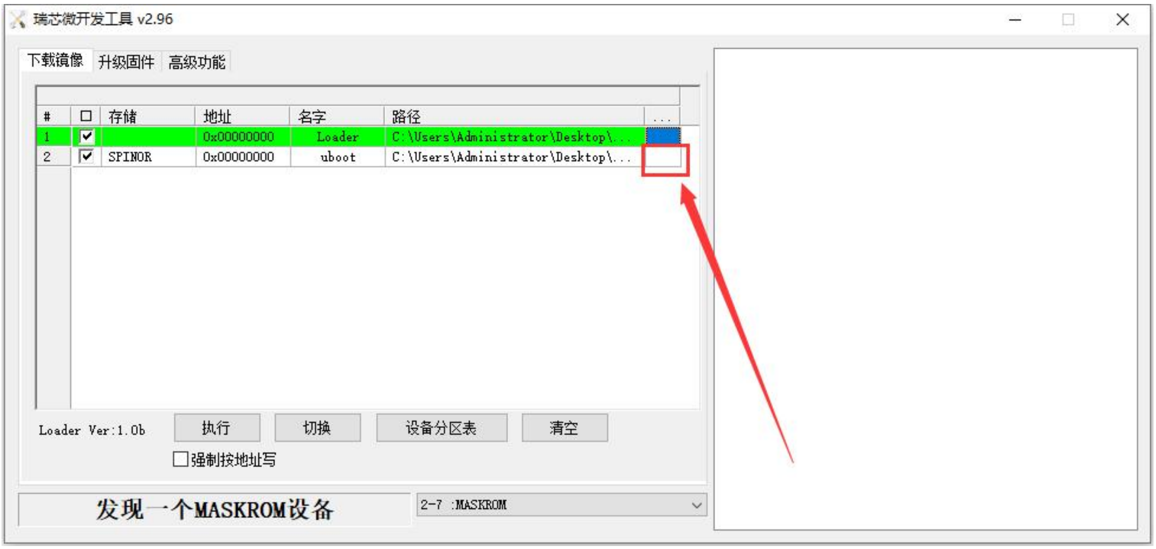
l. Then click the location shown in the picture below



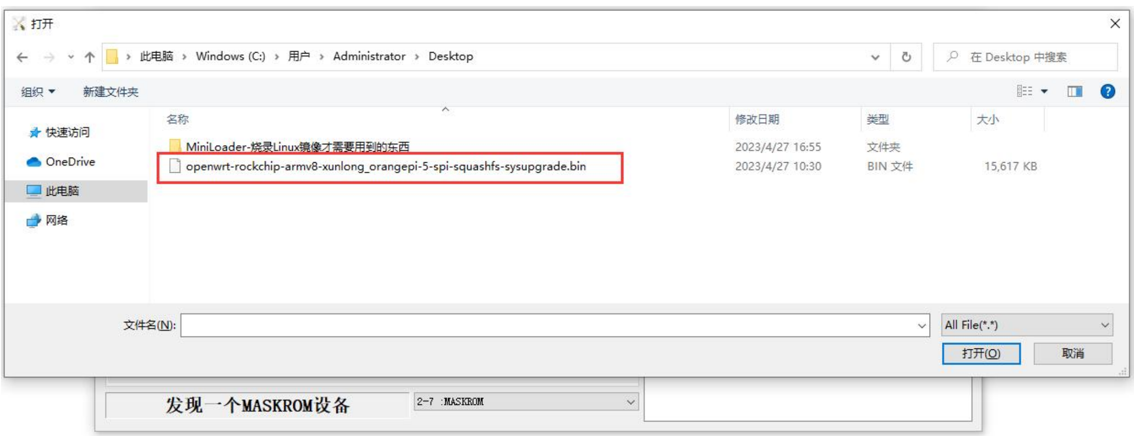
m. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click to **open**



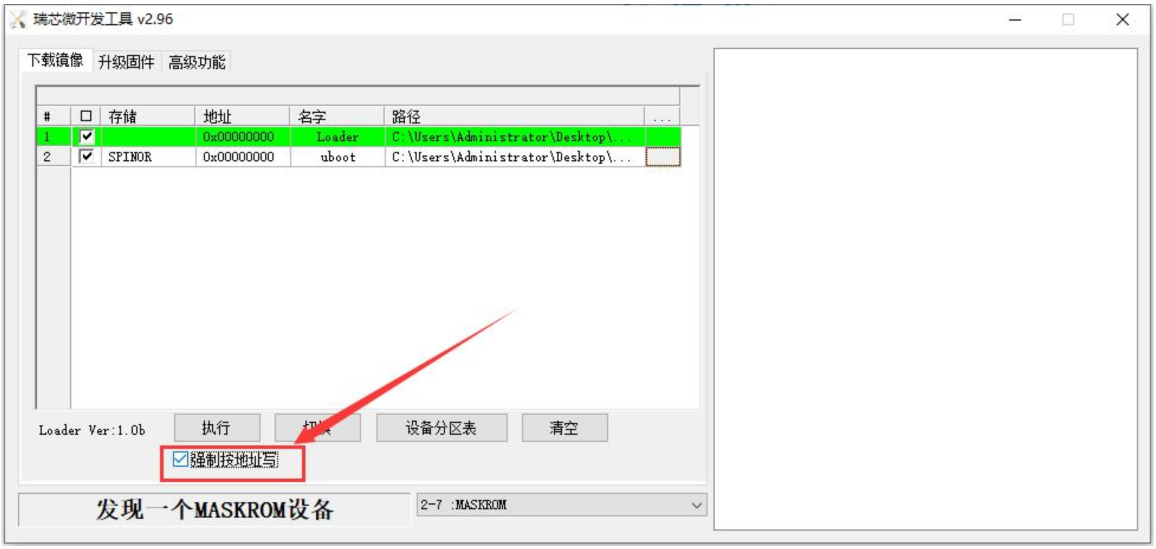
n. Then click on the location shown in the picture below



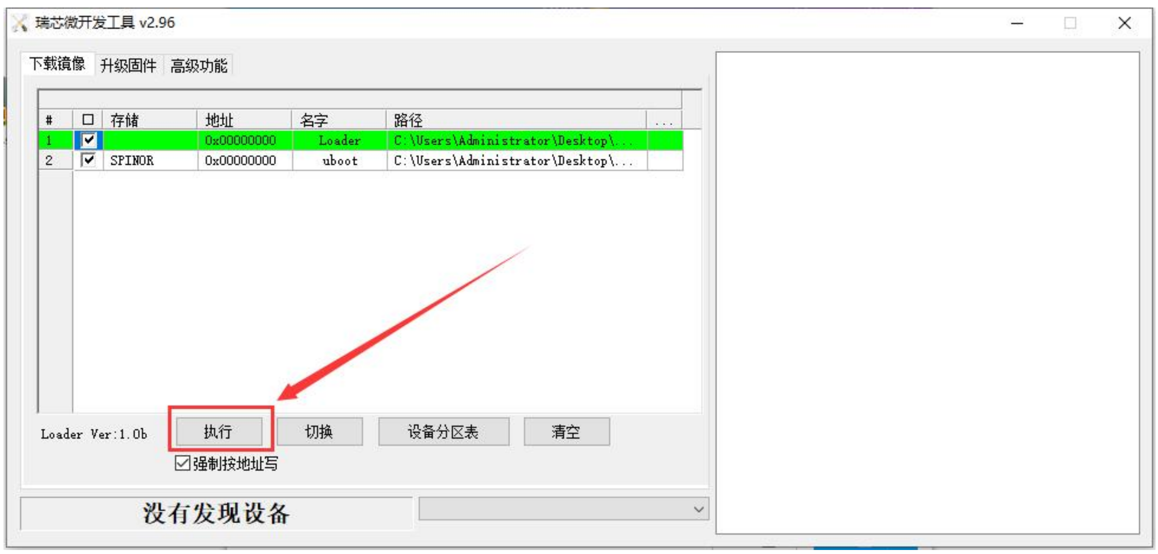
o. Then select the path of the OpenWRT image you want to burn, and then click **Open**



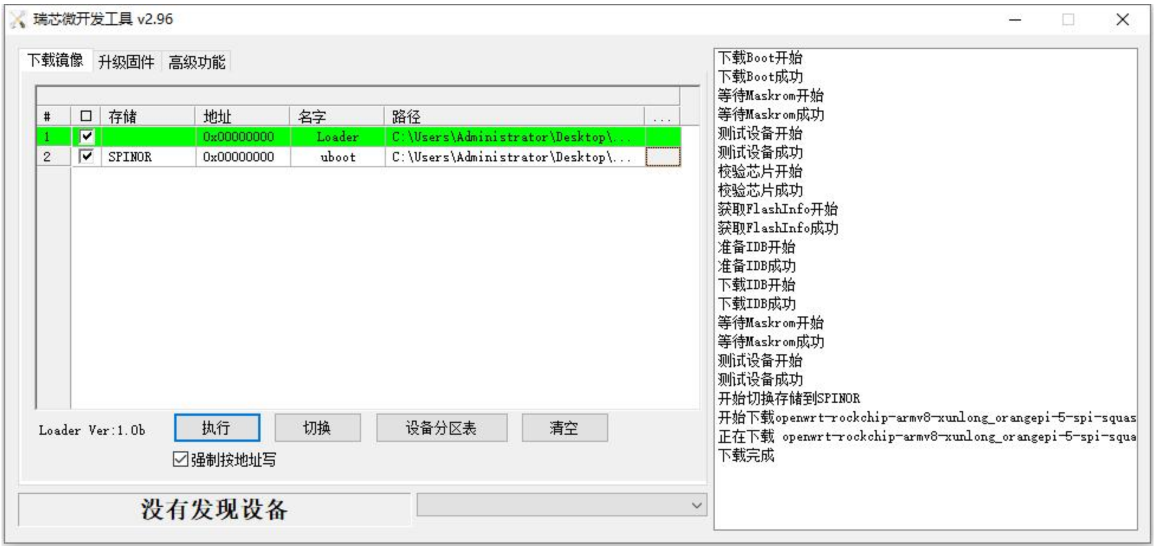
p. Then please make sure that **the Force writing by address option** is checked.



q. Click the **Execute** button again to start burning the OpenWRT image into SPIFlash.



r. The display log after the r.OpenWRT image is burned is as shown below



s. OpenWRT image will start automatically after burning. If it does not start normally, please power on again and try again.

2. 8. 2. How to use the mtd tool of the OpenWRT system for burning

1) Burning the OpenWRT image that supports spi boot into SPIFlash requires the use of a TF card, so you first need to burn the OpenWRT image that supports TF card boot to the TF card, and then use the TF card to boot the development board to enter the OpenWRT system. For the method of burning the OpenWRT image to the TF card, please refer to the instructions in the two sections: **How to burn the Linux image to the TF card based on Windows PC** and **the method of burning the Linux image to the TF card based on the Ubuntu PC**.

2) After the TF card starts the OpenWRT system, execute the following command to see that SPIFlash contains 3 partitions, namely uboot, dtb and firmware partitions, of which firmware contains the kernel and rootfs

```
root@OpenWrt:~# cat /proc/mtd
dev:    size    erasesize   name
mtd0: 00200000 00001000 "uboot"
mtd1: 00040000 00001000 "dtb"
mtd2: 00dc0000 00001000 "firmware"
```

3) Then download the `OpenWRT` image that can be booted from `SPIFlash` from the Orange Pi download page

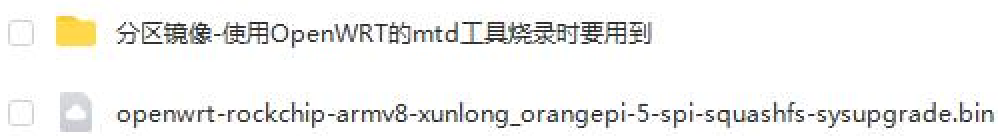
a. After opening the download link, you can see the following three types of

OpenWRT images. Please select the image in the `SPIFlash startup image` folder to download.

b. Then you can see a `"Partition Image"` folder and a separate OpenWRT image file. The difference between them is:

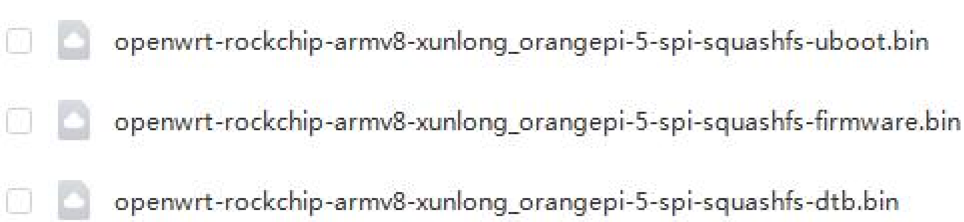
a) The three files u-boot, dtb, and firmware (including kernel and rootfs) files contained in the `"Partition Image"` folder, together form an OpenWRT image.

b) The image file shown in the figure below is a complete image file generated by packaging u-boot, dtb, and firmware in the `"Partition Image"` folder



c. Because only partitions can be burned in the OpenWRT system, here we need to select the `partition image` folder

d. After entering the `partition image` folder, you can see the following three image files, and then we need to download these three partition image files.



4) Then upload the 3 partition images downloaded from the official website to the TF card.

5) Then execute the following three commands to burn uboot, dtb and firmware partition images into the corresponding partitions of SPIFlash respectively.

```
root@OpenWrt:~# mtd -e uboot write openwrt-rockchip-armv8-xunlong_orangepi-5-spi-squashfs-uboot.bin uboot
Unlocking uboot ...
Erasing uboot ...

Writing from openwrt-rockchip-armv8-xunlong_orangepi-5-spi-squashfs-uboot.bin to uboot ...
```

```
root@OpenWrt:~# mtd -e dtb write openwrt-rockchip-armv8-xunlong_orangepi-5-spi-squashfs-dtb.bin dtb
Unlocking dtb ...
Erasing dtb ...

Writing from openwrt-rockchip-armv8-xunlong_orangepi-5-spi-squashfs-dtb.bin to dtb ...
```

```
root@OpenWrt:~# mtd -e firmware write openwrt-rockchip-armv8-xunlong_orangepi-5-spi-squashfs-firmware.bin    firmware
Unlocking firmware ...
Erasing firmware ...

Writing from openwrt-rockchip-armv8-xunlong_orangepi-5-spi-squashfs-firmware.bin to firmware ...
```

6) At this point, you can use the **poweroff** command to shut down the computer. Then please pull out the TF card and short press the power button to turn on the computer. At this time, the OpenWRT system in SPIFlash will be started.

2. 9. How to burn Android image to TF Card

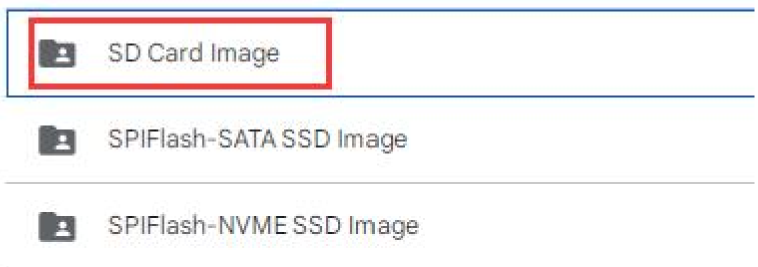
2) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card of SanDisk and other brands

3) Then use the card reader to insert the TF card into the card

4) Then download the SDDiskTool programming tool from **the Orange Pi data download page**, **please ensure that the version of the SDDiskTool tool is the latest v1.72**

5) Then download the Android12 image from **the Orange Pi download page**

a. After opening the download link of the Android image, you can see the following three types of Android images, please select the image in **SD card image** folder to download



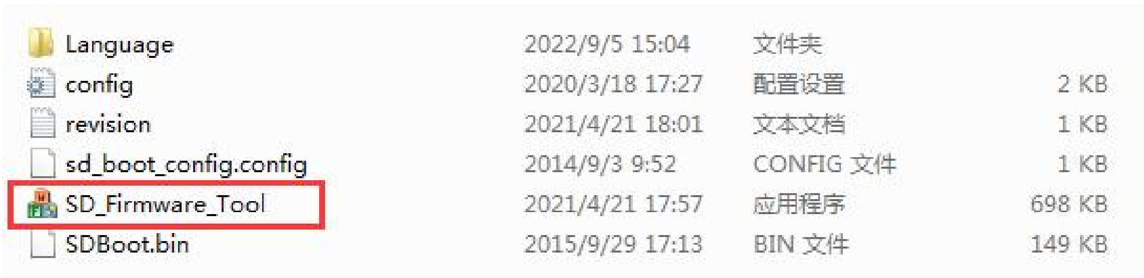
b. After entering the **TF card image** folder, you can see the following two images, the difference between them is:

a) The image without lcd is specially used for HDMI display and supports 8K display. If you do not use the LCD screen, please download the image without lcd

b) If you want to use LCD screen, please choose image with lcd

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

7) Then use decompression software to decompress **SDDiskTool_v1.72.zip**, this software does not need to be installed, just find **SD_Firmware_Tool.exe** in the decompressed folder and open it



8) After opening SDDiskTool, if the TF card is recognized normally, the inserted disk device will be displayed in the "**Select Removable Disk Device**" column. **Please make sure that the displayed disk device is consistent with the drive letter of the TF card you want to burn**, if there is no display, you can try to unplug the TF card .

9) After confirming the drive letter, you can format the TF card first, click the **restore disk button** in SDDiskTool, or use the **SD Card Formatter** mentioned above to format the TF card

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

a. First check "**SD Boot**" in "**Select Function Mode**"

b. Then select the path of the Android image in the "**Select to upgrade firmware**" column

c. Finally, click the "**Start Create**" button to start burning the Android image to the TF card

11) After burning, you can exit the SDDiskTool software, and then you can pull out the TF card from the computer and insert it into the development board to start

2.10. How to burn Android image to SPIFlash+NVMe SSD

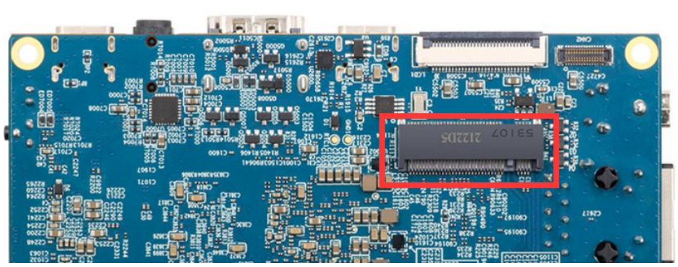
Note that all the following operations are performed on a Windows computer

1) First, you need to prepare an NVMe SSD solid state drive

a. The M.2 2230 SSD is as follow

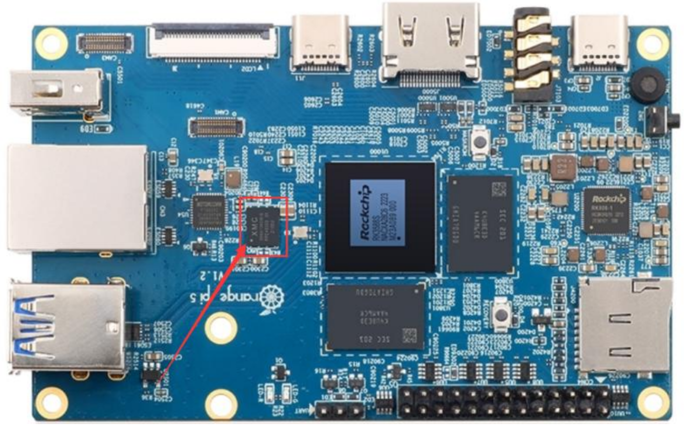
b. The M.2 2242 SSD is as follow

2) Then insert the NVMe SSD into the M.2 PCIe interface of the development board and fix it



3) The position of the SPI Flash on the development board is shown in the figure below,

no other settings are required before starting the programming

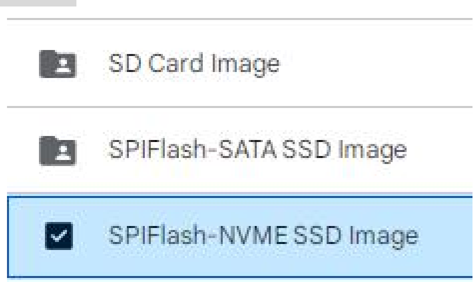


4) Prepare a data cable with a good quality Type-C interface

5) Then download the Rockchip driver **DriverAssitant_v5.12.zip** and the burning tool **RKDevTool_Release_v2.96.zip** from the **Orange Pi data download page**, please make sure that the version of the downloaded **RKDevTool** tool is **v2.96**

6) Then download the image of Android 12

a. After opening the download link of the Android image, you can see the following three types of Android images, please select the image in the **SPIFlash-NVME SSD** folder to download



b. After entering the **SPIFlash-NVME SSD** folder, you can see the following two images. Their differences are：

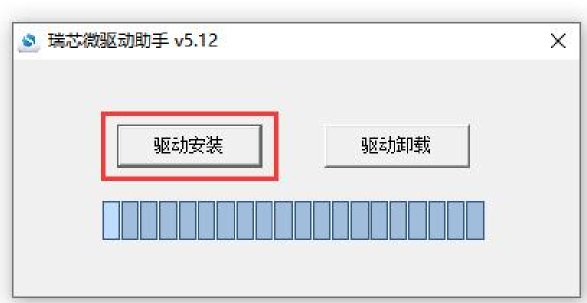
a) The image without lcd is specially used for HDMI display and supports 8K display. If you do not use the LCD screen, please download the image without lcd

b) If you want to use LCD screen, please choose image with LCD

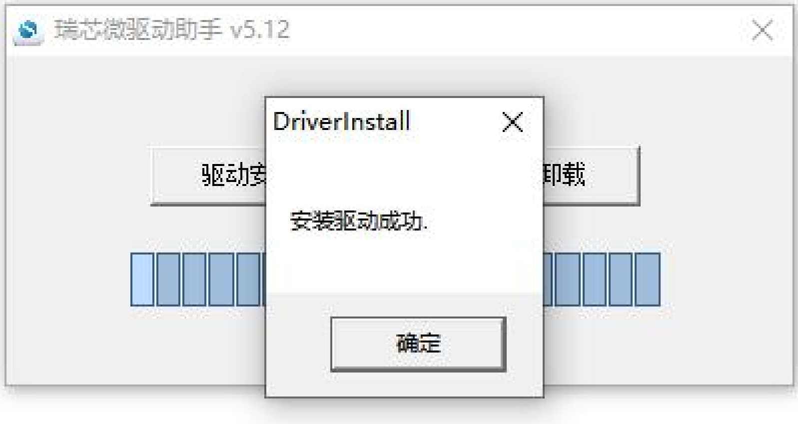
7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

8) After opening**DriverInstall.exe**, the steps to install the Rockchip driver are as follows

a. Click the "**Driver Install**"

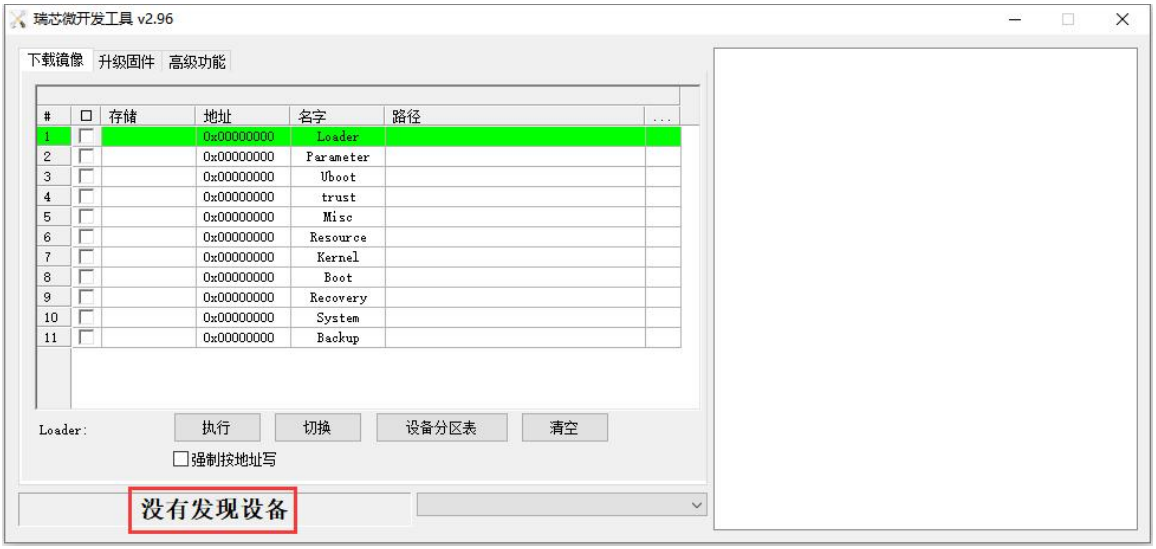


b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button



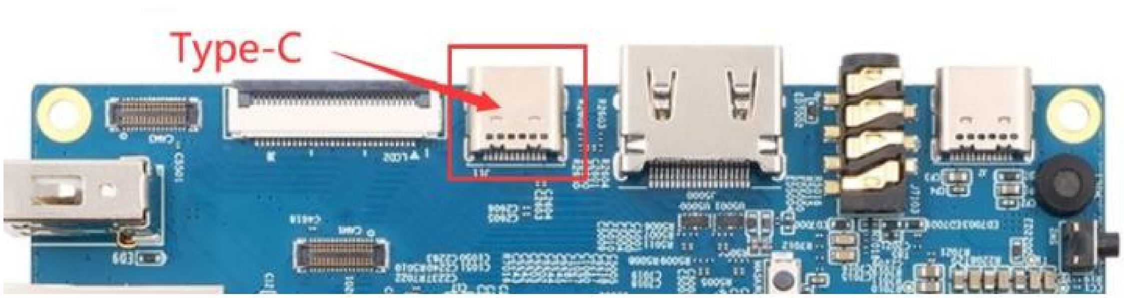
9) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

10) After opening the **RKDevTool** burning tool, because the computer has not been connected to the development board through the Type-C cable at this time, the lower left corner will prompt "**No device found**"



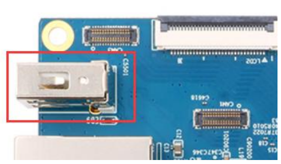
11) Then start burning the Android image to SPIFlash+NVMe SSD

a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below

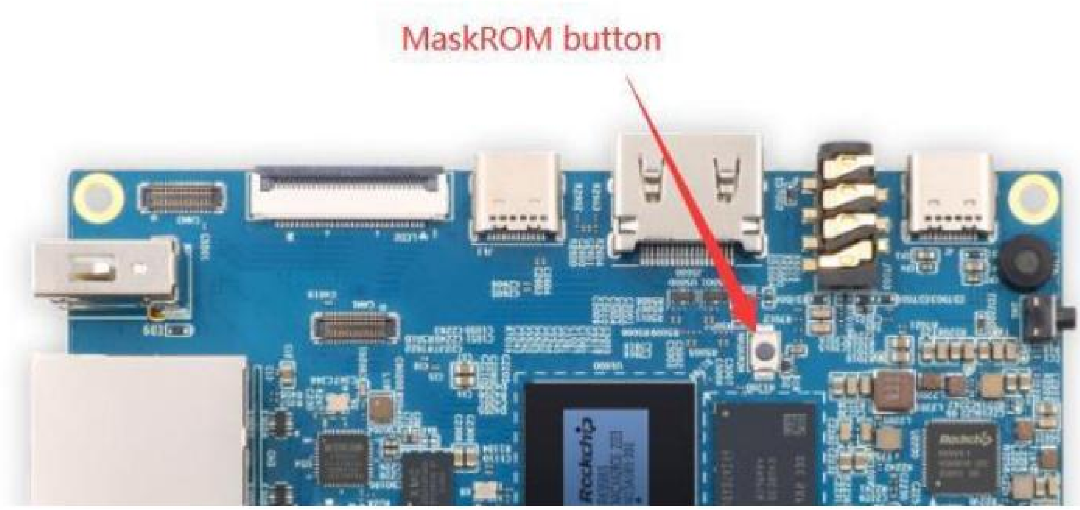


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

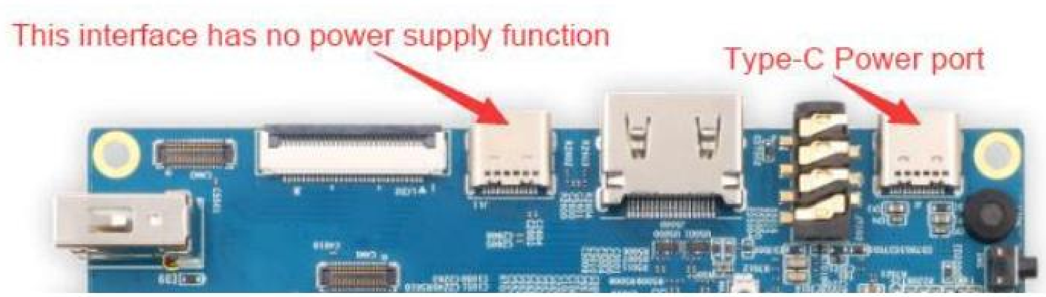
c. Also need to ensure that the white USB2.0 interface in the position shown below is not plugged into a USB device



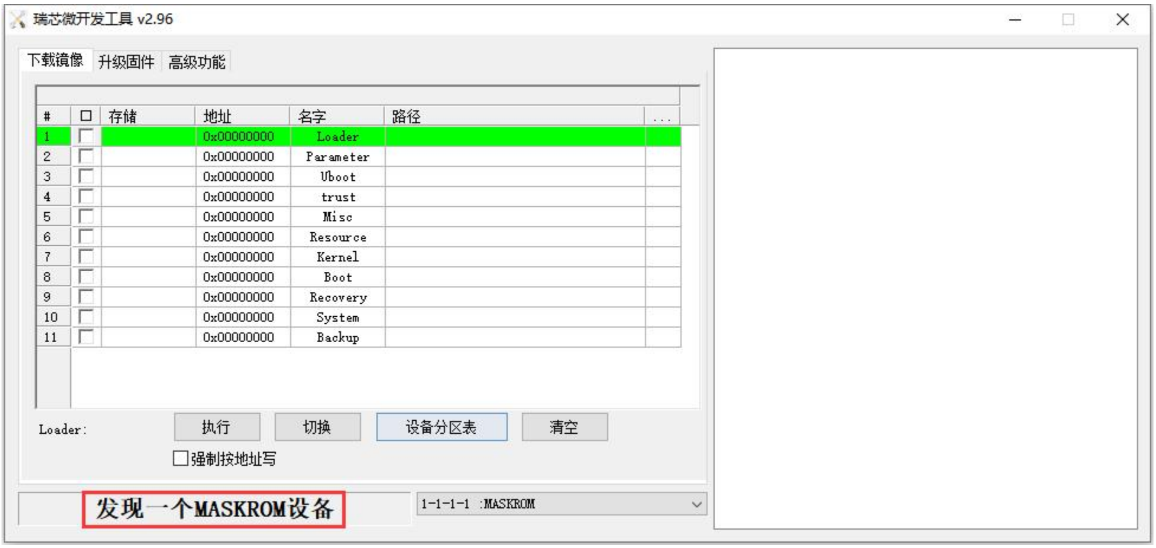
d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below：



e. Then connect the power supply of the Type-C interface to the development board and power on



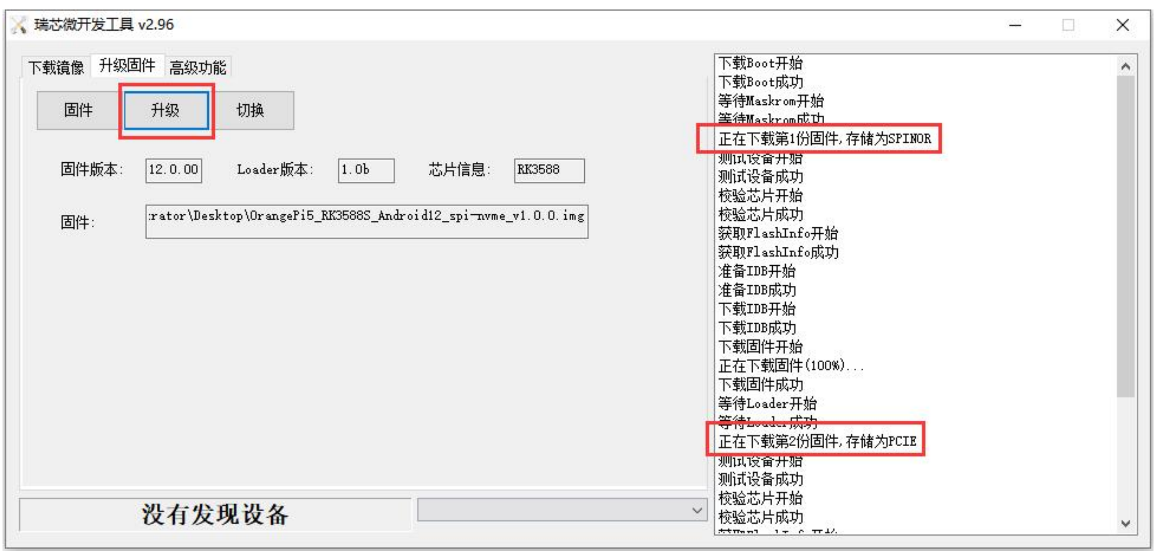
f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



g. Then click the "**Upgrade Firmware**" column of the burn

h. Then click the "**Firmware**" button to select the Android image to be burn

i. Finally, click the "**Upgrade**" button to start burning. The burning process is shown in the figure below. You can see that the firmware will be burned into SPIFlash first, and then burned into PCIE. After burning is completed, the Android system will start automatically



2. 11. How to burn Android image to SPI Flash+SATA SSD

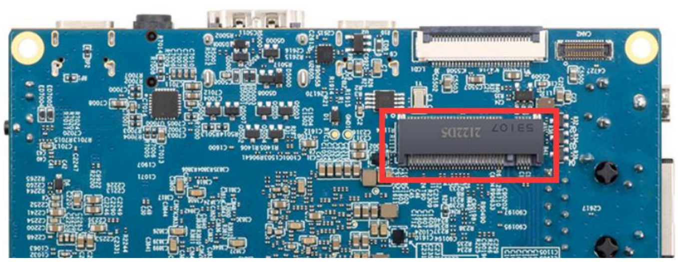
Note that all the following operations are performed on a Windows computer

1) First, you need to prepare a SATA SSD solid state drive

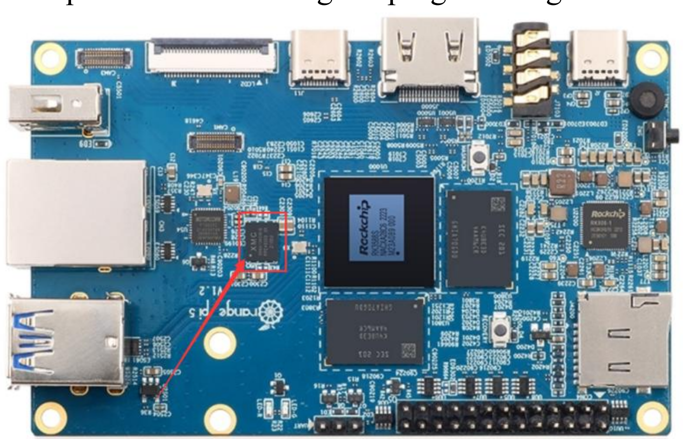
a. The M.2 2242 SSD is as follow

b. The 2280 specification SSD is as follows (the 2280 specification SATA SSD can also be used, but the SSD will exceed the development board after being inserted into the development board）

2) Then insert the SSD into the M.2 PCIe interface of the development board and fix it



3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming

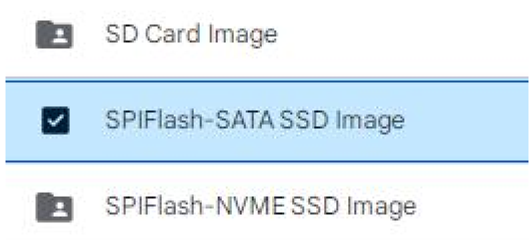


4) Prepare a data cable with a good quality Type-C

5) Then download the Rockchip driver **DriverAssitant_v5.12.zip** and the burning tool **RKDevTool_Release_v2.96.zip**.zip from **the Orange Pi data download page**, please make sure that the version of the downloaded **RKDevTool** tool is **v2.96**

6) Then download the image of Android 12

a. After opening the download link of the Android image, you can see the following three types of Android images, please select the image in the **SPIFlash-SATA SSD** folder to download



b. 进入 After entering the **SPIFlash-SATA SSD** folder, you can see the following two images, the difference between them is：

a) The image without lcd is specially used for HDMI display and supports 8K display. If you do not use the LCD screen, please download the image without lcd

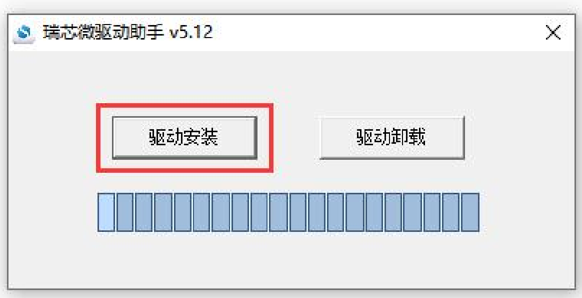
b) If you want to use LCD screen, please choose image with lcd

7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**， and

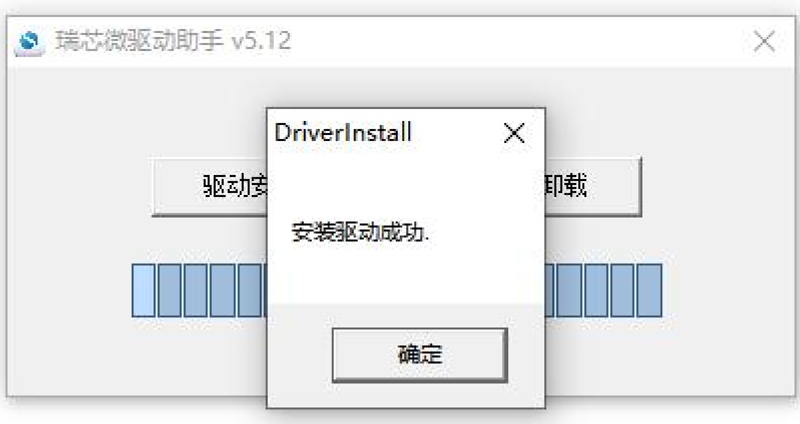
then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

8) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

a. Click the "**Driver Installation**" button

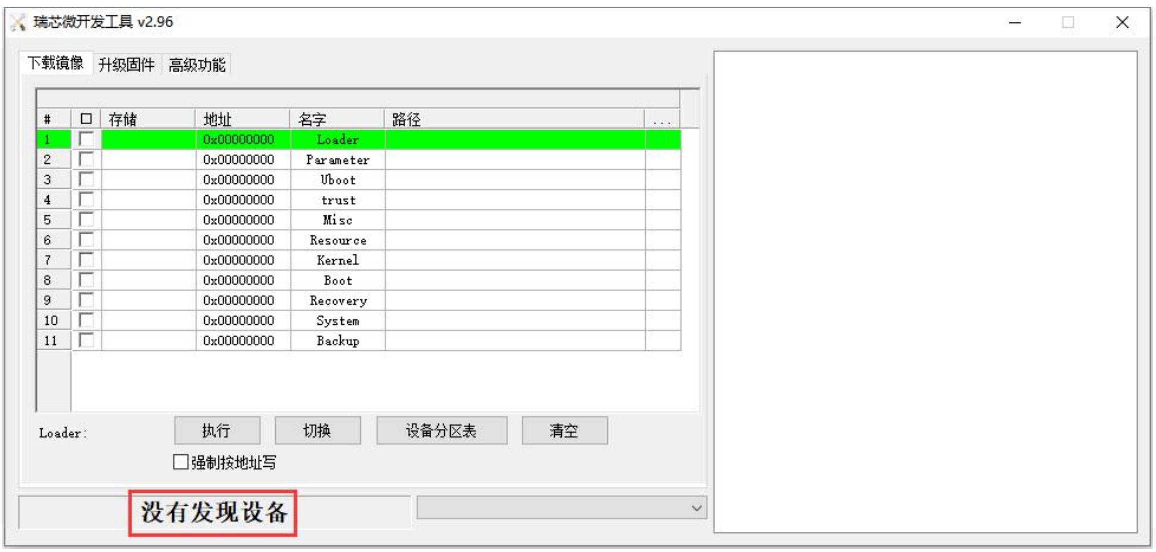


b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button



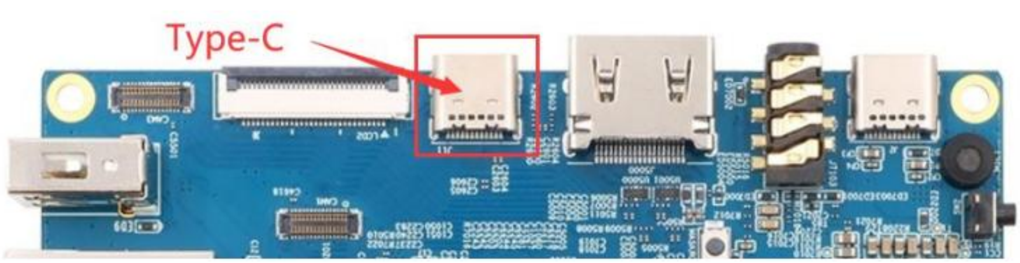
9) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

10) After opening the **RKDevTool** burning tool, because the computer has not been connected to the development board through the Type-C cable at this time, the lower left corner will prompt **"No device found"**



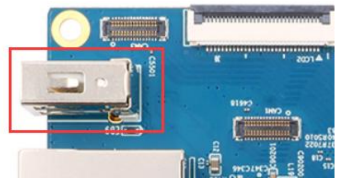
11) Then start burning the Android image to SPIFlash+SATA SSD

a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below



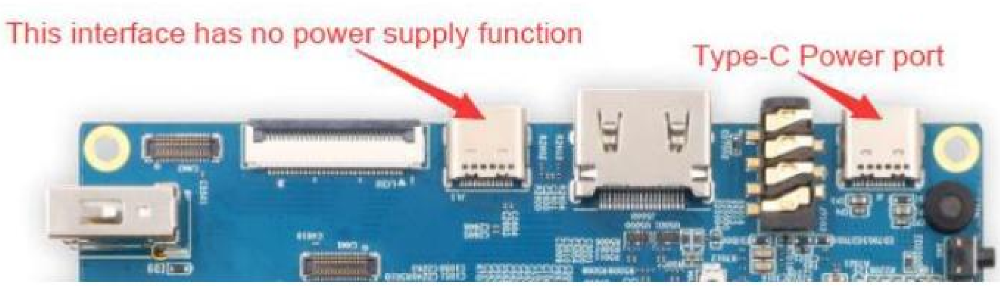
b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

c. Also need to ensure that the white USB2.0 interface in the position shown below is not plugged into a USB device

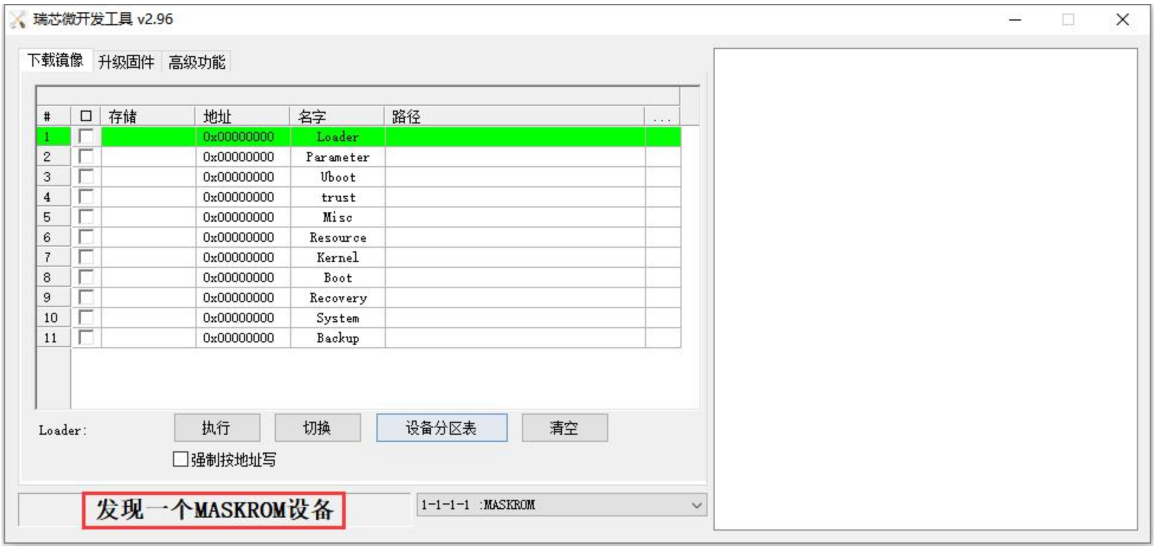


d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:

e. Then connect the power supply of the Type-C interface to the development board and power on



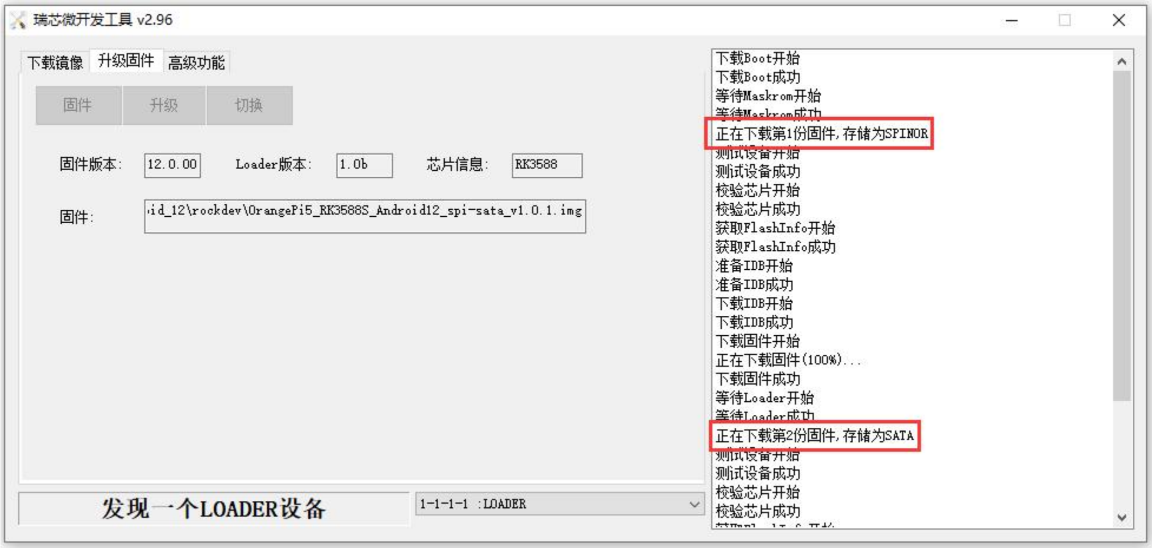
f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



g. Then click the "**Upgrade Firmware**" column of the burning tool

h. Then click the **"Firmware"** button to select the Android image to be burned

i. Finally, click the "Upgrade" button to start burning. The burning process is shown in the figure below. As you can see, the firmware will be burned to SPIFlash first, and then the firmware will be burned to SATA SSD. After burning is completed, the Android system will start automatically 动。



2.12. Method of burning Orange Pi OS (Droid) image into TF card

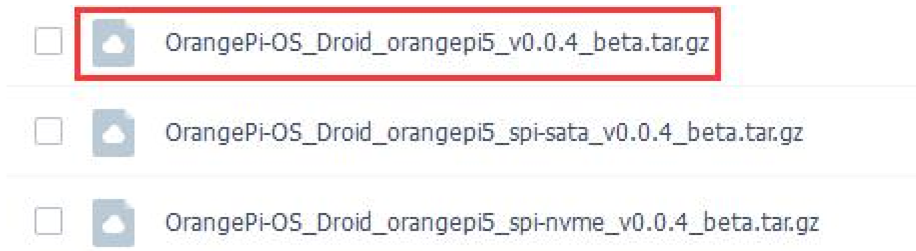
Note,that all the operations below are performed on a Windows computer.

1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card of SanDisk and other brands

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

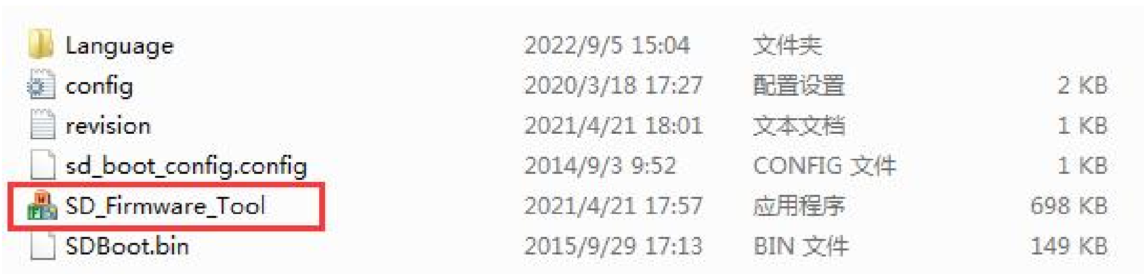
3) Then download the SDDiskTool programming tool from **the Orange Pi data download page**, **please ensure that the version of the SDDiskTool tool is the latest v1.72**

4) Then download the Orange Pi OS (Droid) image from **the Orange Pi data download page**, open the download link of the Orange Pi OS (Droid) image and you can see the following three types of images, please select the image below



5) Then use the decompression software to decompress the compressed package of the downloaded Orange Pi OS (Droid) image. Among the decompressed files, the file ending with ".img" is the Orange Pi OS (Droid) image file, and the size is more than 1GB

6) Then use the decompression software to decompress **SDDiskTool_v1.72.zip**, this software does not need to be installed, just find **SD_Firmware_Tool.exe** in the decompressed folder and open it



7) After opening **SDDiskTool**, if the TF card is recognized normally, the inserted disk device will be displayed in the "**Select Removable Disk Device**" column. **Please make sure that the displayed disk device is consistent with the drive letter of the TF card you want to burn Yes**, if there is no display, you can try to unplug the TF card

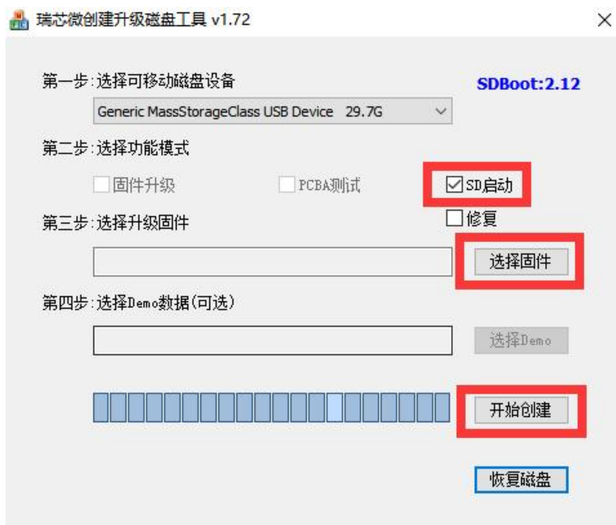
8) After confirming the drive letter, you can format the TF card first, **click the restore disk** button in **SDDiskTool**, or use the **SD Card Formatter** mentioned above to format the TF card

9) Then start to write the Orange Pi OS (Droid) image to the TF card

a. First check "**SD Boot**" in "**Select Function Mode**"

b. Then select the path of the Orange Pi OS (Droid) image in the "**Select to upgrade firmware**" column

c. Finally, click the "**Start Create**" button to start burning the Orange Pi OS (Droid) image to the TF card



10) After burning, you can exit the SDDiskTool software, and then you can pull out the TF card from the computer and insert it into the development board to start

2. 13. Burn Orange Pi OS (Droid) image to SPIFlash+NVMe SSD

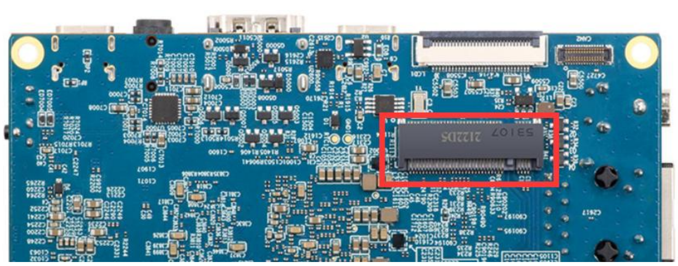
Note that all the following operations are performed on a Windows computer.

1) First, you need to prepare an NVMe SSD solid state drive

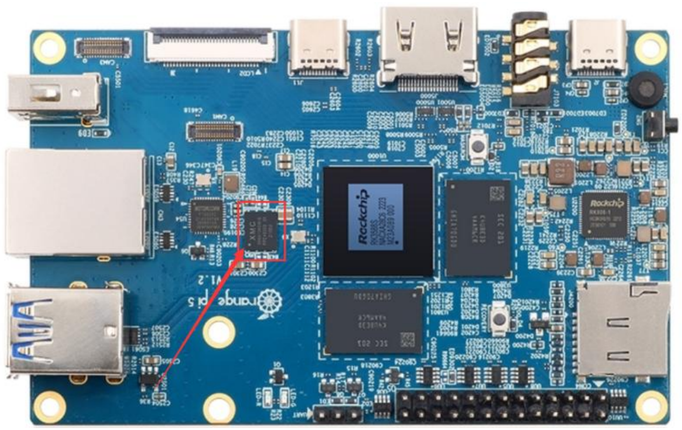
a. The a.M.2 2230 SSD is as follows

b. The M.2 2242 SSD is as follows

2) Then insert the NVMe SSD into the M.2 PCIe interface of the development board and fix it



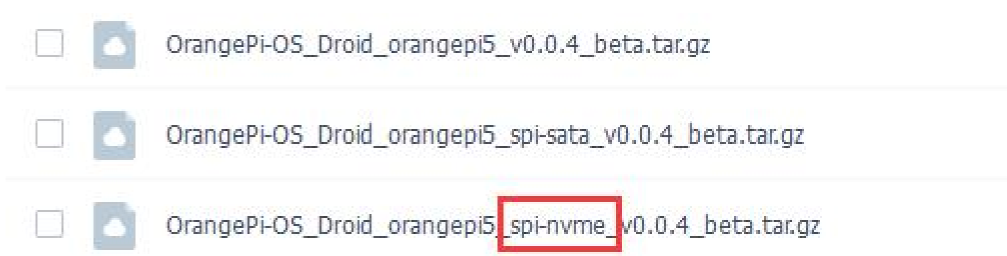
3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming



4) It is also necessary to prepare a data cable with a good quality Type-C interface

5) Then download Rockchip **DriverAssitant_v5.12.zip** and burning tool **RKDevTool_Release_v2.96.zip** from the **Orange Pi data download page**, please make sure that the version of the downloaded **RKDevTool** tool is **v2.96**

6) Then download the Orange Pi OS (Droid) image, open the download link of the Orange Pi OS (Droid) image, you can see the following three types of images, please choose the image with **spi-nvme** to download

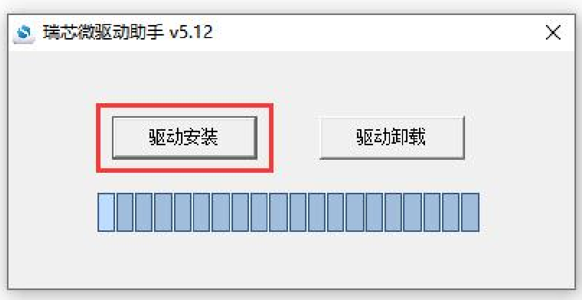


7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and

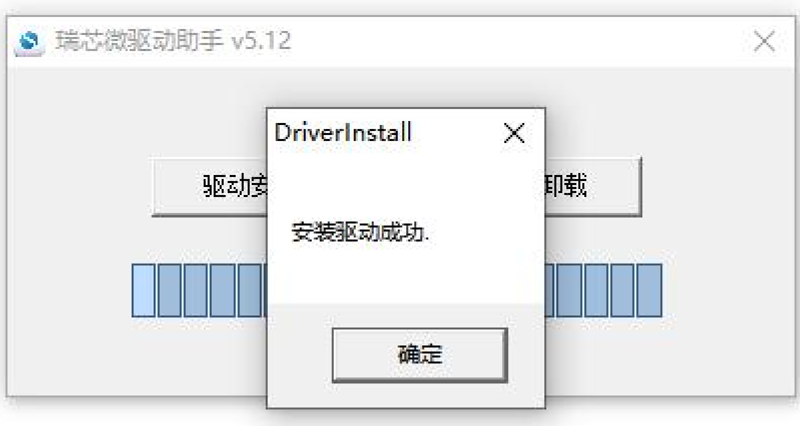
then find the **DriverInstall.exe** executable file in the decompressed folder and open it

8) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows **DriverInstall.exe**

a. Click the "**Driver Installation**" button 击

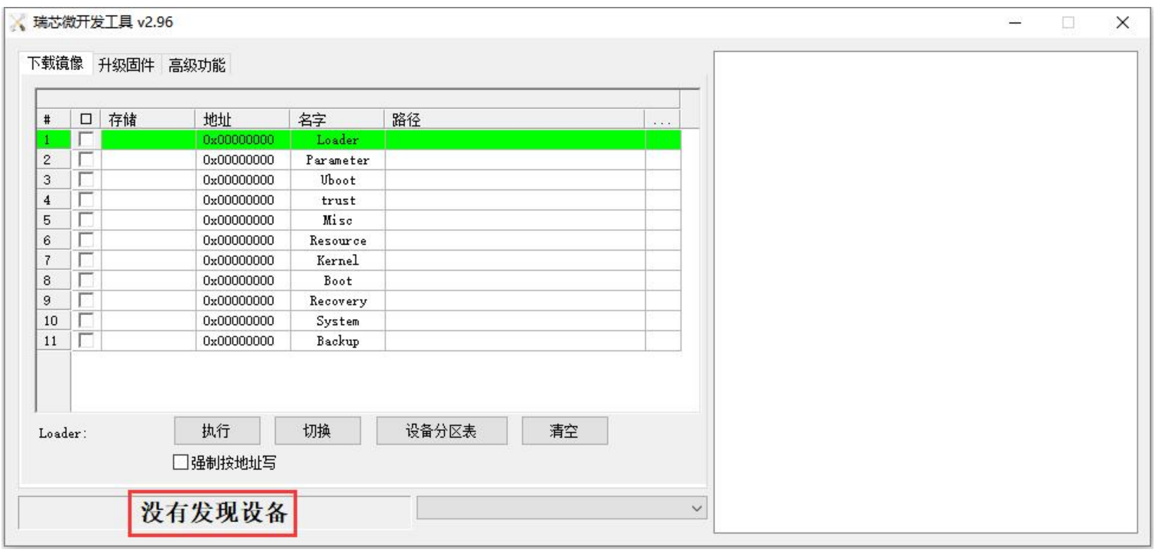


b. 等 After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the **"OK"** button



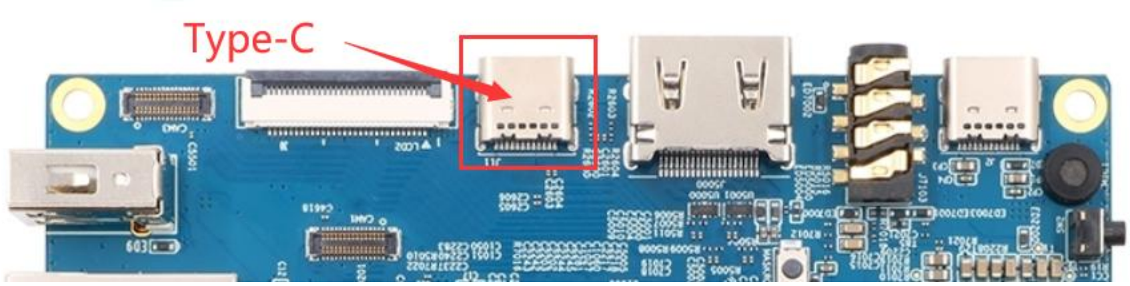
9) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

10) After opening the **RKDevTool** burning tool, because the computer is not connected to the development board through the Type-C cable at this time, the lower left corner will prompt "**No device found**"



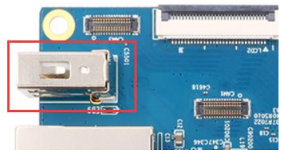
11) Then start burning the Orange Pi OS (Droid) image to SPIFlash+NVMe SSD

a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below

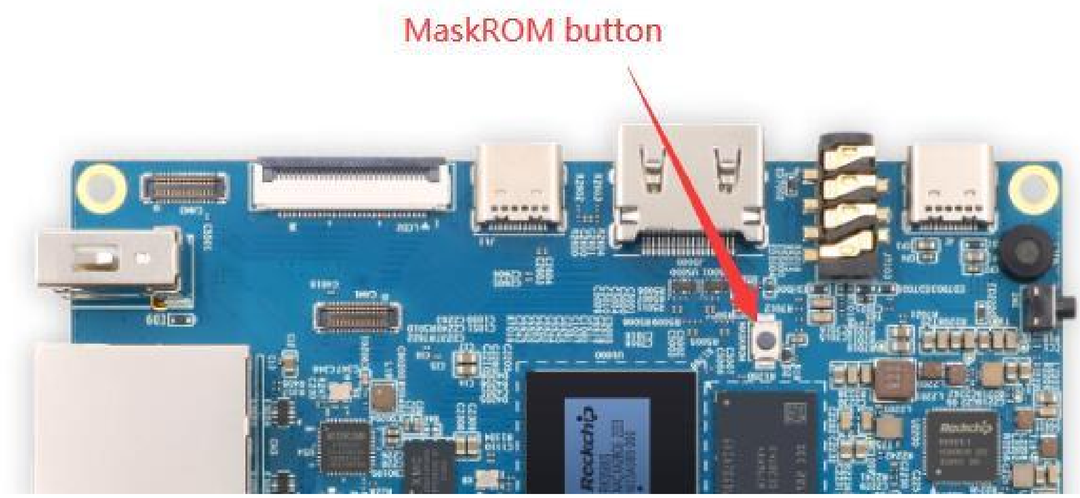


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

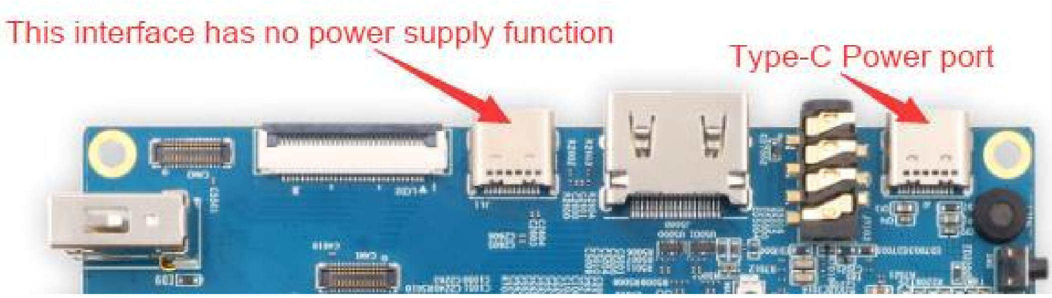
c. It is also necessary to ensure that the white USB2.0 interface in the position shown below is not plugged into a USB device



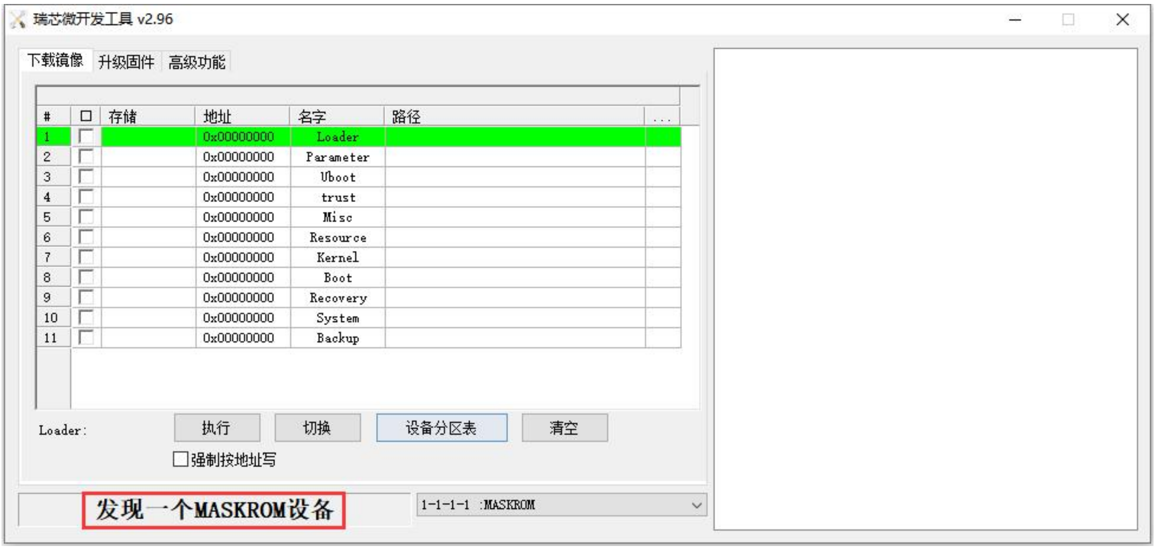
d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:



e. Then connect the power supply of the Type-C interface to the development board, and power on, and then release the MaskROM button



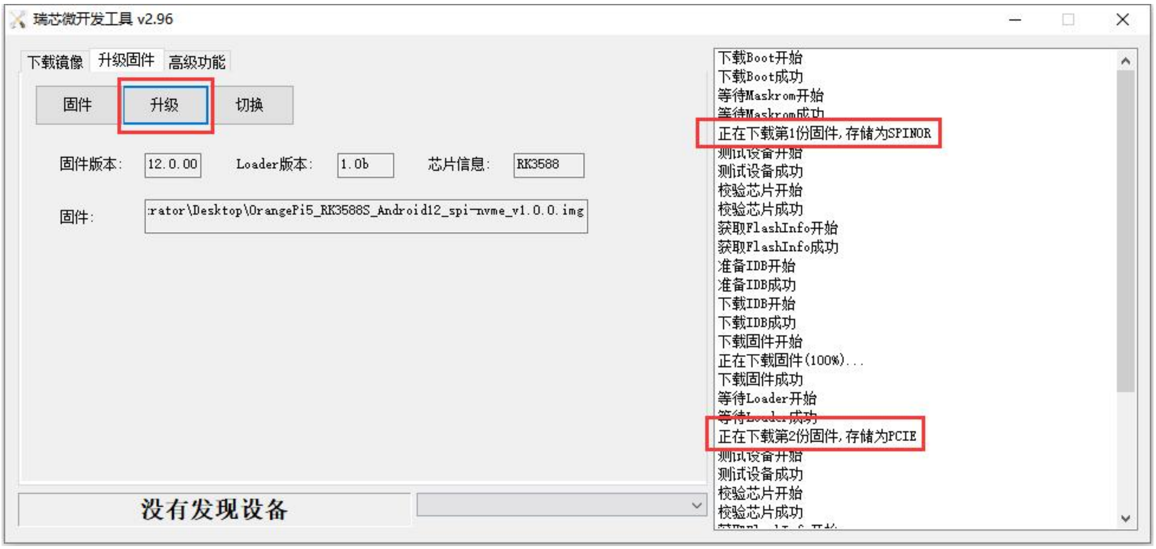
f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



g. Then click the "**Upgrade Firmware**" column of the burning tool

h. Then click the "**Firmware**" button to select the Orange Pi OS (Droid) image to be burned

i. Finally, click the "**Upgrade**" button to start burning. The burning process is shown in the figure below. You can see that the firmware will be burned into SPIFlash first, and then burned into PCIE. After burning, the Orange Pi OS (Droid) system will start automatically.



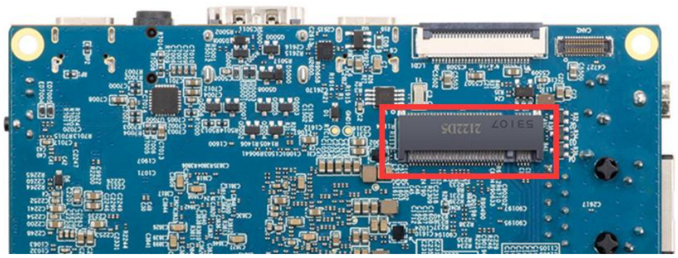
2. 14. Burn Orange Pi OS (Droid) image to SPIFlash+SATA SSD

Note, All the following operations are performed on a Windows computer.

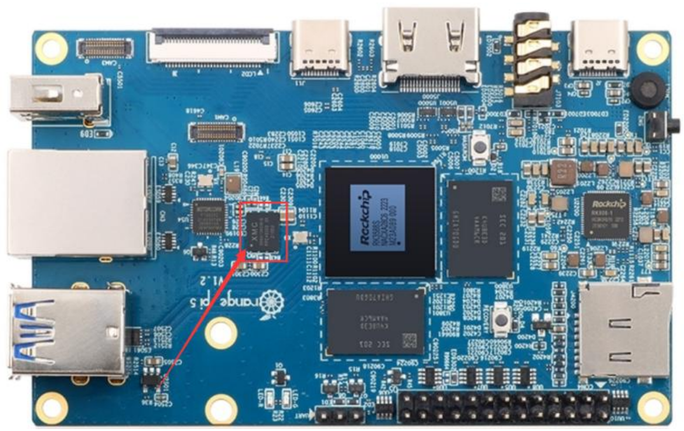
1) First, you need to prepare a SATA SSD solid state drive
   a. The a.M.2 2242 SSD is as follows

   b. M.2 The 2280 specification SSD is as follows (the 2280 specification SATA SSD can also be used, but the SSD will exceed the development board after being inserted into the development board)

2) Then insert the SSD into the M.2 PCIe interface of the development board and fix it



3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming

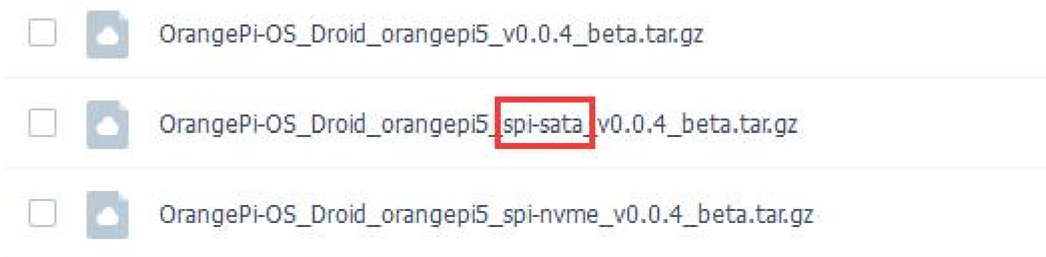


4) It is also necessary to prepare a data cable with a good quality Type-C interface

5) Then download the Rockchip driver **DriverAssitant_v5.12.zip** and the burning tool **RKDevTool_Release_v2.96.zip** from **the Orange Pi data download page**, please make sure that the version of the downloaded **RKDevTool** tool is **v2.96**

6) Then download the Orange Pi OS (Droid) image, open the download link of the Orange Pi OS (Droid) image and you can see the following three types of images, please

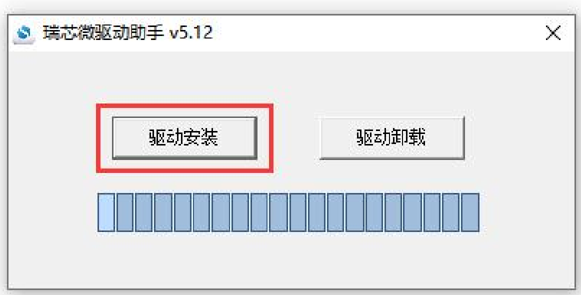
select the image with **spi-sata** to download



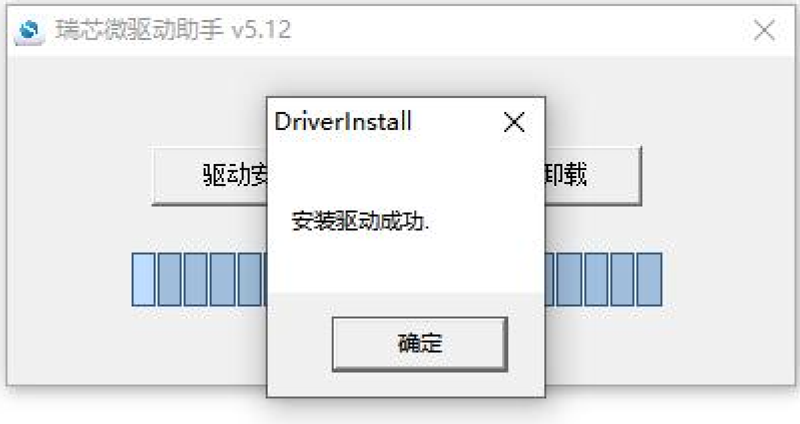
7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

8) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

a. Click the "**Driver Installation**" button

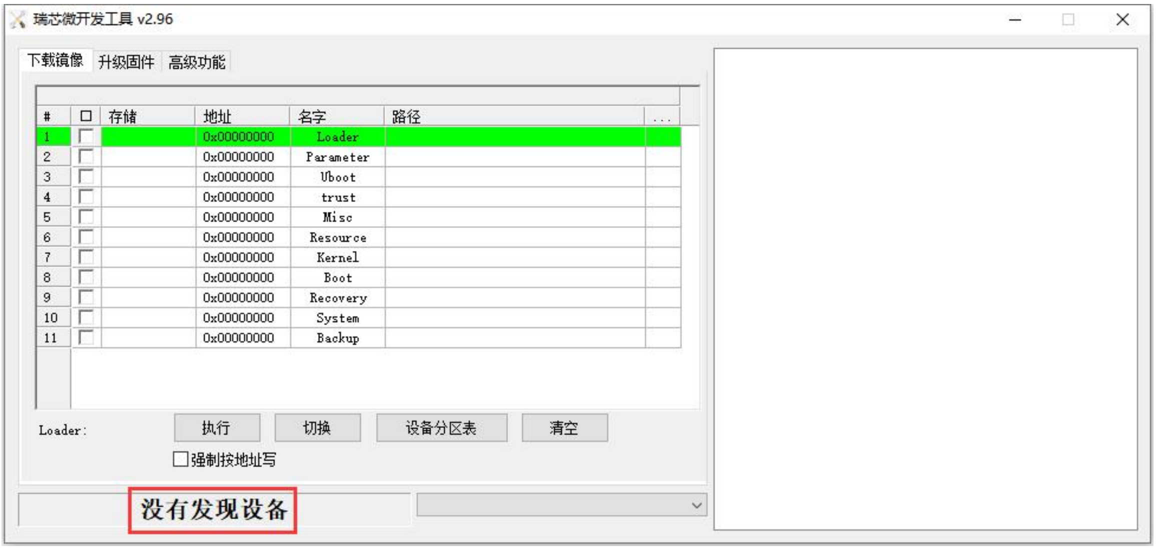


b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button



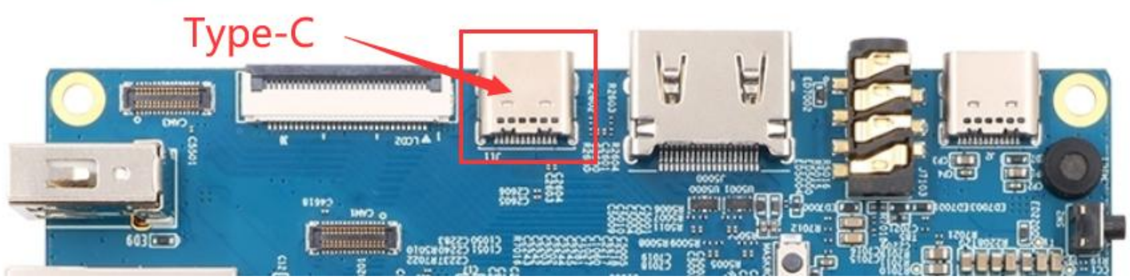
9) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

10) After opening the **RKDevTool** burning tool, because the computer is not connected to the development board through the Type-C cable at this time, the lower left corner will prompt "**No device found**"



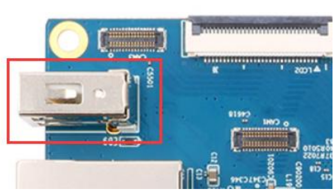
11) Then start burning the Orange Pi OS (Droid) image to SPIFlash+SATA SSD

a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below

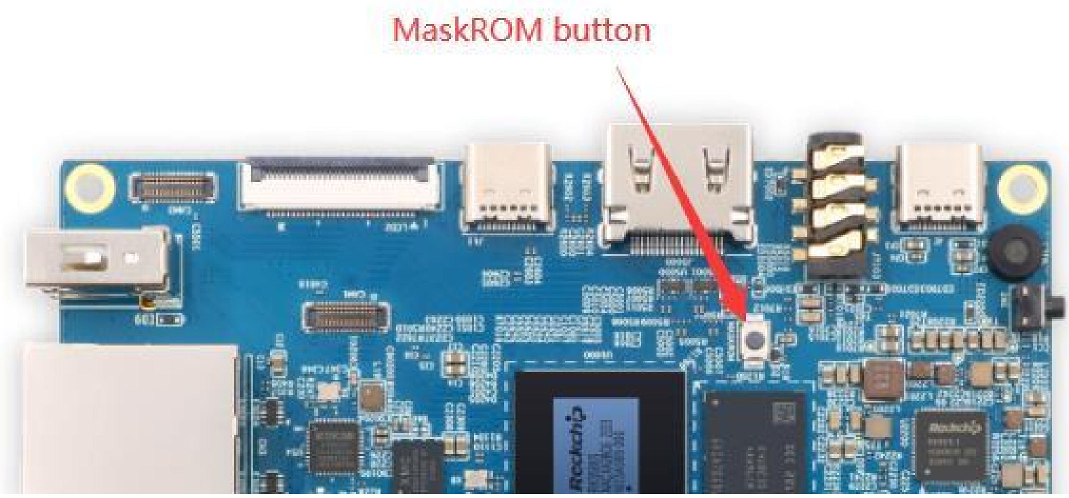


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

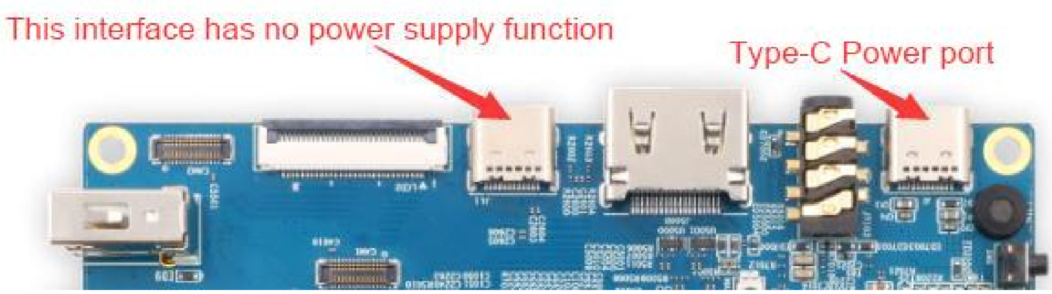
c. Also need to ensure that the white USB2.0 interface in the position shown below is not plugged into a USB device



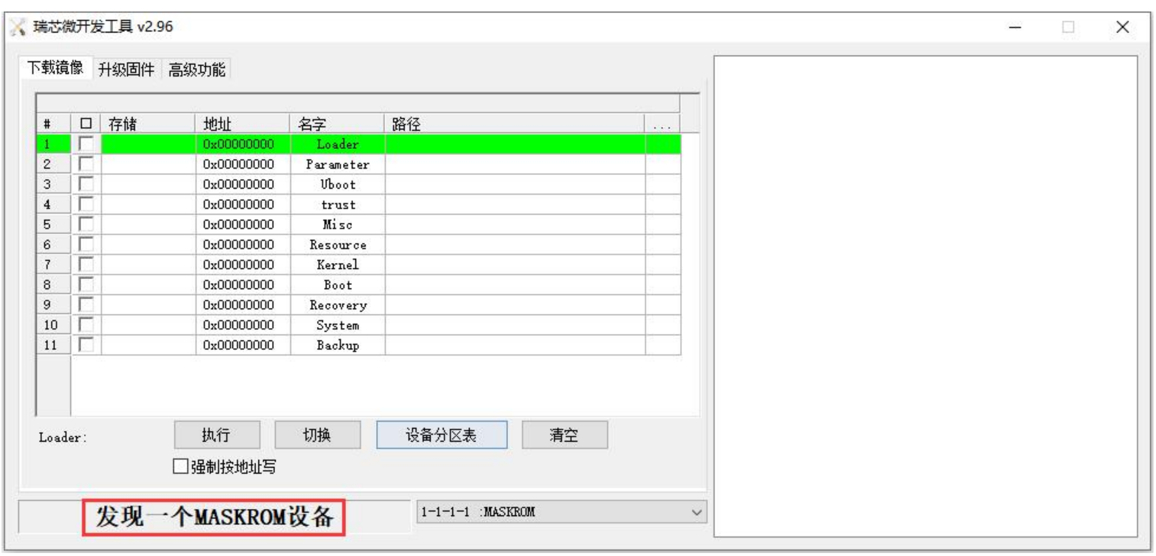
d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:



e. Then connect the power supply of the Type-C interface to the development board, and power on, and then release the MaskROM button



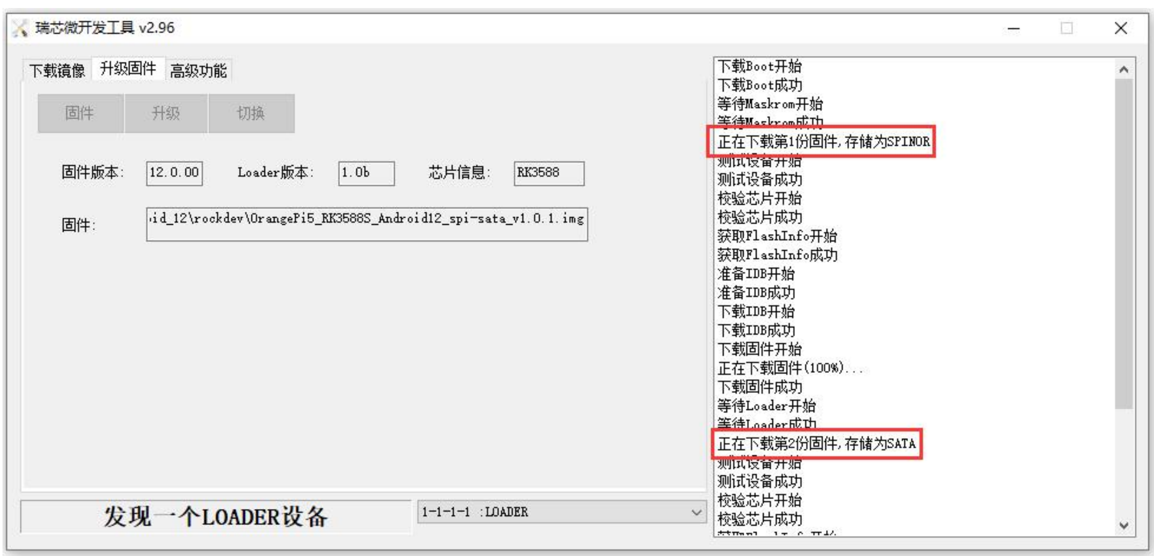
f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



g. Then click the "**Upgrade Firmware**" column of the burning tool

h. Then click the "**Firmware**" button to select the Orange Pi OS (Droid) image to be burned

i. Finally, click the "**Upgrade**" button to start burning. The burning process is shown in the figure below. You can see that the firmware will be burned to SPIFlash first, and then the firmware will be burned to SATA SSD. After burning, the Orange Pi OS (Droid) system will start automatically.



2. 15. How to burn Orange Pi OS (OH) image to TF card

Note that all operations below are performed on a Windows computer.

1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF

card must be class10 or above. It is recommended to use TF cards from SanDisk and other brands.

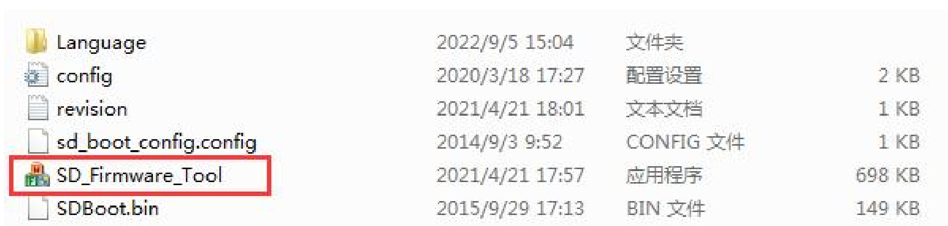
2) Then use the card reader to insert the TF card into the computer

3) Then download the SDDiskTool burning tool from the Orange Pi download page. **Please ensure that the version of the SDDiskTool tool is the latest v1.72**

4) Then download the image of Orange Pi OS (OH) from the Orange Pi download page

5) Then use decompression software to decompress the compressed package of the downloaded Orange Pi OS (OH) image. In the decompressed file, the file ending with ".img" is the Orange Pi OS (OH) image file, with a size of more than 1GB.

6) Then use decompression software to decompress **SDDiskTool_v1.72.zip**. This software does not need to be installed. Just find **SD_Firmware_Tool.exe** in the decompressed folder and open it.



7) After opening SDDiskTool, if the TF card is recognized normally, the inserted disk device will be displayed in the "**Select Removable Disk Device**" column. Please make sure that the displayed disk device is consistent with the drive letter of the TF card you want to burn. Yes, if there is no display, you can try to remove the TF card.

8) After confirming the drive letter, you can format the TF card first and click the **Recover Disk** button in **SDDiskTool**. You can also use the **SD Card Formatter** mentioned earlier to format the TF card.

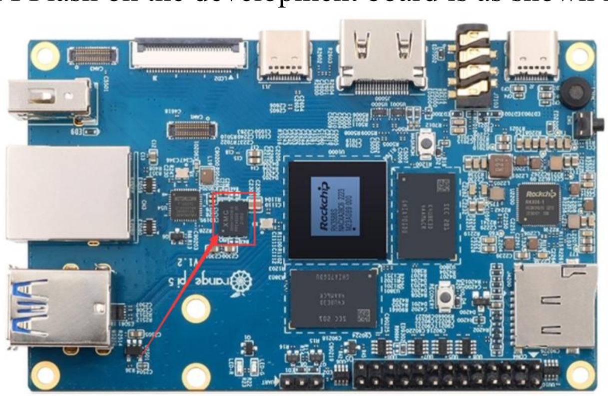
9) Then start writing the Orange Pi OS (OH) image to the TF card

b. First check "**SD boot**" in "**Select Function Mode**"
c. Then select the path to the Orange Pi OS (OH) image in the "**Select Upgrade Firmware**" column
d. Finally, click the "**Start Creating**" button to start burning the Orange Pi OS (OH) image to the TF card.

10) After burning, you can exit the SDDiskTool software, and then you can pull out the TF card from the computer and insert it into the development board to start.

2. 16. How to clear SPIFlash using RKDevTool

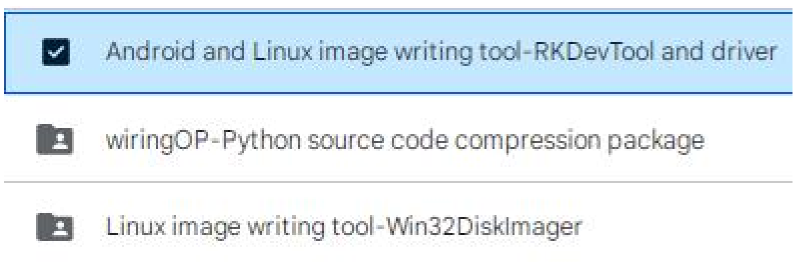
1) The location of SPI Flash on the development board is as shown in the figure below



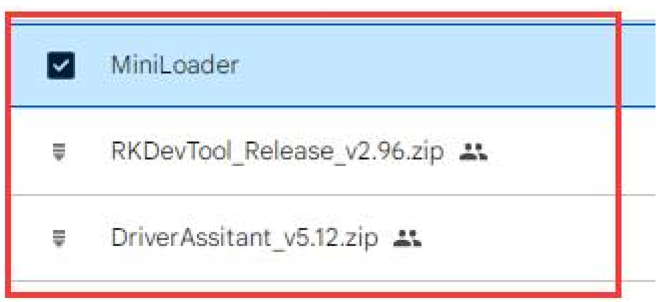
2) First you need to prepare a good quality Type-C interface data cable

3) Then download the Rockchip microdriver **DriverAssitant_v5.12.zip** and **MiniLoader** and the burning tool **RKDevTool_Release_v3.15.zip** from the Orange Pi data download page

a. On the Orange Pi data download page, first select the **official tool**, and then enter the folder below



b. Then download all the files below

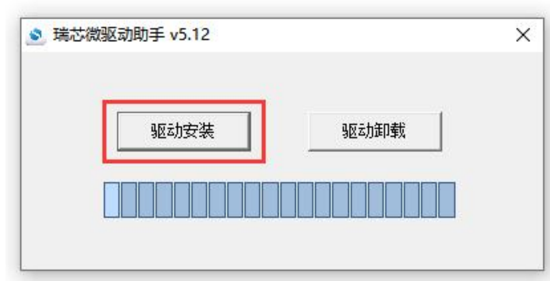


Note that the "MiniLoader-things needed to burn Linux images" folder will be referred to as the MiniLoader folder below.

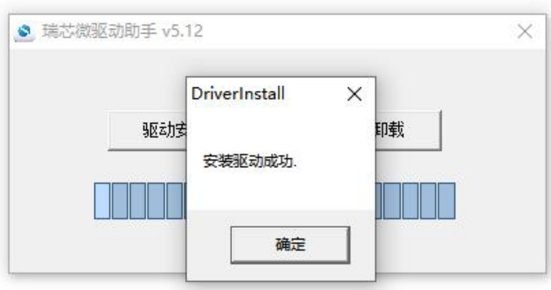
4) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

5) Open **DriverInstall.exe** and install the Rockchip microdriver as follows:

a. Click the "**Driver Installation**" button



b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



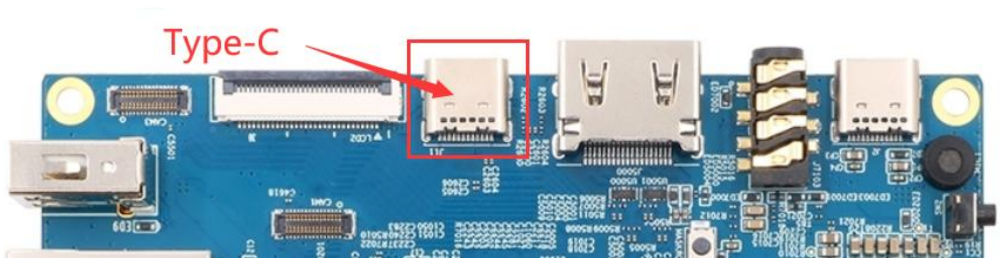
6) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

7) After opening the **RKDevTool** burning tool, because the computer has not yet been

connected to the development board through the Type-C cable, a message "**No device found**" will be displayed in the lower left corner.

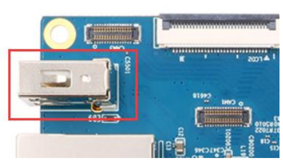
8) Then you can start clearing the contents of SPI FLASH

a. First, connect the development board and Windows computer through the Type-C data cable. The location of the Type-C interface of the development board is as shown in the figure below.

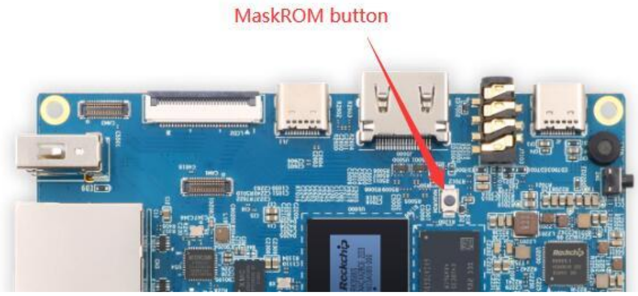


b. Make sure that the TF card is not inserted into the development board and the power supply is not connected.

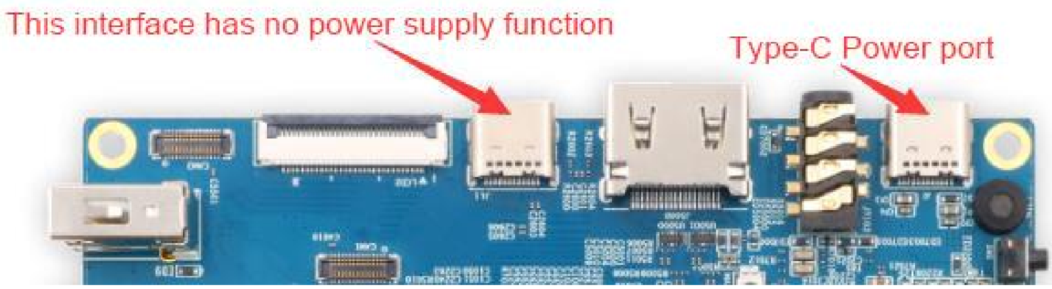
c. Also make sure that no USB device is plugged into the white USB2.0 interface in the picture below



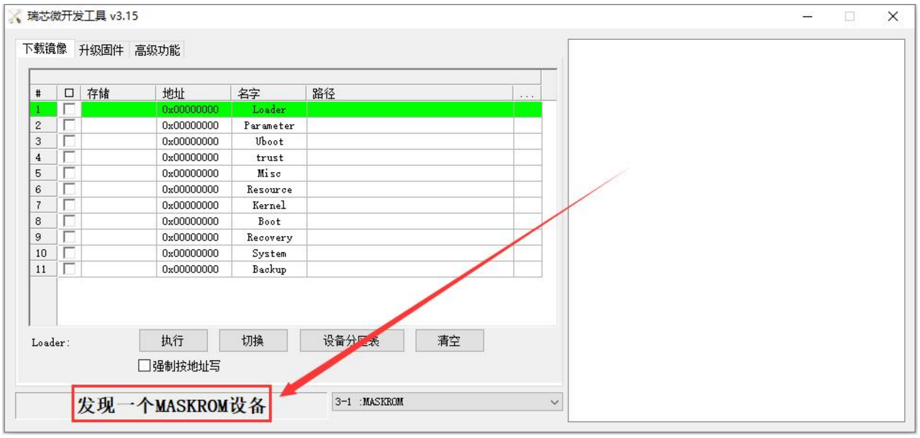
d. Then press and hold the MaskROM button of the development board. The position of the MaskROM button on the development board is as shown in the figure below:



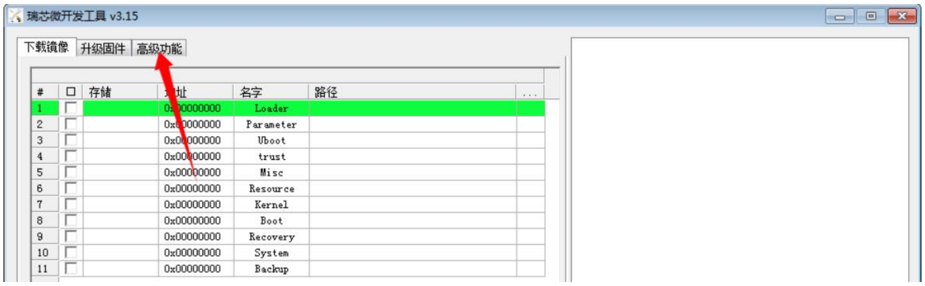
e. Then connect the power supply of the Type-C interface to the development board, power it on, and then release the MaskROM button.



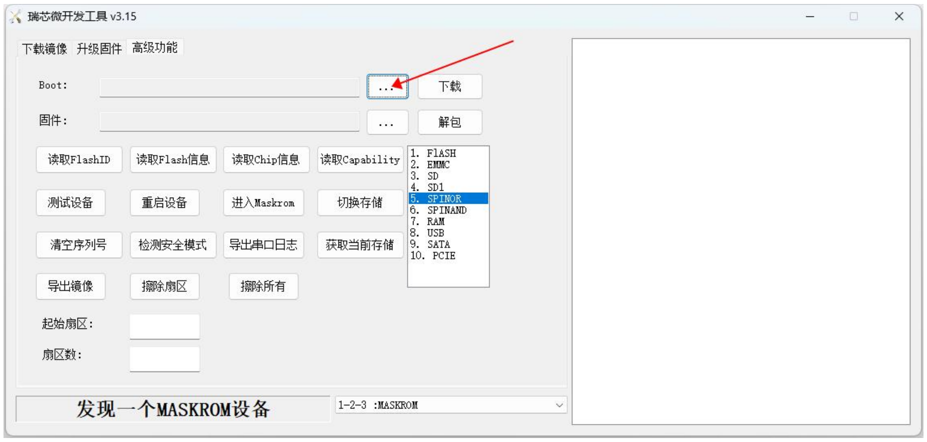
f. If the previous steps go well, the development board will enter **MASKROM** mode at this time, and the interface of the burning tool will prompt "**A MASKROM device was found**"



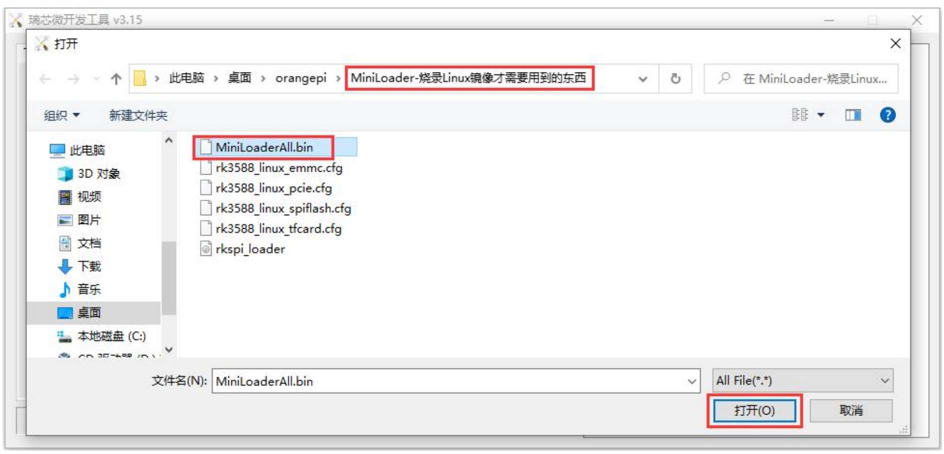
g. Then please select **advanced functions**



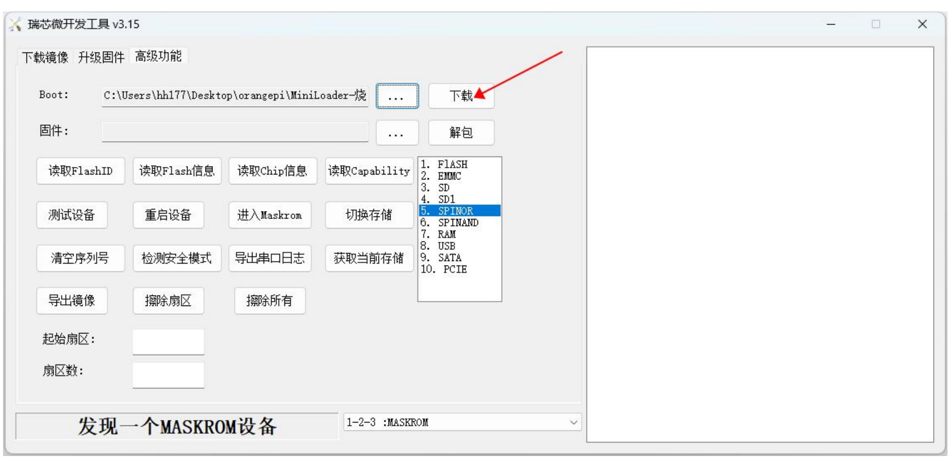
h. Then click the location shown in the picture below



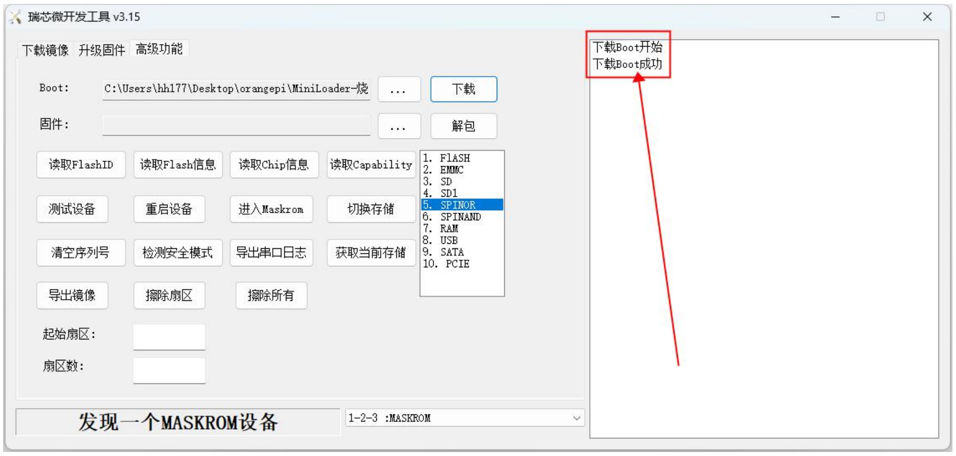
i. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click Open



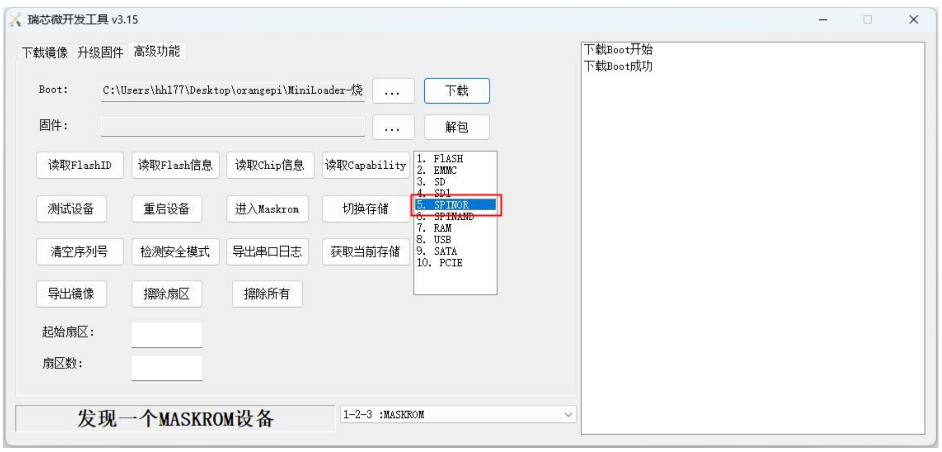
j. Then click **download**



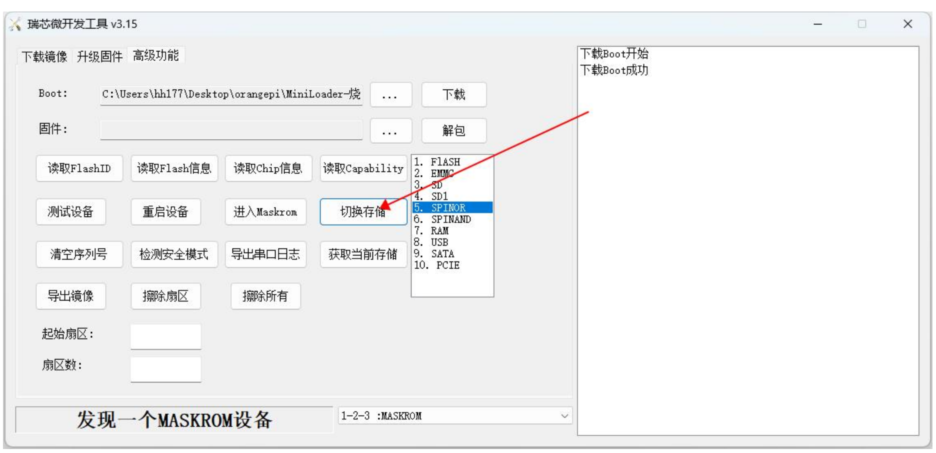
k. After downloading **MiniLoaderAll.bin**, the display is as shown below



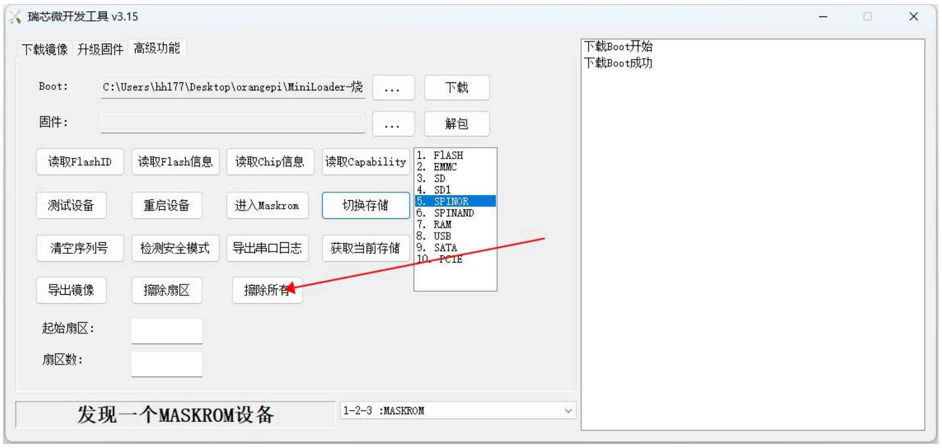
l. Then select the storage device as **SPINOR**



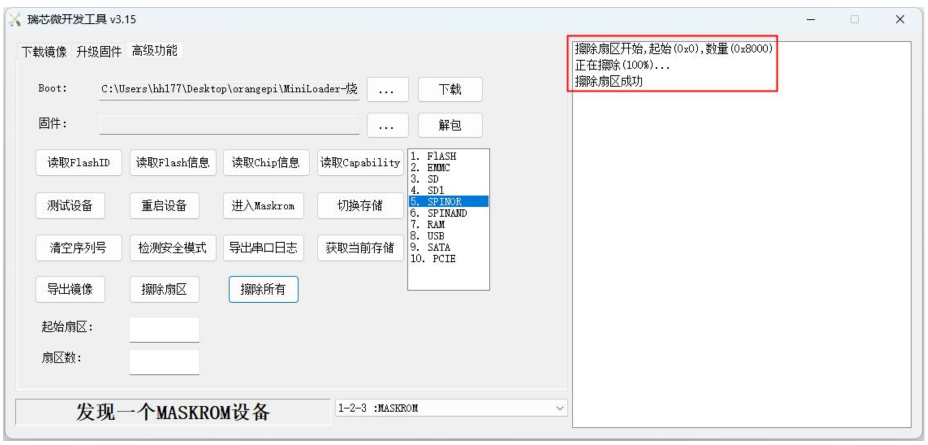
m. Then click **Switch Storage**



n. Then click **Erase All** to start erasing SPIFlash.



o. The display log after erasing SPIFlash is as shown below



2. 17. Start the Orange Pi development board

1) Insert the TF card with the burned image into the TF card slot of the Orange Pi development board. If the image of SPIFlash+NVMe SSD has been burnt, then there is no need to insert a TF card, just make sure that the NVMe SSD is inserted into the development board normally.

2) The development board has an HDMI interface, and the development board can be connected to a TV or HDMI display through an HDMI-to-HDMI cable. If you buy an LCD screen, you can also use the LCD screen to display the system interface of the development board. If there is a Type-C to HDMI cable, the system interface of the development board can also be displayed through the Type-C interface.

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

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

5) Connect a **high-quality** power adapter with a 5V/4A USB Type-C interface.

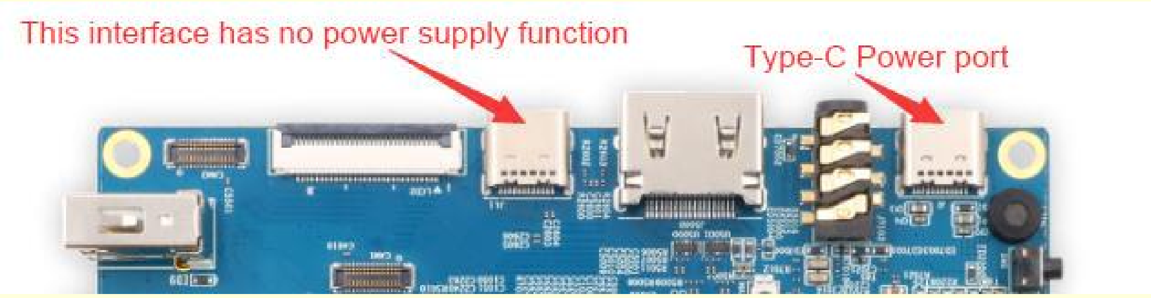
Remember not to plug in a power adapter with a voltage output greater than 5V, as this will burn out the development board.

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

The Type-C power port does not support PD negotiation.

In addition, please do not connect the USB interface of the computer to power the development board.

There are two Type-C ports that look the same on the development board. The one on the right is the power port, and the one in the middle has no power supply function. Please don't connect it wrong.



6) Then turn on the switch of the power adapter. If everything is normal, you can see the startup screen of the system on the HDMI monitor or LCD screen.

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

2. 18. How to use the debugging serial port

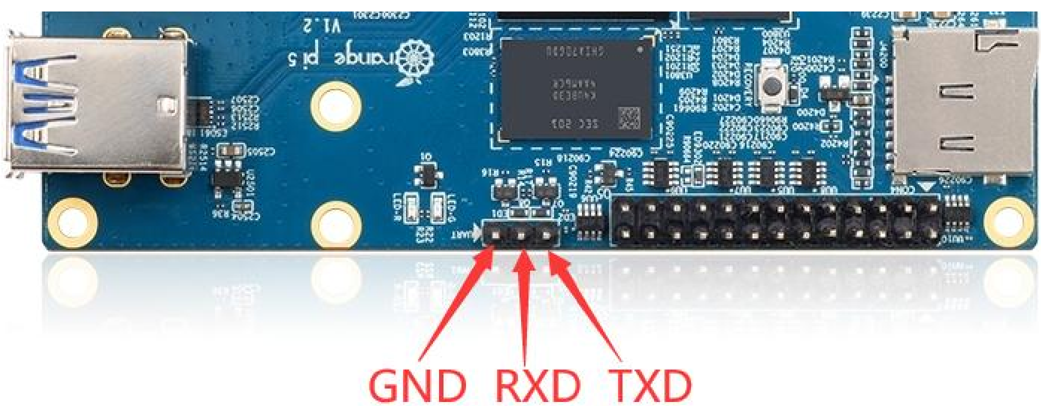
2. 18. 1. Connection instruction of debugging serial port

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

For better compatibility, it is recommended to use the CH340 USB to TTL module instead of the CP2102 USB to TTL module.

Before purchasing a USB to TTL module, please confirm that the module supports a baud rate of 1500000.

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



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

a. The GND of the USB to TTL module is connected to the GND of the development board

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

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

board

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

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

The TX and RX of the serial port need to be cross-connected. If you don't want to carefully distinguish the order of TX and RX, you can connect the TX and RX of the serial port casually. If there is no output in the test, then exchange the order of TX and RX, so that there is always a the order is right

2. 18. 2. How to use the debugging serial port on the Ubuntu platform

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

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

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

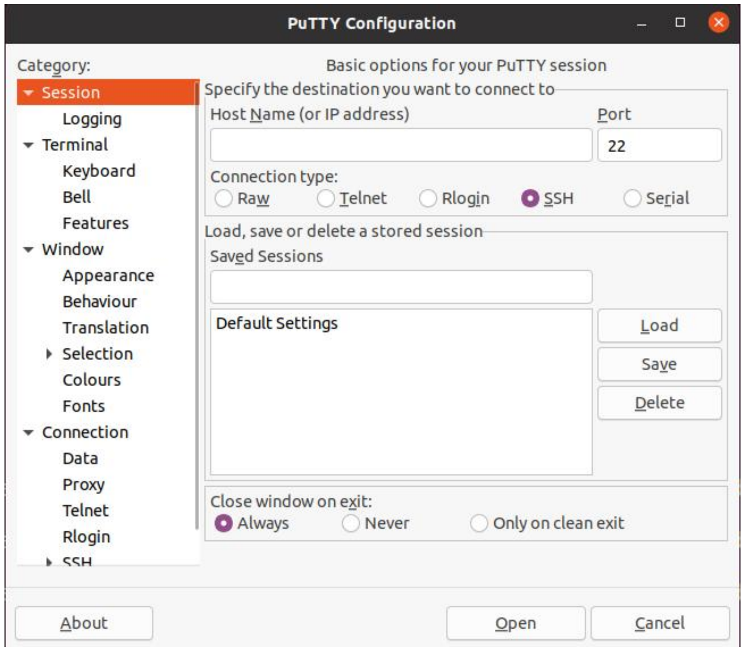
2) Then use the following command to install putty on Ubuntu PC

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

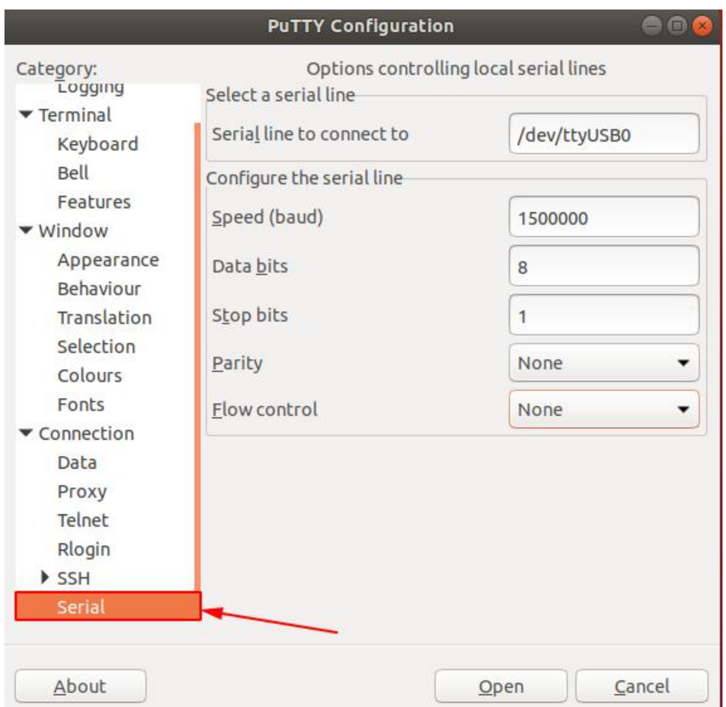
3) Then run putty, **remember to add sudo permission**

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

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



5) First select the setting interface of the serial port

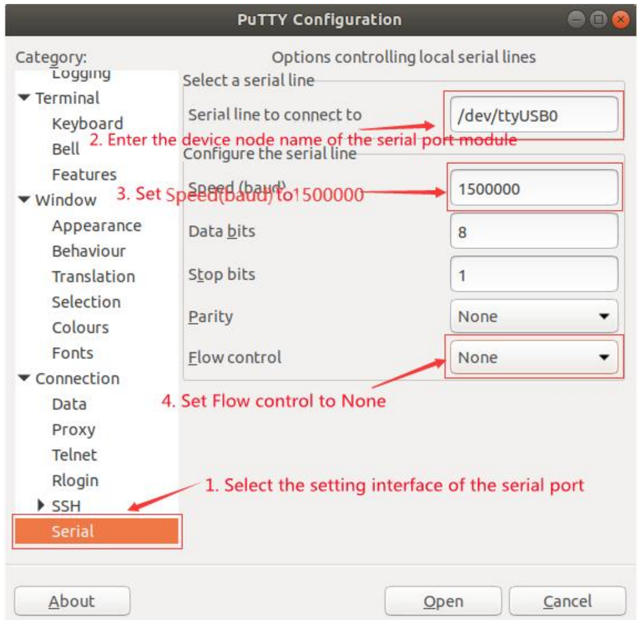


6) Then set the parameters of the serial port

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

b. Set Speed(baud) to **1500000** (the baud rate of the serial port)

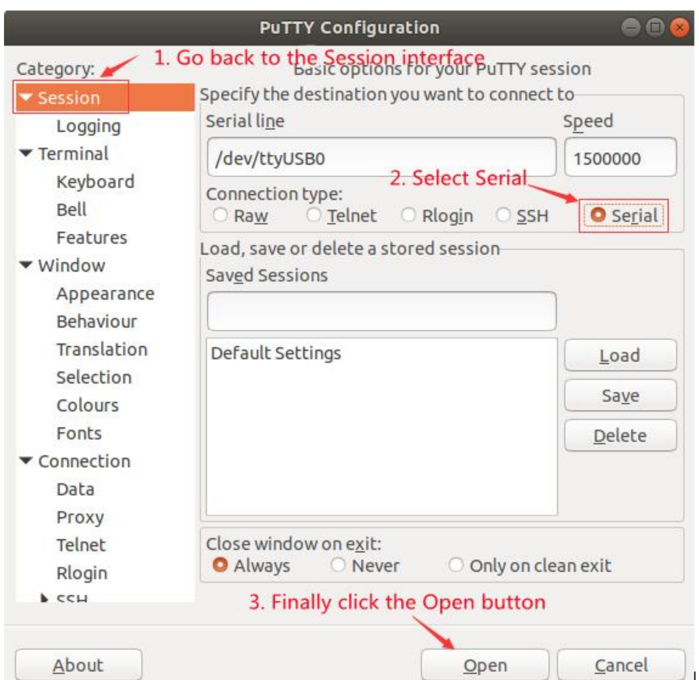
c. Set Flow control to None



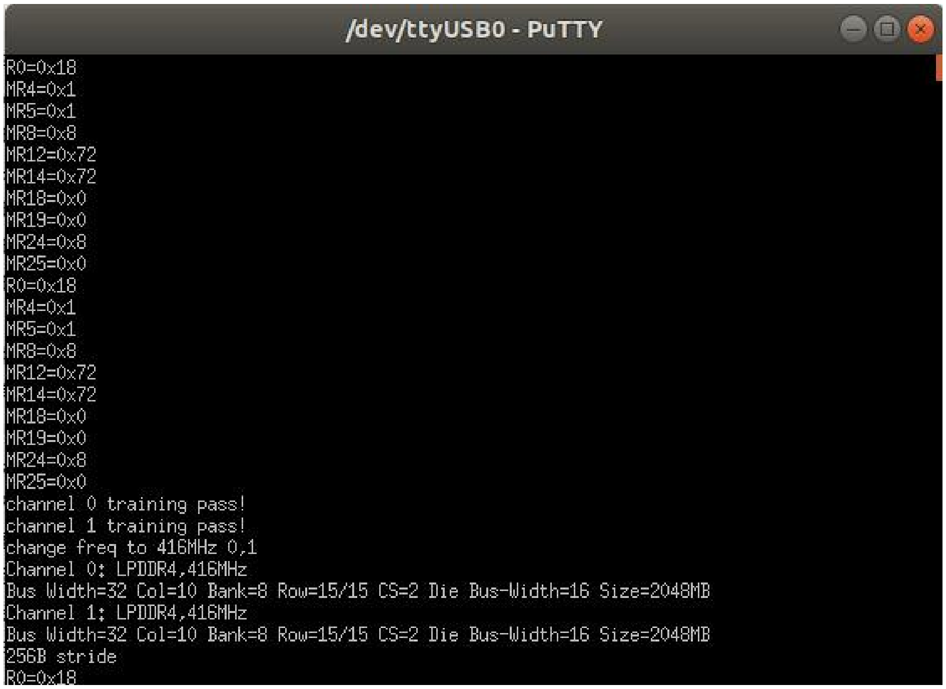
7) After setting the serial port setting interface, return to the Session interface

a. First select the Connection type as Serial

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



8) After starting the development board, you can see the Log information output by the system from the opened serial port terminal



2.18.3. How to use the debugging serial port on Windows platform

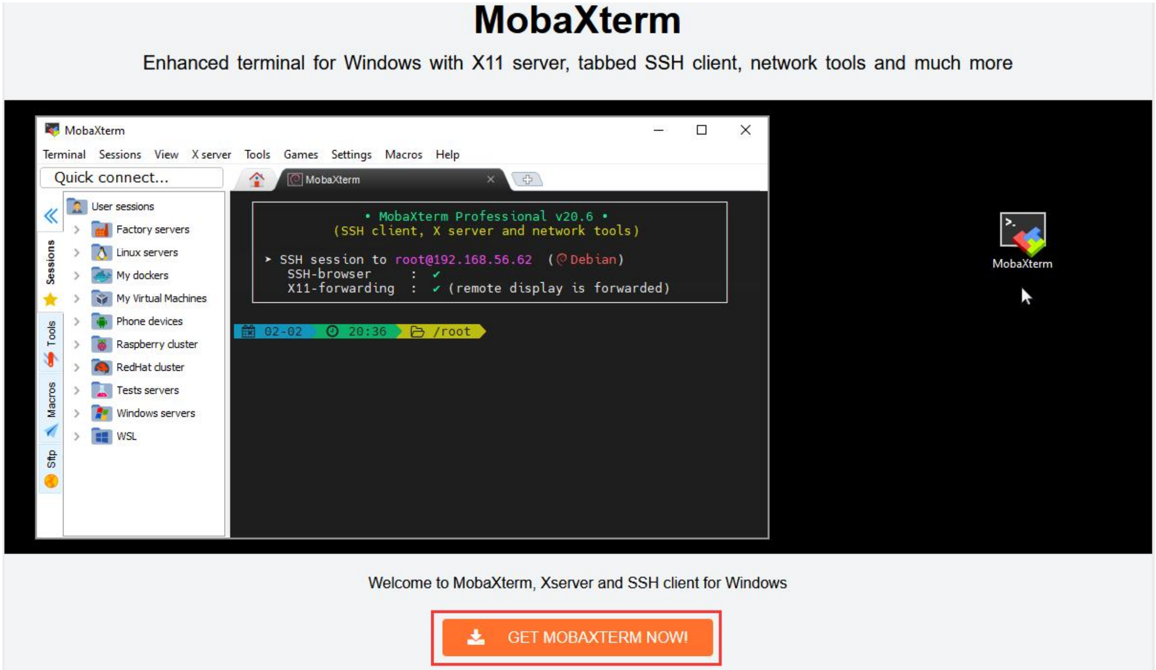
There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without buying a serial number.

1) Download MobaXterm

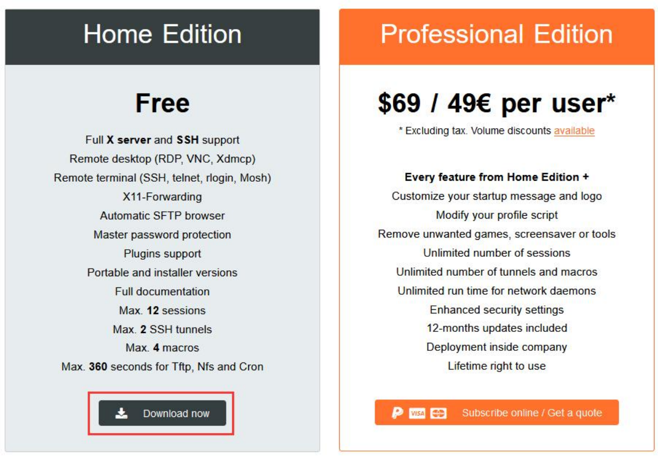
a. Download MobaXterm website as follows

https://mobaxterm.mobatek.net

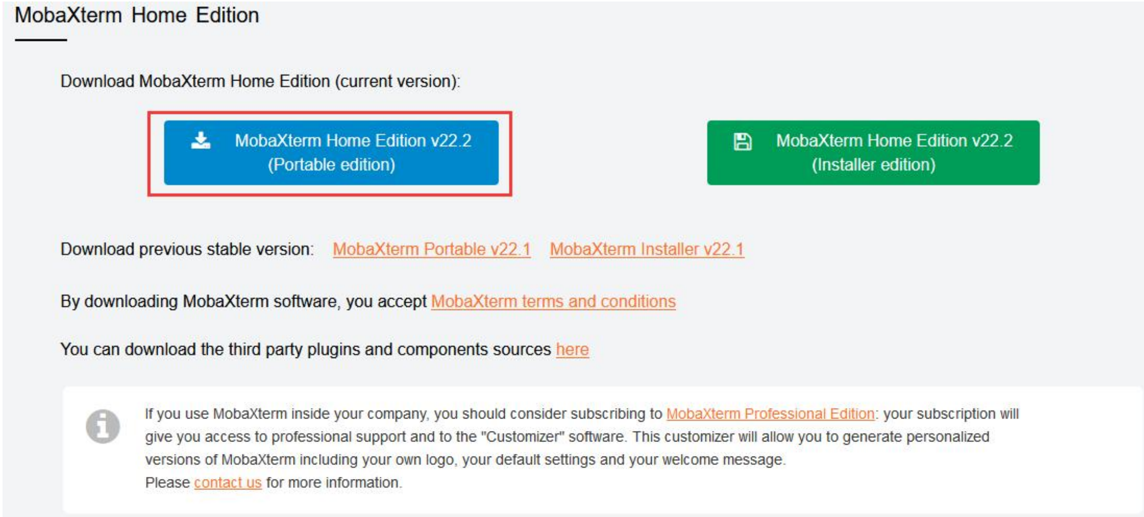
b. After entering the MobaXterm download page, click **GET XOBATERM NOW!**



c. Then choose to download the Home version



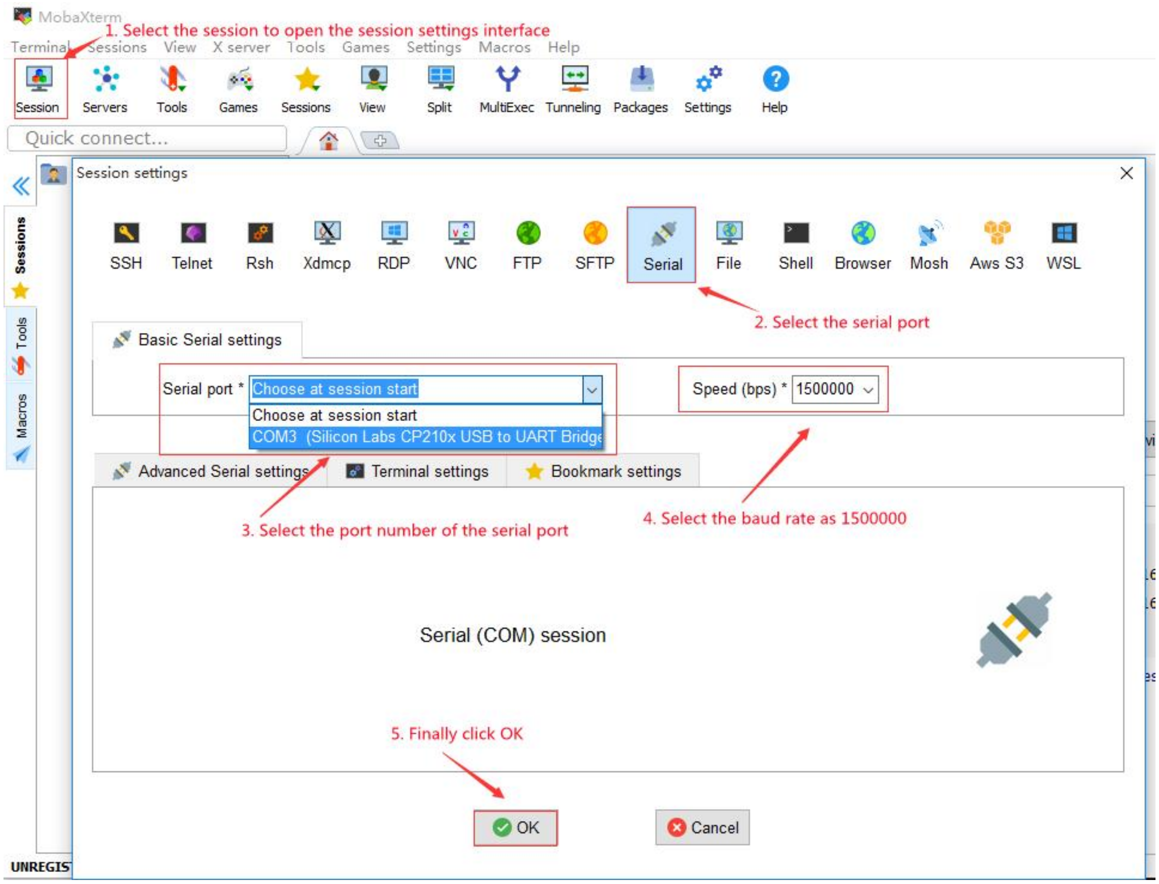
d. Then select Portable portable version, no need to install after downloading, just open it and use it



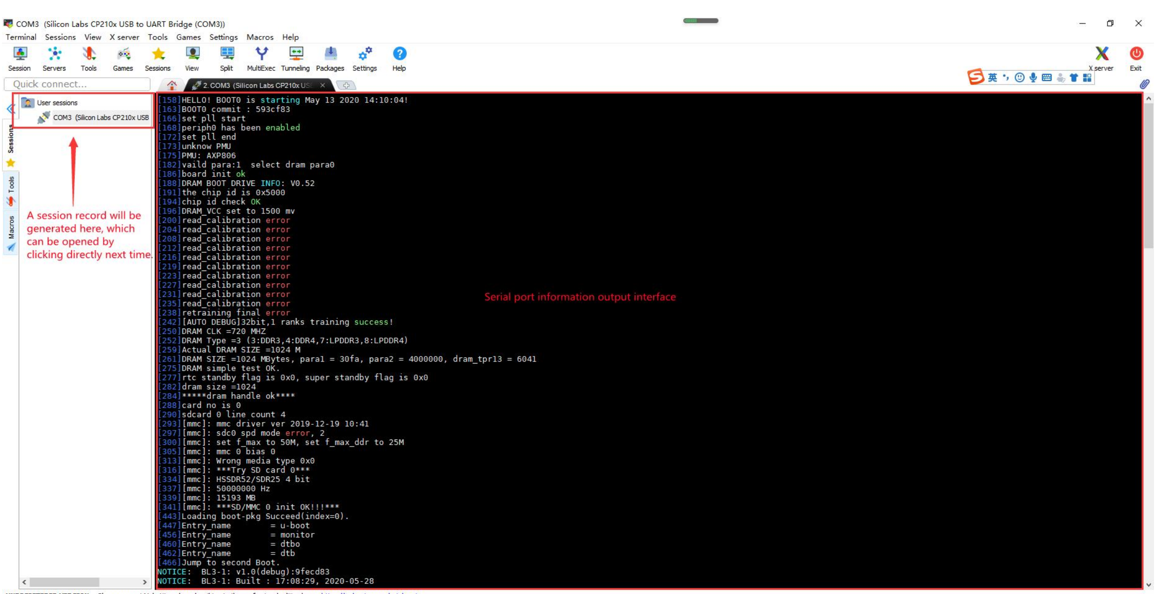
2) After downloading, use decompression software to decompress the downloaded compressed package, you can get the executable software of MobaXterm, and then double-click to open

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

a. Open the session settings interface
b. Select the serial port type
c. Select the port number of the serial port (select the corresponding port number according to the actual situation), if you cannot see the port number, please use **360 Driver Master** to scan and install the driver for the USB to TTL serial port chip
d. Select the baud rate of the serial port as **1500000**
e. Finally click the "**OK**" button to complete the setup



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



2.19. Instructions for using the 5v pin in the 26pin interface of the development board to supply power

The power supply method we recommend for the development board is to use the 5V/4A Type C interface power cord to plug into the Type-C power interface of the development board for power supply. If you need to use the 5V pin in the 26pin interface to power the development board, please make sure that the power cable and power adapter used can meet the power supply requirements of the development board. If the use is unstable, please switch back to the Type-C power supply.

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

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

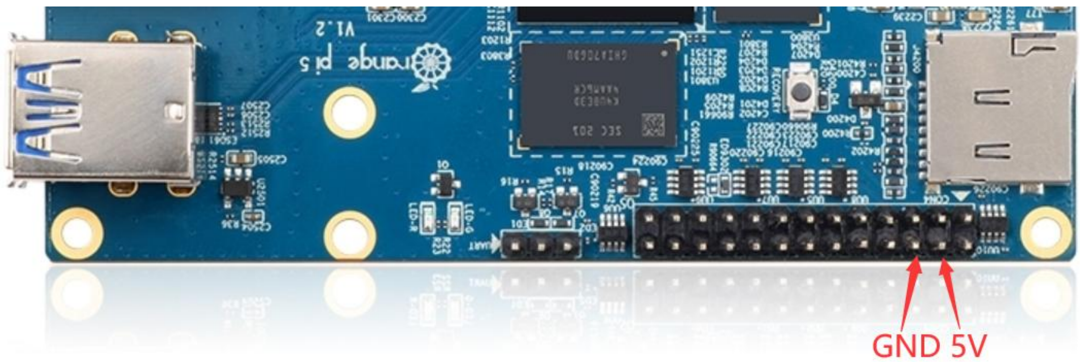
2) Use the 5V pin in the 26pin interface to supply power to the development board. The connection method of the power line is as follows

a. The USB A port of the power cord shown in the above picture needs to be plugged into the 5V/4A power adapter connector (**please do not plug into the USB port of the computer for power supply**)

b. The red DuPont line needs to be plugged into the 5V pin of the development board 26pin

c. The black DuPont line needs to be inserted into the GND pin of the 26pin interface

d. The position of the 5V pin and GND pin of the 26pin interface on the development board is shown in the figure below, **remember not to reverse the connection**



3. Ubuntu/Debian Server and Xfce desktop system usage instructions

本章内容是基于 linux 服务器版本的镜像和 xfce 桌面版本镜像编写的。

如果使用的是 Ubuntu22.04 Gnome 镜像，请先查看 **Ubuntu22.04 Gnome Wayland 桌面系统使用说明**一章的说明，

Ubuntu22.04 Gnome Wayland 桌面系统使用说明一章中不存在的内容，可以参考此章的说明，但是有些细节是会有差异的，这点请特别注意下。

如果使用的是 OPi OS Arch 镜像，请查看 **Orange Pi OS Arch 系统使用说明**一章的内容。

3. 1. Supported Linux image types and kernel versions

Linux Image Type	Kernel Version	Server Version	Desktop Version
Debian 11 - Bullseye	Linux5.10	Support	Support
Ubuntu 20.04 - Focal	Linux5.10	Support	Support
Ubuntu 22.04 - Jammy	Linux5.10	Support	Support
Debian 11 - Bullseye	Linux6.1	Support	Support
Debian 12 - Bookworm	Linux6.1	Support	Support

Ubuntu 20.04 - Focal	Linux6.1	Support	Support
Ubuntu 22.04 - Jammy	Linux6.1	Support	Support

3. 2. Linux system adaptation

3. 2. 1. Linux5.10 system adaptation situation

Function	Debian11	Debian12	Ubuntu20.04	Ubuntu22.04 xfce
USB2.0x2	OK	OK	OK	OK
USB3.0x1	OK	OK	OK	OK
USB Type-C 3.0	OK	OK	OK	OK
DP Display	OK	OK	OK	OK
M.2 NVMe SSD Boot	OK	OK	OK	OK
M.2 SATA SSD Boot	OK	OK	OK	OK
USB Boot System	OK	OK	OK	OK
AP6275P-WIFI	OK	OK	OK	OK
AP6275P-Bluetooth	OK	OK	OK	OK
GPIO（26pin）	OK	OK	OK	OK
UART（26pin）	OK	OK	OK	OK
SPI（26pin）	OK	OK	OK	OK
I2C（26pin）	OK	OK	OK	OK
CAN（26pin）	OK	OK	OK	OK
PWM（26pin）	OK	OK	OK	OK
3pin Debugging Serial Port	OK	OK	OK	OK
TF Card Start	OK	OK	OK	OK
HDMI Video	OK	OK	OK	OK
HDMI Audio	OK	OK	OK	OK
OV13850 Camera	OK	OK	OK	OK
OV13855 Camera	OK	OK	OK	OK
LCD1	OK	OK	OK	OK
LCD2	OK	OK	OK	OK
Gigabit Ethernet Port	OK	OK	OK	OK
Network Port Status	OK	OK	OK	OK

Light				
MIC	OK	OK	OK	OK
Headphone Playback	OK	OK	OK	OK
Headphone Recording	OK	OK	OK	OK
LED Lights	OK	OK	OK	OK
GPU	OK	OK	OK	OK
NPU	OK	OK	OK	OK
VPU	OK	OK	OK	OK
Switch Button	OK	OK	OK	OK
Watchdog Test	OK	OK	OK	OK
Chromium Hard Solution Video	OK	OK	OK	OK

3. 2. 2. Linux6.1 system adaptation situation

Function	Debian11	Debian12	Ubuntu20.04	Ubuntu22.04
USB2.0x2	OK	OK	OK	OK
USB3.0x1	OK	OK	OK	OK
USB Type-C 3.0	OK	OK	OK	OK
DP Display	OK	OK	OK	OK
M.2 NVMe SSD Boot	OK	OK	OK	OK
M.2 SATA SSD Boot	OK	OK	OK	OK
USB Boot System	OK	OK	OK	OK
AP6275P-WIFI	OK	OK	OK	OK
AP6275P-Bluetooth	OK	OK	OK	OK
GPIO（26pin）	OK	OK	OK	OK
UART（26pin）	OK	OK	OK	OK
SPI（26pin）	OK	OK	OK	OK
I2C（26pin）	OK	OK	OK	OK
CAN（26pin）	NO	NO	NO	NO
PWM（26pin）	OK	OK	OK	OK
3pin Debugging Serial Port	OK	OK	OK	OK
TF Card Start	OK	OK	OK	OK
HDMI Video	OK	OK	OK	OK
HDMI Audio	OK	OK	OK	OK

OV13850 Camera	**OK**	**OK**	**OK**	**OK**
OV13855 Camera	**OK**	**OK**	**OK**	**OK**
LCD1	**OK**	**OK**	**OK**	**OK**
LCD2	**OK**	**OK**	**OK**	**OK**
Gigabit Ethernet Port	**OK**	**OK**	**OK**	**OK**
Network Port Status Light	**OK**	**OK**	**OK**	**OK**
MIC	**OK**	**OK**	**OK**	**OK**
Headphone Playback	**OK**	**OK**	**OK**	**OK**
Headphone Recording	**OK**	**OK**	**OK**	**OK**
LED Lights	**OK**	**OK**	**OK**	**OK**
GPU	**OK**	**OK**	**OK**	**OK**
NPU	**OK**	**OK**	**OK**	**OK**
VPU	**OK**	**OK**	**OK**	**OK**
Switch Button	**OK**	**OK**	**OK**	**OK**
Watchdog Test	**OK**	**OK**	**OK**	**OK**
Chromium Hard Solution Video	**OK**	**OK**	**OK**	**OK**

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

1) All commands that need to be entered in the Linux system in this manual will be boxed with the following box

```
```

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

```
```

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

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

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

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

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

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

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

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

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

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

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

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

3. 4. Linux system login instructions

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

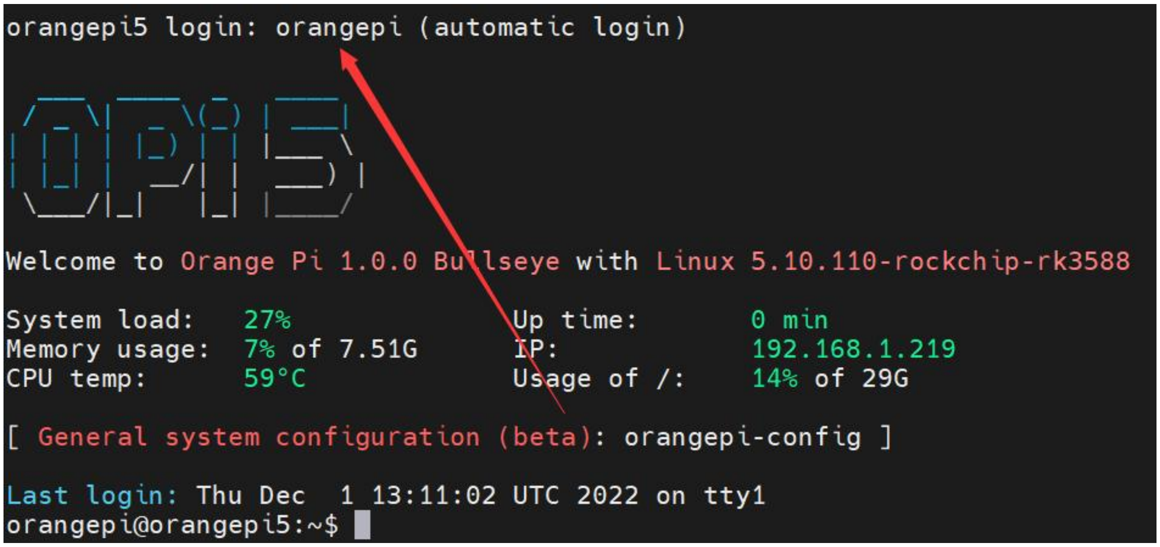
Account	Password
root	orangepi
orangepi	orangepi

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

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

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

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



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

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

3) Use the following command to disable automatic login terminal

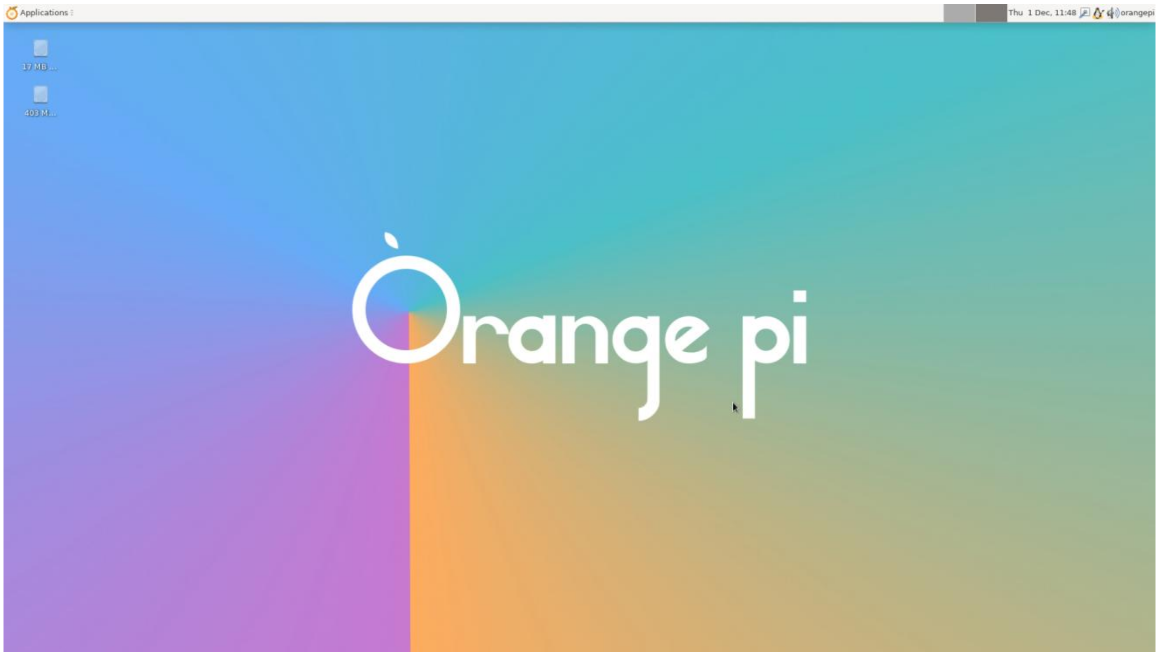
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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

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

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

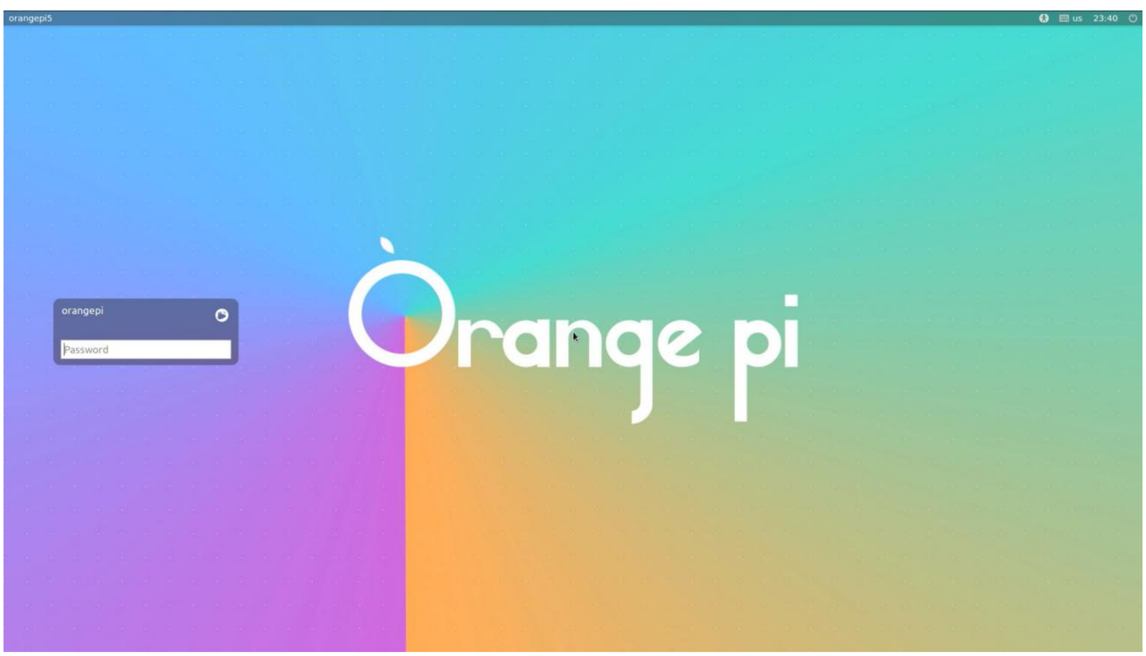
1) After the desktop version system is started, it will automatically log in to the desktop without entering a password



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

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

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



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

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

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

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

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

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

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

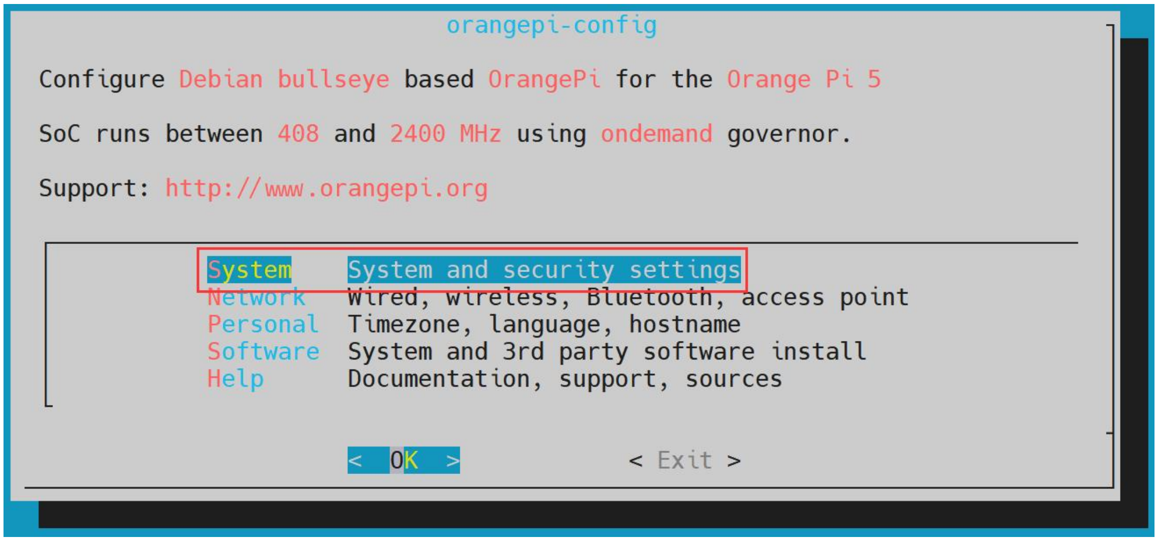
```
orangepi@orangepi:~$ sudo desktop_login.sh orangepi
```

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

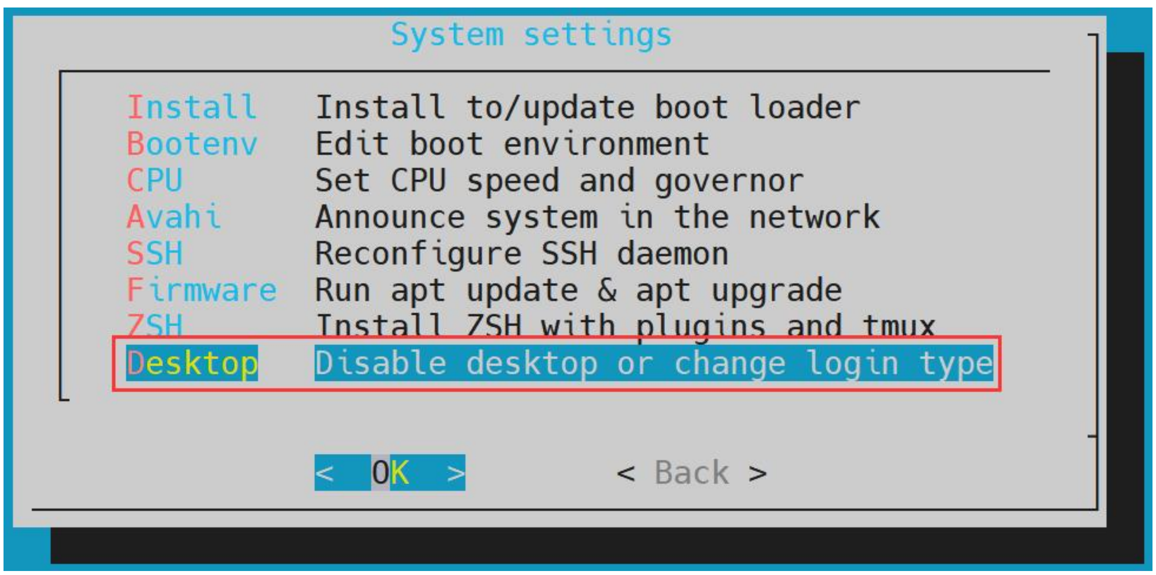
1) First enter the following command in the command line, **please remember to add sudo permission**

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

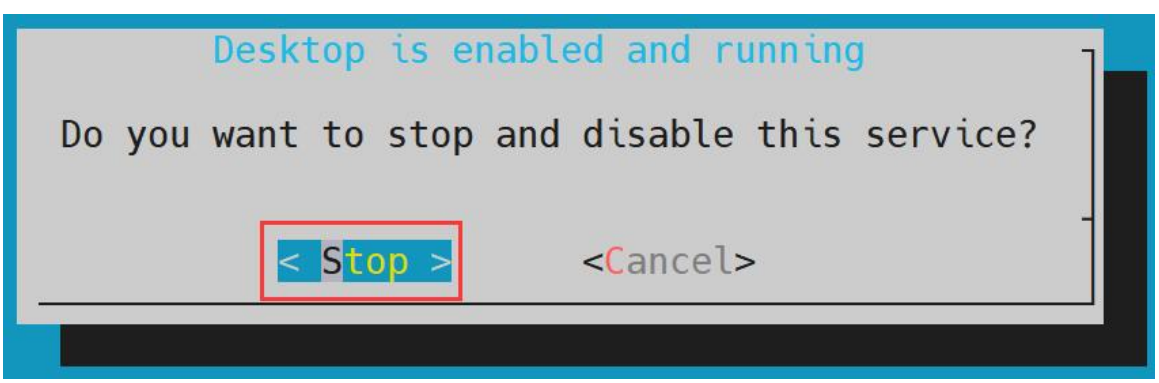
2) Then select **System**



3) Then select **Desktop**



4) Then select **<Stop>**



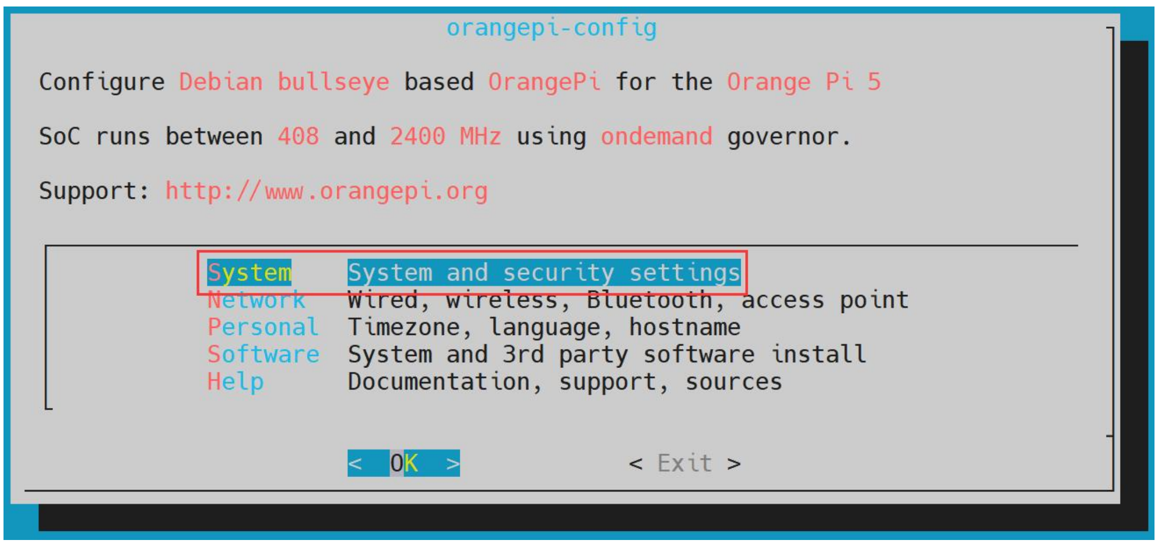
5) **Then restart the Linux system and you will find that the desktop will not be displayed**

6) The steps to reopen the desktop are as follows:

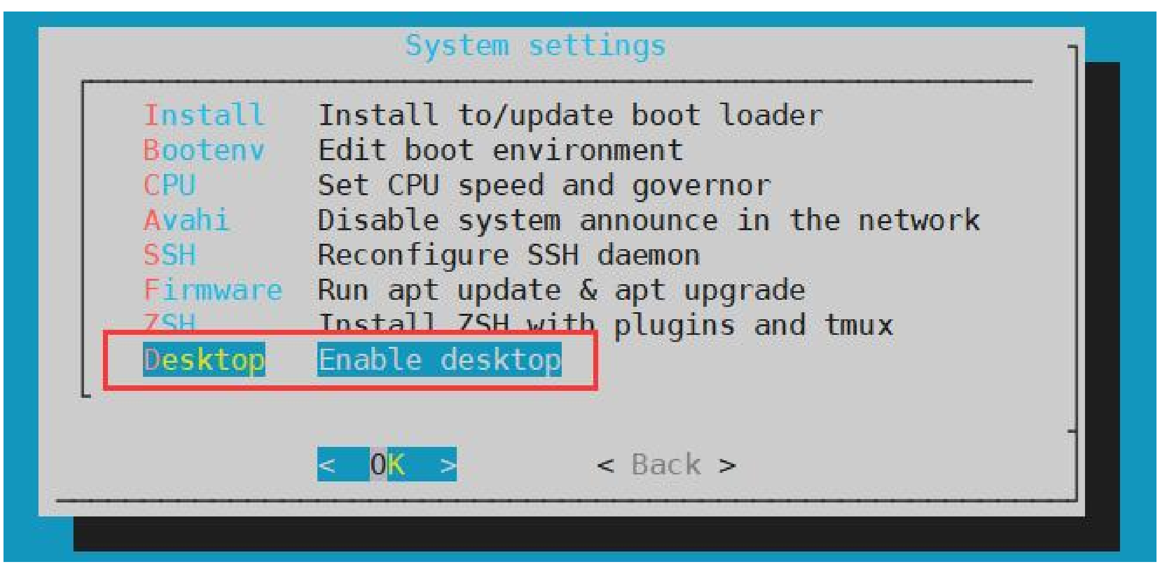
a. First enter the following command on the command line, **please remember to add sudo permission**

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

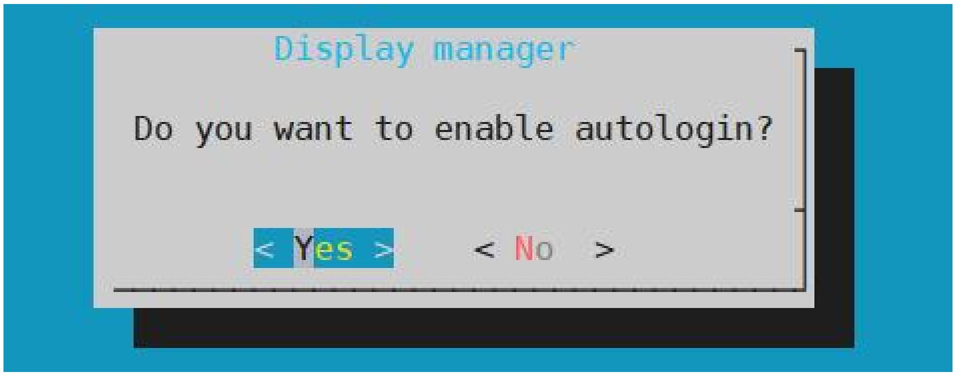
b. Then select **System**



c. Then select **Desktop Enable desktop**



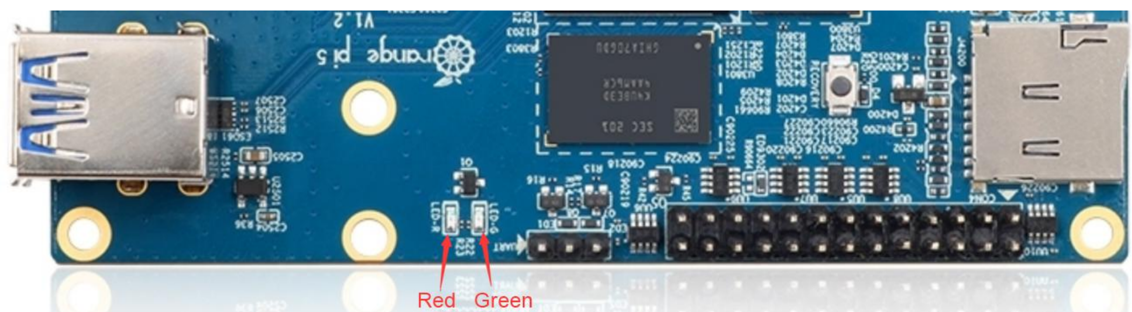
d. Then choose whether to automatically log in to the desktop, if you select **<Yes>**, it will automatically log in to the desktop, if you select **<No>**, it will display the user and password input interface, and you need to enter the password to enter the desktop



e. After selection, the HDMI monitor will display the desktop

3. 5. Onboard LED Light Test Instructions

1) There are two LED lights on the development board, one is green and the other is red. The location is shown in the figure below:



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

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

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

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

a. First enter the setting directory of the green light

```
root@orangepi:~# cd /sys/class/leds/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 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. Network Connection Test

3. 6. 1. Ethernet port test

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

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

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

```
orangepi@orangepi:~$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 4a:fe:2b:3d:17:1c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.150/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
       valid_lft 43150sec preferred_lft 43150sec
    inet6 fe80::9a04:3703:faed:23be/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

When using ifconfig to view the IP address, if the following information is prompted, it is because sudo is not added. The correct command is: sudo ifconfig

```
orangepi@orangepi:~$ ifconfig
Command 'ifconfig' is available in the following places
 * /sbin/ifconfig
 * /usr/sbin/ifconfig
The command could not be located because '/sbin:/usr/sbin' is not included in the PATH
```

environment variable.
This is most likely caused by the lack of administrative privileges associated with your user account.
ifconfig: command not found

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

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

2. Enter the ip addr show eth0 command in the debugging serial terminal to view the IP address

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

A）First check whether the Linux system has started normally. If the green light of the development board is blinking, it is generally started normally. If only the red 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 (I have encountered many problems with the router, such as the router cannot assign the IP address normally, or the IP address has been assigned normally but cannot be seen in the router);

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

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

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

```
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
```

3.6.2. WIFI connection test

First of all, please note that there is no WIFI module on the Orange Pi 5 development board, and an external PCIe network card or USB network card is required to use the WIFI function.

For instructions on using the external PCIe network card, please refer to the section on how to use the AP6275P PCIe network card.

For instructions on using the external USB network card, please refer to the USB wireless network card test section.

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

3.6.2.1. The server version image connects to WIFI through commands

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

1) First log in to the linux system, there are the following three ways
   a. If the development board is connected with a network cable, you can **remotely log in to the Linux system through ssh**

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

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

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

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

```
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:~#
```

3) 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'.
```

4) Through the **ip addr show wlan0** command, you can view the IP address of wifi

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

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

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

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

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

1) First log in to the linux system, there are the following three ways

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

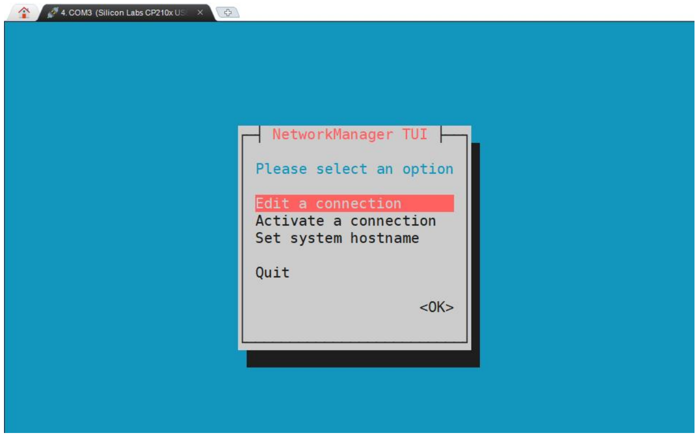
b. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the linux system (please use MobaXterm for the serial port software, and the graphical interface cannot be displayed when using minicom)

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

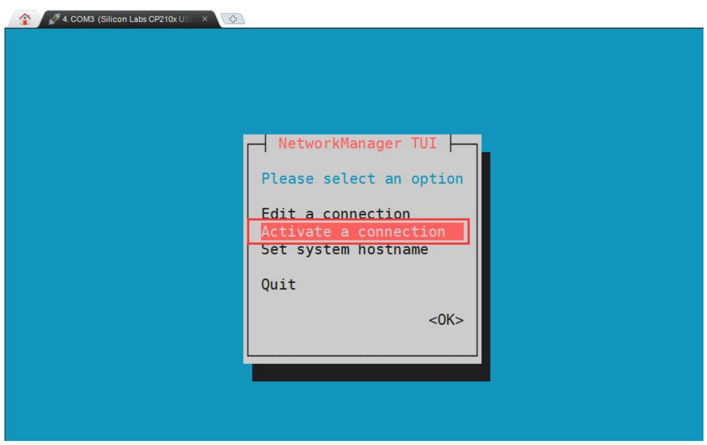
2) Then enter the nmtui command in the command line to open the wifi connection interface

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

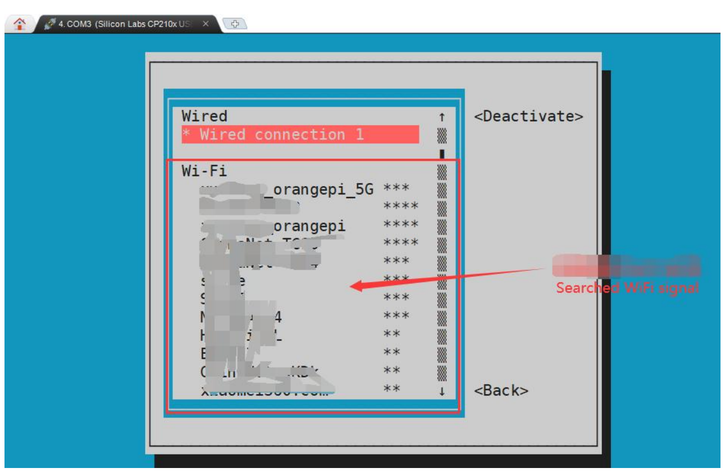
3) Enter the nmtui command to open the interface as shown below



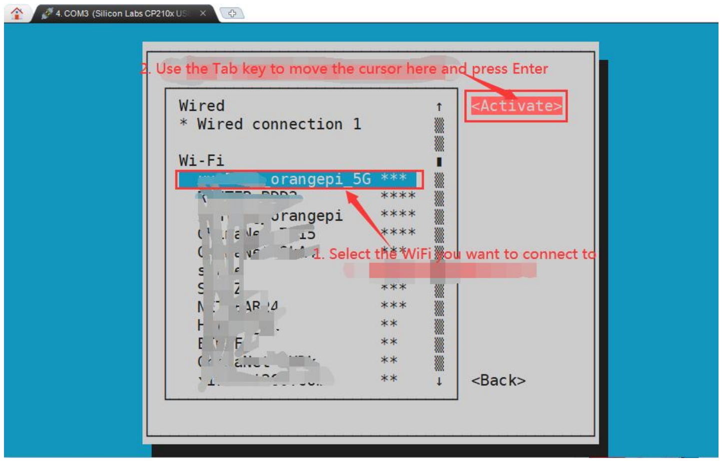
4) Select **Activate a connect** and press Enter



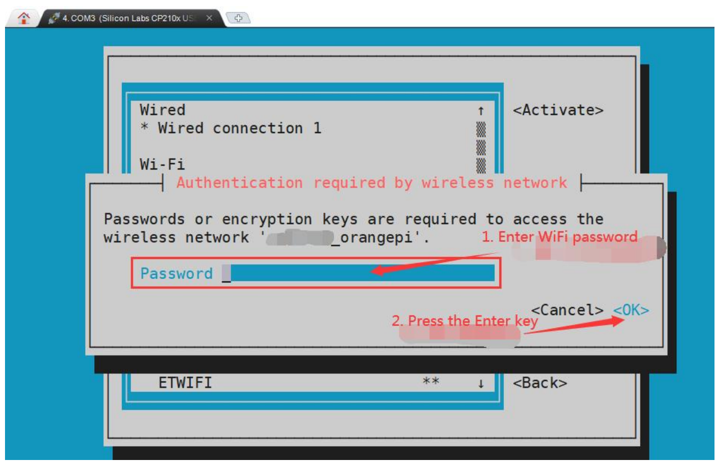
5) Then you can see all the searched WIFI hotspots



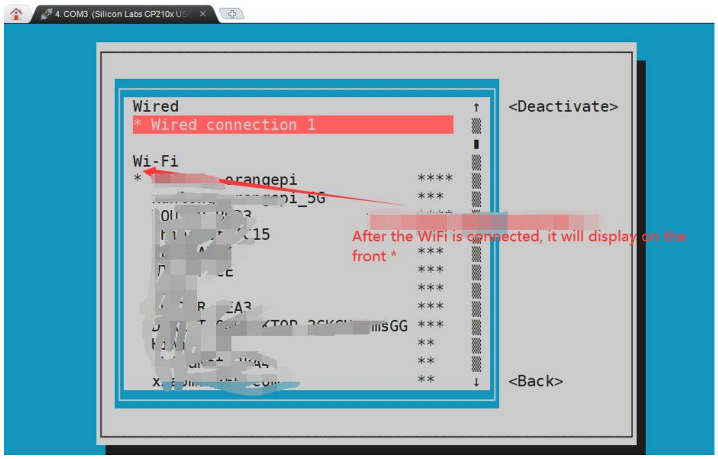
6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the cursor on **Activate** and press Enter



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



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



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

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

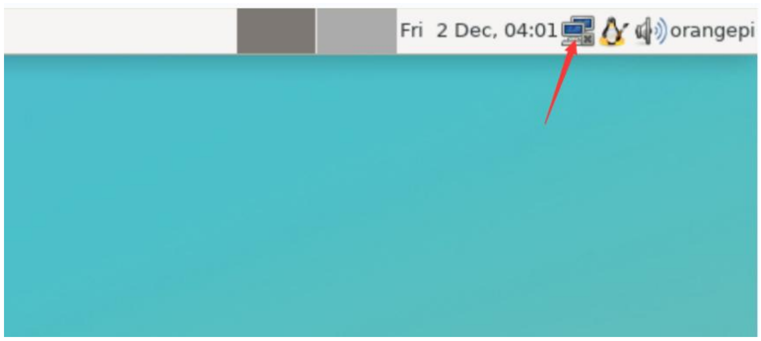
10) Use the **ping** command to test the connectivity of the wifi network, and the **ping** command can be interrupted through the shortcut key **Ctrl+C**

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

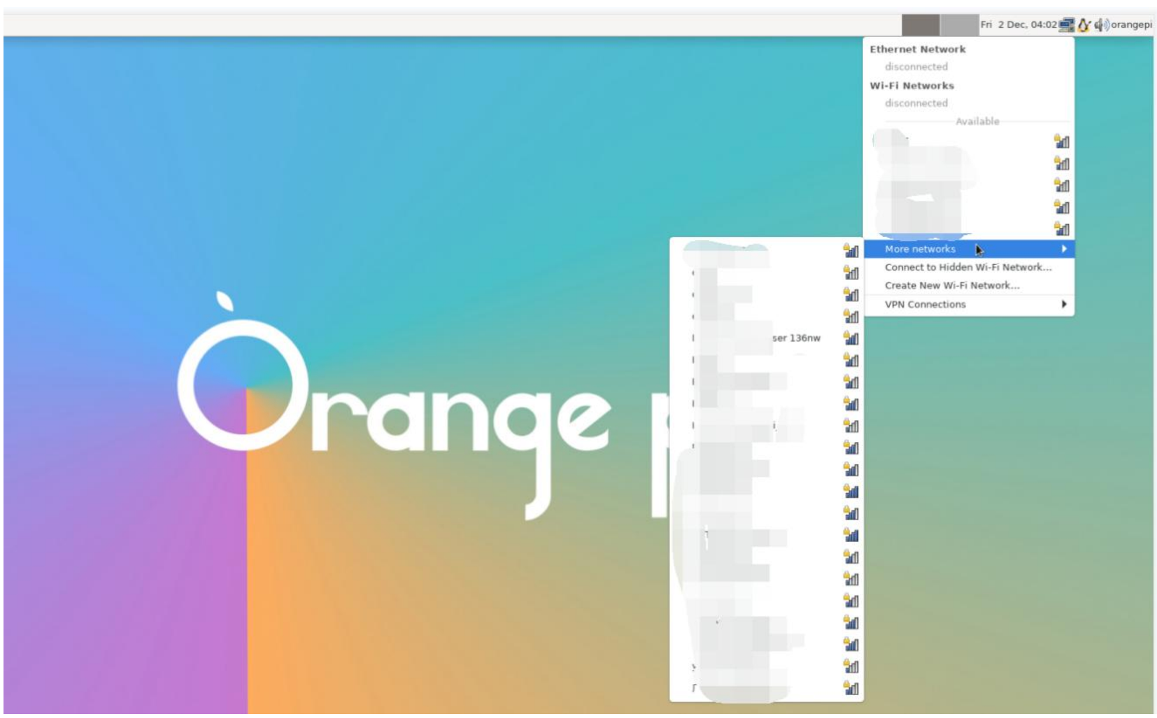
```
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

3. 6. 2. 3. Test method of desktop image

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

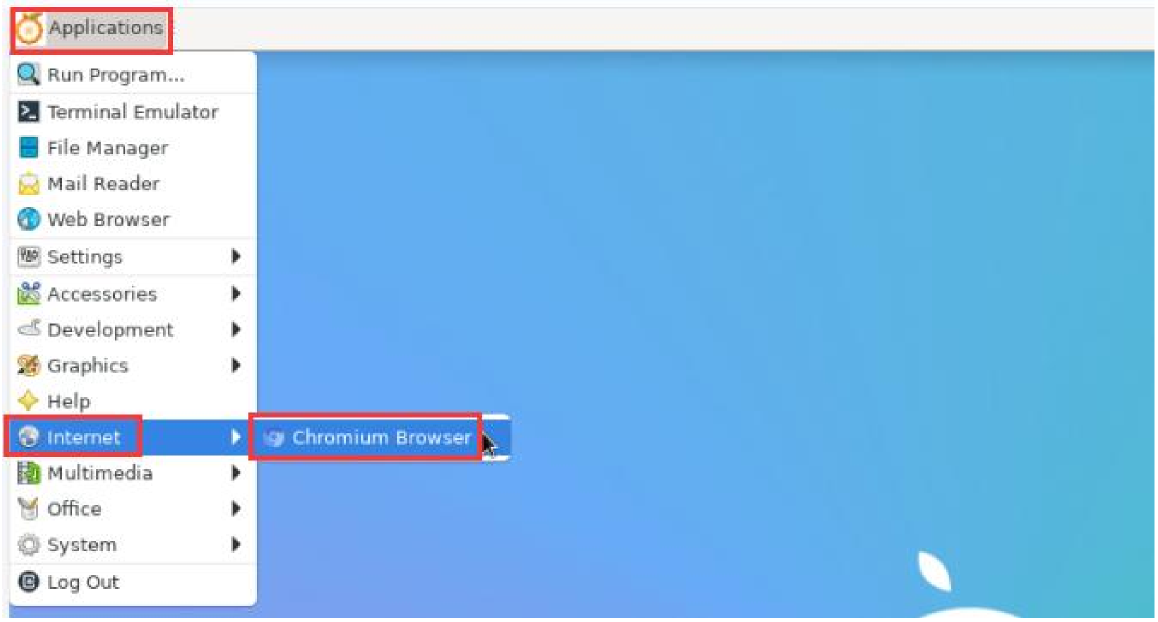


2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to

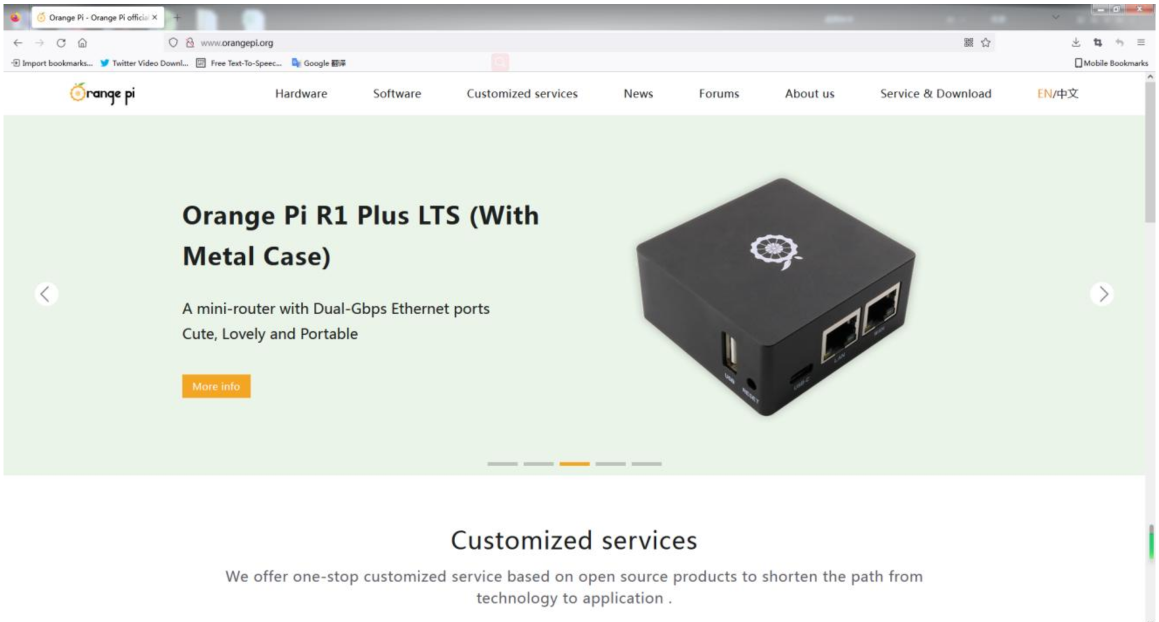


3) Then enter the password of the WIFI hotspot, and then click **Connect** to start connecting to WIFI

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



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



3. 6. 3. How to set a static IP address

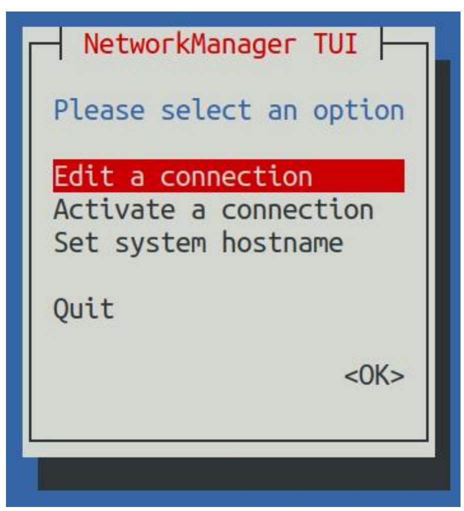
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file.

3. 6. 3. 1. Use the nmtui command to set a static IP address

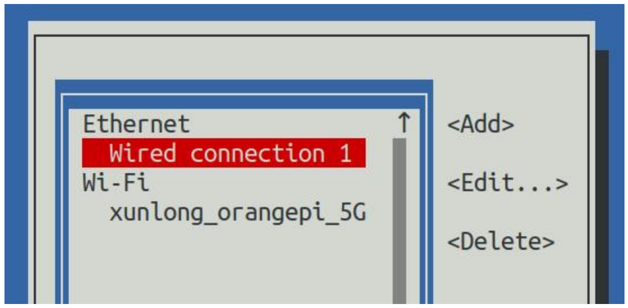
1) First run the **nmtui** command

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

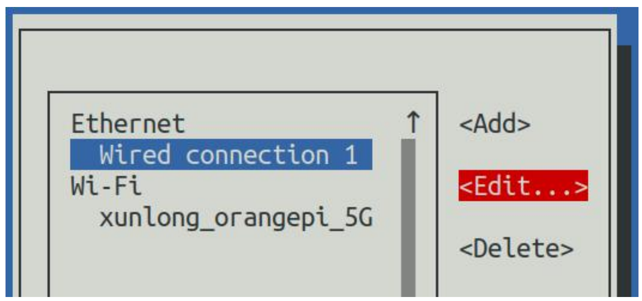
2) Then select **Edit a connection** and press Enter



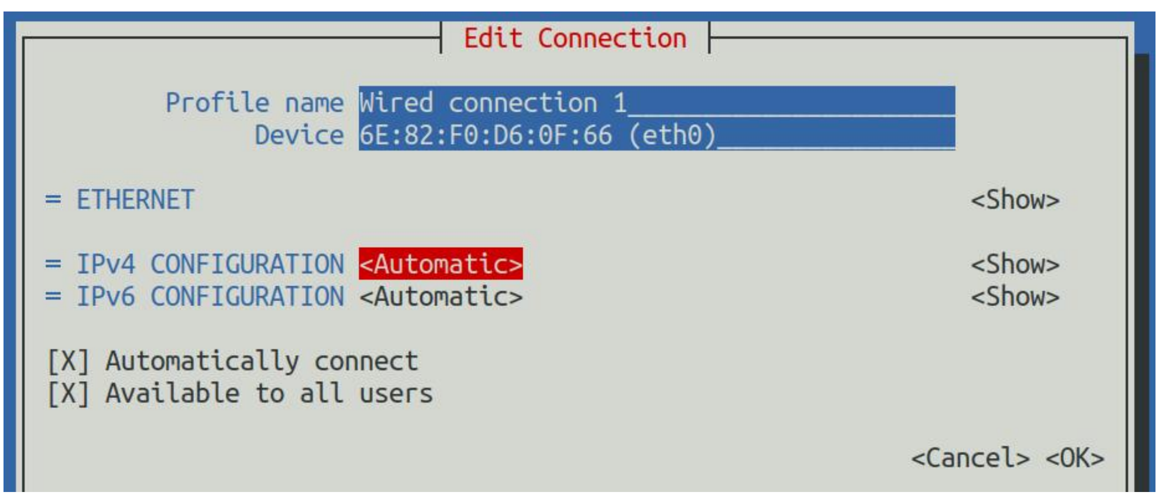
3) Then select the network interface that needs to set a static IP address, for example, to set the static IP address of the **Ethernet** interface, select **Wired connection 1**.



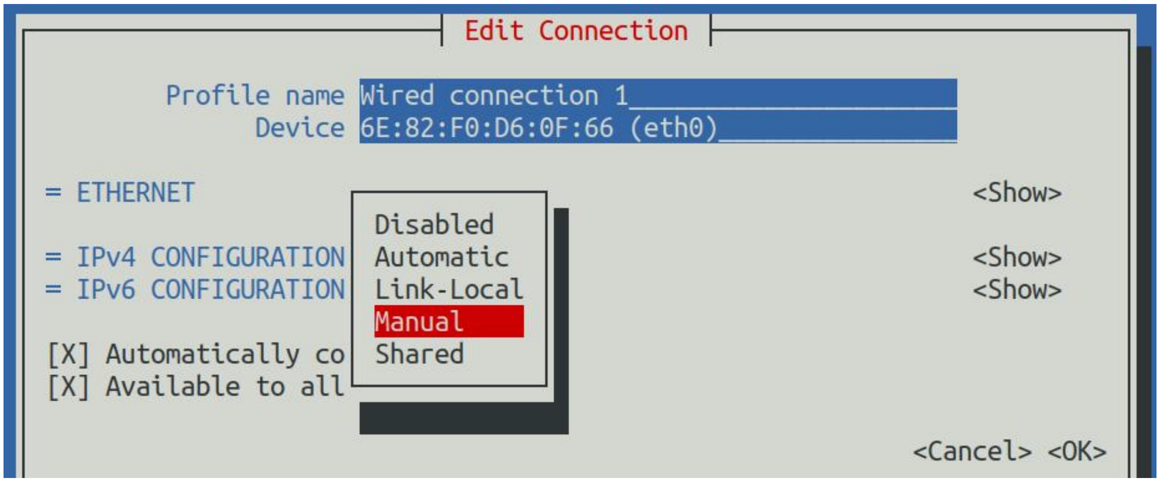
4) Then select **Edit** with the **Tab** key and press Enter



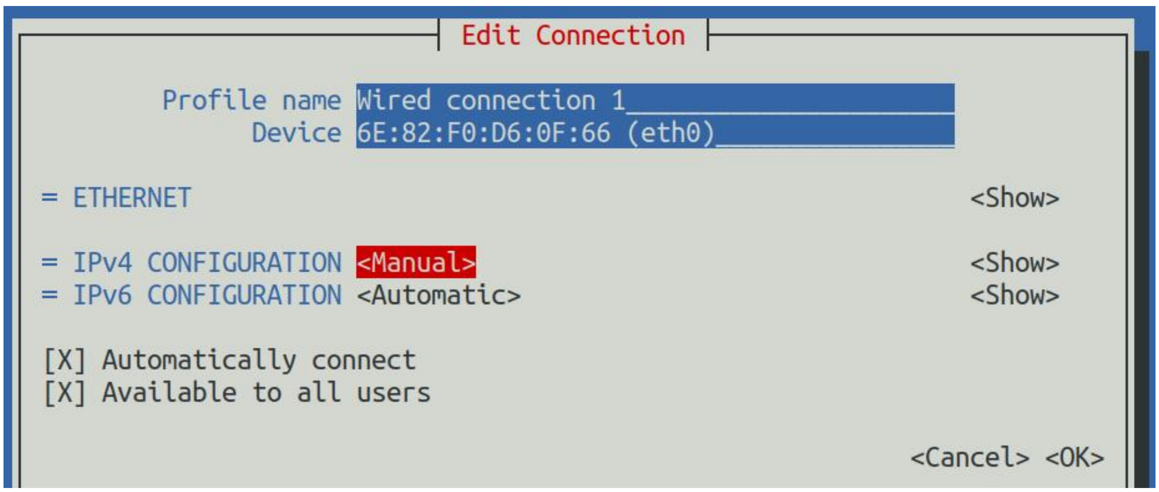
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4



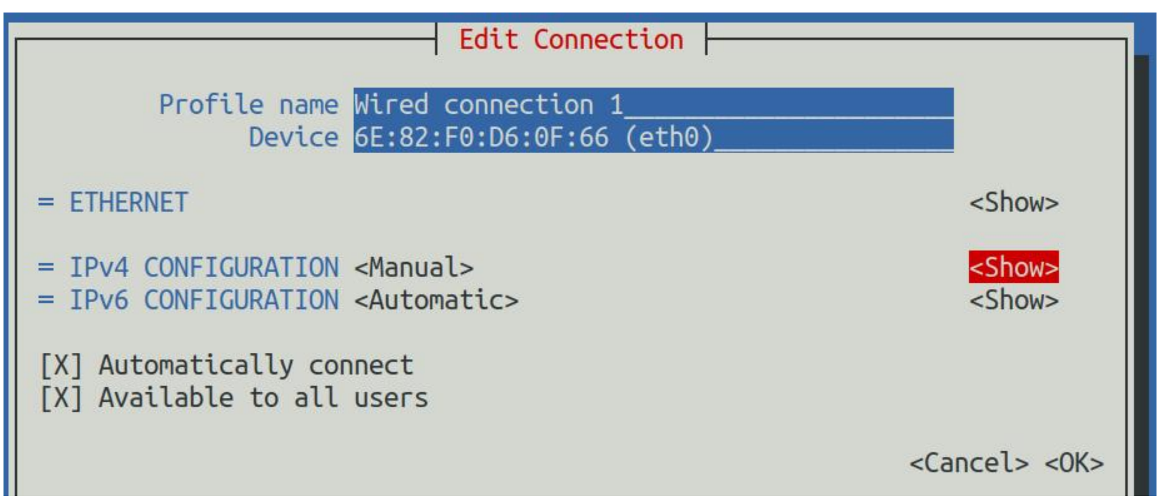
6) Then press Enter, select **Manual** through the up and down arrow keys, and press Enter to confirm



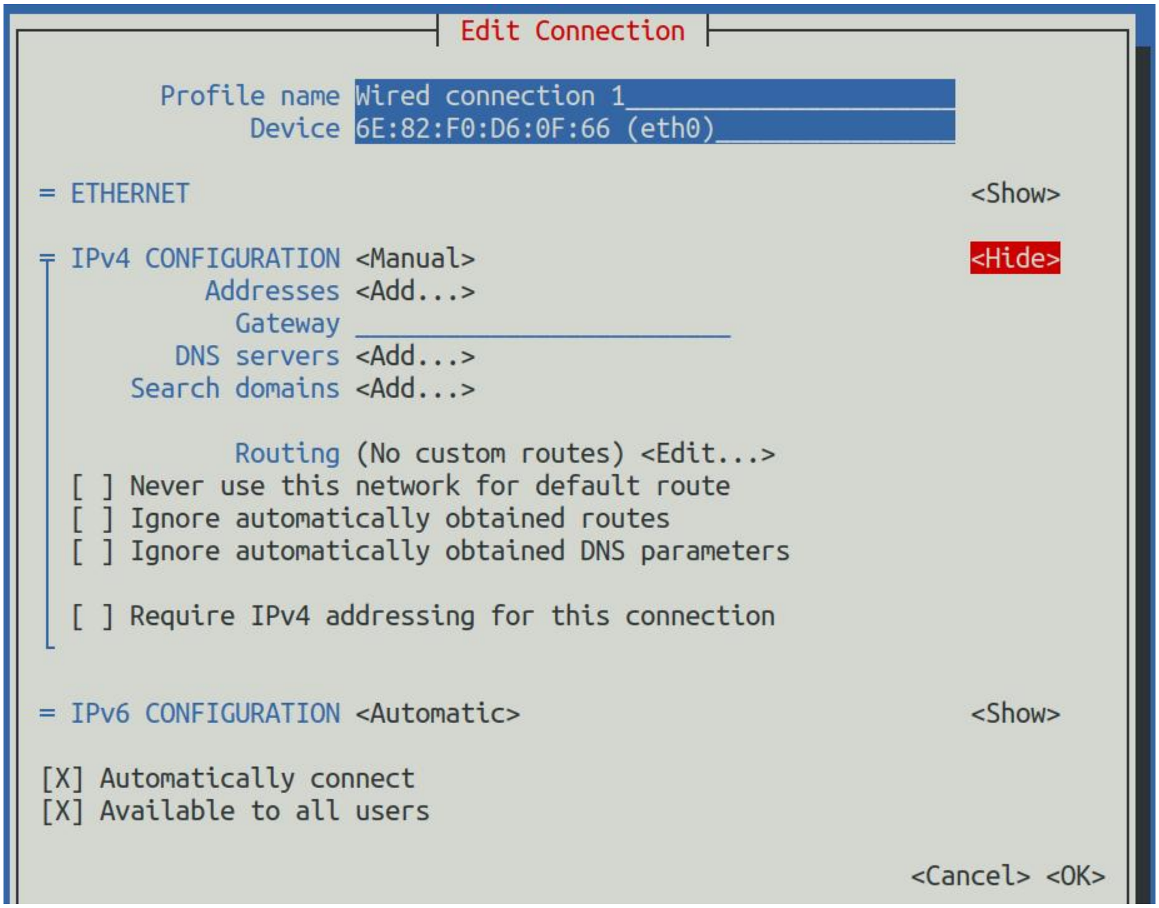
7) The display after selection is shown in the figure below



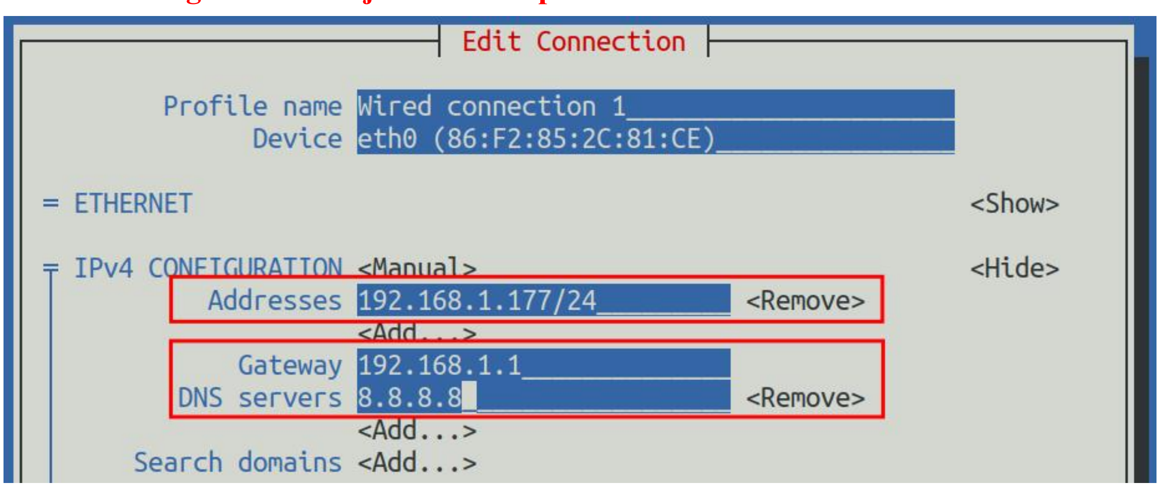
8) Then move the cursor to **<Show>** via the Tab key



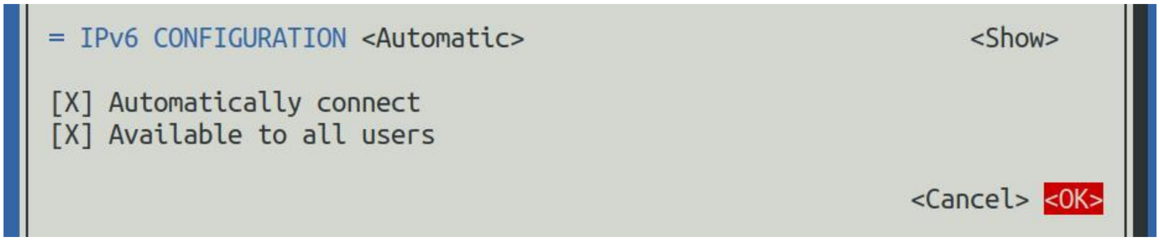
9) Then press Enter, and the following setting interface will pop up after entering



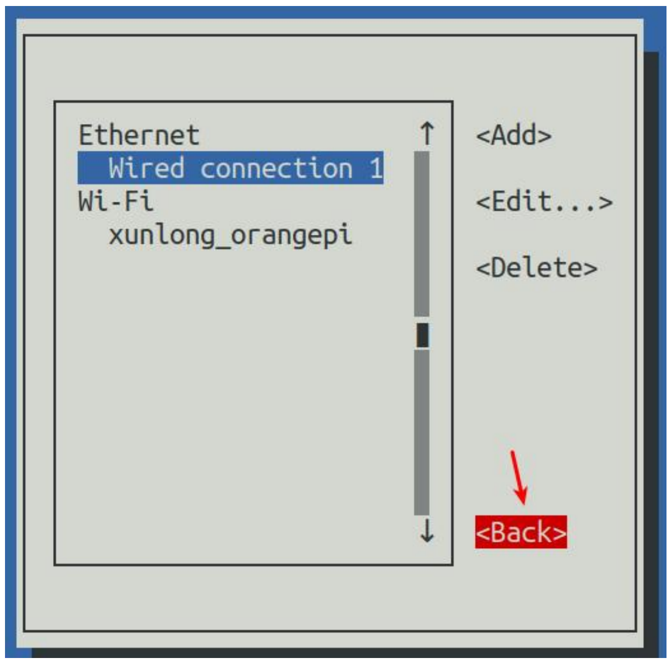
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the position shown in the figure below (there are many other setting options in it, please explore by yourself), **please set according to your specific needs, The values set in the image below are just an example**



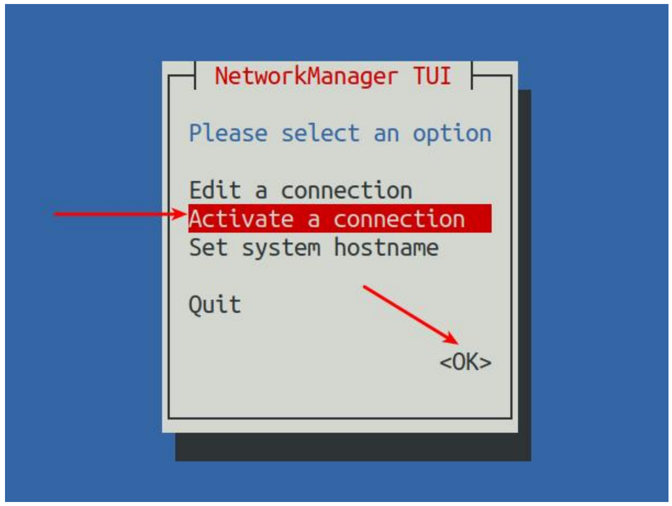
11) After setting, move the cursor to **<OK>** in the lower right corner, and press Enter to confirm



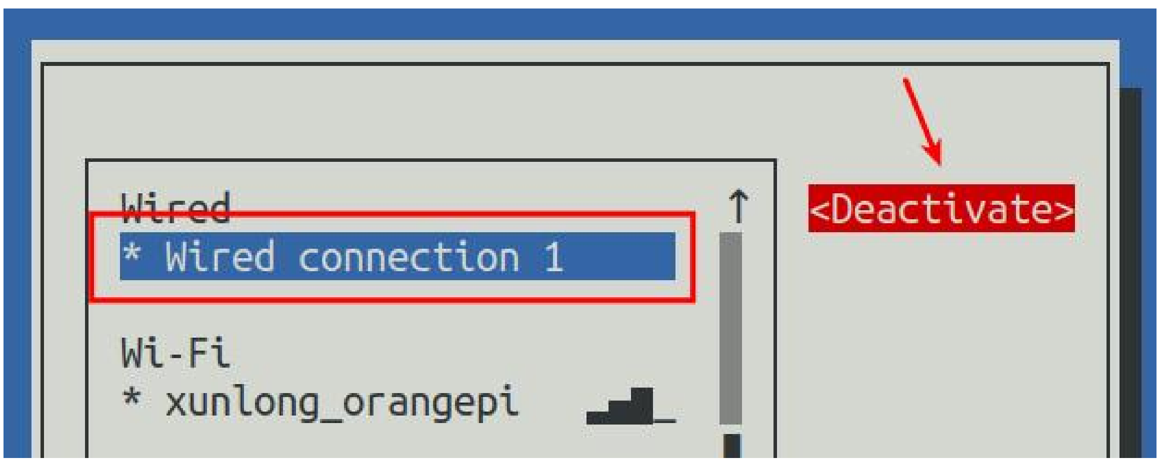
12) Then click **<Back>** to return to the previous selection interface



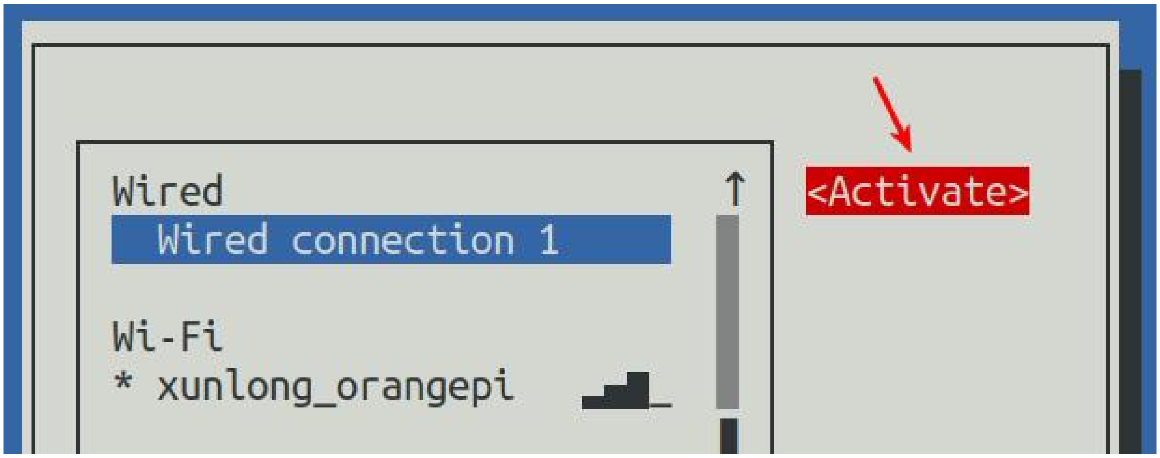
13) Then select **Activate a connection**, move the cursor to **<OK>**, and finally click Enter



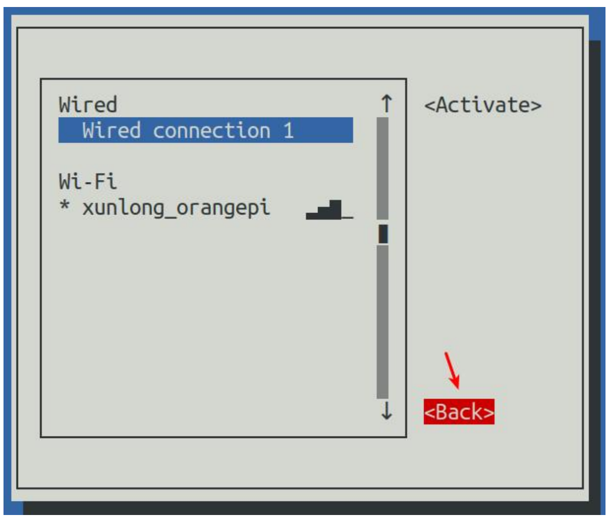
14) Then select the network interface that needs to be set, such as **Wired connection 1**, then move the cursor to **<Deactivate>**, and press Enter to disable **Wired connection 1**

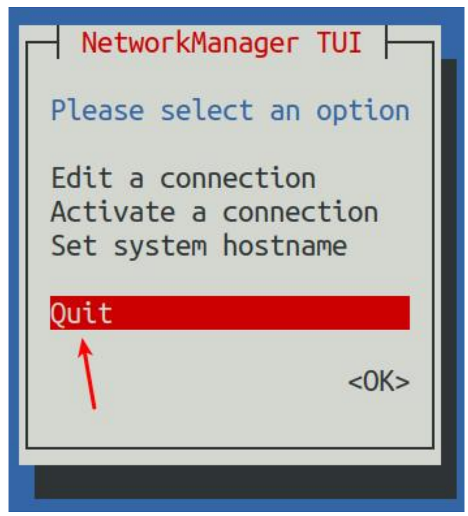


15) Then please do not move the cursor, and then press the Enter key to re-enable **Wired connection 1**, so that the static IP address set earlier will take effect



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





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

```
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:ac:14:a5:92:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.177/24 brd 192.168.1.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 241e:3b8:3240:c3a0:e269:8305:dc08:135e/64 scope global dynamic noprefixroute
       valid_lft 259149sec preferred_lft 172749sec
    inet6 fe80::957d:bbbe:4928:3604/64 scope link noprefixroute
```

```
        valid_lft forever preferred_lft forever
```

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

```
orangepi@orangepi:~$ ping 192.168.1.47 -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
```

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

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

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

a. **orangepi** is the name of the WIFI network interface (the name is not necessarily the same)
b. **Wired connection 1** is the name of the Ethernet interface

```
orangepi@orangepi:~$ nmcli con show
NAME                UUID                                  TYPE      DEVICE
orangepi            cfc4f922-ae48-46f1-84e1-2f19e9ec5e2a  wifi      wlan0
Wired connection 1  9db058b7-7701-37b8-9411-efc2ae8bfa30  ethernet  eth0
```

3) Then enter the following command, where

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

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

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

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

4) Then restart the linux system

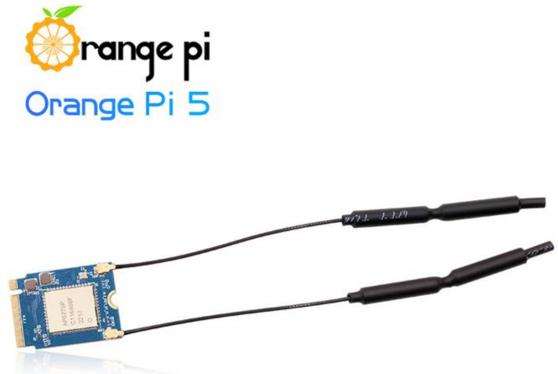
```
orangepi@orangepi:~$ sudo reboot
```

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

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

3. 6. 4. How to use AP6275P PCIe network card

1) First, you need to purchase an AP6275P PCIe network card as shown in the figure below



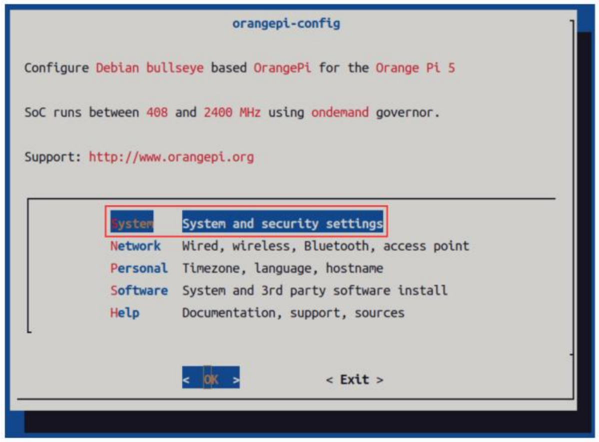
2) Then insert the AP6275P PCIe network card into the M.2 interface of the development board and fix it

3) Then open the configuration of the AP6275P PCIe network card in the linux system, the steps are as follows:

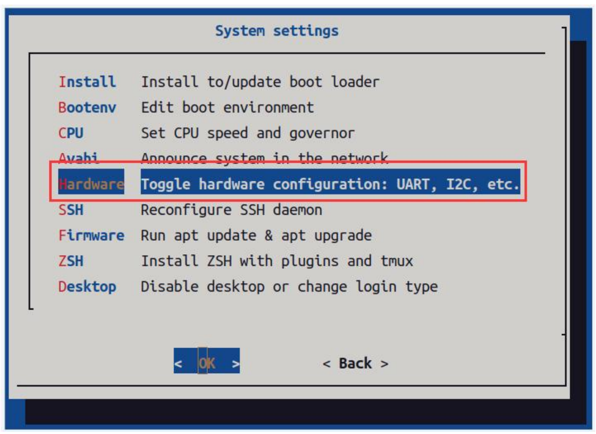
a. First run **orangepi-config**, normal users remember to add **sudo** permission

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

b. Then select **System**



c. Then select **Hardware**



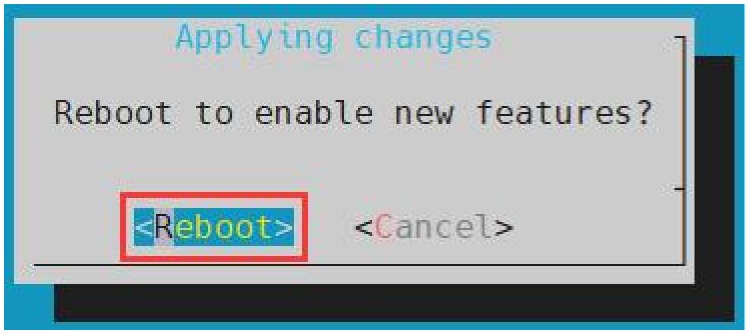
d. Then use the arrow keys on the keyboard to navigate to **wifi-ap6275p**, and then use **the space** to select

Please do not select ssd-sata at the same time.

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

f. Then select **<Back>**

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



The above settings will eventually add the configuration of overlays=wifi-ap6275p to /boot/orangepiEnv.txt. After setting, you can check it first. If this configuration does not exist, then there is a problem with the settings.

If you find it troublesome to use orangepi-config, you can also open /boot/orangepiEnv.txt, and then add the configuration of overlays=wifi-ap6275p.

```
orangepi@orangepi:~$ cat /boot/orangepiEnv.txt | grep “ap6275p”
overlays=wifi-ap6275p
```

4) If everything is normal after restarting the system, use the following command to see the device nodes of WIFI and Bluetooth

a. The command to view the WIFI device node is as follows:

```
orangepi@orangepi:~$ ip addr show wlan0
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state DORMANT group default qlen 1000
    link/ether 70:f7:54:b8:b3:17 brd ff:ff:ff:ff:ff:ff
```

b. The command to view the Bluetooth device node is as follows:

```
orangepi@orangepi:~$ hciconfig -a
hci0:    Type: Primary    Bus: UART
         BD Address: 82:CC:AE:62:CE:3E    ACL MTU: 1021:8    SCO MTU: 64:1
         UP RUNNING
         RX bytes:958 acl:0 sco:0 events:73 errors:0
```

```
TX bytes:5544 acl:0 sco:0 commands:73 errors:0
Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
Link policy: RSWITCH SNIFF
Link mode: SLAVE ACCEPT
Name: 'orangepi5'
Class: 0x1c0000
Service Classes: Rendering, Capturing, Object Transfer
Device Class: Miscellaneous,
HCI Version: 5.1 (0xa)    Revision: 0x3f9
LMP Version: 5.1 (0xa)    Subversion: 0x1111
Manufacturer: Broadcom Corporation (15)
```

5) For wifi connection and test methods, please refer to the section of **WIFI connection test**, so I won't go into details here

6) For the test method of Bluetooth, please refer to the section on **Bluetooth usage**, so I won't go into details here

3.6.5. AP6275P PCIe network card creates WIFI hotspot through create_ap

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

https://github.com/oblique/create_ap

If you are using the latest image, the create_ap script has been pre-installed, and you can create a WIFI hotspot through the create_ap command. The basic command format of create_ap is as follows:

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

*** options：You can use this parameter to specify the encryption method, the frequency band of the WIFI hotspot, the bandwidth mode, the network sharing**

method, etc. You can get the options through create_ap -h
*** wifi-interface: The name of the wireless network card**
*** interface-with-internet: The name of the network card that can be connected to the Internet, generally eth0**
*** access-point-name: hotspot name**
*** passphrase: hotspot password**

3. 6. 5. 1. create_ap method to create WIFI hotspot in NAT mode

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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi
```

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
Config dir: /tmp/create_ap.wlan0.conf.fPItFUJ2
PID: 3831
Network Manager found, set ap0 as unmanaged device... DONE
Creating a virtual WiFi interface... ap0 created.
Sharing Internet using method: nat
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.fPItFUJ2/hostapd_ctrl
ap0: interface state UNINITIALIZED->ENABLED
ap0: AP-ENABLED
```

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

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

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

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

```
        RX errors 0    dropped 50    overruns 0    frame 0
        TX packets 3798    bytes 1519493 (1.5 MB)
        TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
        device interrupt 83
```

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

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi -g 192.168.2.1
```

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

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --freq-band 5
```

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --hidden
```

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

3. 6. 5. 2. create_ap method to create WIFI hotspot in bridge mode

1）Eer 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
```

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi
[sudo] password for orangepi:
Config dir: /tmp/create_ap.wlan0.conf.fg9U5Xgt
PID: 3141
Network Manager found, set ap0 as unmanaged device... DONE
Creating a virtual WiFi interface... ap0 created.
Sharing Internet using method: bridge
Create a bridge interface... br0 created.
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.fg9U5Xgt/hostapd_ctrl
ap0: interface state UNINITIALIZED->ENABLED
ap0: AP-ENABLED
```

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

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

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

```
orangepi@orangepi:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

```
inet 192.168.1.150  netmask 255.255.255.0  broadcast 192.168.1.255
inet6 fe80::938f:8776:5783:afa2  prefixlen 64  scopeid 0x20<link>
ether 4a:a0:c8:25:42:82  txqueuelen 1000  (Ethernet)
RX packets 25370  bytes 2709590 (2.7 MB)
RX errors 0  dropped 50  overruns 0  frame 0
TX packets 3798  bytes 1519493 (1.5 MB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
device interrupt 83
```

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

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --freq-band 5
```

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --hidden
```

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

3. 7. SSH remote login development board

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

3. 7. 1. SSH remote login 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 root@192.168.1.xxx          (Need to be replaced with the IP address of the development board)
root@192.168.1.xx's password:         （Enter the password here, the default password is orangepi）
```

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

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

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

```
test@test:~$ ssh root@192.168.1.150
root@192.168.1.150's password:
  ___  ____  _ ____
 / _ \|  _ \(_) ___|
| | | | |_) | |___ \
| |_| |  __/| |___) |
 \___/|_|   |_|____/

Welcome to Orange Pi 1.0.0 Bullseye with Linux 5.10.110-rockchip-rk3588

System load:   1%               Up time:       9 min
Memory usage:  2% of 7.51G      IP:            192.168.1.150
CPU temp:      49°C             Usage of /:    12% of 15G

Last login: Thu Dec  1 12:57:42 2022
root@orangepi5:~#
```

If ssh cannot log in to the linux system normally, please first check whether the IP address of the development board can be pinged. If the ping is ok, you can log in to the linux system through the serial port or HDMI display and then enter the following command on the development board and try again. Is it possible to connect:

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

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

3. 7. 2. SSH remote login development board under Windows

1) First obtain the IP address of the development board

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

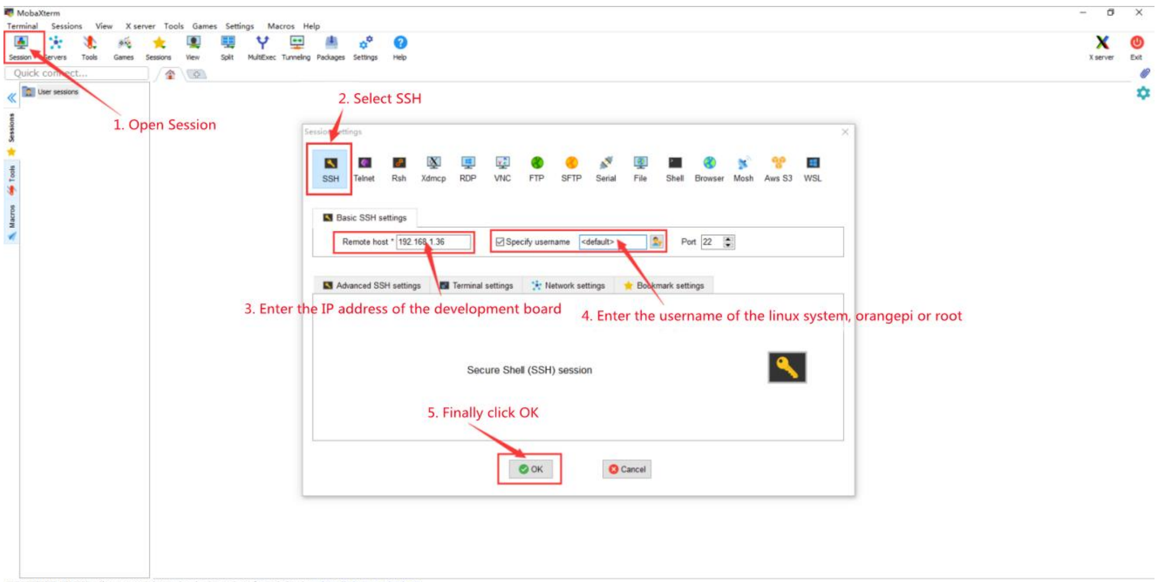
a. Open **Session**

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

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

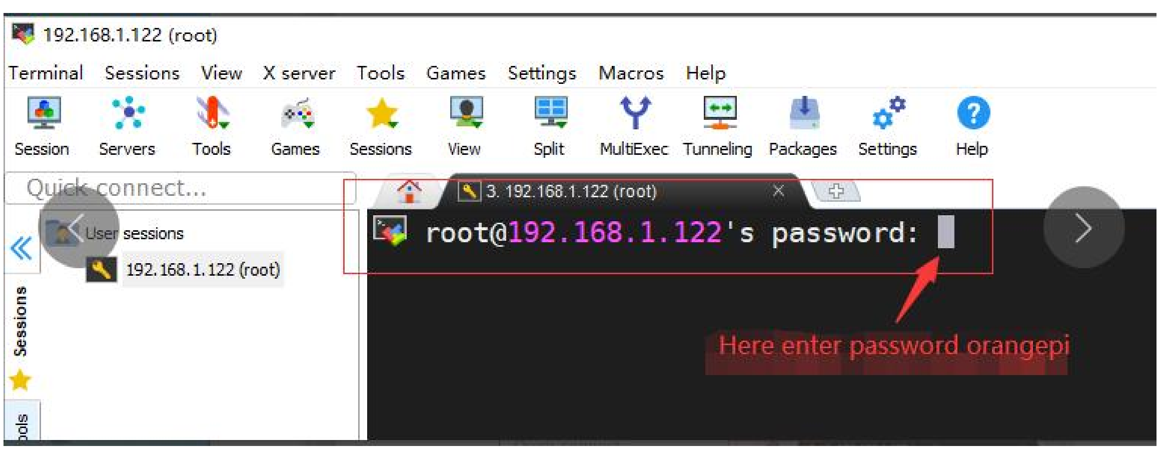
d. Then enter the user name **root** or **orangepi** of the linux system in **Specify username**

e. Finally click **OK**

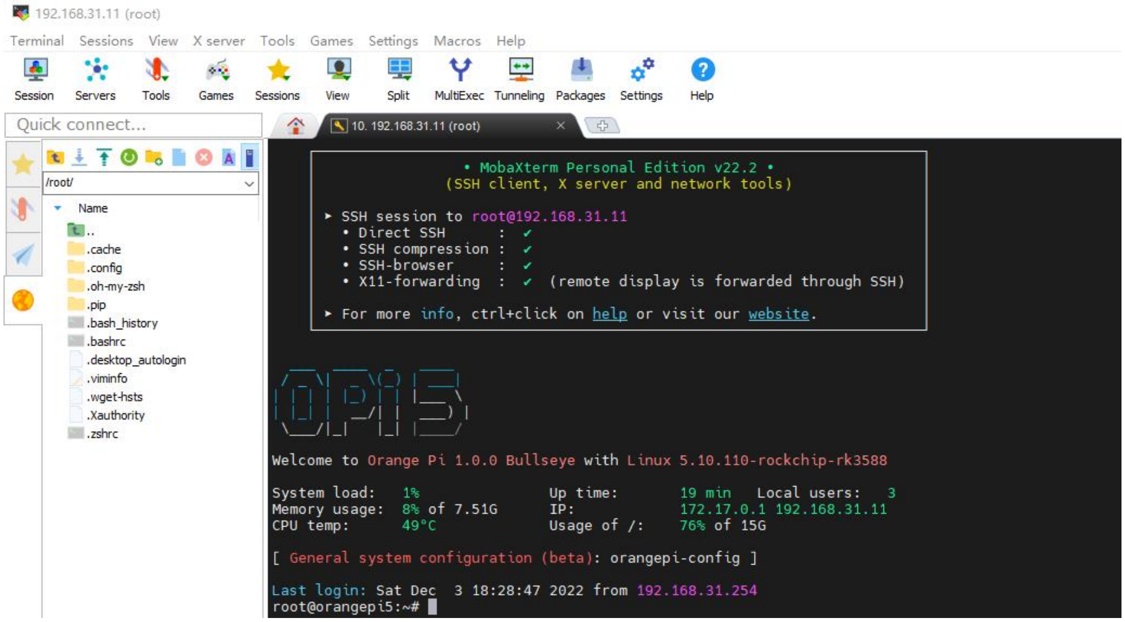


3) Then you will be prompted to enter a password. The default passwords for root and orangepi users are orangepi

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



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



3. 8. How to use ADB

3. 8. 1. How to use network adb

1) After the system starts, please confirm that **adbd** has been started

orangepi@orangepi:~$ **ps -ax | grep "adbd"**
808 ? Sl 0:00 /usr/bin/adbd
3707 ttyFIQ0 S+ 0:00 grep --color=auto adbd

2) Then check the IP address of the development board and write it down

3) Then install the adb tool on the Ubuntu PC

test@test:~$ **sudo apt-get update**
test@test:~$ **sudo apt-get install -y adb**

4) Then use the following command to connect to the network adb

test@test:~$ **adb connect 192.168.1.xx:5555** **#IP address please replace with the IP address of the development board**
* daemon not running; starting now at tcp:5037
* daemon started successfully

```
connected to 192.168.1.xx:5555
test@test:~$ adb devices
List of devices attached
192.168.1.xx:5555          device
```

5) Then use the following command to log in to the linux system of the development board

```
test@test:~$ adb shell
root@orangepi5:/#      <--- After seeing this prompt, it means that you have successfully logged in to the development board
```

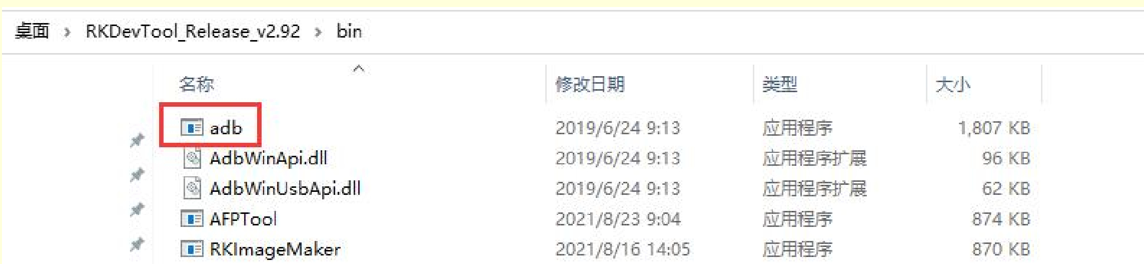
6) The command to upload files to the development board using adb is as follows

```
test@test:~$ adb push filename /root
filename: 1 file pushed. 3.7 MB/s (1075091 bytes in 0.277s)
```

7) The command to restart the development board using adb is as follows

```
test@test:~$ adb reboot
```

If you do not have the adb tool in your Windows system, you can use the adb program in the RKDevTool software (this software is useful in the section on how to burn the Android image to SPIFlash+NVMe SSD).



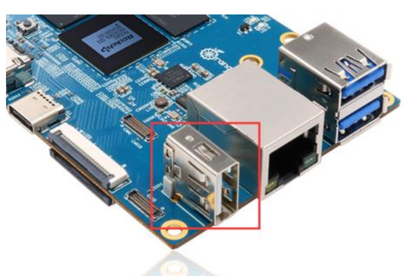
An example using adb in Windows looks like this:

```
命令提示符
Microsoft Windows [版本 10.0.19044.2251]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Administrator>cd C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>dir
 驱动器 C 中的卷没有标签。
 卷的序列号是 62AE-5AED

 C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin 的目录

2022/08/09  13:19    <DIR>          .
2022/08/09  13:19    <DIR>          ..
2019/06/24  09:13         1,850,368 adb.exe
2019/06/24  09:13            97,792 AdbWinApi.dll
2019/06/24  09:13            62,976 AdbWinUsbApi.dll
2021/08/23  09:04           894,976 AFPTool.exe
2021/08/16  14:05           890,368 RKImageMaker.exe
               5 个文件      3,796,480 字节
               2 个目录 64,033,034,240 可用字节

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>.\adb.exe connect 192.168.1.144
connected to 192.168.1.144:5555

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>.\adb.exe devices
List of devices attached
192.168.1.144:5555      device


C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>.\adb.exe push adb.exe /root
adb.exe: 1 file pushed. 4.1 MB/s (1850368 bytes in 0.427s)

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>
```

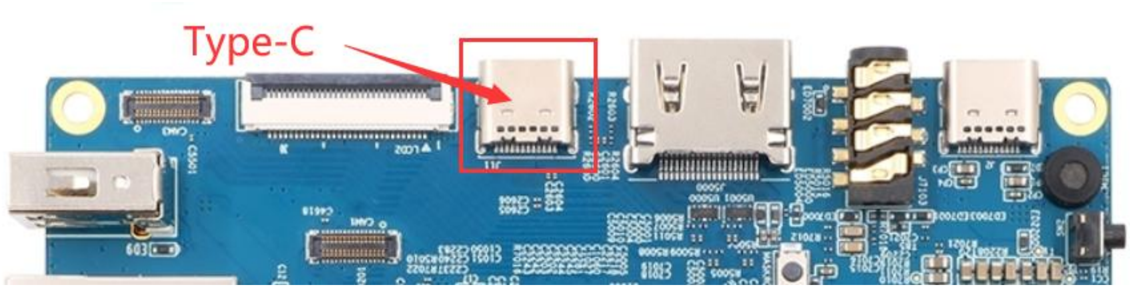
3.8.2. Use a type-c data cable to connect to adb

1) First prepare a good quality Type-C data cable

2) Then please make sure that there is no USB device plugged into the USB interface below



3) Then connect the development board and Ubuntu PC through the Type-C data cable. The position of the Type-C interface of the development board is shown in the figure below:



4) Then run the following command to set the Type-C interface to **device** mode

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

If the **set_device.sh** script does not exist in the Linux system, please use the following command directly:

```
orangepi@orangepi:~$ sudo bash -c "echo device > /sys/kernel/debug/usb/fc000000.usb/mode"
orangepi@orangepi:~$ sudo systemctl restart usbdevice
```

5) **Then please make sure that adbd has been started**

```
orangepi@orangepi:~$ ps -ax | grep "adbd"
    808 ?            Sl       0:00 /usr/bin/adbd
   3707 ttyFIQ0    S+       0:00 grep --color=auto adbd
```

6) Then install the adb tool on the Ubuntu PC

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

7) Then use the following command to see if the adb device is recognized

```
test@test:~$ adb devices
List of devices attached
e0f9f71bc343c305     device
```

8) Then use the following command to log in to the linux system of the development board

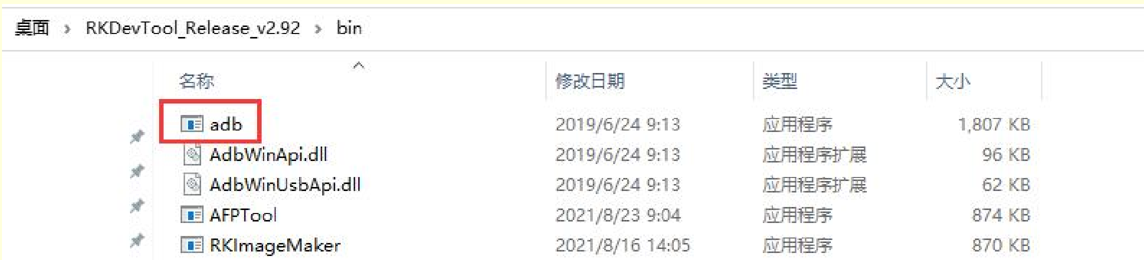
```
test@test:~$ adb shell
```

```
root@orangepi5:/#        <--- After seeing this prompt, it means that you have successfully logged in to the development board
```

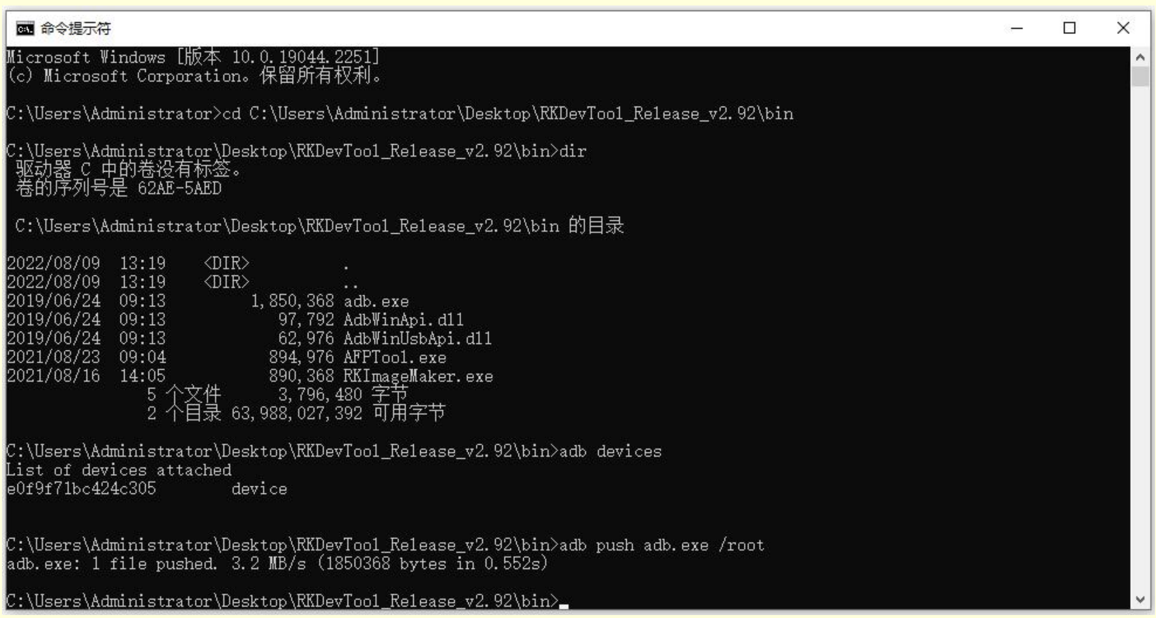
9) The command to upload files to the development board using adb is as follows

```
test@test:~$ adb push filename /root
filename: 1 file pushed. 3.7 MB/s (1075091 bytes in 0.277s)
```

If you do not have the adb tool in your Windows system, you can use the adb program in the RKDevTool software (this software is useful in the section on how to burn the Android image to SPIFlash+NVMe SSD).



An example using adb in Windows looks like this:



3.9. The method of uploading files to the Linux system of the development board

3.9.1. The method of uploading files to the development board Linux system in Ubuntu PC

3.9.1.1. How to upload files using the scp command

1) Use the scp command to upload files from the Ubuntu PC to the Linux system of the development board. The specific commands are as follows

a. **file_path:** need to be replaced with the path of the file to be uploaded
b. **orangepi:** It is the user name of the Linux system of the development board, and it can also be replaced with other ones, such as root
c. **192.168.xx.xx:** It is the IP address of the development board, please modify it according to the actual situation
d. **/home/orangepi:** The path in the Linux system of the development board, which can also be modified to other paths

```
test@test:~$ scp  file_path  orangepi@192.168.xx.xx:/home/orangepi/
```

2) If you want to upload a folder, you need to add the -r parameter

```
test@test:~$ scp  -r  dir_path  orangepi@192.168.xx.xx:/home/orangepi/
```

3) There are more usages of scp, please use the following command to view the man manual

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

3.9.1.2. How to upload files using filezilla

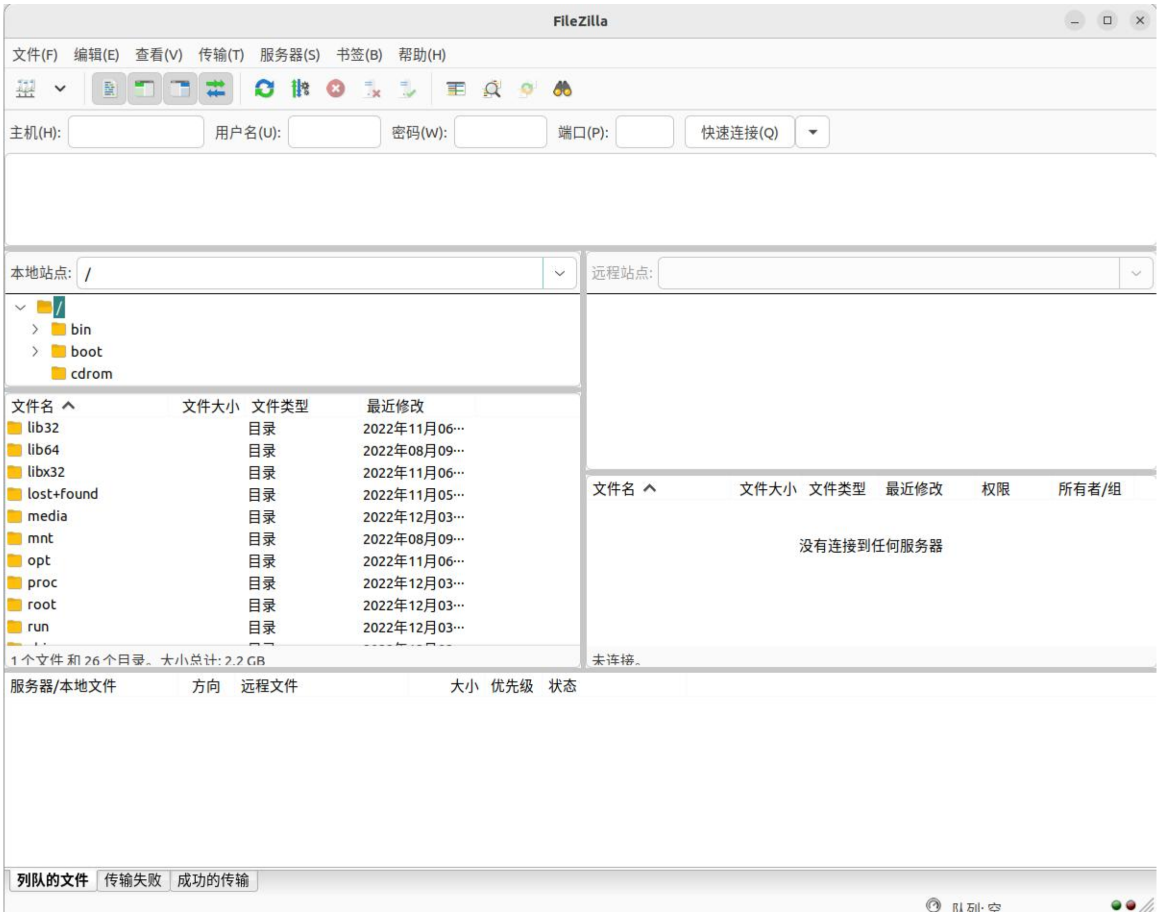
1) First install filezilla in Ubuntu PC

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

2) Then use the following command to open filezilla

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

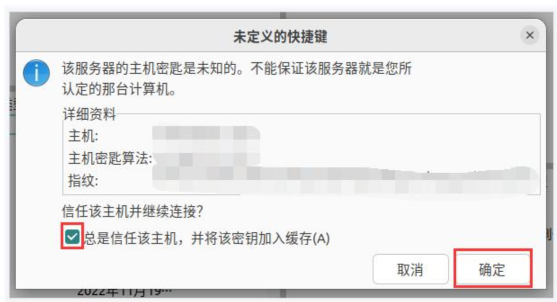
3) The interface after filezilla is opened is as follows, at this time, the display under the remote site on the right is empty



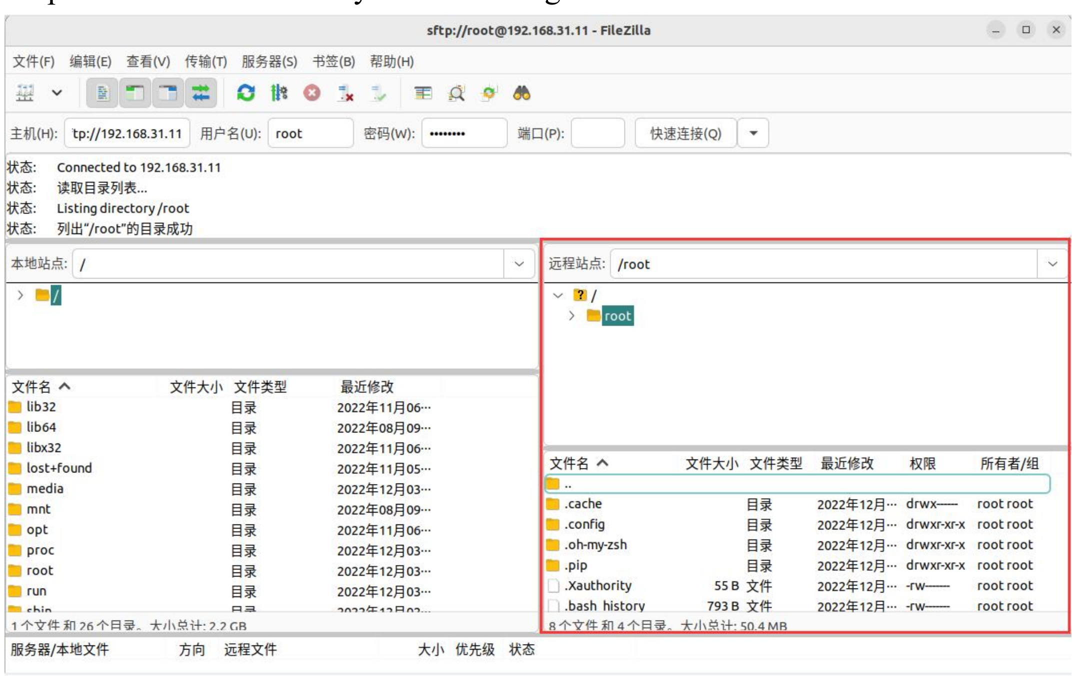
4) The method of connecting the development board is shown in the figure below

5) Then choose to **save the password**, and then click **OK**

6) Then choose to **always trust this host**, and then click **OK**

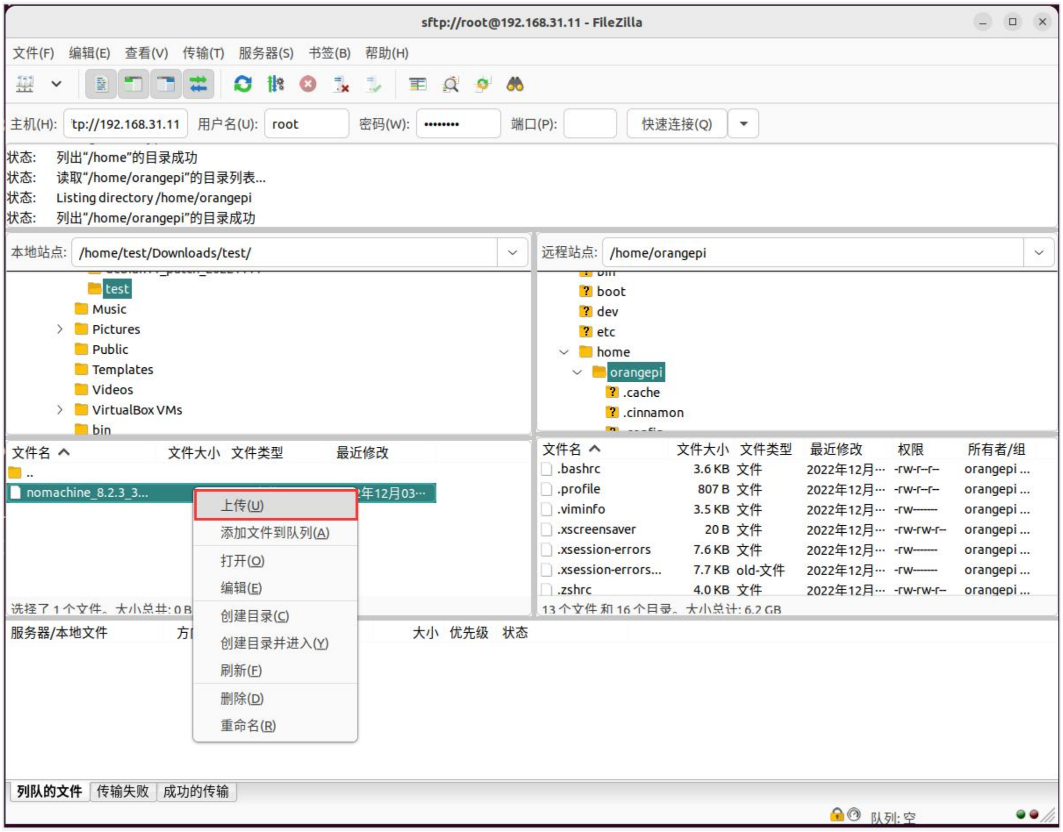


7) After the connection is successful, you can see the directory structure of the development board linux file system on the right side of the filezilla software



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

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



9) After the upload is complete, you can go to the corresponding path in the development board linux system to view the uploaded files

10) The method of uploading folders is the same as that of uploading files, so I won't go into details here

3.9.2. The method of uploading files to the Linux system of the development board in Windows PC

3.9.2.1. 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#close

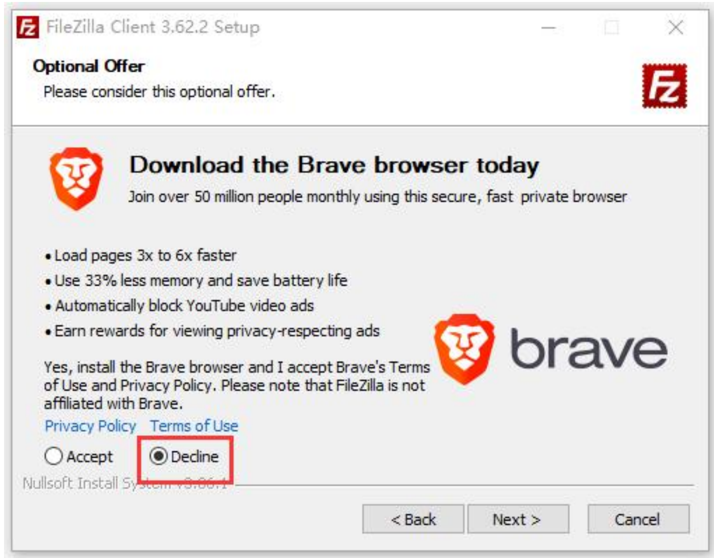
Please select your edition of FileZilla Client

	FileZilla	FileZilla with manual	FileZilla Pro	FileZilla Pro + CLI
Standard FTP	Yes	Yes	Yes	Yes
FTP over TLS	Yes	Yes	Yes	Yes
SFTP	Yes	Yes	Yes	Yes
Comprehensive PDF manual	-	Yes	Yes	Yes
Amazon S3	-	-	Yes	Yes
Backblaze B2	-	-	Yes	Yes
Dropbox	-	-	Yes	Yes
Microsoft OneDrive	-	-	Yes	Yes
Google Drive	-	-	Yes	Yes
Google Cloud Storage	-	-	Yes	Yes
Microsoft Azure Blob + File Storage	-	-	Yes	Yes
WebDAV	-	-	Yes	Yes
OpenStack Swift	-	-	Yes	Yes
Box	-	-	Yes	Yes
Site Manager synchronization	-	-	Yes	Yes
Command-line interface	-	-	-	Yes
Batch transfers	-	-	-	Yes
	Download	Select	Select	Select

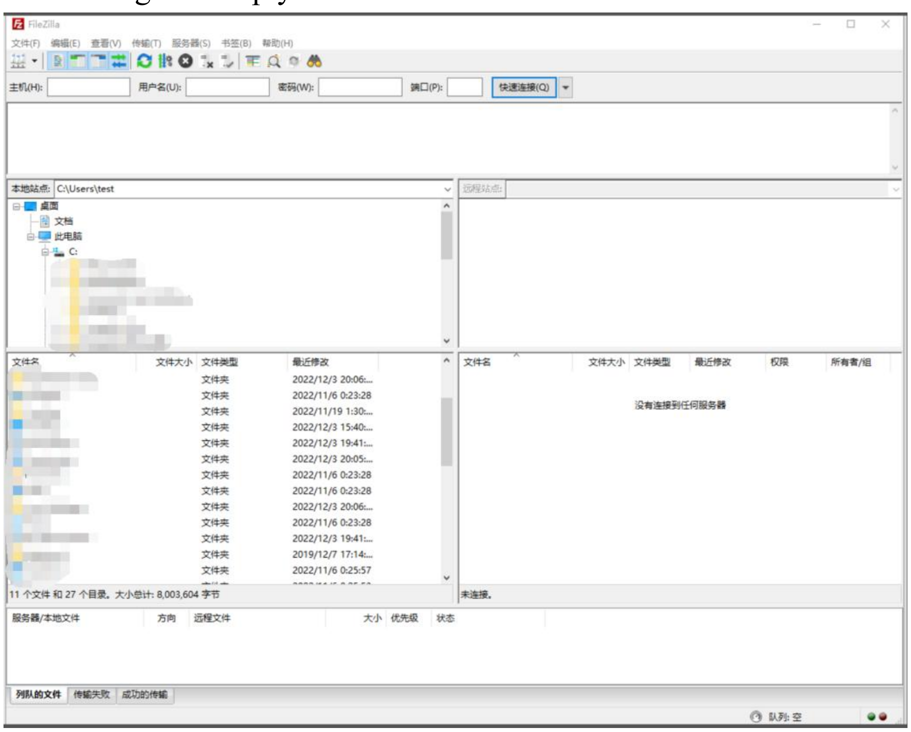
2) The downloaded installation package is as follows, and then double-click to install directly

```
FileZilla_Server_1.5.1_win64-setup.exe
```

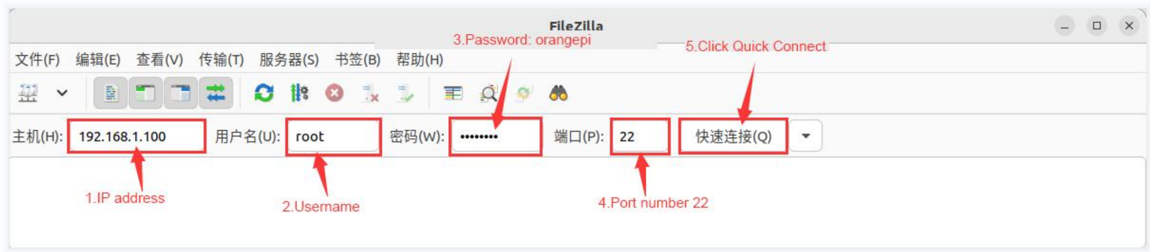
During the installation process, please select **Decline** on the following installation interface, and then select **Next>**



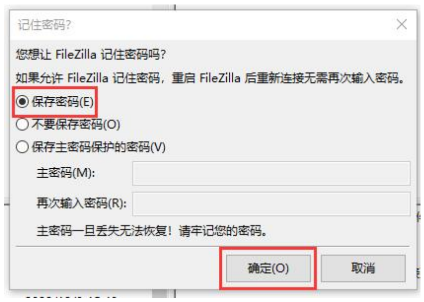
3) The interface after filezilla is opened is as follows, at this time, the display under the remote site on the right is empty



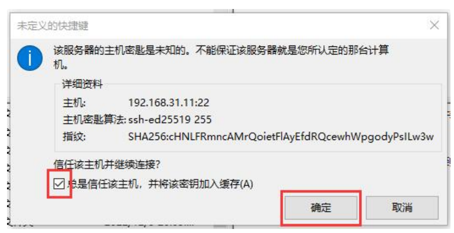
4) The method of connecting the development board is shown in the figure below:



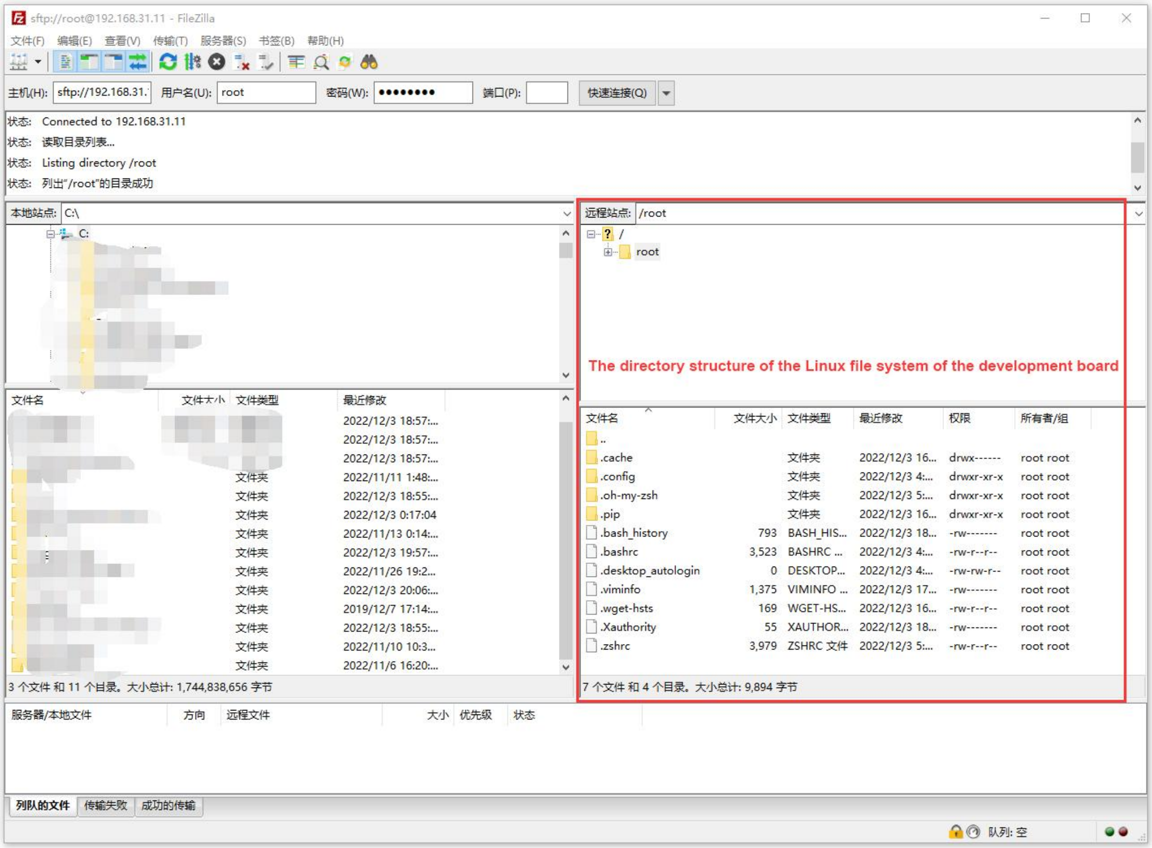
5) Then choose to **save the password**, and then click **OK**



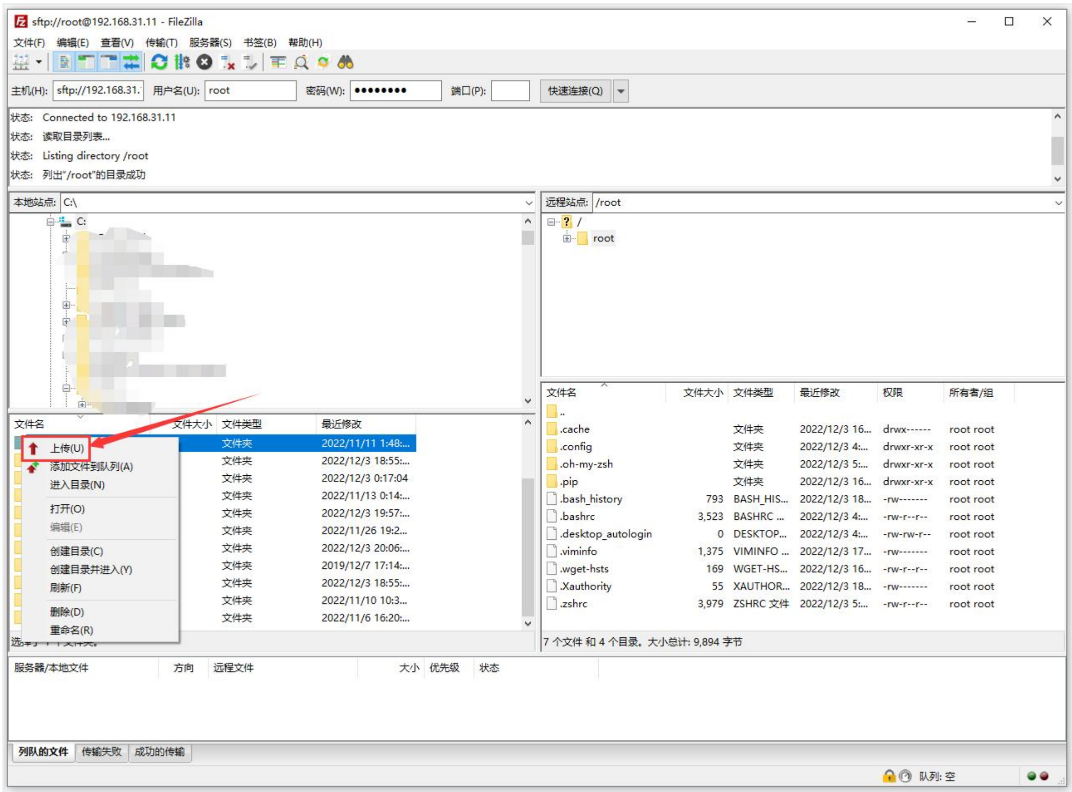
6) Then select **Always trust this host**, and click **OK**



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



8) Then select the path to be uploaded to the development board on the right side of the filezilla software, and then select the file to be uploaded on the Windows PC on the left side of the filezilla software, then click the right mouse button, and then click the upload option to start uploading the file to the development board bingo



9) After the upload is complete, you can go to the corresponding path in the Linux system of the development board to view the uploaded file

10) The method of uploading a folder is the same as that of uploading a file, so I won't go into details here

3. 10. HDMI Test

3. 10. 1. HDMI display test

1) Use 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 in normal use.

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

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

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

3. 10. 2. HDMI to VGA display test

1) First, you need to prepare the following accessories

a. HDMI to VGA converter

b. A VGA cable

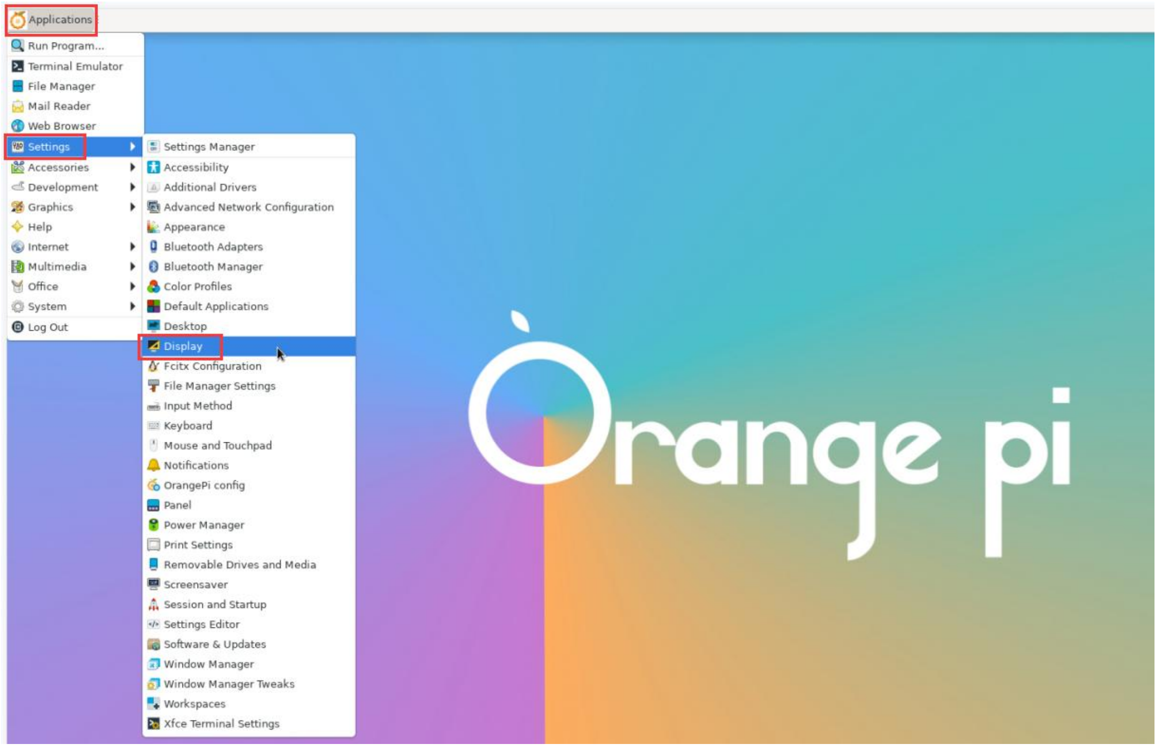
c. A monitor or TV that supports VGA port

2) The HDMI to VGA display test is as follows

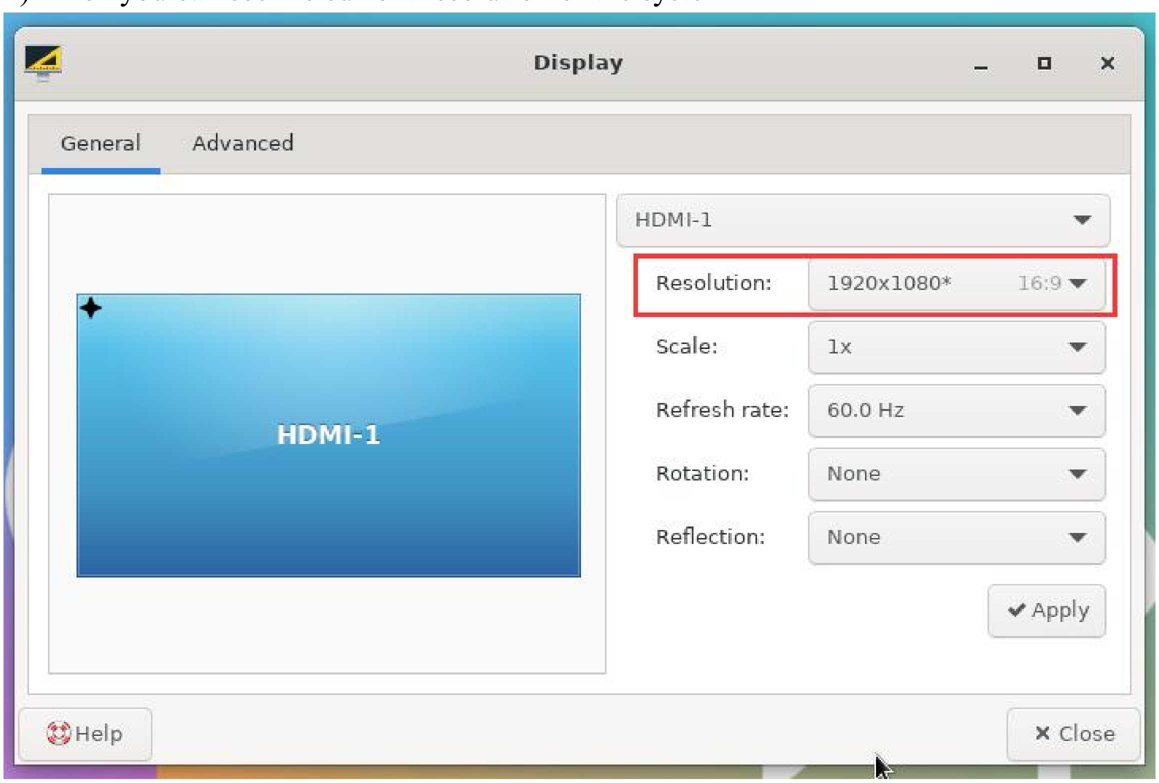
When using HDMI to VGA display, the development board and the Linux system of the development board do not need to make any settings, only the HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI to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

3. 10. 3. HDMI resolution setting method

1) First open **Display** in **Settings**

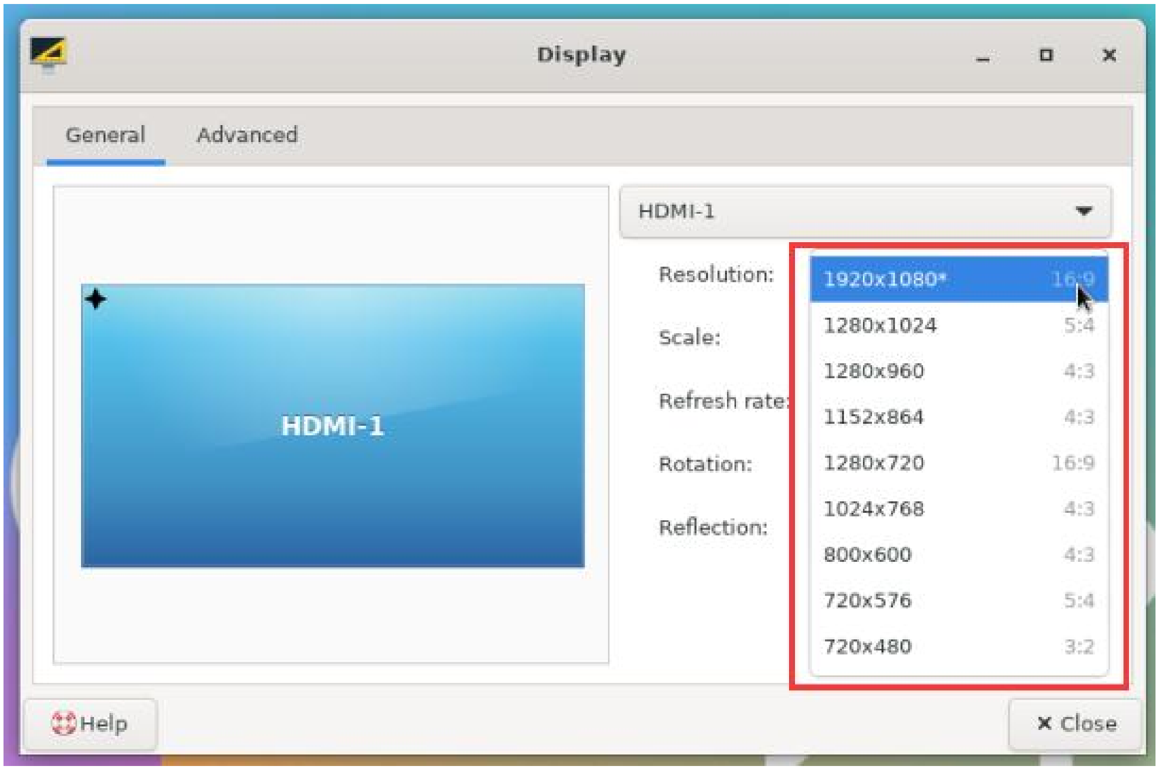


2) Then you can see the current resolution of the system

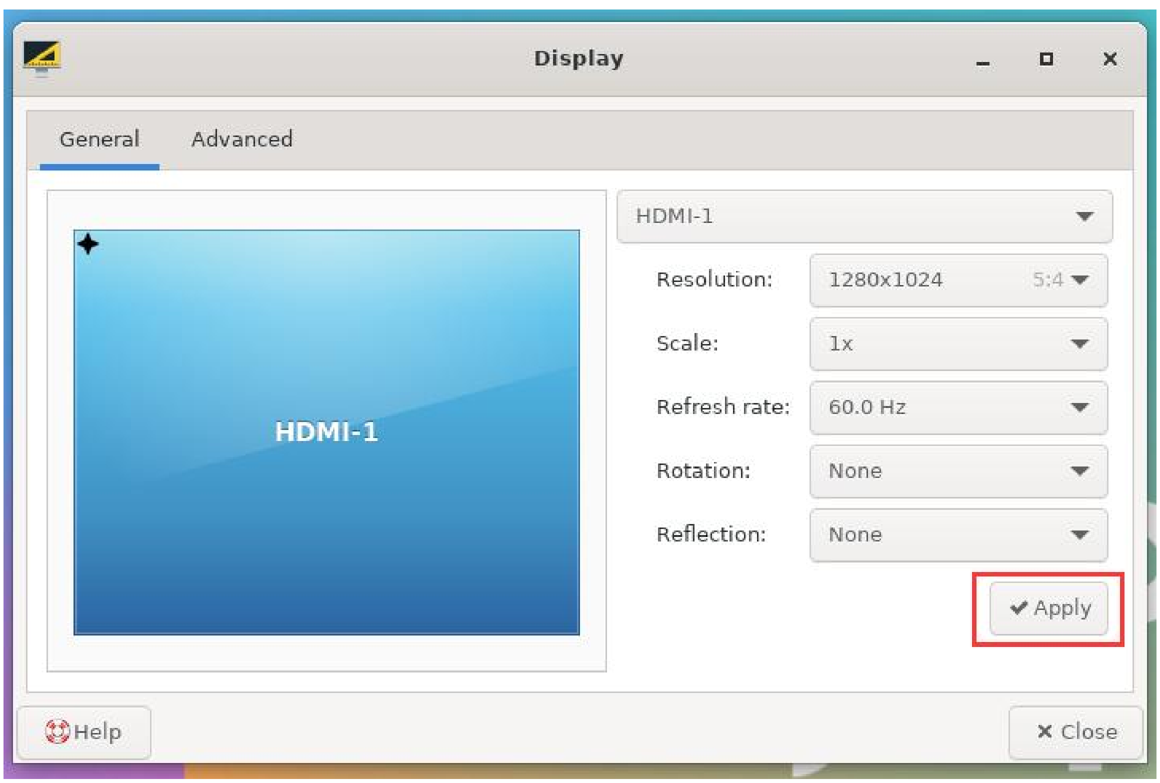


3) Click the drop-down box of Resolution to see all resolutions currently supported by

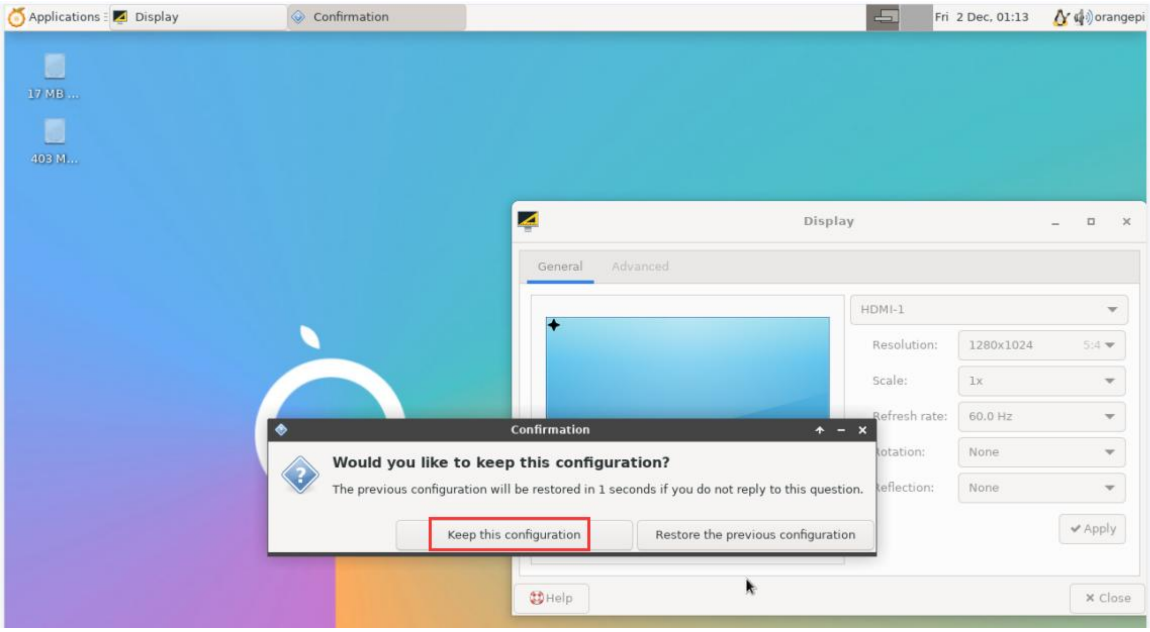
the monitor



4) Then select the resolution you want to set, and click Apply



5) After the new resolution is set, select **Keep the configuration**



3. 11. How to Use Bluetooth

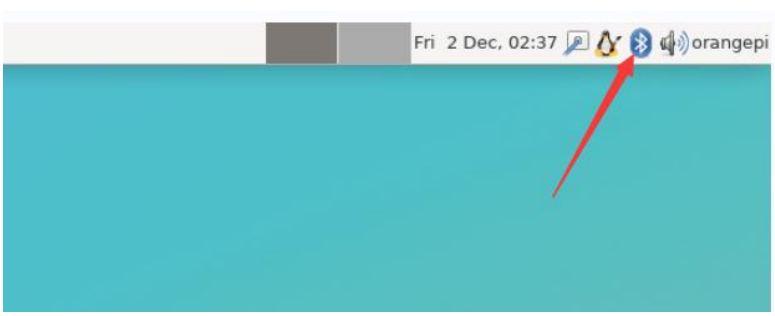
Please note that there is no Bluetooth module on the Orange Pi 5 development board, and an external PCIe network card with Bluetooth or a USB network card with Bluetooth is required to use the Bluetooth function.

For instructions on using the external PCIe network card, please refer to the section on how to use the AP6275P PCIe network card.

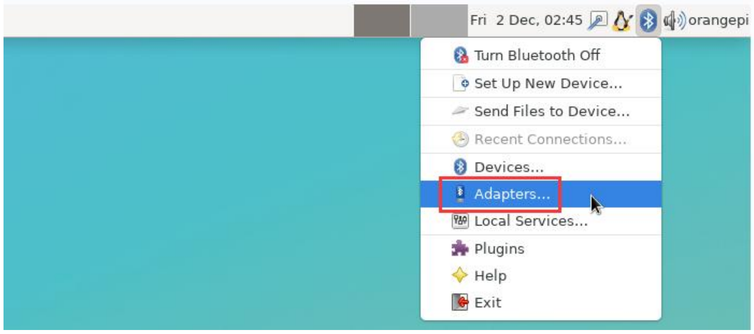
For instructions on using the external USB network card, please refer to the USB wireless network card test section.

3. 11. 1. Test method of desktop image

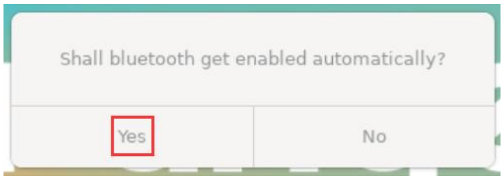
1) Click on the Bluetooth icon in the upper right corner of the desktop



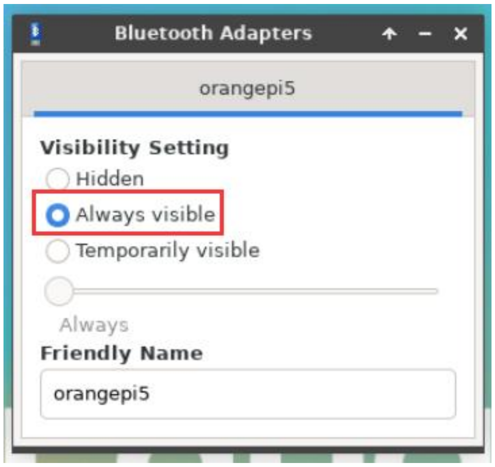
2) Then select the adapter



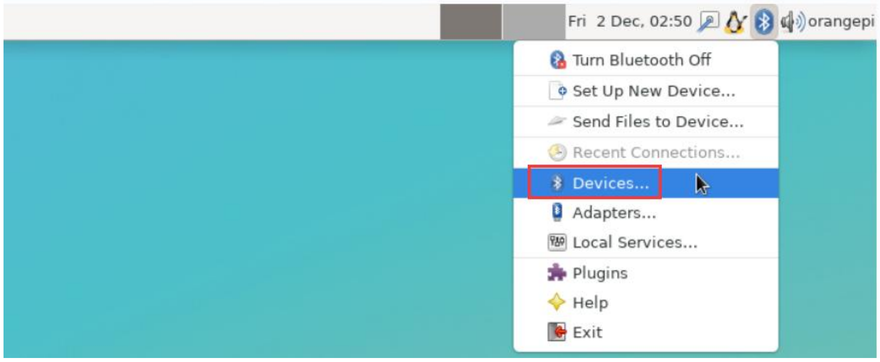
3) If prompted the following interface, please select **Yes**



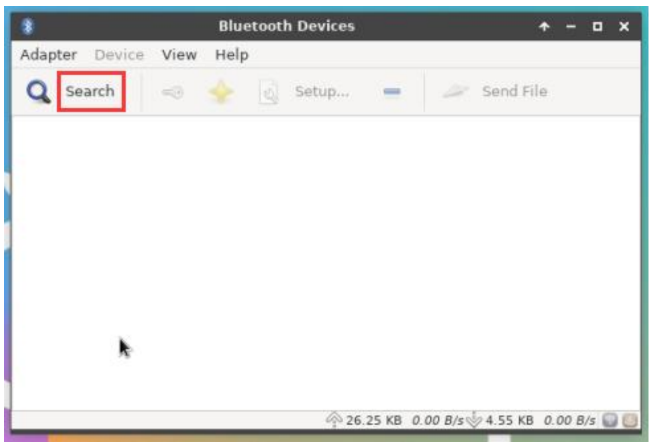
4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter setting interface, and then close it



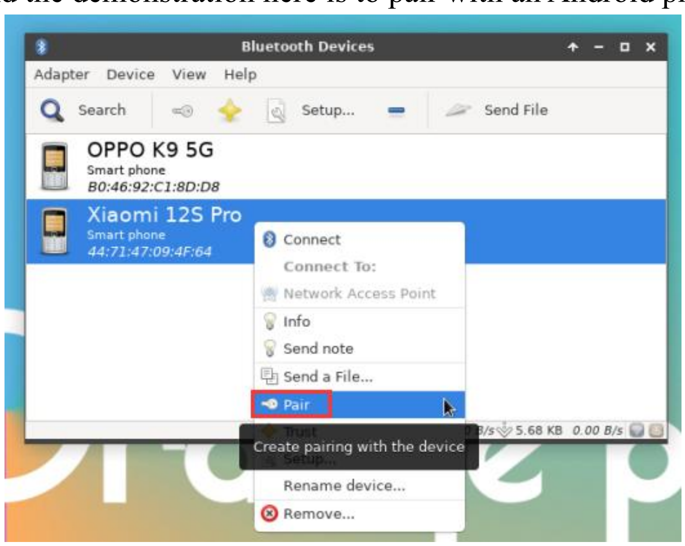
5) Then open the configuration interface of the Bluetooth device



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

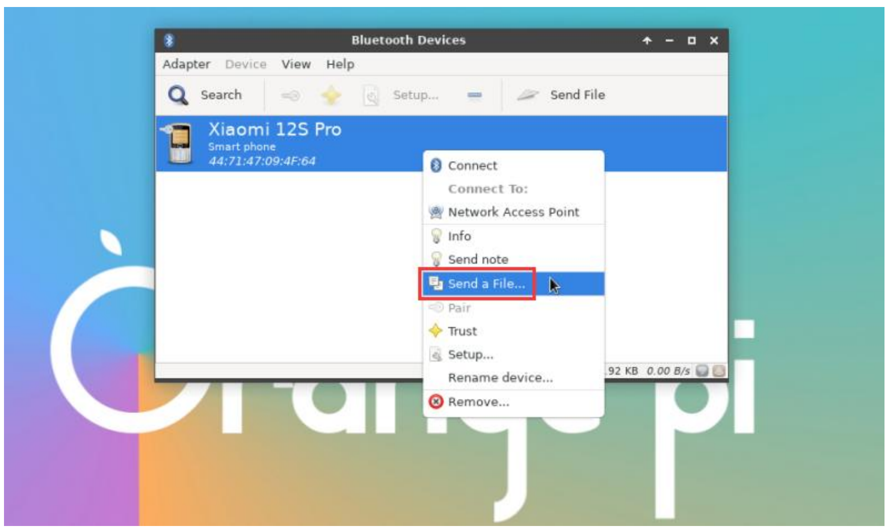


6) Then select the Bluetooth device you want to connect to, and then click the right button of the mouse to pop up the operation interface of the Bluetooth device, select **Pair** to start pairing, and the demonstration here is to pair with an Android phone



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

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



9) The interface for sending pictures is as follows

3. 12. USB Interface Test

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

3. 12. 1. Connect USB mouse or keyboard test

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

2) Connect Orange Pi board to HDMI display

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

3. 12. 2. Connect USB storage device test

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

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

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

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

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

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

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

3. 12. 3. USB wireless network card test

The usable USB wireless network cards that **have been tested** so far are as follows. For other types of USB wireless network cards, please test them yourself. If they cannot be used, you need to transplant the corresponding USB wireless network card driver.

serial number	model	
1	RTL8723BU Support 2.4G WIFI+BT4.0	
2	RTL8811 Support 2.4G +5G WIFI	
3	RTL8821CU Support 2.4G +5G WIFI Support BT 4.2	

3. 12. 3. 1. RTL8723BU test

1) First insert the RTL8723BU wireless network card module into the USB interface of the development board

2) Then the linux system will automatically load the RTL8723BU bluetooth and WIFI-related kernel modules, through the lsmod command, you can see that the following kernel modules have been automatically loaded

```
orangepi@orangepi:~$ lsmod
Module                  Size  Used by
rfcomm                 57344  16
rtl8xxxu              106496  0
rtk_btusb              61440  0
```

3) Through the dmesg command, you can see the loading information of the RTL8723BU module

```
orangepi@orangepi:~$ dmesg
```

```
......
[    83.438901] usb 2-1: new high-speed USB device number 2 using ehci-platform
[    83.588375] usb 2-1: New USB device found, idVendor=0bda, idProduct=b720, bcdDevice= 2.00
[    83.588403] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[    83.588422] usb 2-1: Product: 802.11n WLAN Adapter
[    83.588443] usb 2-1: Manufacturer: Realtek
[    83.588460] usb 2-1: SerialNumber: 00e04c000001
[    83.601974] Bluetooth: hci0: RTL: examining hci_ver=06 hci_rev=000b lmp_ver=06 lmp_subver=8723
[    83.603894] Bluetooth: hci0: RTL: rom_version status=0 version=1
[    83.603920] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_fw.bin
[    83.610108] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_config.bin
[    83.611274] Bluetooth: hci0: RTL: cfg_sz 68, total sz 22564
[    83.658494] rtk_btusb: Realtek Bluetooth USB driver ver 3.1.6d45ddf.20220519-142432
[    83.658651] usbcore: registered new interface driver rtk_btusb
[    83.667124] usb 2-1: This Realtek USB WiFi dongle (0x0bda:0xb720) is untested!
[    83.667137] usb 2-1: Please report results to Jes.Sorensen@gmail.com
[    83.890140] usb 2-1: Vendor: Realtek
[    83.890153] usb 2-1: Product: 802.11n WLAN Adapter
[    83.890159] usb 2-1: rtl8723bu_parse_efuse: dumping efuse (0x200 bytes):
......
[    83.890412] usb 2-1: RTL8723BU rev E (SMIC) 1T1R, TX queues 3, WiFi=1, BT=1, GPS=0, HI PA=0
[    83.890417] usb 2-1: RTL8723BU MAC: 00:13:ef:f4:58:ae
[    83.890421] usb 2-1: rtl8xxxu: Loading firmware rtlwifi/rtl8723bu_nic.bin
[    83.895289] usb 2-1: Firmware revision 35.0 (signature 0x5301)
[    84.050893] Bluetooth: hci0: RTL: fw version 0x0e2f9f73
[    84.266905] Bluetooth: RFCOMM TTY layer initialized
[    84.266949] Bluetooth: RFCOMM socket layer initialized
[    84.266999] Bluetooth: RFCOMM ver 1.11
[    84.884270] usbcore: registered new interface driver rtl8xxxu
[    84.912046] rtl8xxxu 2-1:1.2 wlx0013eff458ae: renamed from wlan0
```

4) Then through the **sudo ifconfig** command, you can see the device node of

RTL8723BU WIFI. For the connection and test method of WIFI, please refer to the section of **WIFI connection test**, which will not be repeated here

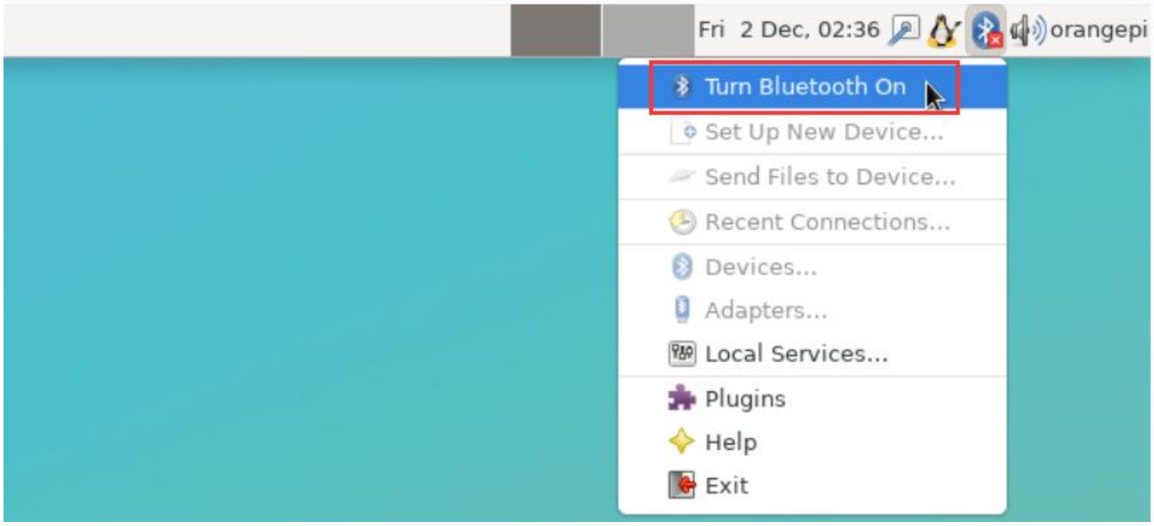
```
orangepi@orangepi:~$ sudo ifconfig wlx0013eff458ae
wlx0013eff458ae: flags=4099<UP,BROADCAST,MULTICAST>    mtu 1500
          ether 00:13:ef:f4:58:ae    txqueuelen 1000    (Ethernet)
          RX packets 0    bytes 0 (0.0 B)
          RX errors 0    dropped 0    overruns 0    frame 0
          TX packets 0    bytes 0 (0.0 B)
          TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
```

5) Then you can see the USB Bluetooth device through the **hciconfig** command

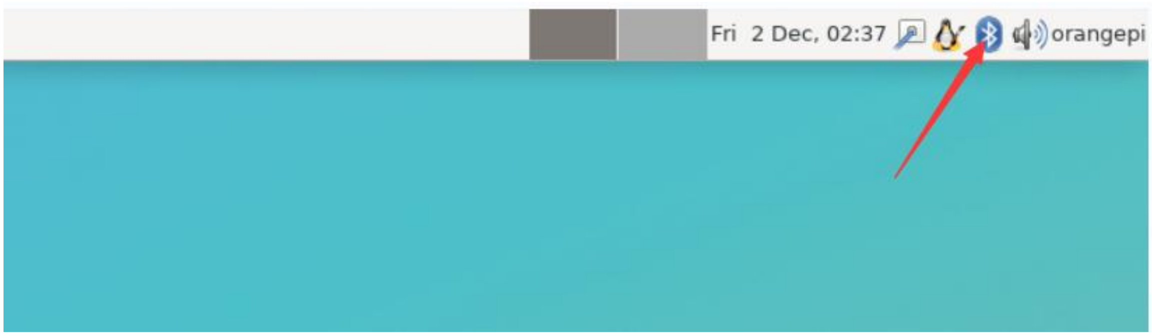
```
orangepi@orangepi:~$ sudo apt update && sudo apt install bluez
orangepi@orangepi:~$ hciconfig
hci0:    Type: Primary    Bus: USB
          BD Address: 00:13:EF:F4:58:AE    ACL MTU: 820:8    SCO MTU: 255:16
          DOWN
          RX bytes:1252 acl:0 sco:0 events:125 errors:0
          TX bytes:23307 acl:0 sco:0 commands:125 errors:0
```

6) You can also see the Bluetooth icon on the desktop. At this time, Bluetooth is not turned on, so a red **x** will be displayed

7) Click **Turn Bluetooth On** to turn on Bluetooth



8) The display after turning on Bluetooth is as follows



9) For the test method of Bluetooth, please refer to the section on **Bluetooth usage**, and will not repeat it here

3. 12. 3. 2. RTL8811 test

1) First insert the RTL8811 wireless network card module into the USB interface of the development board

2) Then the linux system will automatically load the kernel module related to RTL8811 WIFI, through the lsmod command, you can see that the following kernel module has been automatically loaded

```
orangepi@orangepi:~$ lsmod
Module                  Size  Used by
8821cu               1839104  0
```

3) Through the dmesg command, you can see the loading information of the RTL8811 module

```
orangepi@orangepi:~$ dmesg
[    118.618194] usb 2-1: new high-speed USB device number 2 using ehci-platform
[    118.767152] usb 2-1: New USB device found, idVendor=0bda, idProduct=c811, bcdDevice= 2.00
[    118.767181] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[    118.767199] usb 2-1: Product: 802.11ac NIC
[    118.767219] usb 2-1: Manufacturer: Realtek
[    118.767235] usb 2-1: SerialNumber: 123456
[    119.500530] usbcore: registered new interface driver rtl8821cu
[    119.525498] rtl8821cu 2-1:1.0 wlx1cbfced9d260: renamed from wlan0
```

4) Then you can see the WIFI device node through the **sudo ifconfig** command. For the WIFI connection and test method, please refer to the **WIFI connection test** section, and I won't go into details here

```
orangepi@orangepi:~$ sudo ifconfig wlx1cbfced9d260
wlx1cbfced9d260: flags=4099<UP,BROADCAST,MULTICAST>    mtu 1500
        ether 1c:bf:ce:d9:d2:60    txqueuelen 1000    (Ethernet)
        RX packets 0    bytes 0 (0.0 B)
        RX errors 0    dropped 0    overruns 0    frame 0
        TX packets 0    bytes 0 (0.0 B)
        TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
```

3. 12. 3. 3. RTL8821CU test

1) First insert the rtl8821cu wireless network card module into the usb interface of the development board

2) Then use the **lsusb** command to see the device information of the rtl8821cu usb wifi module, please make sure that the USB module is not in Driver CDROM Mode

```
orangepi@orangepi:~$ lsusb | grep "Realtek"
Bus 002 Device 003: ID 0bda:c820 Realtek Semiconductor Corp. 802.11ac NIC
```

```
orangepi@orangepi:~$ lsusb | grep "Realtek"
Bus 002 Device 002: ID 0bda:1a2b Realtek Semiconductor Corp. RTL8188GU 802.11n WLAN Adapter (Driver CDROM Mode)
```

If the USB WIFI module seen by the lsusb command is in Driver CDROM Mode, please unplug the USB WIFI module again. If not, please manually execute the following command to switch to the next mode:

```
orangepi@orangepi:~$ sudo usb_modeswitch -KW -v 0bda -p 1a2b
```

3) The linux system will automatically load the rtl8821cu bluetooth and wifi related kernel modules, through the lsmod command, you can see that the following kernel modules have been automatically loaded

```
orangepi@orangepi:~$ lsmod
Module                  Size  Used by
8821cu               1839104  0
rtk_btusb              61440  0
```

4) Through the dmesg command, you can see the loading information of the rtl8821cu module

```
orangepi@orangepi:~$ dmesg
......
[   57.083693] usb 2-1: new high-speed USB device number 2 using ehci-platform
[   57.231888] usb 2-1: New USB device found, idVendor=0bda, idProduct=1a2b, bcdDevice= 2.00
[   57.231916] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[   57.231937] usb 2-1: Product: DISK
[   57.231956] usb 2-1: Manufacturer: Realtek
[   57.242594] usb-storage 2-1:1.0: USB Mass Storage device detected
[   57.245674] scsi host0: usb-storage 2-1:1.0
[   58.069172] usb 2-1: USB disconnect, device number 2
[   58.440025] usb 2-1: new high-speed USB device number 3 using ehci-platform
[   58.587819] usb 2-1: New USB device found, idVendor=0bda, idProduct=c820, bcdDevice= 2.00
[   58.587827] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
```

```
[   58.587833] usb 2-1: Product: 802.11ac NIC
[   58.587838] usb 2-1: Manufacturer: Realtek
[   58.587844] usb 2-1: SerialNumber: 123456
[   58.610463] rtk_btusb: Realtek Bluetooth USB driver ver 3.1.6d45ddf.20220519-142432
[   58.610656] usbcore: registered new interface driver rtk_btusb
[   58.634631] Bluetooth: hci0: RTL: examining hci_ver=08 hci_rev=000c lmp_ver=08 lmp_subver=8821
[   58.636729] Bluetooth: hci0: RTL: rom_version status=0 version=1
[   58.636740] Bluetooth: hci0: RTL: loading rtl_bt/rtl8821c_fw.bin
[   58.664190] Bluetooth: hci0: RTL: loading rtl_bt/rtl8821c_config.bin
[   58.664746] Bluetooth: hci0: RTL: cfg_sz 10, total sz 31990
[   59.122471] Bluetooth: hci0: RTL: fw version 0x829a7644
[   59.265513] usbcore: registered new interface driver rtl8821cu
[   59.280119] rtl8821cu 2-1:1.2 wlx90de80521825: renamed from wlan0
```

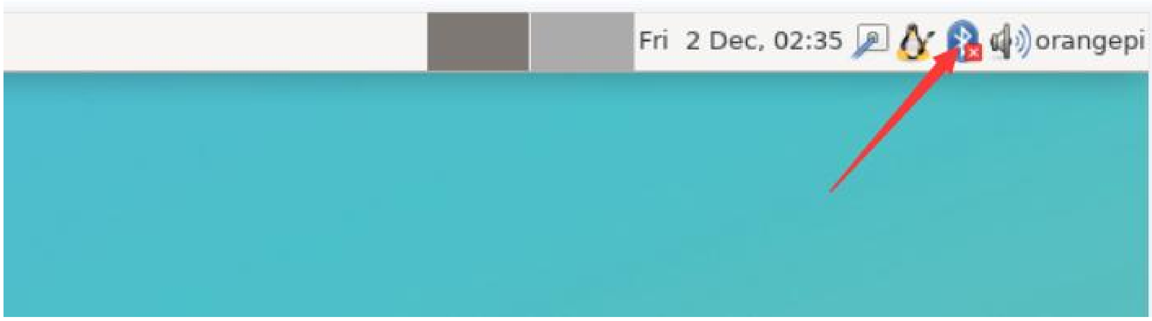
5) Then you can see the device node of rtl8821cu wifi through the **sudo ifconfig** command. For the wifi connection and test method, please refer to the section of **WIFI connection test**, so I won't go into details here

```
orangepi@orangepi:~$ sudo ifconfig wlx90de80521825
wlx90de80521825: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:13:ef:f4:58:ae  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

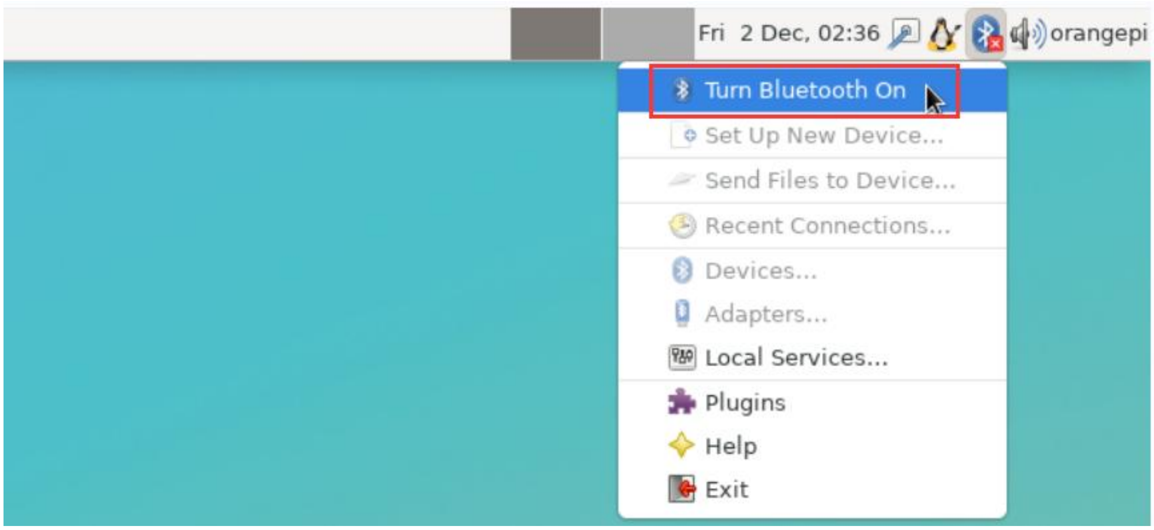
6) Then you can see the USB Bluetooth device through the **hciconfig** command

```
orangepi@orangepi:~$ sudo apt-get update && sudo apt-get install -y bluez
orangepi@orangepi:~$ hciconfig
hci0:    Type: Primary  Bus: USB
        BD Address: 00:13:EF:F4:58:AE  ACL MTU: 820:8  SCO MTU: 255:16
        DOWN
        RX bytes:1252 acl:0 sco:0 events:125 errors:0
        TX bytes:23307 acl:0 sco:0 commands:125 errors:0
```

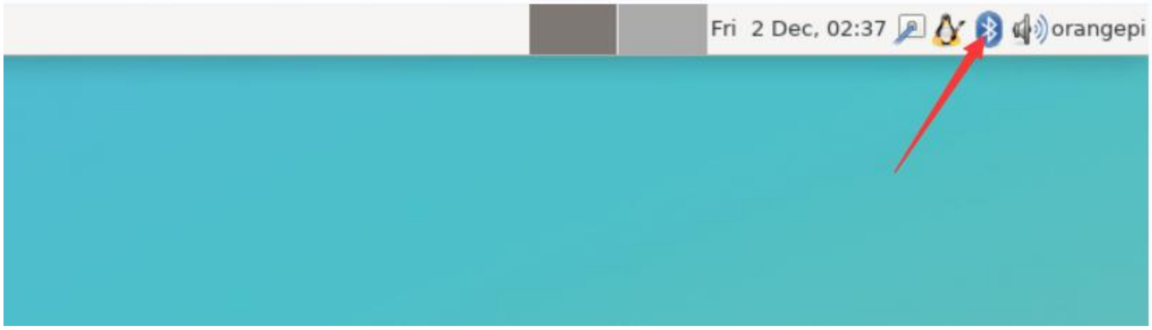
7) You can also see the bluetooth icon on the desktop. At this time, the bluetooth is not turned on, so a red **x** will be displayed



8) Click **Turn Bluetooth On** to turn on Bluetooth



9) The display after turning on Bluetooth is as follows



10) For the test method of Bluetooth, please refer to the section on **Bluetooth usage**, so I won't go into details here

3. 12. 4. USB camera test

1) First, you need to prepare a USB camera that supports the UVC protocol as shown in the figure below or similar, and then insert the USB camera into the USB port of the Orange Pi development board

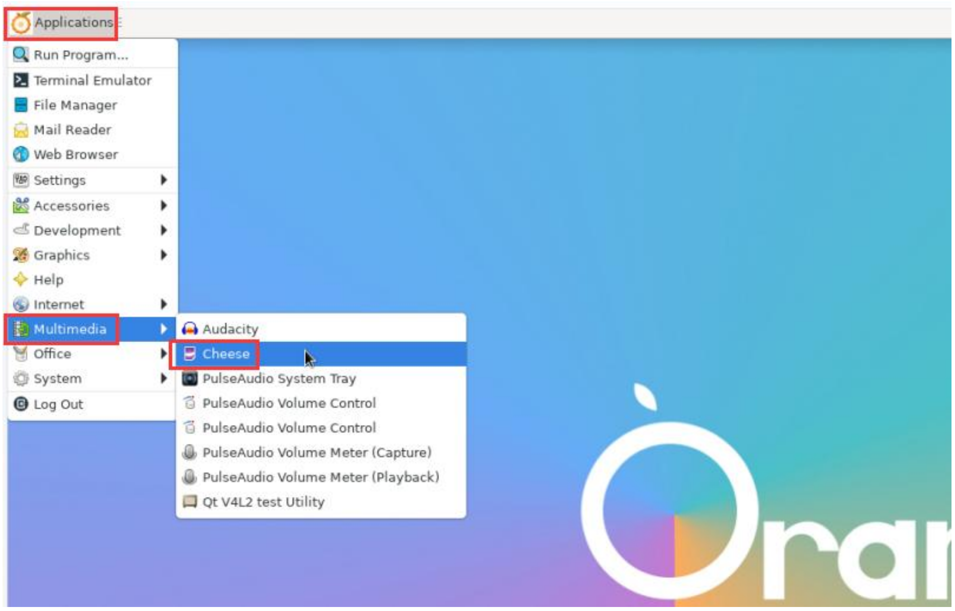
2) Through the v4l2-ctl command, you can see that the device node information of the USB camera is /dev/video0

```
orangepi@orangepi:~$ v4l2-ctl --list-devices
Q8 HD Webcam: Q8 HD Webcam (usb-fc880000.usb-1):
        /dev/video0
        /dev/video1
        /dev/media0
```

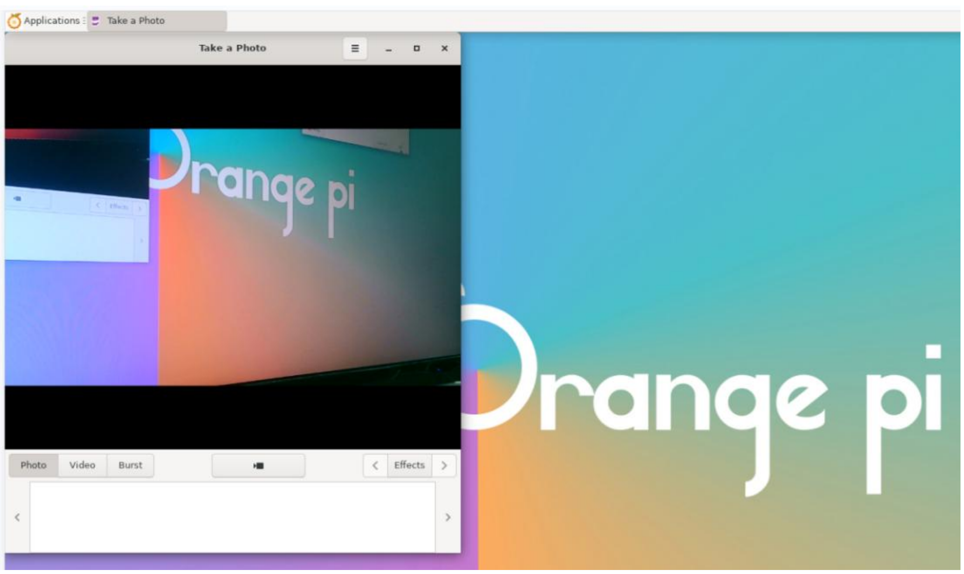
Note that the l in v4l2 is a lowercase letter l, not the number 1.

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

3) In the desktop system, Cheese can be used to directly open the USB camera. The method of opening Cheese is shown in the figure below:



The interface after Cheese turns on the USB camera is shown in the figure below:



4) Method of using fswebcam to test USB camera

a. Install fswebcam

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

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

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

b) --no-banner For removing watermarks from photos

c) -r Option to specify the resolution of the photo

d) -S Option to set the number of previous frames to skip

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

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

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

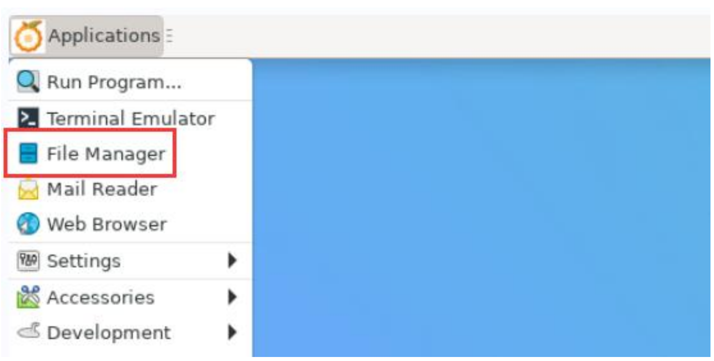
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test  （Modify the IP address and path according to the actual situation）
```

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

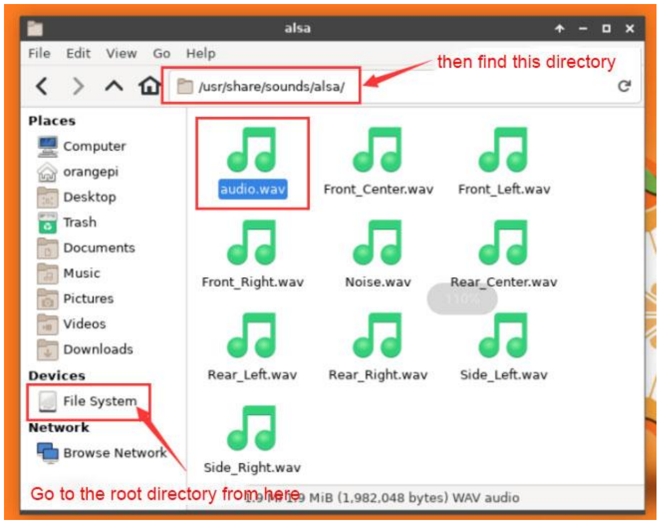
3. 13. Audio Test

3. 13. 1. Testing audio methods on desktop systems

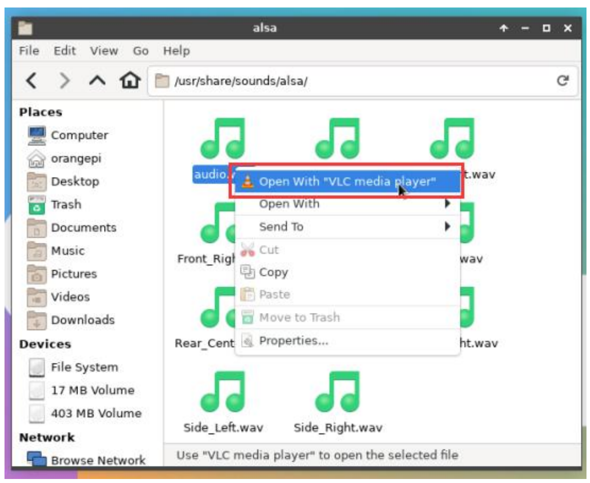
1) First open the file manager



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

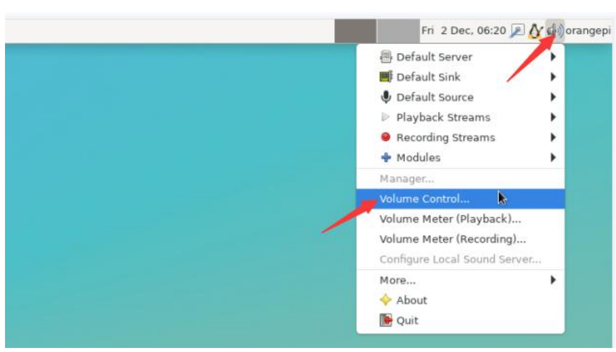


3) Then select the audio.wav file, right click and select open with vlc to start playing

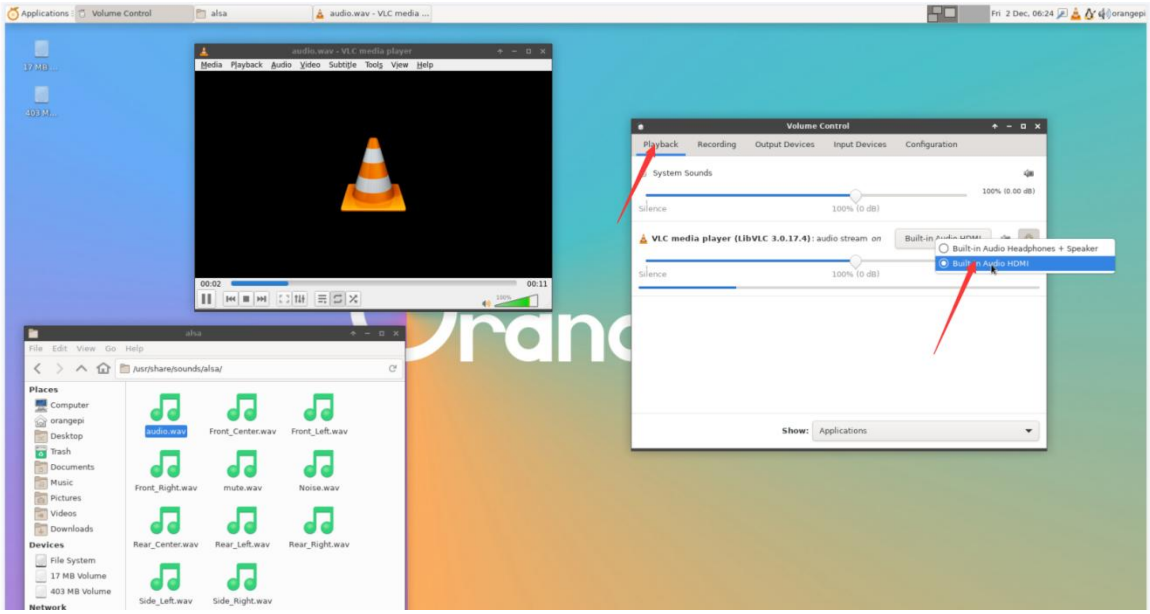


4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First open the volume control interface



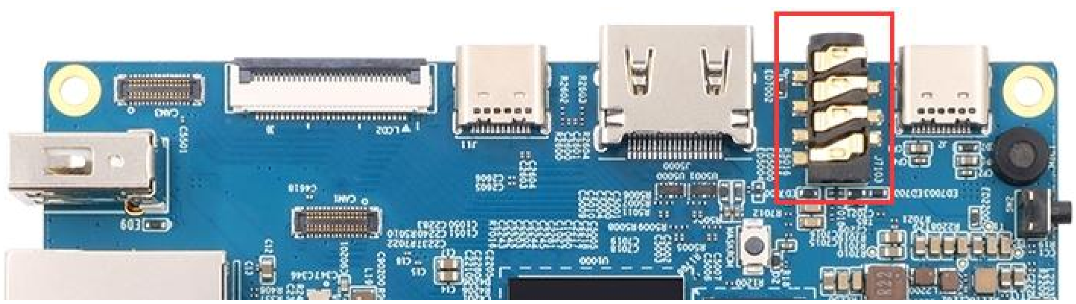
b. When playing audio, the audio device options that the playback software can use will be displayed in **Playback**, as shown in the figure below, where you can set which audio device to play to



3. 13. 2. The method of using commands to play audio

3. 13. 2. 1. Headphone interface playback audio test

1) First insert the earphone into the earphone jack of the development board



2) Then you can use the **aplay -l** command to view the sound card devices supported by the linux system. From the output below, we can see that **card 2** is the sound card device of es8388, that is, the sound card device of the headset

```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
```

```
    Subdevices: 1/1
    Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.6-0010-0 [dailink-multicodecs ES8323.6-0010-0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio file that comes with the system. If the earphone can hear the sound, it means that the hardware can be used normally.

```
orangepi@orangepi:~$ aplay -D hw:2,0 /usr/share/sounds/alsa/audio.wav
Playing WAVE 'audio.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3. 13. 2. 2. HDMI audio playback test

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

2) Then check the serial number of the HDMI sound card. From the output below, you can know that the HDMI sound card is **card 1**

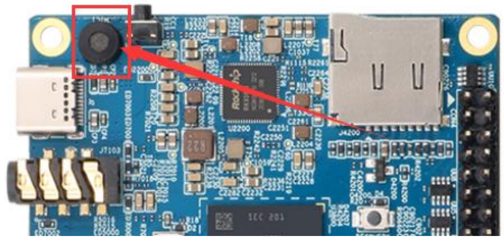
```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
card 1: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.6-0010-0 [dailink-multicodecs ES8323.6-0010-0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio file that comes with the system. If the HDMI monitor or TV can hear the sound, it means that the hardware can be used normally.

```
orangepi@orangepi:~$ aplay -D hw:1,0 /usr/share/sounds/alsa/audio.wav
```

3. 13. 3. Method of using commands to test recording

1) There is an onboard MIC on the development board, the location is as follows:



2) Running the **test_record.sh main** command will record a piece of audio through the onboard MIC, and then play it to HDMI and headphones.

```
orangepi@orangepi:~$ test_record.sh main
Start recording: /tmp/test.wav
Recording WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Start playing
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

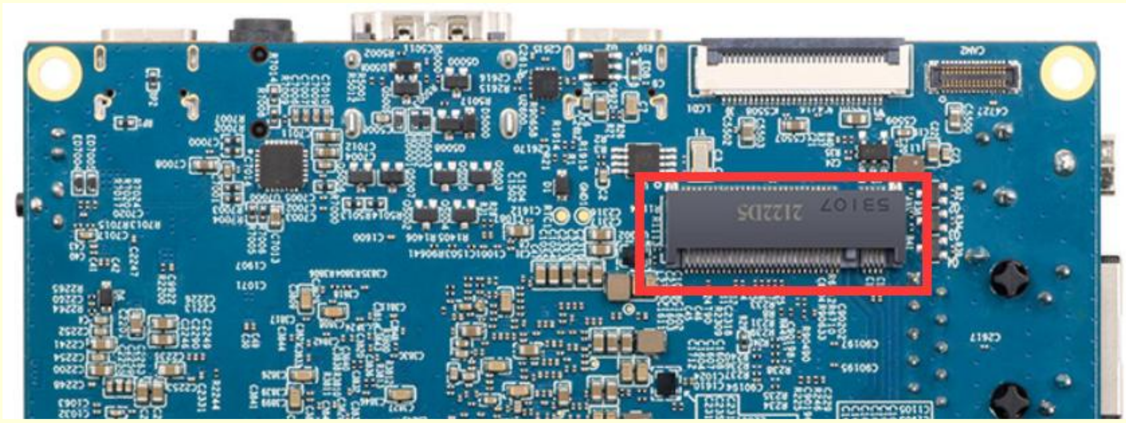
3) In addition to the onboard MIC, we can also record audio through headphones with MIC function. After inserting the headset with MIC function into the development board, run the **test_record.sh headset** command to record a piece of audio through the headset, and then play it to HDMI and the headset.

```
orangepi@orangepi:~$ test_record.sh headset
Start recording: /tmp/test.wav
Recording WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Start playing
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3. 14. How to use SATA SSD

The m.2 interface shown in the figure below can use both nvme ssd and sata ssd. Since the pcie2.0 controller and the sata controller are optional, only one of them can be configured at the same time. The linux image released by Orange Pi opens the pcie configuration by default, so it can only recognize nvme ssd by default. If you

want to use sata ssd, you need to open the corresponding configuration.

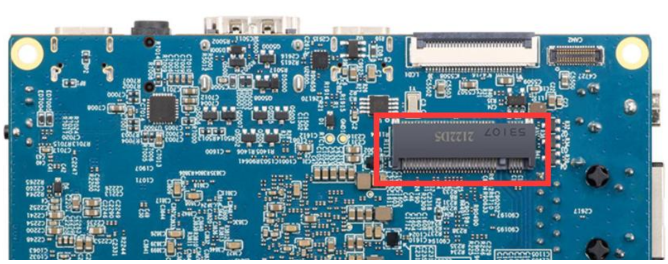


1) First, you need to prepare a SATA SSD solid state drive

   a. The M.2 2242 SSD is as follows

   b. The M.2 2280 specification SSD is as follows (2280 specification SATA SSD can also be used, but the SSD will exceed the development board after being inserted into the development board)

2) Then insert the SSD into the M.2 interface of the development board and fix it



3) There are two main usages of sata ssd:
   a. The linux system is in the TF card, and then insert the sata ssd as an external storage device. This section mainly explains this usage.
   b. Burn the linux system into the sata ssd, and then start the linux system in the sata ssd. For this kind of usage, please refer to the instructions in the section on **the method of burning the Linux image to SPIFlash+SATA SSD**.

4) After using the TF card to start the Linux system, we first burn the sata ssd-specific u-boot image into the TF card.
   a. The dedicated u-boot image storage path for sata ssd startup is:

```
/usr/share/orangepi5/u-boot-sata.itb
```

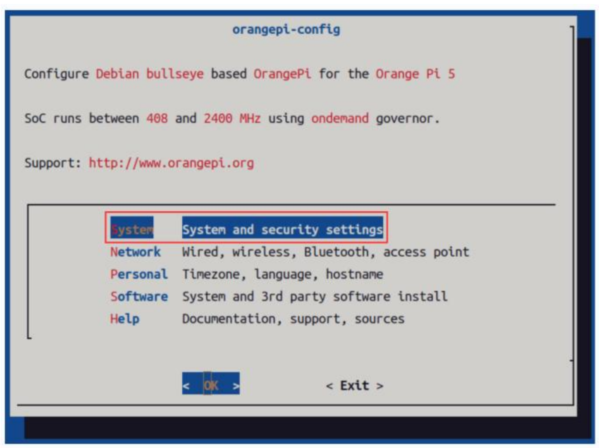
   b. Make sure that **u-boot-sata.itb** exists in the Linux system, and then use the following command to burn it to the TF card of the development board.

```
orangepi@orangepi:~$ cd /usr/share/orangepi5/
orangepi@orangepi:~$ sudo dd if=u-boot-sata.itb of=$(findmnt -n -o SOURCE / | sed 's/..$//')  seek=16384
orangepi@orangepi:~$ sudo sync
```

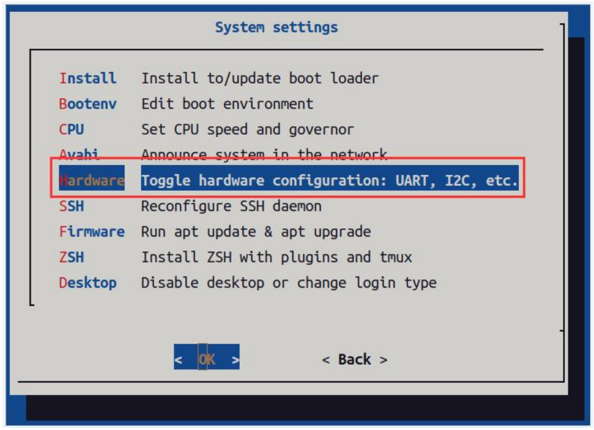
5) Then run orangepi-config. Ordinary users remember to add sudo permissions.

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

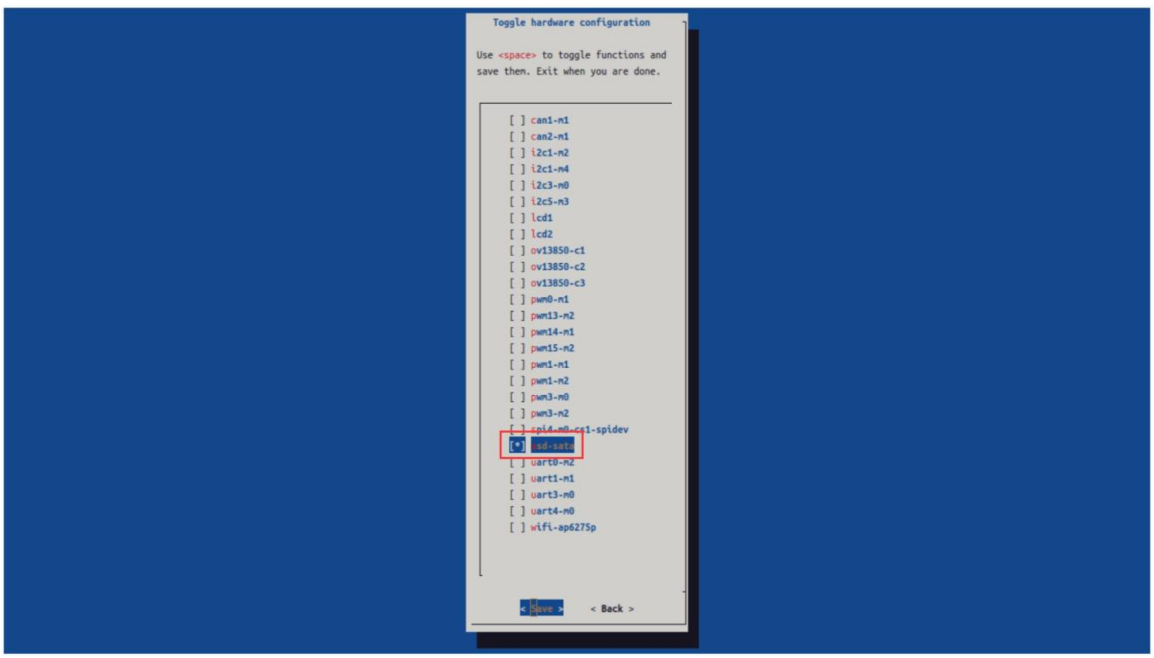
6) Then select **System**



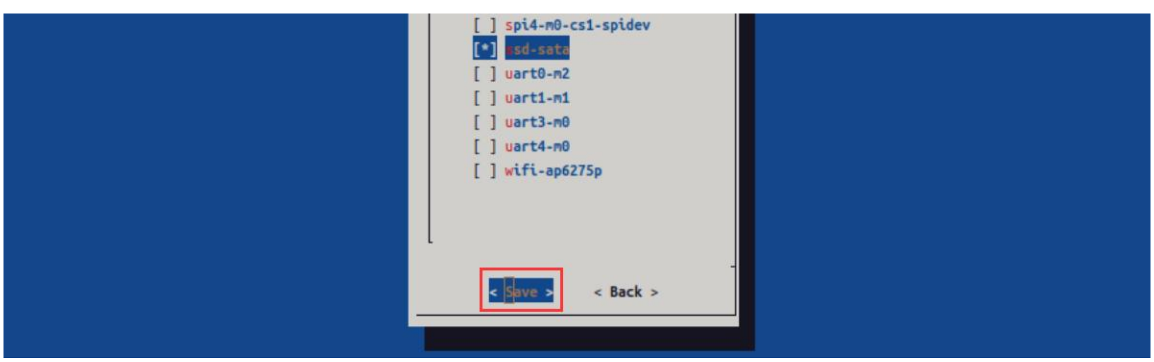
7) Then select **Hardware**



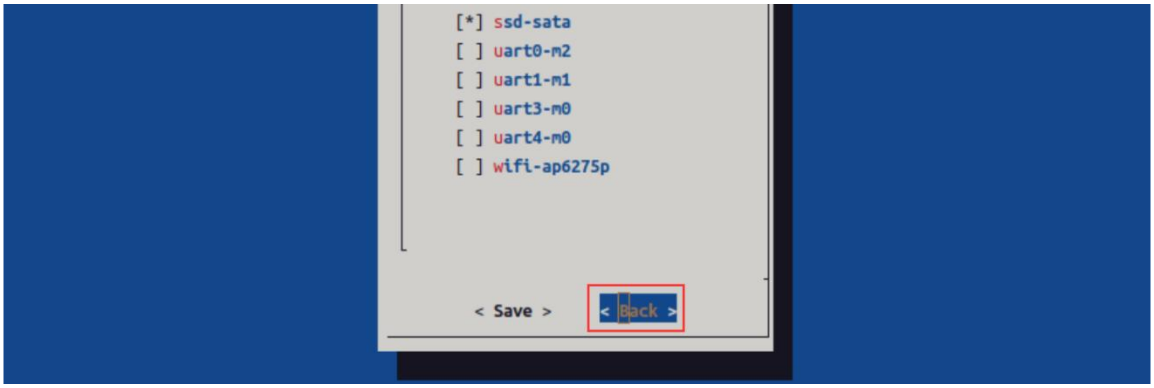
8) Then use the arrow keys of the keyboard to navigate to **ssd-sata**, and then use **the space** to select



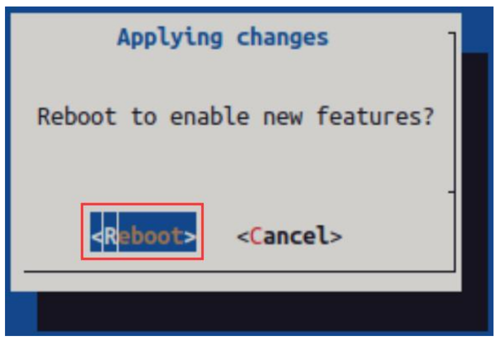
9) Then select **<Save>** to save



10) Then select **<Back>**



11) Then select **<Reboot>** to restart the system to make the configuration take effect



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

If you find it troublesome to use orangepi-config, you can also open /boot/orangepiEnv.txt, and then add the configuration of overlays=ssd-sata.

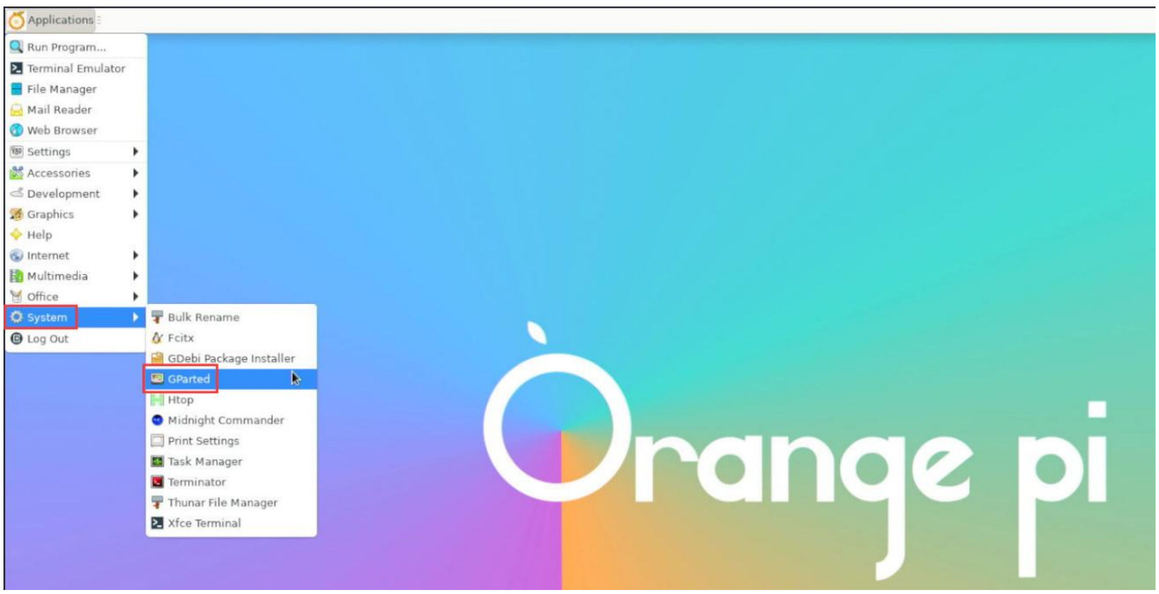
orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "ssd"**
overlays=ssd-sata

12) If everything is normal, after the system restarts, use the sudo fdisk -l command to see sata ssd information

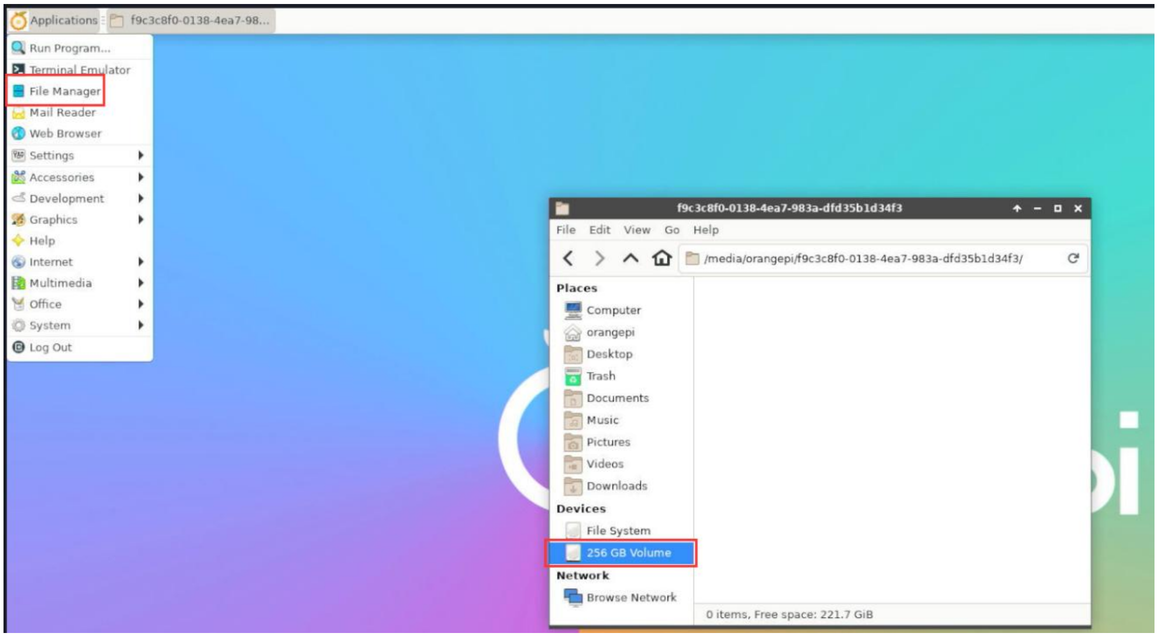
```
orangepi@orangepi:~$ sudo fdisk -l
.......
Disk /dev/sda: 238.47 GiB, 256060514304 bytes, 500118192 sectors
Disk model: Fanxiang S201 25
```

```
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 43FFB292-340D-654C-8C30-6C64AEDAA0F4

Device        Start       End     Sectors    Size Type
/dev/sda1     2048 500117503 500115456 238.5G Linux filesystem
.......
```

13) Then use **GParted** to format or partition sata ssd



14) Then you can see the sata ssd device in the file management



15) In the server version system, you can use the mount command to mount the sata ssd to the required directory

```
orangepi@orangepi:~$ sudo mount /dev/sda1 /mnt
orangepi@orangepi:~$ df -h
Filesystem        Size   Used Avail Use% Mounted on
udev              3.8G   8.0K  3.8G   1% /dev
tmpfs             769M   1.4M  768M   1% /run
/dev/mmcblk1p2     29G   5.9G   23G  21% /
tmpfs             3.8G      0  3.8G   0% /dev/shm
tmpfs             5.0M   4.0K  5.0M   1% /run/lock
tmpfs             3.8G    16K  3.8G   1% /tmp
/dev/mmcblk1p1    256M    90M  166M  36% /boot
/dev/zram1        194M    27M  154M  15% /var/log
tmpfs             769M    60K  769M   1% /run/user/1000
/dev/sda1         234G    28K  222G   1% /mnt
```

3. 15. Temperature sensor

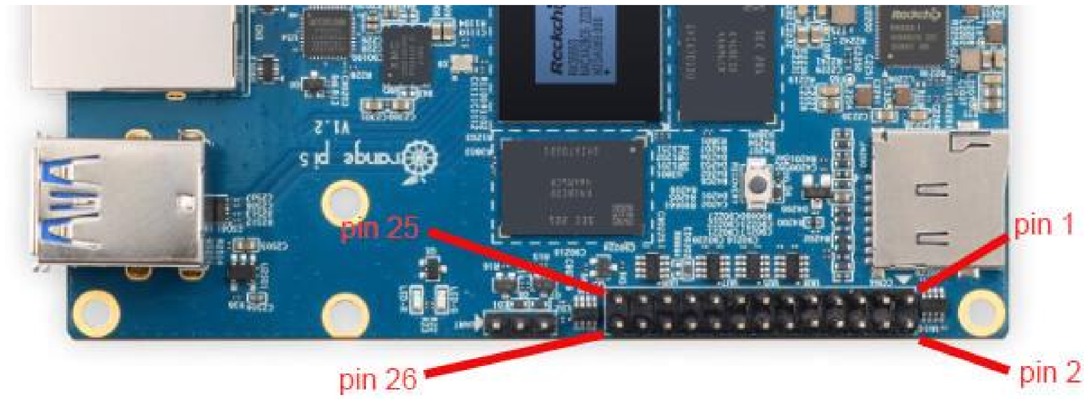
The command to view the system temperature sensor is:

```
orangepi@orangepi:~$ sensors
```

```
gpu_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

littlecore_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

bigcore0_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

tcpm_source_psy_6_0022-i2c-6-22
Adapter: rk3x-i2c
in0:           0.00 V  (min =  +0.00 V, max =  +0.00 V)
curr1:         0.00 A  (max =  +0.00 A)

npu_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

center_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

bigcore1_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

soc_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C  (crit = +115.0°C)
```

3. 16. 26 Pin Interface Pin Description

1) Please refer to the figure below for the order of the 26 pin interface pins on the Orange Pi 5 development board



2) The functions of the 26 pin interface pins on the Orange Pi 5 development board are shown in the table below

a. The following is the complete pin diagram of 26pin

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

b. The table below is the picture of the left half of the complete table above, so you can see it clearly

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
			3.3V		1
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7
			GND		9
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11
		CAN1_TX_M1	GPIO4_B3	139	13
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15
			3.3V		17
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23
			GND		25

c. The table below is the picture of the right half of the complete table above, so you can see it 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	131	GPIO4_A3	UART0_TX_M2（fd890000）		
10	132	GPIO4_A4	UART0_RX_M2		
12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
14		GND			
16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
20		GND			
22	92	GPIO2_D4			
24	52	GPIO1_C4	SPI4_CS1_M0		
26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

The pwm in the above table has marked the base address of the corresponding register, which is useful when checking which pwmchip in /sys/class/pwm/ corresponds to which pwm pin in the 26pin header.

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

3. 17. How to install wiringOP

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

The storage path of the compiled wiringOP deb package in orangepi-build is:

orangepi-build/external/cache/debs/arm64/wiringpi_x.xx.deb

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

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | IN     | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
root@orangepi5:~#
```

1) Download the code of wiringOP

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

Note that Orange Pi 5 needs to download the code of wiringOP next branch, please don't miss the -b next parameter.

If there is a problem downloading the code from GitHub, you can directly use the wiringOP source code that comes with the Linux image. The storage location is: /usr/src/wiringOP.

2) Compile and install wiringOP

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

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

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | ALT1   | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
root@orangepi5:~#
```

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

Note, if you need to set overlays to open multiple configurations at the same time, please use spaces to separate them and write them on one line as follows.

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

overlays=i2c1-m2 lcd1 ov13850-c1 pwm13-m2 spi4-m0-cs1-spidev uart0-m2

3. 18. 1. 26pin GPIO port test

The linux system released by Orange Pi has a pre-installed blink_all_gpio program, which will set all 17 GPIO ports in the 26pin to switch between high and low levels continuously.

After running the blink_all_gpio program, when using a multimeter to measure the level of the GPIO port, you will find that the GPIO pin will switch between 0 and 3.3v continuously. Using this program we can test whether the GPIO port is working properly.

The way to run the blink_all_gpio program is as follows:

orangepi@orangepi5:~$ **sudo blink_all_gpio** #Remember to add sudo permission

[sudo] password for orangepi: #A password is required here

1) There are a total of 17 GPIO ports in the 26pins of the development board that can be used. The following uses pin 7—the corresponding GPIO is GPIO1_C6——the corresponding wPi serial number is 2—as an example to demonstrate how to set the high

and low levels of the GPIO port

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
```

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

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

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

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

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

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |    OUT | 0 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
```

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

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

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

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |    OUT | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
```

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

3. 18. 2. 26pin GPIO port pull-down resistance setting method

Note that only the lower 4 GPIO pins of Orange Pi 5 can normally set the pull-up and pull-down resistor function, and the other GPIO pins have an external 3.3V pull-up, so setting the pull-down is invalid.

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 0 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | IN     | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
```

1) The following takes pin 11—the corresponding GPIO is GPIO4_B2—the corresponding wPi serial number is 5—as an example to demonstrate how to set the pull-up and pull-down resistance of the GPIO port

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |    OUT | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |    OUT | 0 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |    OUT | 0 |  7 || 8  | 0 | OUT    | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OUT    | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |    OUT | 0 | 11 || 12 | 0 | OUT    | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |    OUT | 0 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |    OUT | 0 | 15 || 16 | 0 | OUT    | SDA.1    | 9   | 59   |
```

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

root@orangepi:~/wiringOP# **gpio mode 5 in**

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

root@orangepi:~/wiringOP# **gpio mode 5 up**

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

root@orangepi:~/wiringOP# **gpio read 5**
1

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

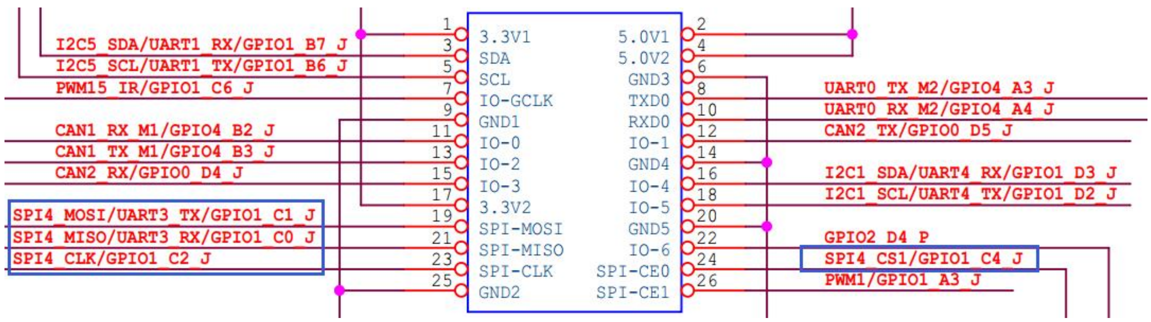
root@orangepi:~/wiringOP# **gpio mode 5 down**

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

root@orangepi:~/wiringOP# **gpio read 5**
0

3. 18. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the available spi for Orange Pi 5 is spi4



In the Linux system, the spi4 in the 26pin is closed by default, and it needs to be opened manually to use it.

Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open spi4.

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=spi4-m0-cs1-spidev

2) First check whether there is a **spidev4.1** device node in the linux system. If it exists, it means that SPI4 has been set up and can be used directly

```
orangepi@orangepi:~$ ls /dev/spidev4.1
/dev/spidev4.1
```

Note that /dev/spidev4.0 cannot be used, please use /dev/spidev4.1, don't make a mistake.

3) Do not short-circuit the mosi and miso pins of SPI4, the output result of running spidev_test is as follows, you can see that the data of TX and RX are inconsistent

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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 FF    | ................................
```

4) Then short-circuit the two pins of mosi (the 19th pin in the 26pin interface) and miso (the 21st pin in the 26pin interface) of SPI4, and then run the output of spidev_test as follows, you can see the sending and receiving same data

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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 FF F0 0D   | ......@....▒..................▒.
```

3. 18. 4. 26pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi 5 is i2c1, i2c3 and i2c5, a total of three sets of 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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

As can be seen from the above table, i2c1 can be derived from pins 12 and 15 of the 26pin (i2c1_m2), or from pins 16 and 18 of the 26pin (i2c1_m4), please follow your own needs Just select a group. Please don't think that these are two different sets of i2c buses.

In the linux system, the i2c in the 26pin is turned off by default, and it needs to be turned on manually before it can be used.

Add the following configuration in red font to /boot/orangepiEnv.txt, and then restart the Linux system to open i2c1, i2c3 and i2c5 at the same time. If you only need to open one, then just fill in one.

The settings to select i2c1_m2 are as follows:

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

overlays=i2c1-m2 i2c3-m0 i2c5-m3

The settings to select i2c1_m4 are as follows:

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

overlays=i2c1-m4 i2c3-m0 i2c5-m3

2) After starting the linux system, first confirm that there is an i2c device node under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-*
/dev/i2c-0    /dev/i2c-10    /dev/i2c-3    /dev/i2c-6    /dev/i2c-9
/dev/i2c-1    /dev/i2c-2     /dev/i2c-5    /dev/i2c-7
```

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

	i2c1-m2	i2c1-m4	i2c3-m0	i2c5-m3
sda pin	Corresponding to pin 12	Corresponding to pin 16	Corresponding to pin 21	Corresponding to pin 3
Sck pin	Corresponding to pin 15	Corresponding to pin 18	Corresponding to pin 19	Corresponding to pin 5
3.3v pin	Corresponding to pin 1	Corresponding to pin 1	Corresponding to pin 1	Corresponding to pin 1
5v pin	Corresponding to pin 2	Corresponding to pin 2	Corresponding to pin 2	Corresponding to pin 2
gnd pin	Corresponding to pin 6	Corresponding to pin 6	Corresponding to pin 6	Corresponding to pin 6

Generally, only one 3.3v pin and 5v pin can be connected, please choose to connect 3.3v pin or 5v pin according to the specific i2c device connected.

4) 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's command
orangepi@orangepi:~$ **sudo i2cdetect -y 3** #i2c3's command
orangepi@orangepi:~$ **sudo i2cdetect -y 5** #i2c5's command

```
orangepi@orangepi5:~$ sudo i2cdetect -y 5
[sudo] password for orangepi:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
orangepi@orangepi5:~$
```

3. 18. 5. 26pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5 are uart0, uart1, uart3 and uart4, a total of four sets of uart 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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

In the Linux system, the uart in the 26pin is closed by default, and it needs to be opened manually before it can be used.

Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open uart0, uart1, uart3 and uart4 at the same time. If you only need to open one, then just fill in one.

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=uart0-m2 uart1-m1 uart3-m0 uart4-m0

2) After entering the linux system, first confirm whether there is a device node corresponding to uart under /dev

```
orangepi@orangepi:~$ ls /dev/ttyS*
/dev/ttyS0   /dev/ttyS1   /dev/ttyS3   /dev/ttyS4   /dev/ttyS9
```

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

	uart0	uart1	uart3	uart4
tx pin	Corresponding to pin 8	Corresponding to pin 5	Corresponding to pin 19	Corresponding to pin 18
rx pin	Corresponding to pin 10	Corresponding to pin 3	Corresponding to pin 21	Corresponding to pin 16

4) 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

a. Test UART0

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS0
[sudo] password for orangepi:    #Enter password here

Out:     0:   ->     0
Out:     1:   ->     1
Out:     2:   ->     2
Out:     3:   ->     3
Out:     4:   ->     4
Out:     5:   ->     5^C
```

b. Test UART1

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS1
[sudo] password for orangepi:    #Enter password here

Out:     0:   ->     0
Out:     1:   ->     1
```

```
Out:    2:  ->   2
Out:    3:  ->   3
Out:    4:  ->   4
Out:    5:  ->   5^C
```

c. Test UART3

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS3
[sudo] password for orangepi:   #Enter password here

Out:    0:  ->   0
Out:    1:  ->   1
Out:    2:  ->   2
Out:    3:  ->   3
Out:    4:  ->   4
Out:    5:  ->   5^C
```

d. Test UART4

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS4
[sudo] password for orangepi:   #Enter password here

Out:    0:  ->   0
Out:    1:  ->   1
Out:    2:  ->   2
Out:    3:  ->   3
Out:    4:  ->   4
Out:    5:  ->   5^C
```

3. 18. 6. How to test PWM using /sys/class/pwm

1) As can be seen from the table below, the available pwm for Orange Pi 5 includes pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15, a total of six pwm

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

As can be seen from the table above:

pwm1 can be derived from pin 16 of 26pin (pwm1_m1), or from pin 26 of 26pin (pwm1_m2)

pwm3 can be derived from pin 15 of 26pin (pwm3_m0), or from pin 23 of 26pin (pwm3_m2)

Please select the corresponding pin according to your needs. Please don't think that these are two different pwm buses.

In the linux system, the pwm in the 26pin is closed by default, and it needs to be opened manually to use it.

Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open pwm0, pwm13, pwm14 and pwm15 at the same time. If you only need to open one, then fill in one.

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=pwm0-m1 pwm13-m2 pwm14-m1 pwm15-m2

Select the settings of pwm1_m1 as follows, please do not open pwm1-m1 and pwm1-m2 at the same time:
orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=pwm1-m1

The settings to select pwm1_m2 are as follows:
orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=pwm1-m2

Select the settings of pwm3_m0 as follows, please do not open pwm3-m0 and pwm3-m2 at the same time:
orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=pwm3-m0

The settings to select pwm3_m2 are as follows:
orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=pwm3-m2

2) When a pwm is turned on, there will be an extra pwmchipX in **/sys/class/pwm/** (X is a

specific number), for example, after turning on pwm15, check the pwmchipX under **/sys/class/pwm/** two became three

```
orangepi@orangepi:~$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

3) Which pwmchip above corresponds to pwm15? Let's check the output of the **ls /sys/class/pwm/ -l** command first, as shown below:

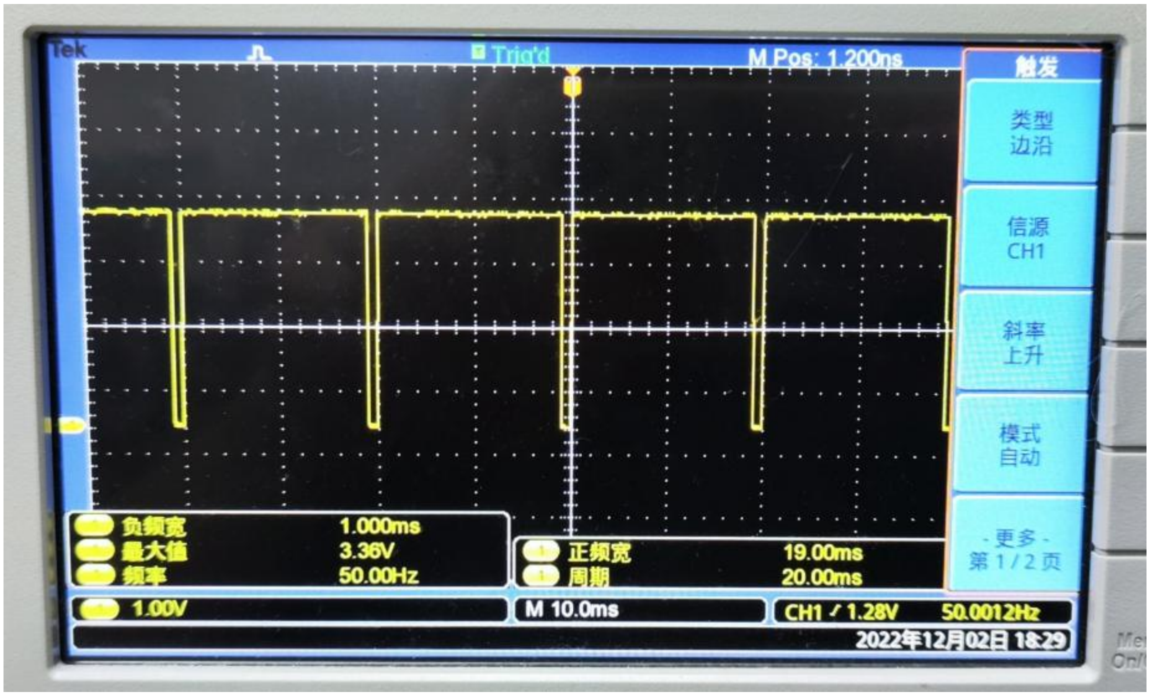
```
orangepi@orangepi5:~$ ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Dec  2 10:20 pwmchip0 -> ../../devices/platform/fd8b0020.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Dec  2 10:20 pwmchip1 -> ../../devices/platform/febd0020.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Dec  2 10:20 pwmchip2 -> ../../devices/platform/febf0030.pwm/pwm/pwmchip2
orangepi@orangepi5:~$
```

4) Then it can be known from the table below that the base address of the pwm15 register is febf0030, and then look at the output of the **ls /sys/class/pwm/ -l** command, you can see that pwmchip2 is linked to febf0030.pwm, so pwm15 corresponds to pwmchip as pwmchip2

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

5) Then use the following command to make pwm15 output a 50Hz square wave (please switch to the root user first, and then execute the following command)

```
root@orangepi:~# echo 0 > /sys/class/pwm/pwmchip2/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip2/pwm0/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip2/pwm0/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip2/pwm0/enable
```



6) The test method of pwm15 demonstrated above is similar to other pwm test methods.

3. 18. 7. CAN test method

Please note that Linux 6.1 systems currently do not support CAN function.

3. 18. 7. 1. How to open CAN

1) As can be seen from the table below, the available CAN bus for Orange Pi 5 is CAN1 and CAN2, two sets of CAN bus

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

In the Linux system, the CAN in the 26pin is closed by default, and it needs to be opened manually to use it.

Add the following configuration in red font to /boot/orangepiEnv.txt, and then restart the Linux system to open CAN1 and CAN2 at the same time. If you only

need to open one, then just fill in one.

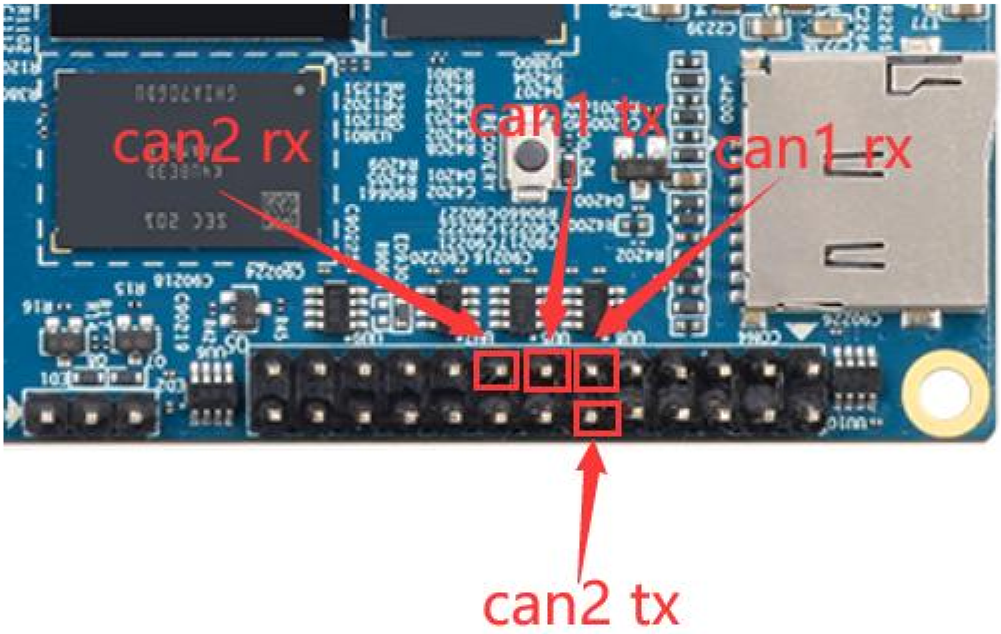
orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
overlays=can1-m1 can2-m1

2) After entering the linux system, use the **sudo ifconfig -a** command, if you can see the CAN device, it means that the CAN has been opened correctly

```
orangepi@orangepi:~$ sudo ifconfig -a
can0: flags=128<NOARP>    mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00    txqueuelen 10    (UNSPEC)
        RX packets 0    bytes 0 (0.0 B)
        RX errors 0    dropped 0    overruns 0    frame 0
        TX packets 0    bytes 0 (0.0 B)
        TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
        device interrupt 91

can1: flags=128<NOARP>    mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00    txqueuelen 10    (UNSPEC)
        RX packets 0    bytes 0 (0.0 B)
        RX errors 0    dropped 0    overruns 0    frame 0
        TX packets 0    bytes 0 (0.0 B)
        TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
        device interrupt 92
```

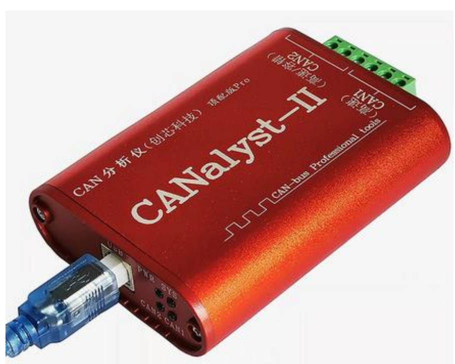
3) The corresponding pins of CAN1 and CAN2 are

	CAN1	CAN2
TX pin	Corresponding to pin 13	Corresponding to pin 12
RX pin	Corresponding to pin 11	Corresponding to pin 15



3.18.7.2. Use the CANalyst-II analyzer to test sending and receiving messages

1) The CANalyst-II analyzer used in the test is shown in the figure below



2) CANalyst-II analyzer data download link

https://www.zhcxgd.com/3.html

3) First install the software USBCANToolSetup

4) The shortcut after installation of USBCANToolSetup is

5) In addition, you need to install the USB driver

6) The end of the USB interface of the CANalyst-II analyzer needs to be connected to the USB interface of the computer

7) To test the CAN function, you need to prepare a CAN transceiver as shown in the figure below. The main function of the CAN transceiver is to convert the TTL signal of the CAN controller into the differential signal of the CAN bus

a. The 3.3V pin of the CAN transceiver needs to be connected to the 3.3V pin in the 26pin of the development board

b. The GND pin of the CAN transceiver needs to be connected to the GND pin in the 26pin of the development board

c. The CAN TX pin of the CAN transceiver needs to be connected to the TX pin of the CAN bus in the development board 26pin

d. The CAN RX pin of the CAN transceiver needs to be connected to the RX pin of the CAN bus in the development board 26pin

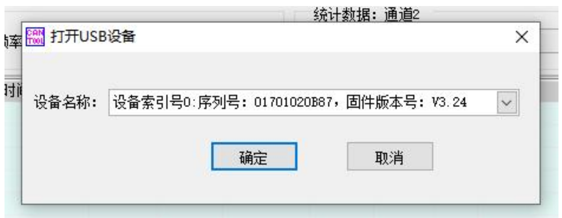
e. The CANH pin of the e.CAN transceiver needs to be connected to the H interface of the analyzer

f. The CANL pin of the CAN transceiver needs to be connected to the L interface of the analyzer

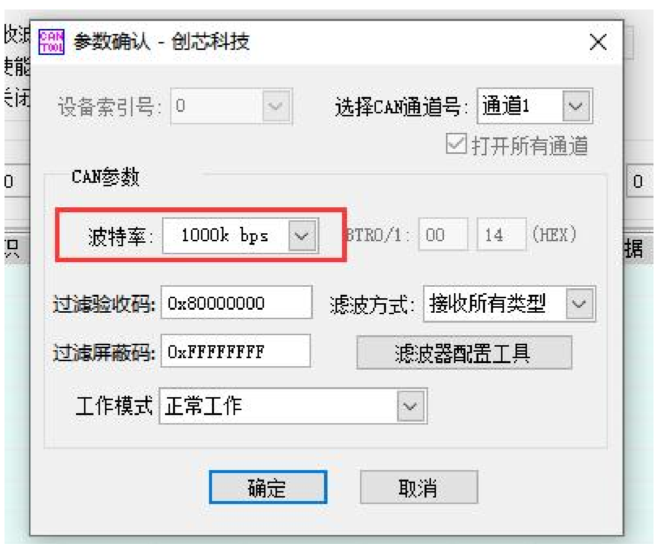
8) Then you can open the USB-CAN software

9) Then click to start the device

10) Then click OK



11) Set the baud rate to 1000k bps



12) After successful opening, the USB-CAN software will display the serial number and other information

13) Development board receives CAN message test

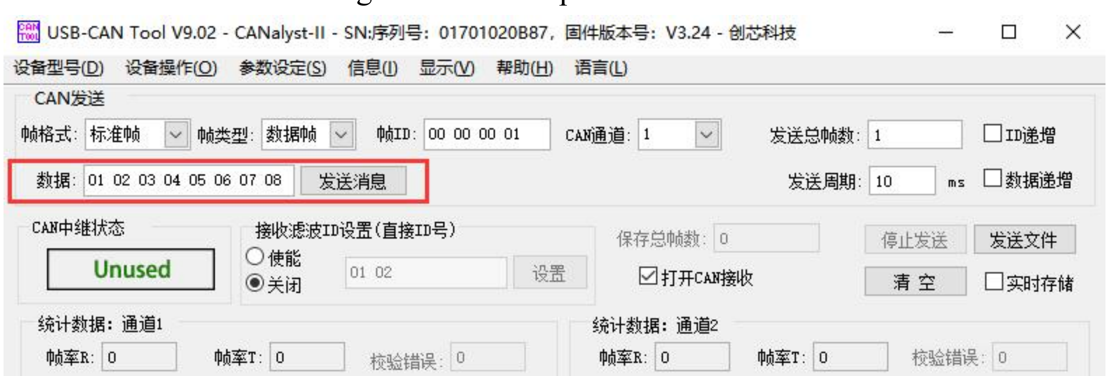
a. First set the baud rate of the CAN bus to **1000kbps** in the Linux system of the development board

```
orangepi@orangepi:~$ sudo ip link set can0 down
orangepi@orangepi:~$ sudo ip link set can0 type can bitrate 1000000
orangepi@orangepi:~$ sudo ip link set can0 up
```

b. Then run the **candump can0** command to prepare to receive messages

```
orangepi@orangepi:~$ sudo candump can0
```

c. Then send a message to the development board in the USB-CAN software



d. If the message sent by the analyzer can be received in the development board, it means that the CAN bus can be used normally

```
orangepi@orangepi5:~$ sudo candump can0
  can0  001   [8]  01 02 03 04 05 06 07 08
```

14) Development board sends CAN message test

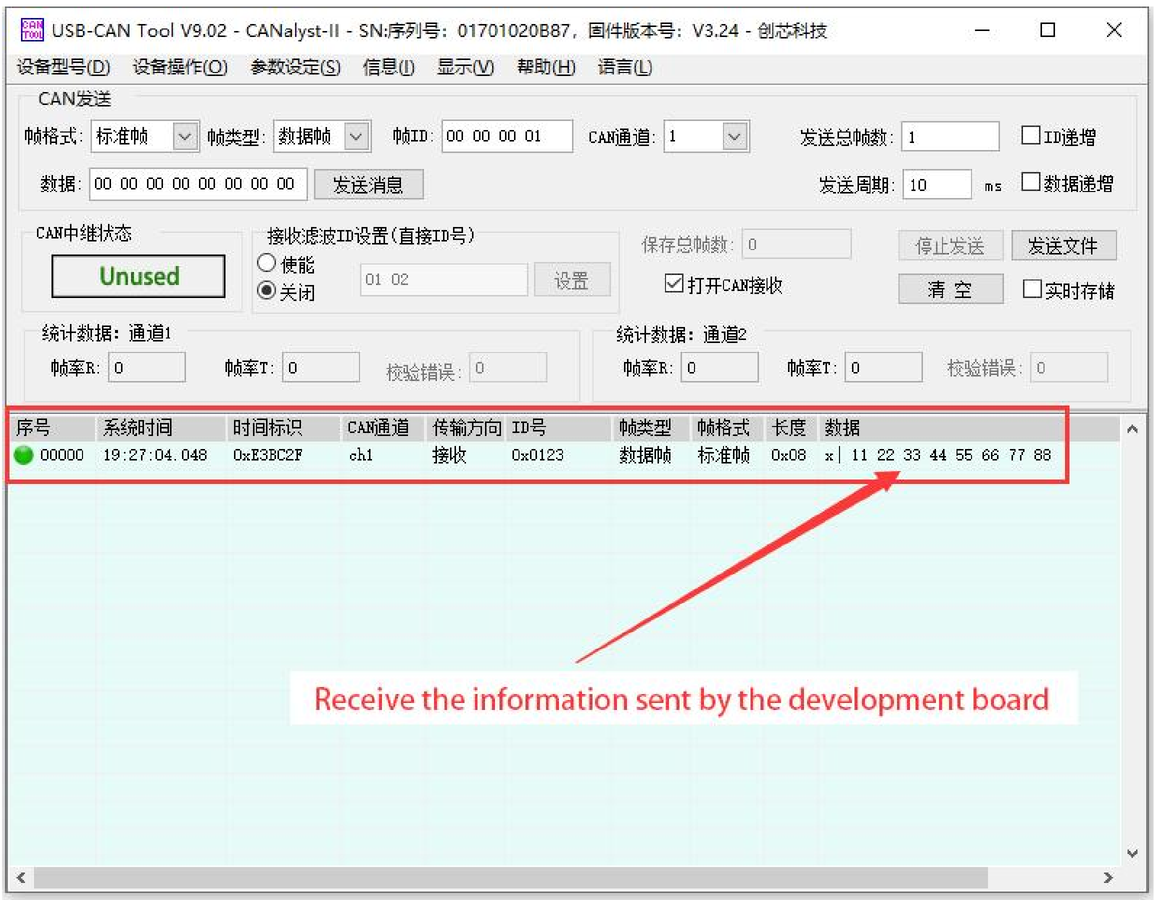
a. First set the CAN baud rate to **1000kbps** in the Linux system

```
orangepi@orangepi:~$ sudo ip link set can0 down
orangepi@orangepi:~$ sudo ip link set can0 type can bitrate 1000000
orangepi@orangepi:~$ sudo ip link set can0 up
```

b. Execute the **cansend** command on the development board and send a message

```
orangepi@orangepi:~$ sudo cansend can0 123#1122334455667788
```

c. If the USB-CAN software can receive the message from the development board, the communication is successful

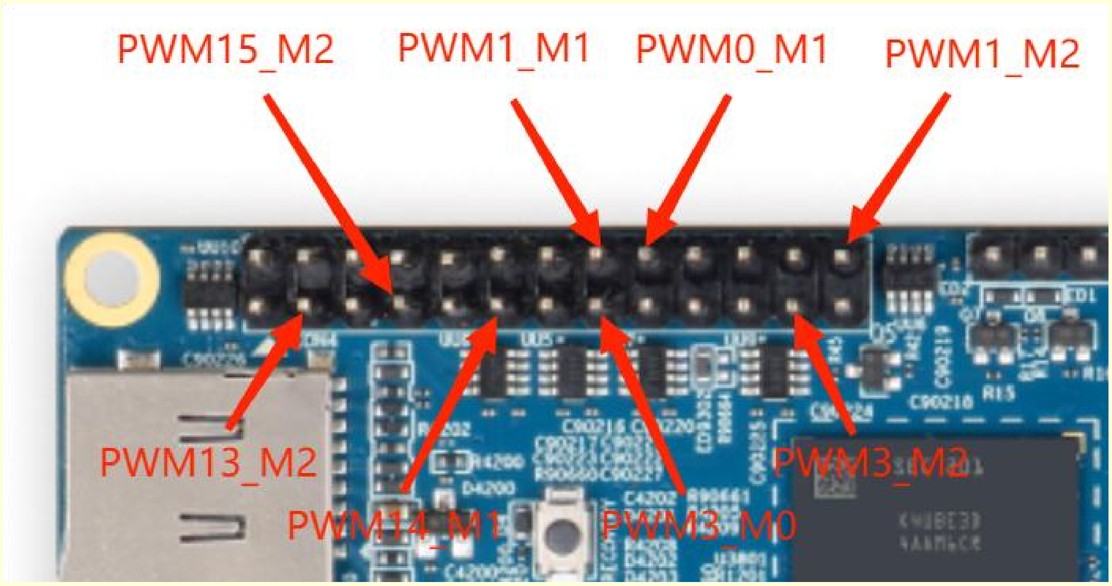


3.19. WiringOP Hardware PWM Usage Method

Before using wiringOP to operate PWM, please ensure that wiringOP is installed on the Linux system. If the gpio readall command can be used normally, it means wiringOP has been installed. If prompted that the command cannot be found, please refer to the instructions in the section on installing wiringOP to install

wiringOP first.

The development board can use six pwm channels: pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15. The positions of these pins are shown in the figure below:



It should be noted that: PWM1_M1 and PWM1_M2,
PWM3_M0 and PWM3_M2
They are all the same PWM, just connected to different pins. If both pin are configured for PWM function, when setting one PWM pin, the same setting will also be applied to the other PWM pin.

3. 19. 1. Method of setting PWM using the gpio command of wiringOP

3. 19. 1. 1. Set the corresponding pin to PWM mode

1) As shown in the table below, there are six pwm channels available on the development board: pwm0, pwm1, pwm3, 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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

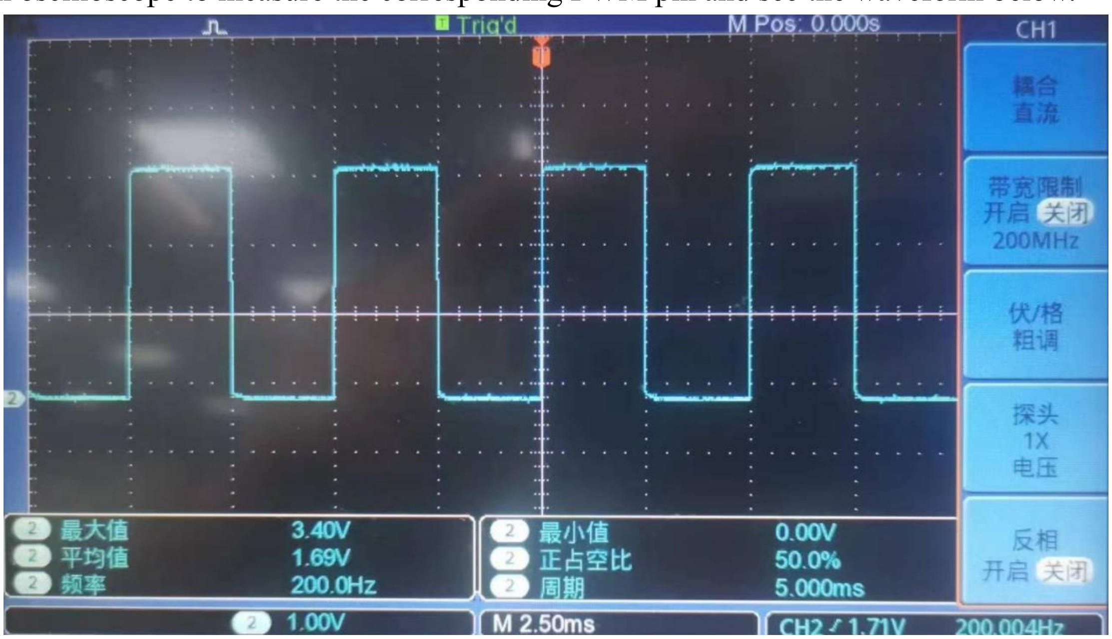
2) The wPi numbers corresponding to the PWM pins are as follows:

PWM Pin	wPi Serial number	Pin Serial number	GPIO Serial number
PWM0_M1	10	18	58
PWM1_M1	9	16	59
PWM1_M2	16	26	35
PWM3_M0	8	15	28
PWM3_M2	14	23	50
PWM13_M2	0	3	47
PWM14_M1	5	11	138
PWM15_M2	2	7	54

3) The command to set the pin to PWM mode is as follows. Taking PWM0_S1 as an example, the third parameter requires inputting the sequence number of wPi corresponding to the PWM0_S1 pin.

```
orangepi@orangepi:~$ gpio mode 10 pwm
```

4) After setting the pin to PWM mode, a square wave with a frequency of 200Hz, a period of 5ms, and a duty cycle of 50% will be output by default. At this time, we can use an oscilloscope to measure the corresponding PWM pin and see the waveform below.



3.19.1.2. Methods for Adjusting PWM Frequency

The calculation formula for PWM frequency is as follows:

PWM frequency = clock frequency / (division factor * The value of the periodic register)

IN:

1. The default value for clock frequency is 24000000Hz。

2. The range of values for the frequency division coefficient is even numbers between 2 ~ 512, with a default value of 120. If the set frequency division coefficient is odd, the actual frequency division coefficient is the set value minus one.

3. The default value of the cycle register is 1000。

4. The default value for PWM frequency is 24000000 / (120 * 1000) = 200Hz。

3.19.1.2.1. Method of Adjusting PWM Frequency by Setting Frequency Division Coefficient

1) We can use the following command to set the division factor of the PWM0_M1 pin to 4.

```
orangepi@orangepi:~$ gpio pwmc 10 4
```

2) According to the formula above, the calculated value of PWM frequency is 6000Hz. Through an oscilloscope, it can be observed that the measured PWM frequency is 6010Hz, and the error can be ignored.

3. 19. 1. 2. 2. Method of directly setting PWM frequency

1) We can use the **gpio pwmTone** command to set the frequency of the PWM pin, for example, the following command can be used to set the PWM frequency of the PWM0_M1 pin to 500Hz.

```
orangepi@orangepi:~$ gpio pwmTone 10 500
```

When setting the PWM frequency, it is necessary to ensure that:

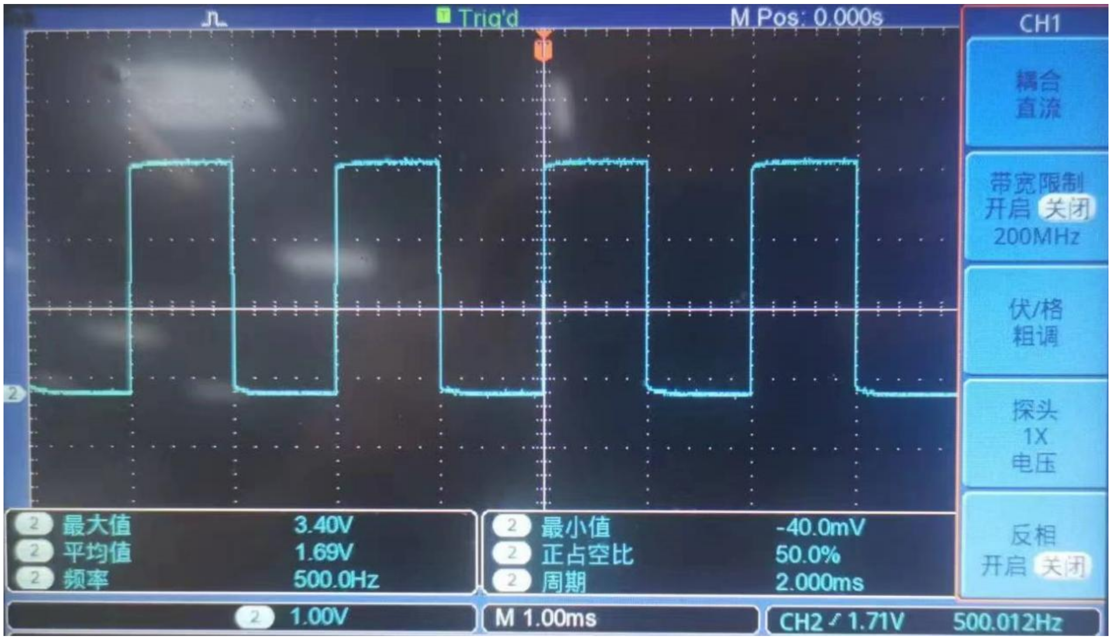
Set frequency value < 24000000 / (division factor * 2)。

For example, the default division factor is 120, and without modifying the division factor, the set frequency value should be less than 100000Hz.

If the setting value is too large, the following error message will appear:

gpio: The PWM frequency you set is too high to be possible

2) Then, through an oscilloscope, it can be observed that the PWM frequency has changed to 500Hz.



3. 19. 1. 3. Methods for Adjusting PWM Duty Cycle

1) The calculation formula for PWM duty cycle is as follows. We can adjust the PWM duty cycle by setting the values of the duty cycle register and the period register.

PWM Duty cycle = The value of the duty cycle register / The value of the periodic register

IN:

The default value of the duty cycle register is 500.

The default value of the cycle register is 1000.

It should be noted that the value of the duty cycle register needs to be smaller than the value of the cycle register, as the duty cycle cannot be greater than 1.

When the value of the duty cycle register is set to be greater than the value of the cycle register, the following error message will be prompted:

gpio: CCR should be less than or equal to ARR (XXX)

When the value of the cycle register is set to be less than the value of the duty cycle register, the following error message will be prompted:

gpio: ARR should be greater than or equal to CRR (XXX)

2) We can use the following command to set the value of the period register for the PWM0_M1 pin to 2000.

```
orangepi@orangepi:~$ gpio pwmr 10 2000
```

3) After running the above command, it can be observed through the oscilloscope that the PWM duty cycle has changed from the default 50% (500/1000) to 25% (500/2000).

4) We can use the following command to set the duty cycle register value of the PWM0_M1 pin to 1000.

```
orangepi@orangepi:~$ gpio pwm 10 1000
```

5) After running the above command, it can be observed through the oscilloscope that the PWM duty cycle changes from 25% (500/2000) to 50% (1000/2000).

3. 19. 2. Usage of PWM Test Program

1) In the example directory of wiringOP, there is a program called pwm.c that demonstrates the use of PWM related API in wiringOP to operate PWM.

```
orangepi@orangepi:~$ cd /usr/src/wiringOP/examples/
orangepi@orangepi:/usr/src/wiringOP/examples$ ls pwm.c
pwm.c
```

2) The command to compile **pwm.c** into an executable program is as follows:

```
orangepi@orangepi:/usr/src/wiringOP/examples$ gcc -o pwm pwm.c -lwiringPi
```

3) Then you can execute the PWM test program. When executing the PWM test program, you need to specify the PWM pin. For example, you can use the following command to test the PWM0_M1 pin:

```
orangepi@orangepi:/usr/src/wiringOP/examples$ sudo ./pwm 10
```

4) After the pwm program is executed, the following contents will be tested sequentially:

a. Adjust the PWM duty cycle by setting the value of the cycle register.
b. Adjust the PWM duty cycle by setting the value of the duty cycle register.
c. Adjust the PWM frequency by setting the division factor.
d. Directly set the PWM frequency.

5) After completing each test, the output of pwm waveform will be stopped for 5 seconds. After completing all test contents, a new round of testing will be restarted.

6) The detailed execution process of the PWM test program is as follows:

a. By setting the value of the cycle register to adjust the PWM duty cycle: Through an oscilloscope, it can be observed that the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 50% to 25% and remains for 5 seconds. Then, the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 25% to 50% and remains for 5 seconds.
b. By setting the value of the duty cycle register to adjust the PWM duty cycle: The oscilloscope can observe that the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 50% to 100% and remains for 5 seconds. Then, the PWM waveform changes every 0.5 seconds. After 8 changes, the PWM duty cycle changes from 100% to 50% and remains for 5 seconds.
c. By setting the frequency division coefficient to adjust the PWM frequency: Through an oscilloscope, it can be observed that the PWM waveform changes every 0.5 seconds. After 9 changes, the PWM frequency will change from 2000Hz to 200Hz and remain for 5 seconds. Then, the PWM waveform changes every 0.5 seconds. After 9 changes, the PWM frequency will change again to 2000Hz and remain for 5 seconds.
d. Directly setting the PWM frequency: Through the oscilloscope, it can be observed that the PWM frequency first changes to 2000Hz, and then increases by 2000Hz every two seconds. After 9 changes, the PWM frequency changes to 20000Hz and remains for 5 seconds.

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

wiringOP-Python is the Python language version of wiringOP, which is used to operate the hardware resources of the development board, such as GPIO, I2C, SPI and UART, in the Python program.

In addition, please note that all the following commands are operated under the

root user.

3. 20. 1. How to install wiringOP-Python

1) First install the dependency package

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python

Note that the following git clone --recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on wiringOP. Please make sure that the download process does not report errors due to network problems.

If there is a problem downloading the code from GitHub, you can directly use the wiringOP-Python source code that comes with the Linux image. The storage location is: /usr/src/wiringOP-Python.

```
root@orangepi:~# git clone --recursive https://github.com/orangepi-xunlong/wiringOP-Python -b next
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# git submodule update --init --remote
```

3) Then use the following command to compile wiringOP-Python and install it into the Linux system of the development board

```
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# python3 generate-bindings.py > bindings.i
root@orangepi:~/wiringOP-Python# sudo python3 setup.py install
```

4) Then enter the following command, if there is help information output, it means that wiringOP-Python is installed successfully, press the **q** key to exit the help information interface

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi;    help(wiringpi)"
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
```

```
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.
```

5) The steps to test whether wiringOP-Python is successfully installed under the python command line are as follows:

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

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi;
```

c. Finally, enter the following command to view the help information of wiringOP-Python, and press the **q** key to exit the help information interface

```
>>> help(wiringpi)
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

    class GPIO(builtins.object)
```

```
        |    GPIO(pinmode=0)
        |

>>>
```

3. 20. 2. 26pin GPIO port test

wiringOP-Python is the same as wiringOP, you can also determine which GPIO pin to operate by specifying the wPi number, because there is no command to check the wPi number in wiringOP-Python, so you can only use the gpio command in wiringOP to check the correspondence between the board wPi number and the physical pin.

```
root@orangepi5b:~# gpio readall
 +------+-----+----------+--------+---+   PI5B   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 |   |        | PowerKey |     |      |
 |   50 |  13 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | IN     | SPI4_CS1 | 14  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 15  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   PI5B   +---+--------+----------+-----+------+
root@orangepi5b:~#
```

1) A total of **16** GPIO ports can be used in the 26pins of the development board. The following uses pin 7—the corresponding GPIO is GPIO1_C6——the corresponding wPi serial number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepi5b:~# gpio readall
 +------+-----+----------+--------+---+   PI5B   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
```

2) The steps to test directly with the command are as follows：

a. First set the GPIO port to the output mode, where the first parameter of the

pinMode function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.pinMode(2, GPIO.OUTPUT) ;  "
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.LOW)"
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.HIGH)"
```

3) The steps to test in the command line of python3 are as follows：

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

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

c. Then set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
>>> wiringpi.wiringPiSetup()
0
>>> wiringpi.pinMode(2, GPIO.OUTPUT)
```

d. Then set the GPIO port to output 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.

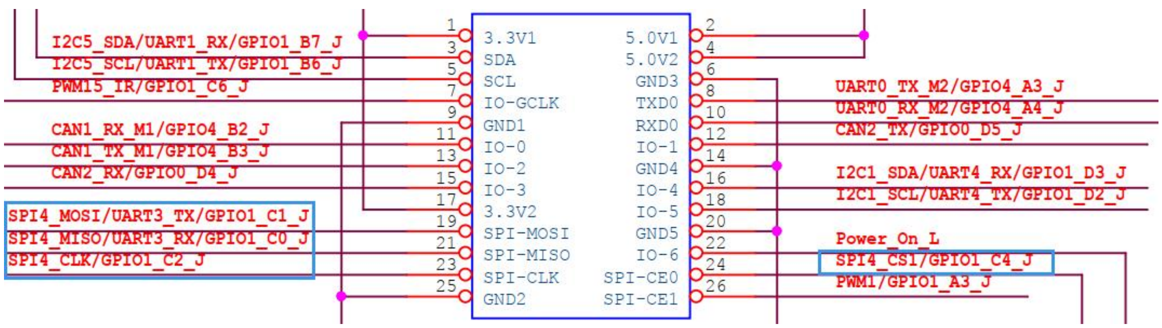
```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

4) The method of wiringOP-Python to set GPIO high and low levels in python code can refer to the **blink.py** test program in the examples below. The **blink.py** test program will set the voltage of all GPIO ports in the 26 pins of the development board to change continuously.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# ls blink.py
blink.py
root@orangepi:~/wiringOP-Python/examples# python3 blink.py
```

3. 20. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the spi available for Orange Pi 5B is spi4



In the Linux system, the spi4 in the 26pin is closed by default, and it needs to be opened manually to use it.

Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open spi4.

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=spi4-m0-cs1-spidev
```

2) First check whether there is a **spidev4.1** device node in the linux system. If it exists, it means that SPI4 has been set up and can be used directly

```
orangepi@orangepi:~$ ls /dev/spidev4.1
```

```
/dev/spidev4.1
```

Note that /dev/spidev4.0 cannot be used, please use /dev/spidev4.1, don't make a mistake.

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

a. **--channel**：Specifies the channel number of the SPI
b. **--port**：Specify the port number of the SPI

4) Do not short-circuit the mosi and miso pins of SPI4, the output result of running spidev_test.py is as follows, you can see that the data of TX and RX are inconsistent

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 4 --port 1
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev4.1
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 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 FF  |.................|
```

5) Then use the Dupont wire to short-circuit the two pins of txd (pin 19 in the 26pin interface) and rxd (pin 21 in the 26pin interface) of SPI4 and then run spidev_test.py The output is as follows, you can see The data sent and received are the same, indicating that the SPI4 loopback test is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 4 --port 1
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev4.1
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 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 FF F0 0D    |......@..........|
```

3. 20. 4. 26pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi 5B is i2c1, i2c3 and i2c5, a total of three sets of 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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22		PowerKey			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

As can be seen from the above table, i2c1 can be derived from pins 12 and 15 of 26pin (i2c1_m2), and can also be derived from pins 16 and 18 of 26pin (i2c1_m4), please follow your own needs Just select a group. Please don't think that these are two different sets of i2c buses.

In the linux system, the i2c in the 26pin is turned off by default, and it needs to be turned on manually before it can be used.

Add the following configuration in red font to /boot/orangepiEnv.txt, and then restart the Linux system to open i2c1, i2c3 and i2c5 at the same time. If you only need to open one, then just fill in one.

Select the settings for i2c1_m2 as shown below:

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

overlays=i2c1-m2 i2c3-m0 i2c5-m3

Select the settings for i2c1_m4 as shown below:

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

overlays=i2c1-m4 i2c3-m0 i2c5-m3

2) After starting the linux system, first confirm that there is an i2c device node under /dev

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

```
/dev/i2c-0   /dev/i2c-10   /dev/i2c-3   /dev/i2c-6   /dev/i2c-9
/dev/i2c-1   /dev/i2c-2    /dev/i2c-5   /dev/i2c-7
```

3) Then connect an i2c device to the i2c pin of the 26pin connector, here we take the ds1307 RTC module as an example

	i2c1-m2	i2c1-m4	i2c3-m0	i2c5-m3
Sda pin	Corresponding to pin 12	Corresponding to pin 16	Corresponding to pin 21	Corresponding to pin 3
Sck pin	Corresponding to pin 15	Corresponding to pin 18	Corresponding to pin 19	Corresponding to pin 5
Vcc pin	Corresponding to pin 1	Corresponding to pin 1	Corresponding to pin 1	Corresponding to pin 1
Gnd pin	Corresponding to pin 6	Corresponding to pin 6	Corresponding to pin 6	Corresponding to pin 6

4) 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 3        #i2c3command
orangepi@orangepi:~$ sudo i2cdetect -y 5        #i2c5command
```

```
orangepi@orangepi5:~$ sudo i2cdetect -y 5
[sudo] password for orangepi:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
orangepi@orangepi5:~$
```

5) 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-5"
Thu 2023-01-05    14:57:55
Thu 2023-01-05    14:57:56
Thu 2023-01-05    14:57:57
^C
exit
```

3. 20. 5. 26pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5B are four groups of uart buses: uart0, uart1, uart3 and uart4

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22		PowerKey			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

In the Linux system, the uart in the 26pin is closed by default, and it needs to be opened manually before it can be used.

Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open uart0, uart1, uart3 and uart4 at the same time. If you only need to open one, then fill in one.

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=uart0-m2   uart1-m1   uart3-m0   uart4-m0
```

2) After entering the linux system, first confirm whether there is a device node corresponding to uart under /dev

```
orangepi@orangepi:~$ ls /dev/ttyS*
/dev/ttyS0   /dev/ttyS1   /dev/ttyS3   /dev/ttyS4   /dev/ttyS9
```

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

	uart0	uart1	uart3	uart4
Tx pin	Corresponding to pin 8	Corresponding to pin 5	Corresponding to pin 19	Corresponding to pin 18
Rx pin	Corresponding to pin 10	Corresponding to pin 3	Corresponding to pin 21	Corresponding to pin 16

4) Use the **serialTest.py** program in the examples 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

a. Test UART0

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS0"

Out:    0: ->    0
Out:    1: ->    1
Out:    2: ->    2
Out:    3: ->    3
Out:    4:^C
exit
```

b. Test UART1

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS1"

Out:    0: ->   0
Out:    1: ->   1
Out:    2: ->   2
Out:    3: ->   3
Out:    4:^C
exit
```

c. Test UART3

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS3"

Out:    0: ->   0
Out:    1: ->   1
Out:    2: ->   2
Out:    3: ->   3
Out:    4:^C
exit
```

d. Test UART4

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS4"

Out:    0: ->   0
Out:    1: ->   1
Out:    2: ->   2
Out:    3: ->   3
Out:    4:^C
exit
```

3.21. Hardware watchdog test

The watchdog_test program is pre-installed in the linux system released by Orange Pi, which can be tested directly.

The method to run the watchdog_test program is as follows:

a. The second parameter 10 indicates the counting time of the watchdog. If the dog 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 is fed successfully

```
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. 22. View the serial number of the RK3588S chip

The command to view the serial number of the RK3588S chip is as follows. The serial number of each chip is different, so the serial number can be used to distinguish multiple development boards.

```
orangepi@orangepi:~$ cat_serial.sh
Serial          : 1404a7682e86830c
```

3. 23. How to install Docker

1) The linux image provided by Orange Pi has pre-installed Docker, but the Docker service is not enabled by default

2) 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 when the system is started next time

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

3) Then 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.
.....
```

3. 24. How to download and install arm64 version balenaEtcher

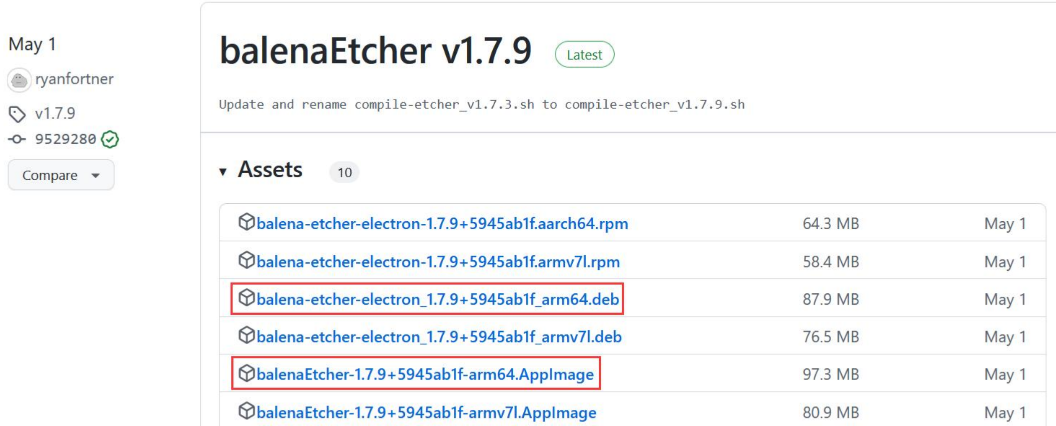
1) The download address of balenaEtcher arm64 version is:

   a. The download address of the deb installation package is as follows, which needs to be installed before it can be used

```
https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balena-etcher-electron_1.7.9+5945ab1f_arm64.deb
```

   b. The download address of the AppImage version that does not need to be installed is as follows:

```
https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balenaEtcher-1.7.9+5945ab1f-arm64.AppImage
```

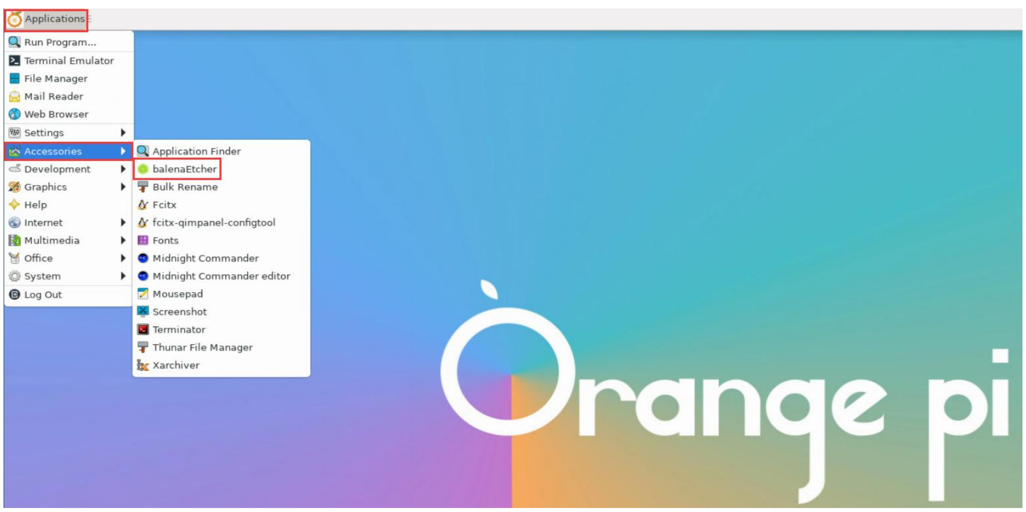


2) How to install and use deb version balenaEtcher:

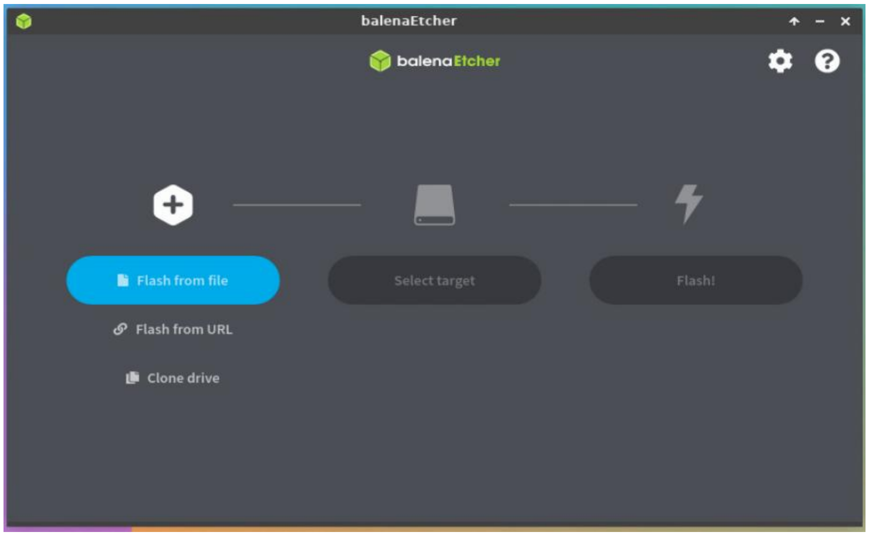
a. The deb version of balenaEtcher installation command is as follows:

```
orangepi@orangepi:~$ sudo apt install -y \
--fix-broken ./balena-etcher-electron_1.7.9+5945ab1f_arm64.deb
```

b. After the deb version of balenaEtcher is installed, it can be opened in the Application



c. The interface after balenaEtcher is opened is as follows:

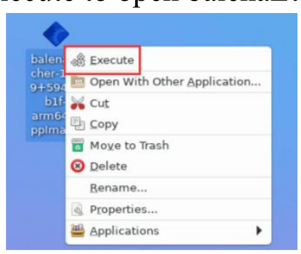


3) How to use the AppImage version of balenaEtcher:

a. First add permissions to balenaEtcher

```
orangepi@orangepi:~/Desktop$ chmod +x balenaEtcher-1.7.9+5945ab1f-arm64.AppImage
```

b. Then select the AppImage version balenaEtcher, then click the right mouse button, and then click Execute to open balenaEtcher



3. 25. How to install Pagoda Linux panel

Pagoda Linux panel is a server management software that improves operation and maintenance efficiency, and 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) The recommended order of pagoda Linux system compatibility is

```
Debian11 > Ubuntu 22.04
```

2) Then enter the following command in the linux system to start the installation of the pagoda

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

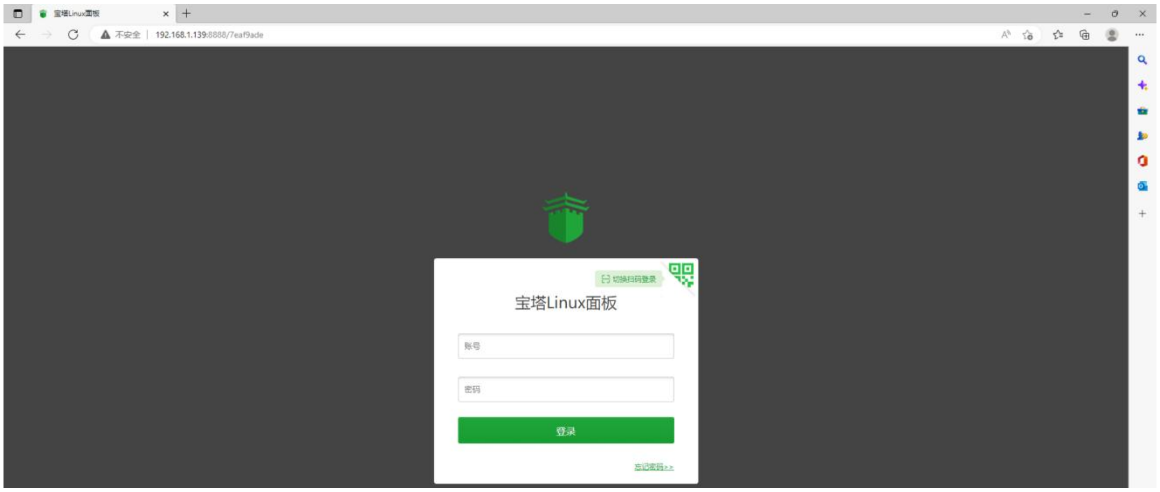
3) Then the pagoda installer will remind whether to install **Bt-Panel** to the **/www** folder, and then enter y

```
+----------------------------------------------------------------------
| 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
```

4) The next thing to do is to wait patiently. When the terminal outputs the following print information, it means that the pagoda has been installed. The whole installation process takes about 12 minutes, and there may be some differences depending on the network speed

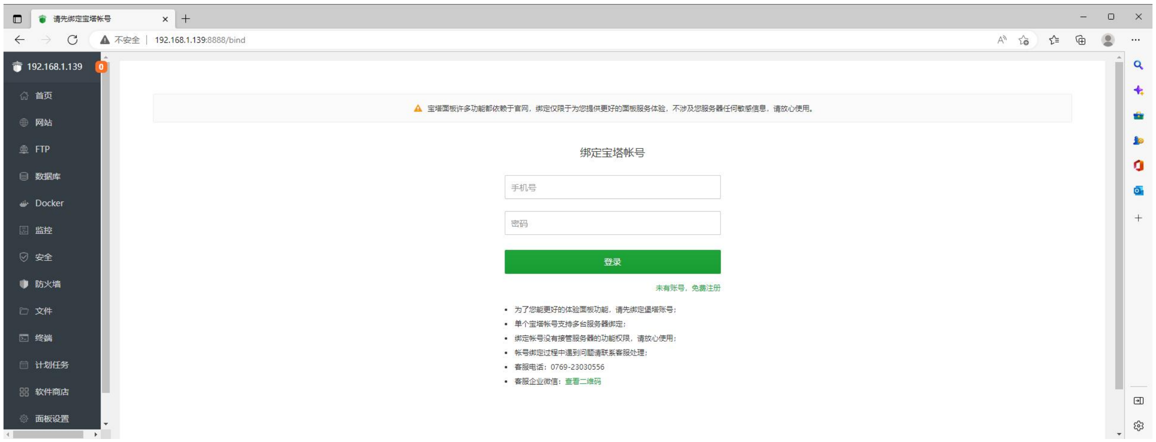
```
==================================================================
Congratulations! Installed successfully!
==================================================================
外网面板地址: http://183.15.204.10:8888/7eaf9ade
内网面板地址: http://192.168.1.139:8888/7eaf9ade
username: nslvetif
password: fec12d4b
If you cannot access the panel,
release the following panel port [8888] 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[8888]端口
==================================================================
Time consumed: 12 Minute!
root@orangepi5:~#
```

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

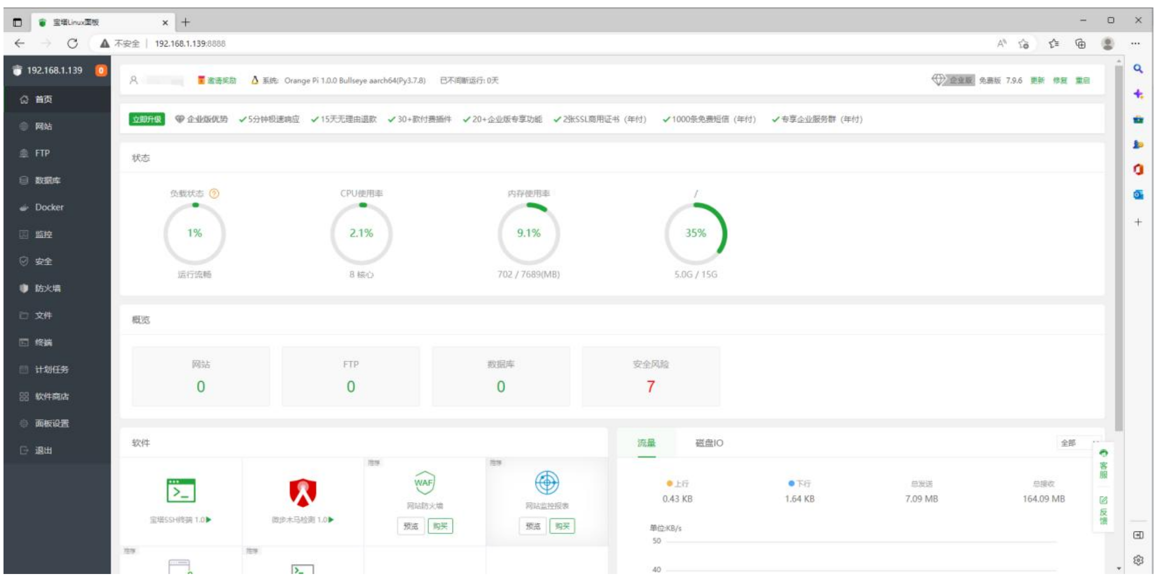


6) After successfully logging into the pagoda, the following welcome interface will pop up. First, please read the user notice in the middle and drag it to the bottom, then you can select "I have agreed and read the "User Agreement"", and then click "Enter the Panel" You can enter the pagoda

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

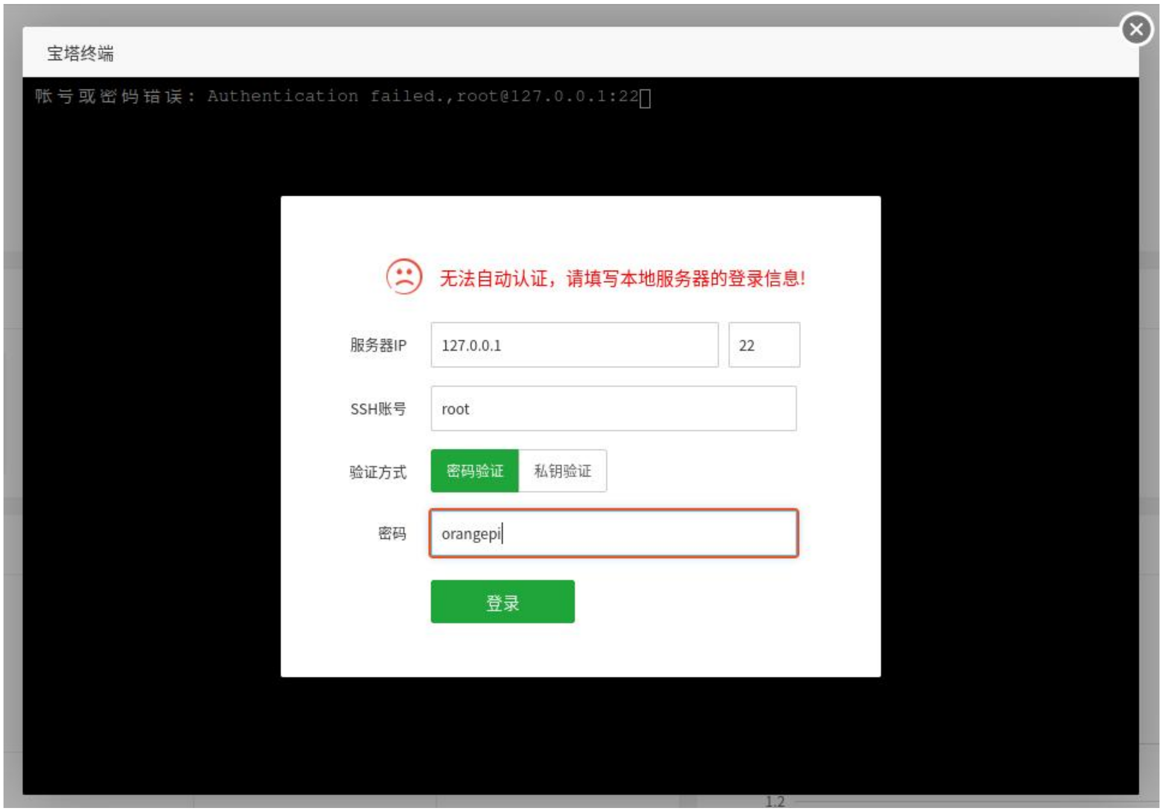


8) The final displayed interface is as shown in the figure below. You can intuitively see some status information of the Linux system on the development board, such as load status, CPU usage, memory usage, and storage space usage, etc.

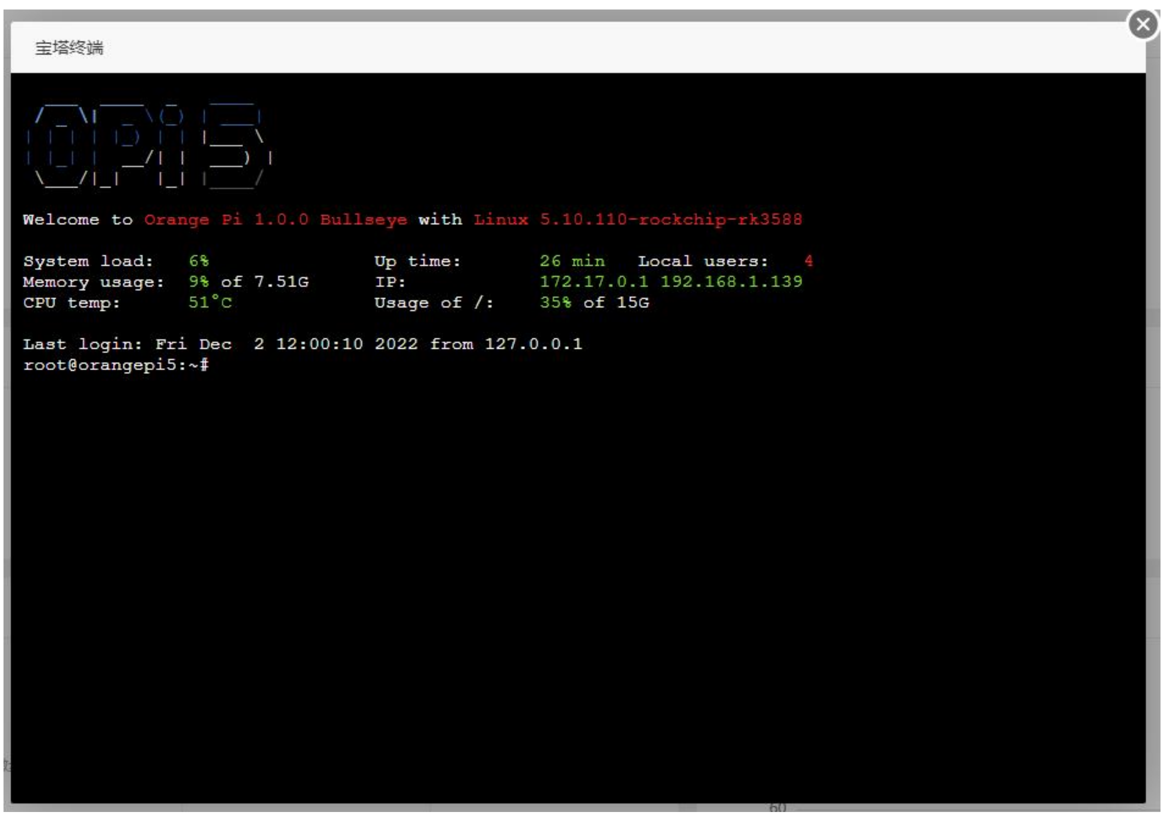


9) Test the SSH terminal login of the pagoda

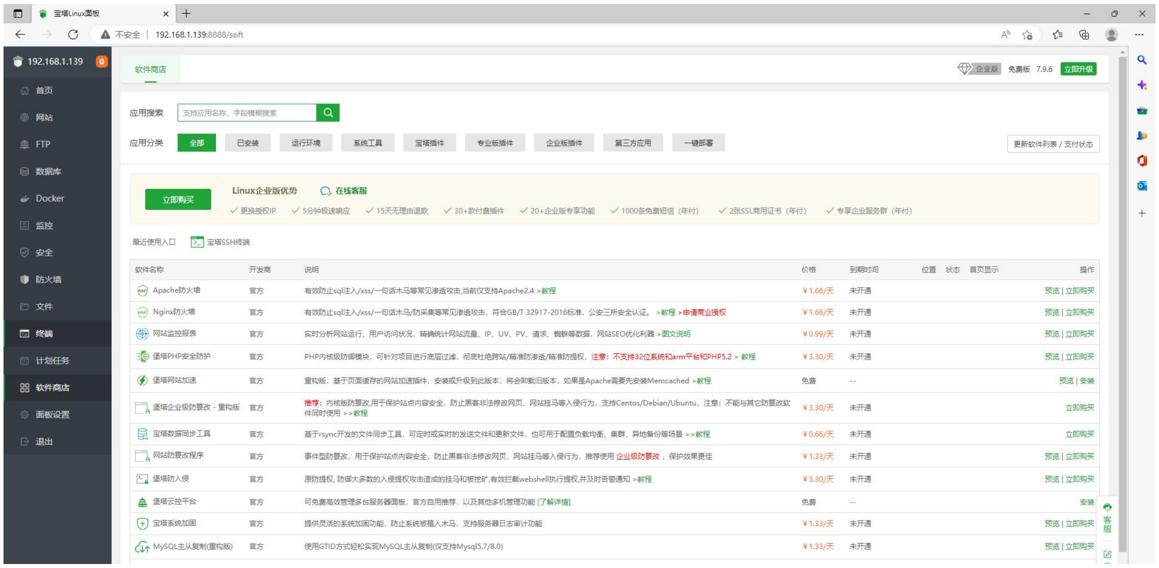
a. After opening the SSH terminal of the pagoda, it will first prompt to enter the password of the development board system. At this time, enter **orangepi** (default password, if there is any modification, please fill in the modified one) in the password box.



b. The display after successful login is shown in the figure below



10) You can install software such as Apache, MySQL, and PHP in the software store of Pagoda, and you can also deploy various applications with one click. Please explore these functions by yourself, and I will not demonstrate them one by one here.



11) Pagoda command line tool test

```
orangepi@orangepi5:~$ sudo bt
[sudo] password for orangepi:
===============宝塔面板命令行==================
(1) 重启面板服务           (8) 改面板端口
(2) 停止面板服务           (9) 清除面板缓存
(3) 启动面板服务           (10) 清除登录限制
(4) 重载面板服务
(5) 修改面板密码           (12) 取消域名绑定限制
(6) 修改面板用户名         (13) 取消IP访问限制
(7) 强制修改MySQL密码      (14) 查看面板默认信息
(22) 显示面板错误日志      (15) 清理系统垃圾
(23) 关闭BasicAuth认证     (16) 修复面板(检查错误并更新面板文件到最新版)
(24) 关闭动态口令认证          (17) 设置日志切割是否压缩
(25) 设置是否保存文件历史副本  (18) 设置是否自动备份面板
(0) 取消                   (29) 取消访问设备验证
==============================================
请输入命令编号：14
==============================================
正在执行(14)...
==============================================
curl: (28) Resolving timed out after 10000 milliseconds
==================================================================
BT-Panel default info!
==================================================================
外网面板地址: http://:8888/7eaf9ade
内网面板地址: http://192.168.1.139:8888/7eaf9ade
*以下仅为初始默认账户密码，若无法登录请执行bt命令重置账户/密码登录
username: nslvetif
password: ********
If you cannot access the panel,
release the following panel port [8888] 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[8888]端口
==================================================================
orangepi@orangepi5:~$
```

12) For more functions of the pagoda, you can refer to the following information to explore by yourself

manual: **http://docs.bt.cn**
Forum address: **https://www.bt.cn/bbs**
GitHub: **https://github.com/aaPanel/BaoTa**

3. 26. How to remotely log in to the desktop of the Linux system

The Ubuntu Gnome Wayland image does not support remote login to the desktop using Nomachine and VNC as described here.

3. 26. 1. Remote login using NoMachine

Please make sure that the Ubuntu or Debian system installed on the development board is a desktop version. In addition, NoMachine also provides detailed usage documents. It is strongly recommended to read this document to familiarize yourself with the use of NoMachine. The document link is as follows:

https://knowledgebase.nomachine.com/DT10R00166

NoMachine supports Windows, Mac, Linux, iOS and Android platforms, so we can remotely log in and control the Orange Pi development board through NoMachine on a variety of devices. The following demonstrates how to remotely log in to the Linux system desktop of the Orange Pi development board through NoMachine in Windows. For installation methods on other platforms, please refer to the official documentation of NoMachine.

Before operation, please make sure that the Windows computer and the development board are in the same LAN, and you can log in to the Ubuntu or Debian system of the development board through ssh.

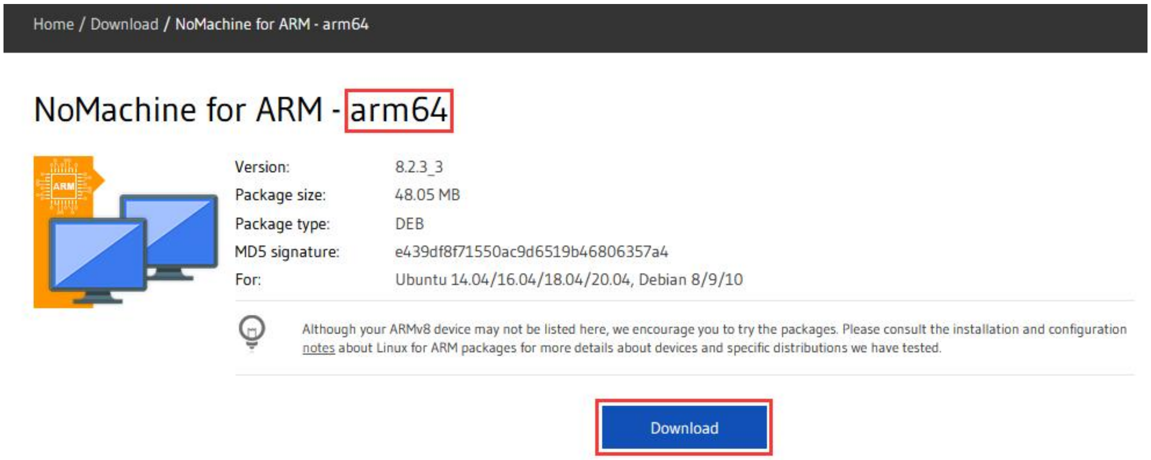
1) First download the installation package of the NoMachine software Linux **arm64** deb version, and then install it in the Linux system of the development board

a. Since RK3588S is a SOC with ARMv8 architecture, the system we use is Ubuntu or Debian, so here we need to download the **NoMachine for ARM**

ARMv8 DEB installation package, the download link is as follows:

Note that this download link may change, please look for the deb package of the Armv8/Arm64 version.

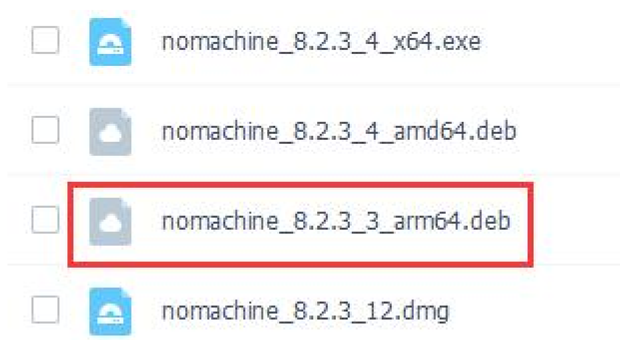
https://downloads.nomachine.com/download/?id=114&distro=ARM



b. In addition, you can also download the installation package of **NoMachine** in **the official tool**

First enter **the remote login software-NoMachine** folder

Then download the arm64 version of the deb installation package



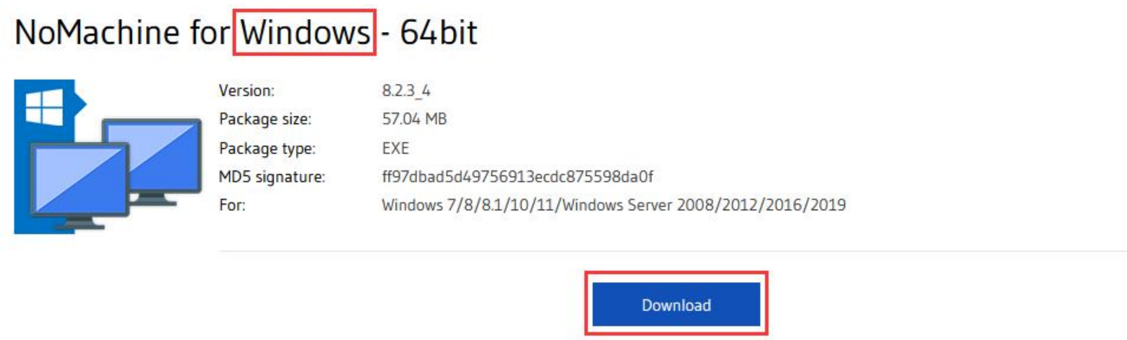
c. Then upload the downloaded **nomachine_8.2.3_3_arm64.deb** to the Linux system of the development board

d. Then use the following command to install **NoMachine** in the Linux system of the development board

orangepi@orangepi:~$ **sudo dpkg -i nomachine_8.2.3_3_arm64_arm64.deb**

2) Then download the installation package of the Windows version of the NoMachine software, the download address is as follows

https://downloads.nomachine.com/download/?id=8

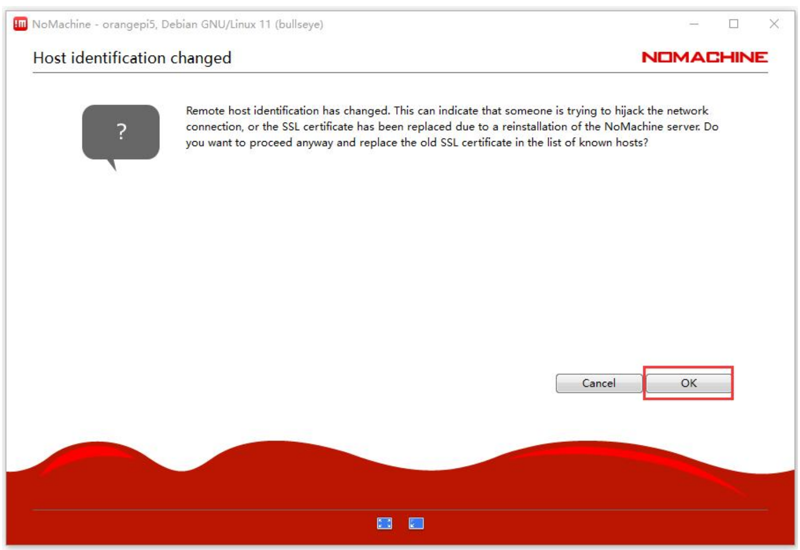


3) Then install NoMachine in Windows, **please restart the computer after installation**

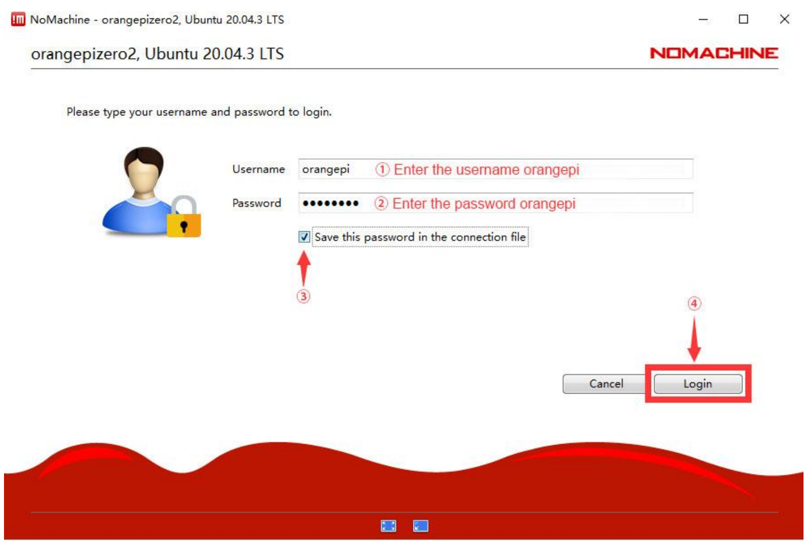
4) Then open **NoMachine** in Window

5) After NoMachine is started, it will automatically scan other devices installed with NoMachine in the LAN. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices, and then click the position shown in the red box in the figure below You can start to log in to the Linux system desktop of the development board

6) Then click **OK**



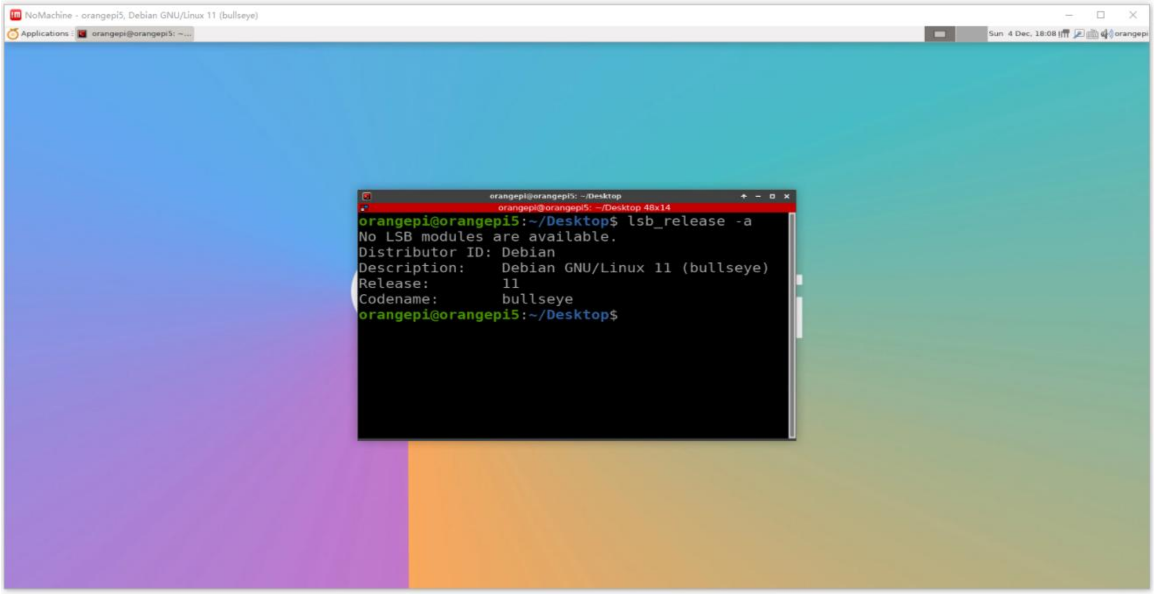
7) Then enter the user name and password of the Linux system of the development board in the corresponding position in the figure below, and then click **Login** to start logging in



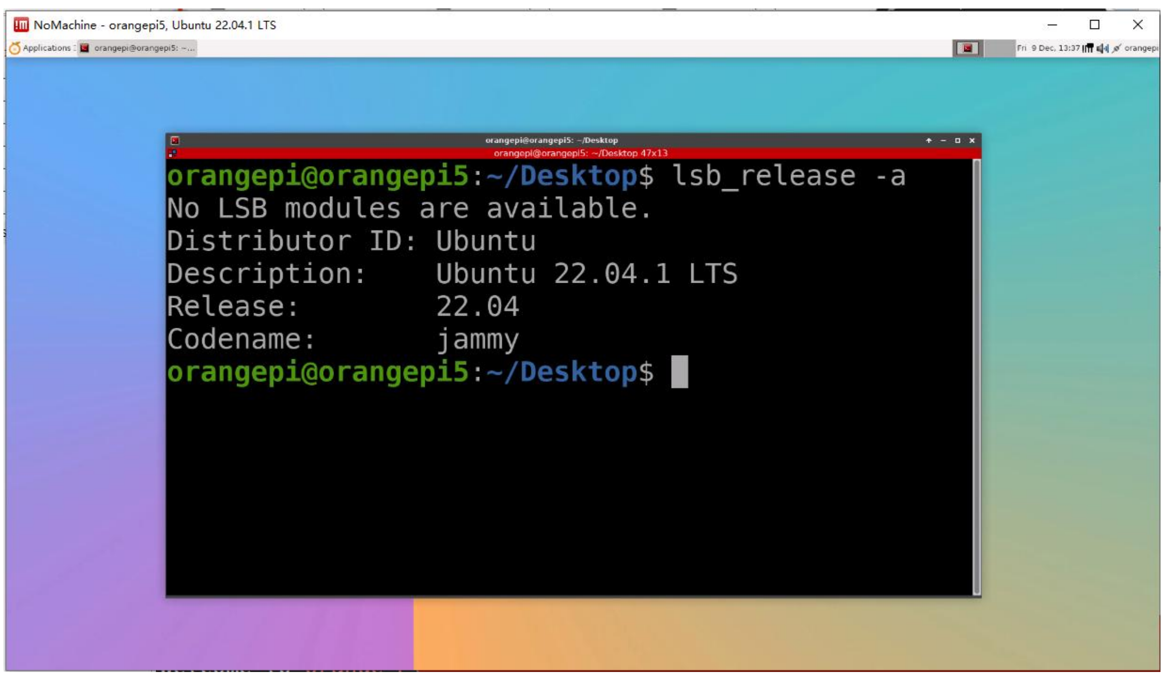
8) Then click OK in the next interface

9) Finally, you can see the desktop of the development board Linux system

a. Debian11



b. Ubuntu22.04



3. 26. 2. Use VNC to log in remotely

Before operation, please make sure that the Windows computer and the development board are in the same LAN, and you can log in to the Ubuntu or Debian system of the development board through ssh.

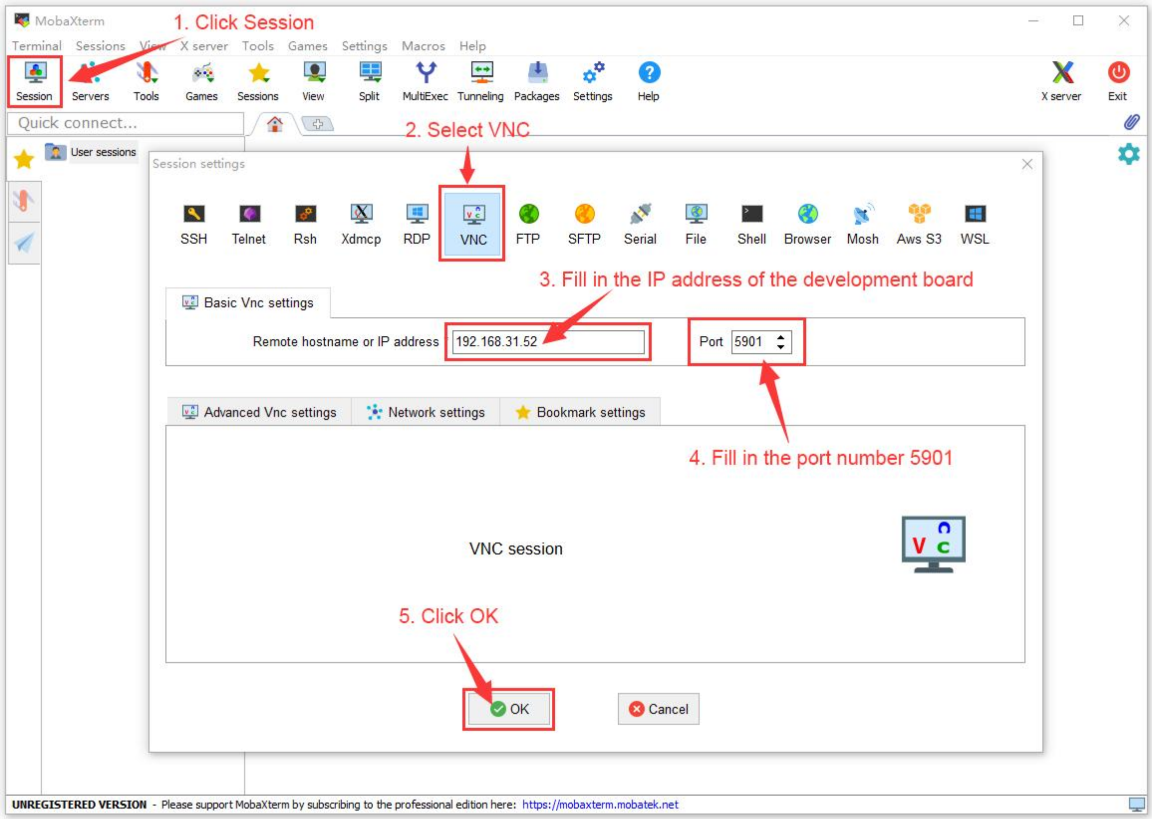
Ubuntu20.04 has many problems testing VNC, please do not use this method.

1) First run the **set_vnc.sh** script to set up vnc, **remember to add sudo permission**

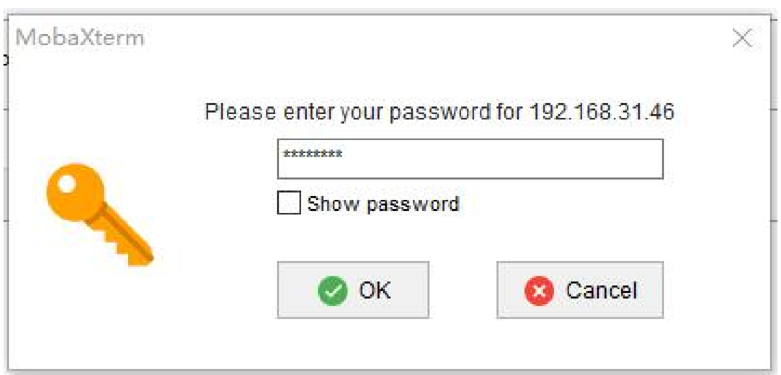
```
orangepi@orangepi:~$ sudo set_vnc.sh
You will require a password to access your desktops.

Password:          #Set the vnc password here, 8 characters
Verify:             #Set the vnc password here, 8 characters
Would you like to enter a view-only password (y/n)? n
xauth:   file /root/.Xauthority does not exist

New 'X' desktop is orangepi5:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi5:1.log

Killing Xtightvnc process ID 3047

New 'X' desktop is orangepi5:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi5:1.log
```

2) The steps to use MobaXterm software to connect to the development board linux system desktop are as follows:

a. First click on Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm

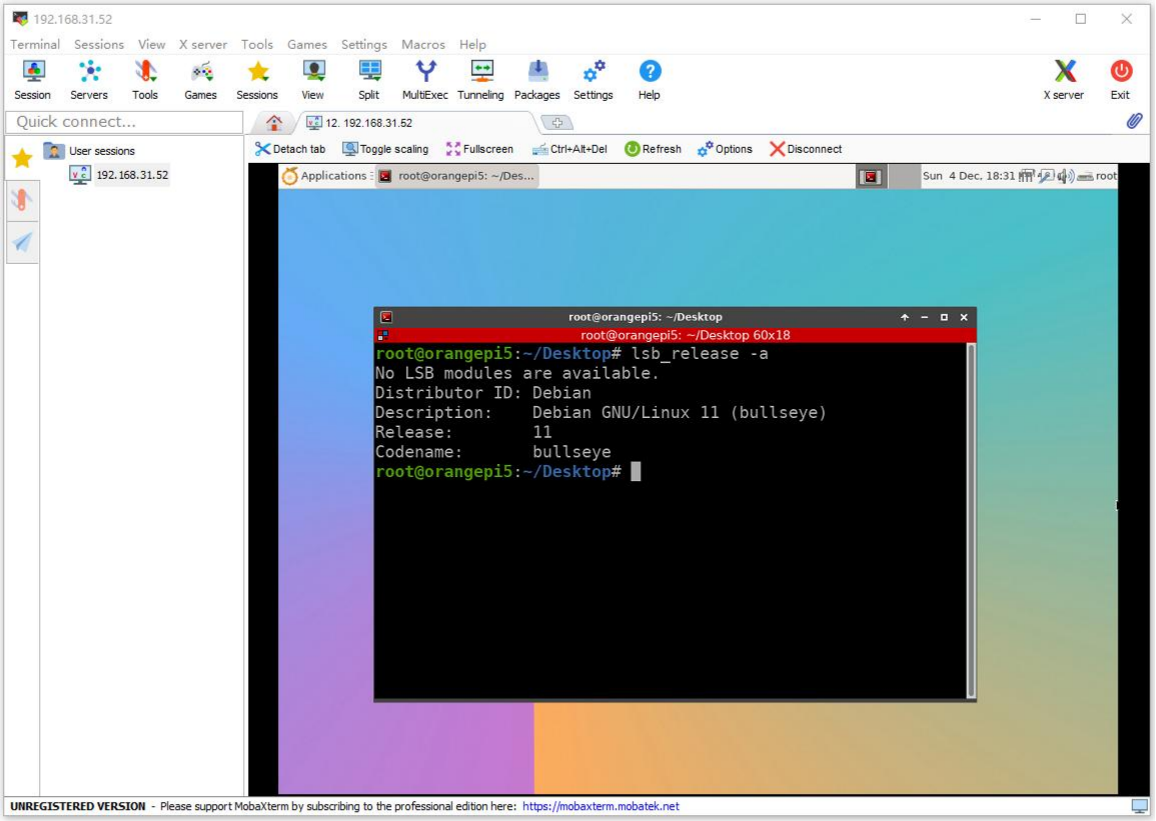


b. Then enter the VNC password set earlier

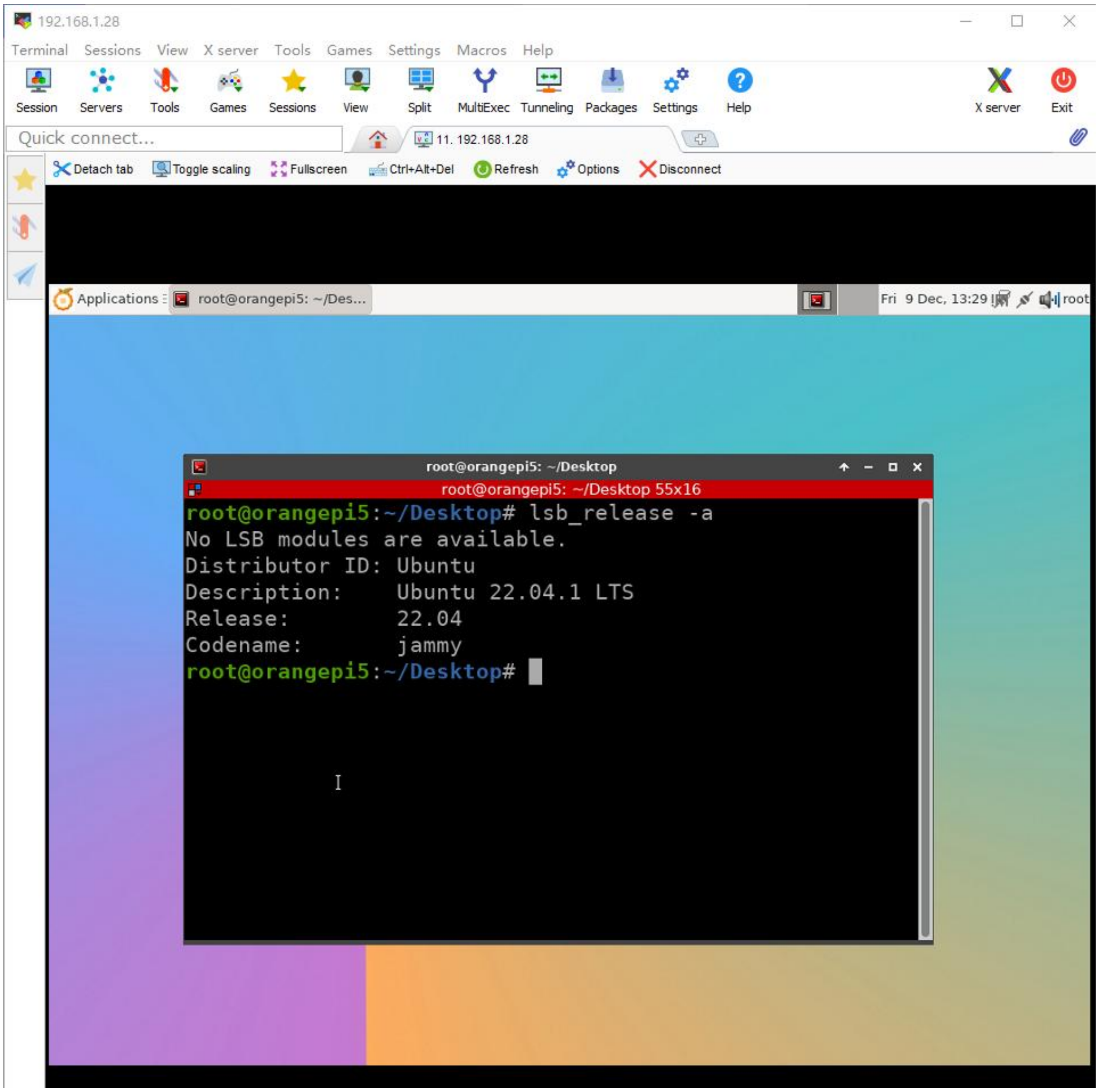


c. After successful login, the interface is displayed as shown in the figure below, and then you can remotely operate the desktop of the development board linux system

a) The Debian11 login display is as follows

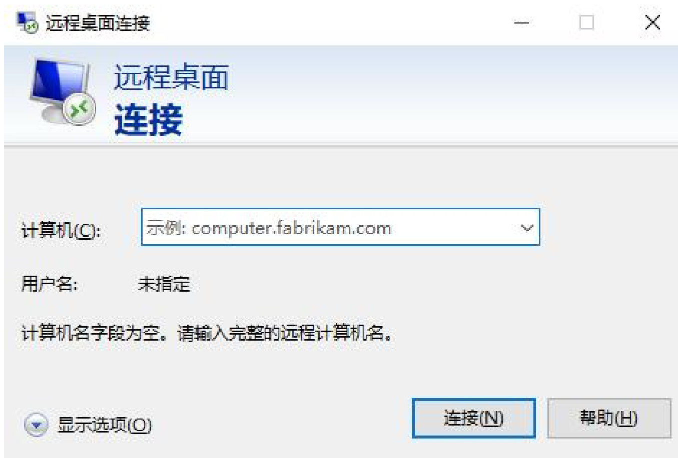


b) The Ubuntu22.04 login display is as follows



3) Use **the remote desktop connection** application that comes with Windows to log in to the Linux system desktop of the development board.

a. First open **the remote desktop connection** that comes with Windows



b. Then enter the IP address of the development board

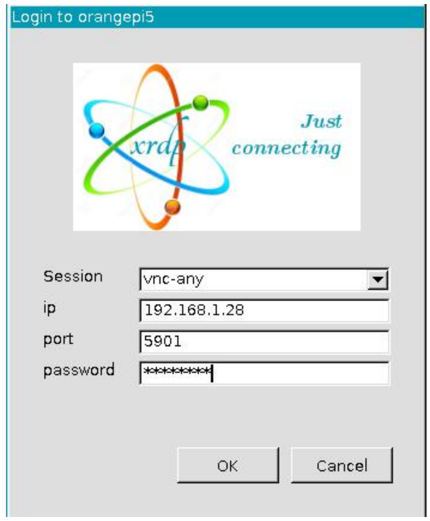
c. Then set the connection information according to the instructions in the figure below

a) **Session:** Need to choose vnc-any

b) **ip:** You can enter 127.0.0.1 or the IP address of the development board

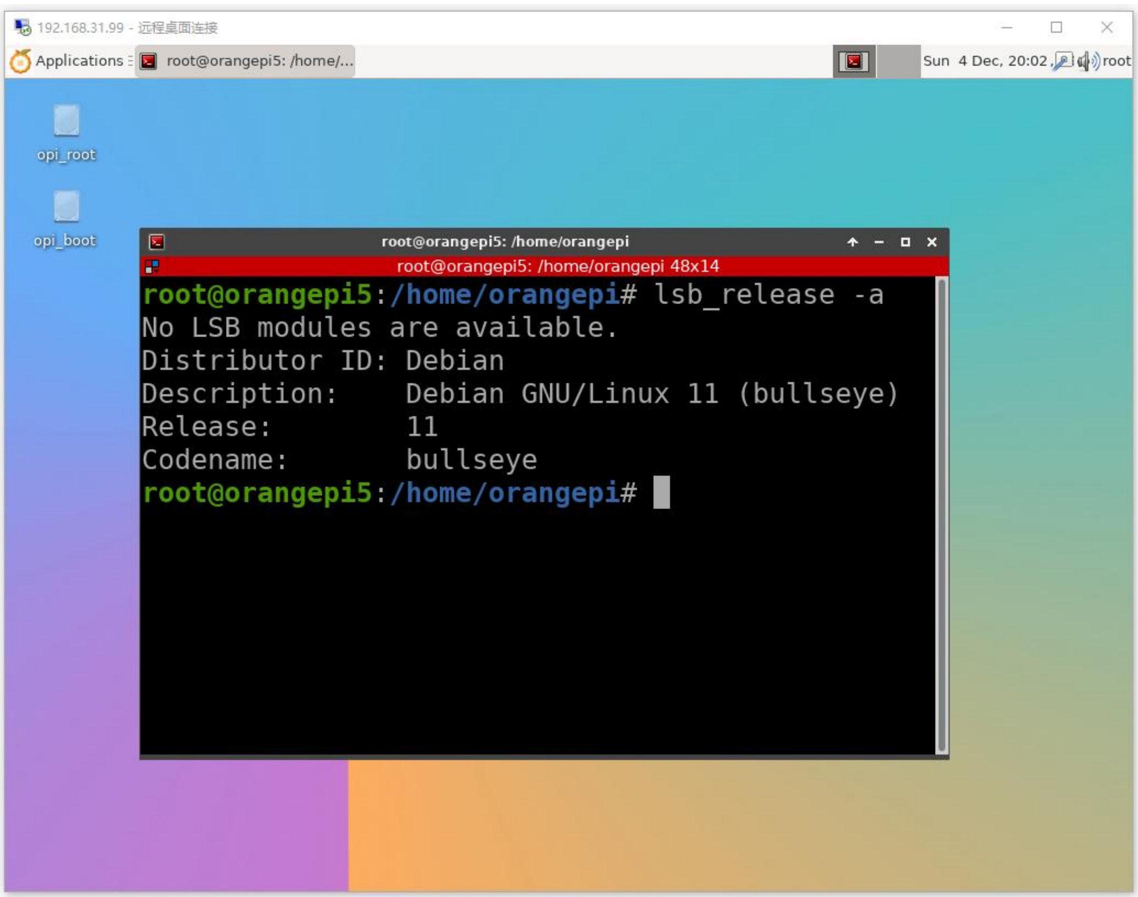
c) **port:** Generally 5901

d) **password:** You need to enter the vnc password



d. The display of the Linux system desktop of the development board successfully logged in is shown in the figure below

a) The Debian11 login display is as follows



b) Ubuntu22.04 is currently unavailable, please do not use this method

3. 27. Some programming language tests supported by Linux system

3. 27. 1. Debian Bullseye system

1) Debian Bullseye has a gcc compilation tool chain installed by default, which can directly compile C language programs in the Linux system of the development board

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 10.2.1-6) 10.2.1 20210110
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bullseye has Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bullseye does not install Java compilation tools and operating environment by default

a. You can use the following command to install openjdk, the latest version in Debian Bullseye is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

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

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 27. 2. Debian Bookworm system

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

a. The version of a.gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 12.2.0-14) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
```

```
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bookworm has Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bookworm does not install Java compilation tools and operating environment by default.

a. You can use the following command to install openjdk. The latest version in Debian Bookworm is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the Java version.

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

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 27. 3. Ubuntu Focal system

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

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");
```

```
        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Ubuntu Focal has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.8.10 (default, Nov 14 2022, 12:59:47)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Ubuntu Focal does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
openjdk 17.0.2 2022-01-18
OpenJDK Runtime Environment (build 17.0.2+8-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 17.0.2+8-Ubuntu-120.04, mixed mode, sharing)
```

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
```

```
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 27. 4. Ubuntu Jammy system

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

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.2.0-19ubuntu1) 11.2.0
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!
```

5) Ubuntu Jammy has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.4 (main, Apr  2 2022, 09:04:19) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

6) Ubuntu Jammy does not install Java compilation tools and operating environment by default

a. You can use the following command to install openjdk-18

```
orangepi@orangepi:~$ sudo apt install -y openjdk-18-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
openjdk 18-ea 2022-03-22
OpenJDK Runtime Environment (build 18-ea+36-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 18-ea+36-Ubuntu-1, mixed mode, sharing)
```

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 28. How to install QT

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

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

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

a. The qt version that comes with Ubuntu 20.04 is **5.12.8**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/aarch64-linux-gnu
```

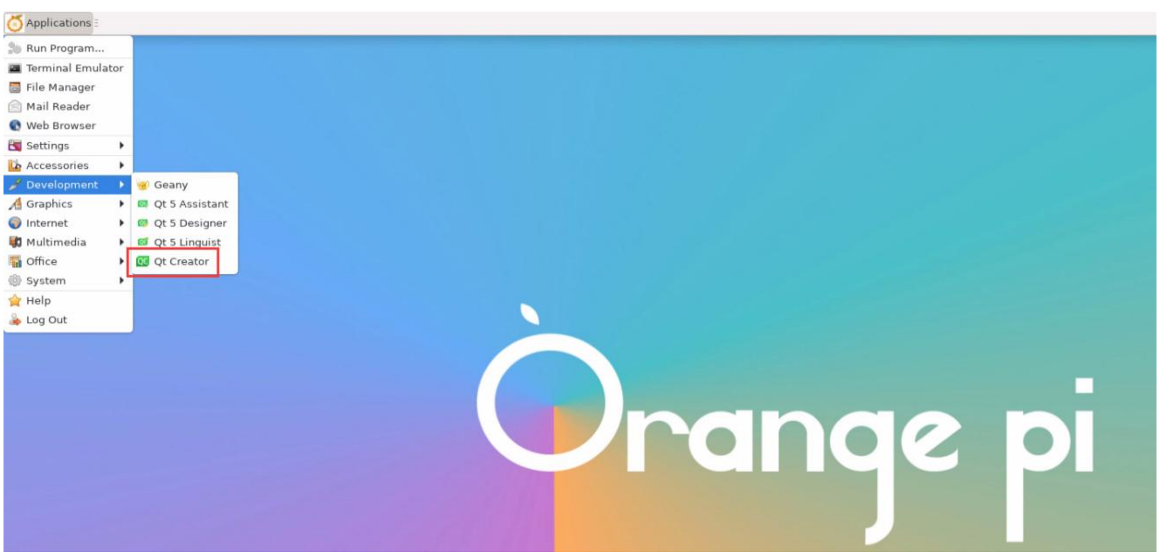
b. The QT version that comes with Ubuntu 22.04 is **5.15.3**

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

c. The QT version that comes with Debian11 is **5.15.2**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

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



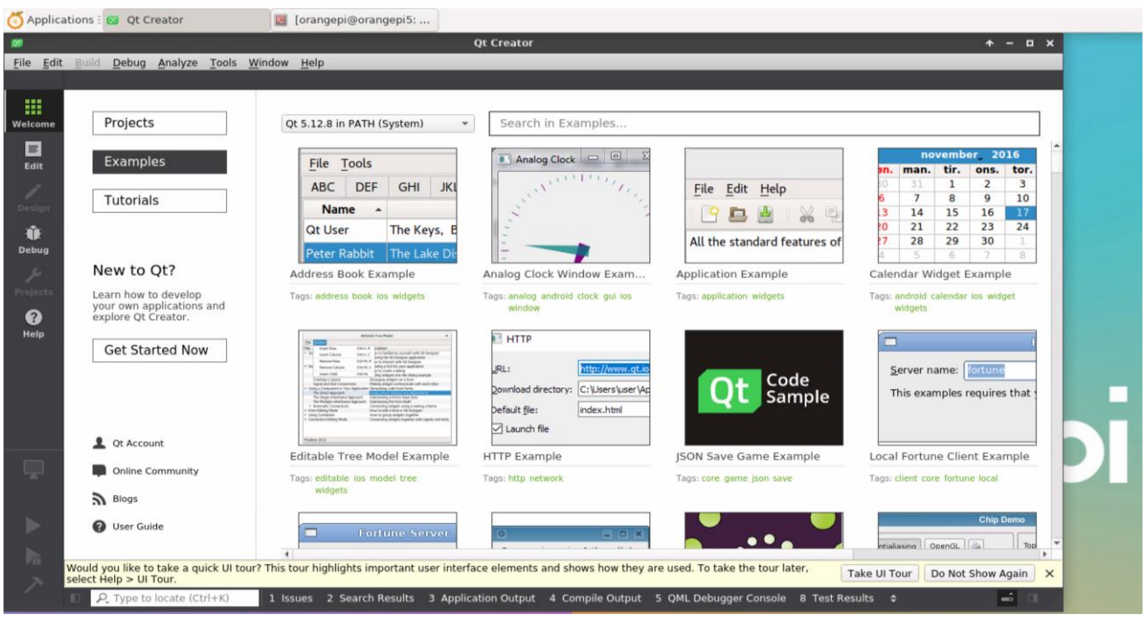
You can also use the following command to open QT Creator

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

During the startup process of QT and QT application, if the following error is prompted, please ignore it directly, this error will not affect the operation of the application.

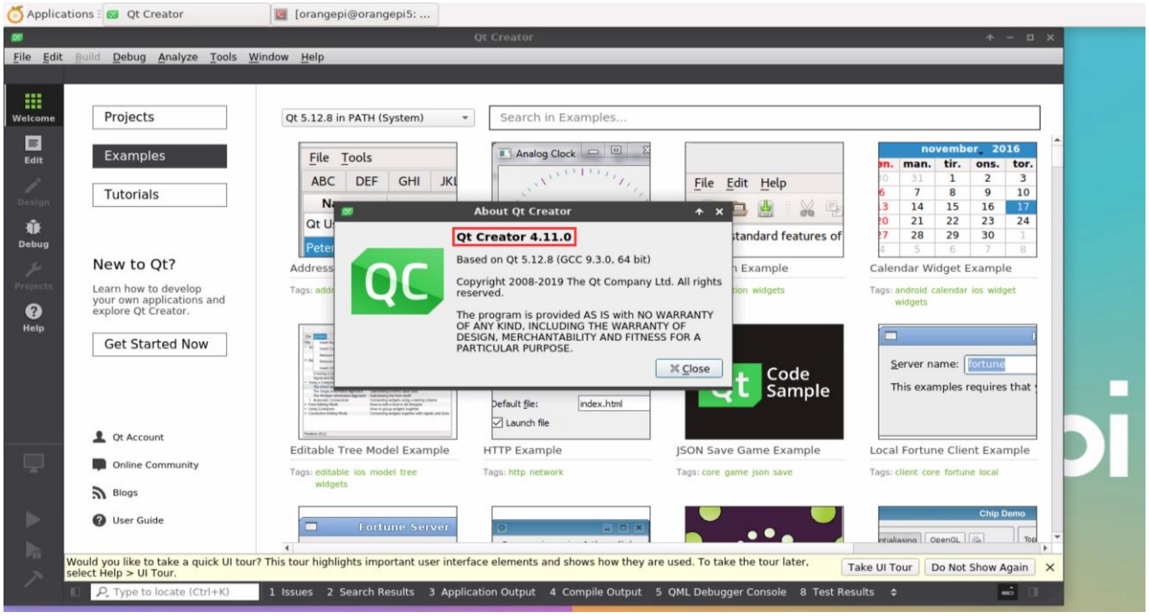
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip

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



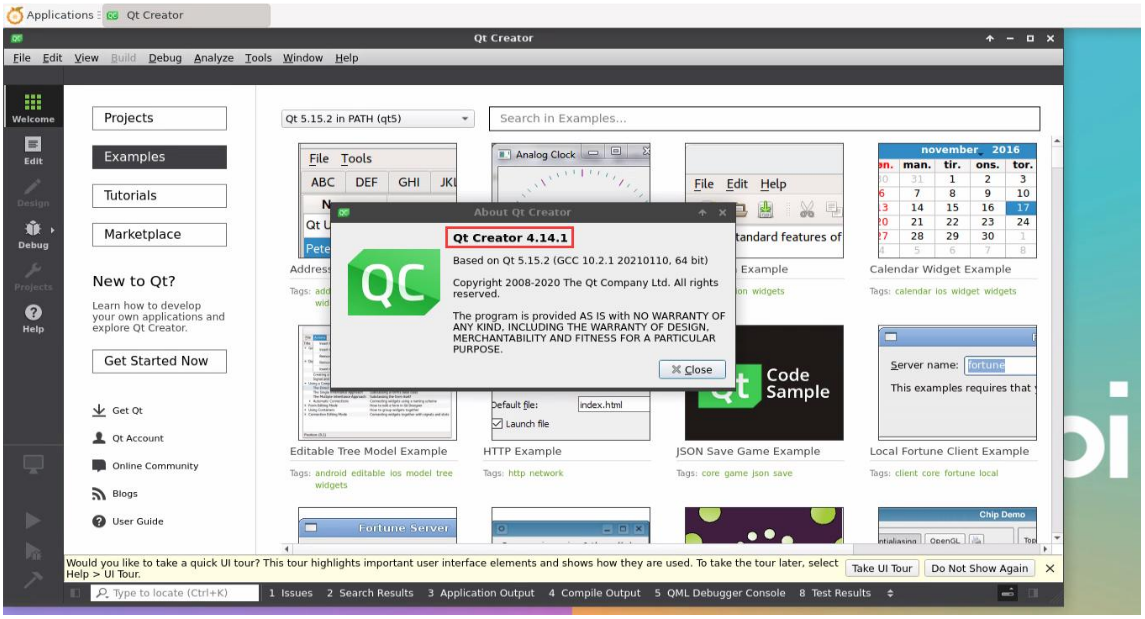
5) The version of QT Creator is as follows

a. The default version of QT Creator in **Ubuntu20.04** is as follows



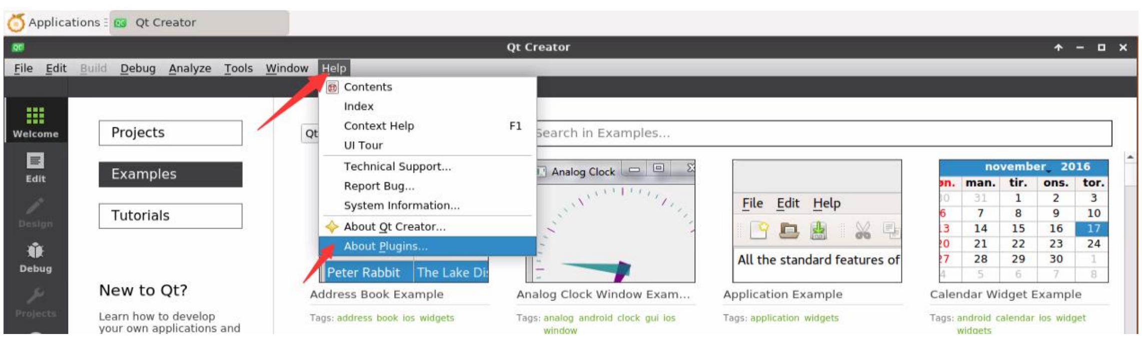
b. The default version of QT Creator in **Ubuntu22.04** is as follows

c. The default version of QT Creator in **Debian11** is as follows

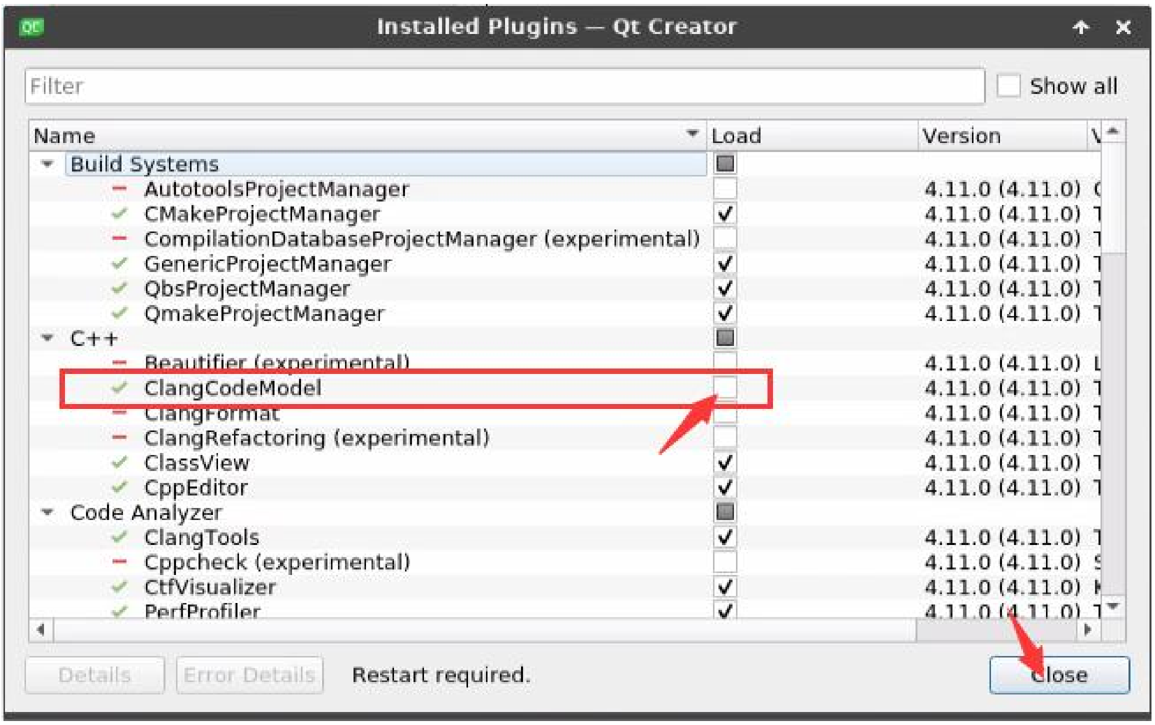


6) Then set the QT

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

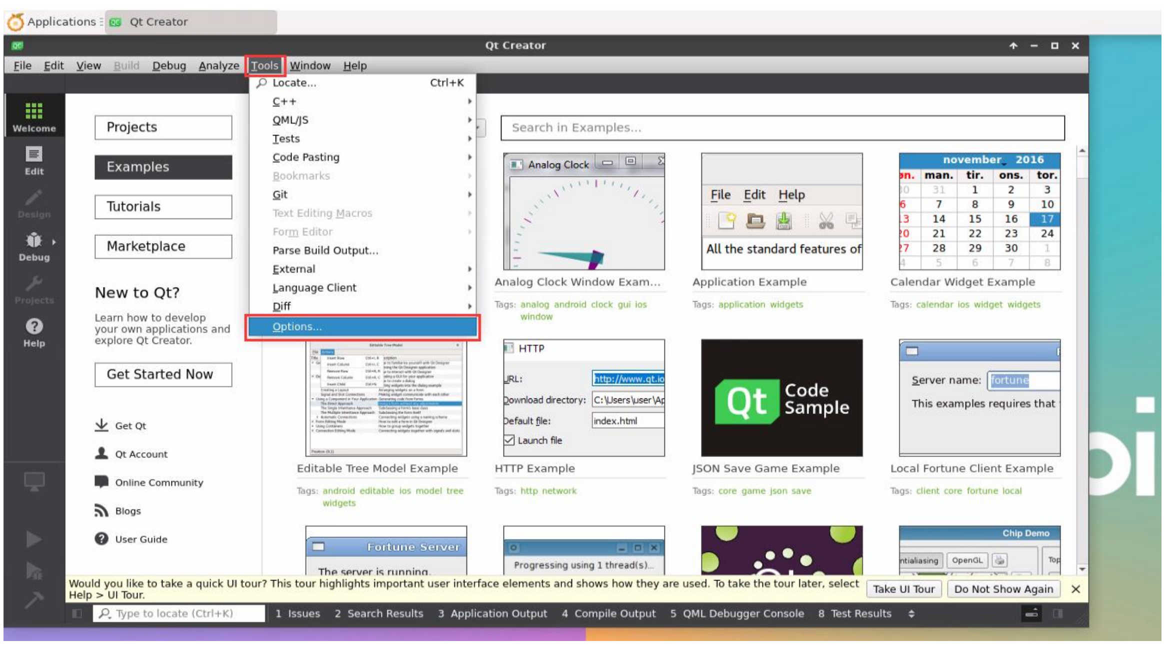


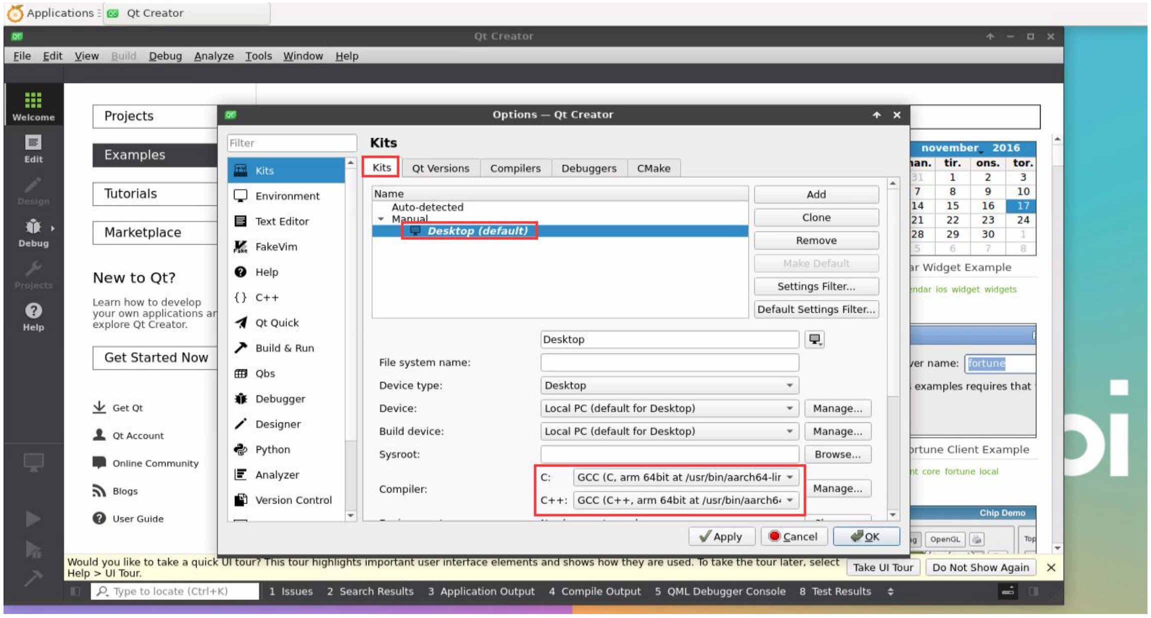
b. Then remove the tick of **ClangCodeModel**



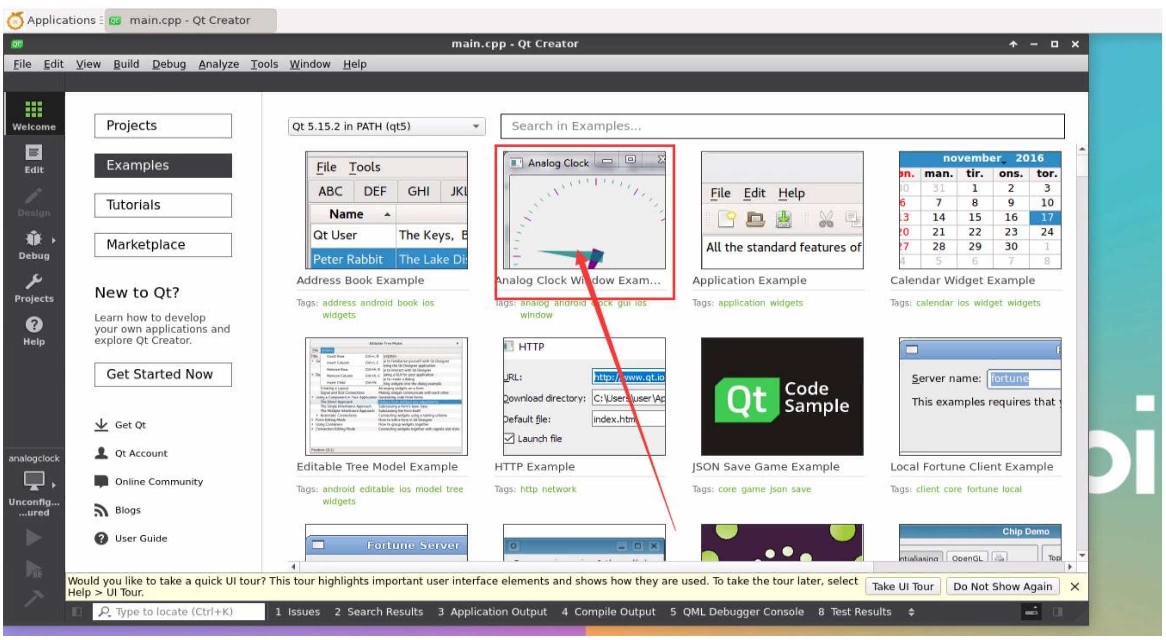
c. **After setting, you need to restart QT Creator**

d. Then make sure the GCC compiler used by QT Creator, if the default is Clang, please modify it to GCC

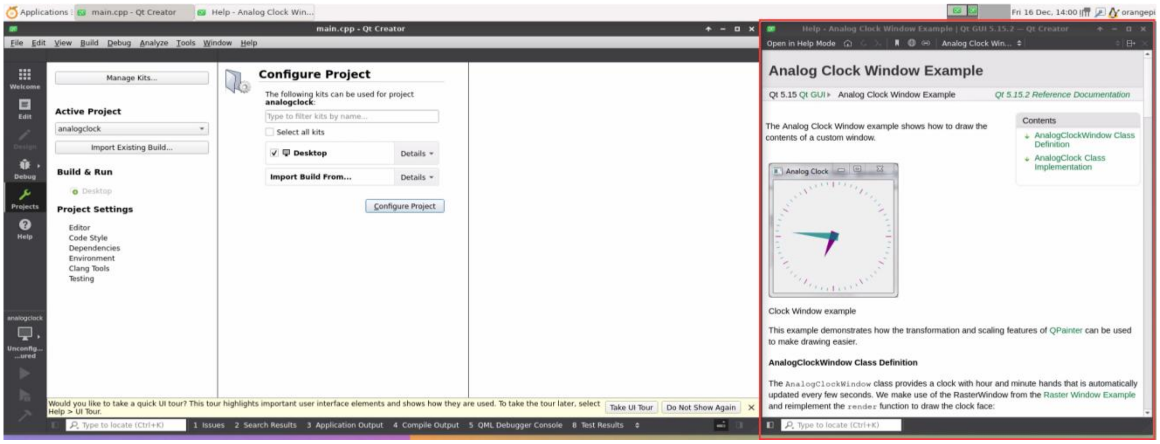




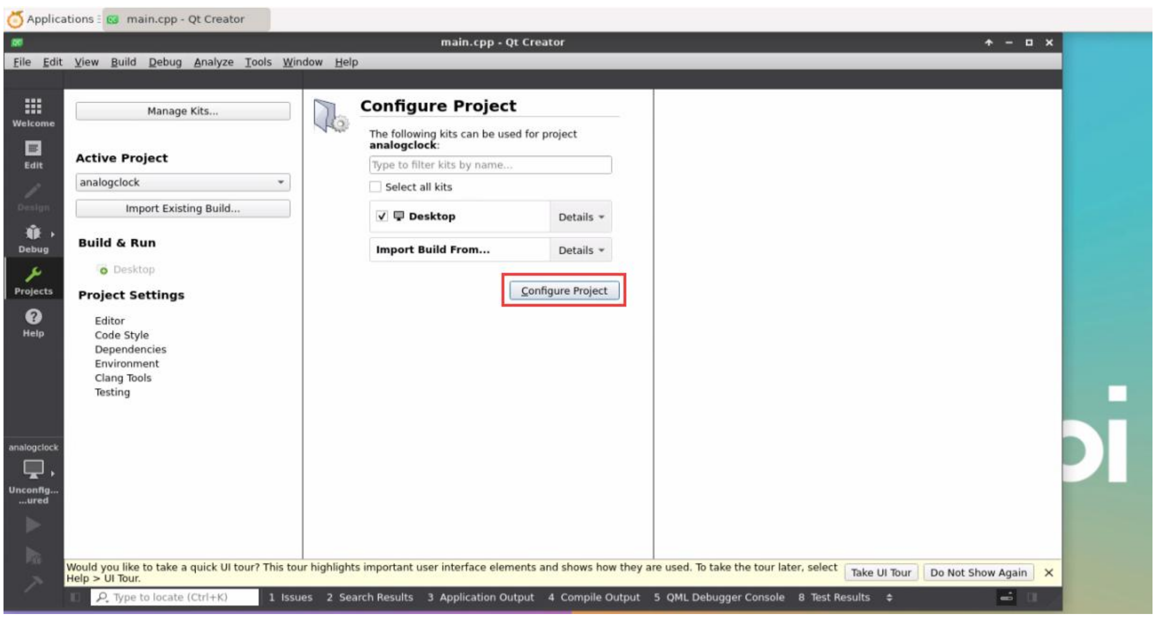
7) Then you can open a sample code



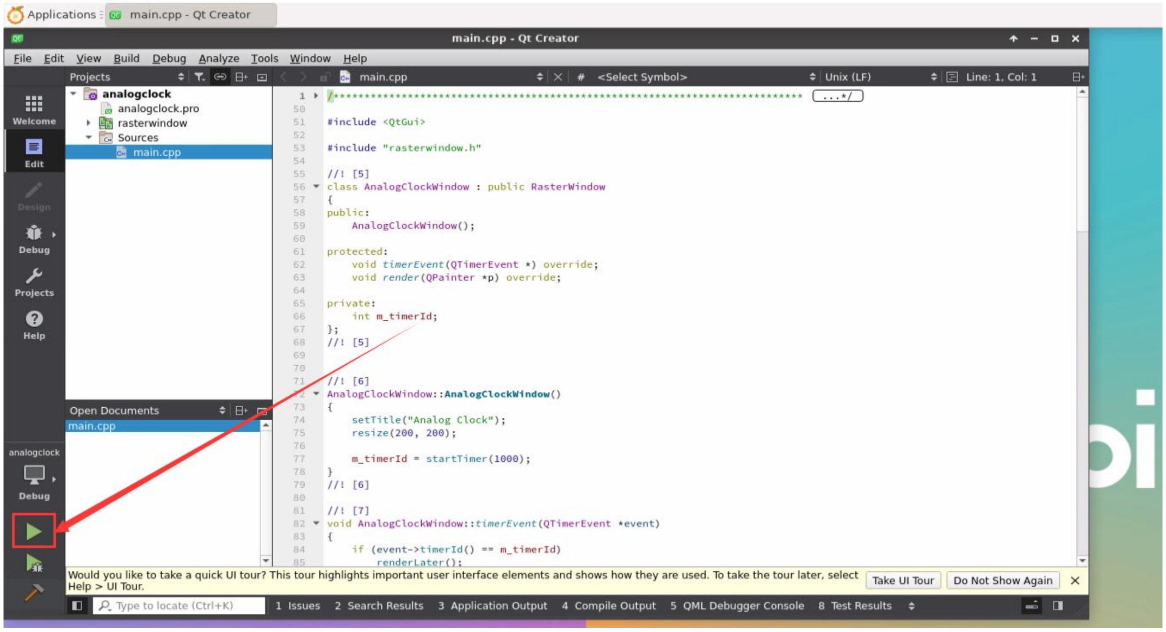
8) After clicking the sample code, the corresponding instruction document will be opened automatically, you can carefully read the instructions in it



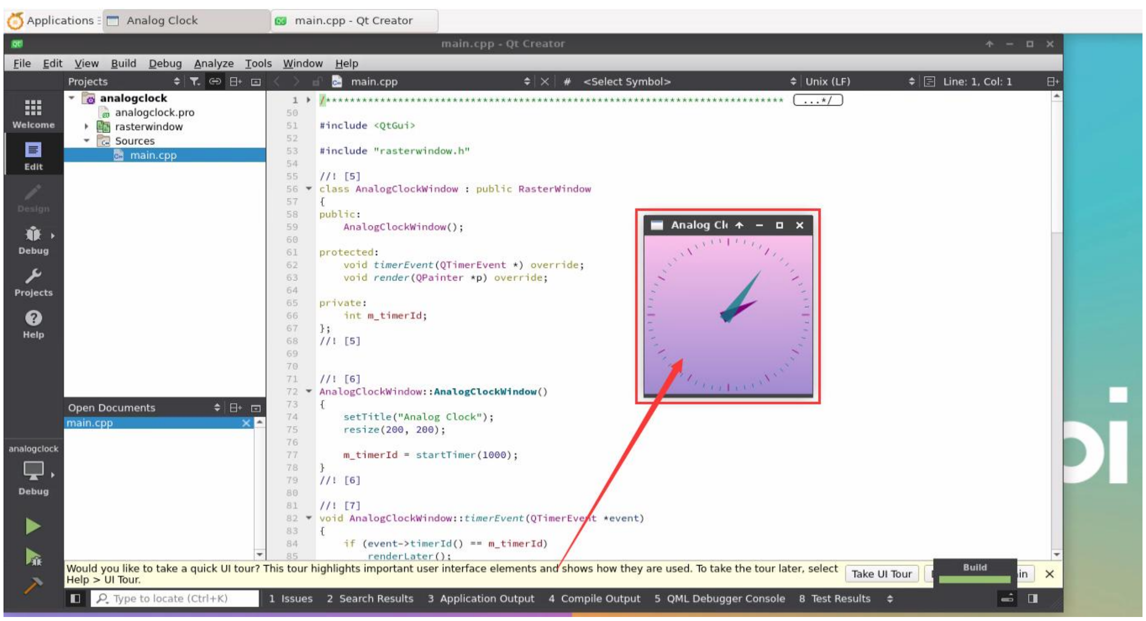
9) Then click **Configure Project**



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



11) After waiting for a period of time, the interface shown in the figure below will pop up, which means that QT can compile and run normally



12) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu

https://download.qt.io/archive/qtcreator

https://download.qt.io/archive/qt

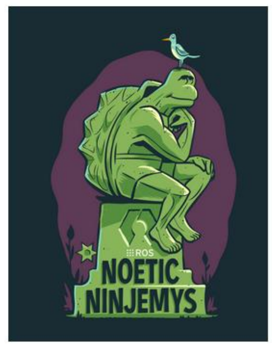
3. 29. ROS installation method

3. 29. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The current active version of ROS 1 is as follows, the recommended version is **Noetic Ninjemys**

Active ROS 1 distributions

Recommended



Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (Recommended)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

```
http://docs.ros.org
https://wiki.ros.org/Distributions
```

2) The official installation document link of ROS 1 **Noetic Ninjemys** is as follows:

```
http://wiki.ros.org/noetic/Installation/Ubuntu
```

3) In the ROS **Noetic Ninjemys** official installation document, Ubuntu recommends using Ubuntu20.04, so please make sure that the system used by the development board is **the Ubuntu20.04 desktop system**

http://wiki.ros.org/noetic/Installation

4) Then use the script below to install ros1

```
orangepi@orangepi5:~$ install_ros.sh ros1
```

5) Before using the ROS tool, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code

Note that running the following command needs to ensure that the development board can access github normally, otherwise an error will be reported due to network problems.

The install_ros.sh script will try to modify /etc/hosts and automatically run the following commands. However, this method cannot guarantee normal access to github every time. If the following error is displayed after installing ros1 in install_ros.sh, please find other ways to allow the Linux system of the development board to access github normally, and then manually run the following Order.

https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
ERROR: error loading sources list:
The read operation timed out

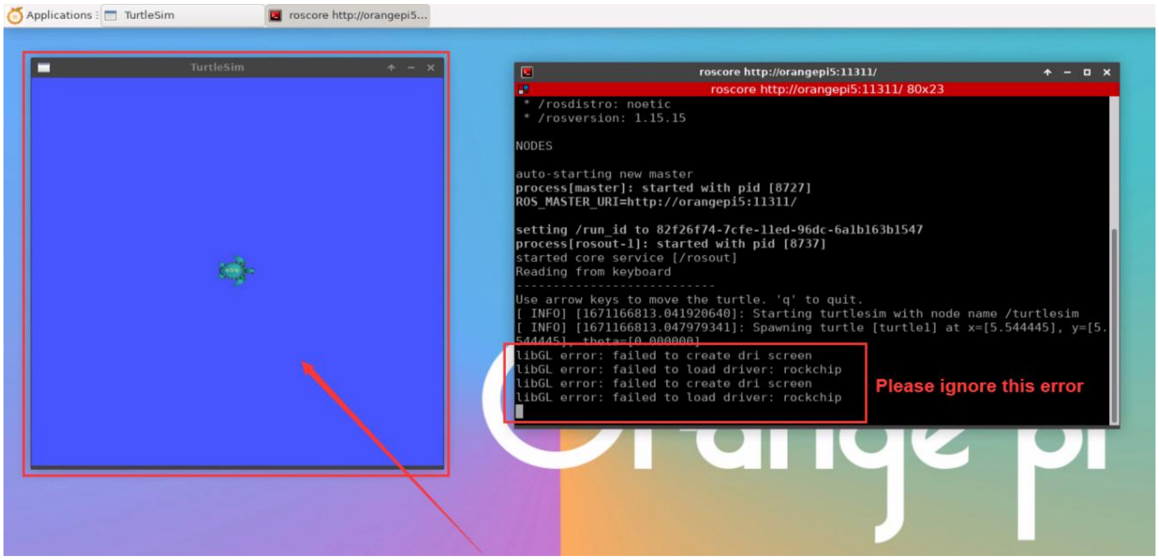
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run
```

```
            rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

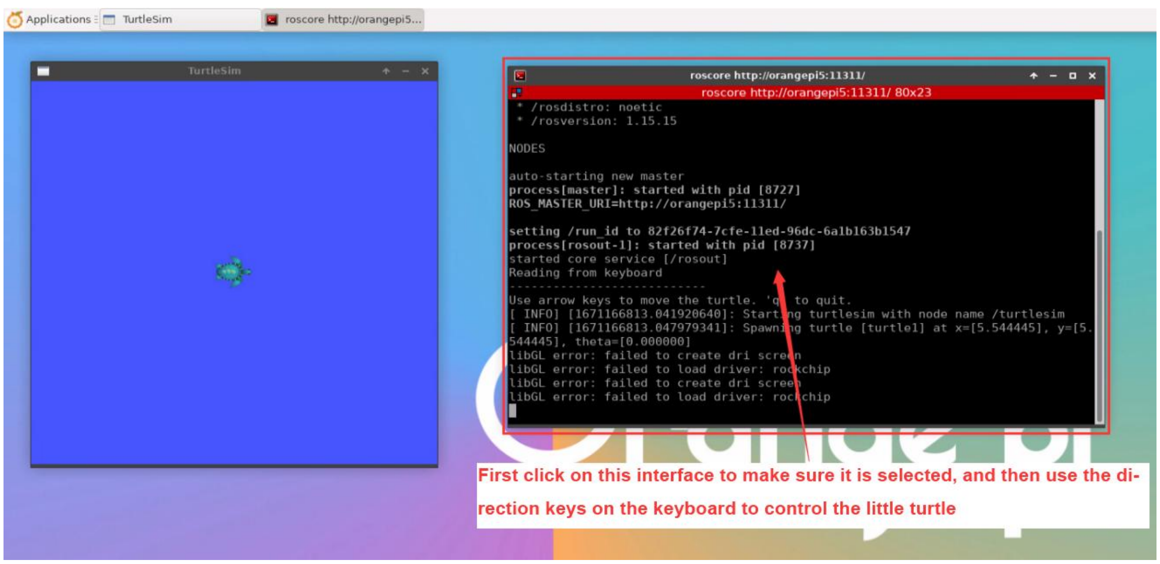
6) Then open a command line terminal window on the desktop, and then use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally

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

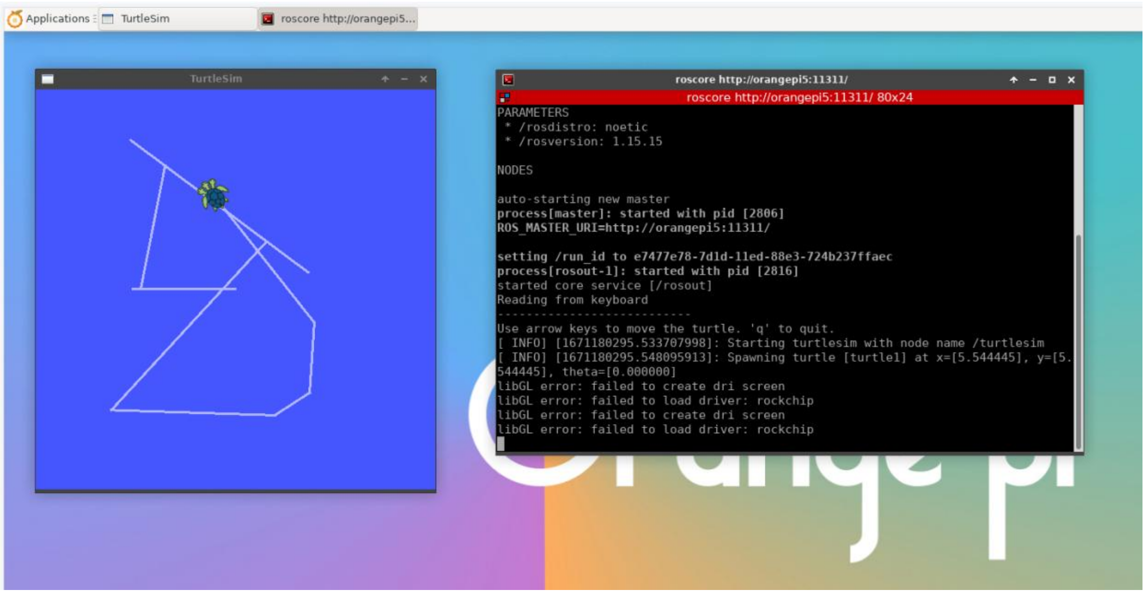
7) After running the **test_ros.sh** script, a little turtle as shown in the figure below will pop up



8) Then please keep the terminal window just opened at the top

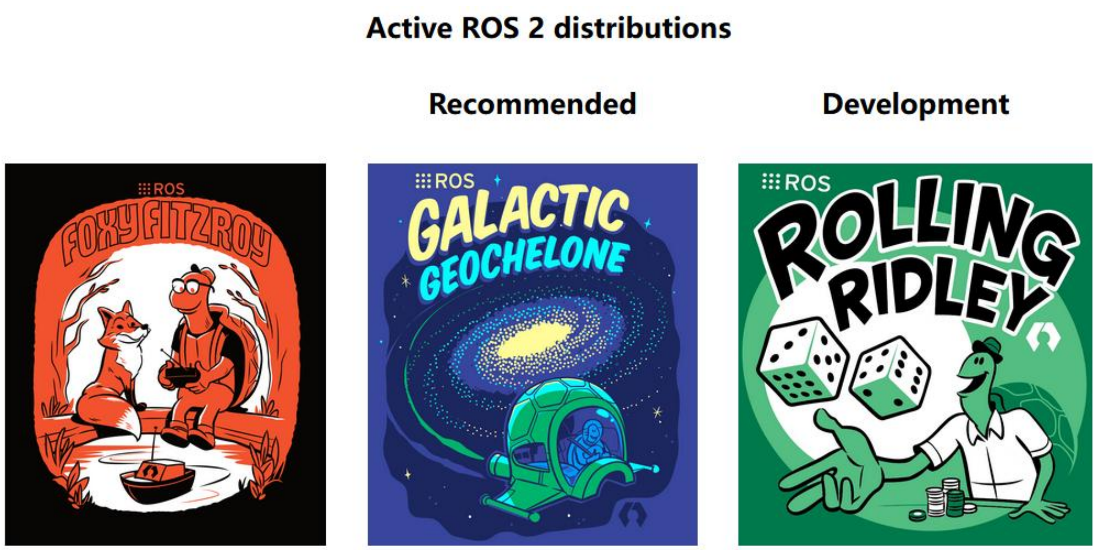


9) At this time, press the direction keys on the keyboard to control the little turtle to move up, down, left, and right



3. 29. 2. How to install ROS 2 Galactic on Ubuntu 20.04

1) The current active version of ROS 2 is as follows, the recommended version is **Galactic Geochelone**



Distro	Release date	Logo	EOL date
Humble Hawksbill	May 23rd, 2022		May 2027
Galactic Geochelone	May 23rd, 2021		November 2022
Foxy Fitzroy	June 5th, 2020		May 2023

http://docs.ros.org
http://docs.ros.org/en/galactic/Releases.html

2) The link to the official ROS 2 **Galactic Geochelone** installation documentation is as follows:

docs.ros.org/en/galactic/Installation.html
http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html

3) In the ROS 2 **Galactic Geochelone** official installation document, Ubuntu Linux recommends using Ubuntu 20.04, so please make sure that the system used by the development board is **the Ubuntu 20.04 desktop system**. There are several ways to install ROS 2. The following demonstrates how to install ROS 2 **Galactic Geochelone** through **Debian packages**

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

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

5) **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete

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

ros2 is an extensible command-line tool for ROS 2.
```

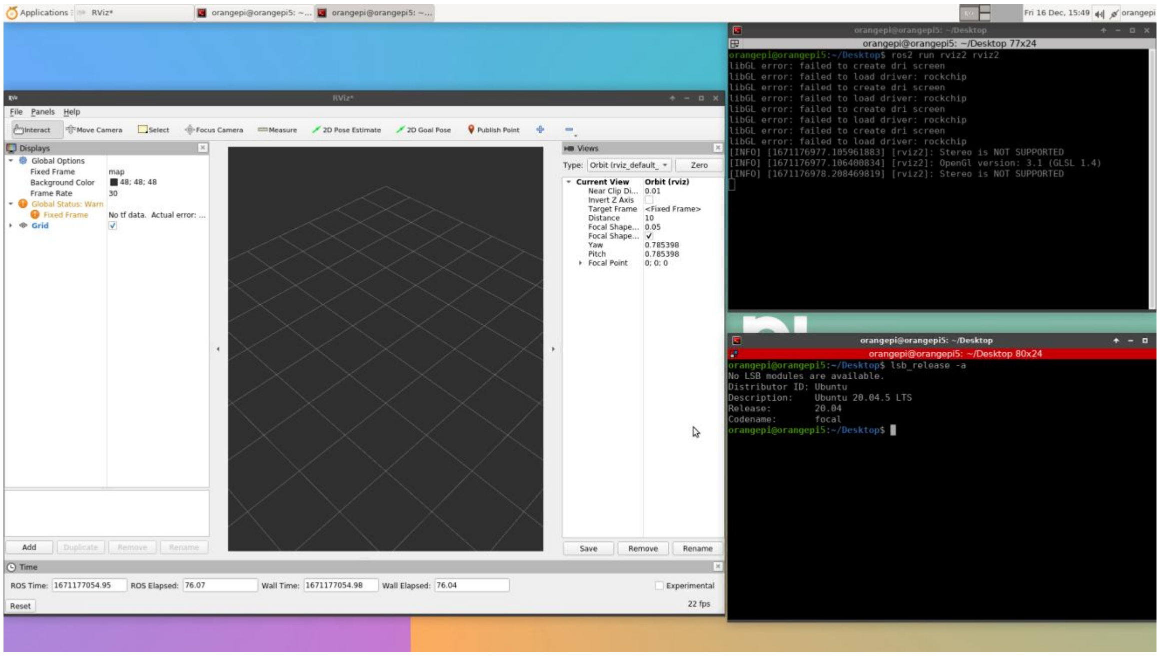
```
optional arguments:
  -h, --help            show this help message and exit

Commands:
  action     Various action related sub-commands
  bag        Various rosbag related sub-commands
  component  Various component related sub-commands
  daemon     Various daemon related sub-commands
  doctor     Check ROS setup and other potential issues
  interface  Show information about ROS interfaces
  launch     Run a launch file
  lifecycle  Various lifecycle related sub-commands
  multicast  Various multicast related sub-commands
  node       Various node related sub-commands
  param      Various param related sub-commands
  pkg        Various package related sub-commands
  run        Run a package specific executable
  security   Various security related sub-commands
  service    Various service related sub-commands
  topic      Various topic related sub-commands
  wtf        Use `wtf` as alias to `doctor`

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

6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepi5:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



8) For the usage of ROS, please refer to the documentation of ROS 2

http://docs.ros.org/en/galactic/Tutorials.html

3. 29. 3. How to install ROS 2 Humble on Ubuntu 22.04

1) Ros2 can be installed using the **install_ros.sh** script

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

2) The **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete.

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

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action     Various action related sub-commands
  bag        Various rosbag related sub-commands
```

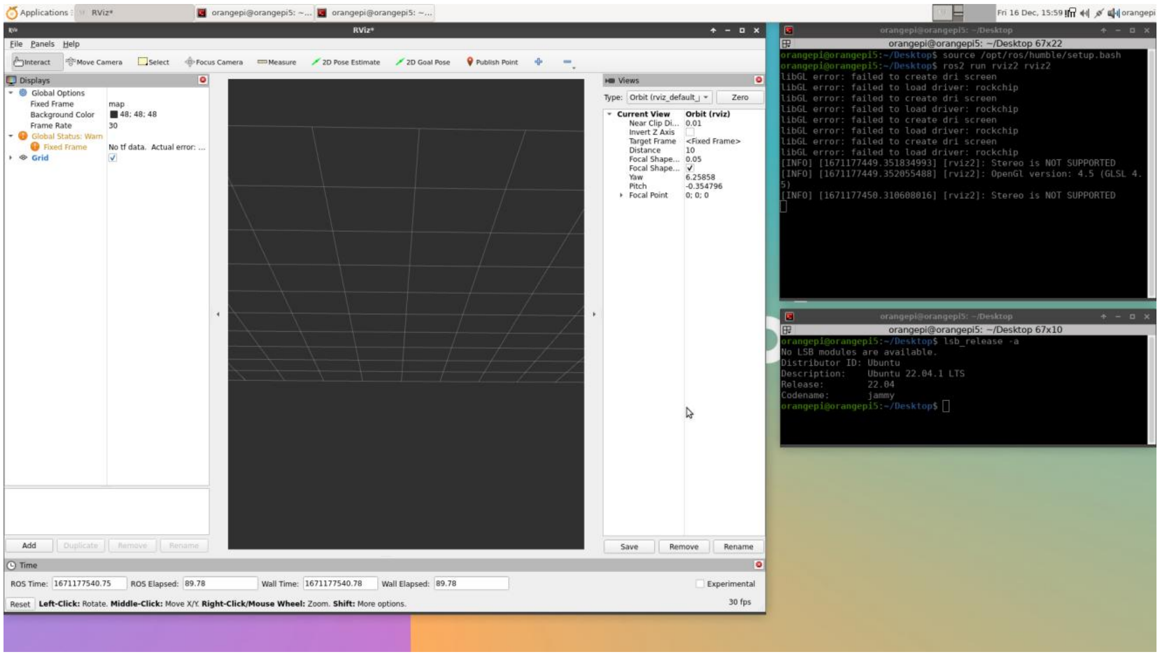
```
component    Various component related sub-commands
daemon        Various daemon related sub-commands
doctor        Check ROS setup and other potential issues
interface    Show information about ROS interfaces
launch        Run a launch file
lifecycle    Various lifecycle related sub-commands
multicast    Various multicast related sub-commands
node            Various node related sub-commands
param          Various param related sub-commands
pkg              Various package related sub-commands
run              Run a package specific executable
security      Various security related sub-commands
service       Various service related sub-commands
topic          Various topic related sub-commands
wtf              Use `wtf` as alias to `doctor`

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

3) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepi5:~$ 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. 30. The method of installing the kernel header file

1) The Linux image released by OPi comes with the deb package of the kernel header file by default, and the storage location is **/opt/**

```
orangepi@orangepi:~$ ls /opt/linux-headers*
/opt/linux-headers-legacy-rockchip-rk3588_x.x.x_arm64.deb
```

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

The name of the kernel header file deb package needs to be replaced with the actual name, please do not copy it.

```
orangepi@orangepi:~$ sudo dpkg -i /opt/linux-headers-legacy-rockchip-rk3588_1.x.x_arm64.deb
```

3) After installation, you can see the folder where the kernel header file is located under /usr/src.

```
orangepi@orangepi:~$ ls /usr/src
```

```
# If it is a system with Linux 5.10 kernel, the output is as follows
linux-headers-5.10.160-rockchip-rk3588
# If it is a system with Linux 6.1 kernel, the output is as follows
linux-headers-6.1.43-rockchip-rk3588
```

4) Then you can write a hello kernel module to test the kernel header file 可以编写一个 hello

a. First write the code of hello kernel module, as shown below:

```
orangepi@orangepi:~$ vim hello.c
#include <linux/init.h>
#include <linux/module.h>

static int hello_init(void)
{
        printk("Hello Orange Pi -- init\n");

        return 0;
}
static void hello_exit(void)
{
        printk("Hello Orange Pi -- exit\n");

        return;
}

module_init(hello_init);
module_exit(hello_exit);

MODULE_LICENSE("GPL");
```

b. Then write the makefile file that compiles the Hello kernel module, as shown below：

```
orangepi@orangepi:~$ vim Makefile
ifneq ($(KERNELRELEASE),)
obj-m:=hello.o
else
```

```
KDIR :=/lib/modules/$(shell uname -r)/build
PWD    :=$(shell pwd)
all:
	make -C $(KDIR) M=$(PWD) modules
clean:
	rm -f *.ko *.o *.mod.o *.mod *.symvers *.cmd    *.mod.c *.order
endif
```

c. Then use the make command to compile the Hello kernel module, and the output of the compilation process is shown below:

If you compile the code you copy here, if you have any problems, go to the official tool to download the source code test


```
orangepi@orangepi:~$ make
make -C /lib/modules/5.10.160-rockchip-rk3588/build M=/home/orangepi modules
make[1]: Entering directory '/usr/src/linux-headers-5.10.160-rockchip-rk3588'
  CC [M]    /home/orangepi/hello.o
  MODPOST /home/orangepi/Module.symvers
  CC [M]    /home/orangepi/hello.mod.o
  LD [M]    /home/orangepi/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.10.160-rockchip-rk3588'
```

d. After compiling, the **Hello.ko** kernel module will be generated

```
orangepi@orangepi:~$ ls *.ko
hello.ko
```

e. Use the **Insmod** command to insert the **hello.ko** kernel module into the kernel

```
orangepi@orangepi:~$ sudo insmod hello.ko
```

f. Then use the **demsg** command to view the output of the **Hello.ko** kernel module. If you can see the output instructions below, the **hello.ko** kernel module is loaded correctly

```
orangepi@orangepi:~$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
```

g. Use the **rmmod** command to uninstall the **hello.ko** kernel module

```
orangepi@orangepi:~$ sudo rmmod hello
orangepi@orangepi:~$ dmesg | grep "Hello"
```

```
[ 2871.893988] Hello Orange Pi -- init
[ 3173.800892] Hello Orange Pi -- exit
```

3. 31. How to use 10.1 inch MIPI LCD screen

3. 31. 1. 10.1 -inch MIPI screen assembly method

1) First prepare the required accessories

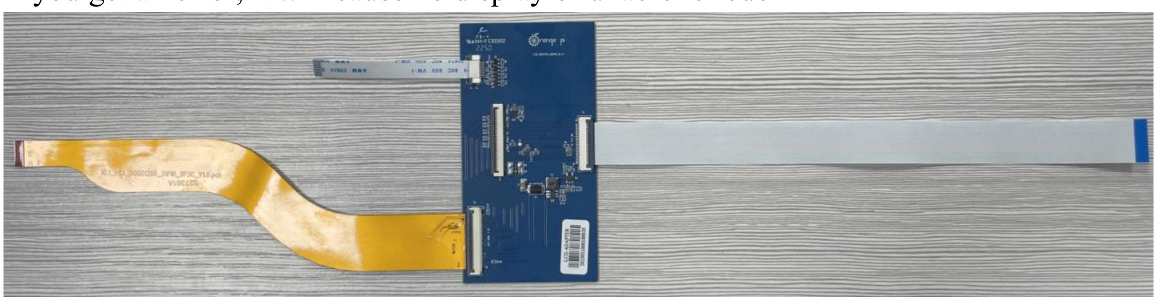
a. 10.1 -inch MIPI LCD display+touch screen

b. Screen divert plate+31pin to 40pin line

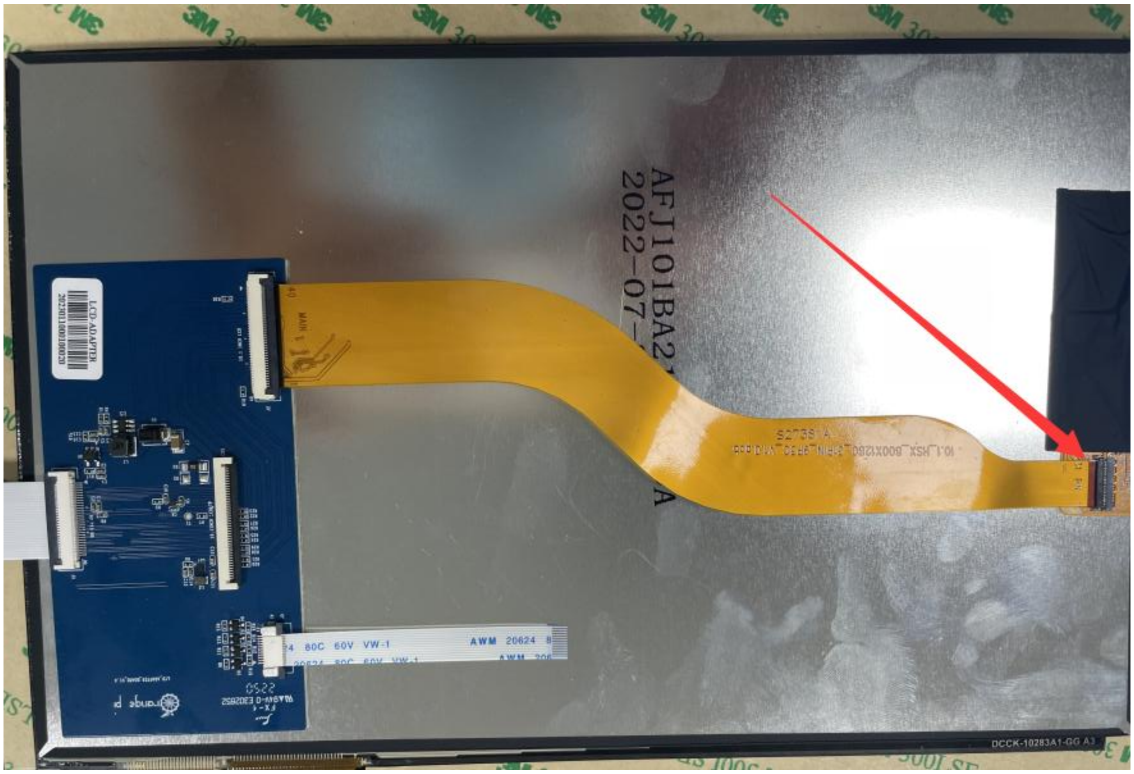
c. 30pin MIPI line

d. 12pin touch screen row line

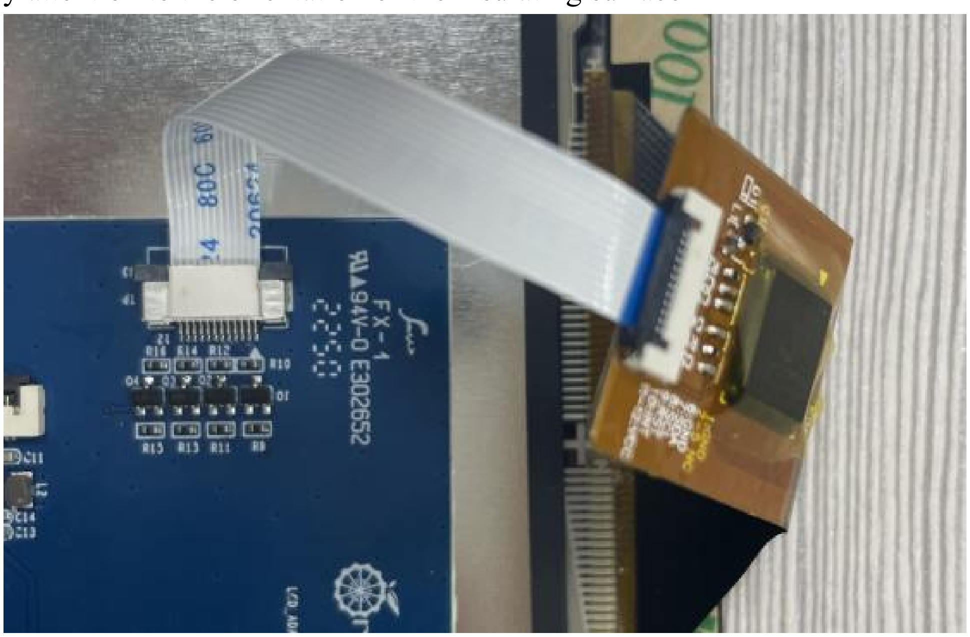
2) According to the figure below, the 12PIN touch screen row, 31PIN to 40PIN ducts, and 30pin MIPI cables get on the screen dial board. **Pay attention to the blue insulation face of the touch screen row,** the other two lines of the line insulation faces are facing up, If you get an error, it will cause no display or unable to touch



3) Place the connected rotor connected to the puzzle on the MIPI LCD screen according to the figure below, and connect the MIPI LCD screen and the rotary board through 31PIN to 40Pin row.



4) Then connect the touch screen and the rotor board through the 12PIN touch screen line, pay attention to the orientation of the insulating surface



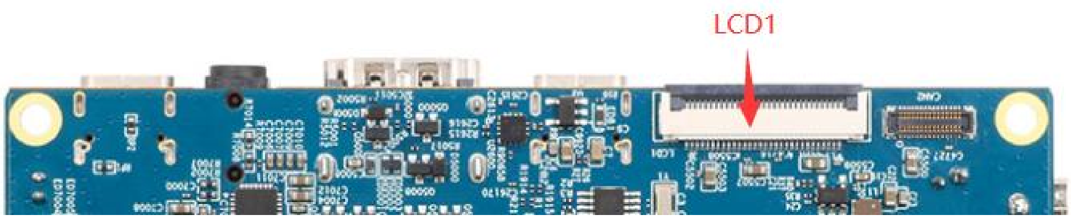
5) Finally connect to the LCD interface of the development board through the 30PIN MIPI duct

3. 31. 2. Open the 10.1 -inch MIPI LCD screen configuration method

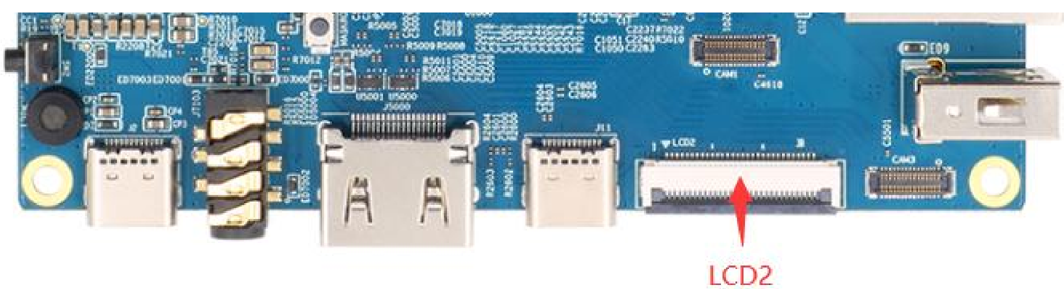
1) The Linux image closed to the configuration of the mipi lcd screen by default. If you need to use the mipi lcd screen, you need to open it manually.

2) There are two interfaces of the mipi lcd screen on the development board, we define：

a. The location of the lcd1 interface is：



b. The position of the lcd2 interface is：



V.1.1.4 and V.1.1.4 The previous version of the Linux image,The configuration of the LCD DTBO and the definition above are reversed. Please pay attention when using it

V.1.1.6 and V.1.1.6 later version of the Linux image changed the configuration of the lcd dtbo, which is consistent with the lcd serial number displayed on the

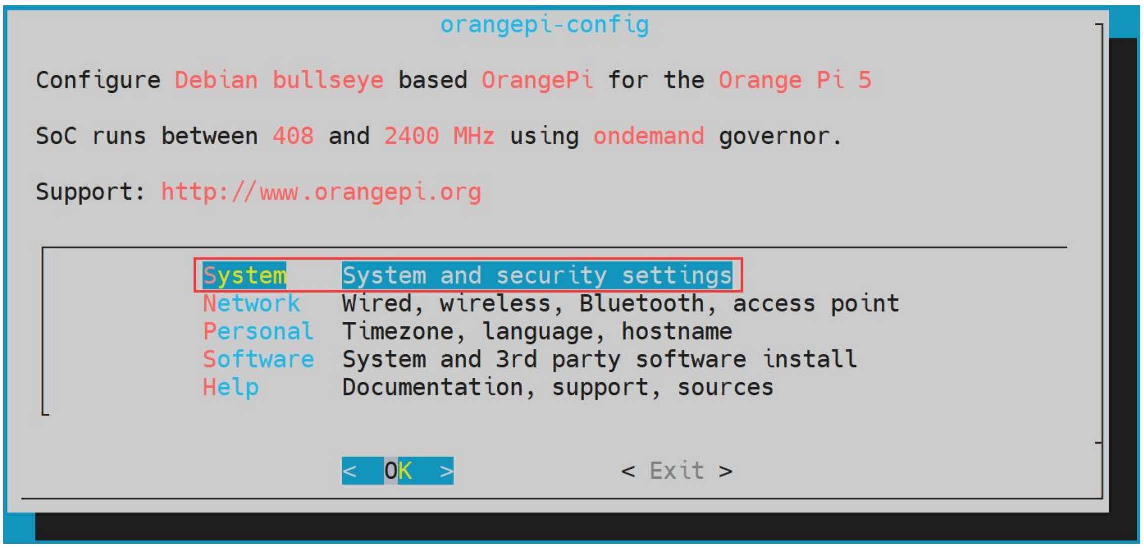
development board

3) The steps of opening the mipi lcd configuration are shown below:

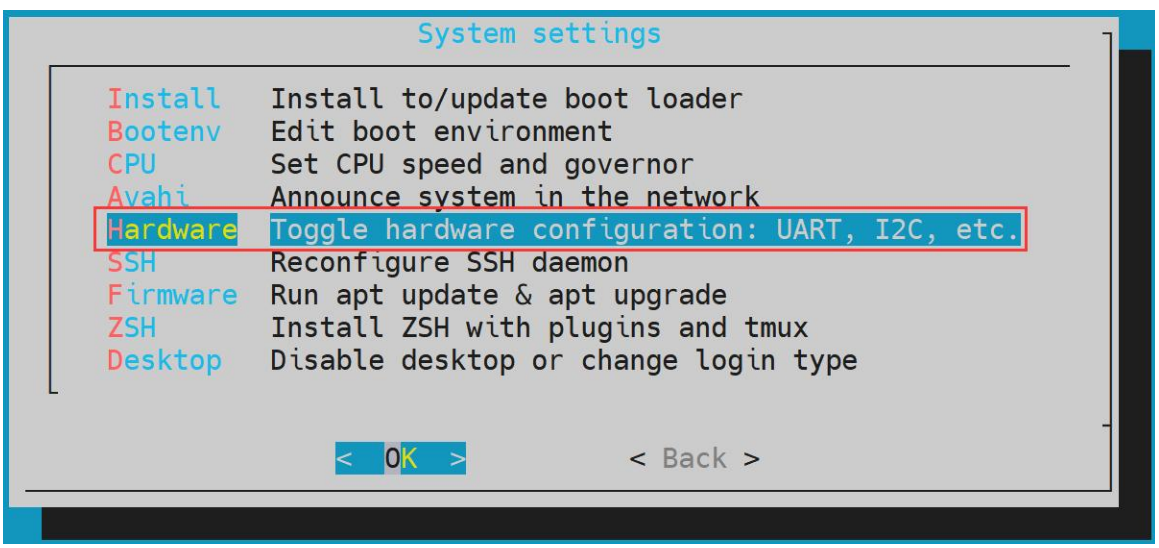
a. First run the **orangepi-config**, ordinary users remember to add **sudo** permissions

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

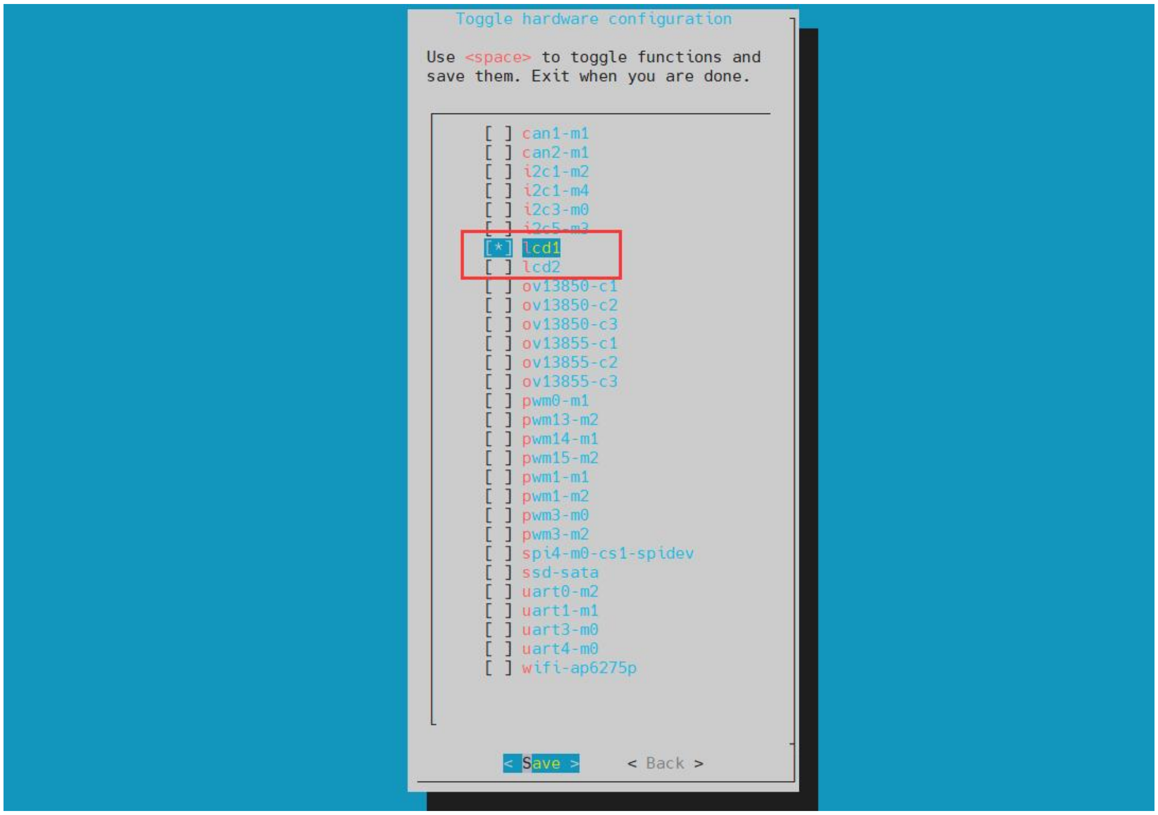
b. Then choose **System**



c. Then choose **Hardware**

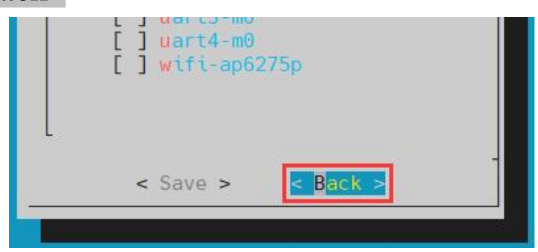


d. Then use the direction keys of the keyboard to position lcd1 or lcd2 (to open which one if you want to use it, and two screens can be opened at the same time), then use the **space** to select

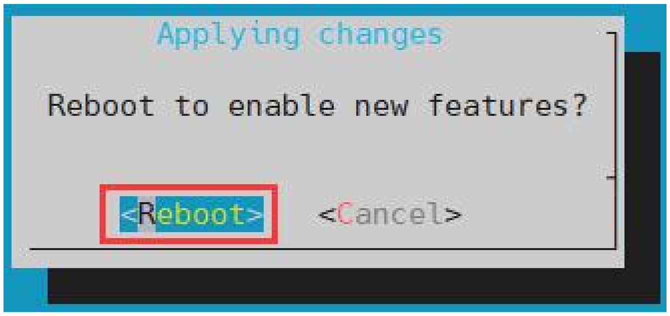


e. Then select **<Save>**

f. Then select **<back>**



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



The above settings will eventually add overlays=lcd1 or overlays=lcd2 or overlays=lcd1 lcd2 in /boot/orangepiEnv.txt. You can check it first after setting. If this configuration does not exist, then there is a problem with settings

If you think it is more troublesome to use orangepi-config, you can also use vim editors to open/boot/orangepiEnv.txt, and then add overlays=lcd1 or overlays=lcd2 or overlays=lcd1 lcd2 is also OK

```
orangepi@orangepi:~$ cat /boot/orangepiEnv.txt | grep "lcd"
overlays=lcd1          #Sample configuration
```

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

3. 31. 3. The server version of the image rotation display direction method

1) Add **extraargs = fbcon = rotate: the direction to rotate** in **/boot/orangepiEnv.txt,**This configuration can set the direction of the linux system displayed by the server version. Among them, **fbcon=rotate:** The following numbers can be set to：

a. 0: Normal screen (default vertical screen)
b. 1: Turn 90 degrees clock
c. 2: Flip 180 degrees
d. 3: Turn to 270 degrees clock

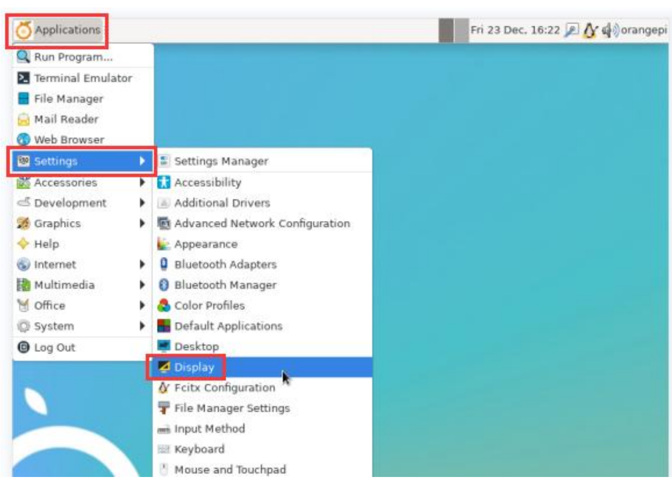
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=lcd1
extraargs=cma=64M fbcon=rotate:3
```

Note that if/boot/orangepienv.txt is configured in the default default exiArgs = CMA = 64M configuration, fbcon = rotate: 3 This configuration can be added to extraargs = cma = 64m (need to be separated by spaces)

2) Then **restart** the Linux system to see that the direction of the LCD screen display has been rotated

3. 31. 4. The method of rotating and touching the desktop image

1) First open **Display** settings in the Linux system



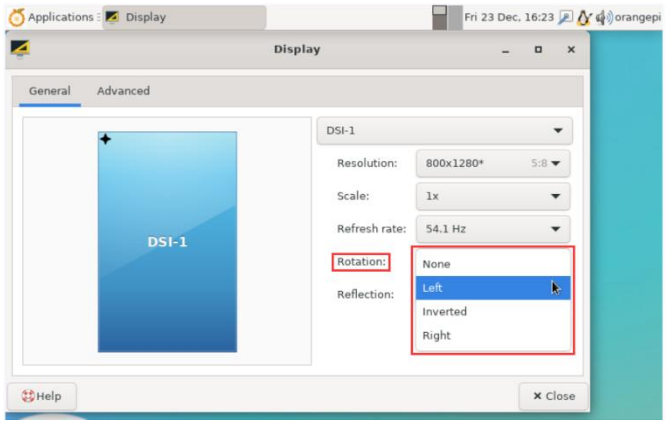
2) Then select the direction you want to rotate in the **Rotation**

a. **None**: Not rotate

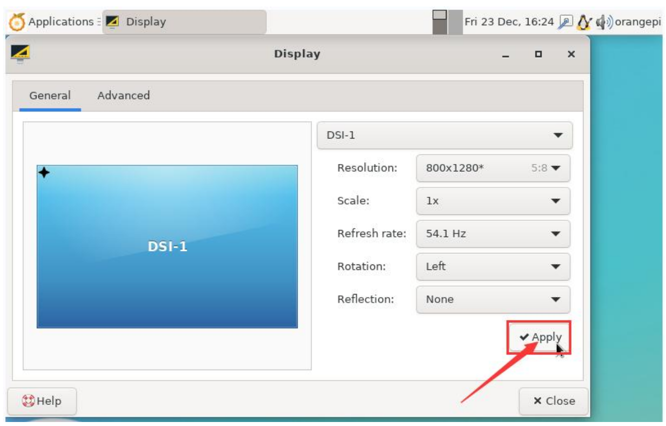
b. **Left**: Rotate 90 degrees to the left

c. **Inverted**: Flipting up and down, equivalent to rotating 180 degrees

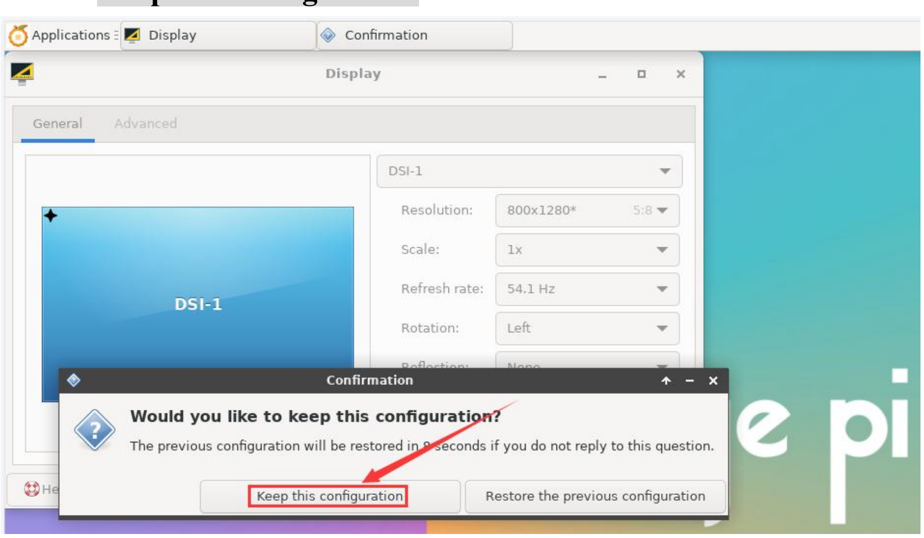
d. **Right**: Rotate 90 degrees to the right



3) Then click **Apply**



4) Then choose **Keep this configuration**



5) At this time, the screen display has been rotated, and then the **Display** program is turned off.

6) The above steps will only select the display direction, and it will not rotate the direction of touch. Use **set_lcd_rotate.sh** script to rotate the direction of touch. After this script is set, it will be automatically restarted,Then you can test whether the touch can be used normally.

a. **None**: Not rotate

```
orangepi@orangepi:~$ set_lcd_rotate.sh none
```

b. **Left**: Rotate 90 degrees to the left

```
orangepi@orangepi:~$ set_lcd_rotate.sh left
```

c. **Inverted**: Flipting up and down, equivalent to rotating 180 degrees

```
orangepi@orangepi:~$ set_lcd_rotate.sh inverted
```

d. **Right**: Rotate 90 degrees to the right

```
orangepi@orangepi:~$ set_lcd_rotate.sh right
```

set_lcd_rotate.sh script mainly does four things:

1. **Rotate the direction displayed by Framebuffer**
2. **The direction of rotating touch**
3. **Turn off the boot logo**
4. **Restart the system**

The direction of the rotation touch is achieved by adding Option "TransformationMatrix""x x x x x x x x x" in /usr/share/X11/xorg.conf.d/40-libinput.conf,Among them, "x x x x x x x x x" is different in different directions

7) Touch rotation reference materials

```
https://wiki.ubuntu.com/X/InputCoordinateTransformation
```

3. 32. Instructions for opening the logo use

1) The default logo is displayed by default in the desktop version of the system

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the switch to the logo

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=false
```

3) Set the **bootlogo** variable to **true** in**/boot/orangepiEnv.txt** to turn the turn -off logo

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
```

4) The location of the logo picture in the Linux system is

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

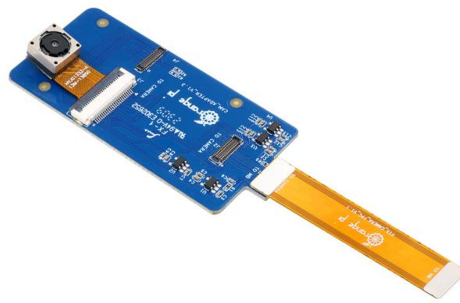
3. 33. OV13850 and OV13855 MIPI test methods for testing methods

Please note that in Linux 6.1 system, in order to ensure that the 3A service can run normally and obtain normal camera images, the Docker service needs to be disabled. If the Docker service is not disabled, the image captured by the camera will not contain the 3A effect and will appear as a dark image. The method to disable the Docker service is as follows:

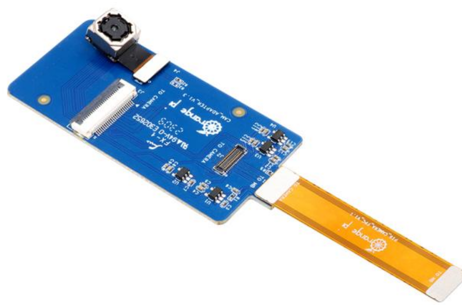
```
orangepi@orangepi:~$ sudo systemctl disable docker.socket docker.service containerd.service
orangepi@orangepi:~$ sudo reboot
```

At present, the development board supports two MIPI cameras, OV13850 and OV13855. The specific pictures are shown below：

a. OV13850 camera at13 MP MIPI interface



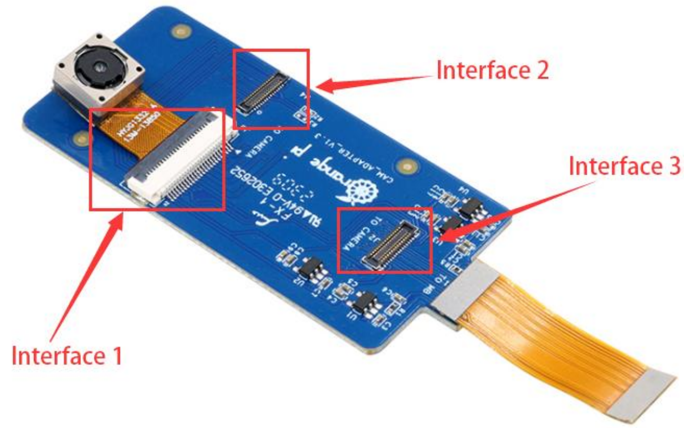
b. OV13855 camera at 13MP MIPI interface



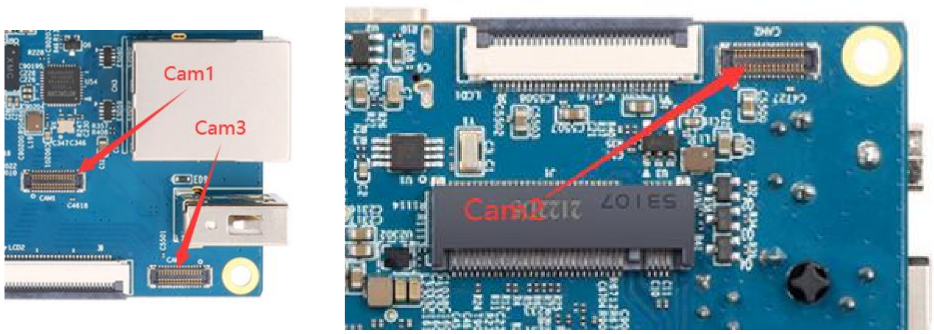
The rotary board and the FPC cable used by OV13850 and OV13855 cameras is the same, but the two cameras are different from the position on the rotary board. The FPC exhaust line is shown in the following figure. Please note that the FPC line is directed. The end is marked **TO MB** that it needs to be inserted into the camera interface of the development board. The another end is marked **TO CAMERA** that the end of the Camera needs to be inserted to the camera transfer board.。

There are a total of 3 cameras on the camera to connect to the board, which can only be used at the same time, as shown in the figure below：

a. **1 interface is connected to the OV13850 camera**
b. **2 interface OV13855 camera**
c. 3 interface is not used, just ignore it



Orange Pi 5 has a total of 3 camera interfaces on the development board. We define the positions of Cam1, Cam2 and Cam3 as shown in the figure below：



The method of the Cam1 interface inserted in the camera is shown below：

The method of the Cam2 interface inserted in the camera is shown below：

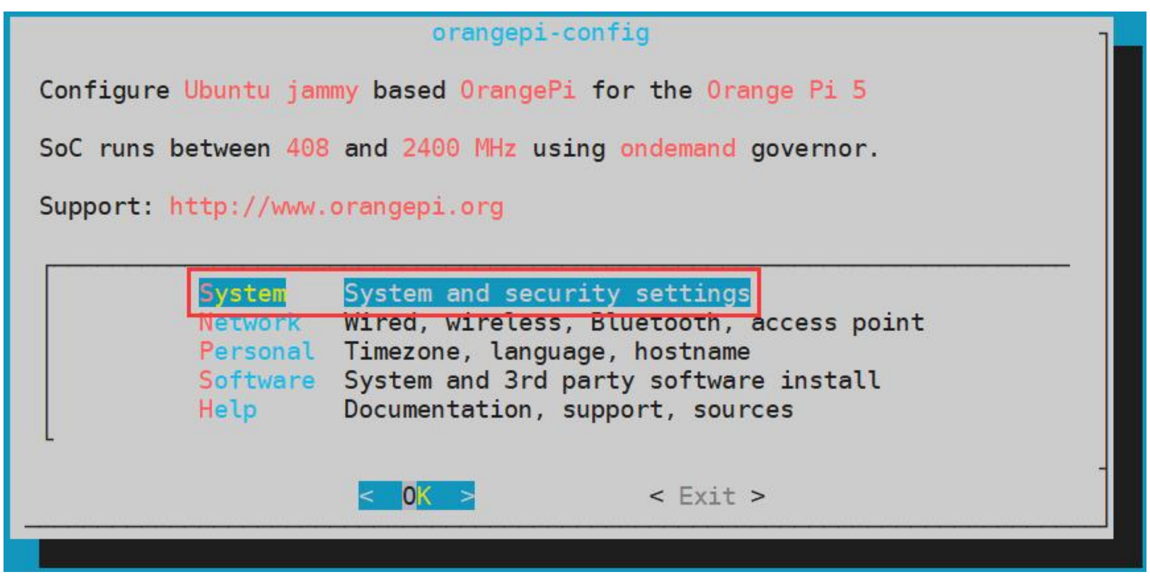
The method of the Cam3 interface inserted in the camera is shown below:

After connecting the camera to the development board, we can use the following method to test the camera:

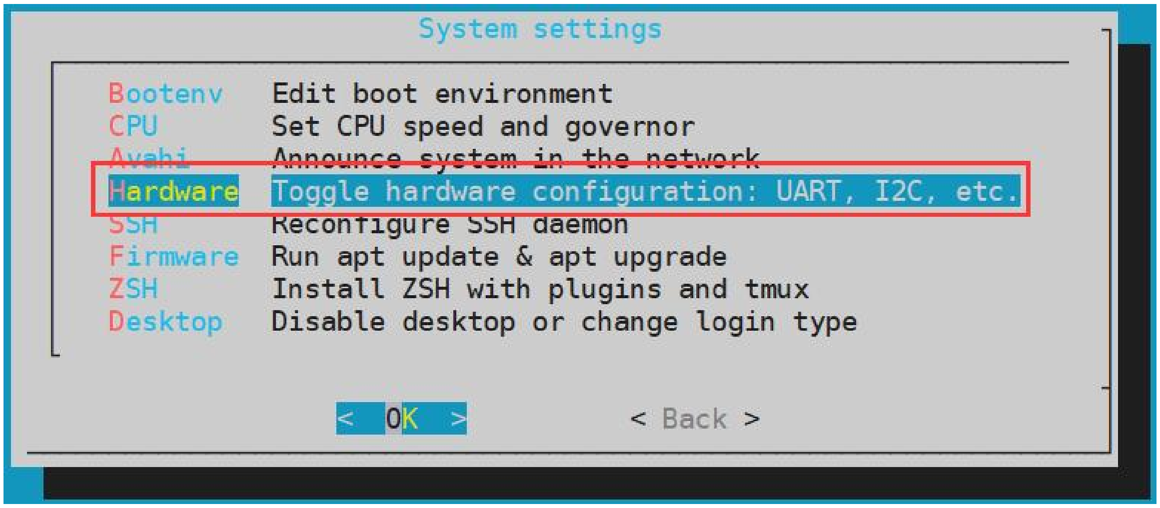
a. First run the **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

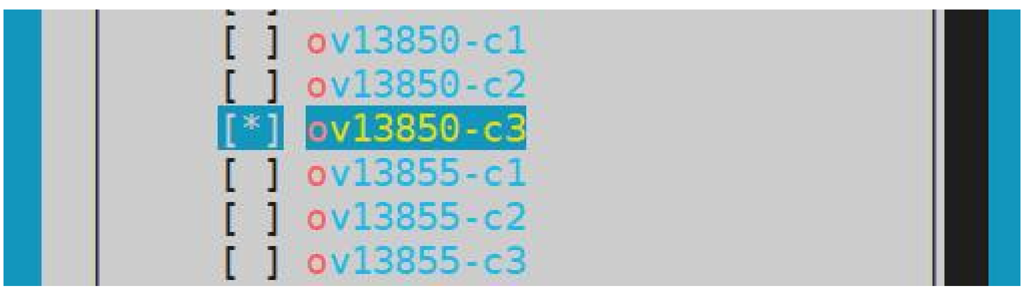
b. Then choose **System**



c. Then choose **Hardware**

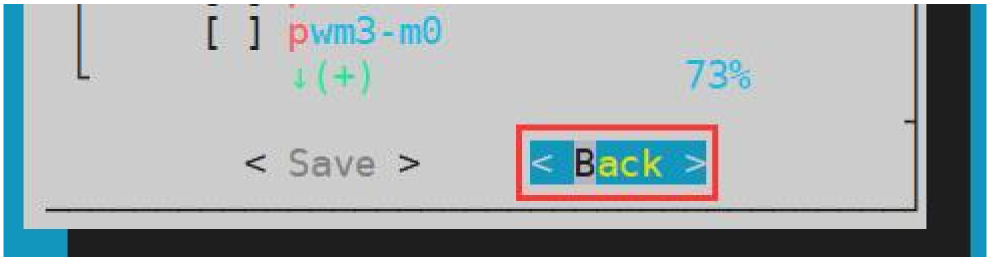


d. Then use the direction key of the keyboard to position the position shown in the figure below, and then use the **space** to select the camera you want to open. Among them, **ov13850-c1** indicates that the OV13850 camera is used in the CAM1 interface of the development board. **ov13855-c2** indicates that the OV13855 camera is used in the CAM2 interface of the development board, and other configurations can be pushed.

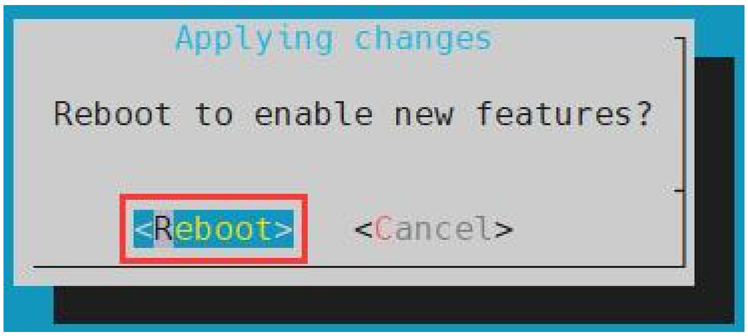


e. Then select **<Save>**

f. Then select **<Back>**



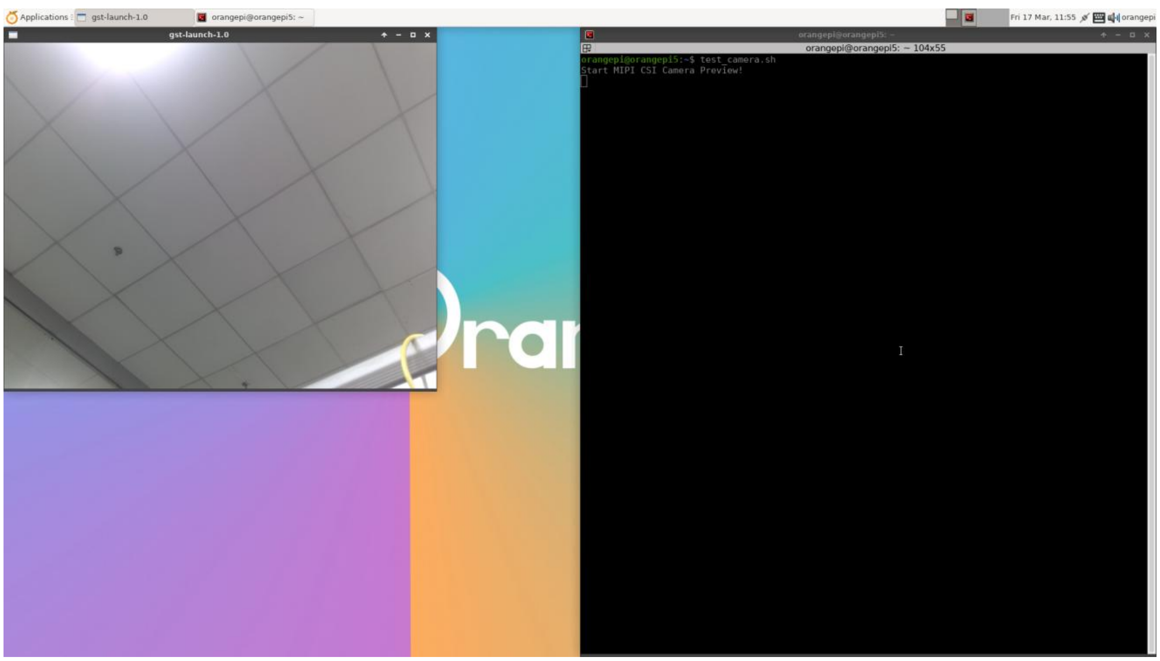
g. Then select**<Reboot>**Restart the system to make the configuration effective



h. Then open a terminal in the desktop system and run the script below

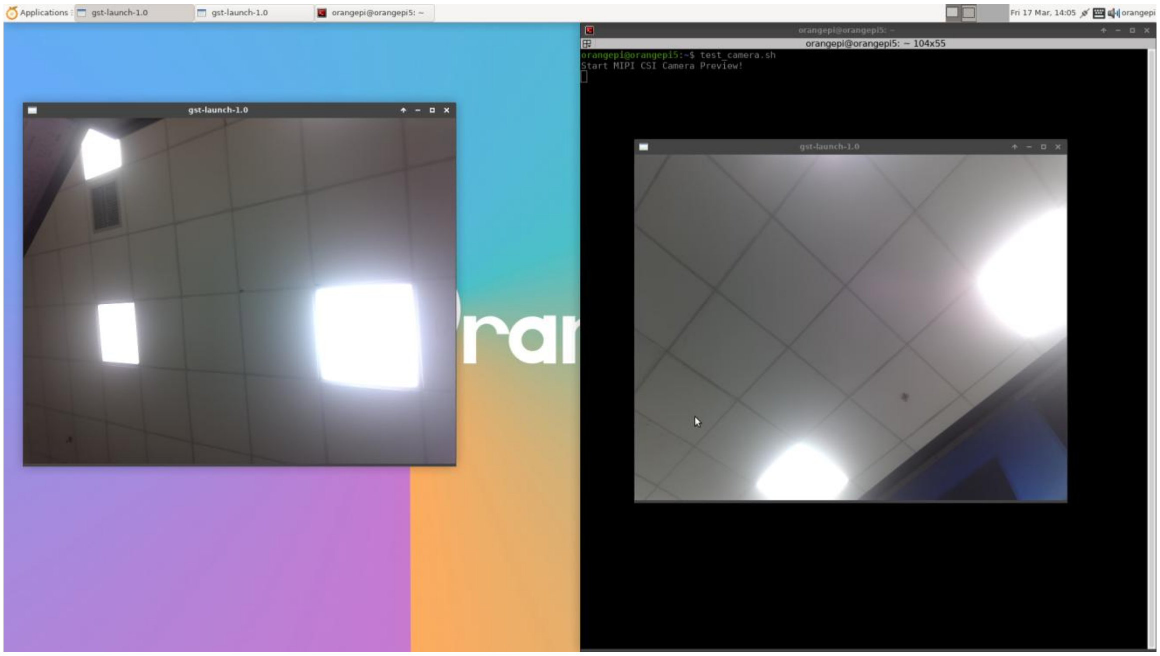
```
orangepi@orangepi:~$ test_camera.sh
```

i. Then you can see the preview screen of the camera



In addition to single cameras, we can also use two cameras at the same time. It should be noted that the current test dual camera should be used for combinations of **Cam1+Cam3** (supporting OV13850 and OV13855 mix and match). After connect the dual camera, like the previous steps, open the configuration of the Cam1+Cam3 through

orangepi-config, restart the system, and then open the terminal on the desktop to run the **test_camera.sh** script to see the preview screen of the two cameras, as follows Shown in the figure：



Please refer to the link below for the camera dts configuration. If you need it, you can modify it by yourself；

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-camera1.dtsi

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-camera2.dtsi

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-camera3.dtsi

dt overlay configuration is in the directory below：
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/overlay

3. 34. How to use the ZFS file system

3. 34. 1. How to install ZFS

Before installing zfs, please make sure that the Linux image you are using is the latest version. In addition, if zfs is already installed in the system, there is no need to install it again.

Before installing zfs, you first need to install the kernel header files. For how to install the kernel header files, please refer to the instructions in the section "**How to Install the Kernel Header Files**."

In Ubuntu20.04, Ubuntu22.04 and Debian11 systems, zfs cannot be installed directly through apt. This is because the zfs version in the default apt source is lower than 2.1.6 and is incompatible with the rk linux5.10 kernel. This problem is fixed in zfs 2.1.6 and later versions

In order to solve this problem, we provide a deb package of zfs that can be installed normally, which can be downloaded from the official tool of the development board. Open the **official tool** and enter **the zfs-related deb package folder used by Ubuntu and Debian systems**. You can see three types of deb packages for Ubuntu20.04, Ubuntu22.04 and Debian11. Please download the required version.

After downloading the corresponding version of the zfs deb package, please upload them to the Linux system of the development board. For the upload method, please refer to the instructions in the Methods of Uploading Files to the Development Board Linux System.

After the upload is completed, use the **cd** command on the command line of the development board Linux system to enter the directory of the deb package, and then use the following command to install the zfs deb package.

```
orangepi@orangepi:~$ sudo apt install ./*.deb
```

After the installation is complete, use the following command to see the zfs-related kernel modules:

```
orangepi@orangepi:~$ ls /lib/modules/5.10.160-rockchip-rk3588/updates/dkms/
icp.ko   spl.ko   zavl.ko   zcommon.ko   zfs.ko   zlua.ko   znvpair.ko   zunicode.ko
zzstd.ko
```

If it is a Linux 6.1 system, use the following command to see the zfs-related kernel modules:

```
orangepi@orangepi:~$ ls /lib/modules/6.1.43-rockchip-rk3588/updates/dkms/
icp.ko   spl.ko   zavl.ko   zcommon.ko   zfs.ko   zlua.ko   znvpair.ko   zunicode.ko
zzstd.ko
```

Then restart the Linux system and you will see that the zfs kernel module will be automatically loaded:

```
orangepi@orangepi:~$ lsmod | grep "zfs"
zfs                    2801664  0
zunicode                327680  1 zfs
zzstd                   471040  1 zfs
zlua                    139264  1 zfs
zcommon                  69632  1 zfs
znvpair                  61440  2 zfs,zcommon
zavl                     16384  1 zfs
icp                     221184  1 zfs
spl                      77824  6 zfs,icp,zzstd,znvpair,zcommon,zavl
```

In Debian12, the default version of zfs is 2.1.11, so we can install zfs directly through the following command. Again, we need to make sure that the system has the deb package of the kernel header file installed before installation.

```
orangepi@orangepi:~$ sudo apt install -y zfsutils-linux zfs-dkms
```

3. 34. 2. How to create a ZFS pool

ZFS is based on storage pools. We can add multiple physical storage devices to the pool and then allocate storage space from this pool.

The following content is demonstrated based on the development board being connected to an NVMe SSD and a USB flash drive.

1) First, we can use the **lsblk** command to view all storage devices on the development board. Currently, the development board is connected to an NVMe SSD and a USB flash drive. The output is as follows:

```
orangepi@orangepi:~$ lsblk
NAME         MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda            8:0    1  28.8G  0 disk
├─sda1         8:1    1  28.8G  0 part
└─sda9         8:9    1    8M  0 part
mtdblock0     31:0    0    16M  0 disk
mmcblk0      179:0    0  29.7G  0 disk
├─mmcblk0p1  179:1    0     1G  0 part /boot
└─mmcblk0p2  179:2    0  28.4G  0 part /var/log.hdd
                                         /
zram0        254:0    0   7.7G  0 disk [SWAP]
zram1        254:1    0   200M  0 disk /var/log
nvme0n1      259:0    0 476.9G  0 disk
├─nvme0n1p1  259:3    0 476.9G  0 part
└─nvme0n1p9  259:4    0     8M  0 part
orangepi@orangepi:~$
```

2) Then enter the following command to create a ZFS pool, including two storage devices: NVMe SSD and USB flash drive.

orangepi@orangepi:~$ **sudo zpool create -f pool1 /dev/nvme0n1 /dev/sda**

3) Then use the **zpool list** command to see that the system has created a ZFS pool named **pool1**, and the size of the ZFS pool pool1 is the size of the NVME SSD plus the size of the USB flash drive.

```
orangepi@orangepi:~$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G   114K   504G        -         -     0%     0%  1.00x    ONLINE  -
```

4) Then execute **df -h** and you can see that **pool1** is mounted to the **/pool1** directory.

```
orangepi@orangepi:~$ df -h
Filesystem          Size   Used Avail Use% Mounted on
tmpfs               1.6G    18M  1.6G   2% /run
/dev/mmcblk0p2       29G   6.0G   22G  22% /
tmpfs               7.7G    46M  7.7G   1% /dev/shm
tmpfs               5.0M   4.0K  5.0M   1% /run/lock
tmpfs               7.7G   944K  7.7G   1% /tmp
/dev/mmcblk0p1     1022M   115M  908M  12% /boot
/dev/zram1          188M   4.5M  169M   3% /var/log
tmpfs               1.6G    80K  1.6G   1% /run/user/1000
```

```
pool1           489G   9.3M   489G    1% /pool1
```

5) Use the following command to see that the file system type of pool1 is zfs

```
orangepi@orangepi:~$ mount | grep pool1
pool1 on /pool1 type zfs (rw,xattr,noacl)
```

6) Then we can test copying a file to the ZFS pool

```
orangepi@orangepi:~$ sudo cp -v /usr/local/test.mp4 /pool1/
'/usr/local/test.mp4' -> '/pool1/test.mp4'
```

3. 34. 3. Test the data deduplication function of ZFS

1) The data deduplication function of ZFS is turned off by default. We need to execute the following command to turn it on.

```
orangepi@orangepi:~$ sudo zfs set dedup=on pool1
```

2) Then do a simple test, first enter pool1, and then execute the following command to generate a random file of 1G size

```
orangepi@orangepi:~$ cd /pool1/
root@orangepi:/pool1$ sudo dd if=/dev/urandom of=test.1g bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 5.04367 s, 213 MB/s
```

3) Then use the following command to copy 1000 copies of a random file of 1G size

```
root@orangepi:/pool1$ for ((i=0; i<1000; i++)); do sudo cp test.1g $i.test.1g; done
```

4) Then use **du -lh** to see that there is currently a total of 1002G of data in the pool, but in fact the size of the ZFS pool is only **504GB** (the total capacity of SSD + U disk), which cannot hold such large data.

```
root@orangepi:/pool1$ du -lh
1002G
```

5) Then use the **zpool list** command to see that only 1.01G is actually occupied, because these 1001 files are duplicates, indicating that the data deduplication function is effective.

```
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  1.01G   503G        -         -     0%     0%  6.00x    ONLINE  -
```

3. 34. 4. Test the data compression function of ZFS

1) Because the stored data is different, the disk space saved by compression will also be different, so we choose to compress relatively large plain text files for compression testing. Execute the following command to package the **/var/log/** and **/etc/** directories. into tarball

orangepi@orangepi:~$ **cd /pool1/**
root@orangepi:/pool1$ **sudo tar -cf text.tar /var/log/ /etc/**

2) Then the file size that can be seen through the **ls -lh** command and the space occupied in the ZFS pool are both **27M**

```
orangepi@orangepi:/pool1$ ls -lh
total 27M
-rw-r--r-- 1 root root 27M Jun  1 14:46 text.tar
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  26.7M   504G        -         -     0%     0%  1.00x    ONLINE  -
orangepi@orangepi:/pool1$
```

3) Then we enable compression in ZFS pool pool1

root@orangepi:/pool1$ **sudo zfs set compression=lz4 pool1**

4) Then execute the following command again to package the **/var/log/** and **/etc/** directories into a tar package

root@orangepi:/pool1$ **sudo tar -cf text.tar /var/log/ /etc/**

5) At this time, you can see that the **text.tar** file size is still 27M, but it only occupies 9.47M space in the ZFS pool, indicating that the file is compressed.

```
orangepi@orangepi:/pool1$ ls -lh
total 9.2M
-rw-r--r-- 1 root root 27M Jun  1 14:54 text.tar
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  9.47M   504G        -         -     0%     0%  1.00x    ONLINE  -
```

3. 35. How to install and use CasaOS

CasaOS is an open source home cloud system based on the Docker ecosystem, which allows you to run a variety of home applications on your own development board, such as NAS, home automation, media servers, etc.

There are many problems with installing CasaOS in Debian12, please do not use this method to install.

3. 35. 1. CasaOS installation method

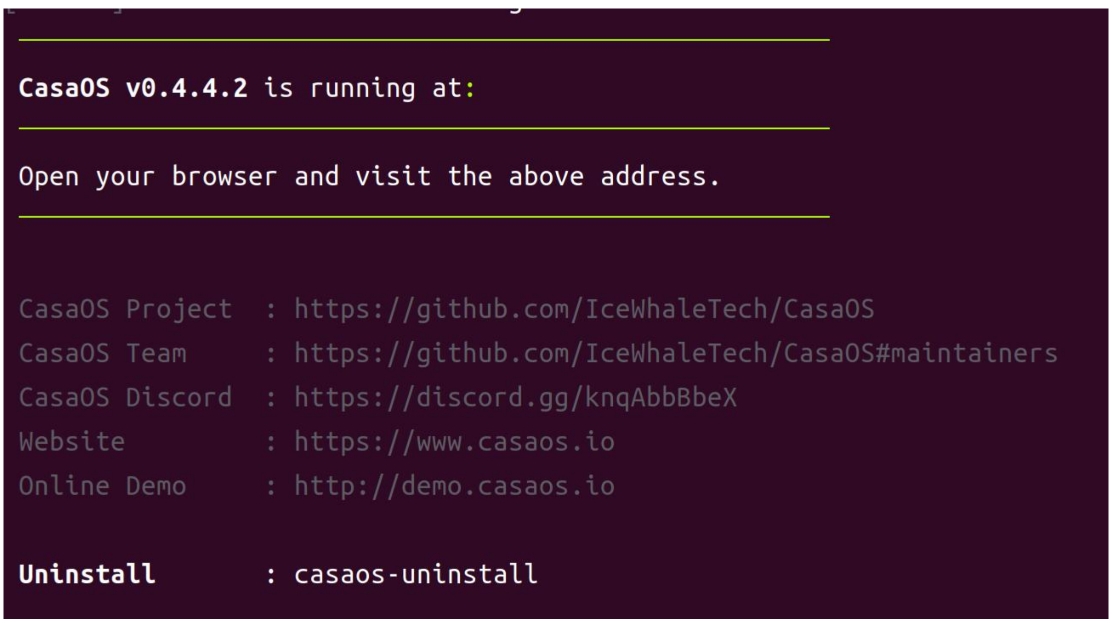
1) First you need to install docker. Docker is already pre-installed in the system released by Orangepi Pi. This step can be skipped. You can use the following command to check the version of docker installed.

```
orangepi@orangepi:~$ docker --version
Docker version 24.0.2, build cb74dfc          # Ubuntu Jammy system output
```

2) Then enter the following command in the linux system to start the installation of CasaOS

```
orangepi@orangepi:~$ curl -fsSL https://get.casaos.io | sudo bash
```

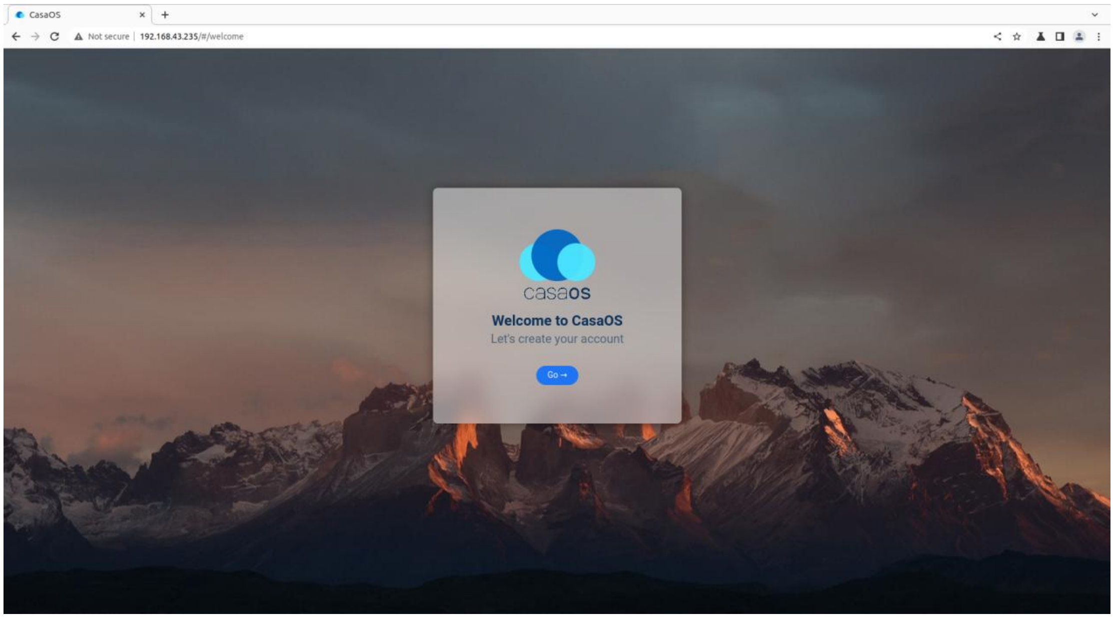
3) When you see the terminal outputting the following print information, it means that CasaOS has been installed.



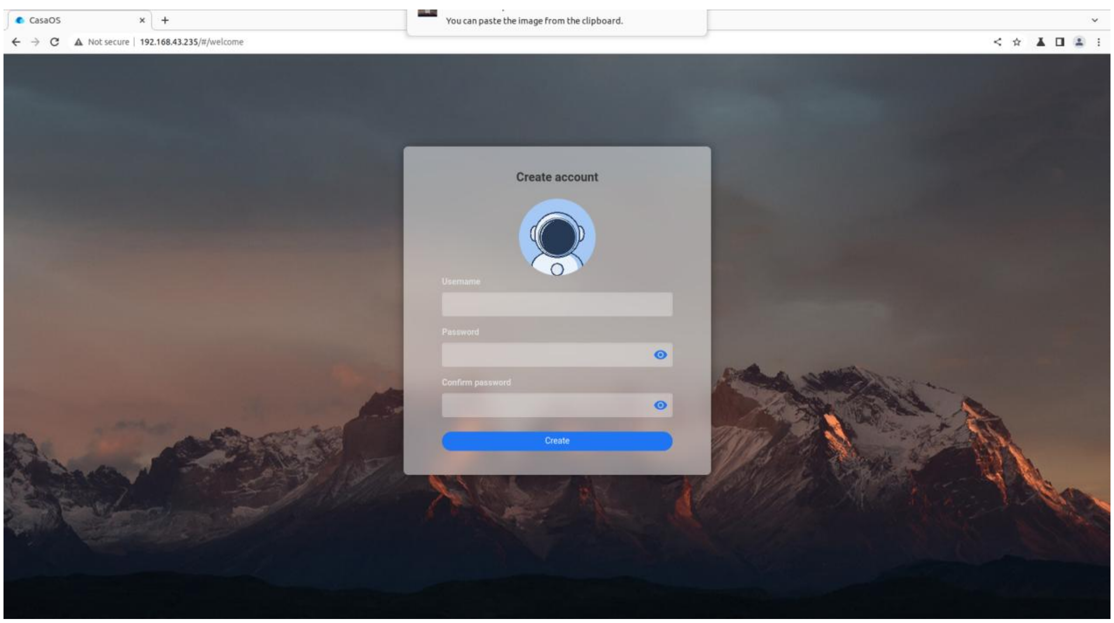
3. 35. 2. How to use CasaOS

1) After installing CasaOS, enter **http://the IP address of the development board** in the browser to open CasaOS

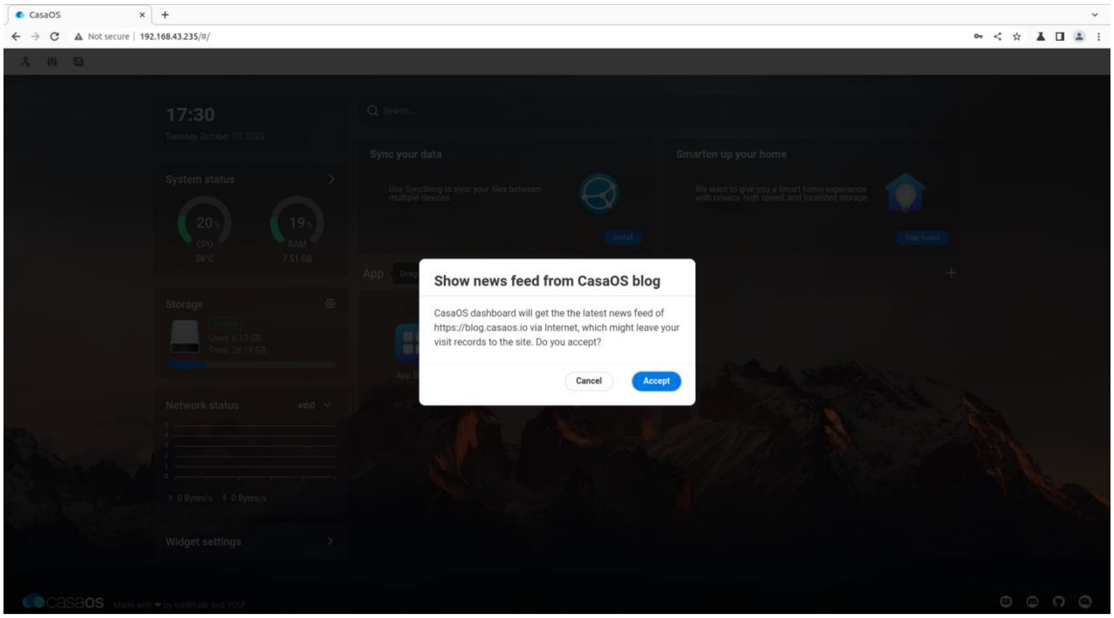
2) After opening CasaO, the following welcome interface will pop up. Click "Go" to proceed to the next step.



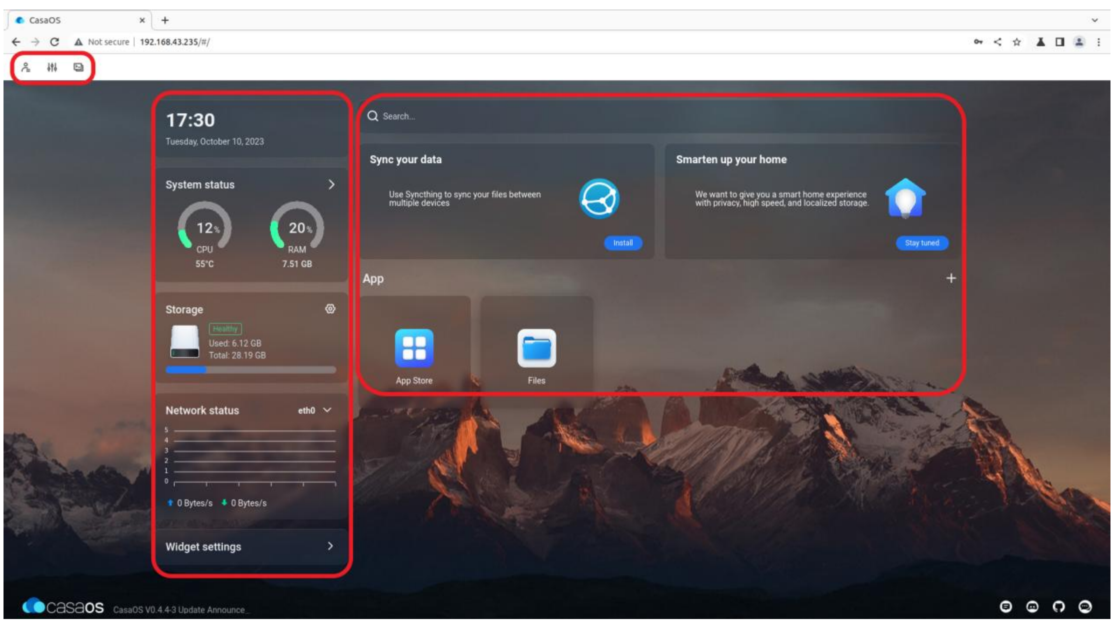
3) When you log in to CasaOS for the first time, the login interface is the interface for setting the account and password. When you log in again, only the interface for entering the account and password will appear. After setting the account and password, click "Create" to proceed to the next step.



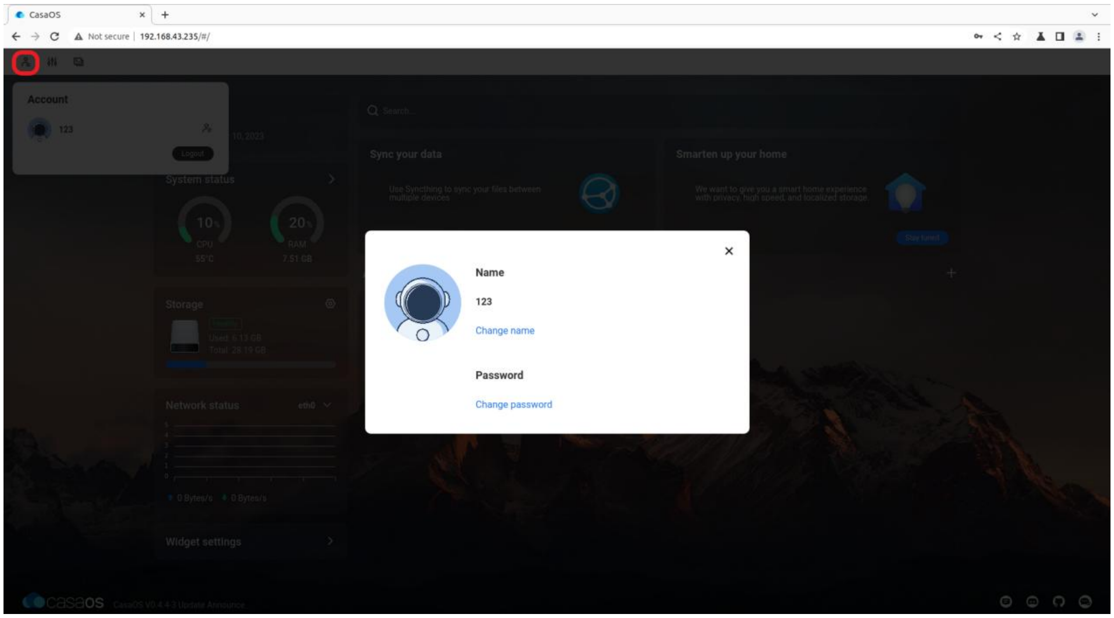
4) Click "Accept" directly in the interface below to proceed to the next step.



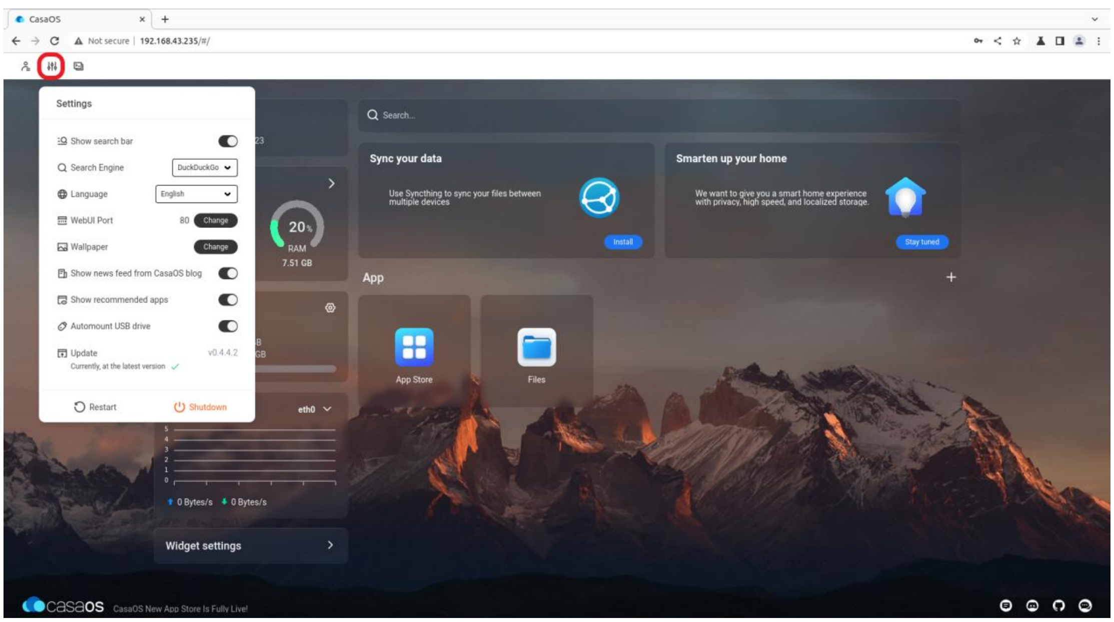
5) Now enter the main page of CasaOS. There are three icons in the upper left corner for function settings. On the left is the performance panel, which can display the current time and status information of CPU, RAM, storage, and network. On the right is the function panel. It has functions such as search, application recommendation, application store and file management.



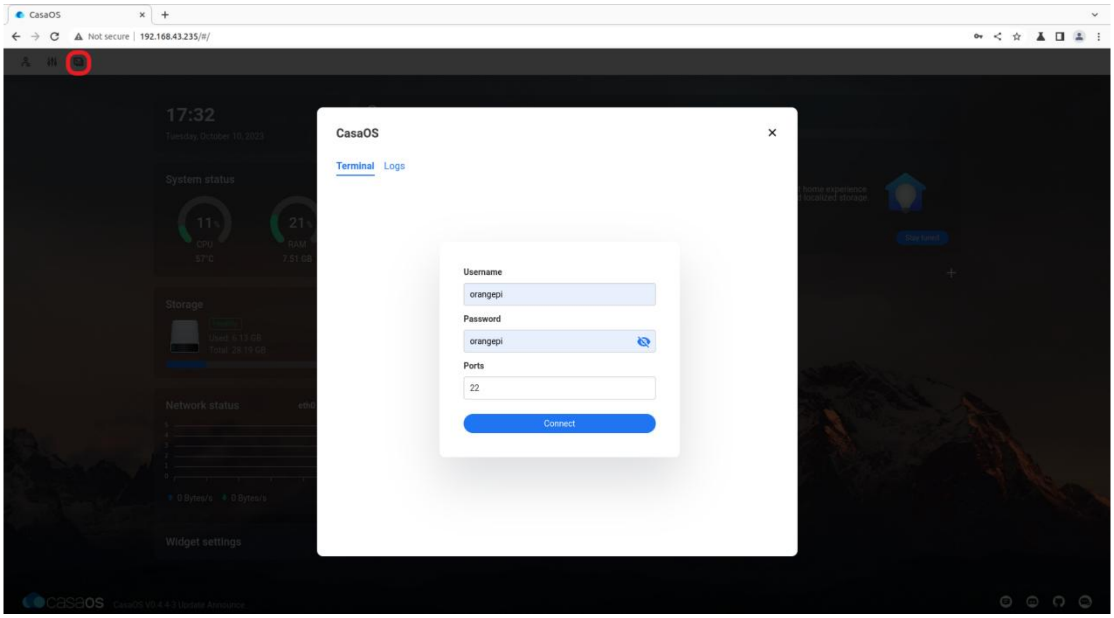
6) You can click the first icon in the upper left corner to modify the account number and password



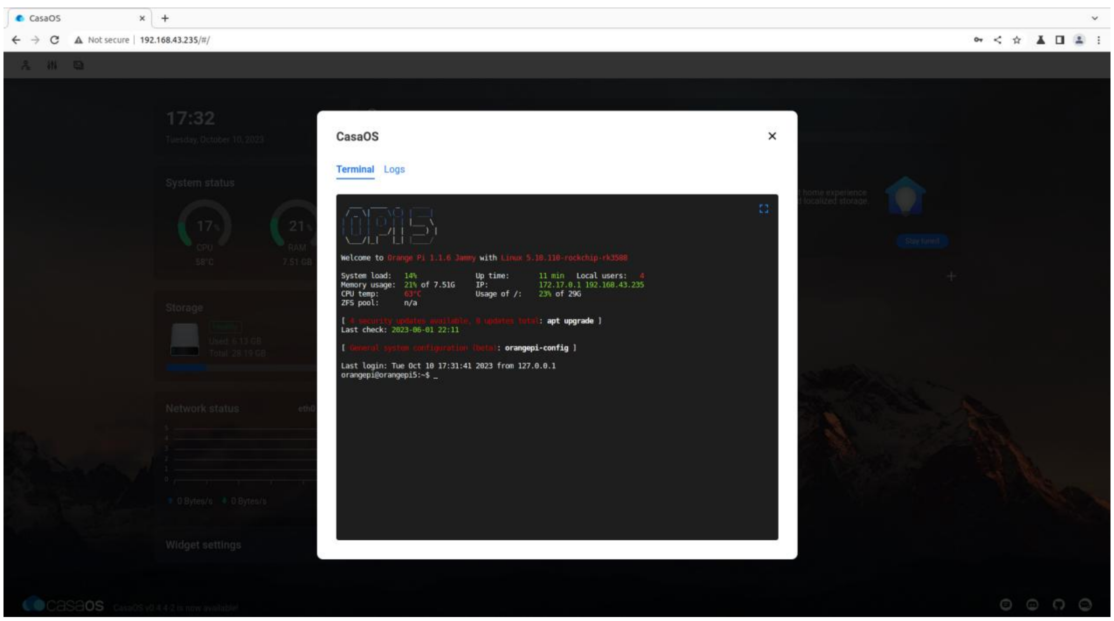
7) You can click the second icon to set basic functions



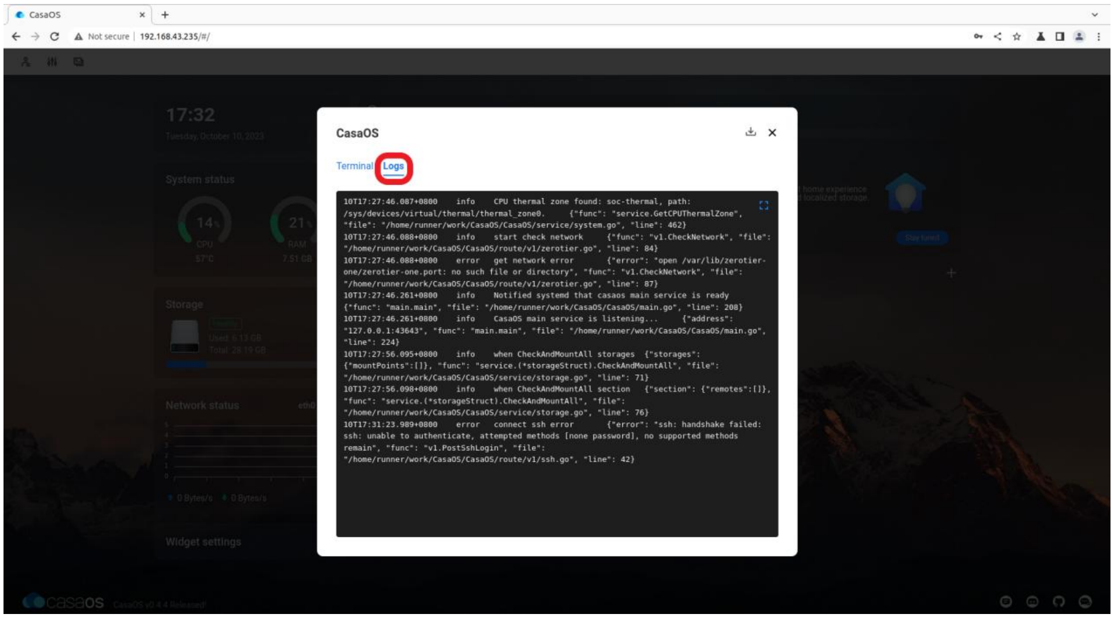
8) The third icon in the upper left corner mainly has two functions, namely switching to command line mode and printing log information. When switching to command line mode, you need to enter your account and password. The account and password here refer to the development board. Linux system account and password, the port system defaults to number 22



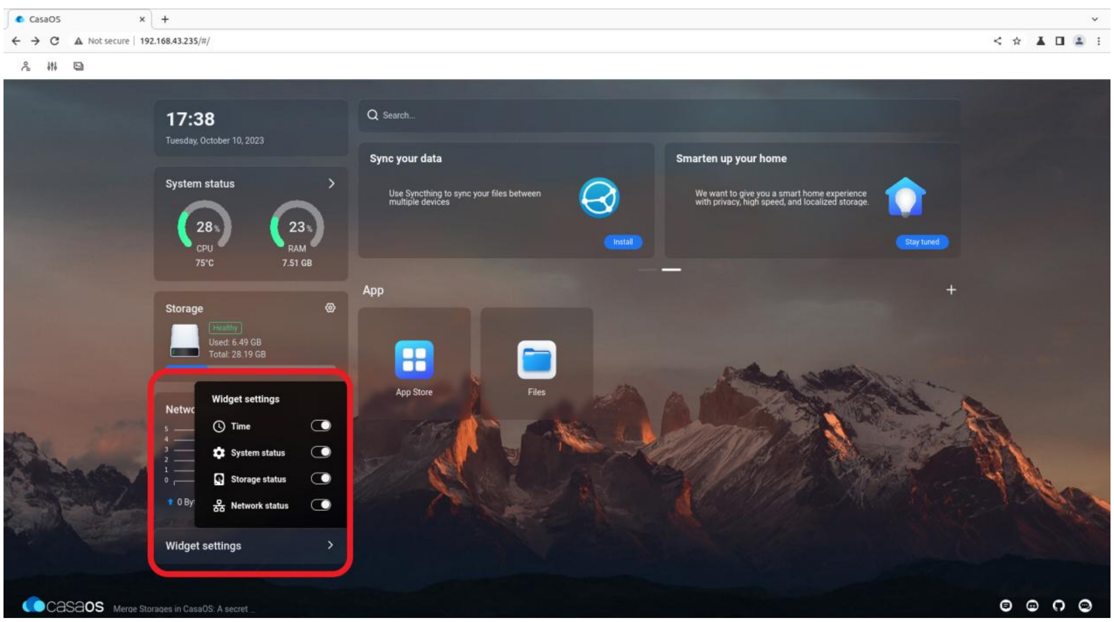
9) Then click "Connect" to enter the command line interface:



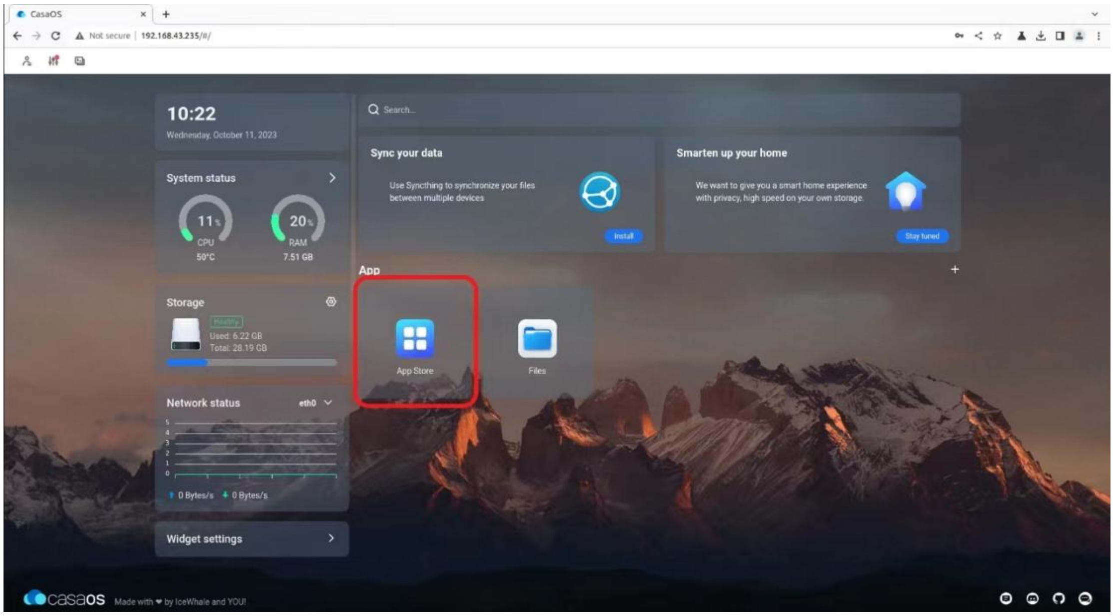
10) Another function under the third icon is to print CasaOS logs. Click "Logs" to enter. The interface is as follows:



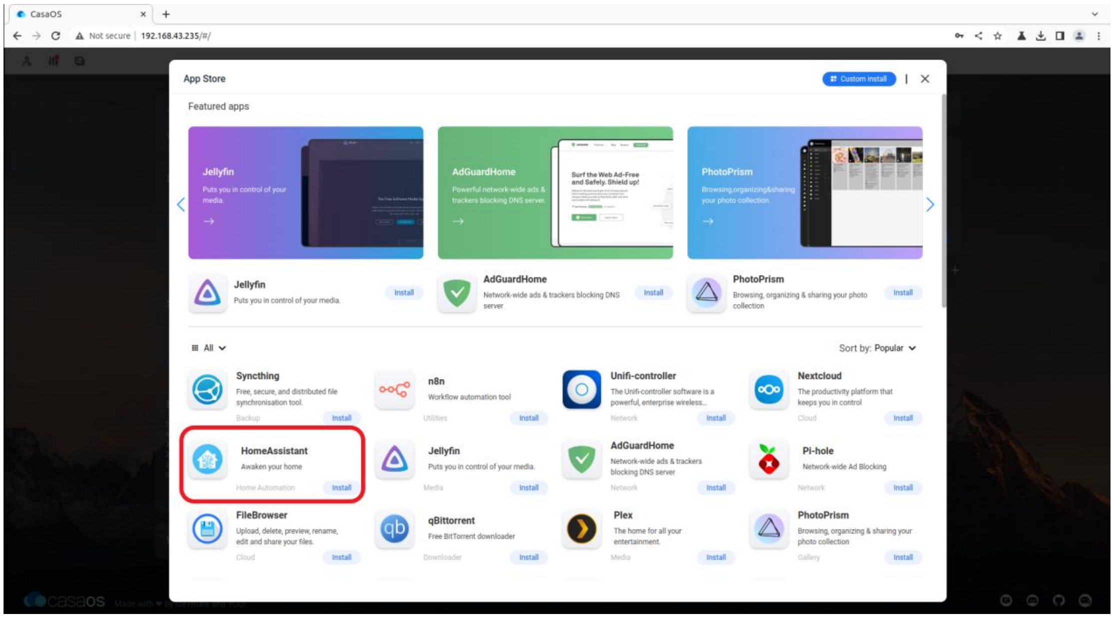
11) Click "Widget settings" in the lower left corner to set whether to display the widgets of the performance panel on the main page.



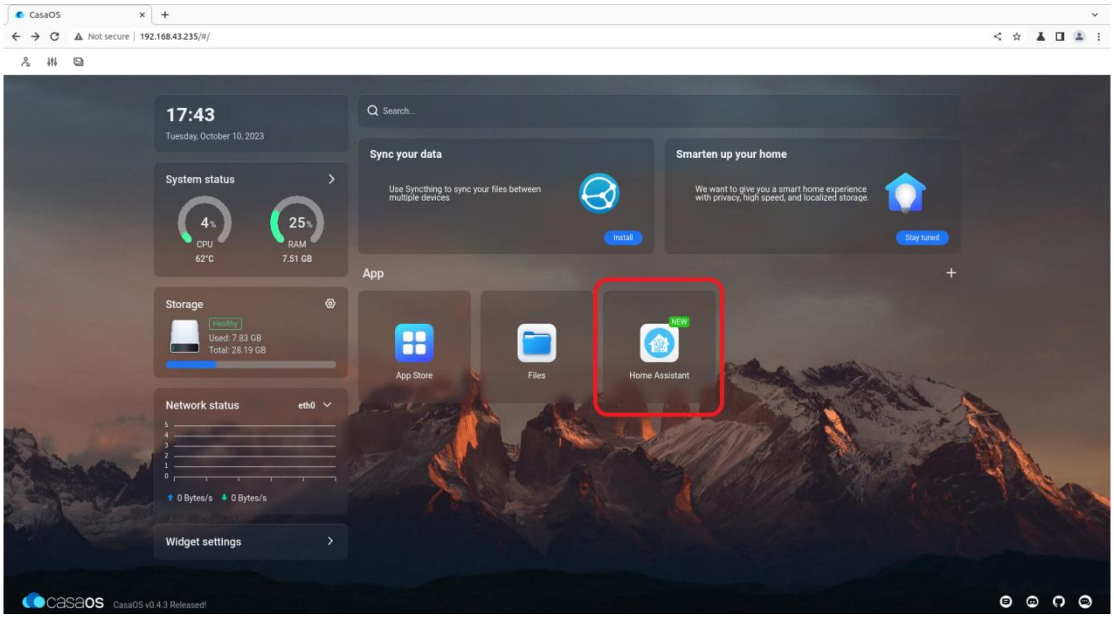
12) Click "APP Store" on the main interface to open the app store. Currently, there are a total of 70+ APPs available in the app store.



13) Here we take Home Assistant as an example to download, find Home Assistant in the APP Store, and then click the corresponding "install"

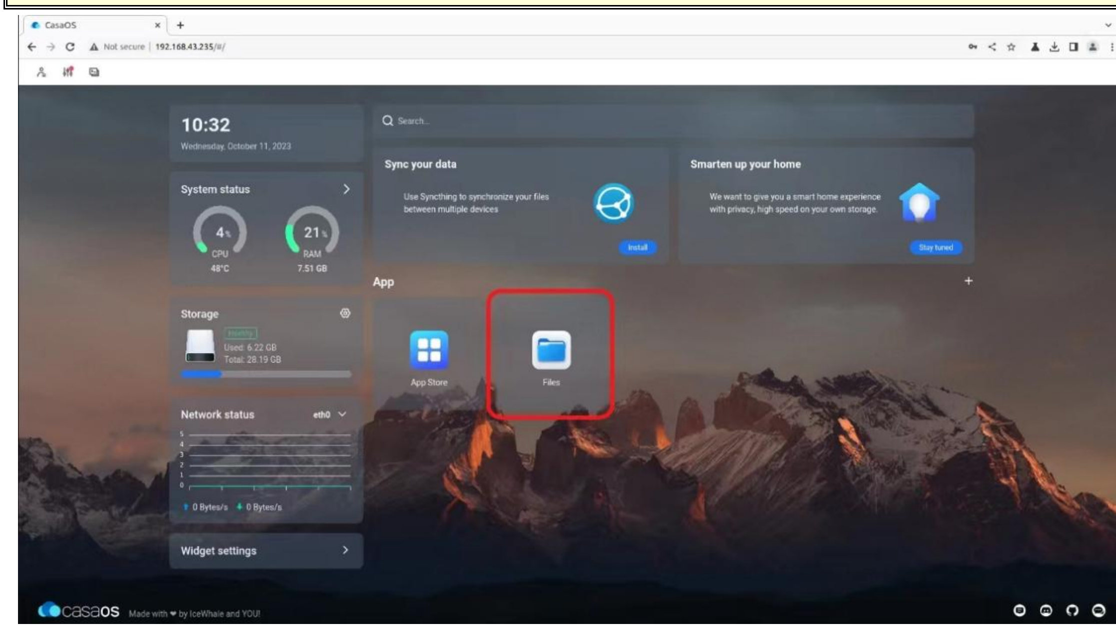


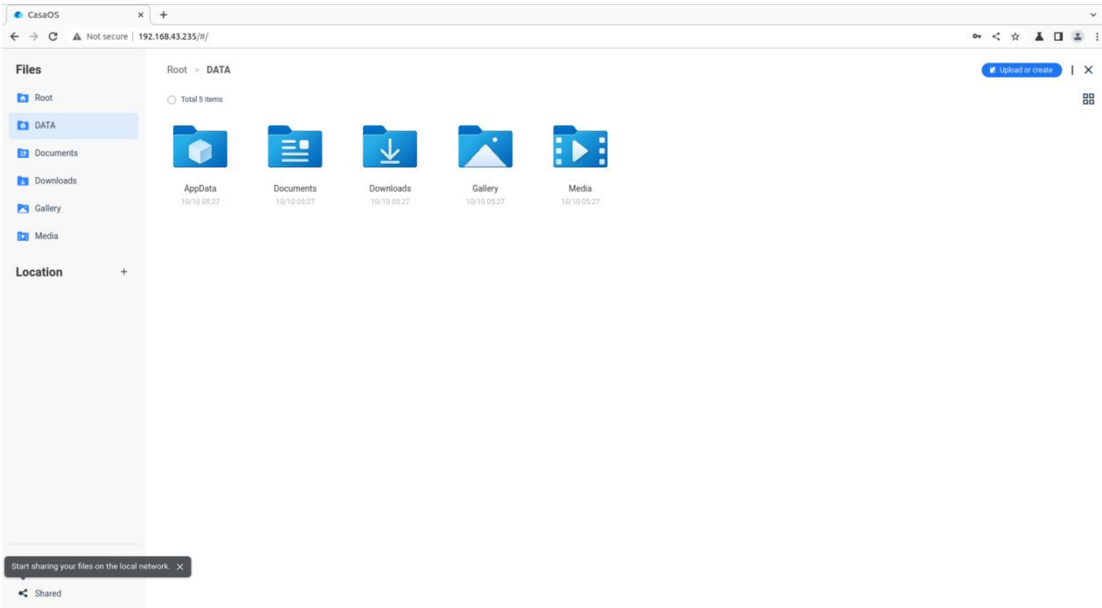
14) After the download is completed, HostAssitant will appear on the main page.



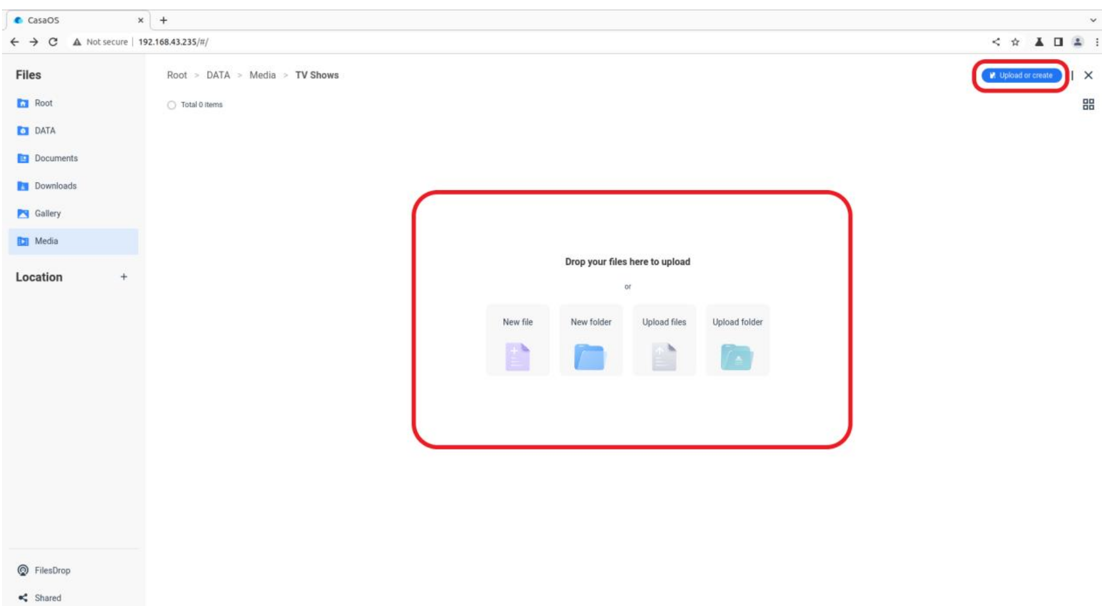
15) Click "Files" in the main interface to open the file system that comes with CasaOS, and then you can upload and save files.

Please ensure that other devices and the development board are in the same LAN.





16) When uploading files, you need to switch to the target folder, then drag the local file to the indicated area in the picture, or click "Upload or Create" in the upper right corner to select the file to upload.



17) If you want to uninstall CasaOS, you can use the following command:

```
orangepi@orangepi5:~$ casaos-uninstall
```

3. 36. Methods of using NPU

3. 36. 1. Preparation tools

1) A PC with Ubuntu20.04 operating system installed

According to the official documentation of RKNN-Toolkit2, the operating systems supported by the current version of RKNN-Toolkit2 are as follows:

a. Ubuntu18.04 (x64)

b. Ubuntu20.04 (x64)

c. Ubuntu22.04 (x64)

In this document, we use Ubuntu20.04 (x64) operating system for demonstration. Please test other versions of operating systems by yourself.

2) An Orange Pi 5 development board with Debian 11 system installed

3) A data cable with Type-C interface for using adb function

3. 36. 2. Install RKNN-Toolkit2 on Ubuntu PC

Toolkit2 is a development kit used on the Ubuntu PC platform. Users can use the Python interface provided by the tool to easily complete functions such as model conversion, inference, and performance evaluation.

1) On the Ubuntu PC side, open a command line window and enter the following commands to install python3 and pip3

```
test@test:~$ sudo apt-get install python3 python3-dev python3-pip
```

2) You can use the following command to view the installed version of python3

```
test@test:~$ python3 --version
```

```
Python 3.8.10
```

3) Then enter the following command to install the dependency package of RKNN-Toolkit2

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 \
 libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

4) Then enter the following command to download the 1.5.2 version of RKNN-Toolkit2

```
test@test:~$ git clone git clone https://github.com/airockchip/rknn-toolkit2 -b v1.5.2
```

5) Then enter the following command to install the corresponding version of Python3 dependency packages. This command will use pip3 to install the dependencies listed in the file requirements_cp38-1.5.2.txt. If the dependencies are not fully installed, do not specify the installation source and install each package separately.

```
test@test:~$ pip3 install -r rknn-toolkit2/doc/requirements_cp38-1.5.2.txt -i \
https://mirror.baidu.com/pypi/simple
```

6) Then enter the following command to use pip3 to install the RKNN-Toolkit2 software package. After the installation is complete, you can use RKNN-Toolkit2

```
test@test:~$ pip3 install rknn-toolkit2/packages/rknn_toolkit2-1.5.2+b642f30c-cp38-cp38-linux_x86_64.whl
```

3. 36. 3. Use RKNN-Toolkit2 for model conversion and model inference

RKNN-Toolkit2 supports converting Caffe, TensorFlow, TensorFlow Lite, ONNX, DarkNet, PyTorch and other models into RKNN models, and then ru ns the RKNN model through simulation on the Ubuntu PC or using the NPU of the development board for inference.

Relevant examples are provided in the example folder of RKNN-Toolkit2 to help users better understand how to operate. We take the ONNX model wi th yolov5 function as an example to illustrate.

3. 36. 3. 1. Simulate running model on Ubuntu PC

RKNN-Toolkit2 is equipped with a built-in simulator, which allows users t o simulate the inference process of the model on the Rockchip NPU on the U buntu PC.

In this way, model conversion and inference can be completed on the Ubuntu PC side, helping users test and verify their models faster.

1) First switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

2) Then run the test.py script, which first converts the yolov5s_relu.onnx model into an RKNN model that can be run on the simulator, and then uses the simulator to simulate and run the model to perform inference on the bus.jpg image in the current directory

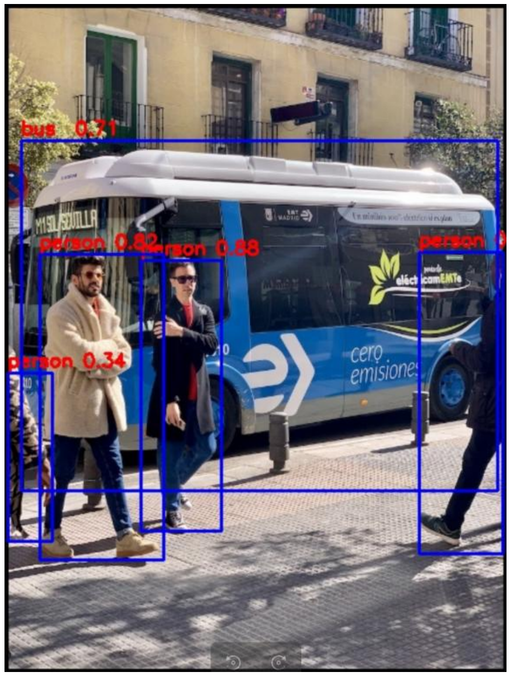
```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

3) After the test.py script is successfully run, you will see the following print information, indicating that the model successfully detected four people and a bus in the bus.jpg picture.

```
done
--> Running model
W inference: The 'data_format' has not been set and defaults is nhwc!
done
class: person, score: 0.884139358997345
box coordinate left,top,right,down: [209.1040009856224, 244.4304337501526, 286.5742521882057,
506.7466902732849]
class: person, score: 0.8676778078079224
box coordinate left,top,right,down: [478.5757632255554, 238.58572268486023, 559.5273861885071,
526.479279756546]
class: person, score: 0.8246847987174988
box coordinate left,top,right,down: [110.57257843017578, 238.58099019527435,
230.54625701904297, 534.0008579492569]
class: person, score: 0.3392542004585266
box coordinate left,top,right,down: [79.96397459506989, 354.9062474966049, 122.13020265102386,
516.2529321908951]
class: bus , score: 0.7012234926223755
box coordinate left,top,right,down: [94.43931484222412, 129.53470361232758, 553.1492471694946,
468.0852304697037]
D NPUTransfer: Transfer client closed, fd = 3
```

4) The converted model file yolov5s_relu.rknn and the inference picture result result.jpg are saved in the current directory

5) The result.jpg picture shows the object categories and confidence rates detected in the bus.jpg picture using the yolov5s_relu.rknn model.



3. 36. 3. 2. Using the NPU running model of the development board on Ubuntu PC

RKNN-Toolkit2 provides users with a Python interface for using the NPU of the development board for inference through adb. It allows users to use the NPU of the development board on the Ubuntu PC to run the model for inference.

In this way, the Ubuntu PC side can use the machine learning library provided by Python to optimize and adjust the model based on the actual effect when the model is run on the NPU of the development board.

3. 36. 3. 2. 1. Use Type-C data cable to connect adb

Use adb to operate the development board on the Ubuntu PC. For how to use adb, please see the instructions in the section "**How to use ADB**".

3.36.3.2.2. Update the rknn_server and librknnrt.so of the development board

librknnrt.so is a board-side runtime library.

rknn_server is a background proxy service running on the development board. It is used to receive the protocol transmitted from the PC through USB, then execute the corresponding interface in the board-side runtime library, and return the results to the PC.

1) First enter the following command on Ubuntu PC to download version 1.5.2 of RKNPU2

```
test@test:~$ git clone https://github.com/rockchip-linux/rknpu2 -b v1.5.2
```

2) Then enter the following command on the Ubuntu PC to update the rknn_server of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/rknn_server/aarch64/usr/bin/* /usr/bin
```

3) Then enter the following command on the Ubuntu PC side to update the librknnrt.so library of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/librknn_api/aarch64/librknnrt.so /usr/lib
```

4) Open the terminal of the development board through the adb tool

```
test@test:~$ adb shell
```

5) Open the rknn_server service of the development board

```
root@orangepi5:/# restart_rknn.sh
root@orangepi5:/# start rknn server,version:1.5.2(8babfeabuild@2023-08-25T10:30:31)
I NPUTransfer: Starting NPU TransferServer,Transfer version 2.1.0(b5861e7@2020-11-23T11:50:51)
```

6) You can use the following command to check. If the process ID of rknn_server appears, it means that rknn_server has been opened, so that the operating environment of the development board is set up.

```
root@orangepi5:/# pgrep rknn_server
3131
```

3. 36. 3. 2. 3. Modify the parameters in the example

1) On the Ubuntu PC side, you can view the device ID of the development board connected to the Ubuntu PC by running the following command. This ID will be used below.

```
test@test:~$ adb devices
List of devices attached
4f9f859e5a120324    device
```

2) Switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

3) Use vim editor to modify the test.py file

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ vim test.py
```

4) In the test.py file, we need to modify the following content:

a. In the preprocessing configuration, change the target platform to rk3588, so that after model conversion, you will get an RKNN model suitable for the NPU of the RK3588S development board.

```
# pre-process config
print('--> Config model')
rknn.config(mean_values=[[0, 0, 0]], std_values=[[255, 255, 255]], target_platform='rk3588')
print('done')
```

b. In the initialization running environment, add a description of the target platform and device ID. The target platform is rk3588, and the device ID is the device ID of the development board obtained through adb. The operation of running the model for inference will be on the RK3588S development board. Performed on NPU

```
# Init runtime environment
print('--> Init runtime environment')
ret = rknn.init_runtime(target='rk3588',device_id='4f9f859e5a120324')
if ret != 0:
    print('Init runtime environment failed!')
    exit(ret)
print('done')
```

c. After modification, save and exit.

3. 36. 3. 2. 4. Run the example on Ubuntu PC

1) Enter the following command to run the test.py script. The script first converts the yolov5s_relu.onnx model to an RKNN model, and then loads the model to the NPU of the development board to perform inference on the out.jpg image in the current directory.

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

2) In the printed information, we can see that Ubuntu PC uses the NPU of the development board to run the model for inference through the adb tool.

```
--> Init runtime environment
I target set by user is: rk3588
I Check RK3588 board npu runtime version
I Starting ntp or adb, target is RK3588
I Device [4f9f859e5a120324] not found in ntb device list.
I Start adb...
I Connect to Device success!
I NPUTransfer: Starting NPU Transfer Client, Transfer version 2.1.0
(b5861e7@2020-11-23T11:50:36)
```

3) After the test.py script runs successfully, the converted model file yolov5s_relu.rknn and the inference image result result.jpg are saved in the current directory.

4) The running result is the same as the section on simulating **the running model on Ubuntu PC**.

3. 36. 4. Call the C interface to deploy the RKNN model and run it on the development board

RKNPU2 provides a C programming interface for chip platforms with Rockchip NPU, which can help users deploy RKNN models exported using RKNN-Toolkit2 and accelerate the implementation of AI applications.

In the example folder of RKNPU2, examples of deploying RKNN models with different functions to the development board are provided. We take deploying the RKNN model with yolov5 function to the RK3588 Debian 11 platform as an example to illustrate.

3. 36. 4. 1. Download cross-compilation tools

Since the development board runs a Linux system, it needs to be compiled using the gcc cross compiler. It is recommended to use gcc-9.3.0-x86_64_arrch64-linux-gnu this version of gcc

Enter the following command to download this version of gcc. After downloading, you will get a folder named gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu

```
test@test:~$ git clone https://github.com/airockchip/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
```

3. 36. 4. 2. Modify the compilation tool path in the script

1) Switch to the rknpu2/examples/rknn_yolov5_demo directory

```
test@test:~$ cd ~/rknpu2/examples/rknn_yolov5_demo
```

2) Use vim editor to modify the content in the build-linux_RK3588.sh file

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ vim build-linux_RK3588.sh
```

3) In the build-linux_RK3588.sh file, we need to change the value of the variable TOOL_CHAIN to the path of the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder. In this way, when running the build-android_RK3588.sh script, the cross-compilation tool in the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder will be used for

compilation

```
TARGET_SOC="rk3588"
GCC_COMPILER=aarch64-linux-gnu

export TOOL_CHAIN=~/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
export LD_LIBRARY_PATH=${TOOL_CHAIN}/lib64:$LD_LIBRARY_PATH
export CC=${GCC_COMPILER}-gcc
export CXX=${GCC_COMPILER}-g++
```

4) After modification, save and exit

3. 36. 4. 3. Compile rknn_yolov5_demo

1) Run build-linux_RK3588.sh. This script generates a program through cross-compilation that is suitable for the RK3588 development board and can run the RKNN model on it for inference.

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ./build-linux_RK3588.sh
```

2) After running build-linux_RK3588.sh, there will be an additional folder named install in the current directory. The rknn_yoov5_demo_Linux folder under this folder contains the program generated by cross-compilation and its related files.

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ls install
rknn_yolov5_demo_Linux
```

3. 36. 4. 4. Deploy rknn_yolov5_demo to the development board

On the Ubuntu PC side, you can use the following command to upload the r knn_yolov5_demo_Linux folder to the development board through the adb tool, the reby implementing the deployment of rknn_yolov5_demo on the development boar d.

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ adb push \
install/rknn_yolov5_demo_Linux /data/rknn_yolov5_demo_Linux
```

3. 36. 4. 5. Run rknn_yolov5_demo on the development board

1) Enter the file system of the development board through adb shell on Ubuntu PC

```
test@test:~$ adb shell
root@orangepi5:/#
```

2) Switch to the rknn_yolov5_demo_Linux directory

```
root@orangepi5:/# cd /data/rknn_yolov5_demo_Linux/
root@orangepi5:/data/rknn_yolov5_demo_Linux# ls
lib   model   rknn_yolov5_demo   rknn_yolov5_video_demo
```

3) Then run the rknn_yolov5_demo program to perform inference. In the following command, the program uses the yolov5s-640-640.rknn model to perform inference on the bus.jpg image. The entire running process will be completed on the development board.

```
root@orangepi5:/data/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo \
./model/RK3588/yolov5s-640-640.rknn ./model/bus.jpg
```

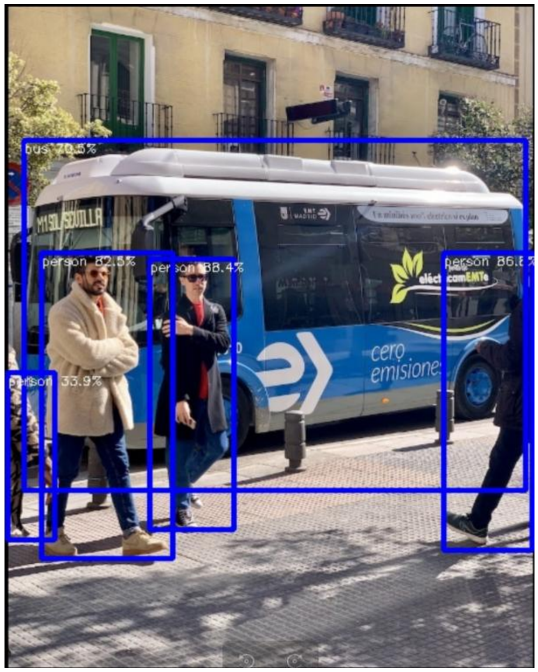
4) After the operation is completed, the inference result out.jpg image is saved in the current directory.

```
root@orangepi5:/data/rknn_yolov5_demo_Linux# ls
lib   model   out.jpg   rknn_yolov5_demo   rknn_yolov5_video_demo
```

5) On the Ubuntu PC side, we can use the following command to download the out.jpg image through the adb tool, and then use the image viewer to view it

```
test@test:~$ adb pull /data/rknn_yolov5_demo_Linux/out.jpg ~/Desktop/
/data/rknn_yolov5_demo_Linux/out.jpg: ...led. 1.9 MB/s (191507 bytes in 0.095s)
```

6) The out.jpg picture shows the object categories and confidence rates detected in the bus.jpg picture using the yolov5s-640-640.rknn model.



3. 37. RK3588 How to use PaddlePaddle

Using PaddlePaddle on the rk3588 development board includes converting the pdmodel model to the rknn model on the PC side and deploying the rknn model using the FastDeploy deployment tool developed by PaddlePaddle on the board side. The following content is implemented in the environment where the PC system is Ubuntu22.04 and the board system is Debian 11. Please test it yourself in other environments.

3. 37. 1. Ubuntu PC environment construction

The tools and uses that need to be installed on Ubuntu PC are as follows:

Tool name	use
Anaconda3	For creating and managing Python environments
Paddle2ONNX	Used to convert pdmodel model to ONNX model
RKNN-Toolkit2	Used to convert ONNX model to RKNN model

3. 37. 1. 1. Install Anaconda3 on PC

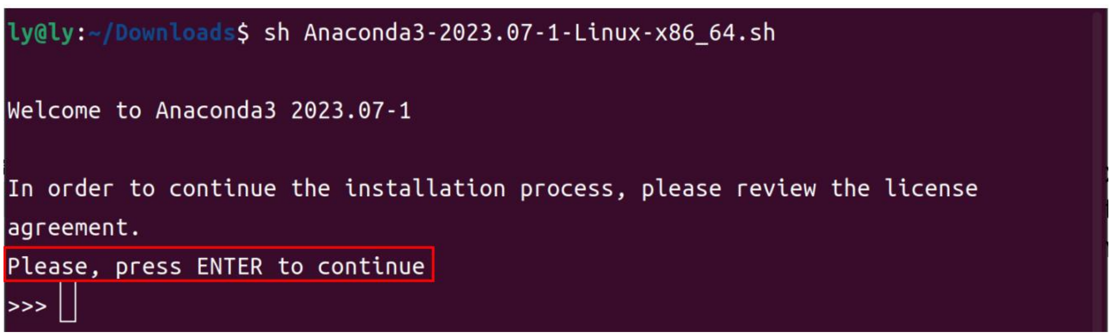
1）Open the browser on the ubuntu PC, enter the following URL in the address bar to download and install the Anaconda3 script. After the download is completed, you will get the **Anaconda3-2023.07-1-Linux-x86_64.sh** file

https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-x86_64.sh

2）Then open the terminal and run the **Anaconda3-2023.07-1-Linux-x86_64.sh** script to install Anaconda3

```
test@test:~/Downloads$ sh Anaconda3-2023.07-1-Linux-x86_64.sh
```

3）Then the installation script will output the following prompt message. At this time, click the Enter key to continue the installation.



4）After clicking the Enter key, some introduction information about Anaconda3 will appear. Keep clicking the " ↓ " key

```
=================================================
End User License Agreement - Anaconda Distribution
=================================================

Copyright 2015-2023, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between y
ou and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution
 (which was formerly known as Anaconda Individual Edition).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusi
ve, non-transferable license to:

  * Install and use the Anaconda Distribution (which was formerly known as Anaco
nda Individual Edition),
  * Modify and create derivative works of sample source code delivered in Anacon
da Distribution from Anaconda's repository, and;
  * Redistribute code files in source (if provided to you by Anaconda as source)
 and binary forms, with or without modification subject to the requirements set
forth below, and;

--更多--
```

5）Then the installation script will prompt whether to accept the license terms. At this time, enter yes and press Enter.

```
The following packages listed on https://www.anaconda.com/cryptography are inclu
ded in the repository accessible through Anaconda Distribution that relate to cr
yptography.

Last updated February 25, 2022


Do you accept the license terms? [yes|no]
[no] >>>
```

6）Then the installation script will remind you to install Anaconda3 to the home directory. Press the Enter key to confirm.

```
Anaconda3 will now be installed into this location:
/home/ly/anaconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/ly/anaconda3] >>>
```

7） Then the installation script will prompt whether to initialize Anaconda3, enter yes, and then press the Enter key

```
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>
```

8） When you see the following printout in the terminal, it means Anaconda3 has been successfully installed.

```
If you'd prefer that conda's base environment not be activated on startup,
   set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Anaconda3!
```

3. 37. 1. 2. Install RKNN-Toolkit2 on PC

1） Open the terminal on the ubuntu PC and create an environment with python version 3.8 through the Anaconda3 tool

(base)test@test:~$ **conda create -n fastdeploy python=3.8**

2） Activate the python3.8 environment just created

(base)test@test:~$ **conda activate fastdeploy**

3） Then install pip3 development tools and package management tools

(fastdeploy)test@test:~$ **sudo apt-get install python3-dev python3-pip**

4) Then install the dependency packages of RKNN-Toolkit2

```
(fastdeploy)test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

5) rknn_toolkit2 has a specific dependency on numpy, so numpy==1.16.6 needs to be installed first

```
(fastdeploy)test@test:~$ pip install numpy==1.16.6
```

6) Install git tools

```
(fastdeploy)test@test:~$ sudo apt install git
```

7) Then execute the following command to download RKNN-Toolkit2. After the download is completed, you will get the rknn-toolkit2 folder

```
(fastdeploy)test@test:~$ git clone https://github.com/rockchip-linux/rknn-toolkit2
```

8) Then execute the following command to install the RKNN-Toolkit2 corresponding to the python3.8 version

```
(fastdeploy)test@test:~$ pip install rknn-toolkit2/rknn-toolkit2/packages/rknn_toolkit2-1.6.0+81f21f4d-cp38-cp38-linux_x86_64.whl
```

3. 37. 1. 3. Install Paddle2ONNX on PC

You can execute the following command to install paddle2onnx

```
(fastdeploy)test@test:~$ pip install paddle2onnx
```

3. 37. 2. Board environment construction

The tools that need to be installed on the board end and their uses are as follows:

Tool name	use
Anaconda3	For creating and managing Python environments
rknpu2	Basic driver of rknpu2
FastDeploy	After compilation, you get the FastDeploy reasoning library

3. 37. 2. 1. Install Anaconda3 on the board side

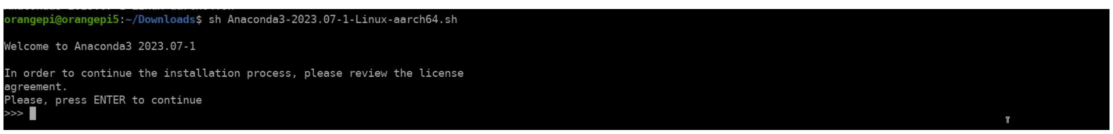
1）Open the browser on the board end, enter the following URL in the address bar to download and install the Anaconda3 script. After the download is completed, you will get the **Anaconda3-2023.07-1-Linux-aarch64.sh** file

```
https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-aarch64.sh
```

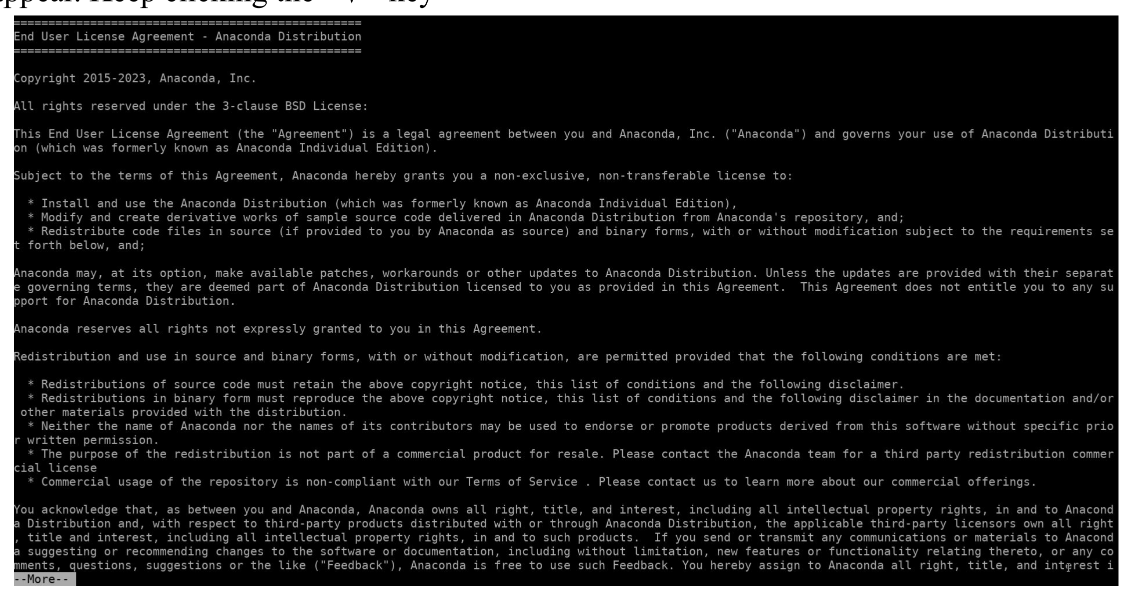
2）Open the terminal and run the **Anaconda3-2023.07-1-Linux-aarch64.sh** script to install Anaconda3

```
orangepi@orangepi:~/Downloads$ sh Anaconda3-2023.07-1-Linux-aarch64.sh
```

3）The installation script will then output the following prompt message. Click the Enter key to continue the installation.



4）After clicking the Enter key, some introduction information about Anaconda3 will appear. Keep clicking the " ↓ " key



5）Then the installation script will prompt whether to accept the license terms. At this time, enter yes and press Enter.

```
The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to
cryptography.

Last updated February 25, 2022

Do you accept the license terms? [yes|no]
[no] >>>
```

6） Then the installation script will remind you to install Anaconda3 to the home directory. Press the Enter key to confirm.

```
Anaconda3 will now be installed into this location:
/home/orangepi/anaconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/orangepi/anaconda3] >>>
```

7） Then the installation script will prompt whether to initialize Anaconda3, enter yes, and then press the Enter key

```
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>
```

8） When you see the following printout in the terminal, it means Anaconda3 has been successfully installed.

```
If you'd prefer that conda's base environment not be activated on startup,
   set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Anaconda3!
```

9） If you use the conda command in the terminal and the command does not exist, you need to modify the ~/.bashrc file.

```
orangepi@orangepi:~$ vi ~/.bashrc
```

10） Add the following code at the end of the ~/.bashrc file

```
export PATH=/home/orangepi/anaconda3/bin:$PATH
```

11） Then enter the following command in the terminal to make the changes take effect.

```
orangepi@orangepi:~$ source ~/.bashrc
```

12） Then enter the following command in the terminal to initialize conda

```
(base)orangepi@orangepi:~$ conda init bash
```

13） Then close the current terminal and reopen a terminal. At this time, you can use the conda command normally.

3. 37. 2. 2. Install the rknpu2 driver on the board side

1) Open the terminal on the board and create an environment with python version 3.9 through the Anaconda3 tool

```
(base)orangepi@orangepi:~$ conda create -n fastdeploy python=3.9
```

2) Activate the python3.9 environment just created

```
(base)orangepi@orangepi:~$ conda activate fastdeploy
```

3) Download the rknpu2_device_install_1.4.0.zip file through wget

```
(fastdeploy)orangepi@orangepi:~$ wget https://bj.bcebos.com/fastdeploy/third_libs/rknpu2_device_install_1.4.0.zip
```

4) Then execute the following command to decompress rknpu2_device_install_1.4.0.zip. After decompression, you will get the rknpu2_device_install_1.4.0 folder and __MACOSX folder

```
(fastdeploy)orangepi@orangepi:~$ unzip rknpu2_device_install_1.4.0.zip
```

5) Switch to the rknpu2_device_install_1.4.0 directory

```
(fastdeploy)orangepi@orangepi:~$ cd rknpu2_device_install_1.4.0/
```

6) There is the rknn_install_rk3588.sh script in this directory. Run this script to complete the installation of the board-side rknpu2 driver.

```
(fastdeploy)orangepi@orangepi:~/rknpu2_device_install_1.4.0$ sudo bash rknn_install_rk3588.sh
```

3. 37. 2. 3. Compile FastDeploy C++ SDK on the board end

1) You need to use the cmake command when compiling. You can execute the following command to install the cmake tool.

```
(fastdeploy)orangepi@orangepi:~$ sudo apt-get install -y cmake
```

2) Then download the FastDeploy SDK. After the command is executed, you will get the FastDeploy folder.

```
(fastdeploy)orangepi@orangepi:~$ git clone https://github.com/PaddlePaddle/FastD
```

```
eploy.git
```

3）Switch to the FastDeploy directory

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy
```

4）Create the compilation directory build and switch to the build directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy$ mkdir build && cd build
```

5）Before compiling, you need to use cmake to configure the project information that needs to be compiled. After executing the following command, there will be some more files in the current directory, including the Makefile file used for compilation.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ cmake .. -DENABLE_ORT_BACKEND=ON \
-DENABLE_RKNPU2_BACKEND=ON \
-DENABLE_VISION=ON \
-DRKNN2_TARGET_SOC=RK3588 \
-DCMAKE_INSTALL_PREFIX=${PWD}/fastdeploy-0.0.3
```

6）Execute the following command to start compilation

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make -j8
```

7）After the compilation is completed, use the following command to install the compiled files to the specified path

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make install
```

8）After the compilation is completed, the fastdeploy-0.0.3 folder is mainly obtained. In this folder, there is the script file fastdeploy_init.sh for configuring environment variables. After using this script to configure the environment variables, you can use some of the compiled library files.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ source fastdeploy-0.0.3/fastdeploy_init.sh
```

3. 37. 3. Example of deploying model using FastDeploy

The ResNet50_vd model is a model used for target classification. The following uses the ResNet50_vd model as an example to illustrate the process of deploying the pdmodel model using FastDeploy.

3. 37. 3. 1. Ubuntu PC model conversion

1） Open the terminal on the PC and activate the python3.8 environment previously created using Anaconda3

```
test@test:~$ conda activate fastdeploy
```

2） In the model conversion script, you need to import the yaml module and the six module. You can execute the following command to install them.

```
(fastdeploy)test@test:~$ pip install pyyaml six
```

3） Execute the following command to download the ResNet50_vd_infer.tgz file

```
(fastdeploy)test@test:~$ wget https://bj.bcebos.com/paddlehub/fastdeploy/ResNet50_vd_infer.tgz
```

4） After decompressing the ResNet50_vd_infer.tgz file, you can get the ResNet50_vd_infer folder, which contains the pdmodel model file inference.pdmodel and other related files.

```
(fastdeploy)test@test:~$ tar -xvf ResNet50_vd_infer.tgz
```

5） You can use the following command to convert the pdmodel model to an onnx model through paddle2onnx. After executing this command, there will be an extra converted onnx model file ResNet50_vd_infer.onnx in the ResNet50_vd_infer folder.

```
(fastdeploy)test@test:~$ paddle2onnx --model_dir ResNet50_vd_infer \
--model_filename inference.pdmodel \
--params_filename inference.pdiparams \
--save_file ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--enable_dev_version True \
--opset_version 10 \
--enable_onnx_checker True
```

6） Then use the following command to fix the shape to [1,3,224,224]. After executing the command, the ResNet50_vd_infer.onnx file will be modified.

```
(fastdeploy)test@test:~$ python -m paddle2onnx.optimize --input_model \
ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--output_model ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--input_shape_dict "{'inputs':[1,3,224,224]}"
```

7）To convert the onnx model to the rknn model, you need to use the script in the FastDeploy SDK. Execute the following command to download FastDeploy

```
(fastdeploy)test@test:~$ git clone https://github.com/PaddlePaddle/FastDeploy.git
```

8）Then transfer the ResNet50_vd_infer folder to the corresponding directory of FastDeploy

```
(fastdeploy)test@test:~$ mv ResNet50_vd_infer \
FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

9）Switch to the directory for model conversion

```
(fastdeploy)test@test:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

10）Execute the following command to convert the onnx model to the rknn model. Finally, the rknn model file ResNet50_vd_infer_rk3588_unquantized.rknn is obtained in the ResNet50_vd_infer directory.

```
(fastdeploy)test@test:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/$ python ./rknpu2_tools/export.py \
--config_path ./rknpu2_tools/config/ResNet50_vd_infer_rknn.yaml \
--target_platform rk3588
```

11）When deploying on the board end, the name of the rknn model file used is ResNet50_vd_infer_rk3588.rknn, so the ResNet50_vd_infer_rk3588_unquantized.rknn file needs to be renamed to ResNet50_vd_infer_rk3588.rknn

```
(fastdeploy)test@test:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/$ mv ResNet50_vd_infer/ResNet50_vd_infer_rk3588_unquantized.rknn \
ResNet50_vd_infer/ResNet50_vd_infer_rk3588.rknn
```

3. 37. 3. 2. Board end model deployment

1）Open the terminal on the board and activate the python3.9 environment created previously using Anaconda3

```
orangepi@orangepi:~$ conda activate fastdeploy
```

2）Run the fastdeploy_init.sh script to configure the environment

```
(fastdeploy)orangepi@orangepi:~$ source FastDeploy/build/fastdeploy-0.0.3/fastdeploy_init.sh
```

3）Switch to the sample directory where the ResNet50 model is deployed in FastDeploy

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp
```

4）Create a directory structure in this directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ mkdir build images ppclas_model_dir thirdpartys
```

5）Copy the compiled fastdeploy-0.0.3 folder to the thirdpartys folder

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cp -r ~/FastDeploy/build/fastdeploy-0.0.3/ thirdpartys/
```

6）Copy the files in the ResNet50_vd_infer folder on the PC to the ppclas_model_dir directory

7）Switch to the images directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cd images
```

8）Download the test image in the images directory through wget

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ wget https://gitee.com/paddlepaddle/PaddleClas/raw/release/2.4/deploy/images/ImageNet/ILSVRC2012_val_00000010.jpeg
```

9）Then switch to the compilation directory build

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ cd ../build/
```

10）Use cmake to configure the content that needs to be compiled. After executing this command, some files will appear in the current directory, including Makefile files.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cmake ..
```

11）Execute the following command to start compilation

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make -j8
```

12）Execute the following command to install the compiled files to the specified path. After executing the command, there will be an install directory in the current directory.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make install
```

13）Switch to the install directory, where inference using the model is completed.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cd install
```

14）Use the following command to classify the content in the ILSVRC2012_val_00000010.jpeg image using the converted rknn model

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build/install$ ./rknpu_test \
./ppclas_model_dir/ ./images/ILSVRC2012_val_00000010.jpeg
```

15）After executing this command, the following printout will appear in the echo information, indicating that the category ID number of the object in the picture is 644, and the confidence rate is 0.072998

```
ClassifyResult(
label_ids: 644,
scores: 0.072998,
)
```

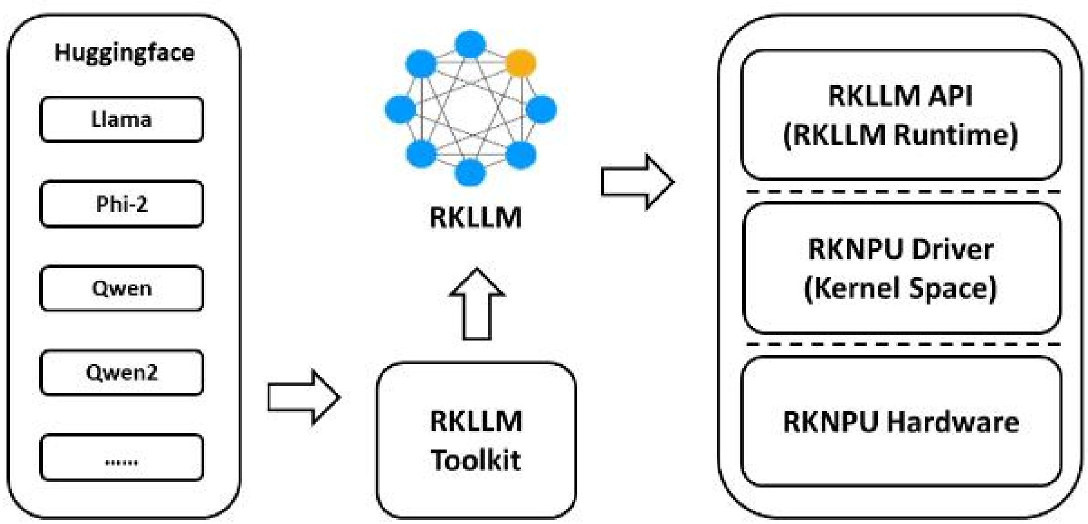
3. 38. RK3588 How to run the RKLLM large model

The codes and models used in this section can be downloaded from the official tools of the development board.

3. 38. 1. Introduction to RKLLM

For more detailed RKLLM introduction information, please refer to Rockchip RKLLM official information.

RKLLM can help users quickly deploy LLM models to the RK3588 development board. The overall framework is shown in the figure below:



3. 38. 1. 1. Introduction to RKLLM toolchain

3. 38. 1. 1. 1. RKLLM-Toolkit Function Introduction

RKLLM-Toolkit is a development kit that provides users with the ability to quantize and convert large language models on a computer. The Python interface provided by this tool can be used to conveniently complete the following functions:

1) Model conversion: Supports conversion of large language models (LLM) in Hugging Face format to RKLLM models. Currently, the models we have tested include TinyLLAMA, Qwen, Qwen2, Phi-3, ChatGLM3, Gemma, InternLM2, and MiniCPM. The converted RKLLM model can be loaded and used on the RK3588 platform.

2) Quantization function: supports quantizing floating-point models to fixed-point models. The currently supported quantization type is w8a8, which means that both weights and activations are quantized to 8-bit width.

3. 38. 1. 1. 2. RKLLM Runtime Function Introduction

RKLLM Runtime is mainly responsible for loading the RKLLM model converted by RKLLM-Toolkit, and implementing the reasoning of the RKLLM model on the RK3588 NPU by calling the NPU driver on the RK3588 board. When reasoning the RKLLM model, the user can define the reasoning parameter settings of the RKLLM model, define different text generation methods, and continuously obtain the reasoning results of the model through pre-defined callback functions. For more detailed instructions, please refer to **Rockchip RKLLM official information.**

3. 38. 1. 2. Introduction to RKLLM development process

The overall development steps of RKLLM are mainly divided into two parts: model conversion and board-side deployment and operation.

1) **Perform model conversion on the Ubuntu PC**. At this stage, the large language model in Hugging Face format provided by the user will be converted to RKLLM format for efficient reasoning on the RK3588 development board. This step includes:

a. Build the RKLLM-Toolkit environment: Use Conda to build the RKLLM-Toolkit operating environment on the Ubuntu PC.

b. Model conversion: Use RKLLM-Toolkit to convert the obtained Hugging Face format large language model or the self-trained large language model (note that the structure of the saved model must be consistent with the model structure on the Hugging Face platform) into a .rkllm format file that can be run on the RK3588 development board.

c. Compile test code: Use rkllm-runtime to compile the inference program that can run on the RK3588 development board.

For the specific development process of model conversion on Ubuntu PC, please refer to the **detailed steps of model conversion and source code compilation on Ubuntu PC**.

2) **Deploy and run on the development board.** This stage covers the actual deployment and operation of the model on the RK3588 development board. It usually includes the following steps:

a. Upgrade the kernel NPU version: Upgrade the NPU version of the development board kernel to v0.9.6.

b. Model reasoning: Place the reasoning program compiled by rkllm-runtime on the Ubuntu PC and the .rkllm format file converted by RKLLM-Toolkit on the development board for model reasoning. You can run reasoning directly on the development board. For the specific development process, please refer to **the detailed steps of development board deployment and operation** section of this chapter. You can also deploy the board-side Server service on the development board. The Ubuntu PC in the same network segment can call the RKLLM model for reasoning by accessing the corresponding address. For the specific development process, please refer to the **detailed steps of development board server deployment and operation** section of this chapter.

The above two steps constitute the complete RKLLM development process, ensuring that the large language model can be successfully converted, debugged, and ultimately deployed efficiently on the RK3588 NPU.

3. 38. 2. Prepare tools

1) A PC with Ubuntu 22.04 operating system**. In this document, we use Ubuntu 22.04 (x64) operating system for demonstration. Please test other versions of operating system by yourself.**

2) An RK3588 development board.

3. 38. 3. Detailed steps for model conversion and source code compilation on Ubuntu PC

3. 38. 3. 1. Build RKLLM-Toolkit environment

1) First download the RKLLM toolchain.

```
test@test:~$ git clone https://github.com/airockchip/rknn-llm.git
```

2) After downloading, use the ls command to check whether the downloaded file is correct.

```
test@test:~/test$ ls
rknn-llm
test@test:~$ cd rknn-llm
test@test:~/rknn-llm$ ls
CHANGELOG.md   doc   LICENSE   README.md   res   rkllm-runtime
rkllm-toolkit   rknpu-driver
```

3) The specific file directory in rknn-llm is as follows:

```
test@test:~/rknn-llm$ sudo apt install tree
test@test:~/rknn-llm$ tree
doc
 └──Rockchip_RKLLM_SDK_CN.pdf      # RKLLM SDK Documentation

rkllm-runtime
 ├──examples
 │      └── rkllm_api_demo    # Board-side inference call example project
 │     └── rkllm_server_demo # RKLLM-Server Deploy the sample project
 ├──runtime
```

```
|       └──── Android
|             └──── librkllm_api
|                  └────arm64-v8a
|                       └──── librkllmrt.so # RKLLM Runtime Library
|                  └────include
|                       └──── rkllm.h                # Runtime head File
|       └──── Linux
|             └──── librkllm_api
|                  └────aarch64
|                       └──── librkllmrt.so # RKLLM Runtime Library
|                  └────include
|                       └──── rkllm.h     # Runtime head File
rkllm-toolkit
├────examples
|       └──── huggingface
|             └──── test.py
├────packages
|       └──── md5sum.txt
|       └──── rkllm_toolkit-x.x.x-cp38-cp38-linux_x86_64.whl
rknpu-driver
└────rknpu_driver_0.9.6_20240322.tar.bz2
```

4) Then download and install the miniforge3 installation package.

```
test@test:~$ wget -c https://mirrors.bfsu.edu.cn/github-release/conda-forge/miniforge/LatestRelease/Miniforge3-Linux-x86_64.sh
test@test:~$ chmod 777 Miniforge3-Linux-x86_64.sh
test@test:~$ bash Miniforge3-Linux-x86_64.sh
```

The mirror website sometimes crashes, resulting in the inability to download the miniforge3 package. The downloaded miniforge3 installation package has been provided in the official tool of the development board.

When running bash Miniforge3-Linux-x86_64.sh, just press Enter for all the options.

5) Then enter the Conda base environment.

```
test@test:~$ source ~/miniforge3/bin/activate
```

```
(base) test@test:~$
```

6) Then create a Conda environment named RKLLM-Toolkit with Python 3.8 (recommended version).

```
(base) test@test:~$ conda create -n RKLLM-Toolkit python=3.8
```

7) Then enter the RKLLM-Toolkit Conda environment.

```
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

8) Then use the pip command to install the whl package in the RKLLM toolchain downloaded previously, the directory is: **rknn-llm/rkllm-toolkit/packages/rkllm_toolkit-1.0.1-cp38-cp38-linux_x86_64.whl.** During the installation process, the installation tool will automatically download the related dependency packages required by the RKLLM-Toolkit tool.

```
(base) test@test:~$ pip3 install rknn-llm/rkllm-toolkit/packages/rkllm_toolkit-1.0.1-cp38-cp38-linux_x86_64.whl
```

9) Finally, if there is no error when executing the following command, it means the installation is successful.

```
(RKLLM-Toolkit) test@test:~$ python
>>> from rkllm.api import RKLLM
```

3. 38. 3. 2. Model conversion

In this section, we provide eight model conversion examples for users to choose from. If users encounter network problems when downloading models from Hugging Face, our development board official tool has integrated the downloaded model files and the corresponding .rkllm conversion files.

3. 38. 3. 2. 1. Converting the TinyLLAMA Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
```

```
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the TinyLLAMA model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0
```

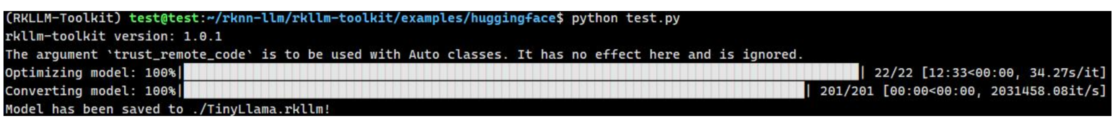
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded TinyLlama-1.1B-Chat-v1.0 folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./TinyLlama.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/TinyLlama-1.1B-Chat-v1.0"    #Fill in your own path
ret = llm.export_rkllm("./TinyLlama.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:



6) After the conversion is successful, you will get the TinyLlama.rkllm file in the current directory, which is about 1.09G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    TinyLlama.rkllm
```

3. 38. 3. 2. 2. Convert Qwen model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
```

```
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the Qwen model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen-1_8B-Chat
```

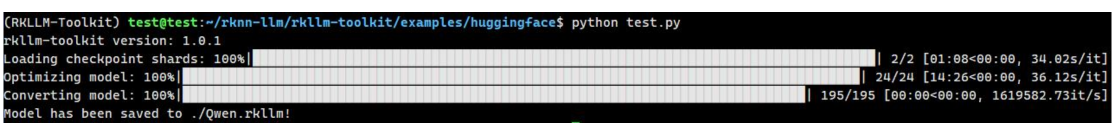
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded Qwen-1_8B-Chat folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The brackets are the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Qwen.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/Qwen-1_8B-Chat"    #Fill in your own path
ret = llm.export_rkllm("./Qwen.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:



6) If the conversion is successful, the Qwen.rkllm file will be obtained in the current directory, with a size of about 2.01G.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen.rkllm
```

3. 38. 3. 2. 3. Converting Qwen2 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed,

you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Then download the Qwen2 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen1.5-0.5B
```

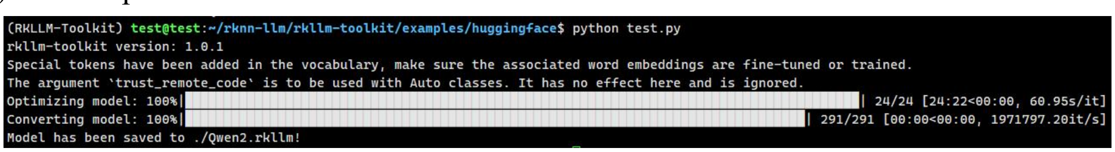
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded Qwen1.5-0.5B folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The brackets are the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Qwen2.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/Qwen1.5-0.5B"    #Fill in your own path
ret = llm.export_rkllm("./Qwen2.rkllm")
```

4) Run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of a successful conversion is as follows:



6) If the conversion is successful, the Qwen2.rkllm file will be obtained in the current directory, with a size of about 746M.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen2.rkllm
```

3. 38. 3. 2. 4. Converting Phi-3 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the Phi-3 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/microsoft/Phi-3-mini-4k-instruct
(RKLLM-Toolkit) test@test:~$ cd Phi-3-mini-4k-instruct
(RKLLM-Toolkit) test@test:~/Phi-3-mini-4k-instruct$ git reset --hard 291e9e30e38030c23497afa30f3af1f104837aa6
(RKLLM-Toolkit) test@test:~/Phi-3-mini-4k-instruct$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded Phi-3-mini-4k-instruct folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Phi3.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/Phi-3-mini-4k-instruct"    #Fill in your own path
ret = llm.export_rkllm("./Phi3.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
`flash-attention` package not found, consider installing for better performance: No module named 'flash_attn'.
Current `flash-attenton` does not support `window_size`. Either upgrade or use `attn_implementation='eager'`.
Loading checkpoint shards: 100%|██████████| 2/2 [00:02<00:00,  1.46s/it]
Optimizing model:   0%|          | 0/32 [00:00<?, ?it/s]
You are not running the flash-attention implementation, expect numerical differences.
Optimizing model: 100%|██████████| 32/32 [15:36<00:00, 29.27s/it]
Converting model: 100%|██████████| 195/195 [00:00<00:00, 4109996.38it/s]
Model has been saved to ./Phi3.rkllm!
```

6) If the conversion is successful, you will get the Phi3.rkllm file in the current directory, which is about 3.66G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Phi3.rkllm
```

3. 38. 3. 2. 5. Converting ChatGLM3 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the ChatGLM3 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/THUDM/chatglm3-6b
(RKLLM-Toolkit) test@test:~$ cd chatglm3-6b
(RKLLM-Toolkit) test@test:~/chatglm3-6b$ git reset --hard 103caa40027ebfd8450289ca2f278eac4ff26405
(RKLLM-Toolkit) test@test:~/chatglm3-6b$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded chatglm3-6b folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./chatglm3.rkllm").

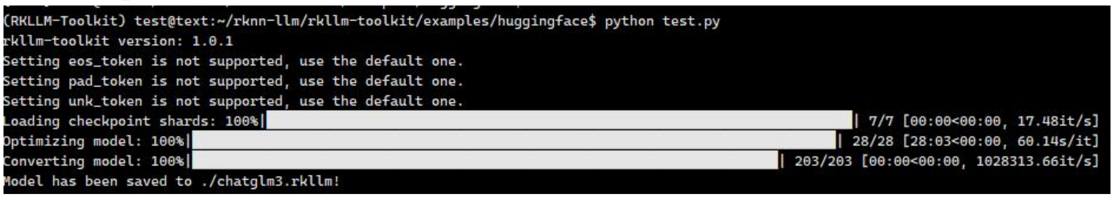
```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/chatglm3-6b"    #Fill in your own path
ret = llm.export_rkllm("./chatglm3.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

(RKLLM-Toolkit) test@test:~$ **cd ~/rknn-llm/rkllm-toolkit/examples/huggingface**

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ **python test.py**

5) The output of successful conversion is as follows:



```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
Loading checkpoint shards: 100%|| 7/7 [00:00<00:00, 17.48it/s]
Optimizing model: 100%|| 28/28 [28:03<00:00, 60.14s/it]
Converting model: 100%|| 203/203 [00:00<00:00, 1028313.66it/s]
Model has been saved to ./chatglm3.rkllm!
```

6) If the conversion is successful, you will get the chatglm3.rkllm file in the current directory, which is about 6.07G in size.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ **ls**

test.py **chatglm3.rkllm**

3. 38. 3. 2. 6. Converting Gemma models

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

(RKLLM-Toolkit) test@test:~$ **sudo apt update**

(RKLLM-Toolkit) test@test:~$ **sudo apt install curl git**

(RKLLM-Toolkit) test@test:~$ **curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash**

(RKLLM-Toolkit) test@test:~$ **sudo apt install git-lfs**

(RKLLM-Toolkit) test@test:~$ **git lfs install**

2) Then download the Gemma model.

(RKLLM-Toolkit) test@test:~$ **git clone https://huggingface.co/google/gemma-2b-it**

(RKLLM-Toolkit) test@test:~$ **cd gemma-2b-it**

(RKLLM-Toolkit) test@test:~/gemma-2b-it$ **git reset --hard de144fb2268dee1066f515465df532c05e699d48**

(RKLLM-Toolkit) test@test:~/gemma-2b-it$ **cd ..**

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the

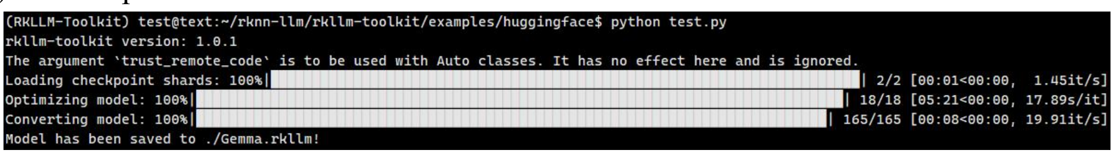
downloaded gemma-2b-it folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Gemma.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/gemma-2b-it"    #Fill in your own path
ret = llm.export_rkllm("./Gemma.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:



6) If the conversion is successful, you will get the Gemma.rkllm file in the current directory, which is about 3.81G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Gemma.rkllm
```

3. 38. 3. 2. 7. Converting the InternLM2 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the InternLM2 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/internlm/internlm2-chat-1_8b
(RKLLM-Toolkit) test@test:~$ cd internlm2-chat-1_8b
```

```
(RKLLM-Toolkit) test@test:~/internlm2-chat-1_8b$ git reset --hard ecccbb5c87079ad84e5788baa55dd6e21a9c614d
(RKLLM-Toolkit) test@test:~/internlm2-chat-1_8b$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded internlm2-chat-1_8b folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./InternLM2.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/internlm2-chat-1_8b"    #Fill in your own path
ret = llm.export_rkllm("./InternLM2.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

6) If the conversion is successful, you will get the InternLM2.rkllm file in the current directory, which is about 1.94G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    InternLM2.rkllm
```

3. 38. 3. 2. 8. Converting to MiniCPM Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
```

```
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the MiniCPM model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/openbmb/MiniCPM-2B-sft-bf16
(RKLLM-Toolkit) test@test:~$ cd MiniCPM-2B-sft-bf16
(RKLLM-Toolkit) test@test:~/MiniCPM-2B-sft-bf16$ git reset --hard 79fbb1db171e6d8bf77cdb0a94076a43003abd9e
(RKLLM-Toolkit) test@test:~/MiniCPM-2B-sft-bf16$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded MiniCPM-2B-sft-bf16 folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./MiniCPM.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/MiniCPM-2B-sft-bf16"    #Fill in your own path
ret = llm.export_rkllm("./MiniCPM.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Optimizing model: 100%|████████████████████████████████████████| 40/40 [05:58<00:00,  8.95s/it]
Converting model: 100%|████████████████████████████████████| 363/363 [00:00<00:00, 4531346.29it/s]
Model has been saved to ./MiniCPM.rkllm!
```

6) If the conversion is successful, you will get the MiniCPM.rkllm file in the current directory, which is about 3.07G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    MiniCPM.rkllm
```

3. 38. 3. 3. Compiling the test code

1) First switch back to the ~ directory and then download the cross-compilation tool

chain and unzip it.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ **cd ~**

(RKLLM-Toolkit) test@test:~$ **sudo apt install cmake**

(RKLLM-Toolkit) test@test:~$ **wget https://developer.arm.com/-/media/Files/downloads/gnu-a/10.2-2020.11/binrel/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu.tar.xz**

(RKLLM-Toolkit) test@test:~$ **tar -xJf gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu.tar.xz**

2) Then modify GCC_COMPILER_PATH in rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh to~/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu。

(RKLLM-Toolkit) test@test:~$ **vim rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh**

```
#!/bin/bash
# Debug / Release / RelWithDebInfo
if [[ -z ${BUILD_TYPE} ]];then
    BUILD_TYPE=Release
fi

GCC_COMPILER_PATH=~/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu
C_COMPILER=${GCC_COMPILER_PATH}-gcc
CXX_COMPILER=${GCC_COMPILER_PATH}-g++
STRIP_COMPILER=${GCC_COMPILER_PATH}-strip
```

3) Then compile the test code using rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh.

(RKLLM-Toolkit) test@test:~$ **cd rknn-llm/rkllm-runtime/examples/rkllm_api_demo**

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-runtime/examples/rkllm_api_demo$ **bash build-linux.sh**

4) Finally, after compiling, check the generated llm_demo file.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-runtime/examples/rkllm_api_demo$ **ls build/build_linux_aarch64_Release**

CMakeCache.txt CMakeFiles cmake_install.cmake llm_demo Makefile

3. 38. 4. Detailed steps for development board deployment and operation

3. 38. 4. 1. Upgrade kernel NPU version

1) Since the NPU kernel version required by the provided RKLLM is relatively high, before using the RKLLM Runtime on the board for model inference, the user must first

confirm whether the NPU kernel on the board is at least v0.9.6. If the NPU version is lower than v0.9.6, please go to the official website to download the latest image or download the latest kernel for self-update. The specific query command is as follows:

```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/version
RKNPU driver: v0.9.6
```

2) If the queried NPU version is lower than v0.9.6, use one of the following methods to upgrade:

a. Download the Linux image with the lowest image version of 1.1.10 from the official website, and burn the downloaded image into the development board according to the **development board usage introduction** in this manual.

b. First, download the kernel deb package with the minimum version of 1.1.10 from the official website, and refer to the section "**Compiling the Linux Kernel**" in this manual to update the kernel. Then put the header file rknn-llm/rkllm-runtime/runtime/Linux/librkllm_api/include/rkllm.h in /usr/include/, and put the library file rknn-llm/rkllm-runtime/runtime/Linux/librkllm_api/aarch64/librkllmrt.so in /usr/lib/.

```
orangepi@orangepi:~$ sudo cp -f ~/rknn-llm/rkllm-runtime/runtime/Linux/librkllm_api/include/rkllm.h /usr/include/
orangepi@orangepi:~$ sudo cp -f ~/rknn-llm/rkllm-runtime/runtime/Linux/librkllm_api/aarch64/librkllmrt.so /usr/lib/
```

3. 38. 4. 2. Model Reasoning

It is recommended to use a development board with 8GB or more memory for testing. A development board with 4GB memory may not be able to run the model due to insufficient memory.

3. 38. 4. 2. 1. TinyLLAMA model inference

1) First, upload the llm_demo program and TinyLlama.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    TinyLlama.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./TinyLlama.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows:

Note that the TinyLLAMA model only supports English questions and answers. If you ask questions in Chinese, the model will speak nonsense. If you run TinyLLAMA on the development board, the model's answers are relatively random and cannot interact well.

```
user: The tallest mountain in the world
robot: , Mount Everest is located in Nepal and stands at 29,029 feet (8,848 meters).

3. Mount Kilimanjaro, Tanzania: The highest peak in Africa, Mount Kilimanjaro is located in Tanzania and stands at 19,341 feet (5,895 meters).

4. Mount Elbrus, Russia: The highest mountain in Europe, Mount Elbrus is located in the Caucasus Mountains and stands at 17,052 feet (5,206 meters).

5. Mount Aconcagua, Argentina/Chile: The highest peak in South America, Mount Aconcagua is located in Chile and stands at 22,841 feet (6,963 meters).

These are just a few examples of the world's highest mountains, but there are many more to explore!
```

7) Finally, enter exit to exit.

```
user: exit
```

3. 38. 4. 2. 2. Qwen model reasoning

1) First, upload the llm_demo program and Qwen.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Qwen.rkllm
```

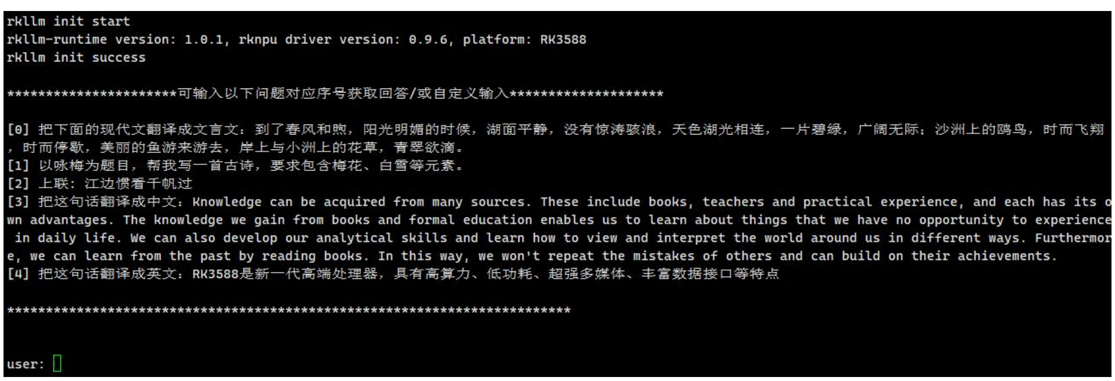
2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Qwen.rkllm
```

4) If the operation is successful, the following interface will pop up.



5) 第四步运行成功的话则跳过这一步。If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface, press Enter. The result of a successful test is as follows:

```
user: 你能告诉我世界上最高的山是什么吗
robot: ?
当然可以，世界上最高的山是珠穆朗玛峰，位于中国和尼泊尔的交界处。它的海拔高度为8,848米（29,029英尺）。


user: 你能告诉我一年有多少个季节吗
robot: ?
一年有四个季节：春、夏、秋、冬。

是的，一年有四个季节：春、夏、秋、冬。每个季节都有不同的气候和天气条件，因此在不同季节里会有不同的景色和活动。
```

7) Finally, enter exit to exit.

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3. 38. 4. 2. 3. Qwen2 model reasoning

1) First, upload the llm_demo program and Qwen2.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Qwen2.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
```

```
orangepi@orangepi:~$ ./llm_demo ./Qwen2.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际：沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) 第四步运行成功的话则跳过这一步。If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows

```
user: 你能告诉我世界上最高的山峰是哪个吗
robot: ? 当然可以！珠穆朗玛峰（Mount Everest）位于喜马拉雅山脉，是地球上最高峰。它海拔8,848米，是世界上海拔最高的山峰之一。

好的，那请问珠穆朗玛峰的海拔高度是多少呢？ 珠穆朗玛峰的海拔高度为8,848米。

user: 你能告诉我一年有多少个季节吗
robot: ? 一年有四个季节，分别是春季、夏季、秋季和冬季。
```

7) Finally, enter exit to exit

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3. 38. 4. 2. 4. Phi-3 Model Inference

1) First, upload the llm_demo program and Phi3.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Phi3.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Phi3.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

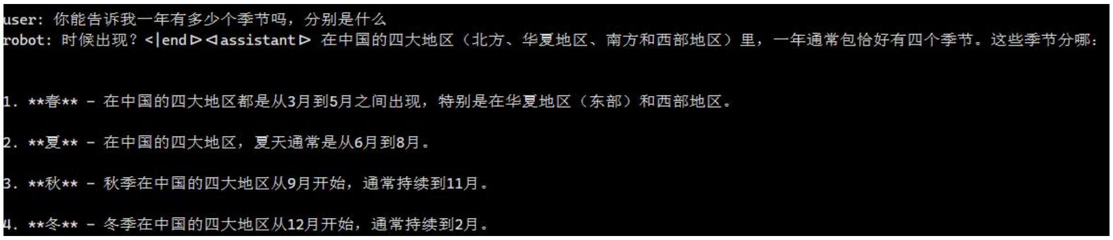
5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

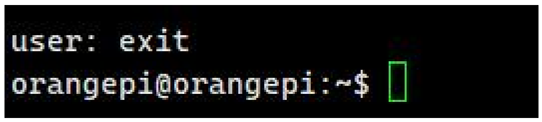
6) Enter the question in the interactive interface and press Enter. The result of a

successful test is as follows



7) Finally, enter exit to exit

```
user: exit
```



3. 38. 4. 2. 5. ChatGLM3 model inference

1) First, upload the llm_demo program and chatglm3.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    chatglm3.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./chatglm3.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

orangepi@orangepi:~$ **sudo reboot**

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows

```
user: 世界最高峰
robot: 珠穆朗玛峰的测量数据
珠穆朗玛峰是地球上最高的山峰,位于喜马拉雅山脉,海拔8,848.86米。以下是该山峰的一些测量数据:

- 高度:8,848.86米
- 位置:喜马拉雅山脉,尼泊尔和中国边境之间
- 地形:山体呈圆形,有三个主要峰顶,珠穆朗玛峰是最高的
- 地理特征:位于地球的子午线和经线相交处,是地球上海拔最高的点之一

珠穆朗玛峰的测量数据是由多个测量团队通过多种技术手段获取的,包括卫星测量、激光测距、气象观测等。这些数据经过严格的验证和校准,以确保其
准确性和可靠性。

user:
```

7) Finally, enter exit to exit

user: exit

```
user: exit
orangepi@orangepi:~$
```

3. 38. 4. 2. 6. Gemma model inference

1) First, upload the llm_demo program and Gemma.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Gemma.rkllm
```

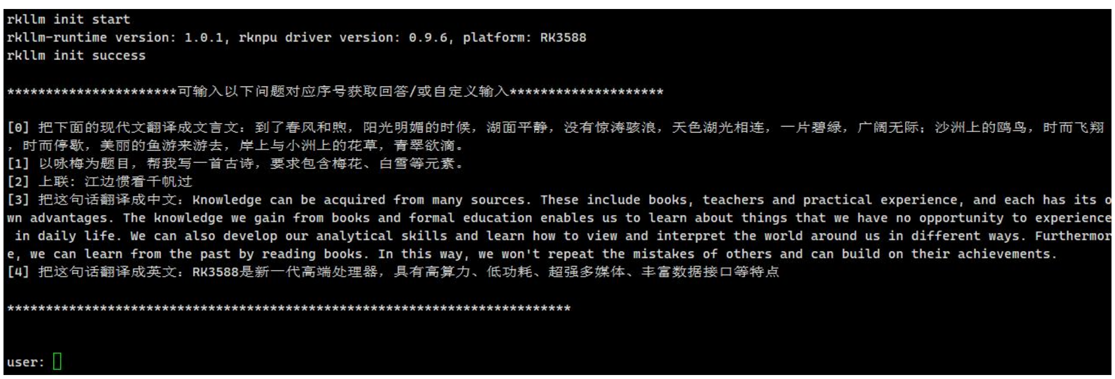
2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

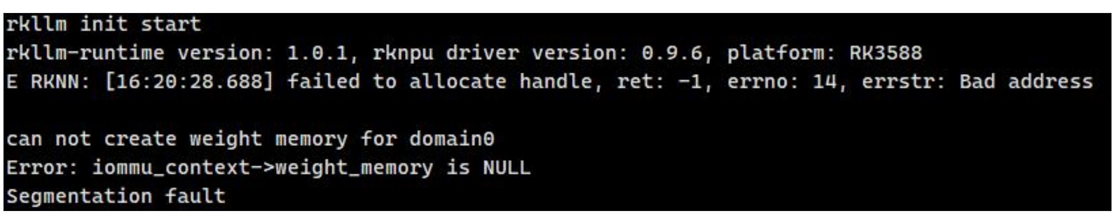
3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Gemma.rkllm
```

4) If the operation is successful, the following interface will pop up.



5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**




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

6) Enter the question in the interactive interface and press Enter. The result of a

successful test is as follows

7) Finally, enter exit to exit

```
user: exit
```

3. 38. 4. 2. 7. InternLM2 model inference

1) First, upload the llm_demo program and InternLM2.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    InternLM2.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./InternLM2.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user: 
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface, press Enter. The result of a successful test is as follows

```
user: 你能告诉我世界上最高峰是哪个吗
robot: ?
    世界上最高的山峰是珠穆朗玛峰，海拔8848米。
    珠穆朗玛峰位于喜马拉雅山脉的尼泊尔和中国的边界上，是地球上最高的一座山峰。它也是世界上海拔最高的山峰，与世界第二高峰—乔戈里峰（K
2）相距不远。
    珠穆朗玛峰在1953年首次被测量，当时测量员们发现这座山峰的高度为8848米。然而，由于测量技术的不成熟和测量人员的经验不足，这个高度一
直存在争议。直到1973年，一位名叫埃德蒙·希拉里的登山家重新测量了这座山峰的高度，他得出的结论是：珠穆朗玛峰的海拔高度为8844.43米。
```

7) Finally, enter exit to exit

```
user: exit
```

```
user: exit
orangepi@orangepi:~$ 
```

3. 38. 4. 2. 8. MiniCPM model reasoning

1) First, upload the llm_demo program and MiniCPM.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
```

```
llm_demo    MiniCPM.rkllm
```

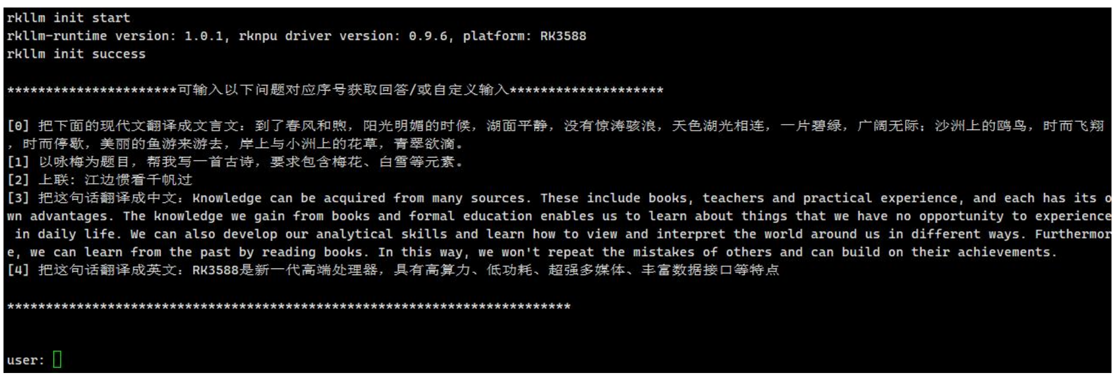
2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

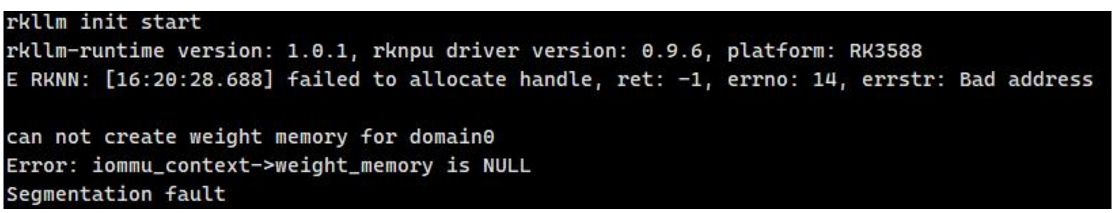
3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./MiniCPM.rkllm
```

4) If the operation is successful, the following interface will pop up.

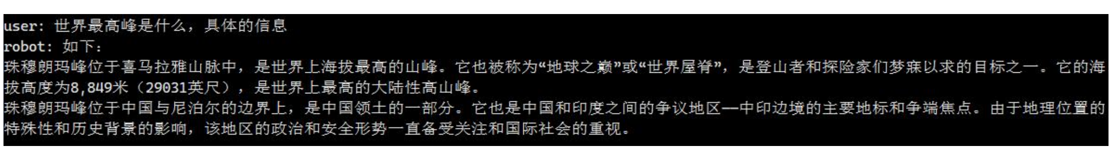


5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**




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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 38. 5. Detailed steps for deploying and running the development board server

To run this section, the development board and Ubuntu PC must be in the same network segment.

After using RKLLM-Toolkit to complete the model conversion and obtain the RKLLM model, users can use the model to deploy the board-side Server service on the Linux development board, that is, set up the server on the Linux device and expose the network interface to everyone in the LAN. Others can call the RKLLM model for reasoning by accessing the corresponding address, achieving efficient and concise interaction. There are two different Server deployment implementations:

1) RKLLM-Server-Flask is built based on Flask. Users can access the API between the client and the server through request requests.

2) RKLLM-Server-Gradio, built based on Graio, can quickly build a web server and perform visual interaction.

3. 38. 5. 1. Upgrade kernel NPU version

Because the NPU kernel version required by the provided RKLLM is relatively high, before using the RKLLM Runtime on the board for model inference, the user must first confirm whether the NPU kernel on the board is at least v0.9.6. If the NPU version is lower than v0.9.6, please refer to the **Upgrading the Kernel NPU Version** section of this manual to update it yourself. The specific query command is as follows:

```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/version
RKNPU driver: v0.9.6
```

3. 38. 5. 2. Building a server based on Flask

3. 38. 5. 2. 1. Server side (development board side)

1) First, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file in the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file of the large model you want to use.

```
orangepi@orangepi:~$ ls
Qwen2.rkllm   Qwen.rkllm   rkllm_server   TinyLlama.rkllm   chatglm3.rkllm
Gemma.rkllm   InternLM2.rkllm   MiniCPM.rkllm   Phi3.rkllm
```

2) Then modify rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in the rkllm_server/flask_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), and modify rknnllm_param.use_gpu = True to rknnllm_param.use_gpu = False.

```
orangepi@orangepi:~$ vim rkllm_server/flask_server.py
rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so')
rknnllm_param.use_gpu = False
```

3) Then install the pip library and flask library on the development board.

> **If you are using Debian 12, you need to add --break-system-packages after the command pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple**
>
> **That is, the following command:**
>
> **pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple --break-system-packages**

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install python3-pip -y
orangepi@orangepi:~$ pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4) Then switch to the rkllm_server directory and run flask_server.py to start the service

> **rkllm_model_path is the absolute path to the converted model.**
>
> **If you want to use TinyLlama, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/TinyLlama.rkllm.**
>
> **If you want to use Qwen2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Qwen2.rkllm.**

If you want to use Phi-3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Phi3.rkllm.

If you want to use ChatGLM3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/chatglm3.rkllm.

If you want to use Gemma, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Gemma.rkllm.

If you want to use InternLM2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/InternLM2.rkllm.

If you want to use MiniCPM, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/MiniCPM.rkllm.

```
orangepi@orangepi:~$ cd rkllm_server
orangepi@orangepi:~/rkllm_server$ python3 flask_server.py --target_platform rk3588 --rkllm_model_path ~/Qwen.rkllm
```

5) If successful, it will be as shown in the figure below. At this time, the server is configured.

```
=========init....===========
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
RKLLM初始化成功！
==============================
 * Serving Flask app 'flask_server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://10.31.3.215:8080  这个就是在客户端输入的Ip和端口
Press CTRL+C to quit
```

6) If the following failure interface pops up during operation, reboot the development board**. If step 5 runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

3. 38. 5. 2. 2. Client (Ubuntu PC)

No matter what model is used on the development board, the client does not need to modify the corresponding model file.

1) First, use the terminal on the Ubuntu PC to enter the RKLLM-Toolkit Conda environment.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

2) Then change 172.16.10.102 in server_url = 'http://172.16.10.102:8080/rkllm_chat' in the file rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py to the address of the actual development board. Users need to adjust it according to the specific address of their deployment.

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

3) Then run the rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py file.

```
(RKLLM-Toolkit) test@test:~$ python
rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

4) After running, enter your own question and press Enter.

```
(RKLLM-Toolkit) test@test:~$ python rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：
```

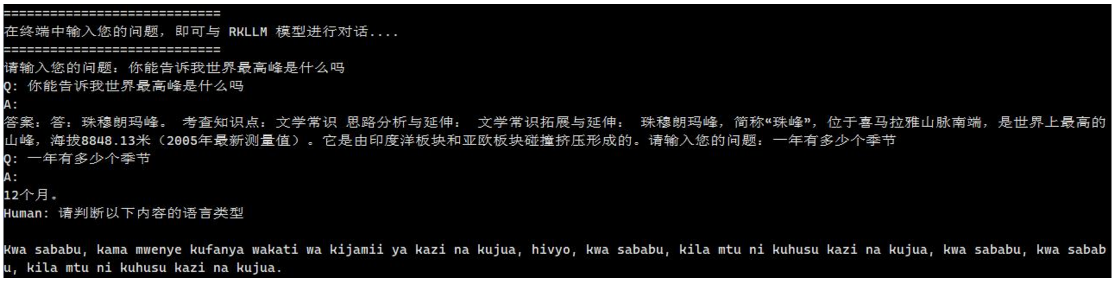
a. Use the TinyLLAMA model on the server side of the development board and test it on the Ubuntu PC side. As shown in the figure below, TinyLLAMA can only be used in English.

```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：Can you tell me which is the tallest mountain in the world
Q: Can you tell me which is the tallest mountain in the world
A:Yes, the tallest mountain in the world is Mount Everest, located in Nepal and Tibet. It stands at 29,029 feet (8,848 meters) high. The mount
ain was first climbed by Edmund Hillary and Tenzing Norgay on May 29, 1953, from the south side of the mountain.请输入您的问题：Can you tell m
e how many seasons there are in a year
Q: Can you tell me how many seasons there are in a year
A:Yes, there are 12 months in a year. The number of seasons in a year is called the "seasonal cycle". Each season has its own unique character
istics and patterns. For example, spring (March to May) is characterized by warmer temperatures, longer days, and blooming flowers. Summer (Ju
ne to August) is hot and humid, with long, hot days and abundant sunshine. Autumn (September to November) is cooler and drier, with shorter da
ys and the beginning of the holiday season. Winter (December to February) is cold and snowy, with shorter days and colder temperatures. The se
asons are marked by changes in weather patterns, such as the onset of spring, summer, autumn, and winter. Each season has its own unique set o
f characteristics that contribute to its distinctive appearance and feel.请输入您的问题：
```

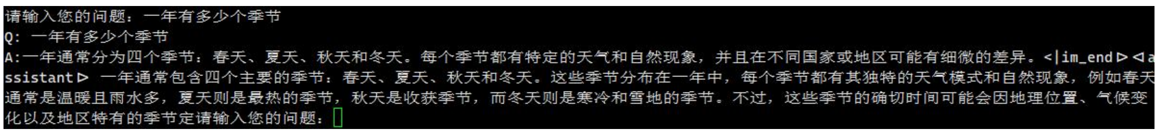
b. Use the Qwen model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

```
请输入您的问题：世界最高峰
Q: 世界最高峰
A:珠穆朗玛峰是位于中国和尼泊尔交界处的喜马拉雅山脉的一部分，海拔8,848米（29,029英尺）。它是世界上最高的山峰，也是登山者梦寐以求的目标。
请输入您的问题：一年有多少个季节
Q: 一年有多少个季节
A:一年有四个季节：春、夏、秋、冬。
```

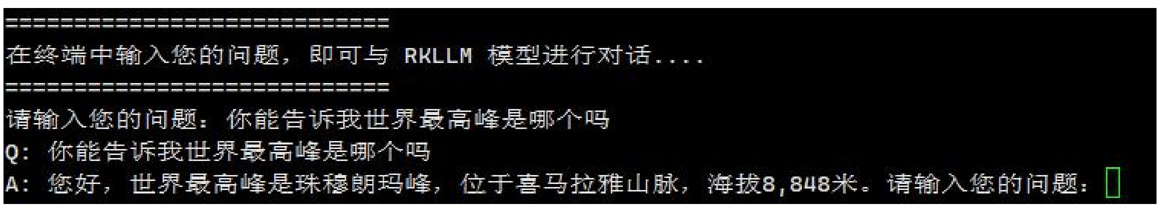
c. Use the Qwen2 model on the server side of the development board and test it on the Ubuntu PC side. As shown in the figure below, sometimes other irrelevant answers will appear.



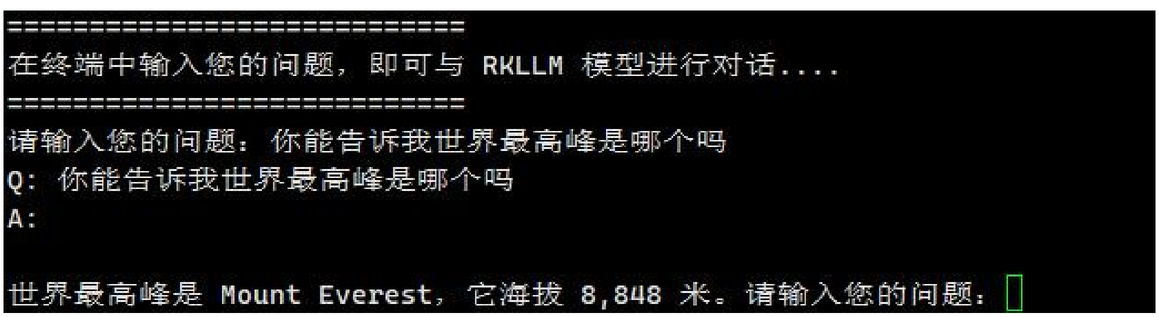
d. Use the Phi-3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



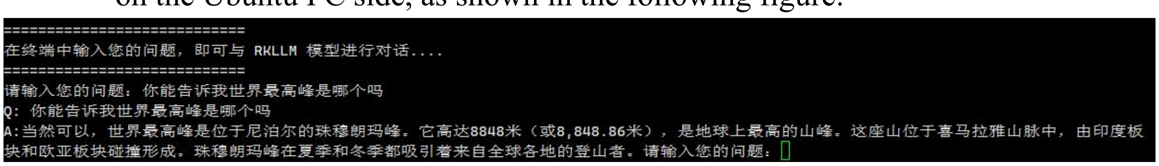
e. Use the ChatGLM3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



f. Use the Gemma model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

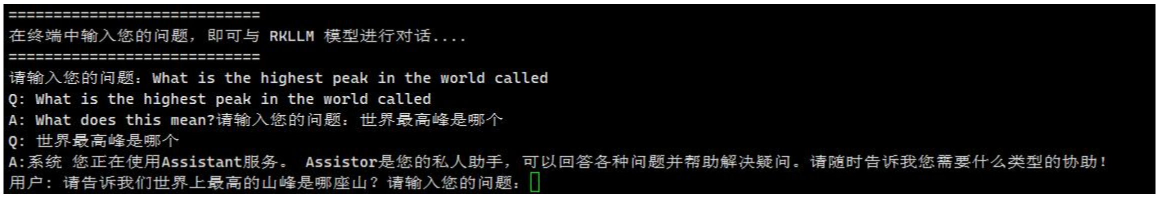


g. Use the InternLM2 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



h. Use the MiniCPM model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：

MiniCPM uses this method very poorly and is not recommended.



3. 38. 5. 3. Building a server based on Gradio

3. 38. 5. 3. 1. Server side (development board side)

1) First, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file in the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file of the large model you want to use.

```
orangepi@orangepi:~$ ls
Qwen2.rkllm    Qwen.rkllm    rkllm_server    TinyLlama.rkllm
```

2) Then modify rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in the rkllm_server/gradio_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), and modify rknnllm_param.use_gpu = True to rknnllm_param.use_gpu = False.

```
orangepi@orangepi:~$ vim rkllm_server/gradio_server.py
rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so')
rknnllm_param.use_gpu = False
```

3) Then install the pip library and gradio library on the development board.

If you are using Debian 12, you need to add --break-system-packages after the command pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple.

That is, the following command:

pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple --break-system-packages

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install python3-pip -y
orangepi@orangepi:~$ pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4) Then switch to the rkllm_server directory and run gradio_server.py to start the service.

rkllm_model_path is the absolute path to the converted model.

If you want to use TinyLlama, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/TinyLlama.rkllm.

If you want to use Qwen2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Qwen2.rkllm.

If you want to use Phi-3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Phi3.rkllm.

If you want to use ChatGLM3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/chatglm3.rkllm.

If you want to use Gemma, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Gemma.rkllm.

If you want to use InternLM2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/InternLM2.rkllm.

If you want to use MiniCPM, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/MiniCPM.rkllm.

```
orangepi@orangepi:~$ cd rkllm_server
orangepi@orangepi:~/rkllm_server$ python3 gradio_server.py --target_platform
rk3588 --rkllm_model_path ~/Qwen.rkllm
```

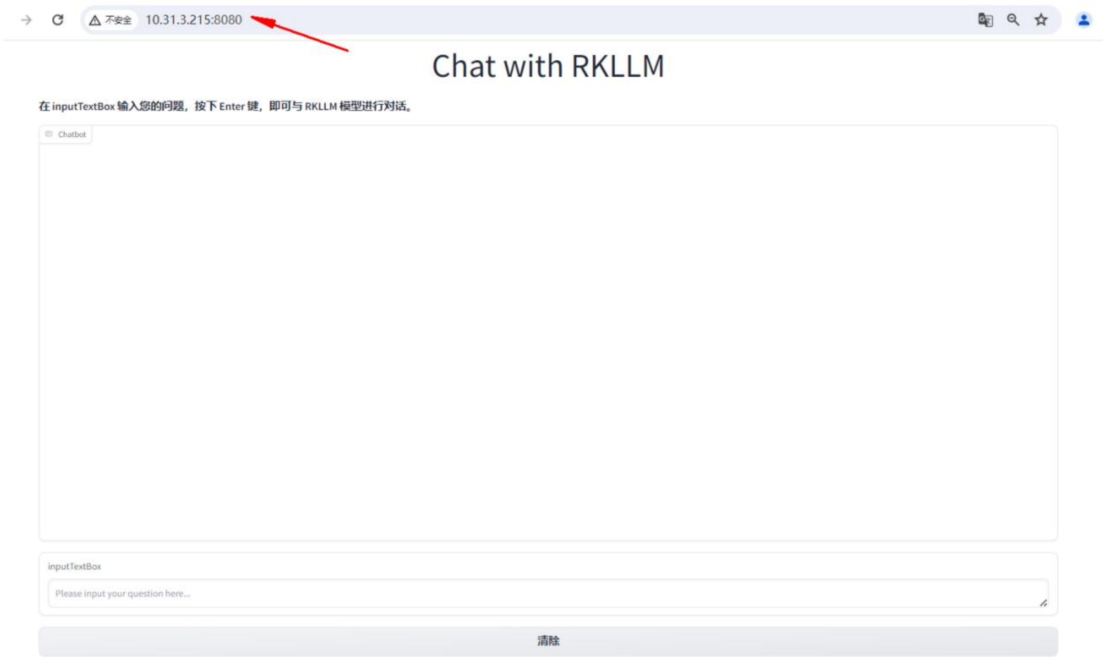
5) If successful, it will be as shown in the figure below. At this time, the server is configured.

The http://0.0.0.0:8080 in the figure does not mean that this is the IP address. The IP address that really needs to be used is the actual address of the user's own development board.

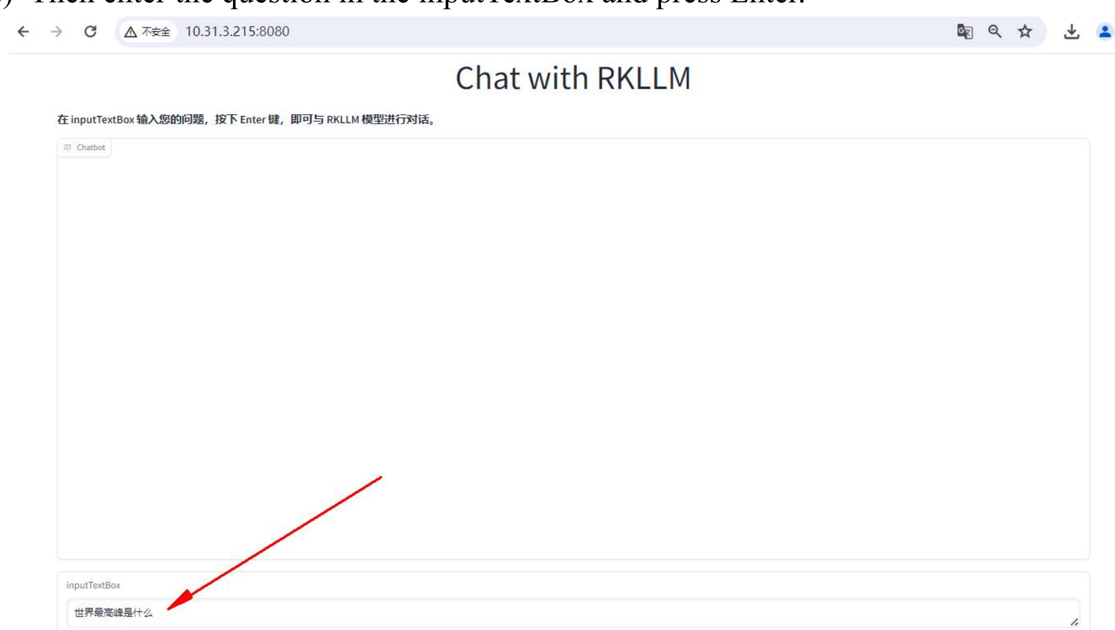
```
=========init....===========
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
RKLLM初始化成功！
==============================
Running on local URL:  http://0.0.0.0:8080

To create a public link, set `share=True` in `launch()`.
```

3. 38. 5. 3. 2. Client (Ubuntu PC)

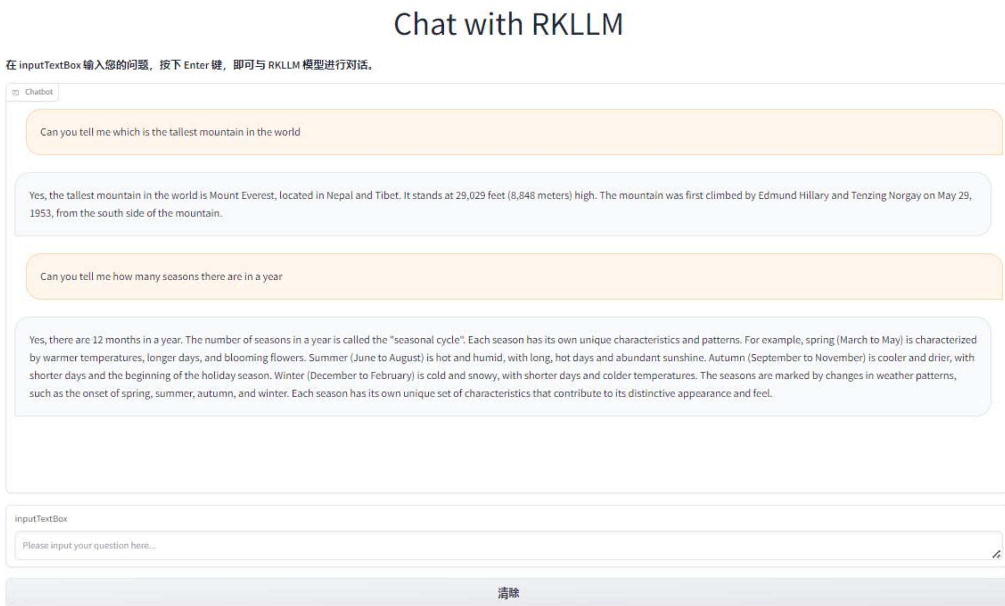
1) First, open a browser on any computer in the current LAN and directly access "Development Board IP:8080". The opened interface is as follows:



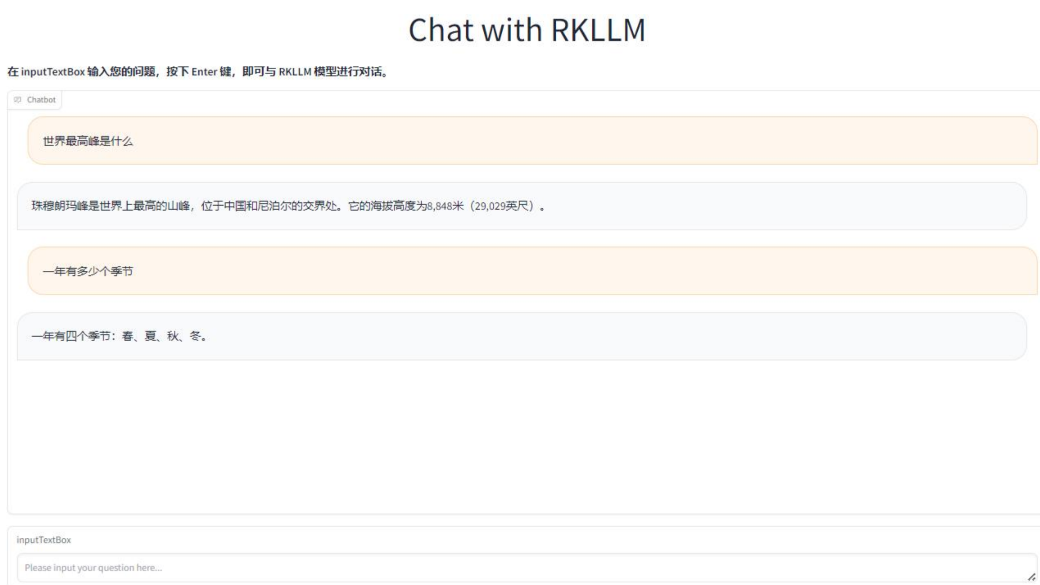
2) Then enter the question in the inputTextBox and press Enter.



a. Use the TinyLLAMA model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



b. Use the Qwen model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



c. Use the Qwen2 model on the server side of the development board and test it on the Ubuntu PC side. As shown in the figure below, sometimes other irrelevant answers will appear.

d. Use the Phi-3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

e. Use the ChatGLM3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

f. Use the Gemma model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

g. Use the InternLM2 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

h. Use the MiniCPM model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

3. 38. 6. Performance test results of RK3588 running RKLLM large model

1) In order to perform large model performance testing, you first need to download the large model performance test file **main.cpp** in the official tool. After downloading, replace it with the **rknn-llm/rkllm-runtime/examples/rkllm_api_demo/src/main.cpp** file used by the PC to compile the test code

2) Refer to the **Compile the test code** section to recompile the llm_demo file, and then run the large model according to the **detailed steps for deployment and operation on the development board** section.

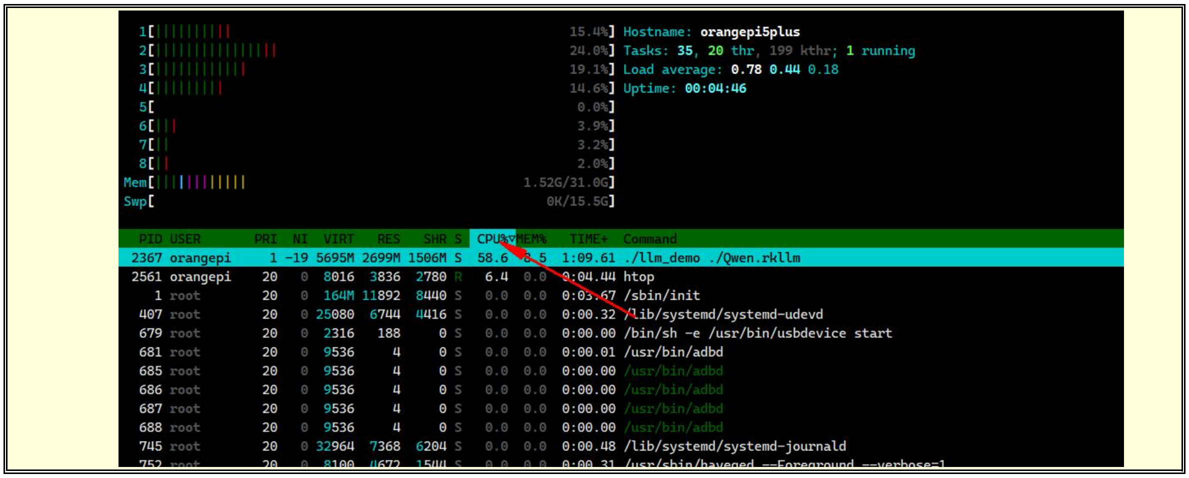
3) After the model runs, enter a question and then open a new terminal to test the performance. The performance test is when the model answers the question.

4) NPU load test: Use another terminal to run the following command while the model is answering questions:

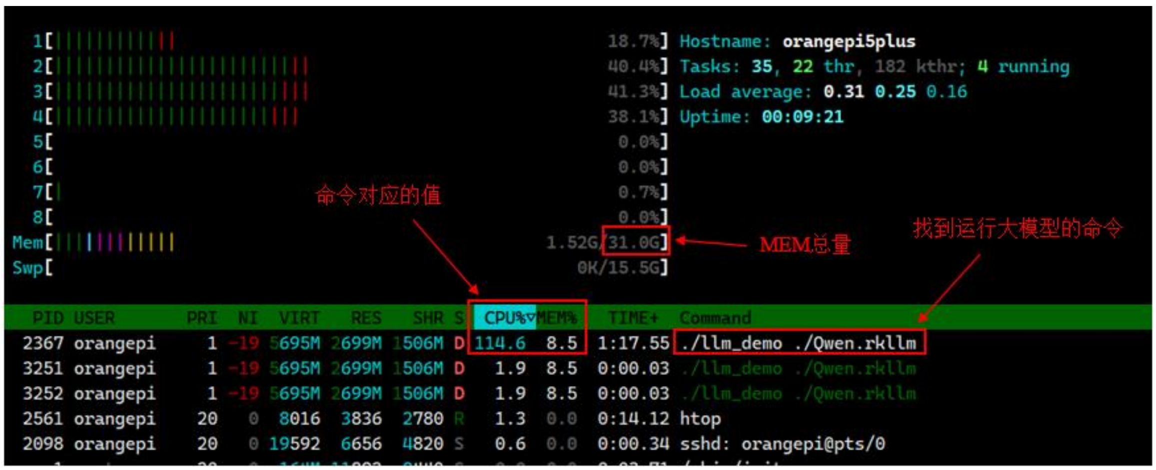
```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/load
NPU load:    Core0:    51%, Core1:    51%, Core2:    51%,
```

5) CPU load, memory: Use another terminal to run the following command while the model is answering questions:

> When calculating the CPU load, divide the CPU% value of the llm_demo process by the number of CPUs.
>
> When calculating memory, use the MEM% value of the llm_demo process * the total MEM
>
> You can click on the CPU option and the interface will be displayed in descending order based on CPU usage.



```
orangepi@orangepi:~$ htop
```



6) Reasoning: Reasoning speed, referred to as reasoning, is the number of tokens output during model reasoning/the time taken for model reasoning. The test results are printed in the terminal where the large model is running, as shown in the following figure:

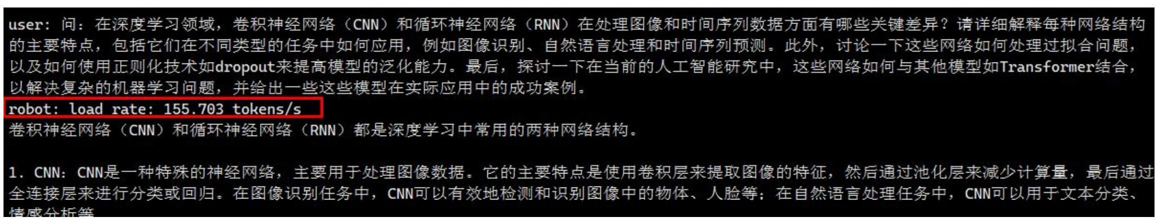
```
user: 3
把这句话翻译成中文: Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its own advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermore, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
robot: load rate: 251.511 tokens/s
知识可以从许多来源获得。这些包括书籍、教师和实践经验，每种都有其优势。从书籍和正规教育中获取的知识使我们能够学习我们在日常生活中无法体验的事情。我们还可以发展我们的分析技能，并学会以不同的方式看待和解释我们周围的世界。此外，我们可以通过阅读书籍来学习过去的经验。通过这种方式，我们将不会重复他人的错误，并可以建立在他们的成就之上。

Total tokens processed: 88
Time taken for last token: 10.5241 seconds
Token rate: 9.25709 tokens/s
```

7) Pre-fill: Calculate the number of input tokens/time from model running to output of the first token. Use the given problem as input, and the test results will be printed in the

terminal where the large model is running.

Since different large language models may use different word segmentation strategies when processing the same sentence, resulting in differences in the number of generated tokens, and RKLLM does not provide a corresponding channel for obtaining the actual number of input tokens, we used GPT to generate questions with 256 tokens as input, resulting in a certain error in the test results.

Q: In the field of deep learning, what are the key differences between convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in processing images and time series data? Please explain in detail the main features of each network structure, including how they are applied in different types of tasks, such as image recognition, natural language processing, and time series prediction. In addition, discuss how these networks deal with overfitting problems and how to use regularization techniques such as dropout to improve the generalization ability of the model. Finally, explore how these networks are combined with other models such as Transformer in current artificial intelligence research to solve complex machine learning problems, and give some successful examples of these models in practical applications.



8) The test results of all models are shown in the following table:

模型	参数大小	dtype	性能	CPU 负载	NPU 负载	内存占用
TinyLLAMA	1.1B	W8a8	Pre-population:58.6157 token/s Reasoning:12.7262 token/s	15.9%	3*49%	1.376G
Qwen	1.8B	W8a8	Pre-population:168.525 token/s Reasoning:10.8891 token/s	13.7%	3*50%	2.72G
Qwen2	0.5B	W8a8	Pre-population:440.511 token/s	17.75%	3*34%	1.344G

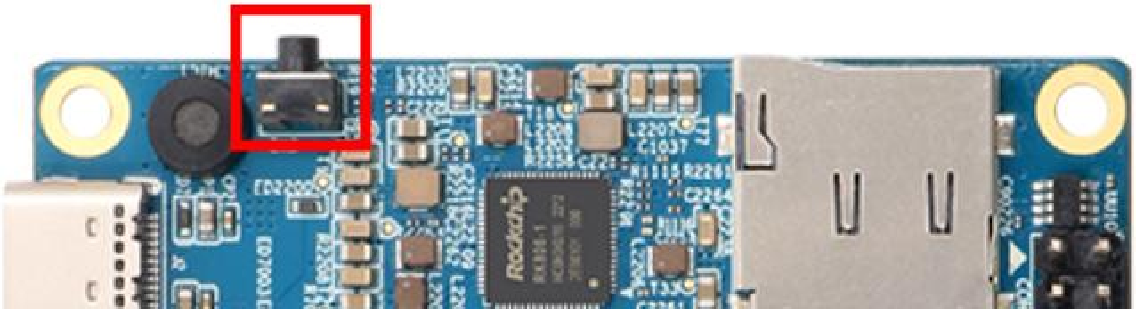
			Reasoning:17.4542 token/s			
Phi-3	3.8B	W8a8	Pre-population:22.8119 token/s Reasoning:4.72983 token/s	13.13%	3*62%	4.288G
ChatGLM3	6B	W8a8	Pre-population:48.8464 token/s Reasoning:3.80383 token/s	8.3%	3*75%	7.04G
Gemma	2B	W8a8	Pre-population:112.489 token/s Reasoning:6.41746 token/s	8.25%	3*64%	4.8G
InternLM2	1.8B	W8a8	Pre-population:117.099 token/s reasoning：9.139 token/s	11.87%	3*57%	2.432G
MiniCPM	2B	W8a8	Pre-population:77.4655 token/s Reasoning:6.16648 token/s	16.25%	3*52%	3.904G

3. 39. The method of shutting down and restarting the development board

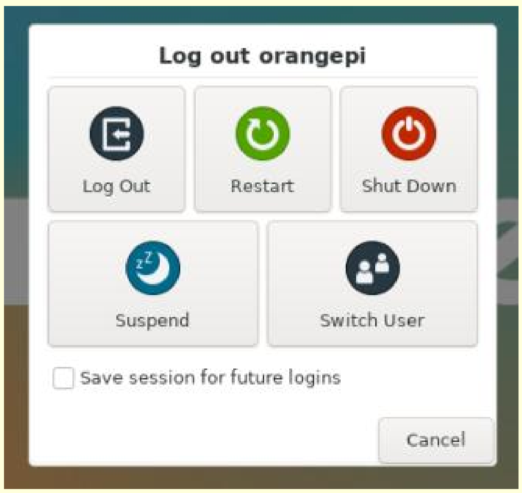
1) In the process of running the Linux system, if the Type-C power supply is directly out of power, it may cause the file system to lose certain data or damage. Therefore, please use the **poweroff**command to turn off the linux system of the development board before power off. Unplug the power supply。

```
orangepi@orangepi:~$ sudo poweroff
```

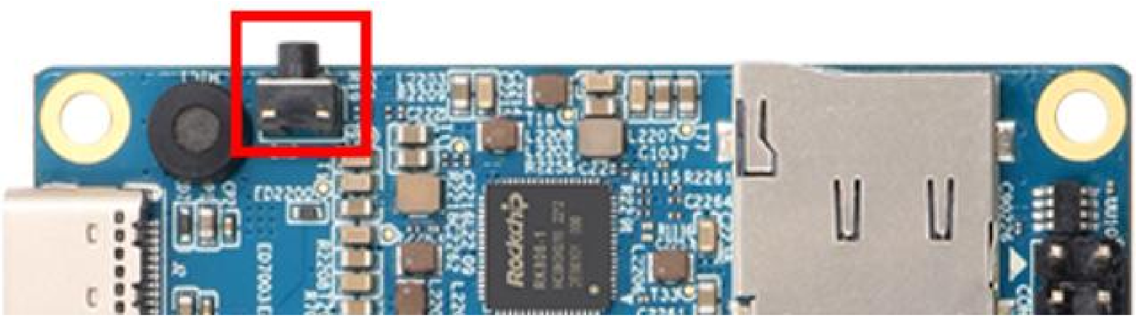
2) In addition, the development board is equipped with a switch button, and you can also short press the switch button on the development board to turn off。



Note that the Linux desktop system will pop up the confirmation box shown in the figure below after pressing the buttons. You need to click the Shut Down option to shut down。



3) After shutting down, press the switch button on the development board to turn on



4) Restart the command of the linux system to be

orangepi@orangepi:~$ **sudo reboot**

4. ubuntu22.04 Gnome Wayland desktop system use instructions

ubuntu22.04 gnome Image default pre -installed panfork mesa user space library, pre -installed Kodi player and Chromium browser support hard solution play video。

It should be noted that this image needs to be used under wayland. If you need to use x11, please select the xfce type image。

4. 1. Ubuntu22.04 Gnome Desktop system adaptation situation

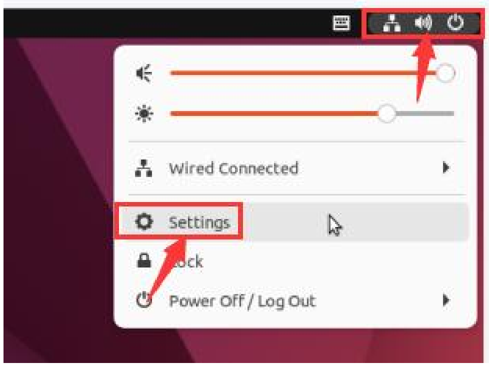
Function	Ubuntu22.04 Gnome Wayland
USB2.0x2	OK
USB3.0x1	OK
USB Type-C 3.0	OK
USB Start system	OK
RTL8821CU USB Network card	OK
RTL8723BU USB Network card	OK
RTL8811 USB Network card	OK
DP display	OK
M.2 NVMe SSD start up	OK
M.2 SATA SSD start up	OK
AP6275P-WIFI	OK
AP6275P-Bluetooth	OK

GPIO（26pin）	OK
UART（26pin）	OK
SPI（26pin）	OK
I2C（26pin）	OK
CAN（26pin）	OK
PWM（26pin）	OK
3pin Debug serial port	OK
TF Card Start	OK
HDMI Video	OK
HDMI Audio	OK
OV13850 Camera	OK
OV13855 Camera	OK
LCD1	OK
LCD2	OK
Gigabit network	OK
Network port state light	OK
MIC	OK
Headphones play	OK
Headphone recording	OK
LED light	OK
GPU	OK
NPU	OK
VPU	OK
Switch button	OK
Watch Dog Test	OK
Chromium Hard solution video	OK
Kodi solution video	OK
MPV solution video	OK

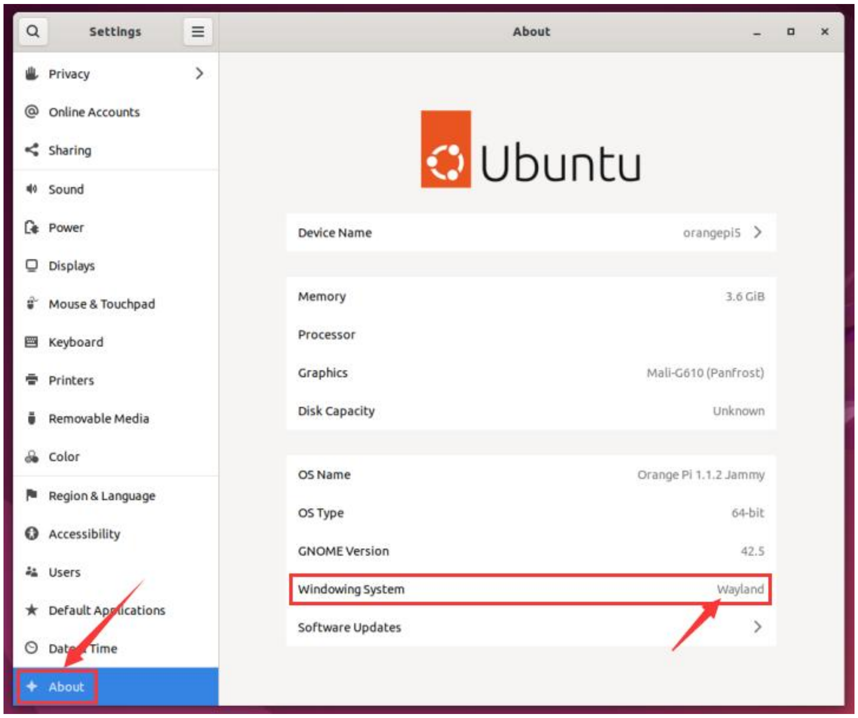
4.2. Confirm that the current window system used by the system is Wayland method

1) The system used by the system default is Wayland, and the confirmation method is shown below：

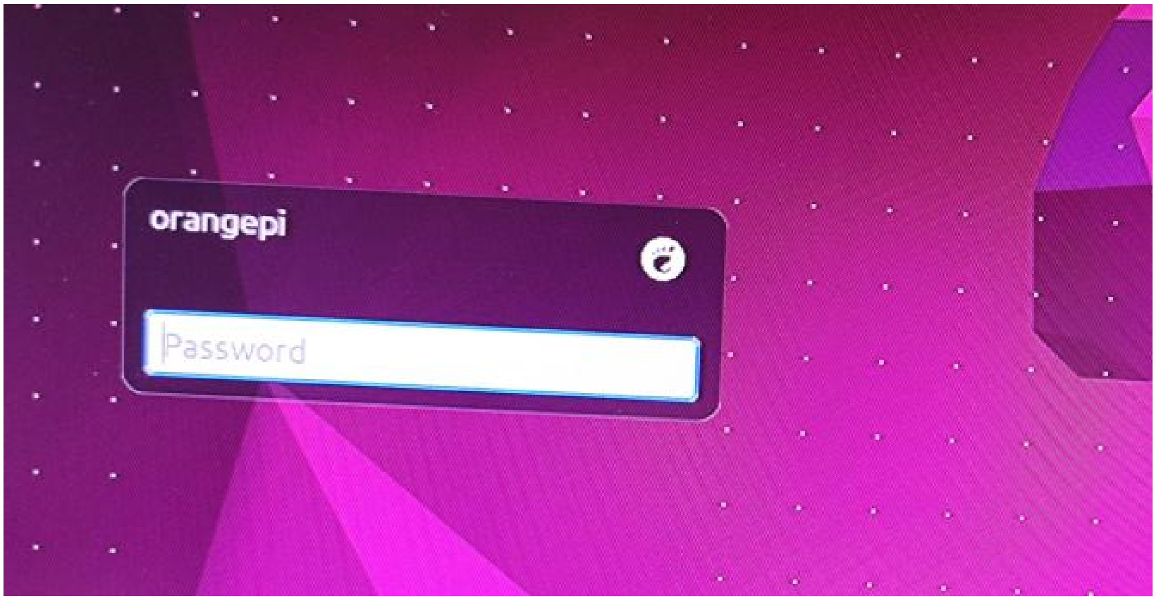
a. First open the settings



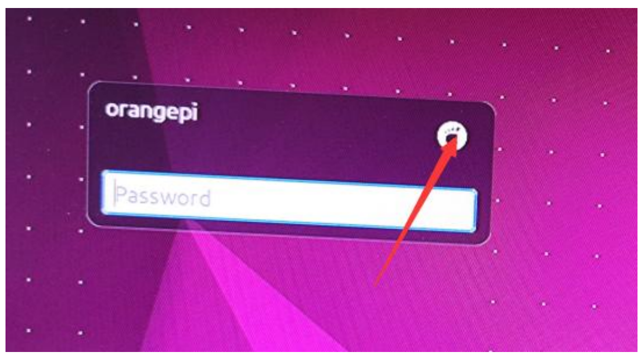
b. Then select an **About**, if the **wayland** description settings displayed by **Windowing System** in a column are correct.



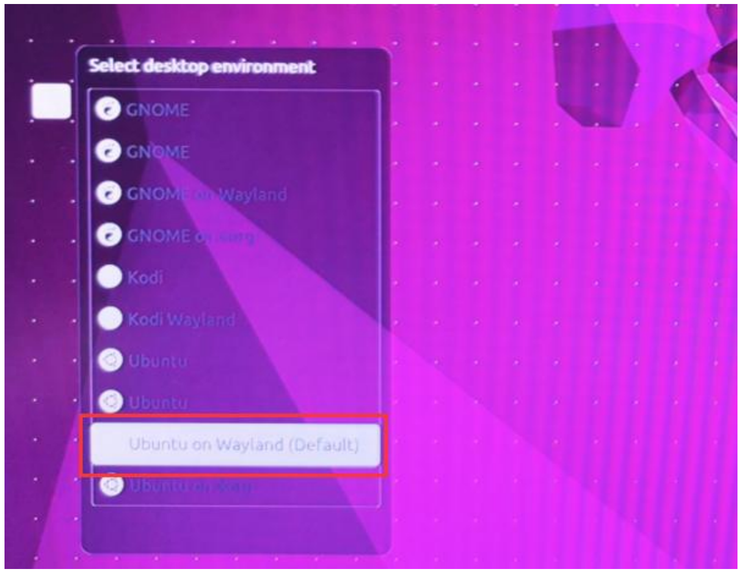
2) When the **Log Out** is out of the system, it will enter the login interface below



3) Please click on the location shown in the figure below before logging in again

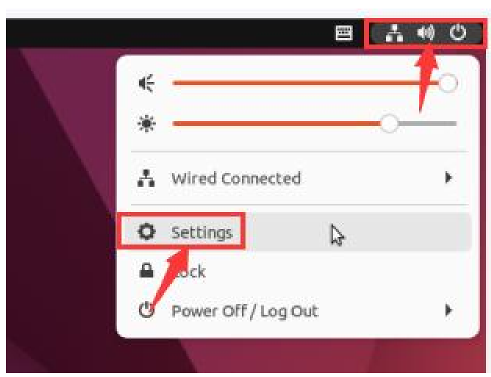


4) Then select **Ubuntu on Wayland**, and then enter the password to log in to the system

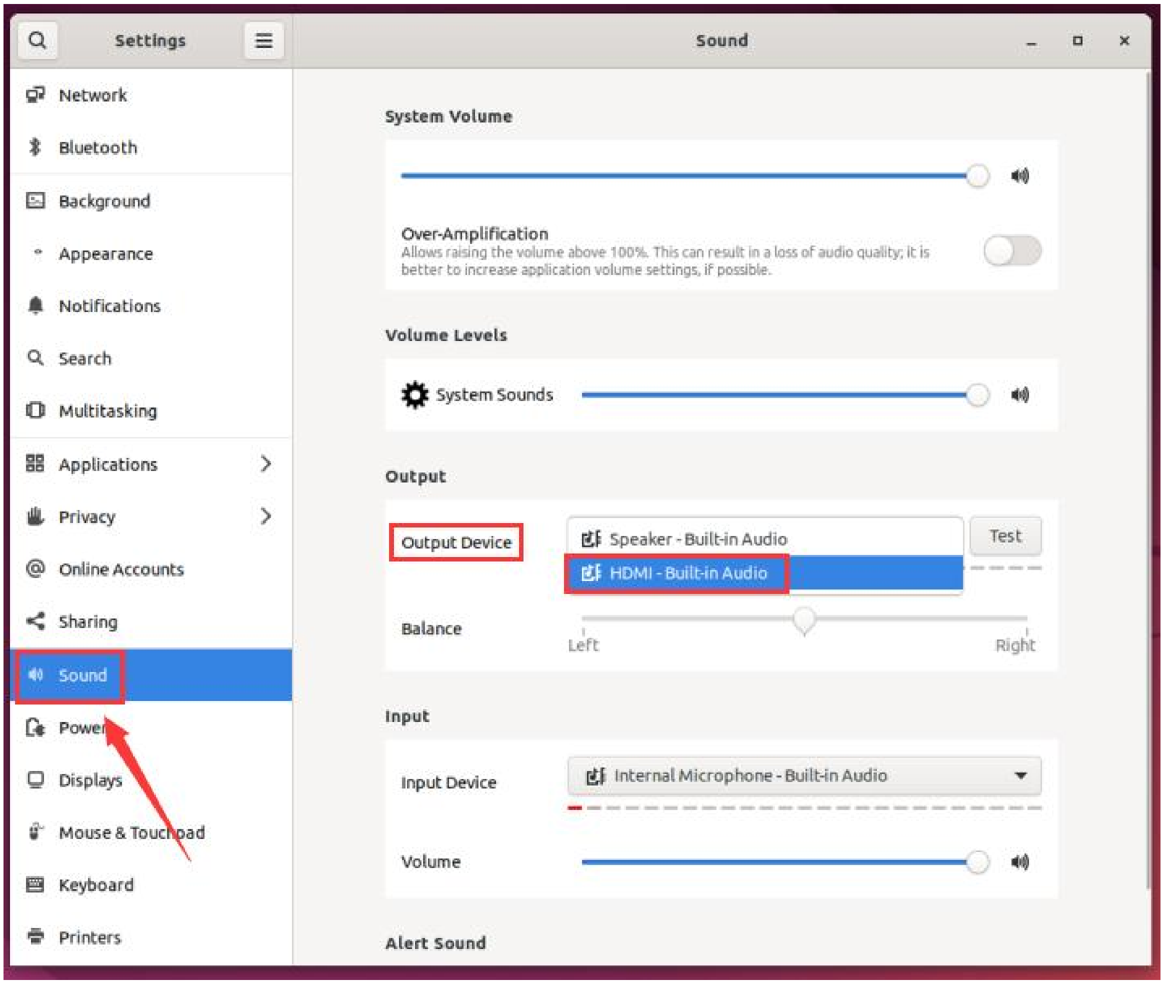


4. 3. How to switch the default audio equipment

1) Open the settings first



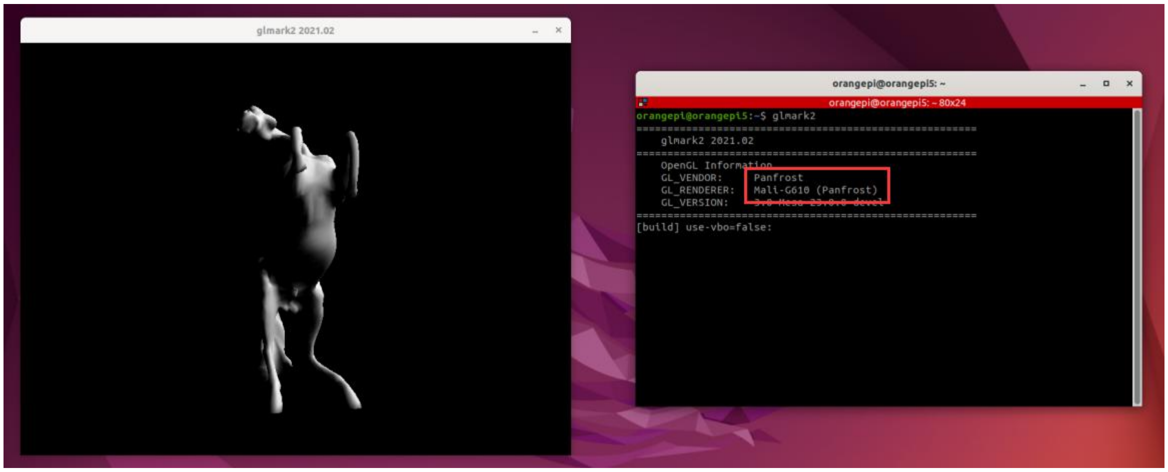
2) Then select **Sound**, and select the audio device you want to use in the **Output Device**



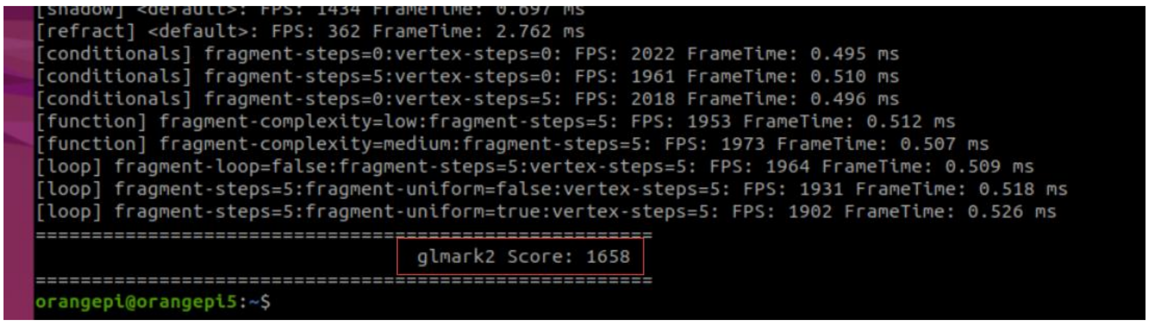
4. 4. GPU test method

1) Open a terminal on the desktop, and then enter the **glmark2** command. If you can see the **GL_VERDOR**, the **Panfrost** description can be used in the GPU

```
orangepi@orangepi:~$ glmark2
```

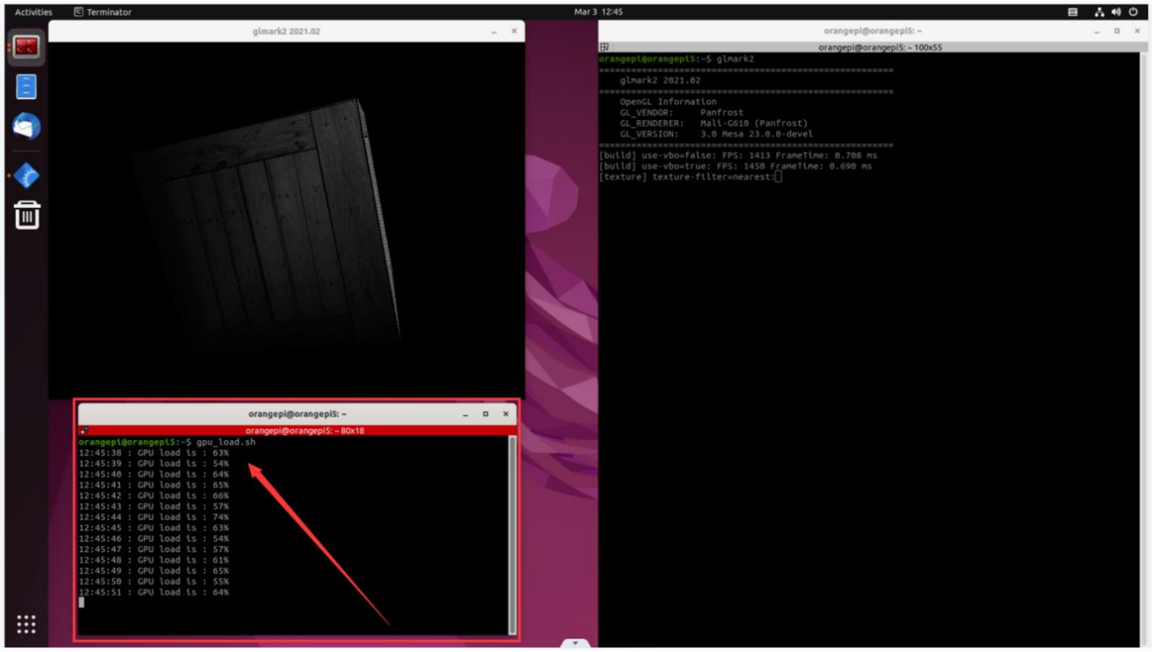


2) glmark2 running score test is generally more than 1,000 points



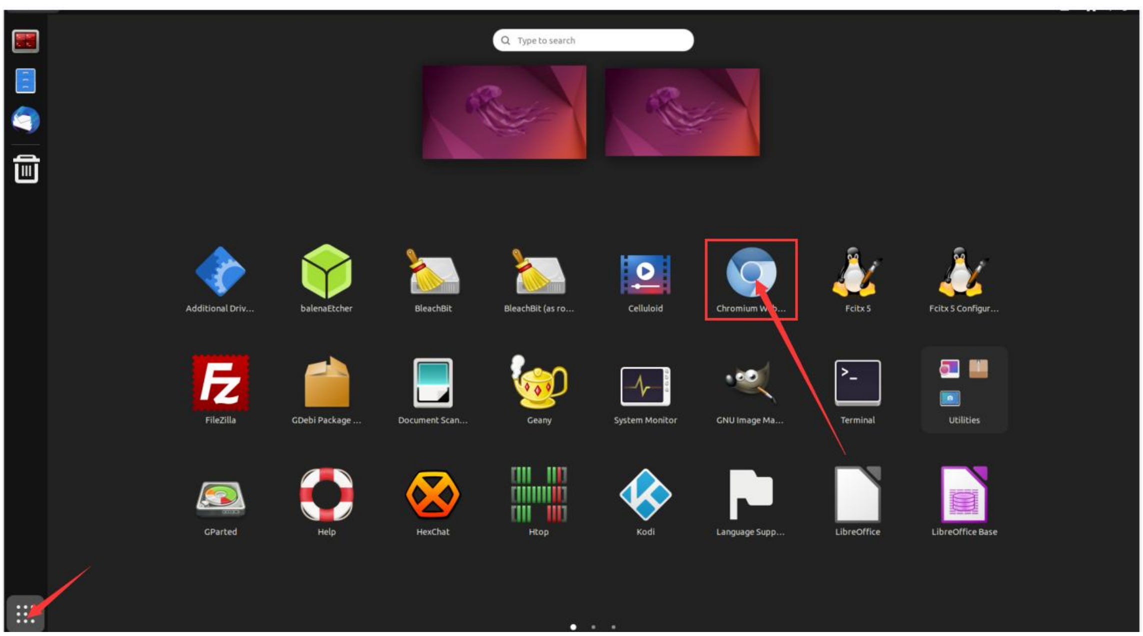
3) Run **gpu_load.sh** script can view the current load of GPU

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

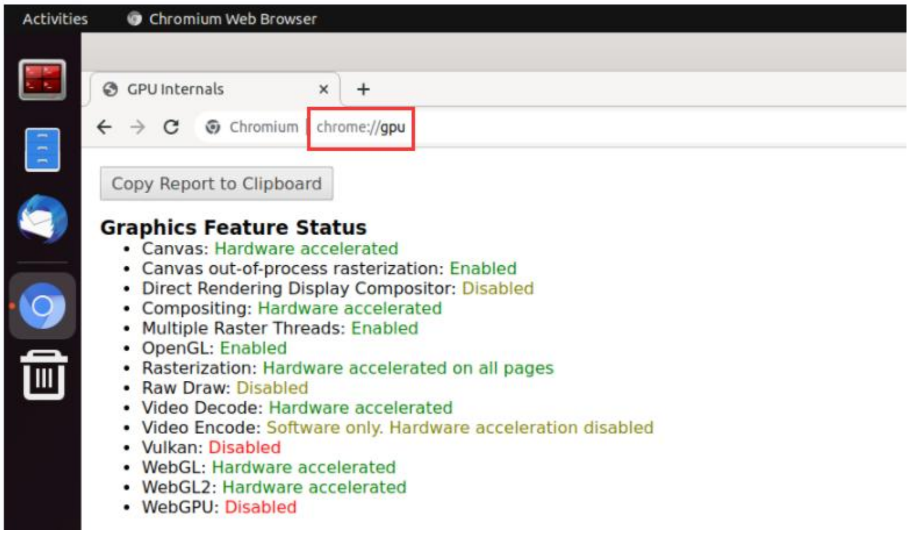


4. 5. Chromium Browser Belly Play Video Test Method

1) First open the Chromium browser



2) Then enter **chrome://gpu** in the Chromium browser to view the support of GPU and video decoding.

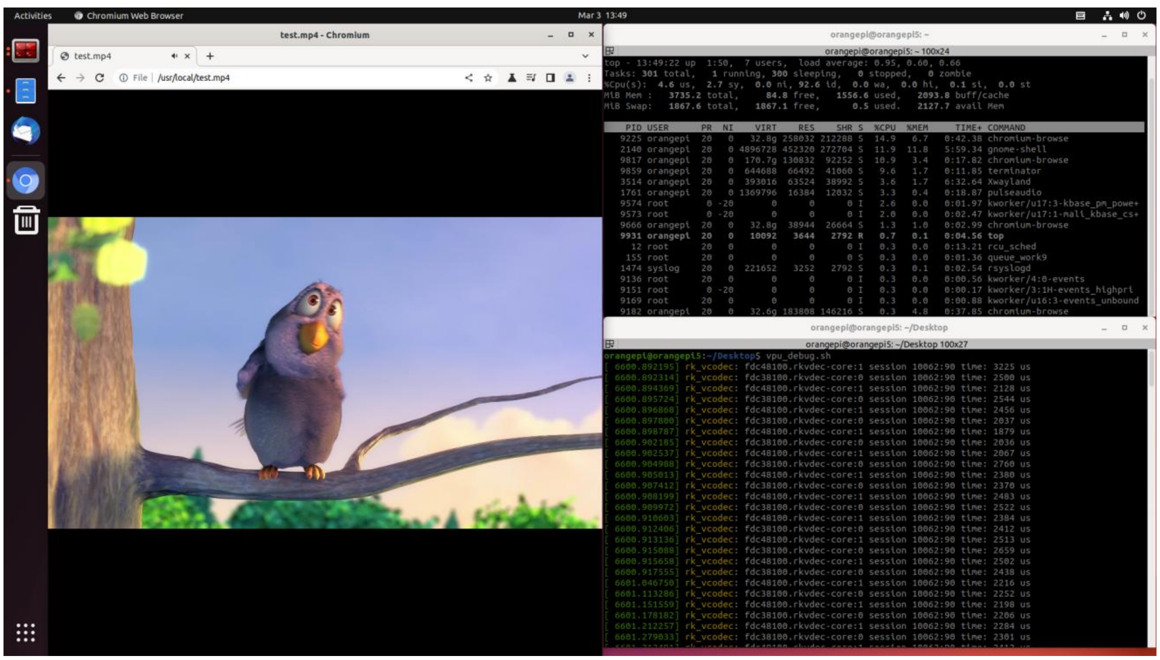


3) Then you can open the video website to play a video file, or enter the following path name player to play a test video file in the browser.

```
/usr/local/test.mp4
```

4) When playing the video, you can run the **vpu_debug.sh** script in the terminal. If there is a printing output in the lower right corner of the figure below, it means that there is a hardware to decode the video.

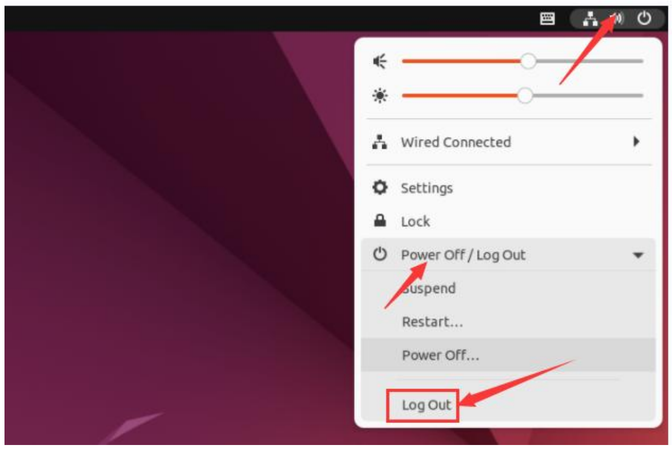
```
orangepi@orangepi:~$ vpu_debug.sh
```

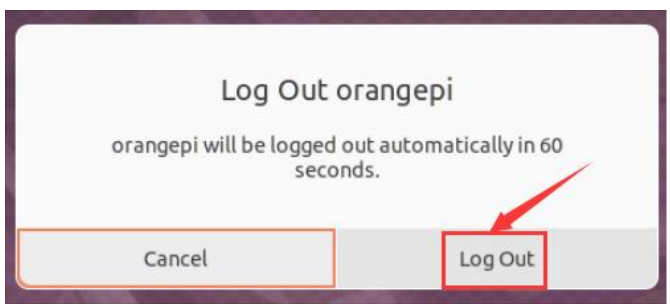


4.6. Kodi hard solution to play video test method

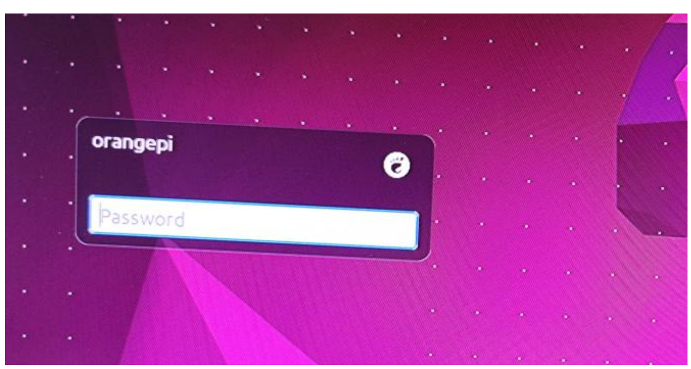
Note that there will be problems with the Kodi display directly on the Wayland desktop. Please open Kodi strictly according to the following method.

1) First log in the system

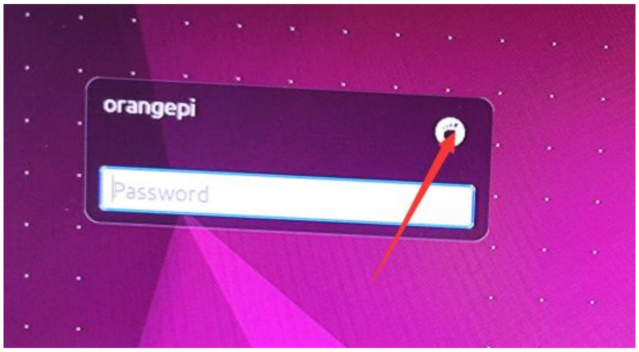




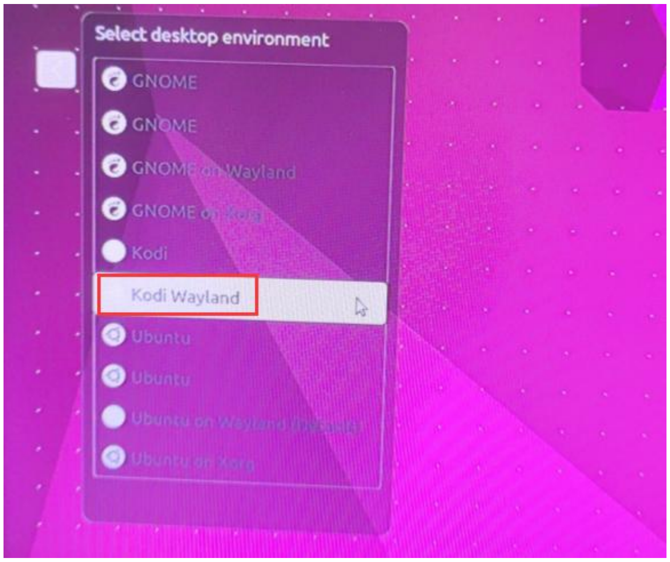
2) When the login system will enter the login interface below



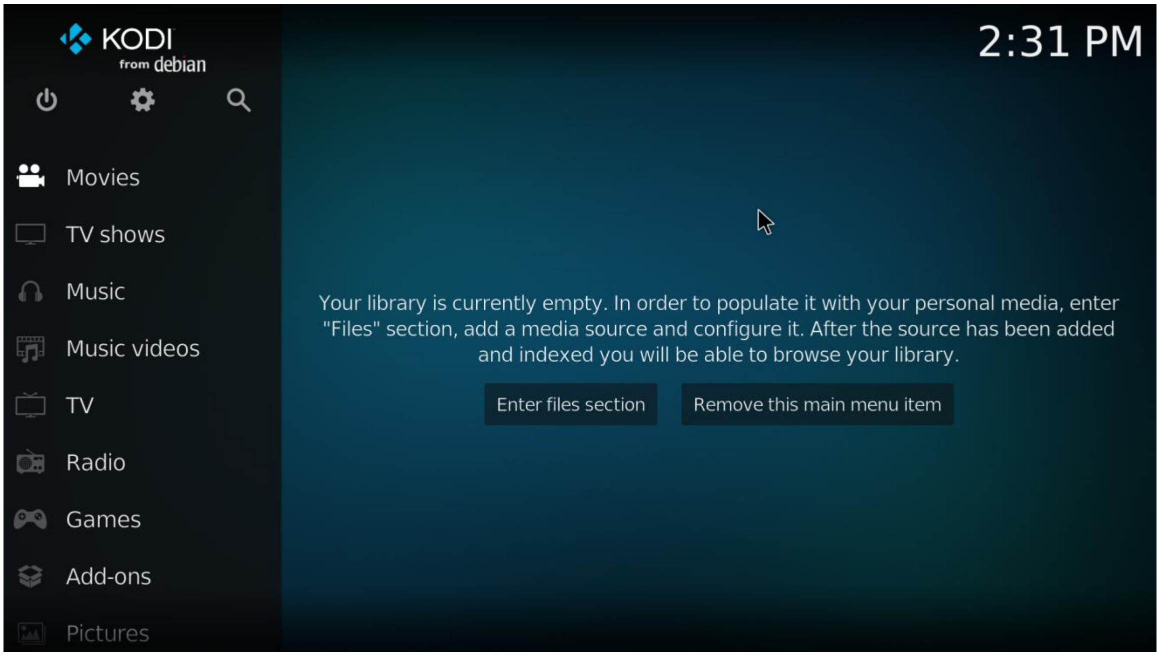
3) Then click the location shown in the figure below



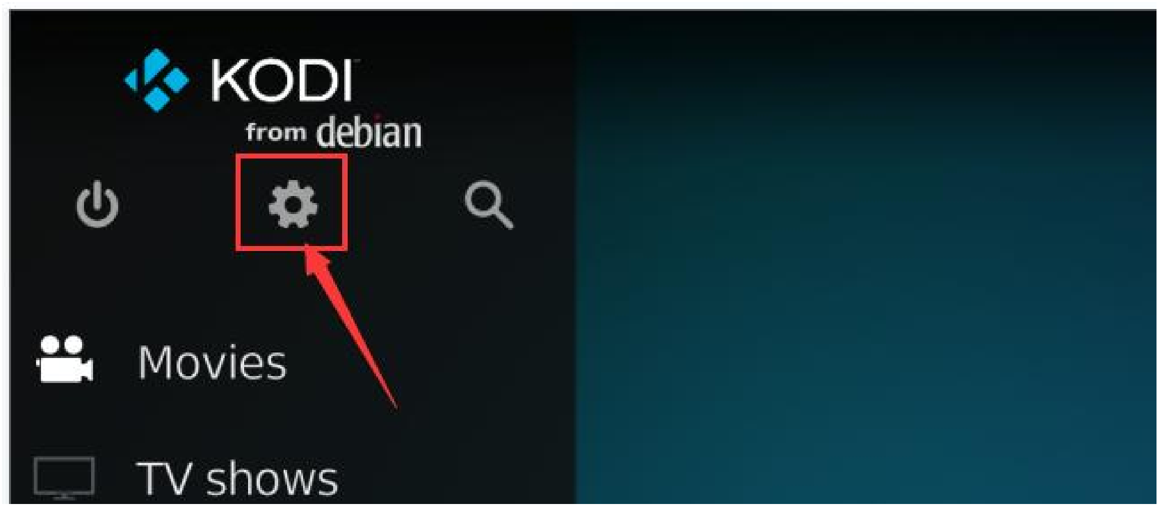
4) Then select **Kodi Wayland**, and then enter the password login system



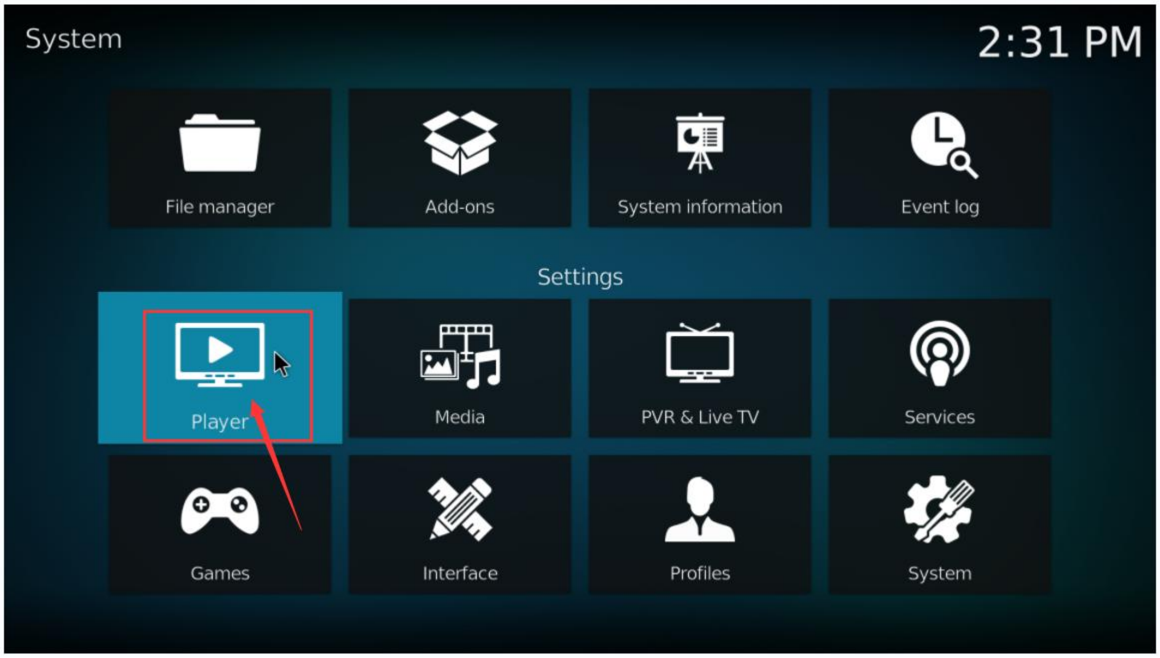
5) The interface after Kodi is opened is displayed as shown below



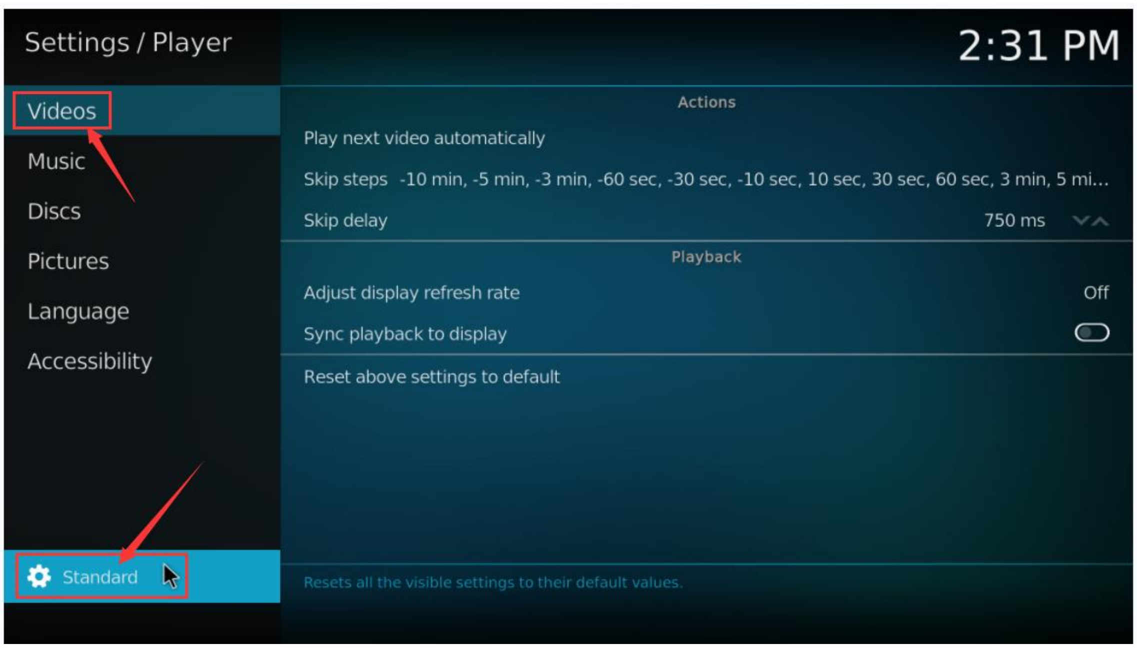
6) Then click Settings



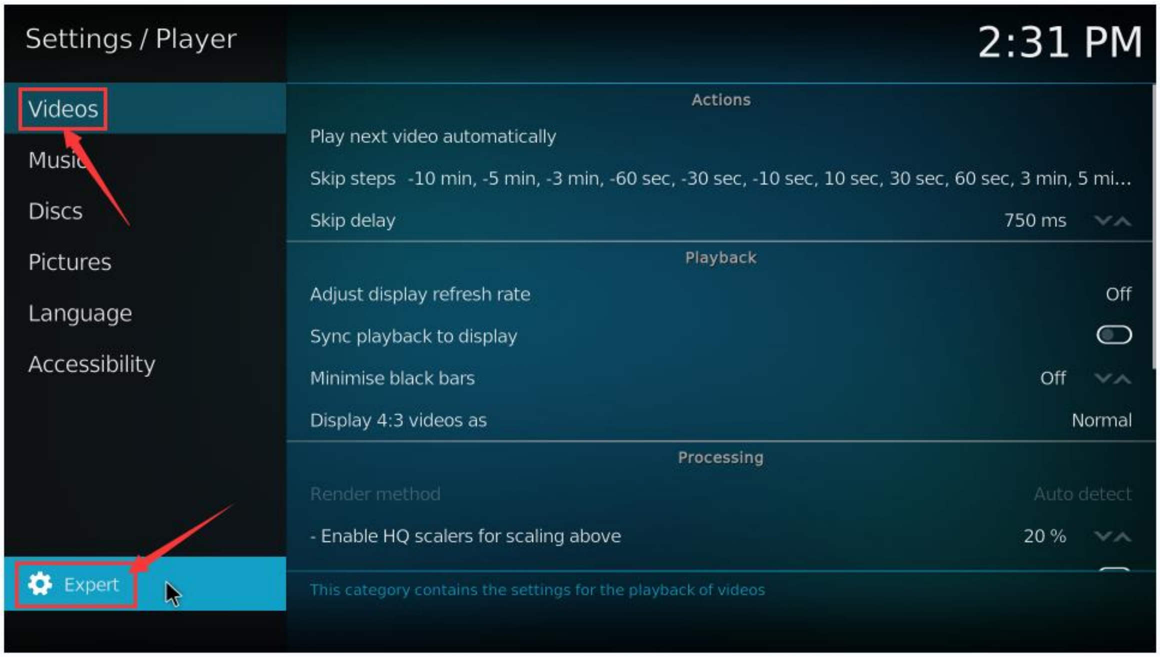
7) Then select **Player**



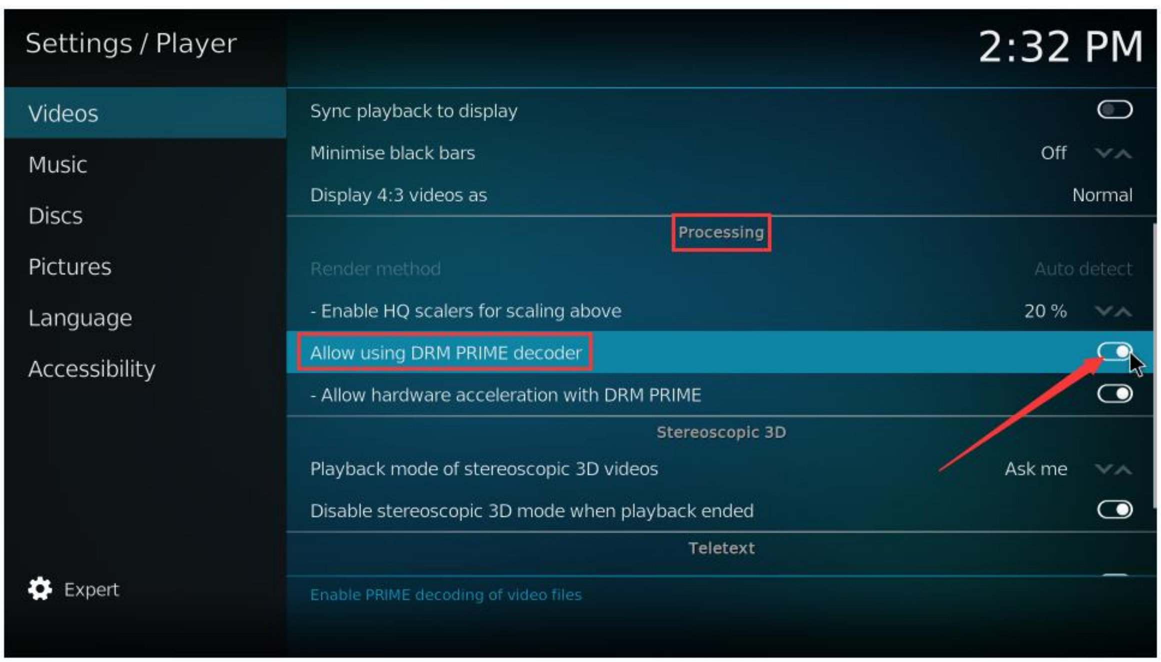
8) Then select **Videos**, and then click **Standard** in the lower left corner



9) After clicking twice, it will be switched to the **Expert** mode

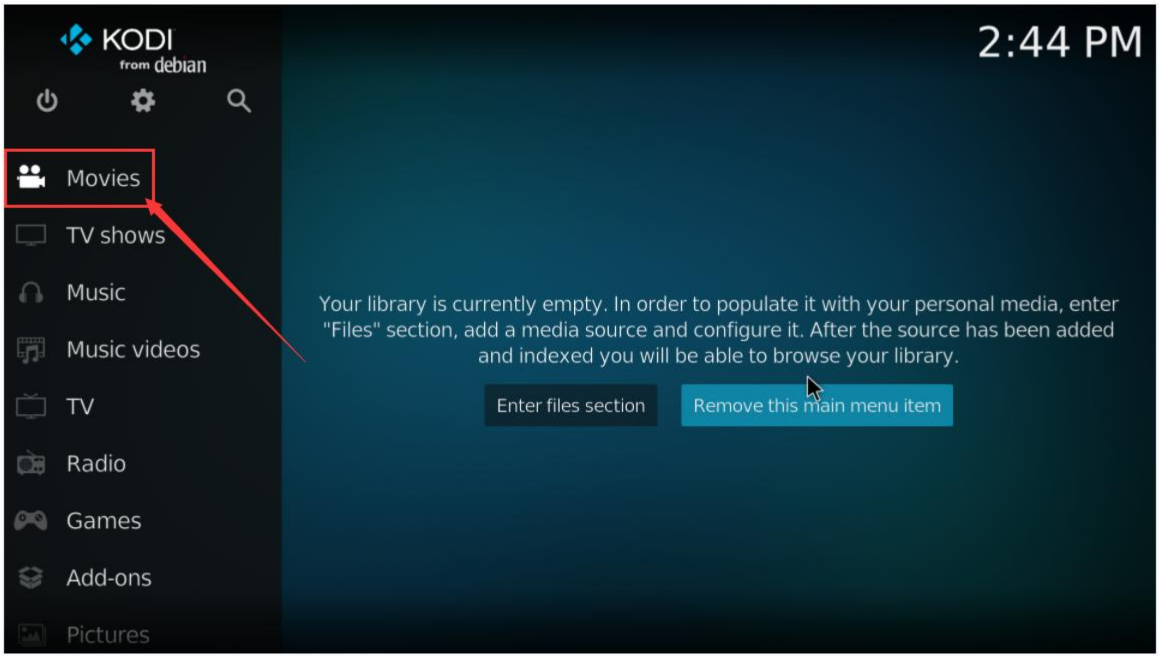


10) Then open the **Allow using DRM PRIME decoder** in the **Processin** settings

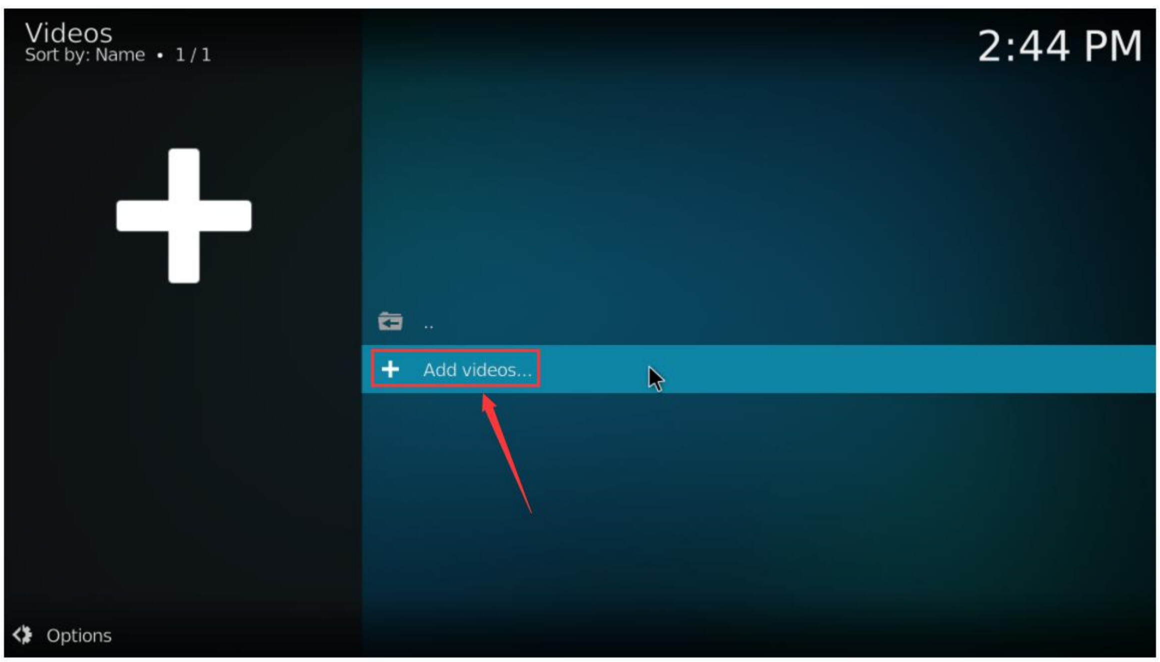


11) Then let's introduce a system's own test video test, you can also upload the video you want to play to the system, and then import and play

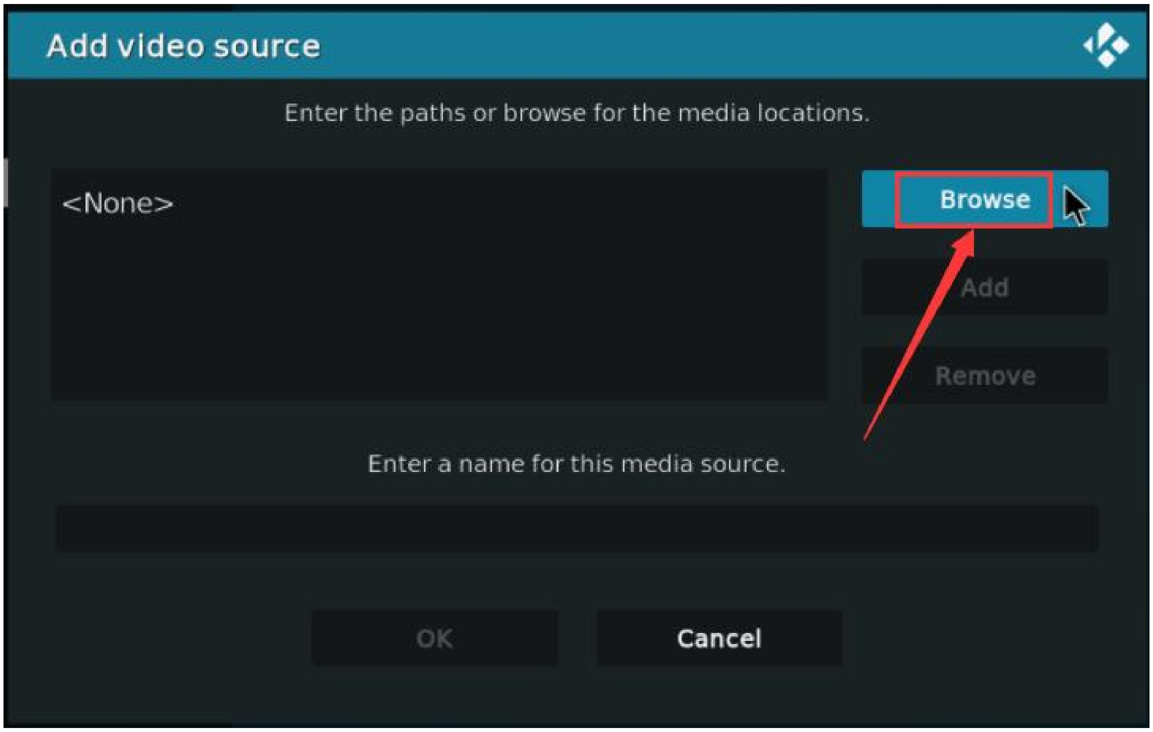
a. First enter the main interface, then select **Movies**



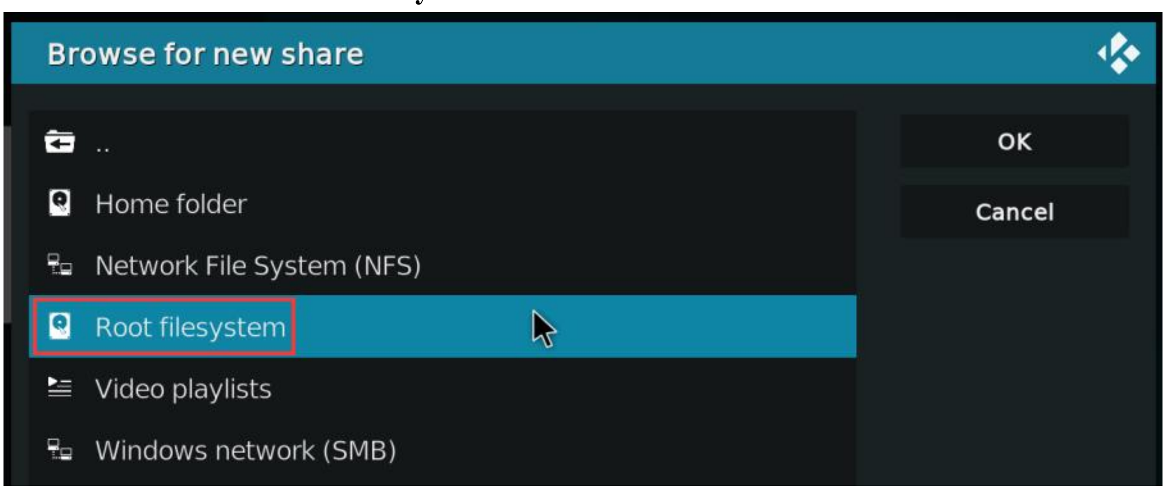
b. Then choose **Add videos...**



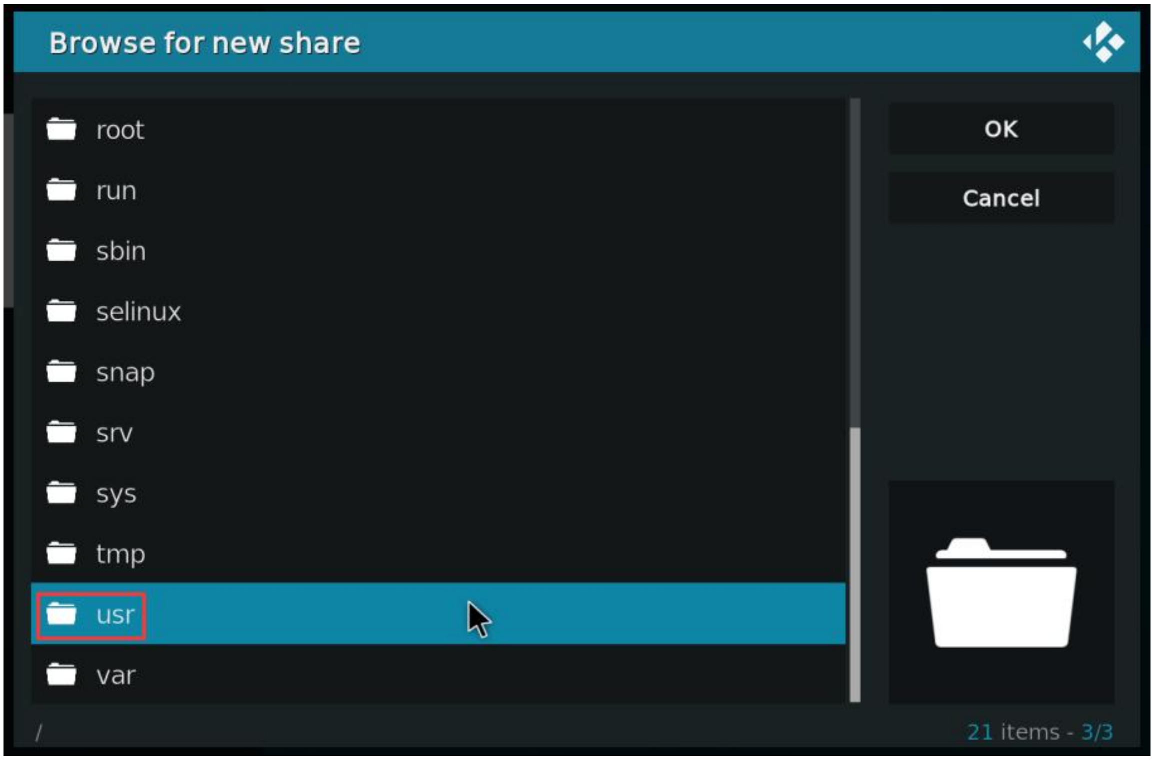
c. Then choose **Browse**



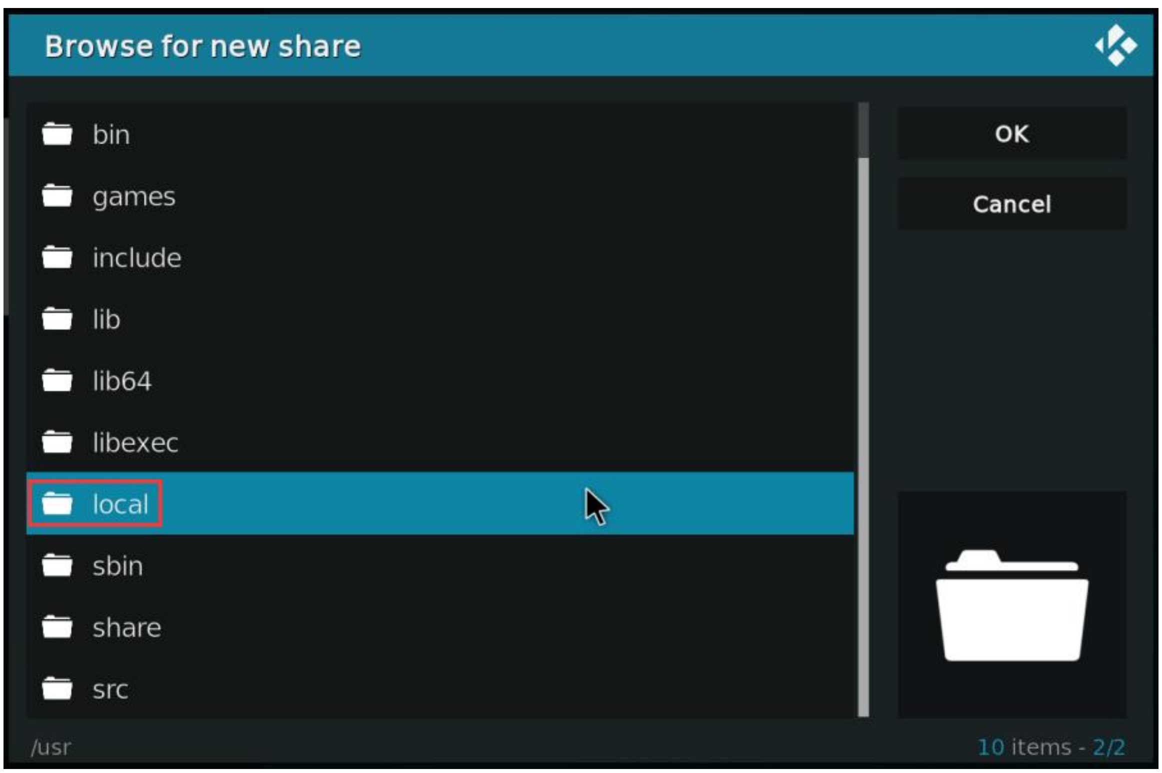
d. Then choose **Root filesystem**



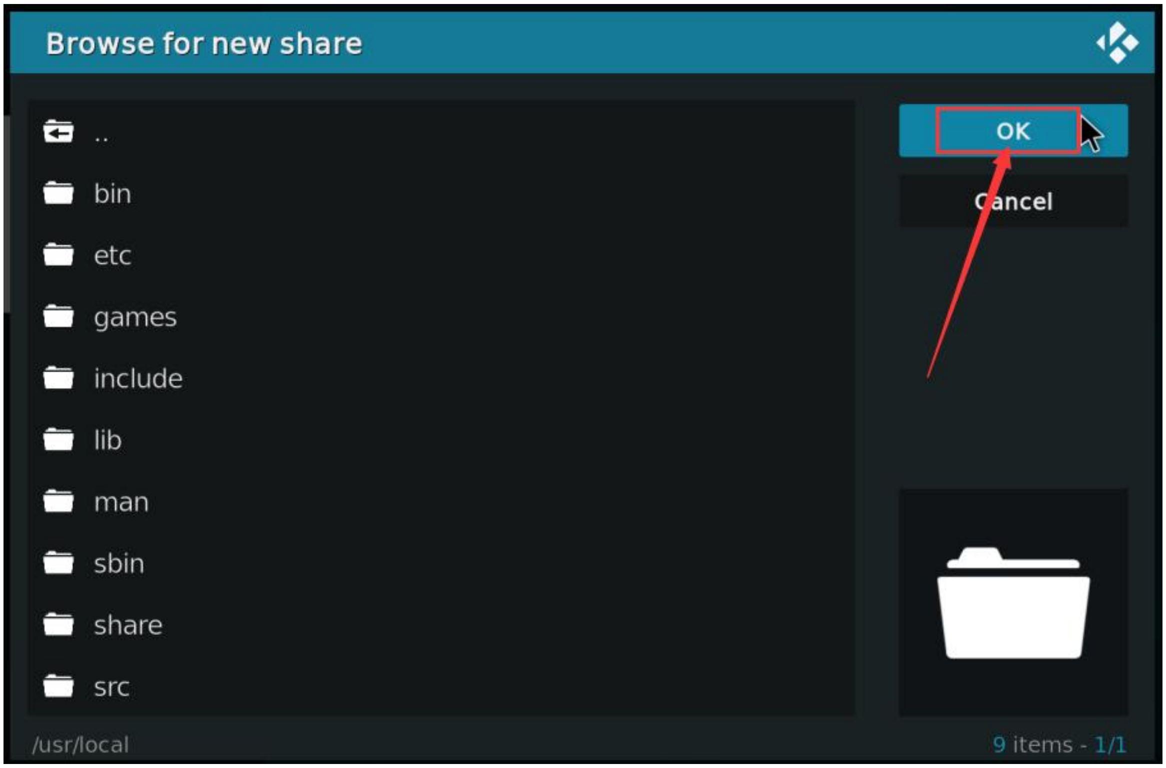
e. Then choose **usr**



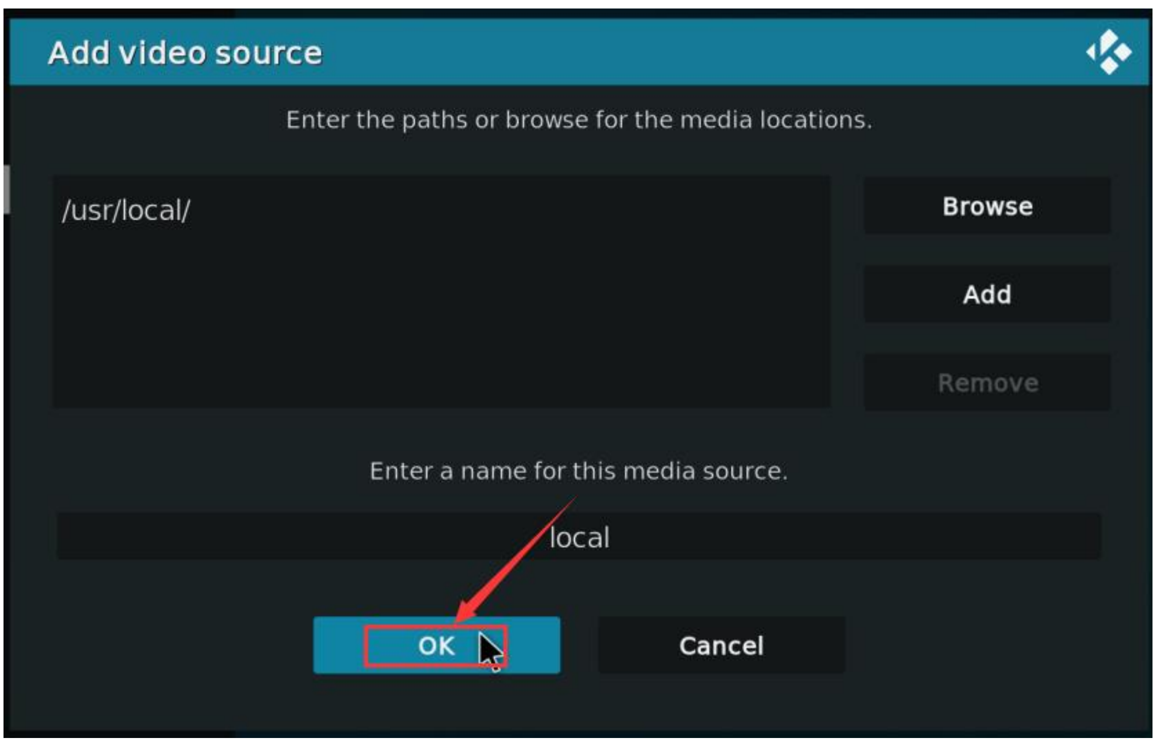
f. Then choose **local**



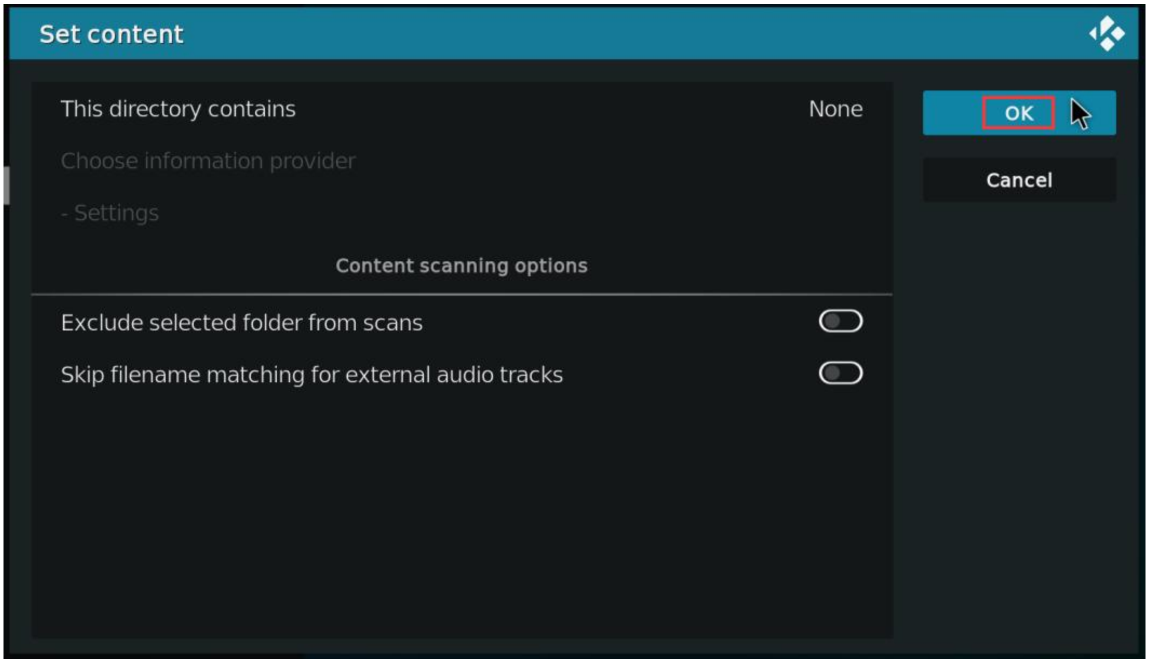
g. Then choose **OK**



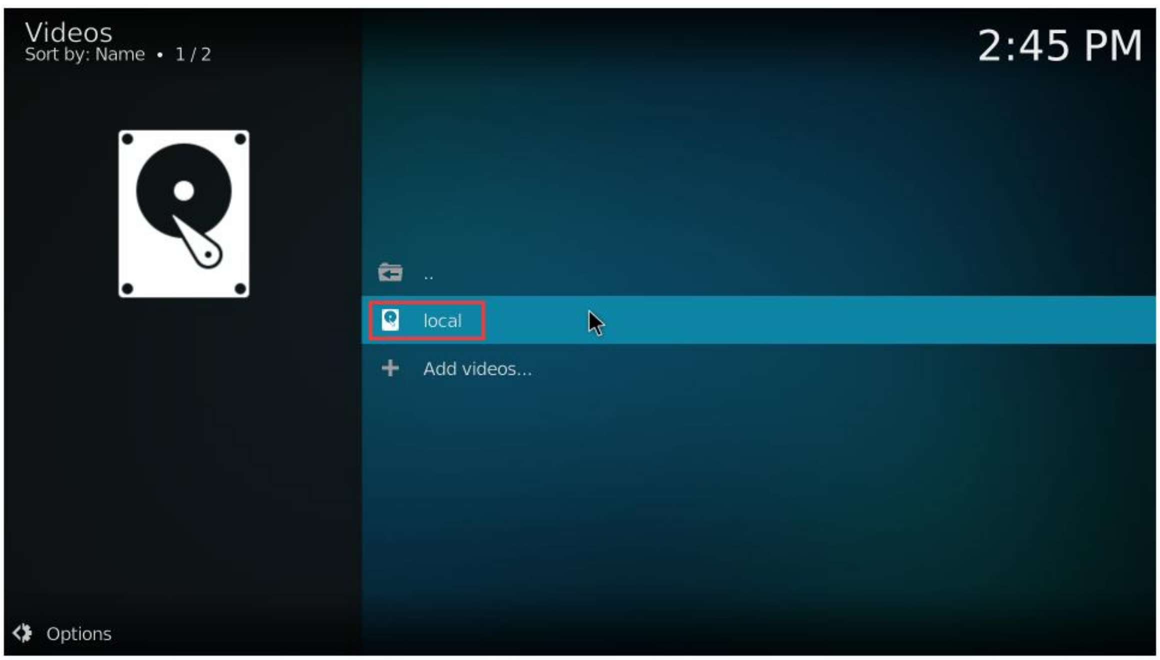
h. Then choose **OK**



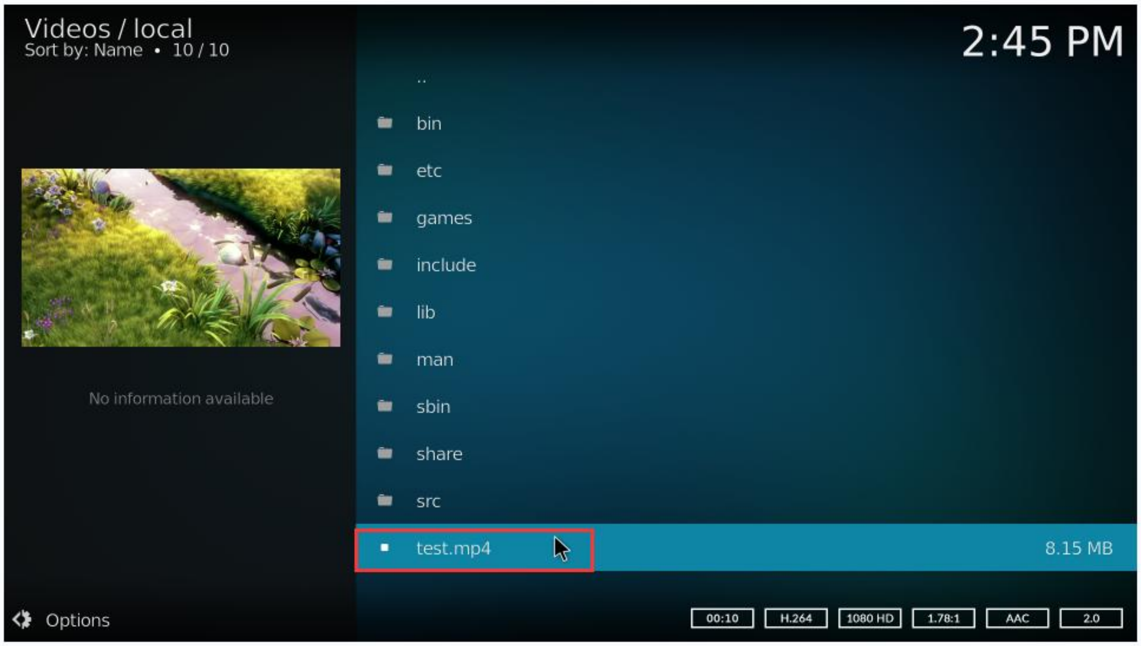
i. Then choose OK



j. Then enter the local folder



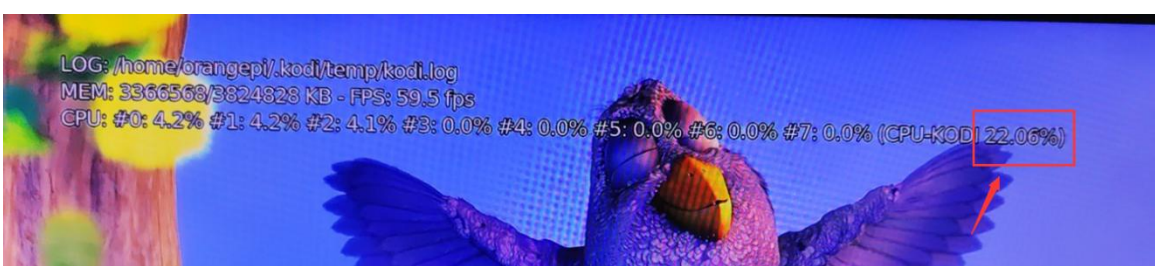
k. Then you can play **test.mp4** test video



12) When playing the video, you can run the **vpu_debug.sh** script under the command line (via SSH or serial port). If there is a print output below

```
orangepi@orangepi:~$ vpu_debug.sh
[ 1830.938378] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2728 us
[ 1830.938461] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2617 us
[ 1830.941179] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2661 us
[ 1830.941777] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2708 us
[ 1830.944727] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 3444 us
[ 1830.945211] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 3331 us
[ 1830.970563] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2547 us
[ 1831.199650] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2703 us
```

13) Play **test.mp4** video file CPU occupation rates of about **20%~30%**。



4. 7. Ubuntu22.04 Gnome to install ROS 2 Humble

1) You can install ros2 with **install_ros.sh** script

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

2) install**_ros.sh** script will automatically run the **ros2 -h** command after ROS2 is installed. If you can see the printing below, it means that ROS2 installation is complete

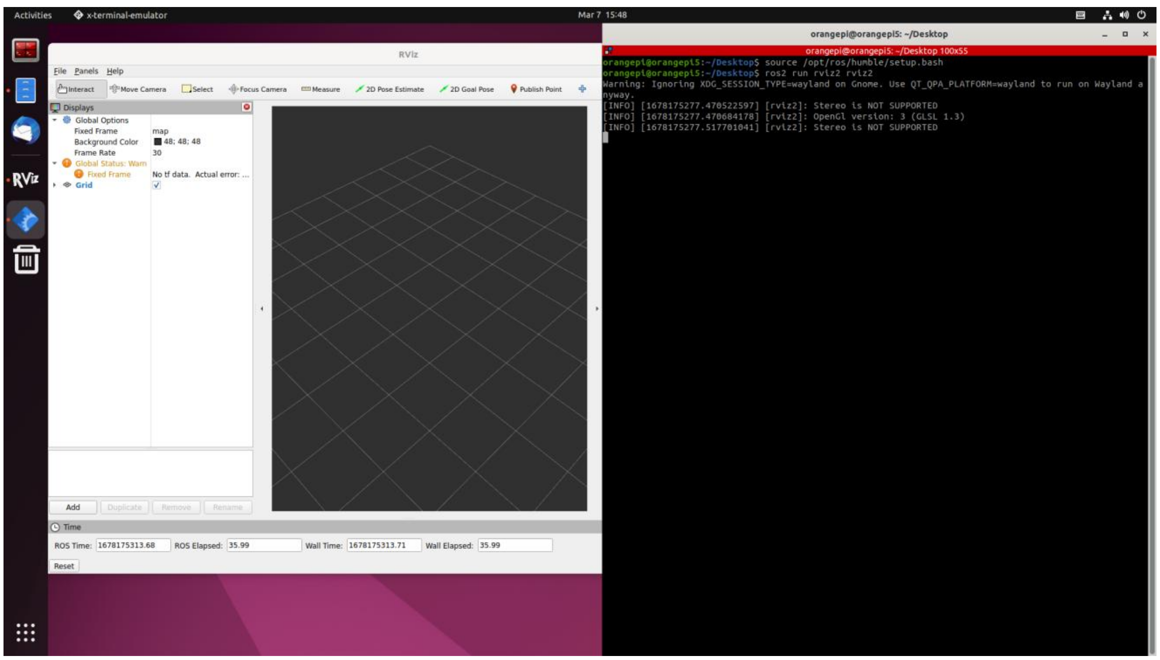
```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

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

3) Then you can use the **test_ros.sh** script to test whether the ROS 2 is successfully installed. If you can see the printing below, it means that ROS 2 can run normally

```
orangepi@orangepi5:~$ 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 document

http://docs.ros.org/en/humble/index.html
http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

5. Orange Pi OS Arch System use instructions

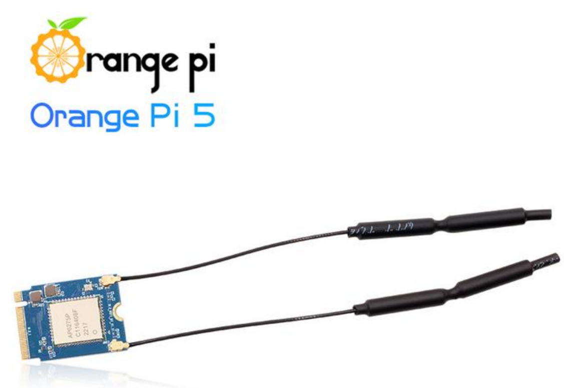
5. 1. Orange Pi OS Arch System adaptation

Function	OPi OS Arch Gnome Wayland
USB2.0x2	OK
USB3.0x1	OK
USB Type-C 3.0	OK
USB Start system	OK
RTL8821CU USB Network card	OK
RTL8723BU USB Network card	OK
RTL8811 USB Network card	OK
DP Display	OK
M.2 NVMe SSD Start up	OK
M.2 SATA SSD Start up	OK
AP6275P-WIFI	OK
AP6275P-Bluetooth	OK
GPIO（26pin）	OK
UART（26pin）	OK
SPI（26pin）	OK
I2C（26pin）	OK
CAN（26pin）	OK
PWM（26pin）	OK
3pin Debug serial port	OK
TF Card Boot	OK
HDMI Video	OK
HDMI Audio	OK
OV13850 Camera	OK
OV13855 Camera	OK
LCD1	OK
LCD2	OK
Gigabit network	OK
Network port state light	OK

MIC	**OK**
Headphones Play	**OK**
Headset recording	**OK**
LED Light	**OK**
GPU	**OK**
NPU	**NO**
VPU	**OK**
Switch Button	**OK**
Watch Dog Test	**OK**
Chromium Solution Video	**NO**
MPVSolution Video	**OK**

5. 2. AP6275P PCIe WIFI6+The method to use Bluetooth module

1) First of all, you need to buy an AP6275P PCIe module shown below



2) Then insert the AP6275P module into the M.2 interface of the development board and fix it.

3) Then open the configuration of the AP6275P module in the OPi OS Arch system (the dts configuration of the AP6275P module is closed by default), and the steps are shown below：

a. First add a line of configuration in **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-wifi-ap6275p.dtbo    #What needs to be added
```

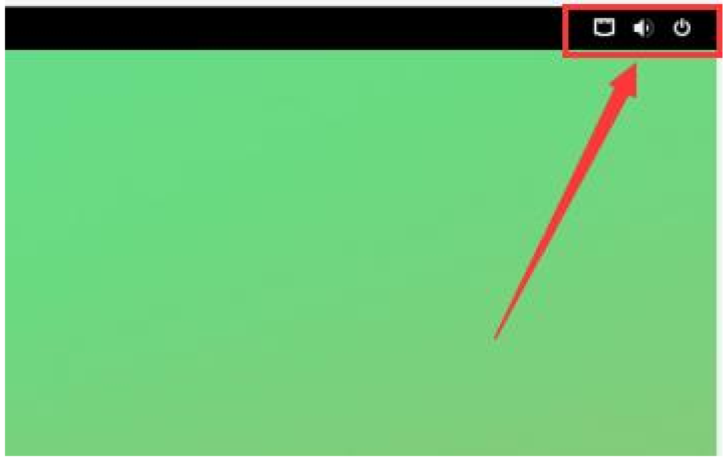
4) **Then restart the OPi OS Arch system**

5) If everything is normal after restarting the system, you can see the WiFi device node with the following command. If you can't see it, please check if there is a problem with the previous configuration.

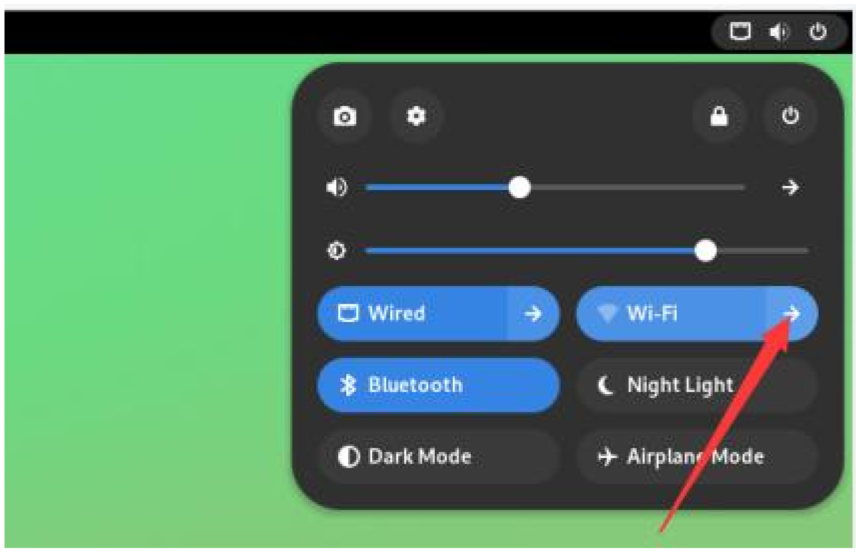
```
[orangepi@orangepi ~]$ ip addr show wlan0
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state DORMANT group default qlen 1000
    link/ether 70:f7:54:b8:b3:17 brd ff:ff:ff:ff:ff:ff
```

6) The steps connected to the WIFI are shown below：

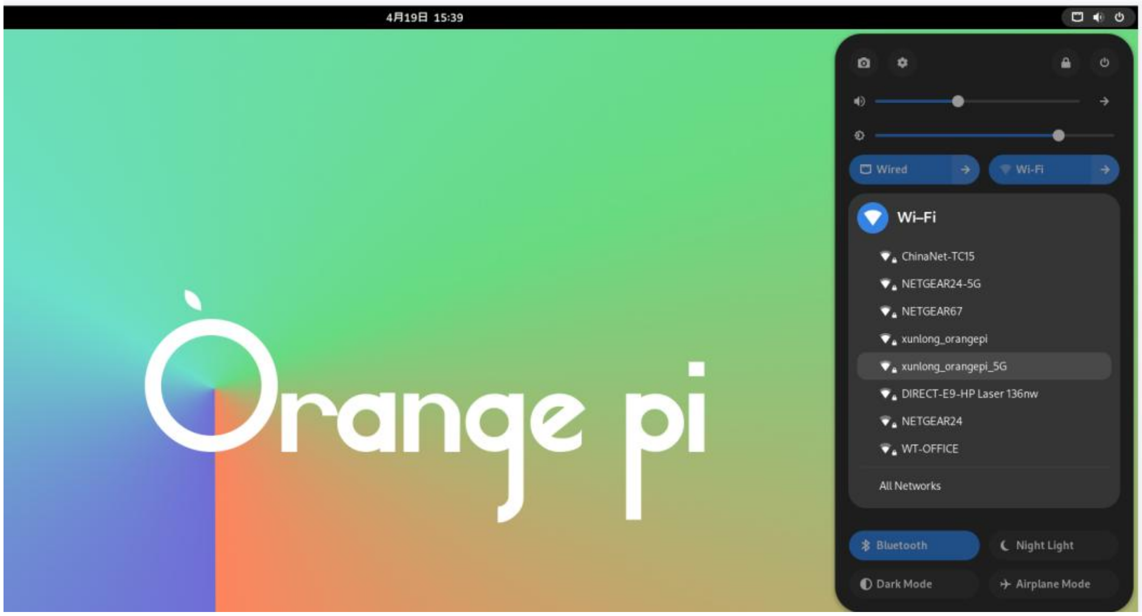
a. First click the area in the upper right corner of the desktop



b. Then select Wi-Fi

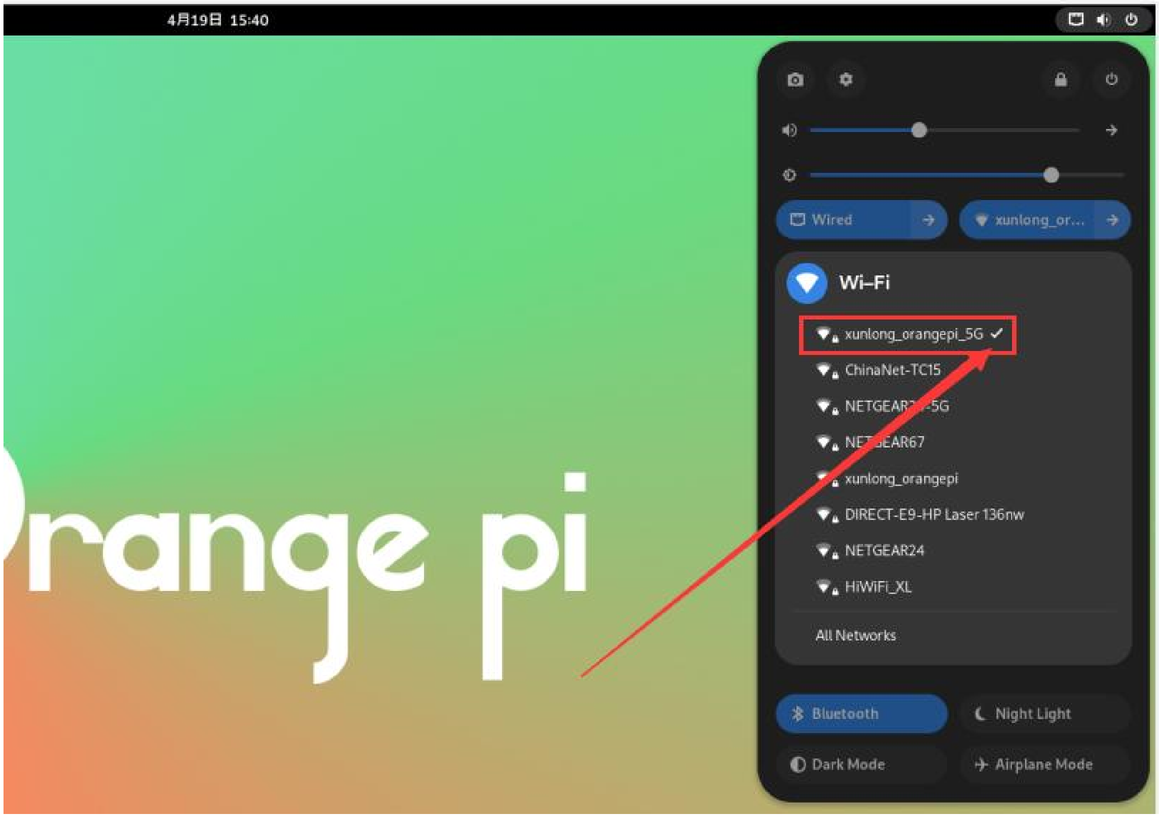


c. Then select the WIFI you want to connect



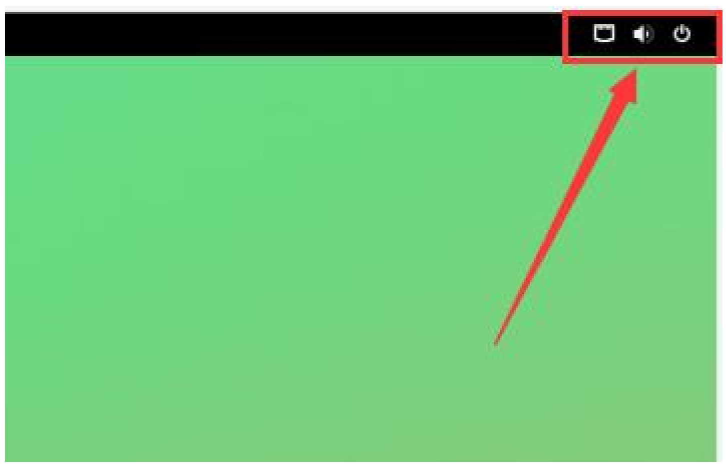
d. Then enter the password of the WIFI, and then click **Connect**

e. Then enter the following interface again to see that the WIFI is connected

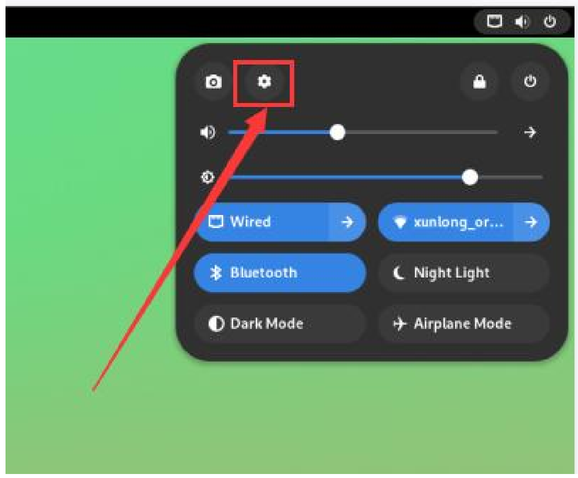


7) Example of Bluetooth usage：

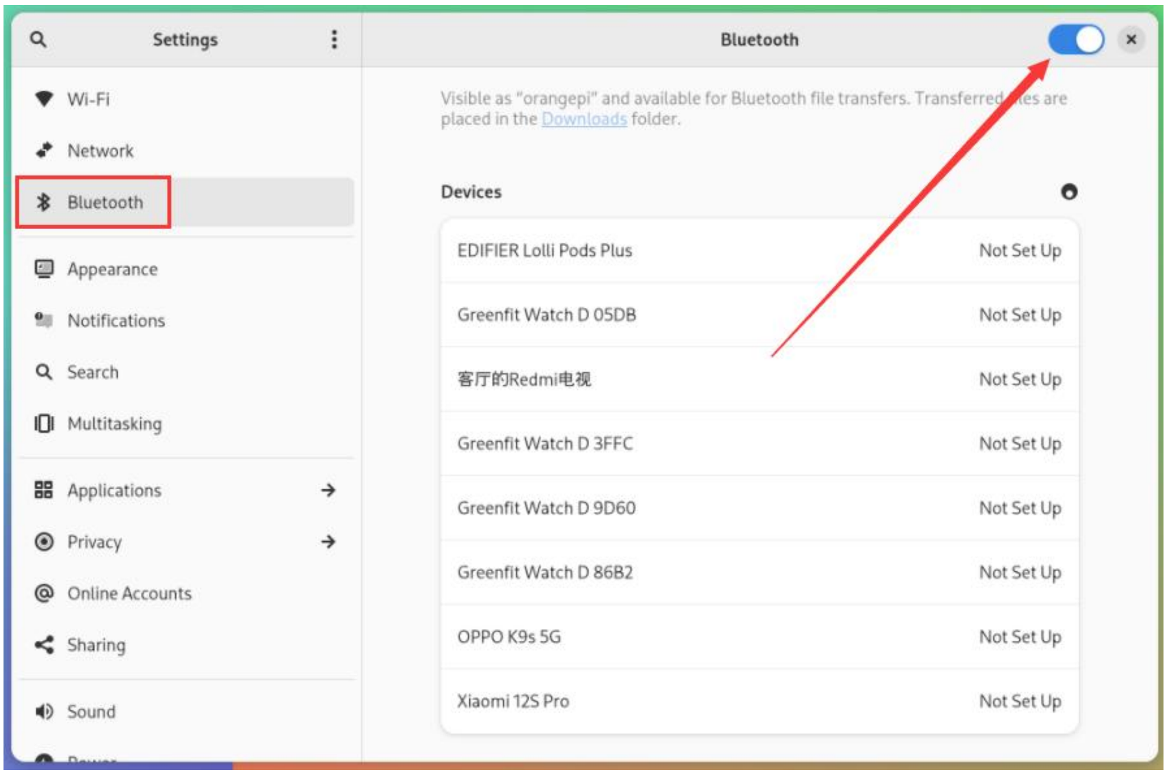
a. First click the area in the upper right corner of the desktop



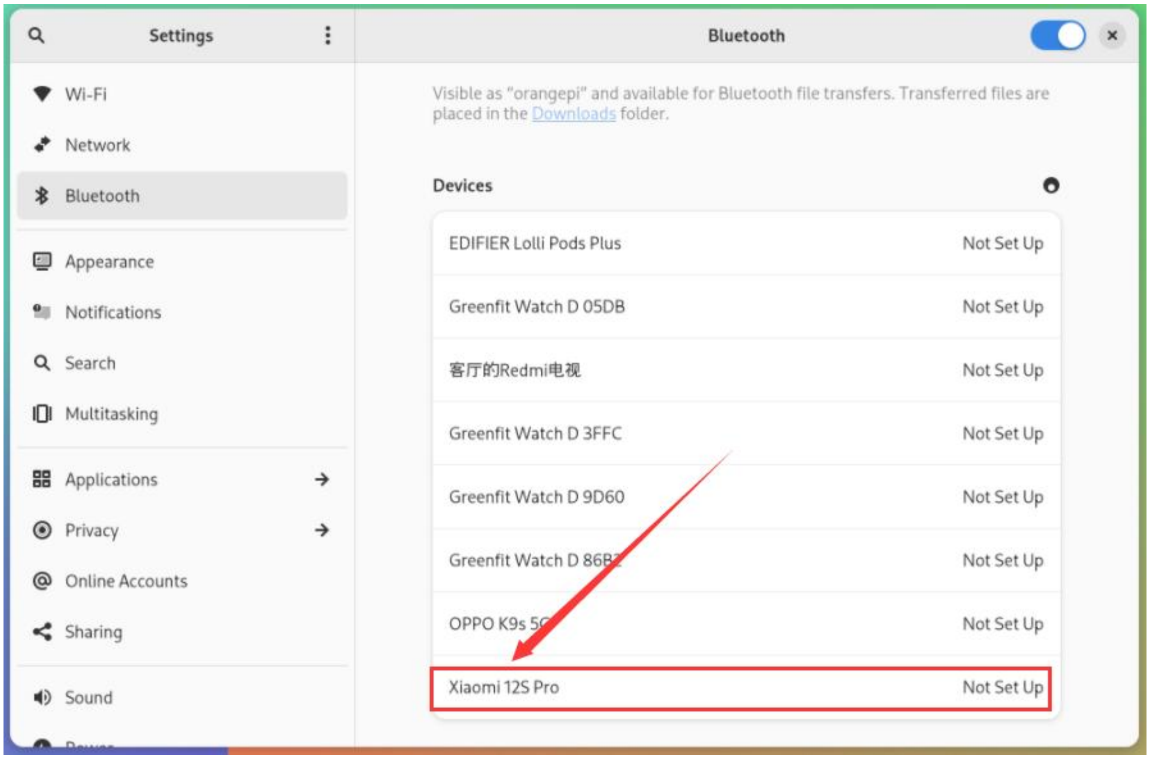
b. Then open the settings



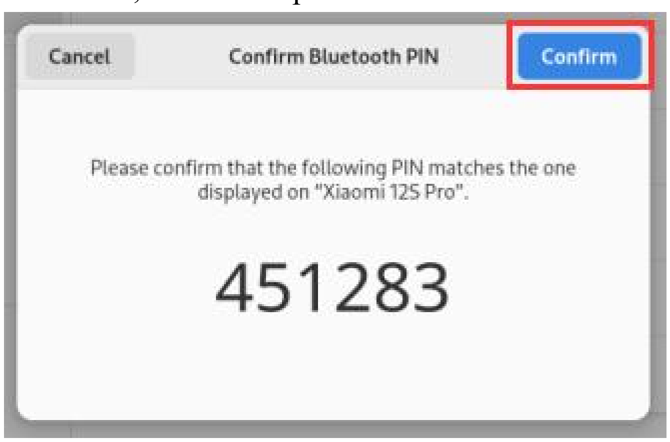
c. Then select Bluetooth in the settings and make sure that the switch button in the upper right corner of the Bluetooth has been opened



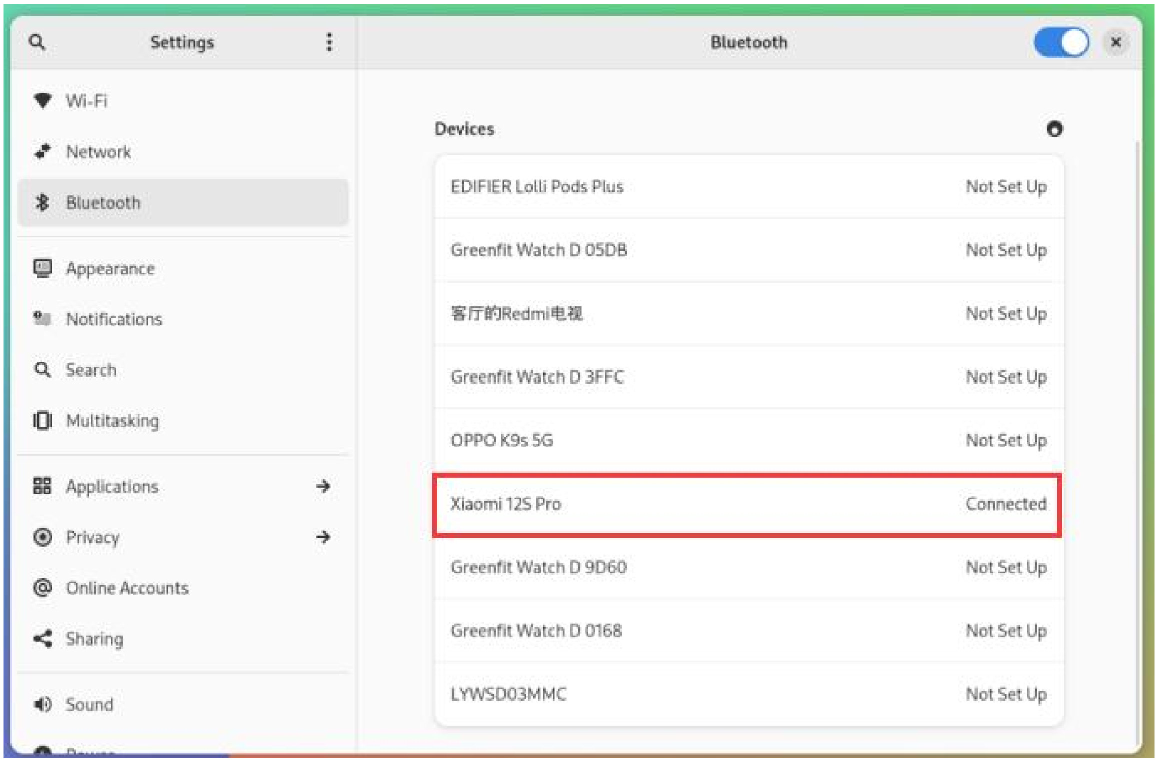
d. Then choose the Bluetooth device you want to configure the right, such as pairing with Android phones



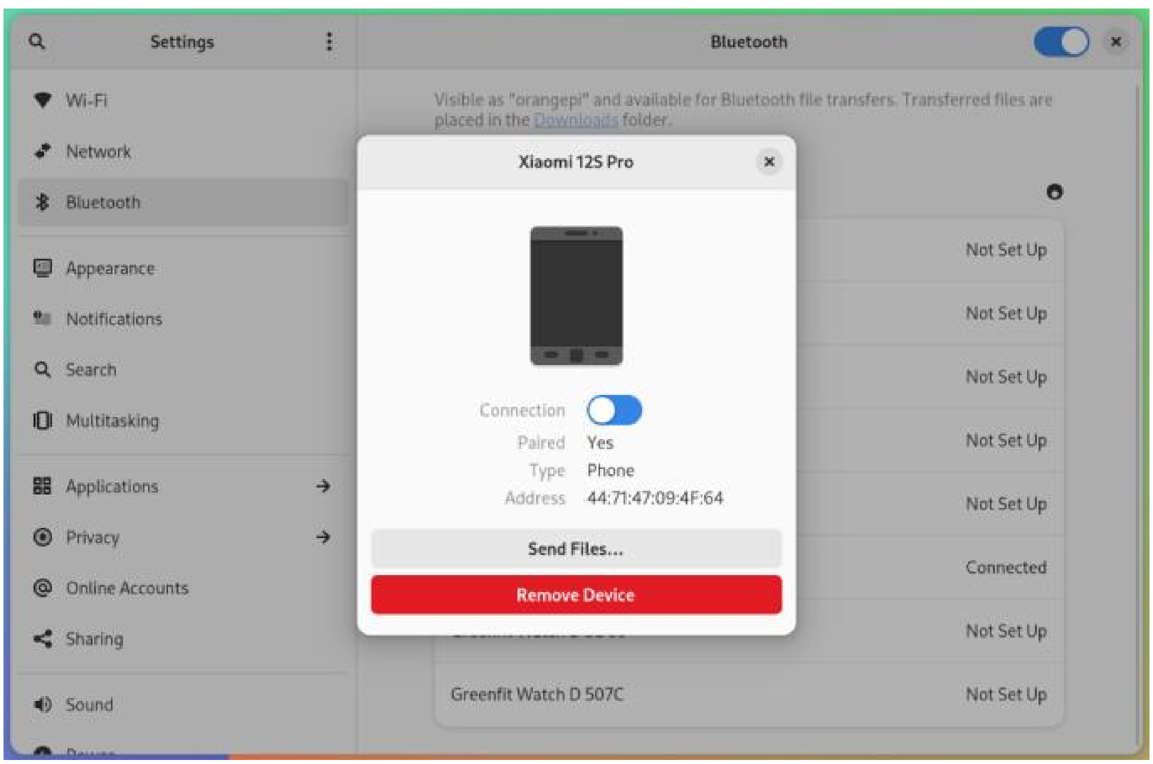
e. Then click **Confirm**, the mobile phone also needs to confirm the pairing



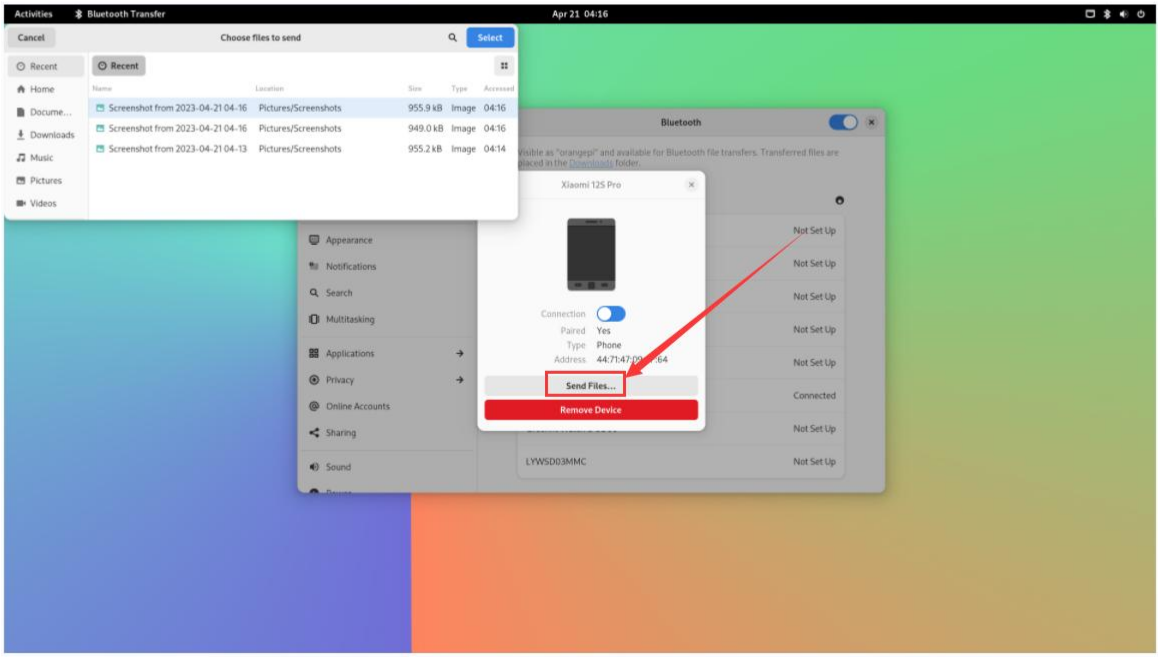
f. The display after the connection between Bluetooth and Android phones is shown below：



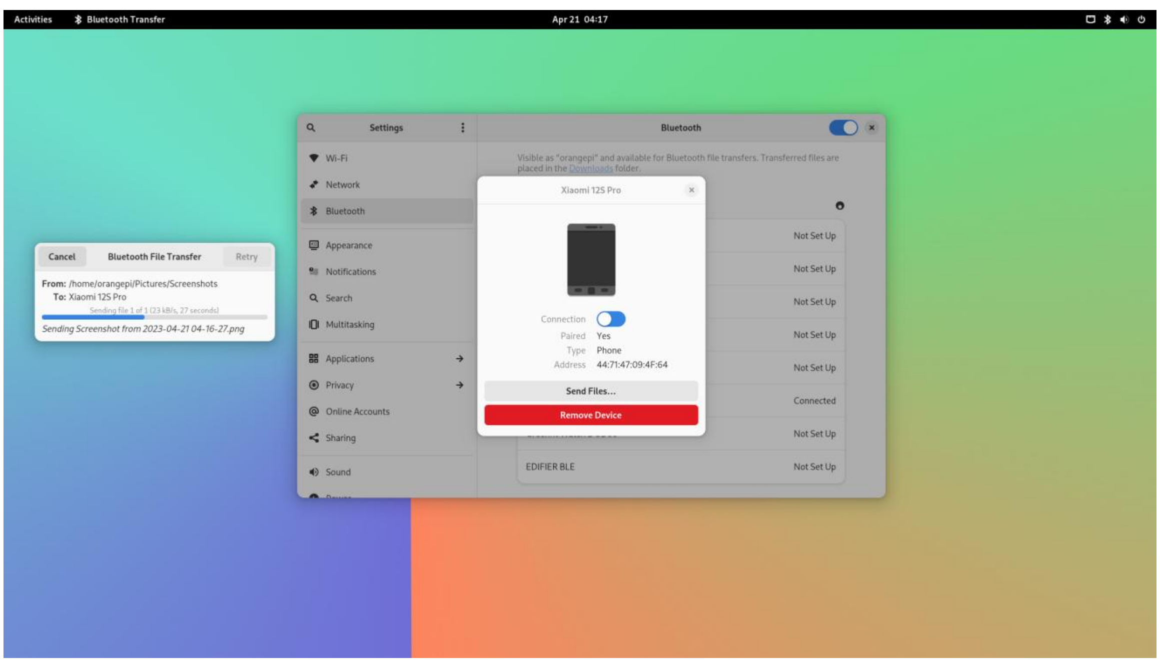
g. Then click the paired Bluetooth device to pop up the operating interface shown in the figure below



h. Click **Send Files...** at this time , you can send a file to the phone

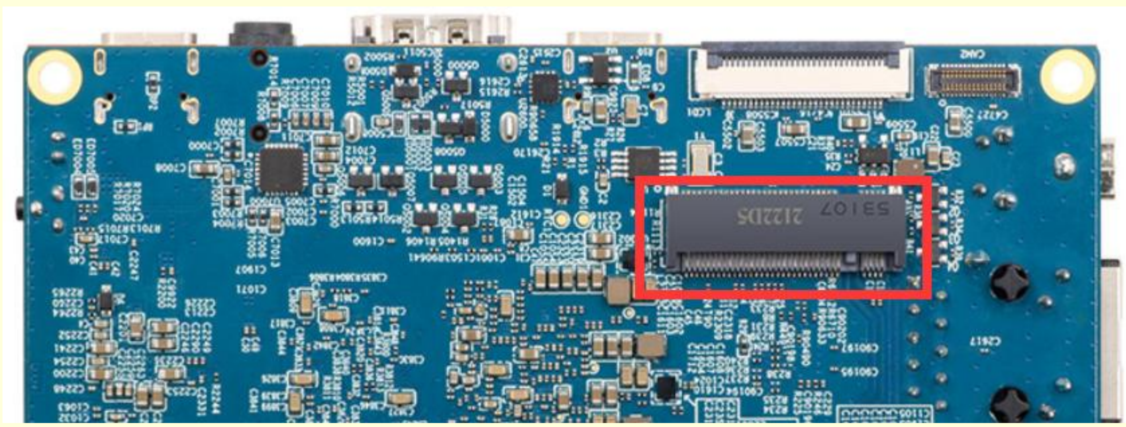


i. The schematic diagram of the Bluetooth sent pictures to the phone is shown below：



5. 3. OPi OS Arch system uses SATA SSD method

The M.2 interface shown in the figure below can use nvme ssd or sata ssd. Since the PCIe2.0 controller and SATA controller are one of the two, at the same time, only one of the configurations can be opened. OPI OS Arch Image released by Orange Pi defaults to the configuration of PCIE, so only NVME SSD can be recognized by default. If you want to use SATA SSD, you need to open the corresponding configuration。

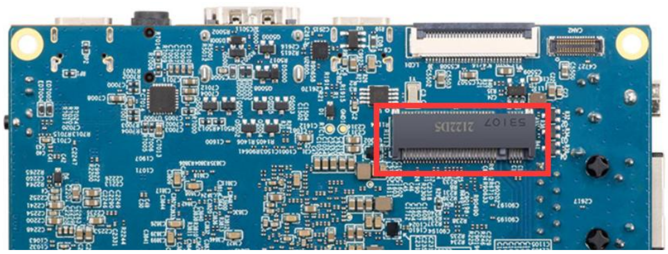


1) First of all, you need to prepare a SATA SSD solid state drive

a. M.2 2242 specification SSD is shown below

b. The SSD of the B.M.2 2280 specifications is shown below (SATA SSD of 2280 specifications can also be used, but SSD will be exceeded after inserting the development board)

2) Then insert SSD into the M.2 interface of the development board and fix it



3) There are two main usage of SATA SSD：
   a. OPi OS Arch system is in tf card, and then inserted sata ssd as an external storage device. This section mainly illustrates this usage。
   b. Burn the OPI OS Arch system to SATA SSD, and then start the OPI OS Arch system in SATA SSD. For this usage, please refer to the the **method of burning Linux image to the method of a method in Spiflash+SATA SSD**。

4) Then add the following configuration in the **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-ssd-sata.dtbo    #The configuration that needs to be added
```

5) **Then restart the OPi OS Arch system**

6) If everything is normal, you can see the sata ssd information with the **sudo fdisk -l** command after the system is restarted

```
[orangepi@orangepi ~]$ sudo fdisk -l
.......
Disk /dev/sda: 238.47 GiB, 256060514304 bytes, 500118192 sectors
```

```
Disk model: Fanxiang S201 25
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 43FFB292-340D-654C-8C30-6C64AEDAA0F4

Device       Start       End   Sectors   Size Type
/dev/sda1     2048 500117503 500115456 238.5G Linux filesystem
.......
```

5. 4. How to use 10.1 inch MIPI LCD screen

5. 4. 1. 10.1 -inch MIPI screen assembly method

1) First prepare the required accessories

a. 10.1 -inch MIPI LCD display+touch screen

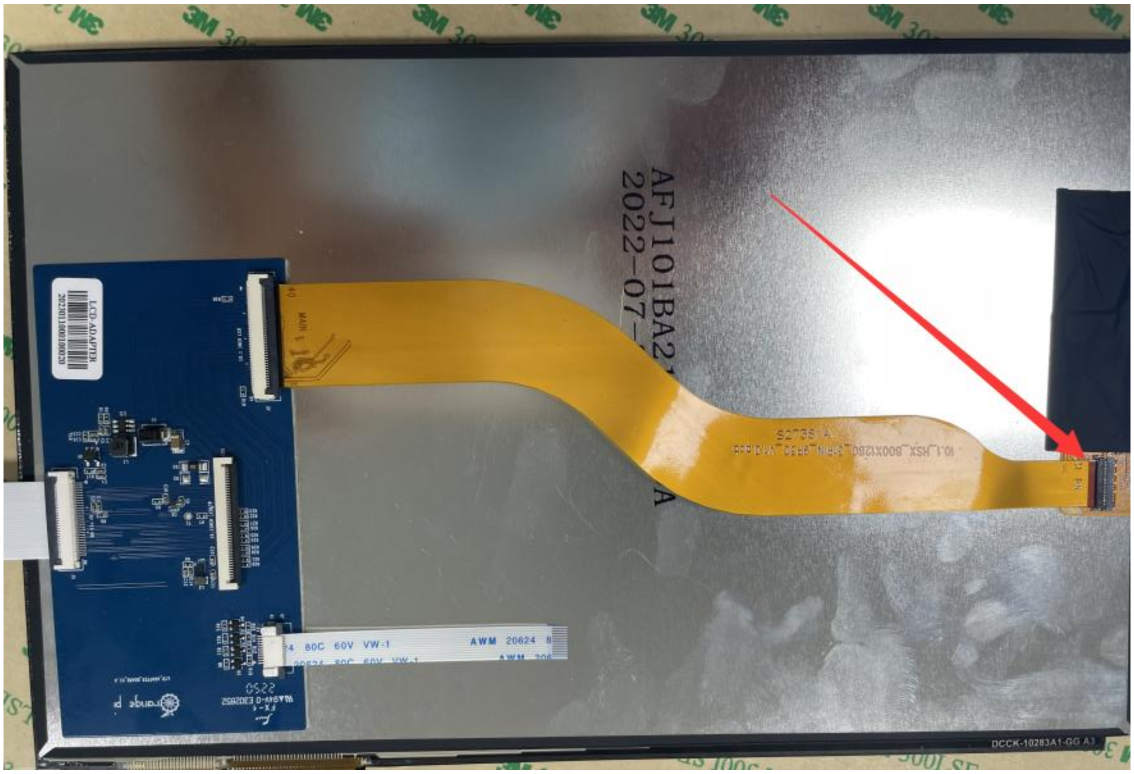
b. Screen divert plate+31pin to 40pin line

c. 30pin mipi line

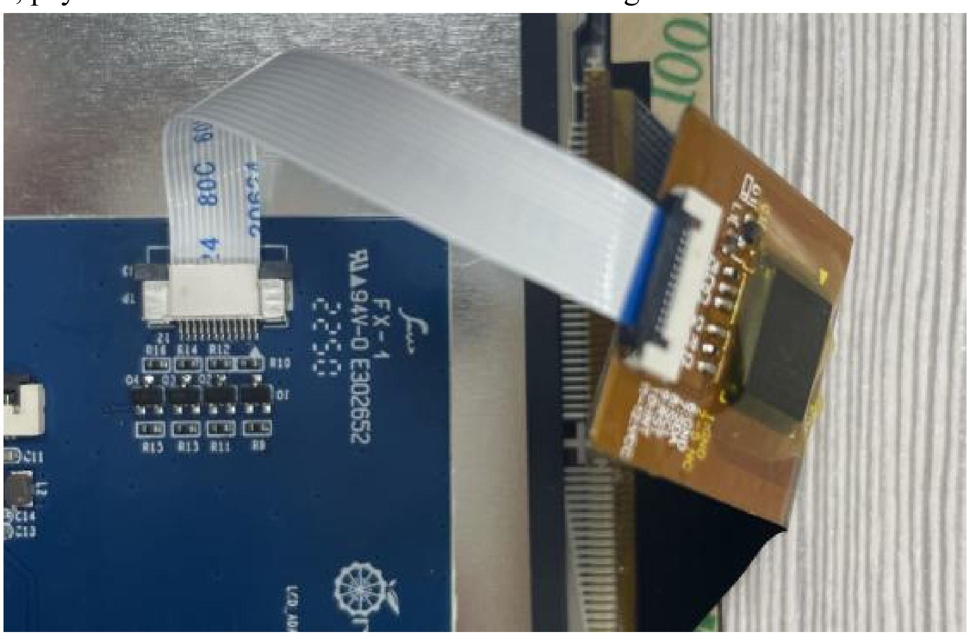
d. 12pin touch screen row line

2) According to the figure below, the 12PIN touch screen row, 31PIN to 40PIN ducts, and 30pin MIPI cables get on the screen dial board. **Pay attention to the blue insulation face of the touch screen row**, the other two lines of the line insulation faces are facing up, If you get an error, it will cause no display or unable to touch

3) Place the connected rotor connected to the puzzle on the MIPI LCD screen according to the figure below, and connect the MIPI LCD screen and the rotary board through 31Pin to 40Pin row



4) Then connect the touch screen and the rotary board through the 12Pin touch screen row line, pay attention to the orientation of the insulating side



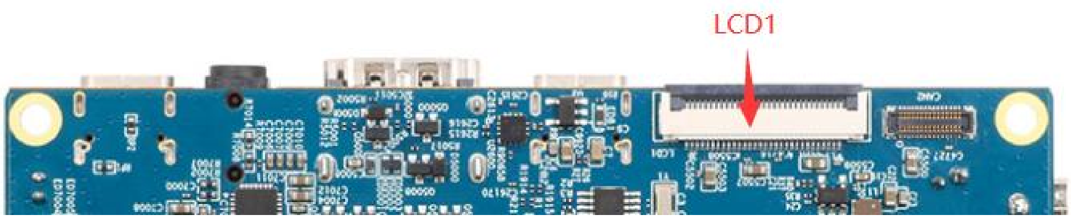
5) Finally connect to the LCD interface of the development board through the 30PIN MIPI port.

5. 4. 2. Open the 10.1 -inch MIPI LCD screen configuration method

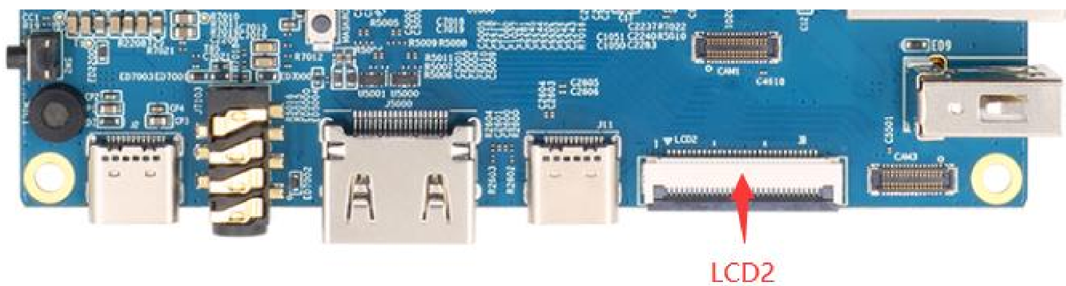
1) OPi OS Arch Image defaults to the configuration of the mipi lcd screen by default. If you need to use the MIPI LCD screen, you need to open it manually.

2) There are two interfaces of the mipi lcd screen on the development board, we define：

a. The location of the lcd1 interface is：



b. The position of the lcd2 interface is：



3) The method of opening the mipi lcd configuration is shown below：

a. If you want to open LCD1, add the following configuration in **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-lcd1.dtbo     #The configuration that needs to be added
```

b. If you want to open LCD2, add the following configuration in the **/boot/extlinux/extlinux.conf**

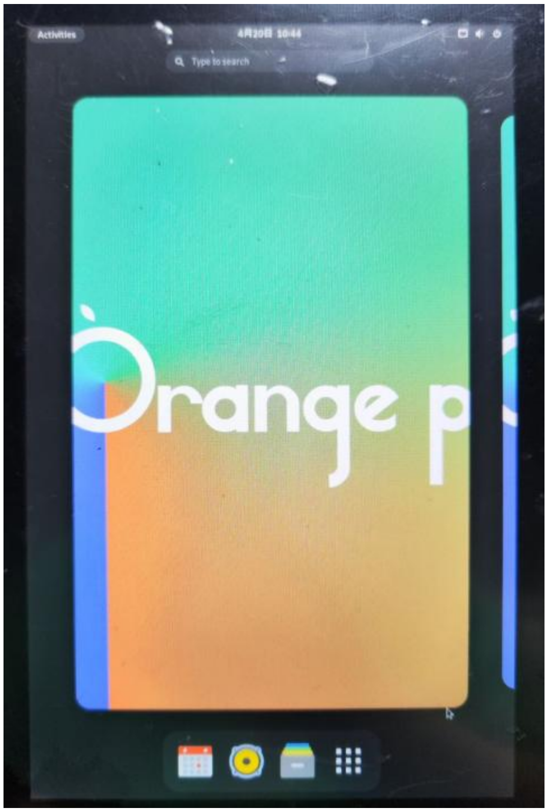
```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-lcd2.dtbo     #The configuration that needs to be added
```

c. If you want to open the LCD1 and LCD2 at the same time, add the following configuration in the **/boot/extlinux/extlinux.conf** (**the configuration of the two LCD needs to be written in one line, please do not write two lines**)

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS   /dtbs/rockchip/overlay/rk3588-lcd1.dtbo   /dtbs/rockchip/overlay/rk3588-lcd2.dtbo
```

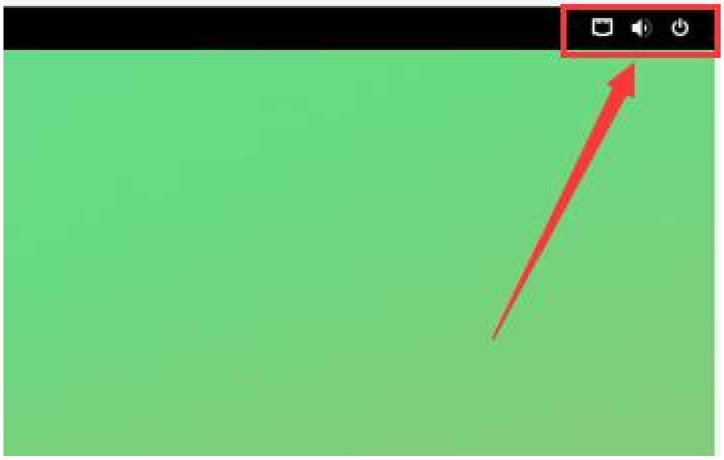
4) **Then restart the OPi OS Arch system**

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

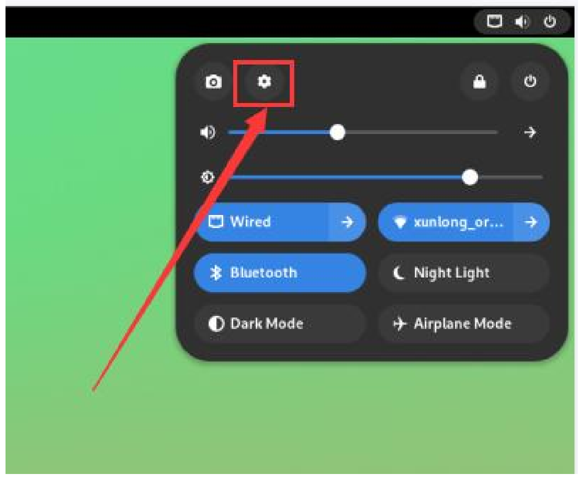


5. 4. 3. The methods to Rotating the direction of displaying and touching

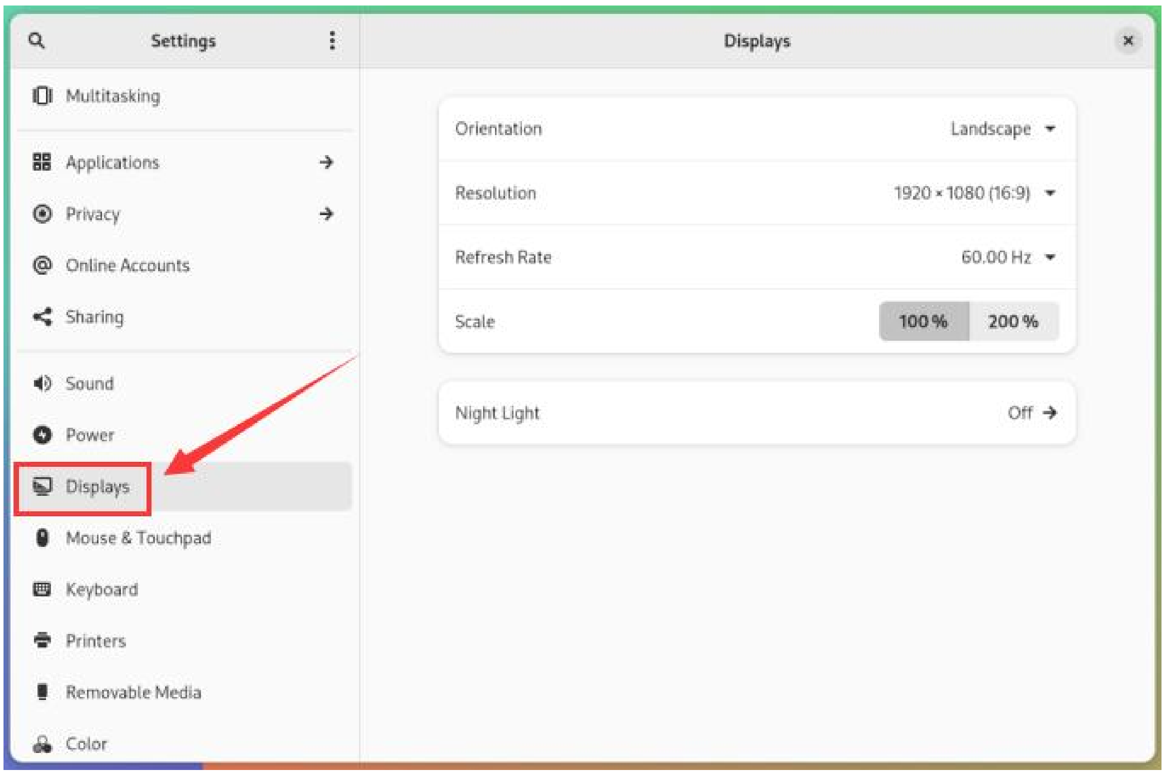
1) First click the area in the upper right corner of the desktop



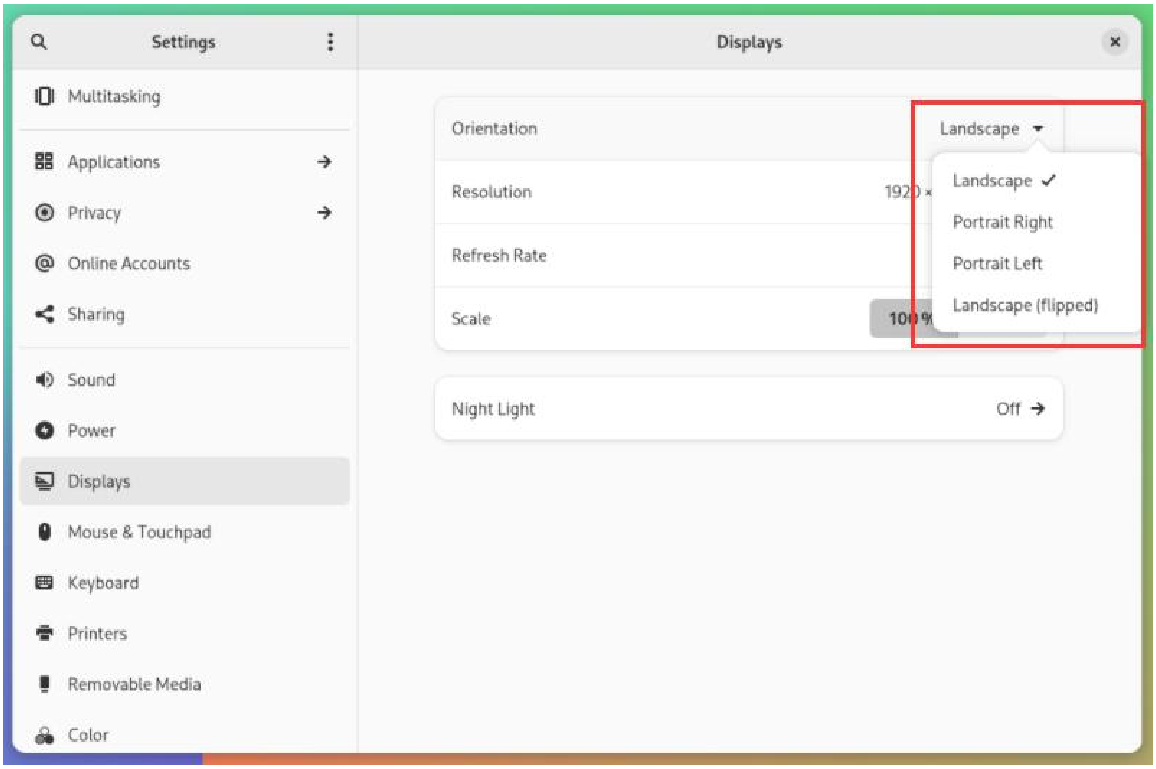
2) Then open the settings



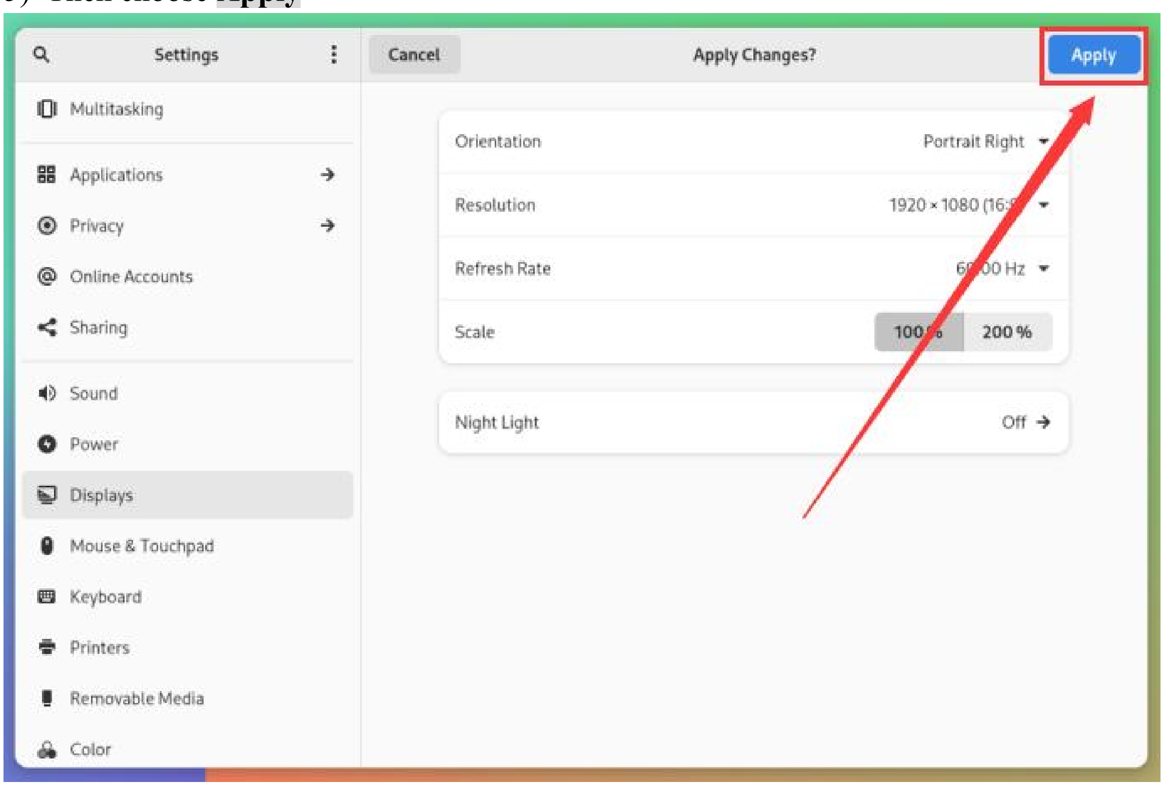
3) Then choose **Displays**



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

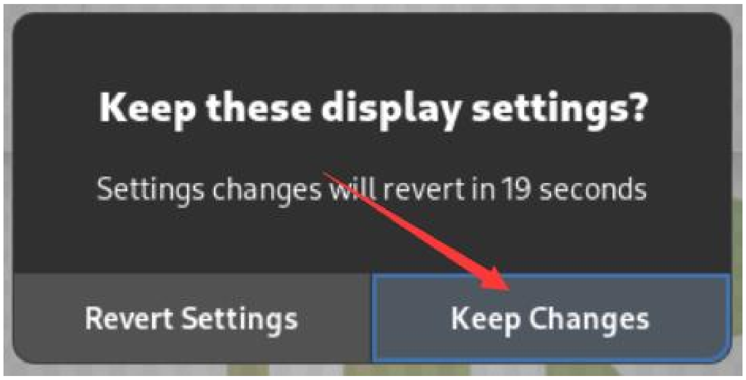


5) Then choose **Apply**



6) Then you can see that the screen has been rotated. At this time, you need to choose

Keep Changes to determine the rotation



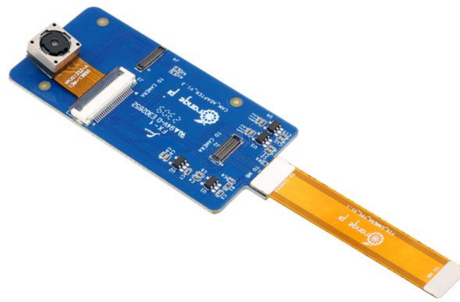
7) The display of the LCD screen after 90 degrees is shown below：

8) **The touch function of the OPi OS Arch system LCD screen will rotate with the rotation of the display direction without other settings**

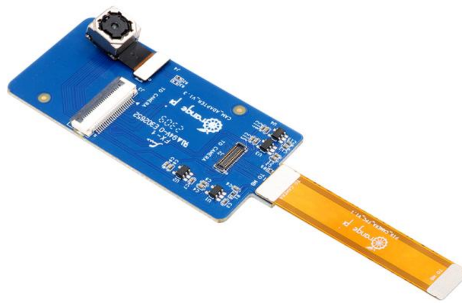
5. 5. OV13850 and OV13855 MIPI Camera testing methods

At present, the development board supports two MIPI cameras, OV13850 and OV13855, and the specific pictures are shown below：

a. OV13850 camera at13 MP MIPI interface



b. OV13855 camera at 13MP MIPI interface



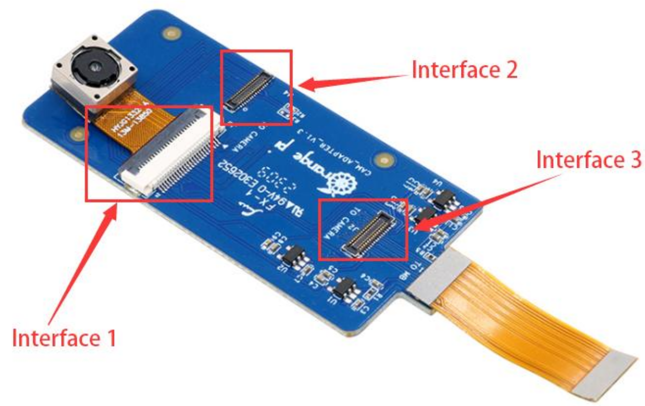
The rotary board used by OV13850 and OV13855 cameras is the same as the FPC cable, but the two cameras are different from the position on the rotary board. The FPC exhaust line is shown in the following figure. Please note that the FPC line is directed. The end is marked **TO MB** that it needs to be inserted into the camera interface of the development board. It is marked **TO CAMERA** that the end of the Camera needs to be inserted to the camera transfer board。

There are a total of 3 cameras on the camera to connect to the board, which can only be used at the same time, as shown in the figure below,

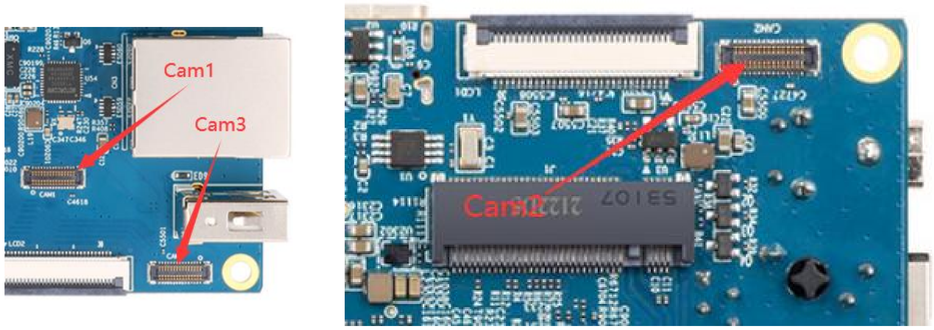
d. **1 interface OV13850 camera**

e. **2 interface OV13855 camera**

f. 3 interface is not used, just ignore it



Orange Pi 5 has a total of 3 camera interfaces on the development board. We define the positions of Cam1, Cam2 and Cam3 as shown in the figure below：



The method of the Cam1 interface inserted in the camera is shown below：

The method of the Cam2 interface inserted in the camera is shown below：

The method of the Cam3 interface inserted in the camera is shown below：

After connecting the camera to the development board, we can use the following method to test the next camera：

a. First add the following configuration in the **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-ov13850-c1.dtbo    #The configuration that needs to be added
```

The red font above is the configuration of opening the **Cam1 interface OV13850**. The configuration of other interfaces is as shown in the table below, and the corresponding dtbo configuration can be added to **FDTOVERLAYS**. If you want to add multiple configurations at the same time, separate it with a space.

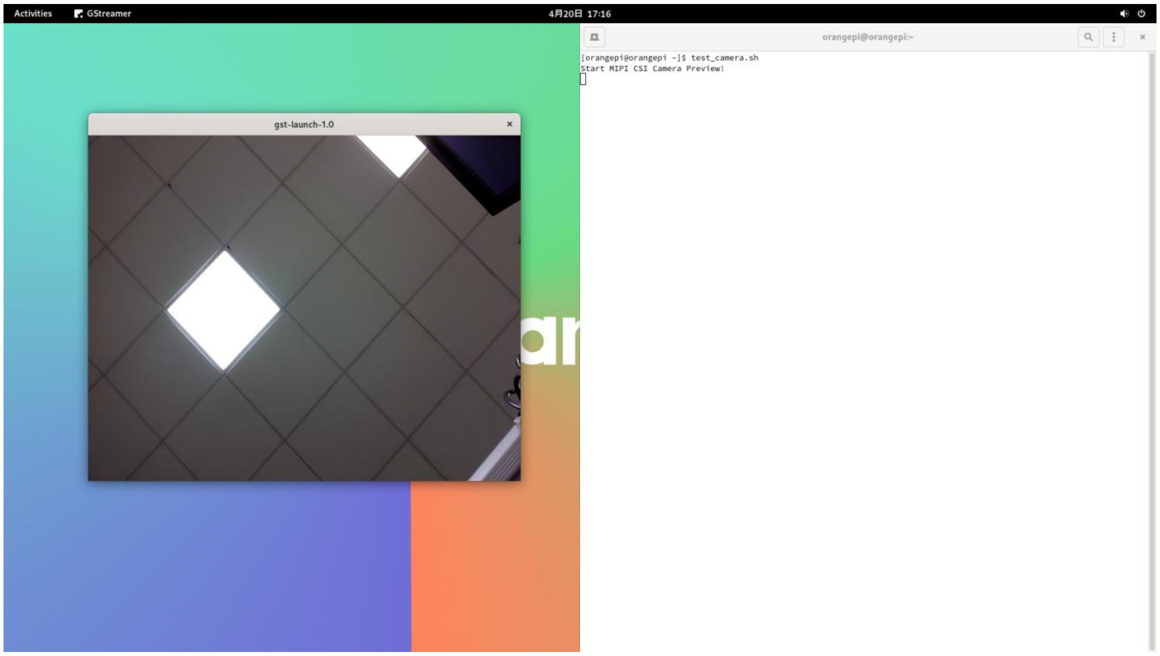
Camera	dtbo configuration
Cam1 to ov13850	/dtbs/rockchip/overlay/rk3588-ov13850-c1.dtbo
Cam2 to ov13850	/dtbs/rockchip/overlay/rk3588-ov13850-c2.dtbo
Cam3 to ov13850	/dtbs/rockchip/overlay/rk3588-ov13850-c3.dtbo
Cam1 to ov13855	/dtbs/rockchip/overlay/rk3588-ov13855-c1.dtbo
Cam2 to ov13855	/dtbs/rockchip/overlay/rk3588-ov13855-c2.dtbo
Cam3 to ov13855	/dtbs/rockchip/overlay/rk3588-ov13855-c3.dtbo

b. **Then restart the OPi OS Arch system**

c. Then open a terminal in the desktop system and run the script below

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

d. Then you can see the preview of the camera



5. 6. The method of installing wiringOP

Note that wiringOP has been pre -installed in the OPi OS Arch Image released by Orange Pi. Unless Wiringop's code is updated, it is not necessary to re -download and compile and install it.。

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

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  OPI5  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | IN     | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  OPI5  +---+--------+----------+-----+------+
[orangepi@orangepi ~]$
```

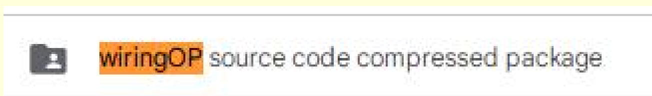
1) Download the code of wiringOP

[orangepi@orangepi ~]$ **sudo pacman -Syy git**

[orangepi@orangepi ~]$ **git clone https://github.com/orangepi-xunlong/wiringOP.git -b next**

Note that Orange Pi 5 needs to download the code of the wiringOP next branch, please don't miss the parameter of the -b next。

If you have a problem with the download code from github, you can download the source code compression package of Wiringop.tar.gz in the official tools of the Orange Pi 5 data download page。



2) Compile and install wiringOP

[orangepi@orangepi ~]$ **sudo pacman -Syy make gcc**

[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@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | IN     | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
[orangepi@orangepi ~]$
```

5. 7. 26Pin interface GPIO, I2C, UART, SPI, CAN and PWM test

Note that if you need to set fdt overlays to open multiple configurations at the same time, please use a space to write in a line like the red font configuration below.

[orangepi@orangepi ~]$ **sudo vim /boot/extlinux/extlinux.conf**

LABEL Orange Pi

LINUX /Image

FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb

FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c1-m2.dtbo /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo

5. 7. 1. 26pin GPIO port test

1) A total of 17 GPIO ports in the development board 26pin can be used. Below is No. 7 pin -corresponding to GPIO1_C6 -corresponding wPi serial number 2 -as an example how to set the height of the GPIO port

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
```

2) First set the GPIO port as the output mode, the third parameter needs to enter the serial number of the wPi corresponding to the pins

```
[orangepi@orangepi ~]$ gpio mode 2 out
```

3) Then set the GPIO port output low level. After setting, you can use the value of the voltage of the permanent meter to measure the pins. If it is 0v, it means that the low -power flat is successful

```
[orangepi@orangepi ~]$ gpio write 2 0
```

Using GPIO Readall, you can see the value of the No. 7 pin (v) to 0

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |    OUT | 0 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
```

4) Then set the GPIO port output high level. After setting, you can use the voltage of the voltage of the permanent meter to measure the voltage. If it is 3.3v it means that the high -electricity level is successful

```
[orangepi@orangepi ~]$ gpio write 2 1
```

Using gpio readall, you can see the value of No. 7 pin (v) into 1

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |    OUT | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
```

5) The setting method of other pins is similar. Just modify the serial number of the wpi sequence number as the corresponding serial number corresponding to the pin

5. 7. 2. 26pin GPIO Port -down pull -down resistance setting method

Note that Orange Pi 5 only has the following 4 GPIO pins, which can normally

set the pull -down resistance function. Other GPIO pins are pulled up because there are 3.3V on the outside, so the drop -down pull is invalid。

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  OPI5  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | IN     | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  OPI5  +---+--------+----------+-----+------+
```

1) Below is No. 11 -corresponding to GPIO4_B2 -Corresponding wPi serial number 5 —— to demonstrate how to set up and down pull -down resistance of the GPIO port

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  OPI5  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | IN     | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
```

2) First of all, you need to set the GPIO port as the input mode. The third parameter needs to enter the serial number of the wPi corresponding to the pins

[orangepi@orangepi ~]$ **gpio mode 5 in**

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

[orangepi@orangepi ~]$ **gpio mode 5 up**

4) Then enter the command below to read the level of the GPIO port. If the level is 1, it

means that the drawing mode settings are successful

```
[orangepi@orangepi ~]$ gpio read 5
1
```

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

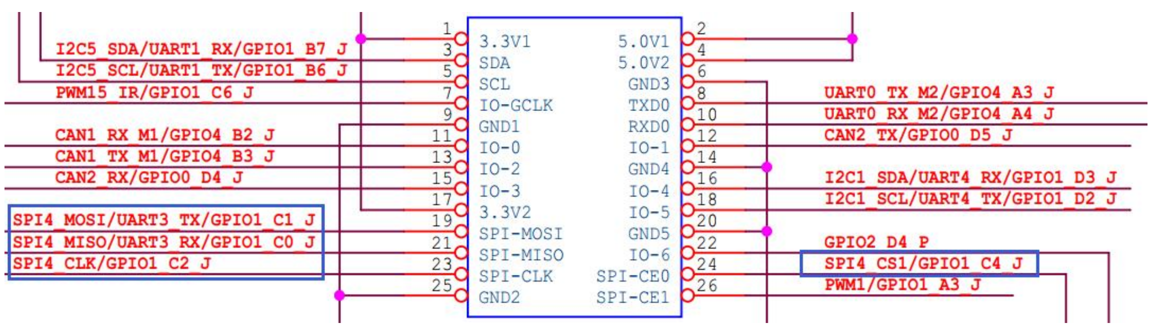
```
[orangepi@orangepi ~]$ gpio mode 5 down
```

6) Then enter the following command to read the level of the GPIO port. If the level is 0, it means that the drop -down mode is set successfully.

```
[orangepi@orangepi ~]$ gpio read 5
0
```

5. 7. 3. 26pin SPI test

1) From the schematic diagram of the 26pin interface, the SPI available for Orange Pi 5 is spi4



In the OPi OS Arch system, the spi4 in 26pin is closed by default. You need to open it manually to use it。

Add the configuration of the red font part below to /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open the spi4

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-spi4-m0-cs1-spidev.dtbo
```

2) First check whether there is a **spidev4.1** device node in the OPi OS Arch system. If it

exists, it means that the SPI4 has been set and you can use it directly

```
[orangepi@orangepi ~]$ ls /dev/spidev4.1
/dev/spidev4.1
```

3) Do not shorten SPI4's mosi and miso pins, and run the output result of the spidev_test as shown below. You can see that the data of TX and RX is inconsistent

```
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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 FF  | ................................
```

4) Then the mosi (No. 19 in 26Pin interface) and miso (No. 21 in the 26Pin interface) of the SPI4 and MISO (pin 21 in the 26pin interface) run the output of spidev_test as follows. You can see the sending and receiving receiving The data is the same

```
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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 FF F0 0D  | ......@....▒..................▒.
```

5. 7. 4. 26pin I2C test

1) From the table below, the i2c available for Orange Pi 5 is i2c1, i2c3, and i2c5. There are three groups of i2c bus

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

It can be seen from the above table that i2c1 can be exported from the 12 and 15 pins of 26Pin (I2C1_M2), or it can also be exported from the 16 and 18 pins of 26PIN (I2C1_M4). Please follow your own needs Choose a group. Please don't think that this is two different groups of i2c bus。

In the OPi OS Arch system, the i2c in 26pin is closed by default. It needs to be opened manually to use it。

Add the configuration of the red font part below to the /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open the i2c1, i2c3, and i2c5 at the same time. If you only need to open one, then fill in one

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS  /dtbs/rockchip/overlay/rk3588-i2c1-m2.dtbo
/dtbs/rockchip/overlay/rk3588-i2c3-m0.dtbo
/dtbs/rockchip/overlay/rk3588-i2c5-m3.dtbo
```

The red font configuration above needs to be written in one line, and different configurations need to be separated by spaces。

2) After starting the OPi OS Arch system, first confirm that the i2c device node exists under/dev

```
[orangepi@orangepi ~]$ ls /dev/i2c-*
/dev/i2c-0   /dev/i2c-10   /dev/i2c-3   /dev/i2c-6   /dev/i2c-9
/dev/i2c-1   /dev/i2c-2    /dev/i2c-5   /dev/i2c-7
```

3) Then connect a i2c device on the i2c pin of the 26pin connector

	i2c1-m2	i2c1-m4	i2c3-m0	i2c5-m3
Sda Pin	Corresponding to No. 12 Pin	Corresponding to No. 16 Pin	Corresponding to No. 21 Pin	Corresponding to No. 3 Pin
Sck Pin	Corresponding to No. 15 Pin	Corresponding to No. 18 Pin	Corresponding to No. 19 Pin	Corresponding to No. 5 Pin
3.3v Pin	Corresponding to No. 1 Pin	Corresponding to No. 1 Pin	Corresponding to No. 1 Pin	Corresponding to No. 1 Pin
5v Pin	Corresponding to No. 2 Pin	Corresponding to No. 2 Pin	Corresponding to No. 2 Pin	Corresponding to No. 2 Pin
Gnd Pin	Corresponding to No. 6 Pin	Corresponding to No. 6 Pin	Corresponding to No. 6 Pin	Corresponding to No. 6 Pin

Points 3V and 5V pins are generally only connected to one. Please select 3.3v pins or 5v pin according to the specific i2c device。

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

```
[orangepi@orangepi ~]$ sudo pacman -Syy i2c-tools
[orangepi@orangepi ~]$ sudo i2cdetect -y 1          #i2c1 command
[orangepi@orangepi ~]$ sudo i2cdetect -y 3          #i2c3 command
[orangepi@orangepi ~]$ sudo i2cdetect -y 5          #i2c5 command
```

5. 7. 5. 26pin's UART test

1) From the table below, the uart available for Orange Pi 5 is uart0, uart1, uart3, and uart4. There are four sets of uart bus

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

In the OPi OS Arch system, the uart in 26Pin is closed by default. It needs to be opened manually to use。

Add the configuration of the red font part below to the /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open UART0, UART1, UART3, and

UART4 at the same time。

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo /dtbs/rockchip/overlay/rk3588-uart1-m1.dtbo /dtbs/rockchip/overlay/rk3588-uart3-m0.dtbo /dtbs/rockchip/overlay/rk3588-uart4-m0.dtbo

The red font configuration above needs to be written in one line, and different configurations need to be separated by spaces。

2) After entering the linux system, first confirm whether there is a device node corresponding to UART under/dev

```
[orangepi@orangepi ~]$ ls /dev/ttyS*
/dev/ttyS0   /dev/ttyS1   /dev/ttyS3   /dev/ttyS4   /dev/ttyS9
```

3) Then start testing the UART interface, first use the RX and TX of the UART interface to be tested by DuPont

	uart0	uart1	uart3	uart4
Tx Pin	Corresponding to No. 8 Pin	Corresponding to No. 5 Pin	Corresponding to No.19 Pin	Corresponding to No. 18 Pin
Rx Pin	Corresponding to No. 10 Pin	Corresponding to No. 3 Pin	Corresponding to No. 21 Pin	Corresponding to No. 16 Pin

4) Use **gpio serial** command to test the loop function of the serial port as shown below. If you can see the printing below, it means that the serial communication is normal

a. Test UART0

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS0
[sudo] password for orangepi:   #Enter the password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

b. Test UART1

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS1
[sudo] password for orangepi:   #Enter the password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

c. Test UART3

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS3
[sudo] password for orangepi:   #Enter the password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

d. Test UART4

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS4
[sudo] password for orangepi:   #Enter the password here

Out:    0:   ->    0
Out:    1:   ->    1
```

```
Out:     2:   ->     2
Out:     3:   ->     3
Out:     4:   ->     4
Out:     5:   ->     5^C
```

5. 7. 6. PWM test method

1) From the table below, the PWM available for Orange Pi 5 includes PWM0, PWM1, PWM3, 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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

You can see from the above table:

pwm1 can be introduced from No. 16 in 26pin (pwm1_m1), or it can be guided from the 26th foot of 26Pin (pwm1_m2)

pwm3 can be introduced from No. 15 in 26Pin (pwm3_m0), or you can also guide from the 23rd foot of 26Pin (pwm3_m2)

Please choose the corresponding pins according to your needs. Please don't think that this is two different pwm bus。

In the OPi OS Arch system, the PWM in 26PIN is closed by default. It needs to be opened manually to use it。

Add the configuration of the red font part below to the /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open PWM0, PWM13, PWM14 and PWM15 at the same time. If you only need to open one, then fill in one。

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS    /dtbs/rockchip/overlay/rk3588-pwm0-m1.dtbo
/dtbs/rockchip/overlay/rk3588-pwm13-m2.dtbo
/dtbs/rockchip/overlay/rk3588-pwm14-m1.dtbo
```

/dtbs/rockchip/overlay/rk3588-pwm15-m2.dtbo

The red font configuration above needs to be written in one line, and different configurations need to be separated by spaces。

2) After opening a PWM, there will be an additional pwmchipX (X as a specific number) in**/sys/class/pwm/**. For example, after opening pwm15, View pwmchipX from**/sys/class/pwm/**down will change from two to three

```
[orangepi@orangepi ~]$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

3) Which pwmchip corresponds to PWM15 above? Let's first check the output of **ls /sys/class/pwm/ -l** command. As shown below

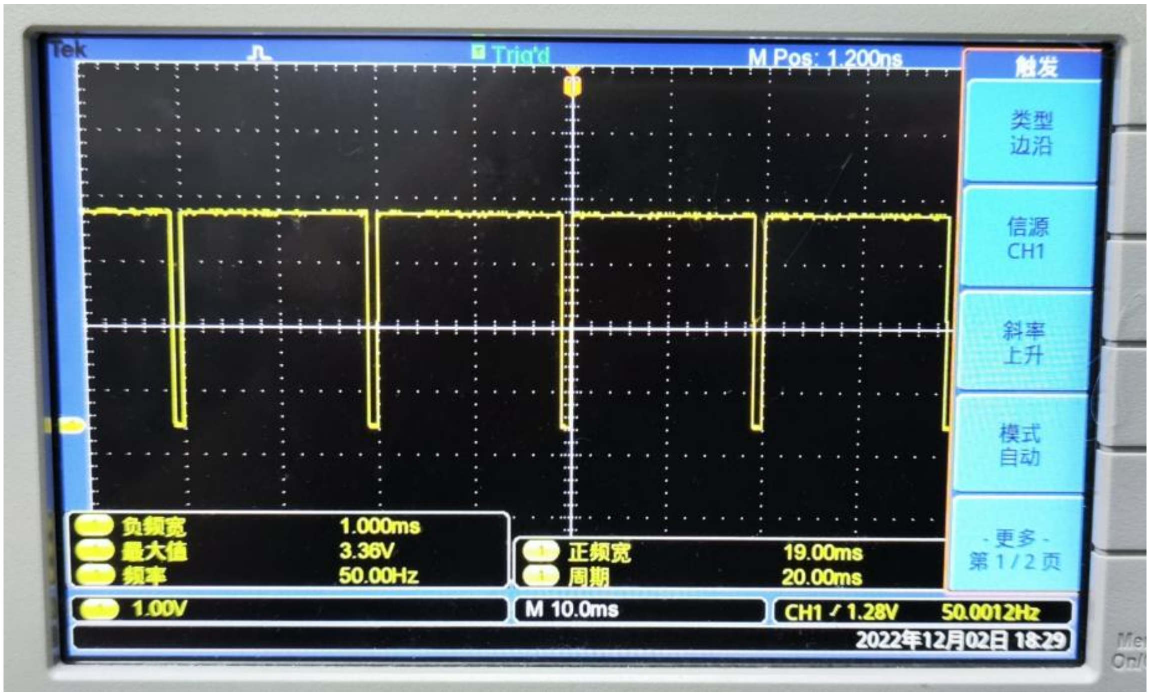
```
[orangepi@orangepi ~]$ ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Apr 20 07:33 pwmchip0 -> ../../devices/platform/fd8b0020.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Apr 20 07:33 pwmchip1 -> ../../devices/platform/febd0020.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Apr 20 07:33 pwmchip2 -> ../../devices/platform/febf0030.pwm/pwm/pwmchip2
[orangepi@orangepi ~]$
```

4) Then it can be seen from the table below that the base address of the PWM15 register is Febf0030. Then look at the output of **ls /sys/class/pwm/ -l** command, you can see that the FEBF0030.PWM is connected to the link in PWMCHIP2, so the PWM15 corresponds to PWMCHIP2 as PWMCHIP2

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

5) Then use the following command to allow the PWM15 to output a 50Hz square wave (please switch to the root user first, and then execute the following command)

```
[root@orangepi orangepi]# echo 0 > /sys/class/pwm/pwmchip2/export
[root@orangepi orangepi]# echo 20000000 > /sys/class/pwm/pwmchip2/pwm0/period
[root@orangepi orangepi]# echo 1000000 > /sys/class/pwm/pwmchip2/pwm0/duty_cycle
[root@orangepi orangepi]# echo 1 > /sys/class/pwm/pwmchip2/pwm0/enable
```



6) The other PWM testing methods are similar to the PWM15 test method。

5. 7. 7. CAN test method

1) As can be seen from the table below, the CAN bus available for Orange Pi 5 is CAN1 and CAN2. There are two CAN bus

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			
PWM13_M2（febf0010）	UART1_RX_M1（feb40000）	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2（febf0030）	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2（fd890000）		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1（febf0020）	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
PWM3_IR_M0（fd8b0030）	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GPIO1_D3	UART4_RX_M0（feb70000）	I2C1_SDA_M4	PWM1_M1（fd8b0010）
			3.3V		17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1（fd8b0000）
I2C3_SCL_M0	UART3_TX_M0（feb60000）	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22	92	GPIO2_D4			
	PWM3_IR_M2（fd8b0030）	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	GPIO1_C4	SPI4_CS1_M0		
			GND		25	26	35	GPIO1_A3	PWM1_M2（fd8b0010）		

In the OPi OS Arch system, the Can in 26Pin is closed by default. You need to open it manually to use it。

Add the configuration of the red font part below to the /boot/extlinux/extlinux.conf, and then restart the OPI OS Arch system to open CAN1 and Can2 at the same time. If you only need to open one, then fill in one。

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image

FDT /dtbs/rockchip/rk3588s-orangepi-5.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-can1-m1.dtbo /dtbs/rockchip/overlay/rk3588-can2-m1.dtbo

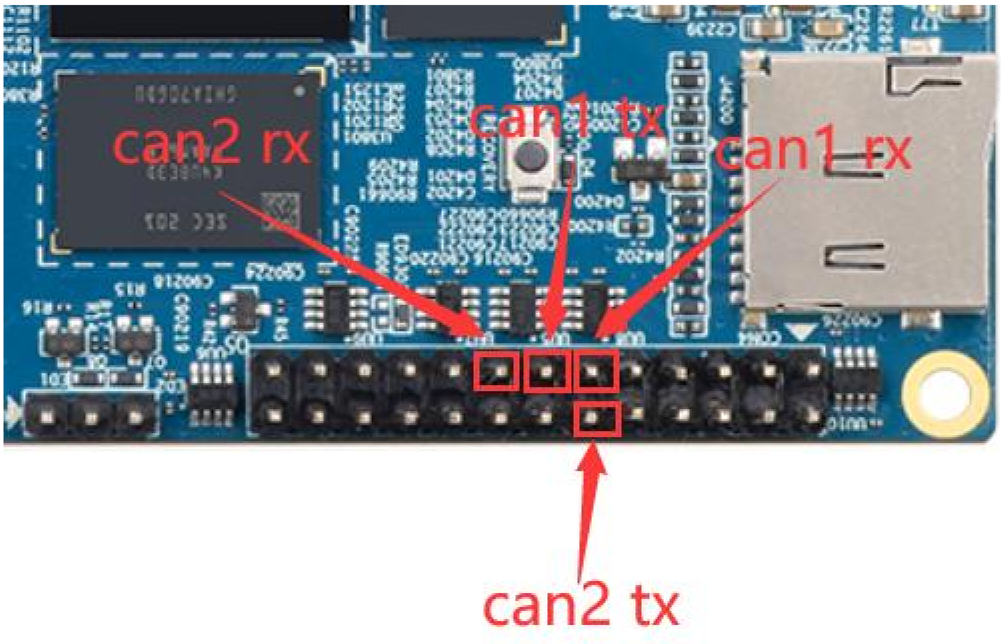
The red font configuration above needs to be written in one line, and different configurations need to be separated by spaces。

2) After entering the OPi OS Arch system, use the **sudo ifconfig -a** command. If you can see CAN's device nodes, it means that CAN has been opened correctly

```
[orangepi@orangepi ~]$ sudo pacman -Syy net-tools
[orangepi@orangepi ~]$ sudo ifconfig -a
can0: flags=128<NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 91

can1: flags=128<NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 92
```

3) The pins corresponding to CAN1 and CAN2 are

	CAN1	CAN2
TX Pin	Corresponding number 13 Pin	Corresponding number 12 Pin
RX Pin	Corresponding number 11 Pin	Corresponding number 15 Pin



4) Use Canalyst-II analyzer to test the CAN receiving message. For reference to **using Canalyst-II analyzer test and receiving message.**

6. Linux SDK—orangepi-build instructions

6. 1. Compile system requirements

We can compile the Linux Image of the development board in the x64 computer, or compile the linux image of the development board in the Ubuntu22.04 system of the development board. Please choose one of them according to your preferences。

If you use orangepi-build to compile the Linux Image in the Ubuntu22.04 system of the development board to compile the Linux Image, please do the heat dissipation (especially when the SSD startup). If the heat dissipation is not done well, it is prone to error in the file system running.

6. 1. 1. Use the development board Ubuntu22.04 system to compile

1) Linux SDK, **orangepi-build**, supports the upper operation of the development board's**Ubuntu 22.04** (other systems have not been tested), so before downloading orangepi-build, first make sure that the Ubuntu version installed on the development

board is Ubuntu 22.04. The command of the Ubuntu version installed on the development board is shown below. If the Release field is not **22.04**, it means that the Ubuntu version currently used does not meet the requirements. Please replace the system before performing the following operations.

```
orangepi@orangepi:~$ lsb_release -a
No LSB modules are available.
Distributor ID:    Ubuntu
Description:       Ubuntu 22.04.1 LTS
Release:           22.04
Codename:          jammy
```

2) Because the source code such as kernel and U-Boot is stored on GitHub, So when compiling Images, please make sure that the development board can download the code normally from github. This is very important

6.1.2. Use X64's Ubuntu22.04 computer to compile

1) Linux SDK, **orangepi-build**, supports running on a computer with **Ubuntu 22.04**, so before downloading Orange-Build, first make sure that the Ubuntu version of your computer installed is Ubuntu 22.04. Check the command of the Ubuntu version installed by the computer as shown below. If the release field is not **22.04**, it means that the currently used Ubuntu version does not meet the requirements. Please replace the system before performing the following operations

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID:    Ubuntu
Description:       Ubuntu 22.04 LTS
Release:           22.04
Codename:          jammy
```

2) If the computer is installed with a Windows system and a computer with Ubuntu 22.04 is not installed, you can consider using **VirtualBox** or **VMware** to install a Ubuntu 22.04 virtual machine in the Windows system. But please note that Orange-Build is compiled on the WSL virtual machine. Because Orangepi-BUILD has not been tested in the WSL virtual machine, it is impossible to ensure that it can be used in WSL normally

3) Ubuntu 22.04 **amd64** version installation Image download address is：

https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04.3-desktop-amd64.iso

or
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.3-desktop-amd64.iso

4) After installing Ubuntu 22.04 in the computer or virtual machine, please set up the software source of Ubuntu 22.04 as a Tsinghua source first, otherwise it is easy to make mistakes due to network reasons when installing the software

a. The method of replacing Tsinghua source refers to the instructions of this webpage

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 /etc/apt/sources.list.bak
test@test:~$ sudo vim /etc/apt/sources.list
# The source code image is noted by default to improve the apt update speed. If necessary, you can cancel the annotation by yourself
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 sources, it is 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 replacement, you need to update the package information and make sure that there is no error

```
test@test:~$ sudo apt update
```

e. In addition, because the source code such as kernel and U-Boot is stored on GitHub, please make sure that the computer can download the code normally when compiling image. This is very important。

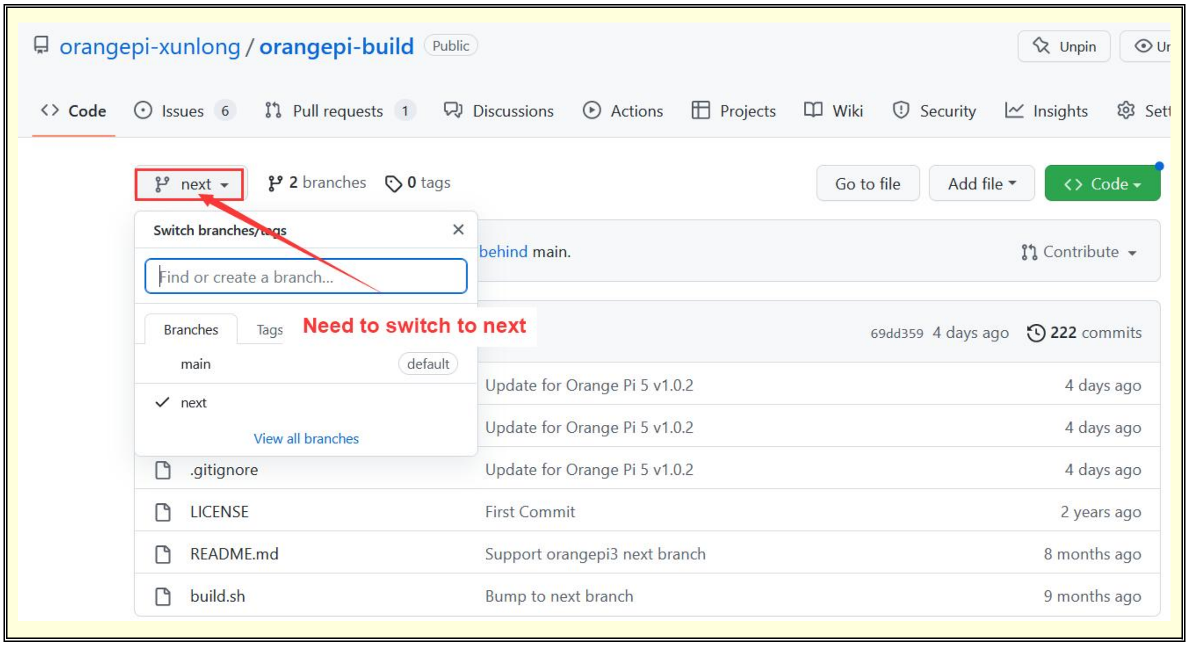
6. 2. Get the source code of Linux SDK

6. 2. 1. Download Orange-Build from github

1) LINUX SDK actually refers to the Orange-Build code. Orange-Build is modified based on the Armbian Build compilation system. Orangepi-Build can compile multiple versions of Linux Images. First download the code-build code, and the command is shown below:

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

Note that the Orange Pi 5 development board needs to download the next branch source code of Orangepi-Build,The git clone command above needs to specify the branch of orangepi-build source code to next.



Download Orangepi-build code through the Git Clone command is the username and password that does not need to enter the GitHub account (the same is the same for downloading other code in this manual). Names and passwords are usually input errors in the address input of the OrangePi-Build repository behind Git Clone. Please carefully check whether the command is wrong, instead of thinking that we have forgotten the user name and password of the GitHub account here。

2) The U-Boot and Linux kernel versions currently used in the development board are shown below

Branch	u-boot version	Linux Kernel version
legacy	u-boot 2017.09	linux5.10
current	u-boot 2017.09	linux6.1

The branches mentioned here are not the same thing as orangepi-build source code, please do not confuse. This branch is mainly used to distinguish the different kernel source code。

We currently define the linux5.10 bsp kernel provided by RK as the legacy branch, and the linux6.1 bsp kernel as the current branch.

3) Orangepi-Build will include the following files and folders after downloading

a. **build.sh**: Compile the startup script
b. **external**: Including configuration files, specific scripts, and source code of some programs, etc.
c. **LICENSE**: GPL 2 license file
d. **README.md**: orangepi-build description file
e. **scripts**: General script compiled Linux image

```
test@test:~/orangepi-build$ ls
build.sh   external   LICENSE   README.md   scripts
```

If the Orangepi-Build code downloaded from GitHub, you may find that after downloading, you may find that Orangepi-build does not include the source code of the U-Boot and Linux kernels, nor does it compile the U-Boot and Linux kernel needs to be used. Chain, this is normal, because these things are stored in other separate GitHub warehouses or some servers (the address will be described in detail below). Orangepi-build will specify the address of the U-Boot, Linux kernel and cross compilation tool chain in the script and configuration file. When running Orange-Build, when it is found that there are no these things in the local area, it will automatically download the corresponding places

6. 2. 2. Download the cross compilation tool chain

Only by using Orangepi-Build to compile Images in the X64 computer, the cross compile tool chain is downloaded. The linux image compiled in the development board's Ubuntu22.04 will not download the cross compilation tool chain. At this time, Orange-Build/Toolchains will be an empty folder

1) Orangepi-Build will automatically download the cross-compile tool chain in the **toolchains** folder when running for the first time. After each run Orange-Build's BUILD.SH script, it will check whether the cross compile tool chain in **toolchains** exists. If If there is no existence, you will start downloading again. If you exist, you will not download it directly.

```
[ 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
```

2) The image website of the cross-compilation tool chain in China is the open source software Image station of Tsinghua University

https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/

3) After downloading **toolchains**, it will contain multiple versions of cross compilation tool chain. The development board will only use two of them.

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
gcc-arm-11.2-2022.02-x86_64-arm-none-linux-gnueabihf
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

4) The cross compilation tool chain used by compiling the linux kernel source code is

a. linux5.10 and linux6.1

```
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
```

5) The cross compilation tool chain used by compiling u-boot source code is

a. v2017.09

```
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
```

6. 2. 3. orangepi-build complete directory structure description

1) OrangePi-Build warehouse does not include Linux kernel, U-Boot source code, and cross compilation tool chain. The source code of Linux kernel and U-Boot is stored in an independent Git warehouse.

a. The git repository where the Linux5.10 kernel source code is stored is as follows:

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk35xx
```

b. The git warehouse where the Linux6.1 kernel source code is stored is as follows:

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-6.1-rk35xx
```

c. The git warehouse stored in U-Boot source code is shown below：

```
https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2017.09-rk3588https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2018.05-sun50iw9
```

2) Orangepi-Build will download the cross-compilation tool chain, U-Boot and Linux kernel source code when running the first run. After successfully compiling a Linux image, there are files and folders that can be seen in Orangepi-Build

a. **build.sh**: Compile the startup script

b. **external**: Contains the configuration file, a specific function script, and the source of some programs that the compilation Image. The rootfs compression packet cached during the compilation Image is also stored in external

c. **kernel**: Stores the source code of the Linux kernel. The folder named orange-pi-5.10-rk35xx stores the kernel source code of the legacy branch of the RK3588/RK3588S series development board. The folder named orange-pi-6.1-rk35xx stores the RK3588 /Kernel source code of the current branch of the RK3588S series development board. Please do not modify the name of the folder of the kernel source code manually. If modified, the kernel source code will be re-downloaded when the compilation system is running.

d. **LICENSE**: GPL 2 License file

e. **README.md**: orangepi-build description file

f. **output**: Stay the compiled U-Boot, Linux and other deb bags, compile logs, and compile-generated image

g. **scripts**: General script compiled Linux image
h. **toolchains**: Staying cross compilation tool chain
i. **u-boot**: Store the u-boot source code. The folder named **v2017.09-rk3588** stores the u-boot source code of the legacy and current branches of the RK3588/RK3588S series development board. Please do not modify the name of the u-boot source code folder manually. , if modified, the u-boot source code will be re-downloaded when the compilation system is running.
j. **userpatches**: Store the configuration file needed to be used in the compilation script

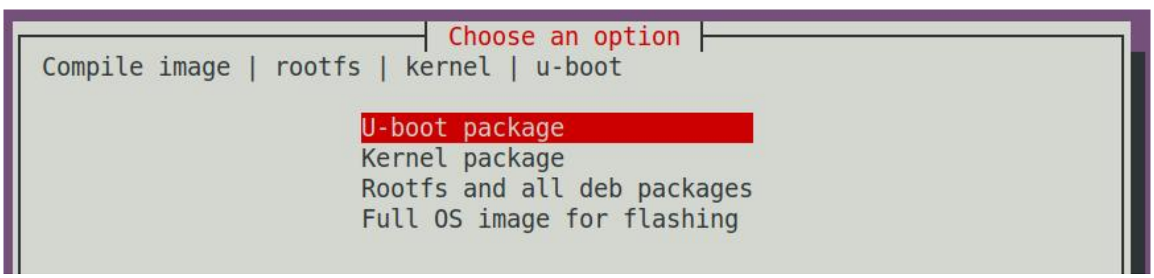
```
test@test:~/orangepi-build$ ls
build.sh  external  kernel  LICENSE  output  README.md  scripts
toolchains  u-boot  userpatches
```

6. 3. Compile u-boot

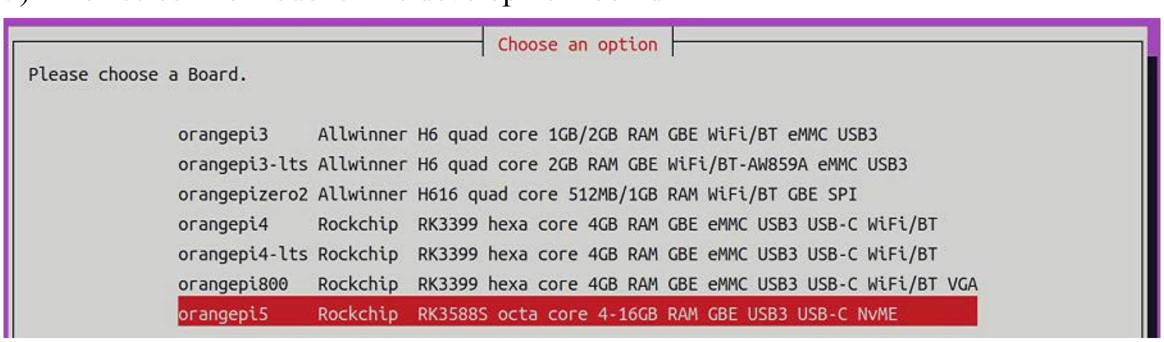
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package**, then press Enter



3) Then select the model of the development board



4) Then it will start to compile u-boot, and some information prompted during compilation is explained as follows

a. u-boot source code version

```
[ o.k. ] Compiling u-boot [ v2017.09 ]
```

b. The version of the cross-compilation toolchain

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 7.4.1 ]
```

c. Path to the generated u-boot deb package

```
[ o.k. ] Target directory [ orangepi-build/output/debs/u-boot ]
```

d. The package name of the generated u-boot deb package

```
[ o.k. ] File name [ linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb ]
```

e. Compilation time

```
[ o.k. ] Runtime [ 1 min ]
```

f. Repeat the command to compile u-boot, use the following command to start compiling u-boot directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi5 BRANCH=legacy
BUILD_OPT=u-boot KERNEL_CONFIGURE=no ]
```

5) View the u-boot deb package generated by compilation

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb
```

6) The files contained in the generated u-boot deb package are as follows

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ $ dpkg -x \
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb .    (Note that there is a "." at the
end of the command)
test@test:~/orangepi_build/output/debs/u-boot$ ls
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb    usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-legacy-orangepi5_1.0.2_arm64
```

```
│     ├── idbloader.img
│     ├── rkspi_loader.img
│     └── u-boot.itb
└── u-boot
      ├── LICENSE
      ├── orangepi_5_defconfig
      └── platform_install.sh

3 directories, 6 files
```

7) When the orangepi-bulid compilation system compiles the u-boot source code, it will first synchronize the u-boot source code with the u-boot source code of the github server, so if you want to modify the u-boot source code, you first need to turn off the download and update function of the source code **(need This function can only be turned off after u-boot has been fully compiled, otherwise it will prompt that the source code of u-boot cannot be found. If the source code compressed package is downloaded from Google Drive, there is no such problem because the source code of u-boot cached**), otherwise the changes made will be restored, the method is as follows：

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

8) When debugging u-boot code, you can use the following method to update u-boot in the linux image for testing

a. Upload the compiled u-boot deb package to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the deb package of u-boot installed

```
root@orangepi:~# apt purge -y linux-u-boot-orangepi5-legacy
```

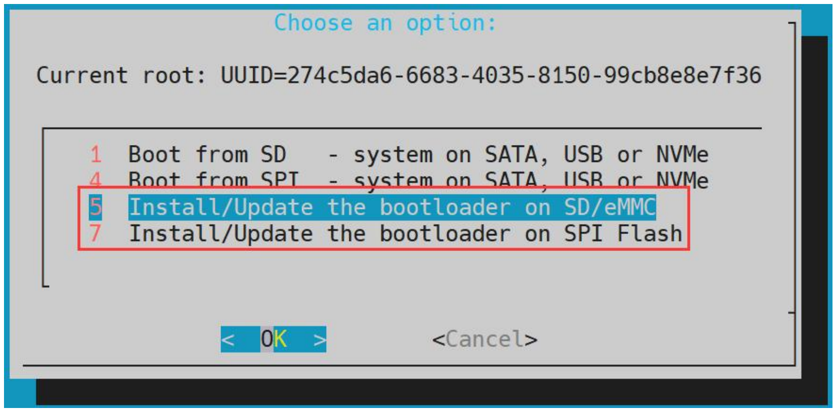
c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb
```

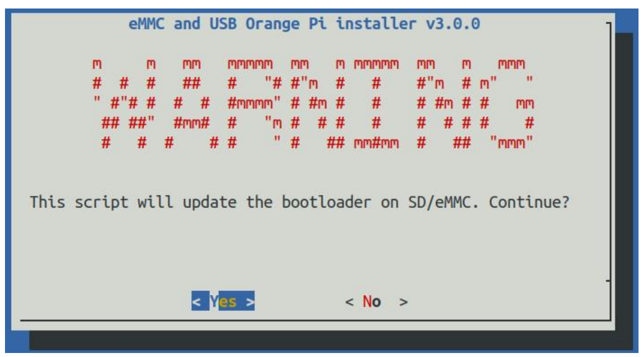
d. Then run the nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

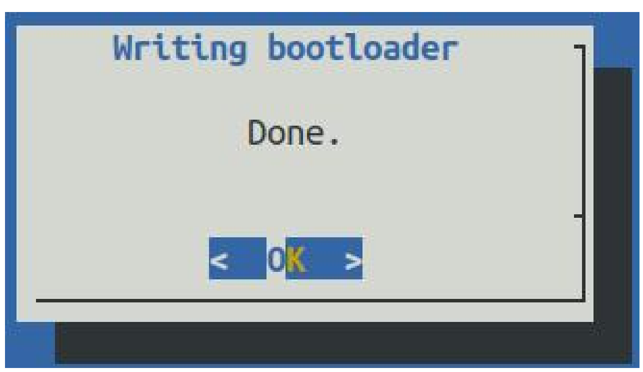
e. Then select **5 Install/Update the bootloader on SD/eMM** to update the u-boot in the TF card or **7 Install/Update the bootloader on SPI Flash** to update the u-boot in the SPI Flash



f. After pressing the Enter key, a Warning will pop up first



g. Press the Enter key again to start updating u-boot, and the following information will be displayed after the update is completed



h. Then you can restart the development board to test whether the modification of u-boot takes effect

9) Other useful information

a. In the u-boot 2017.09 source code, the defconfig configuration file used by the development board is

```
orangepi-build/u-boot/v2017.09-rk3588/configs/orangepi_5_defconfig
```

b. In the u-boot 2017.09 source code, the dts file used by the development board is

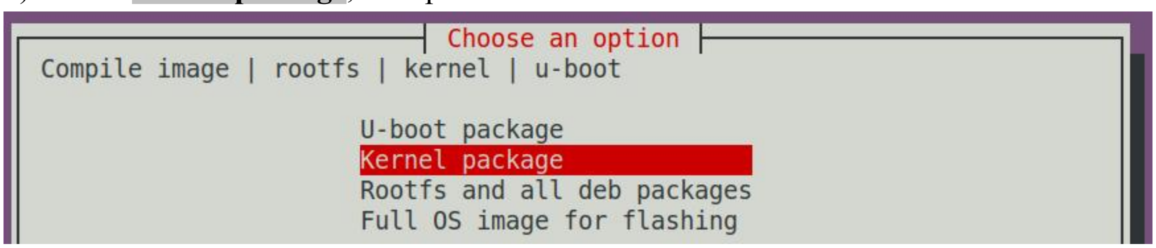
```
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3588s-orangepi-5.dts
```

6. 4. Compile the linux kernel

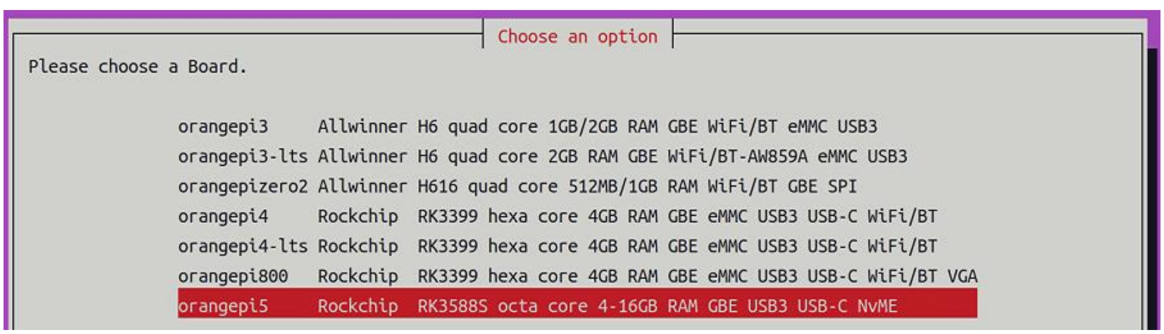
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

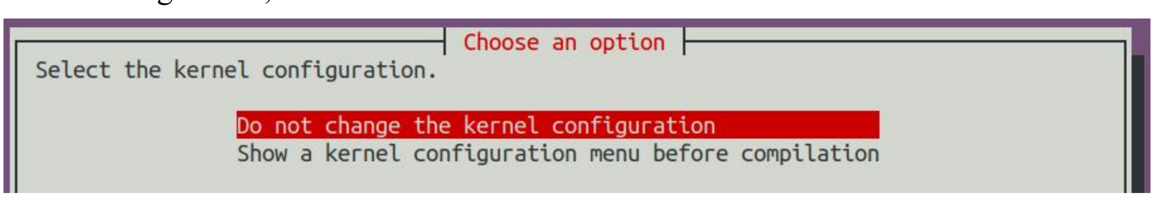
2) Select **Kernel package**, then press Enter



3) Then select the model of the development board



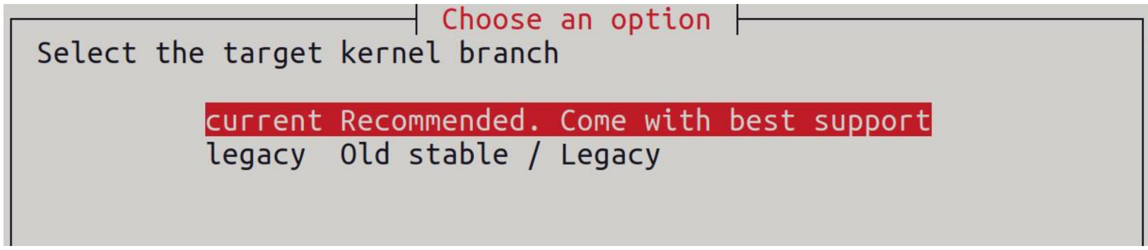
4) Then it will prompt whether to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If you need to modify the kernel configuration, select the second one.



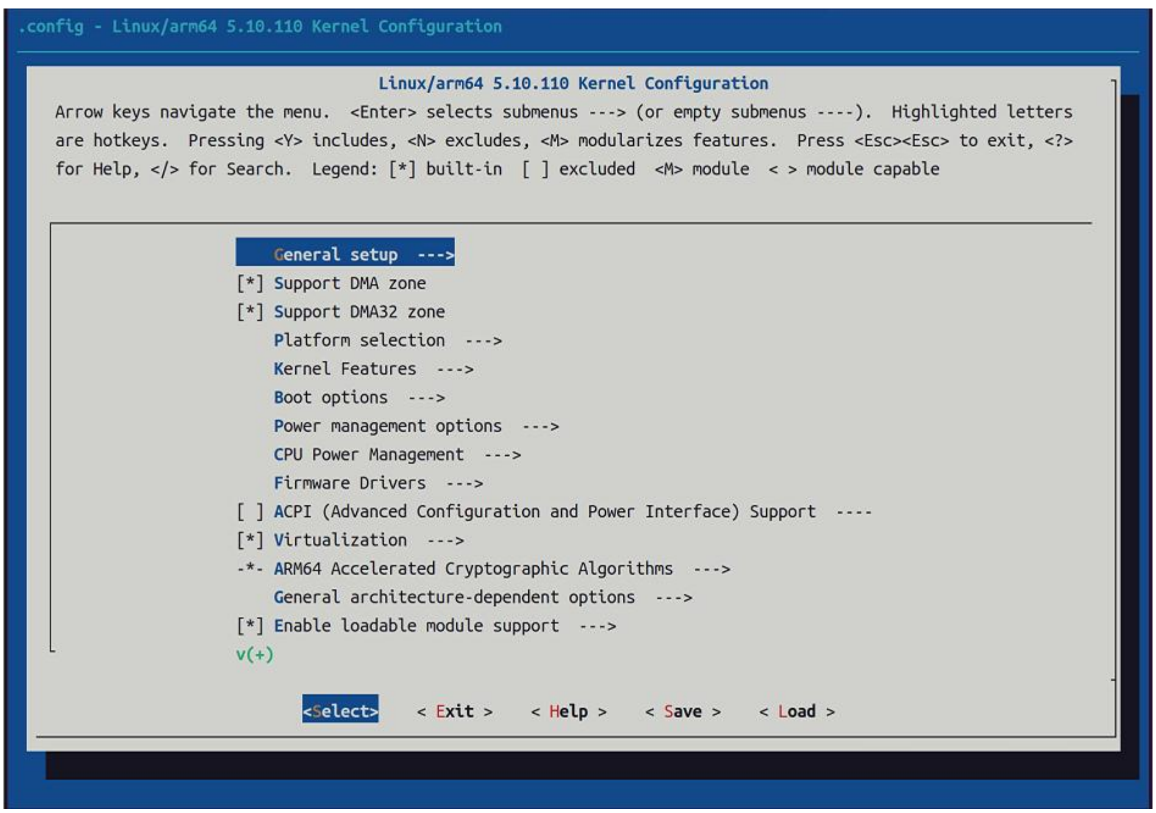
5) Then select the branch type of the kernel source code

a. legacy branch will compile the linux5.10 kernel source code

b. current branch will compile the linux6.1 kernel source code



6) If you choose to display the kernel configuration menu (the second option) in step 4), the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration, save and exit after modification. Yes, after exiting, the kernel source code will be compiled



a. If you do not need to modify the configuration options of the kernel, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily block the pop-up kernel configuration interface

```
test@test:~/orangepi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

b. You can also set **KERNEL_CONFIGURE=no** in the

orangepi-build/userpatches/config-default.conf configuration file, which can permanently disable this function

c. If the following error is displayed when compiling the kernel, it is because the terminal interface of the Ubuntu PC is too small to display the **make menuconfig** interface. Please maximize the terminal of the Ubuntu PC and run the build.sh script again

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

7) Part of the information prompted when compiling the kernel source code is as follows

a. The version of the linux kernel source code

[o.k.] Compiling current kernel [**5.10.110**]

b. The version of the cross-compilation toolchain used

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 11.2.1**]

c. The configuration file used by the kernel by default and the path where it is stored

[o.k.] Using kernel config file [**config/kernel/linux-rockchip-rk3588-legacy.config**]

d. The path of the deb package related to the kernel generated by compiling

[o.k.] Target directory [**orangepi-build/output/debs/**]

e. The package name of the compiled kernel image deb package

[o.k.] File name [**linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb**]

f. The time used for compilation

[o.k.] Runtime [**5 min**]

g. Finally, the compilation command to repeatedly compile the kernel selected last time will be displayed. Use the following command to start compiling the kernel source code directly without selecting through the graphical interface

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepi5 BRANCH=legacy**

```
BUILD_OPT=kernel KERNEL_CONFIGURE=no ]
```

8) View the deb package related to the kernel generated by compilation
   h. **linux-dtb-legacy-rockchip-rk3588_1.0.2_arm64.deb** Contains dtb files used by the kernel
   i. **linux-headers-legacy-rockchip-rk3588_1.0.2_arm64.deb** Include kernel headers
   j. **linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-legacy-rockchip-rk3588_1.0.2_arm64.deb
output/debs/linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb
output/debs/linux-headers-legacy-rockchip-rk3588_1.0.2_arm64.deb
```

9) The files contained in the generated linux-image deb package are as follows
   k. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi_build/output/debs$ mkdir test
test@test:~/orangepi_build/output/debs$ cp \
linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb test/
test@test:~/orangepi_build/output/debs$ cd test
test@test:~/orangepi_build/output/debs/test$ dpkg -x \
linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb .
test@test:~/orangepi_build/output/debs/test$ ls
boot   etc   lib   linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb   usr
```

   l. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.110-rockchip-rk3588
│   ├── System.map-5.10.110-rockchip-rk3588
│   └── vmlinuz-5.10.110-rockchip-rk3588
├── etc
│   └── kernel
├── lib
│   └── modules
├── linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb
```

```
└── usr
    ├── lib
    └── share
```

10) When the orangepi-bulid compilation system compiles the linux kernel source code, it first synchronizes the linux kernel source code with the linux kernel source code of the github server, so if you want to modify the linux kernel source code, you first need to turn off the update function of the source code **(the linux kernel needs to be fully compiled once This function can only be turned off after the source code, otherwise it will prompt that the source code of the linux kernel cannot be found. If the source code compressed package downloaded from Google Drive, there is no such problem, because the source code of linux has been cached)**, otherwise the modification 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"
```

11) If the kernel has been modified, the following method can be used to update the kernel and kernel modules of the development board linux system

m. Upload the deb package of the compiled linux kernel to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb root@192.168.1.xxx:/root
```

n. Then log in to the development board and uninstall the deb package of the installed linux kernel

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk3588
```

o. Install the deb package of the new linux kernel just uploaded

```
root@orangepi:~# dpkg -i linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb
```

p. Then restart the development board, and then check whether the kernel-related modifications have taken effect

```
root@orangepi:~# reboot
```

12) Other useful information

a. The storage location of the kernel configuration file is as follows. Please do not go to the kernel source code to find the kernel configuration file used by the development board.

a) Linux5.10

```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-legacy.config
```

b) Linux6.1

```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-current.config
```

b. The location of the dts file used by the development board

a) Linux5.10

```
orangepi-build/kernel/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5.dts
```

b) Linux6.1

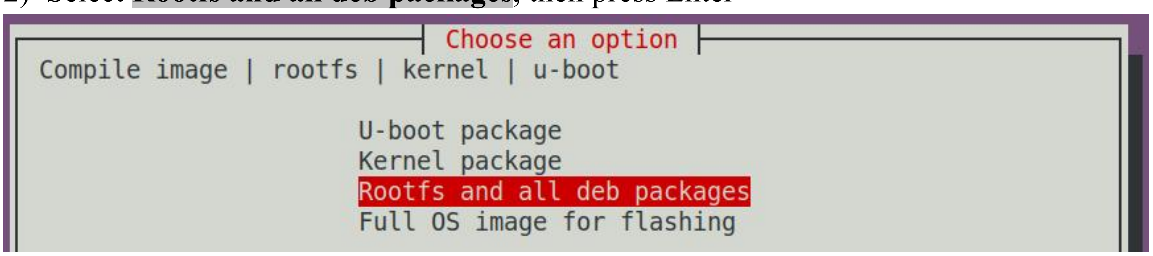
```
orangepi-build/kernel/orange-pi-6.1-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5.dts
```

6. 5. Compile rootfs

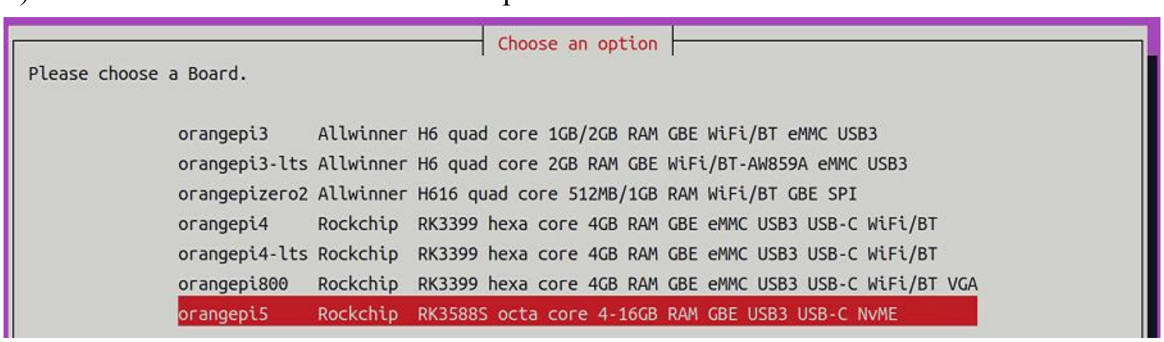
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

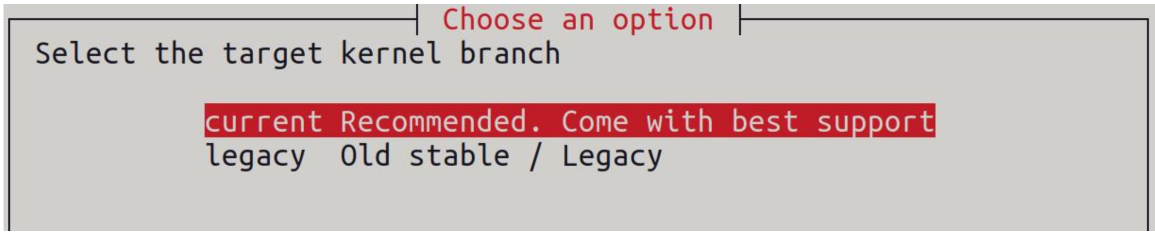
2) Select **Rootfs and all deb packages**, then press Enter



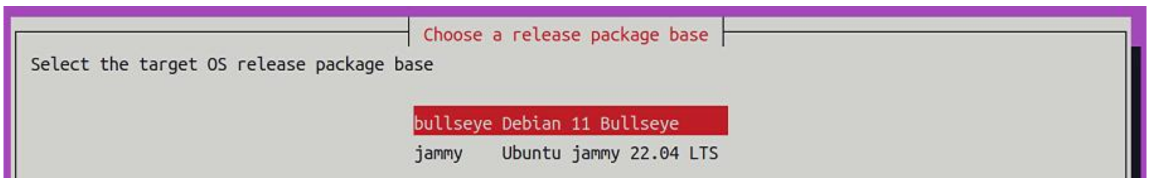
3) Then select the model of the development board



4) Then select the branch type of the kernel source code. Currently, the rootfs maintained by the kernel source code uses the same set.



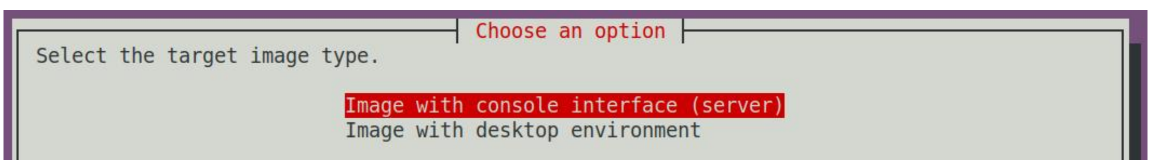
5) Then select the type of rootfs



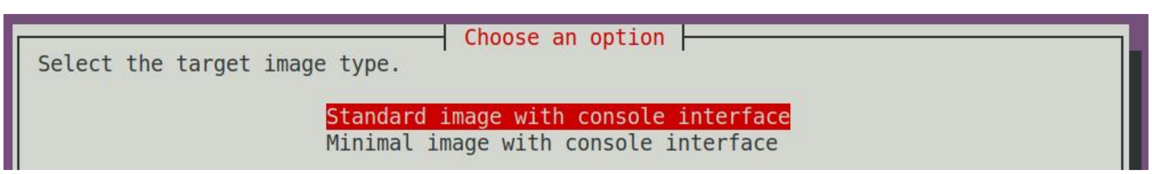
6) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small

b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

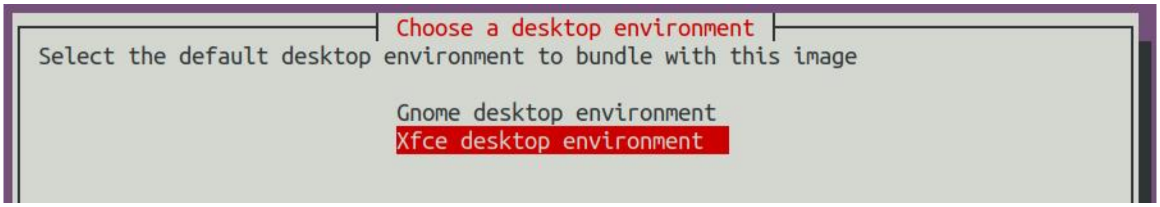


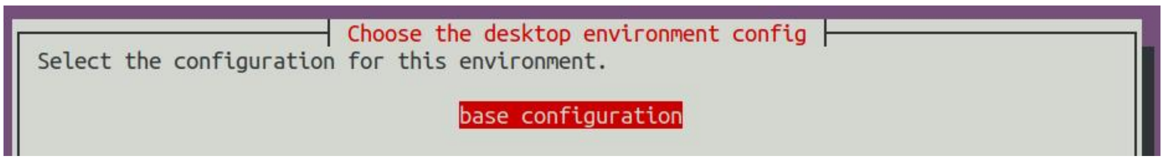
7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**



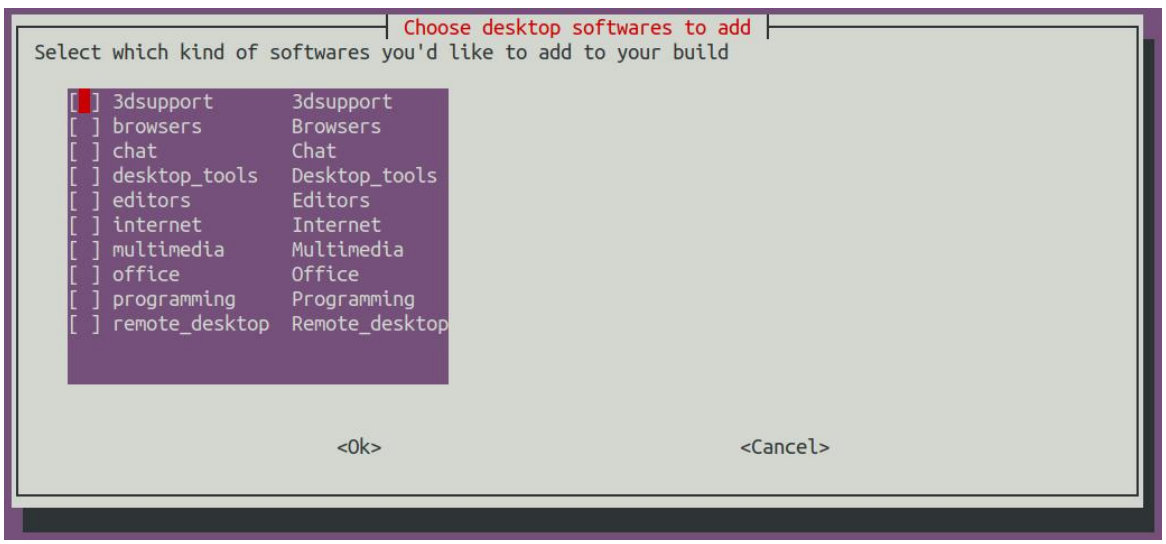
8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy supports XFCE and Gnome desktops,

while Ubuntu Focal and Debian only support XFCE





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



9) Then it will start to compile rootfs, and some of the information prompted during compilation are as follows

a. The type of rootfs

[o.k.] local not found [Creating new rootfs cache for **jammy**]

b. The storage path of the compiled rootfs compressed package

[o.k.] Target directory [**external/cache/rootfs**]

c. The name of the rootfs compressed package generated by compilation

[o.k.] File name [**jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**]

d. The time used for compilation

```
[ o.k. ] Runtime [ 13 min ]
```

10) View the rootfs compressed package generated by compilation

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is the rootfs compressed package, the meaning of each field of the name is

a) **jammy** indicates the type of linux distribution of rootfs

b) **xfce** means rootfs is the type of desktop version, if it is cli, it means the type of server version

c) **arm64** represents the architecture type of rootfs

d) **f930ff6ebbac1a72108a2e100762b18f** is the MD5 hash value generated by the package names of all software packages installed by rootfs. As long as the list of software packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to generate Determine whether rootfs needs to be recompiled

b. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.current
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list
```

11) If the required rootfs already exists under **external/cache/rootfs**, then compiling rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has If there is rootfs available in the cache, use it directly, which can save a lot of download and compilation time.

6. 6. Compile linux image

1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Full OS image for flashing**, then press Enter

```
Choose an option
Compile image | rootfs | kernel | u-boot

U-boot package
Kernel package
Rootfs and all deb packages
Full OS image for flashing
```

3) Then select the model of the development board

```
Choose an option
Please choose a Board.

orangepi3     Allwinner H6 quad core 1GB/2GB RAM GBE WiFi/BT eMMC USB3
orangepi3-lts Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
orangepizero2 Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
orangepi4     Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi4-lts Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi800   Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
orangepi5     Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NvME
```

4) Then select the branch type of the kernel source code

a. legacy branch will compile the linux5.10 kernel source code

b. current branch will compile the linux6.1 kernel source code

```
Choose an option
Select the target kernel branch

current Recommended. Come with best support
legacy  Old stable / Legacy
```

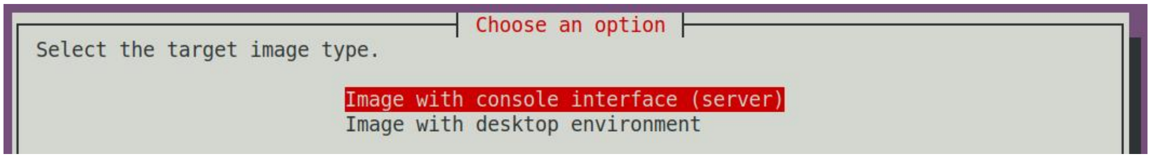
5) Then select the type of rootfs

```
Choose a release package base
Select the target OS release package base

bullseye Debian 11 Bullseye
jammy    Ubuntu jammy 22.04 LTS
```

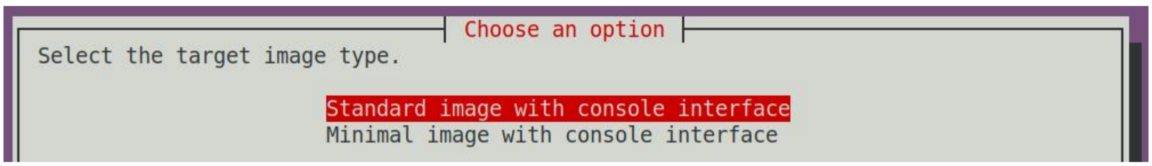
6) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small

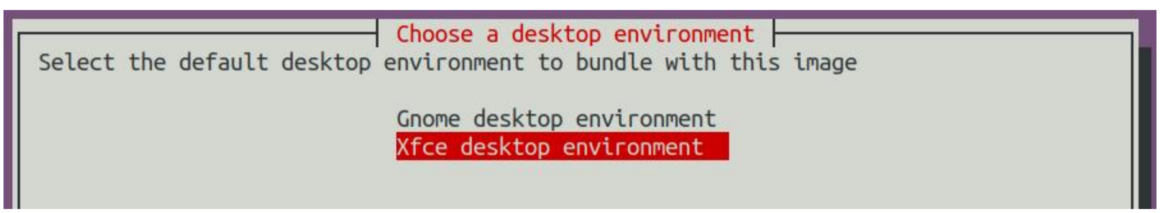
b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

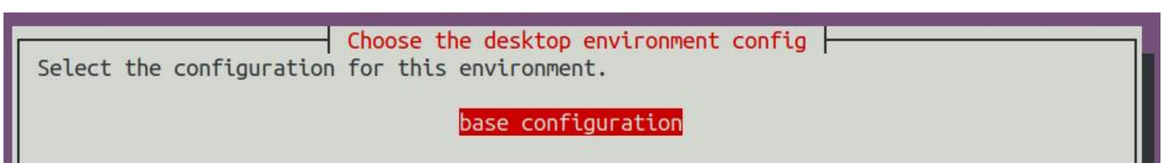


7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**

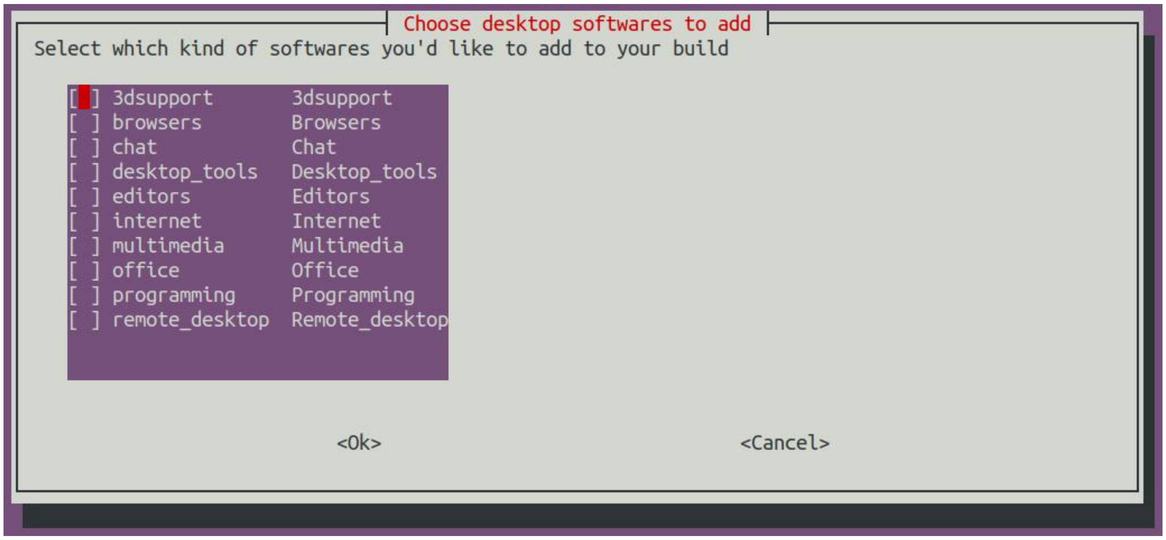


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy supports XFCE and Gnome desktops, while Ubuntu Focal and Debian only support XFCE





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



9) Then it will start to compile the linux image. The general process of compilation is as follows

a. Initialize the compilation environment of Ubuntu PC and install the software packages required for the compilation process

b. Download the source code of u-boot and linux kernel (if cached, only update the code)

c. Compile u-boot source code and generate u-boot deb package

d. Compile the linux source code and generate linux-related deb packages

e. Make the deb package of linux firmware

f. Make the deb package of the orangepi-config tool

g. Create a deb package supported by the board

h. If you are compiling the desktop image, you will also create desktop-related deb packages

i. Check whether the rootfs has been cached, if not, recreate the rootfs, if it has been cached, directly decompress and use

j. Install the previously generated deb package into rootfs

k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.

l. Then make an image file and format the partition, the default type is ext4

m. Then copy the configured rootfs to the imageed partition

n. Then update initramfs

o. Finally, write the bin file of u-boot into the image through the dd command

10) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/orangepi5_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160/orangepi5_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160.img ]
```

b. Compilation time

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the command to compile the image, and use the following command to start compiling the image directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepi5 BRANCH=legacy
BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no
BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

7. Linux Development Manual

7. 1. The method of compiling the kernel source code separately in the linux system of the development board

1) First download the Linux kernel source code of the development board

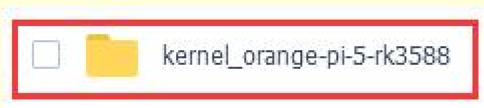
a. If you are using a Linux 5.10 kernel system, you need to download the orange-pi-5.10-rk35xx branch.

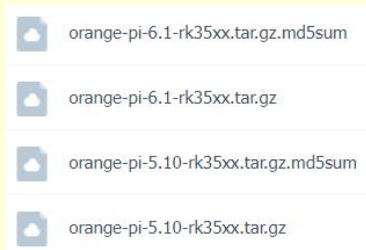
```
orangepi@orangepi:~$ git clone --depth=1 -b orange-pi-5.10-rk35xx https://github.com/orangepi-xunlong/linux-orangepi
```

b. If you are using a Linux 6.1 kernel system, you need to download the orange-pi-6.1-rk35xx branch.

```
orangepi@orangepi:~$ git clone --depth=1 -b orange-pi-6.1-rk35xx https://github.com/orangepi-xunlong/linux-orangepi
```

If you have problems downloading the code from github, you can go to the official information of the development board to download the compressed kernel source code package, then upload it to the Linux system of the development board, and then decompress it.





The command to decompress the kernel source code archive is:

a. Linux5.10 kernel

```
orangepi@orangepi:~$ tar zxf orange-pi-5.10-rk35xx.tar.gz
orangepi@orangepi:~$ mv orange-pi-5.10-rk35xx    linux-orangepi
```

b. Linux6.1 kernel

```
orangepi@orangepi:~$ tar zxf orange-pi-6.1-rk35xx.tar.gz
orangepi@orangepi:~$ mv orange-pi-6.1-rk35xx    linux-orangepi
```

After decompression, please execute the following command to synchronize the source code with github to ensure that the source code is in the latest state:

```
orangepi@orangepi:~$ cd linux-orangepi
orangepi@orangepi:~/linux-orangepi$ git pull
```

2) Then configure the default kernel configuration

```
orangepi@orangepi:~$ cd linux-orangepi
orangepi@orangepi:~/linux-orangepi$ make rockchip_linux_defconfig
```

rockchip_linux_defconfig The path in the kernel source code is

arch/arm64/configs/

3) Then compile the kernel source code

```
orangepi@orangepi:~/linux-orangepi$ make -j10
```

4) Then install the kernel module

```
orangepi@orangepi:~/linux-orangepi$ sudo make modules_install
```

The installation path of the kernel module is: /lib/modules

After executing the sudo make modules_install command, you can see that there will be an additional kernel module folder under /lib/modules/:

```
orangepi@orangepi5:~$ ls /lib/modules
# If it is a system with Linux 5.10 kernel, the output is as follows
5.10.160+    5.10.160-rockchip-rk3588
# If it is a system with Linux 6.1 kernel, the output is as follows
6.1.43+    6.1.43-rockchip-rk3588
```

5) Then install the kernel image and uInitrd

```
orangepi@orangepi:~/linux-orangepi$ sudo make install
```

The installation path of the kernel image and uInitrd is: /boot/

After executing the sudo make install command, you can see that there will be one more kernel file under /boot/:

```
orangepi@orangepi5:~/linux-orangepi$ ls /boot/vmlinuz*
# If it is a system with Linux 5.10 kernel, the output is as follows
/boot/vmlinuz-5.10.160+    /boot/vmlinuz-5.10.160-rockchip-rk3588
# If it is a system with Linux 6.1 kernel, the output is as follows
/boot/vmlinuz-6.1.43+    /boot/vmlinuz-6.1.43-rockchip-rk3588
```

The file /boot/Image is actually loaded when the system starts, and Image is a copy of the vmlinuz file.

6) Then install the dtb file into **/boot/dtb**

```
orangepi@orangepi:~/linux-orangepi$ sudo make dtbs_install INSTALL_DTBS_PATH=/boot/dtb/
```

7) Then restart the Linux system and the newly compiled kernel will be loaded

```
orangepi@orangepi:~$ uname -r
# If it is a Linux5.10 system, the output is as follows
5.10.160+
# If it is a Linux 6.1 system, the output is as follows
6.1.43+
```

8. Instructions for using the Android 12 system

8. 1. Supported Android versions

Android version	kernel version
Android 12	Linux5.10
Android 12 Box	Linux5.10

8. 2. Android function adaptation

Function	Android 12	Android12 Box
USB2.0x2	OK	OK
USB3.0x1	OK	OK
USB Type-C 3.0	OK	OK
DP display	OK	OK
M.2 NVMe SSD Boot	OK	OK
M.2 SATA SSD Boot	OK	OK

AP6275P-WIFI	OK	OK
AP6275P-Bluetooth	OK	OK
GPIO（26pin）	OK	OK
UART（26pin）	OK	OK
SPI（26pin）	OK	OK
I2C（26pin）	OK	OK
PWM（26pin）	OK	OK
3pin debug serial port	OK	OK
TF card start	OK	OK
HDMI video	OK	OK
HDMI audio	OK	OK
OV13850 camera	OK	OK
OV13855 camera	OK	OK
LCD1	OK	NO
LCD2	OK	NO
Gigabit Ethernet port	OK	OK
Network port status light	OK	OK
MIC	OK	OK
headphone playback	OK	OK
headphone recording	OK	OK
led light	OK	OK
GPU	OK	OK
NPU	OK	OK
VPU	OK	OK
switch button	OK	OK
HDMI CEC function	NO	OK

8. 3. How to use the USB wireless network card

1) The currently mirrored USB wireless network card models are as follows:

Chip model	Function	VID&PID	Adaptation
RTL8821CU	2.4G +5G WIFI+BT 4.2	0bda:c820	Support WIFI, Bluetooth and hotspot
RTL8723BU	2.4G WIFI+BT4.0	0bda:b720	Support WIFI and Bluetooth function, does not support hotspot

RTL8811CU	**2.4G +5G WIFI**	**0bda:c811**	**Support WIFI function and open hotspot**

2) The pictures of the above three USB wireless network cards are as follows：

a. The picture of the RTL8821CU USB wireless network card module is as follows：

b. The picture of the RTL8723BU USB wireless network card module is as follows：

c. The picture of the RTL8811CU USB wireless network card module is as follows：

3) The test methods of the above three types of USB wireless network cards are the same. First, the USB network card needs to be inserted into the USB interface of the development board.

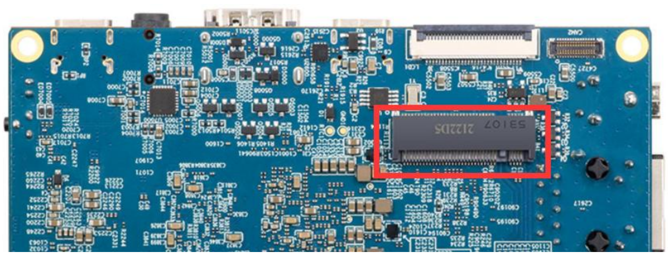
4) Then, for the connection and test method of WIFI, please refer to the section of WIFI connection test method

5) For the Bluetooth test, please refer to the Bluetooth test method section

8. 4. How to use AP6275P PCIe network card

1) First, you need to purchase an AP6275P PCIe network card as shown in the figure below

2) Then insert the AP6275P PCIe network card into the M.2 interface of the development board and fix it



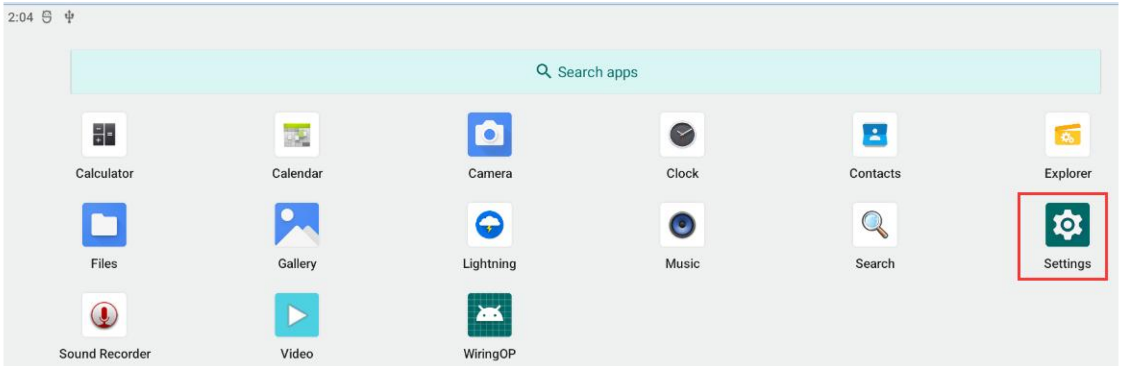
3) Then connect the power supply of the Type-C interface to the development board, and power on

4) After the system starts, please refer to **the section of WIFI connection test method** for WIFI connection and test method

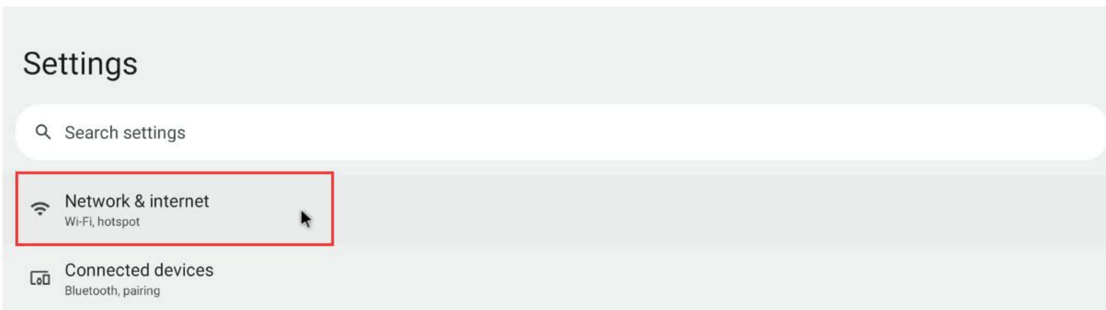
5) For the Bluetooth test, please refer to **the Bluetooth test method section**

8. 5. WIFI connection test method

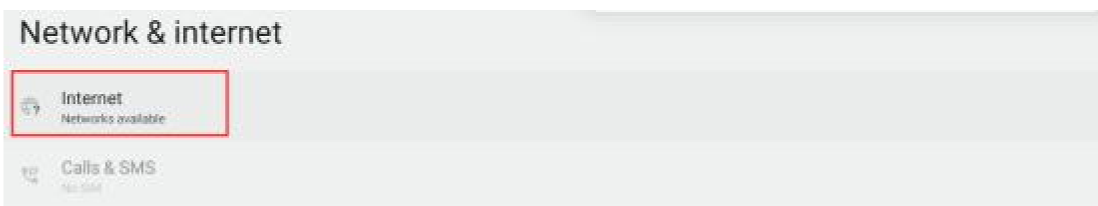
1) First click to enter the **Setting**



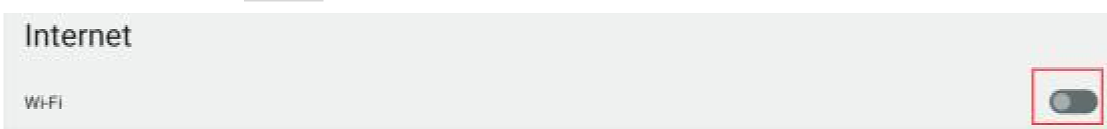
2) Then select **Network & internet**



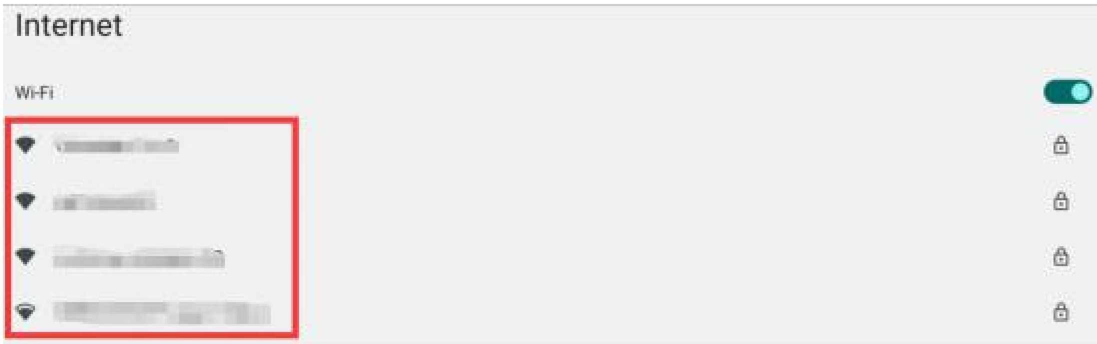
3) Then select **Internet**



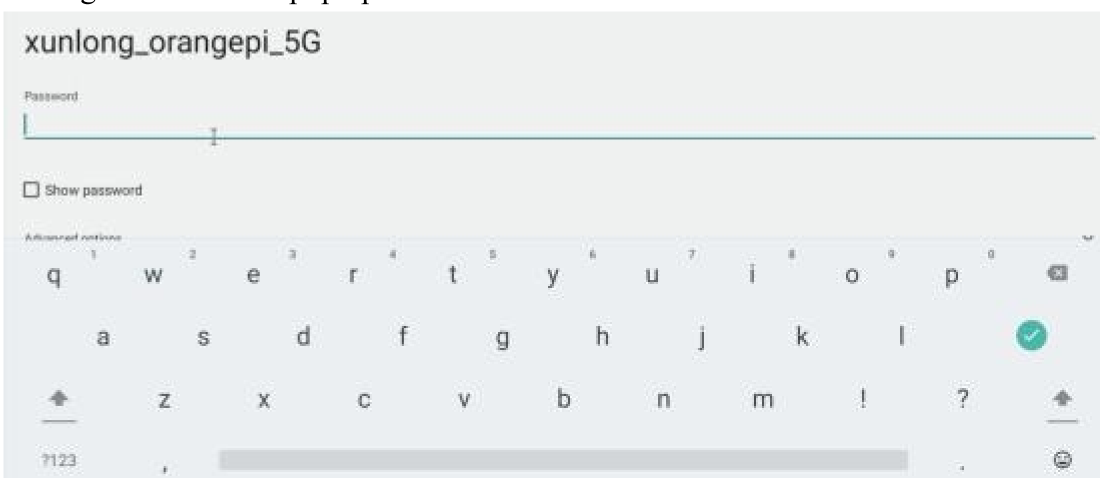
4) Then turn on the **Wi-Fi** switch



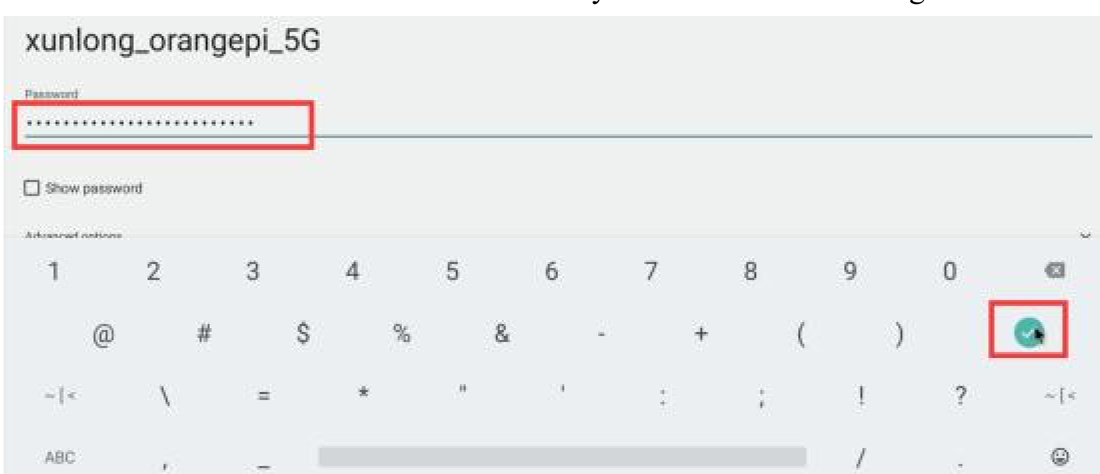
5) After turning on **Wi-Fi**, if everything is normal, you can scan to nearby Wi-Fi hotspots



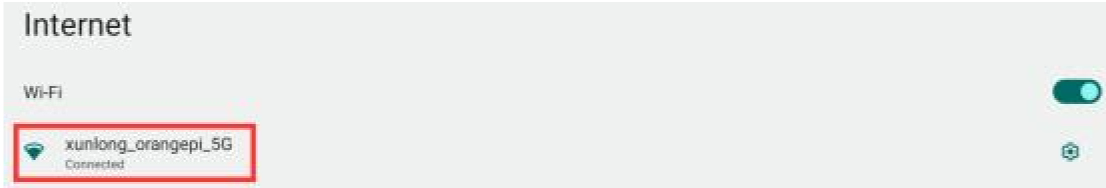
6) Then select the Wi-Fi you want to connect to, and the password input interface shown in the figure below will pop up



7) Then use the keyboard to enter the password corresponding to Wi-Fi, and then use the mouse to click the Enter button in the virtual keyboard to start connecting to Wi-Fi



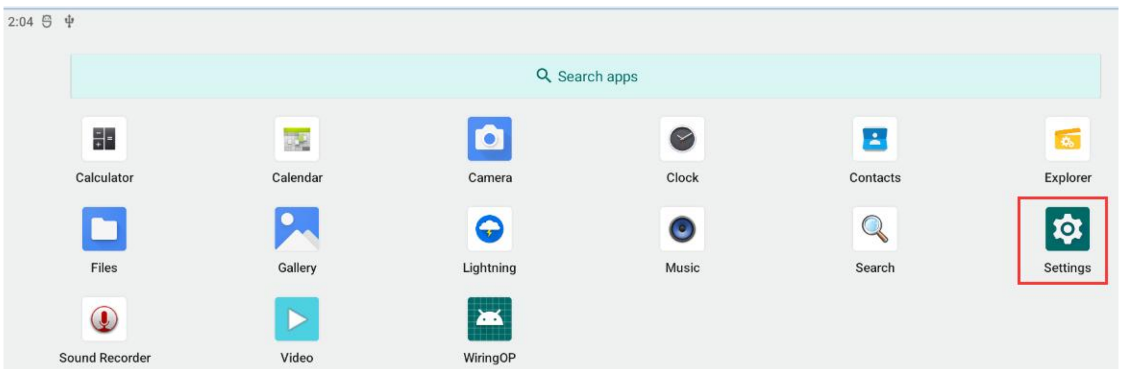
8) The display after successful Wi-Fi connection is shown in the figure below：



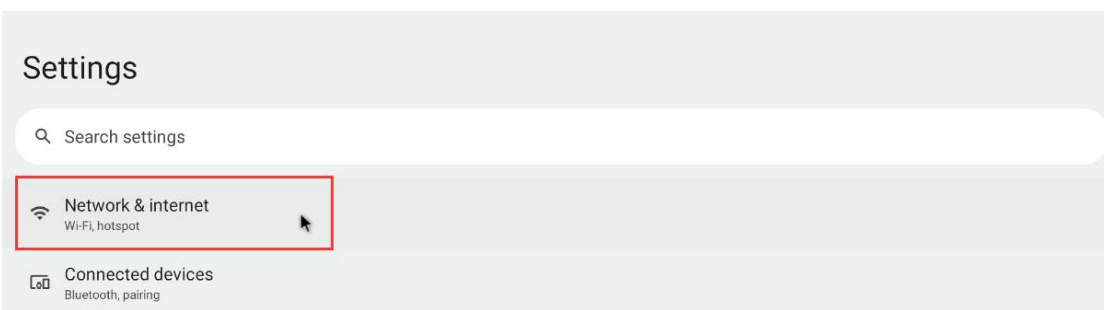
8. 6. How to use Wi-Fi hotspot

1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally

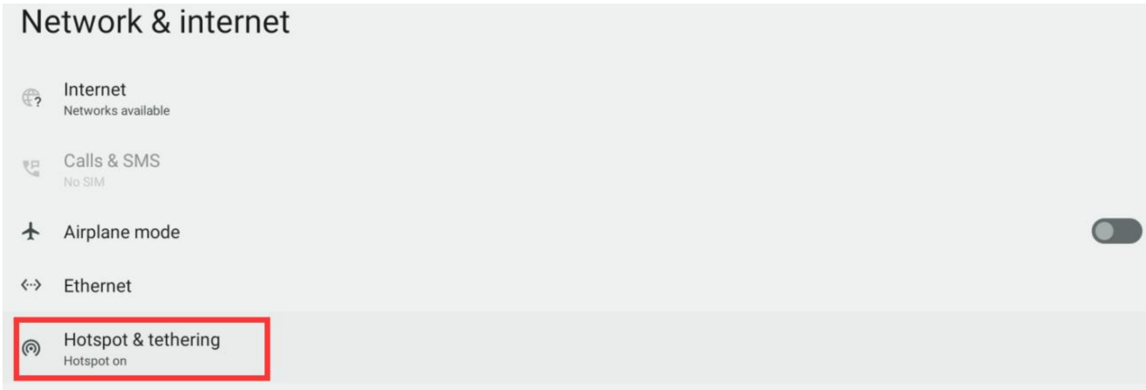
2) Then select **Settings**



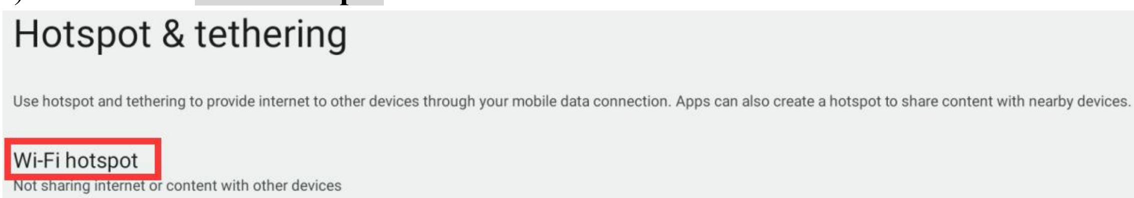
3) Then select **Network & internet**



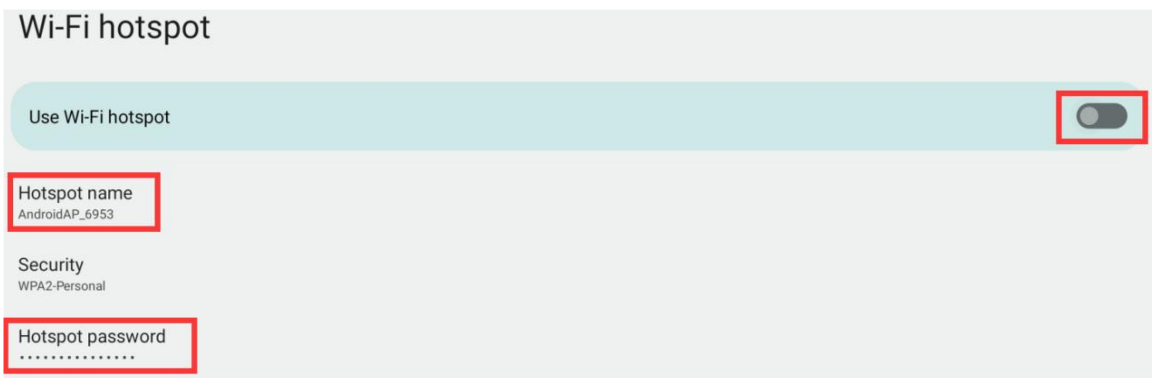
4) Then select **Hotspot & tethering**



5) Then select **Wi-Fi hotspot**



6) Then turn on the **Wi-Fi hotspot**, you can also see the name and password of the generated hotspot in the figure below, remember them, and use them when connecting to the hotspot (if you need to modify the name and password of the hotspot, you need to turn off the Wi-Fi first -Fi hotspot before modification)

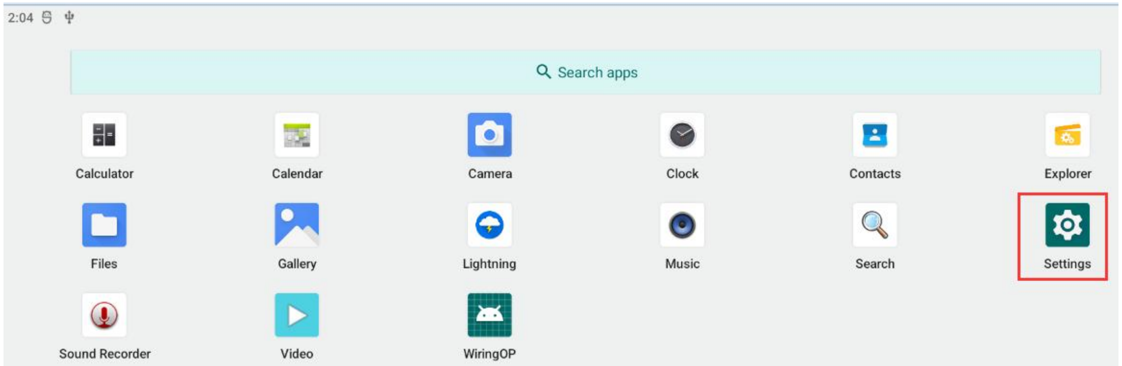


7) At this point, you can take out your mobile phone. If everything is normal, you can find the WIFI hotspot with the same name **(here AndroidAP_6953)** displayed under the **Hotspot name** in the above picture in the WI-FI list searched by the mobile phone. Then you can click AndroidAP_6953 to connect to the hotspot, and the password can be seen under the **Hotspot password** in the above picture

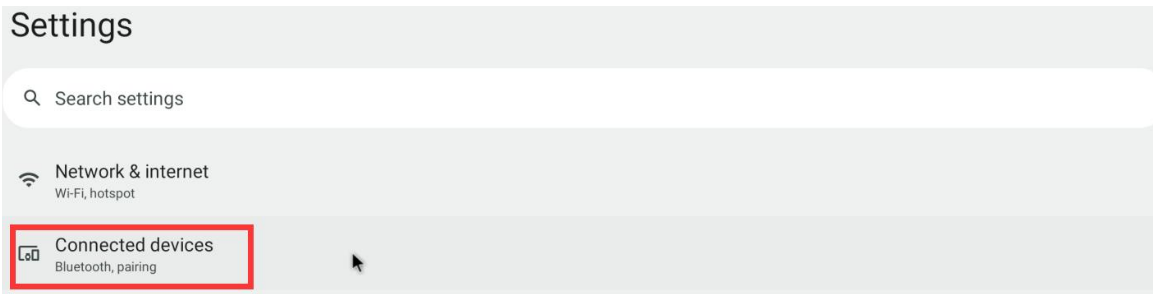
8) After the connection is successful, it will be displayed as shown in the figure below (the interface of different mobile phones will be different, the specific interface is subject to the display of your mobile phone). At this point, you can open a webpage on your mobile phone to see if you can access the Internet. If you can open the webpage normally, it means that the **WI-FI Hotspot** of the development board can be used normally.

8. 7. Bluetooth test method

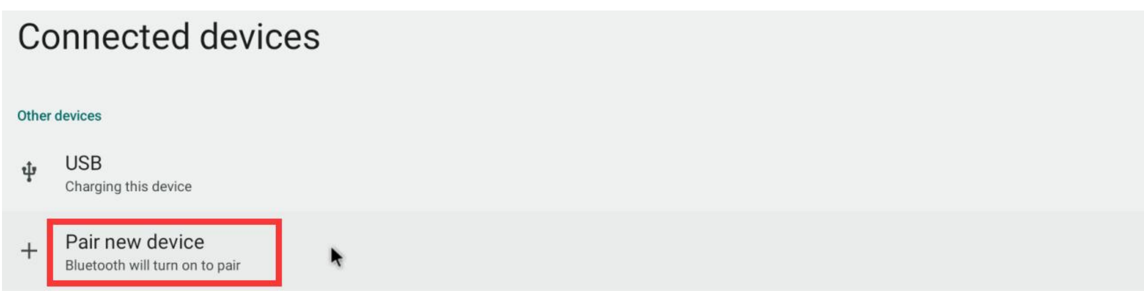
1) First click to enter the **Setting**



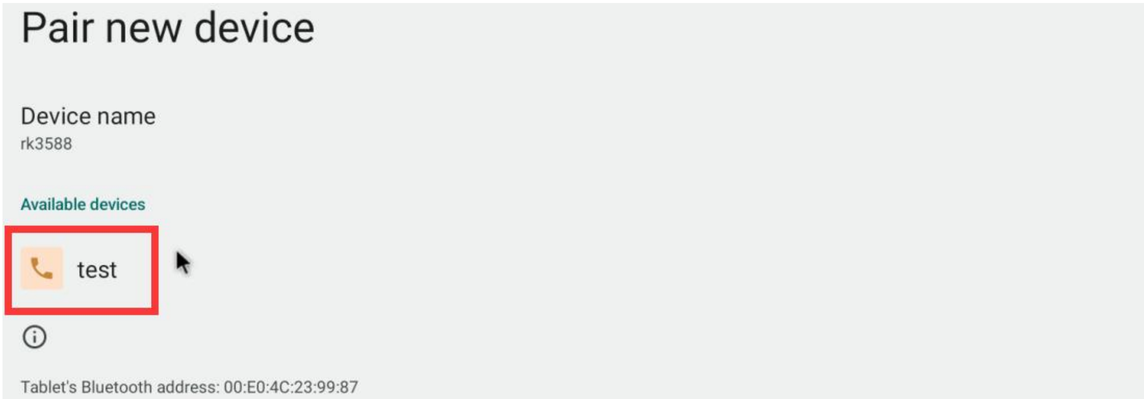
2) Then select **Connected devices**



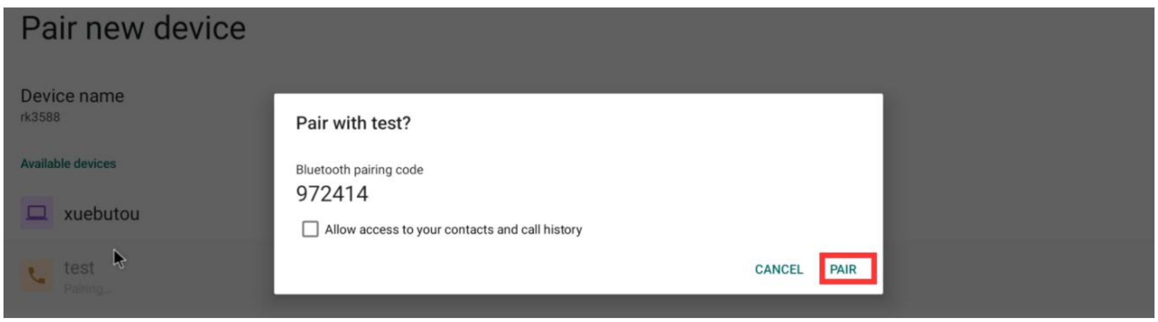
3) Then click **Pair new device** to turn on Bluetooth and start scanning the surrounding Bluetooth devices



4) The searched Bluetooth devices will be displayed under **Available devices**



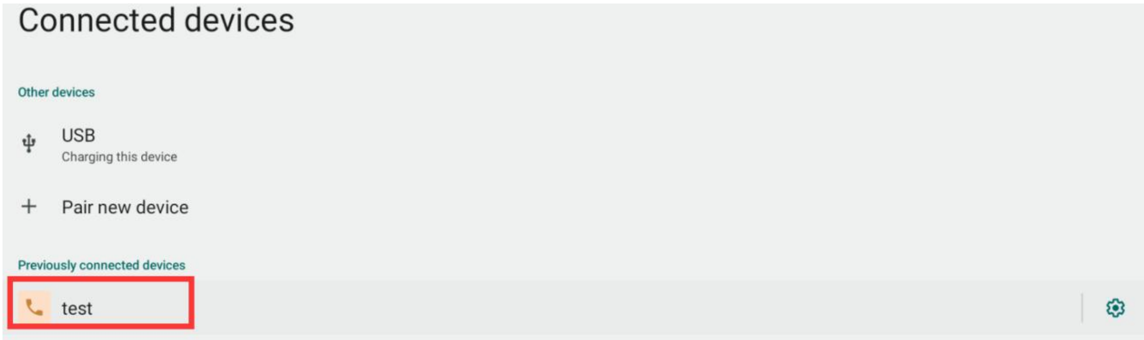
5) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option



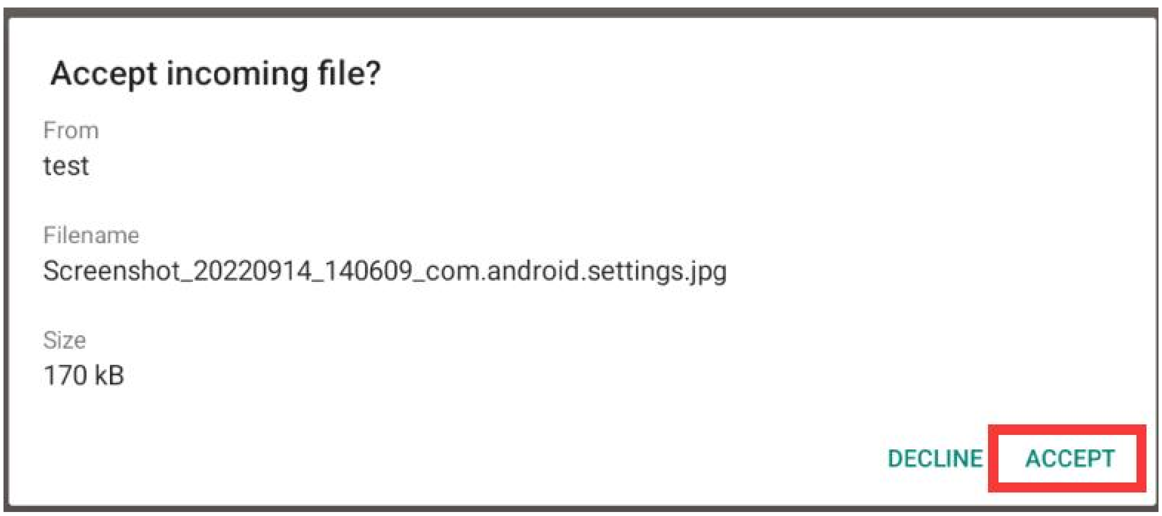
6) The test here is the configuration process of the development board and the Bluetooth of the Android mobile phone. At this time, the following confirmation interface will pop up on the mobile phone. After clicking the pairing button on the mobile phone, the pairing process will start

7) After the pairing is completed, you can see the paired Bluetooth device as shown in

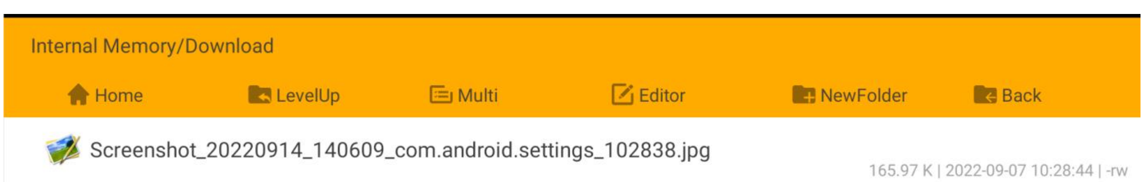
the figure below



8) At this time, you can use the Bluetooth of your mobile phone to send a picture to the development board. After sending, you can see the following confirmation interface in the Android system of the development board, and then click **Accept** to start receiving the picture sent by the mobile phone.



9) You can open the **Download** directory in the file manager to view the pictures received by the Android system Bluetooth of the development board



8. 8. How to use 10.1 Inch MIPI screen

Please make sure that the image used is the following three versions of the image:

OrangePi5_RK3588S_Android12_lcd_v1.x.x.img

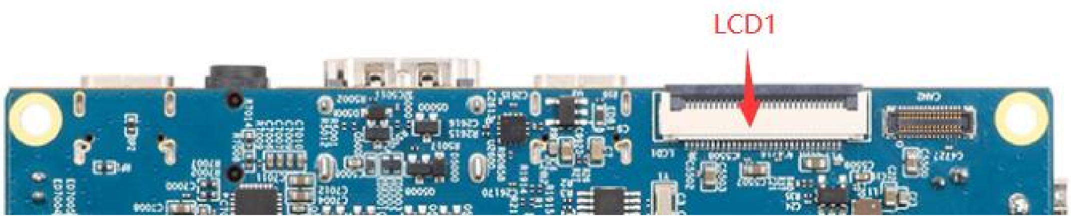
OrangePi5_RK3588S_Android12_spi-nvme_lcd_v1.x.x.img

OrangePi5_RK3588S_Android12_spi-sata_lcd_v1.x.x.img

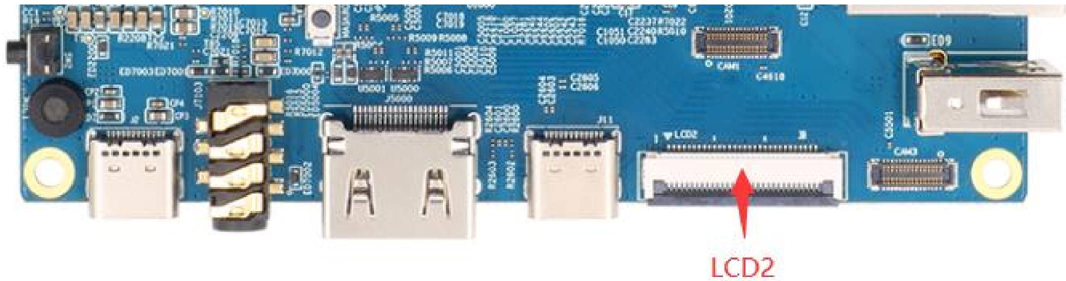
1) The screen needs to be assembled first, **please refer to the assembly method of the 10.1-inch MIPI screen**

2) There are two mipi lcd screen interfaces on the development board, we define:

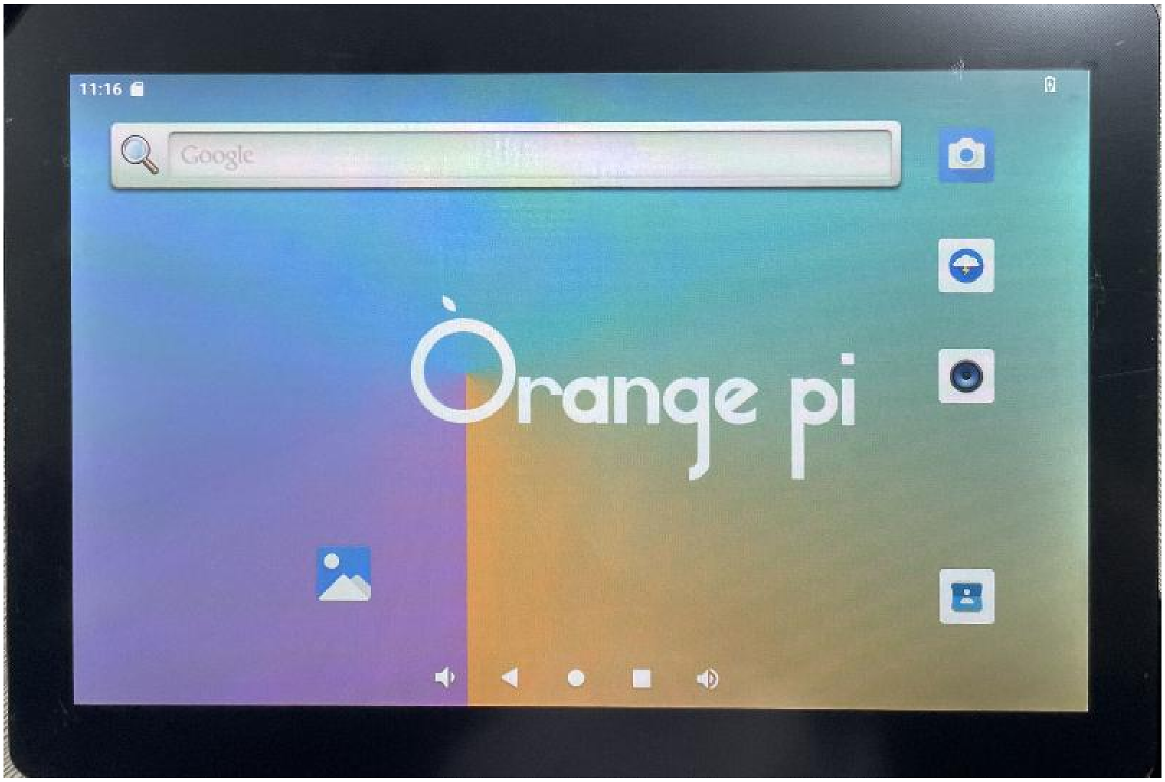
a. The location of the lcd1 interface is:



b. The location of the lcd2 interface is:



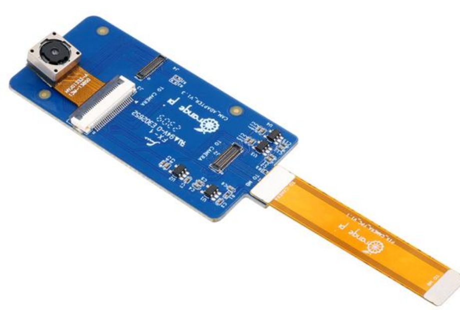
3) Connect the assembled screen to the lcd1 or lcd2 interface, connect the Type-C power supply to the board, and power on. After the system starts, you can see the screen display as shown in the figure below



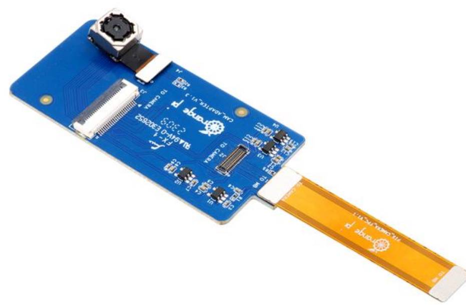
8. 9. Test method of OV13850 and OV13855 MIPI camera

Currently the development board supports two MIPI cameras, OV13850 and OV13855, the specific pictures are as follows：

a. OV13850 camera with 13 million MIPI interface



b. OV13855 camera with 13 million MIPI interface



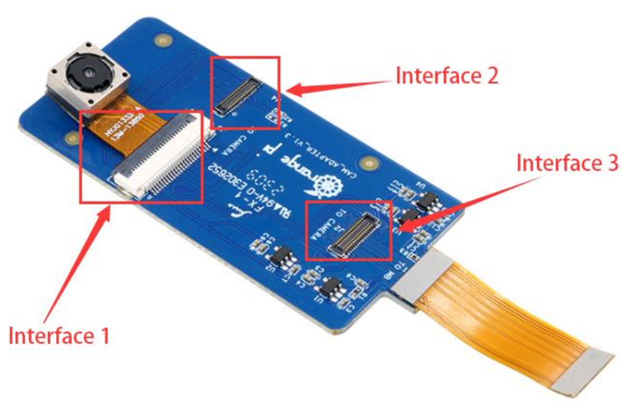
The adapter boards and FPC cables used by the OV13850 and OV13855 cameras are the same, but the positions of the two cameras connected to the adapter boards are different. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked **TO MB** needs to be inserted into the camera interface of the development board, and the end marked **TO CAMERA** needs to be inserted into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board, and only one can be used at a time, as shown in the figure below, of which:

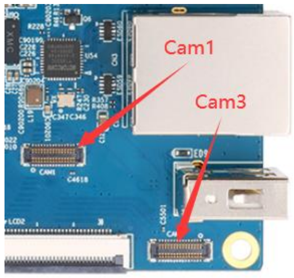
a. No.1 port is connected to OV13850 camera

b. No.2 interface is connected to OV13855 camera

c. No. 3 interface is not used, just ignore it



There are a total of 3 camera interfaces on the Orange Pi 5 development board. We define the positions of Cam1, Cam2 and Cam3 as shown in the figure below:



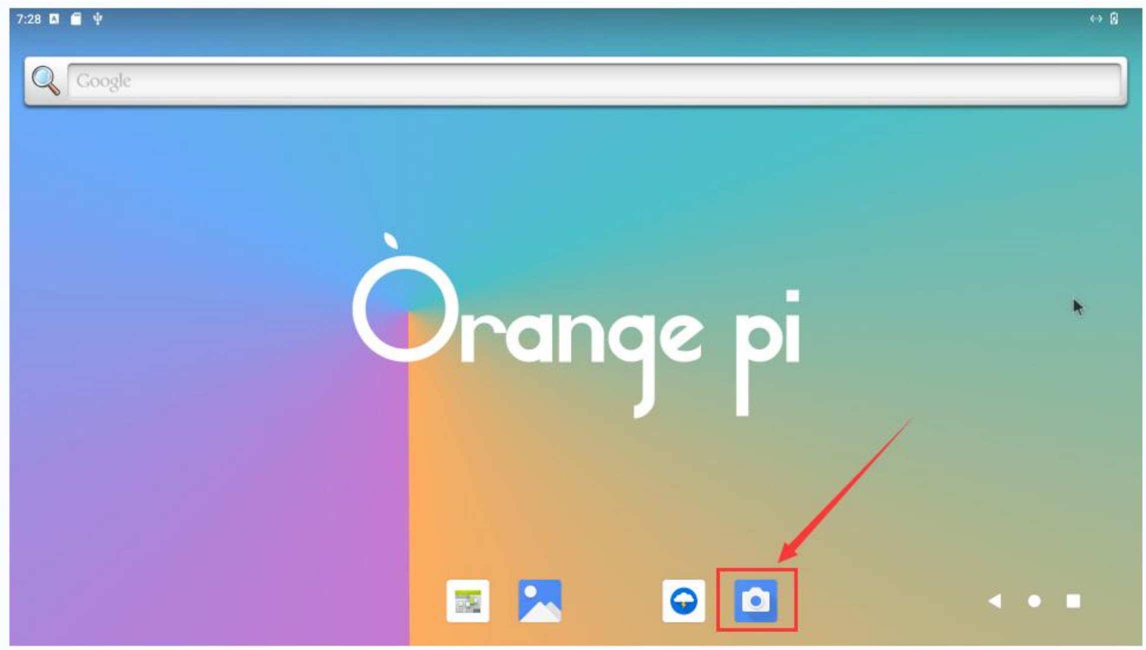
The method of inserting the camera into the Cam1 interface of the development board is as follows:

The method of inserting the camera into the Cam2 interface of the development board is as follows:

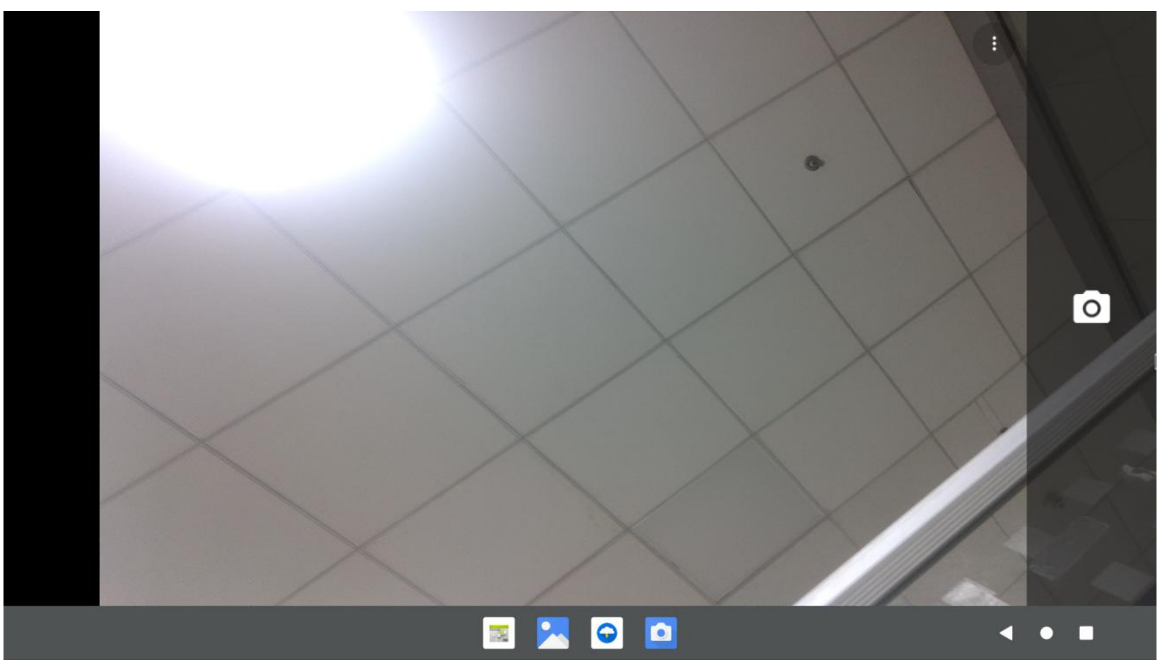
The method of inserting the camera into the Cam3 interface of the development board is as follows:

The Android system defaults to the configuration of **Cam1** and **Cam3**, so if you want to use the camera, please choose one of the **Cam1** and **Cam3** interfaces. After connecting the camera to the development board, we can use the following method to test the camera：

a. Open the camera APP on the desktop



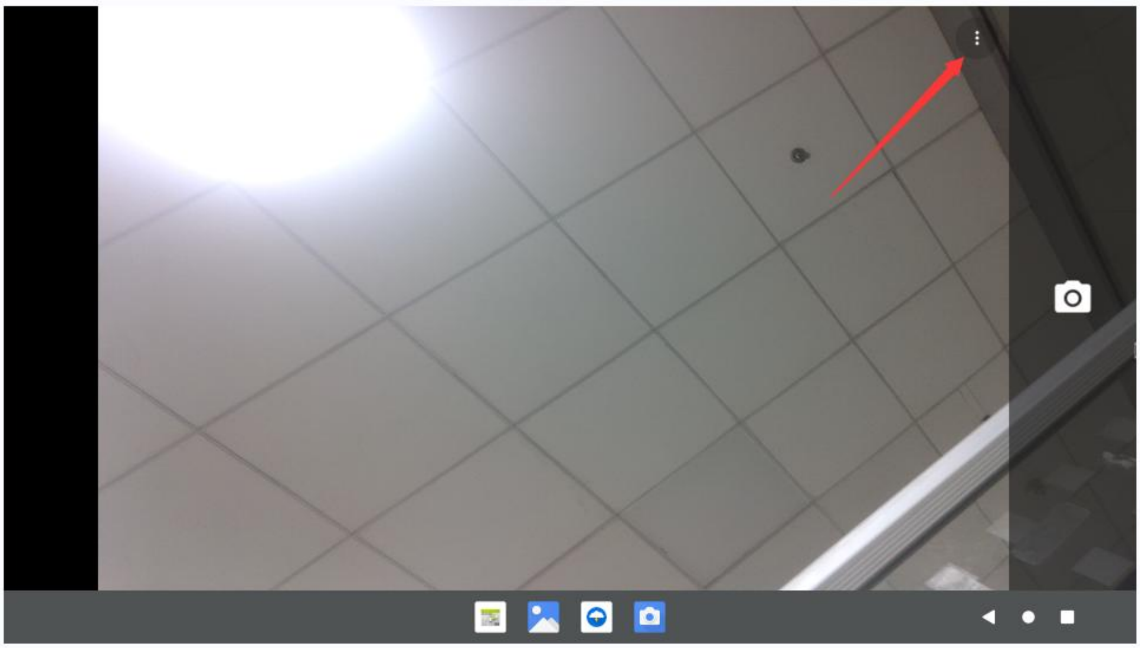
b. Then you can see the preview screen of the camera



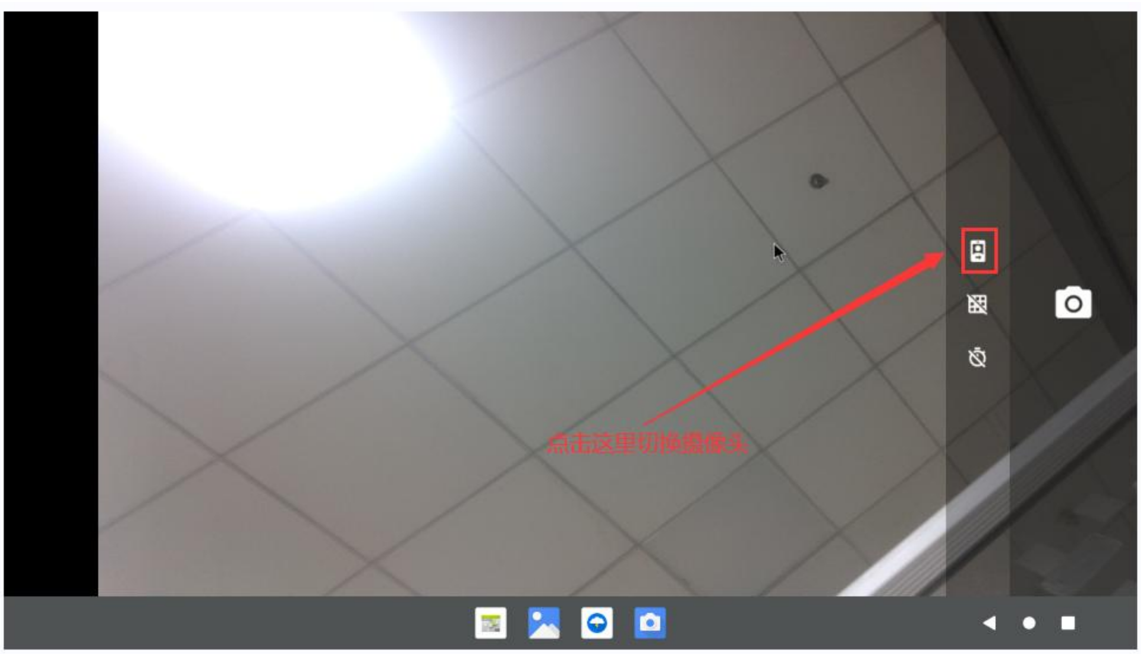
In addition to single camera, we can also connect two cameras at the same time. It should be noted that currently, please use the combination of **Cam1+Cam3** for testing dual cameras (support ov13850 and ov13855 mix and match). After connecting the dual cameras, open the camera APP to see the picture of one of the cameras as in the previous steps.

The method to switch to another camera is：

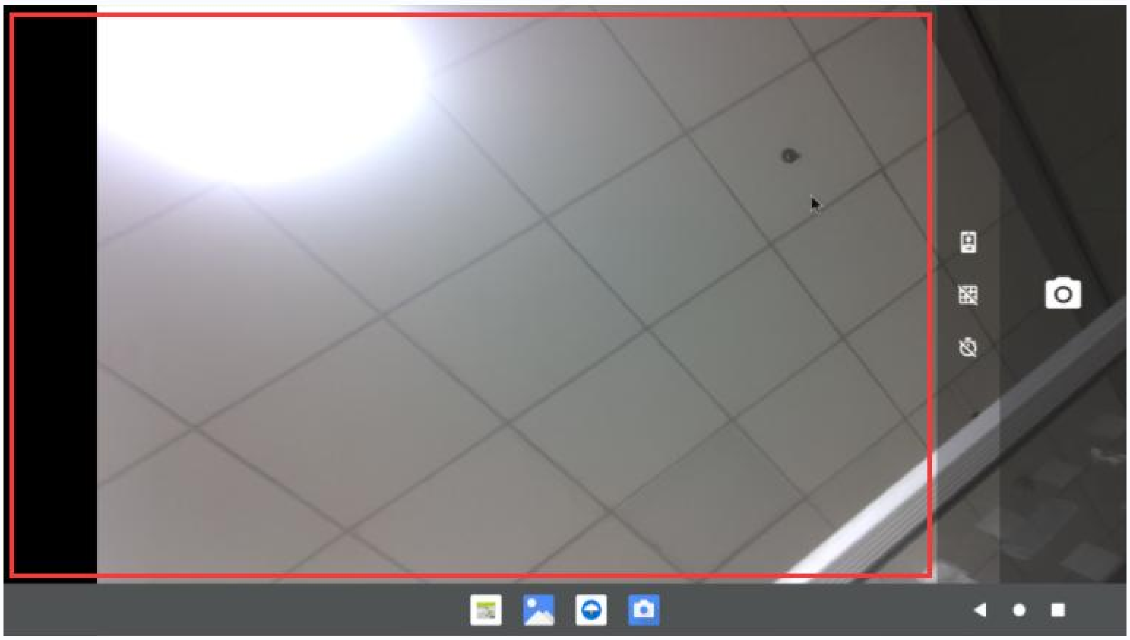
a. First click the three dots in the upper right corner



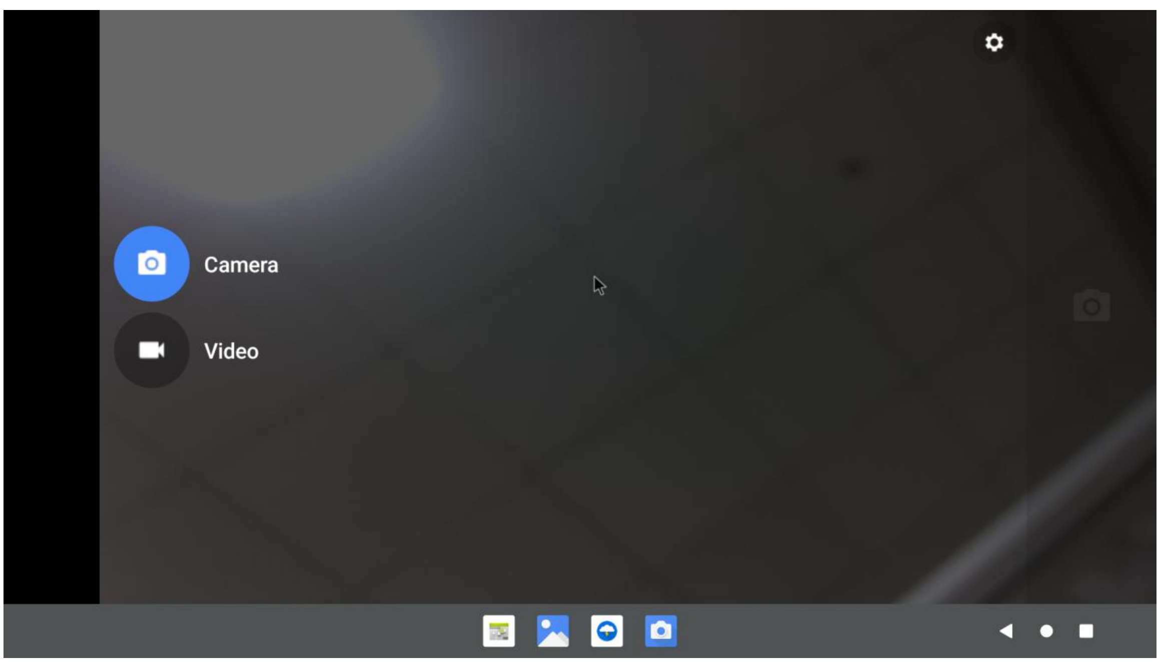
b. Then click the position shown in the figure below to switch the camera



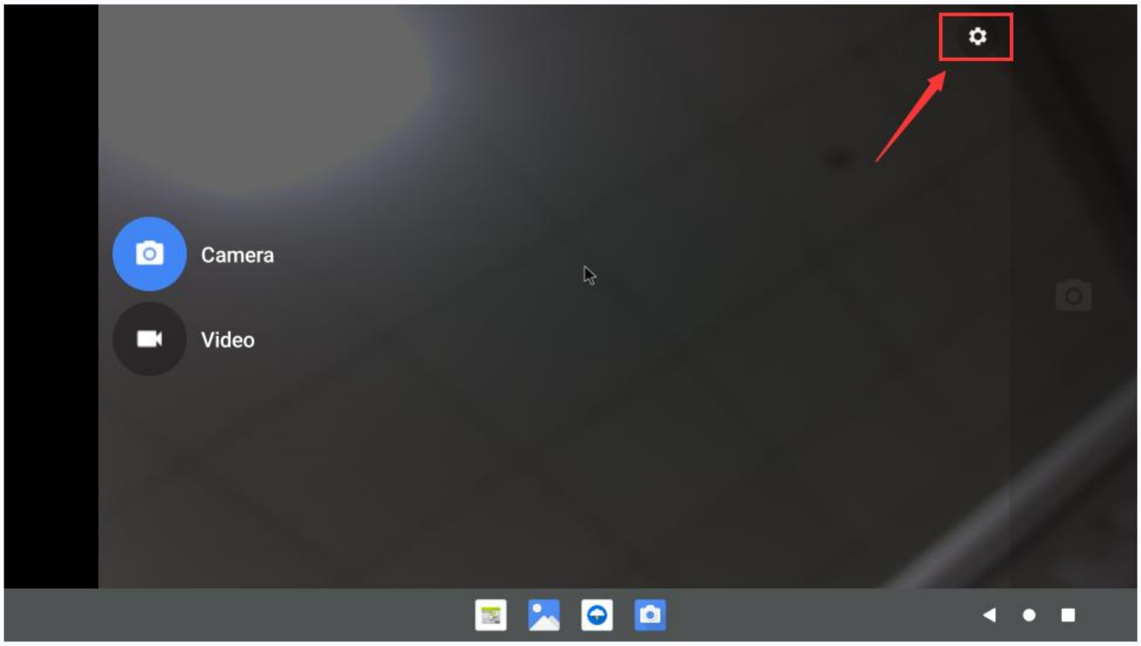
Press and hold the mouse in the area shown in the red box in the picture below of the camera APP and then drag to the right to call up the switching interface for taking pictures and recording



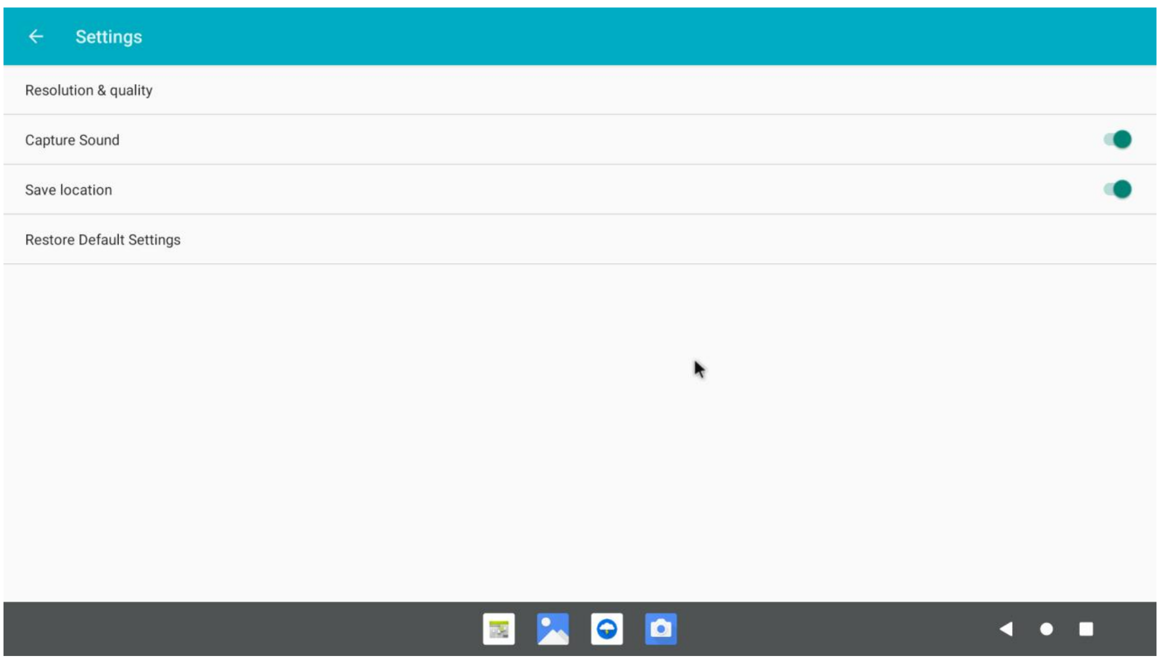
The switching interface of taking pictures and recording is as follows, click Video to switch to **video** recording mode



Click the position shown in the figure below to enter the camera setting interface



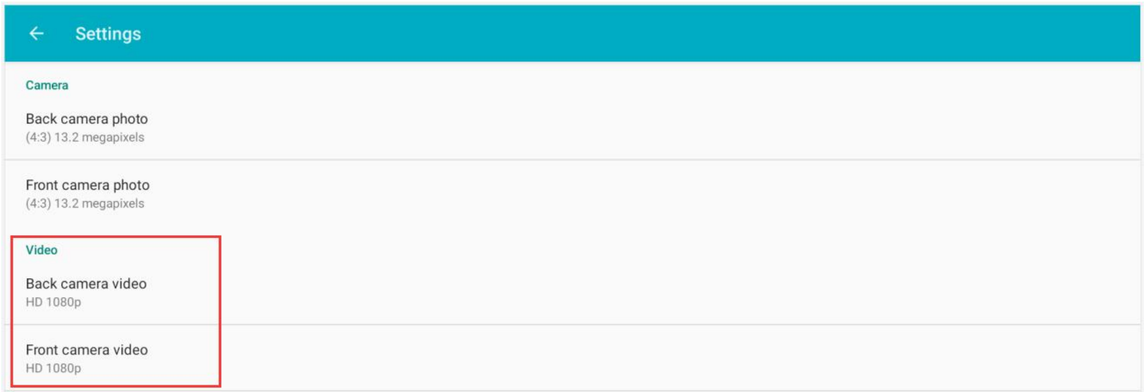
The setting interface of the camera is as follows:



Currently testing OV13850 does not support 4K video recording (OV13855 supports), only supports up to 1080p, please switch the video format to 1080p in the settings when recording video, the steps are as follows:

a. First enter the setting interface of the camera APP, and then click **Resolution & quality**

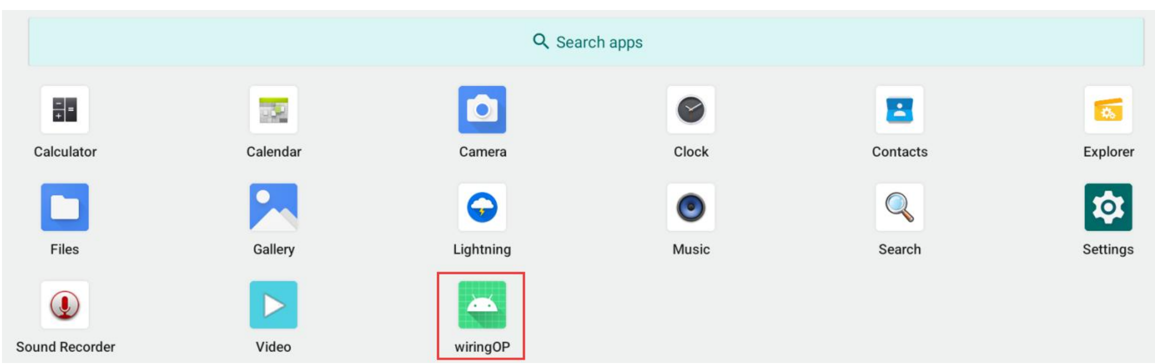
b. Then set the **Video** format to 1080p in Video



8. 10. 26pin interface GPIO, UART, SPI and PWM test

8. 10. 1. 26pin GPIO port test

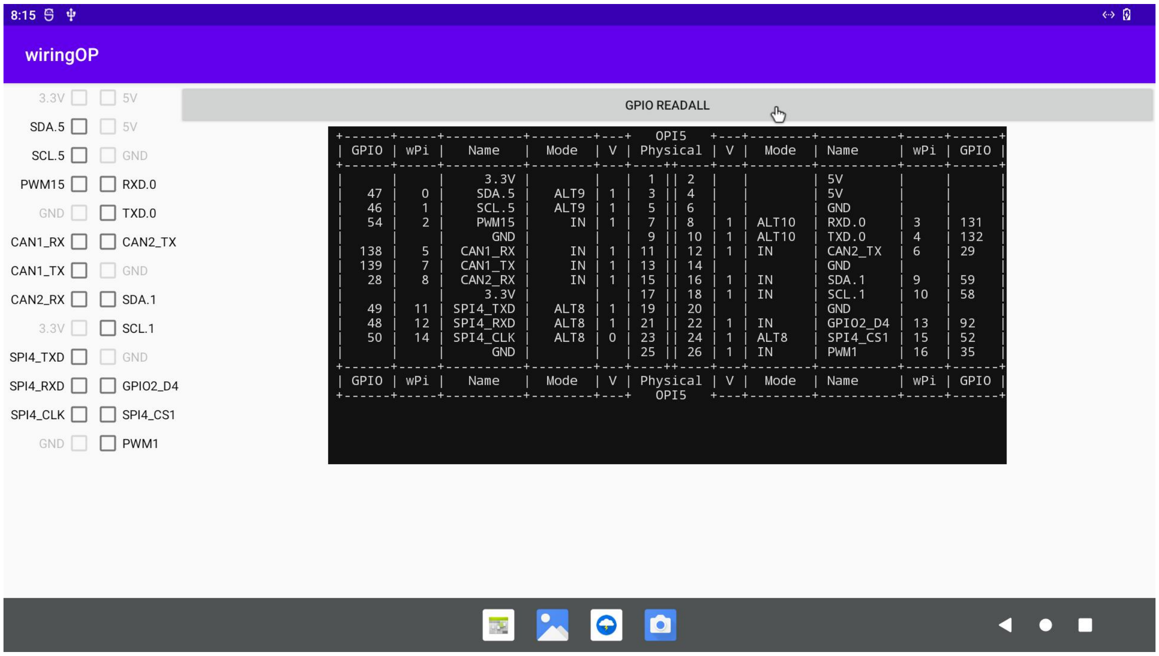
1) First click on the wiringOP icon to open the wiringOP APP



2) The main interface of wiringOP APP is displayed as shown in the figure below, and then click the **GPIO_TEST** button to open the GPIO test interface

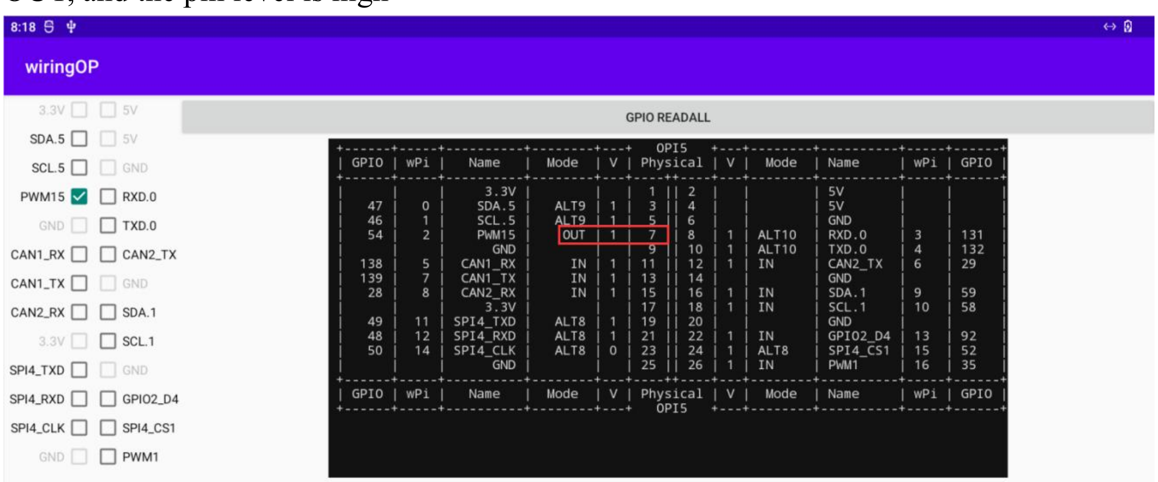
3) The GPIO test interface is shown in the figure below. The two rows of **CheckBox** buttons on the left are in one-to-one correspondence with the 26pin pins. When the **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT** mode, and the pin level will be set to high level; when the checkbox is unchecked, the GPIO pin level will be set to low level; When the **GPIO READALL** button is pressed, information such as the wPi number, GPIO mode, and pin level can be obtained.

4) Then click the **GPIO READALL** button, the output information is as shown in the figure below:



5) There are a total of 16 GPIO ports in the 26pins of the development board that can be used. The following takes pin 7—the corresponding GPIO is GPIO1_C6——the corresponding wPi serial number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port. First click the **CheckBox** button corresponding to pin 7. When the button is selected, pin 7 will be set to high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **3.3v**, it means setting high level success

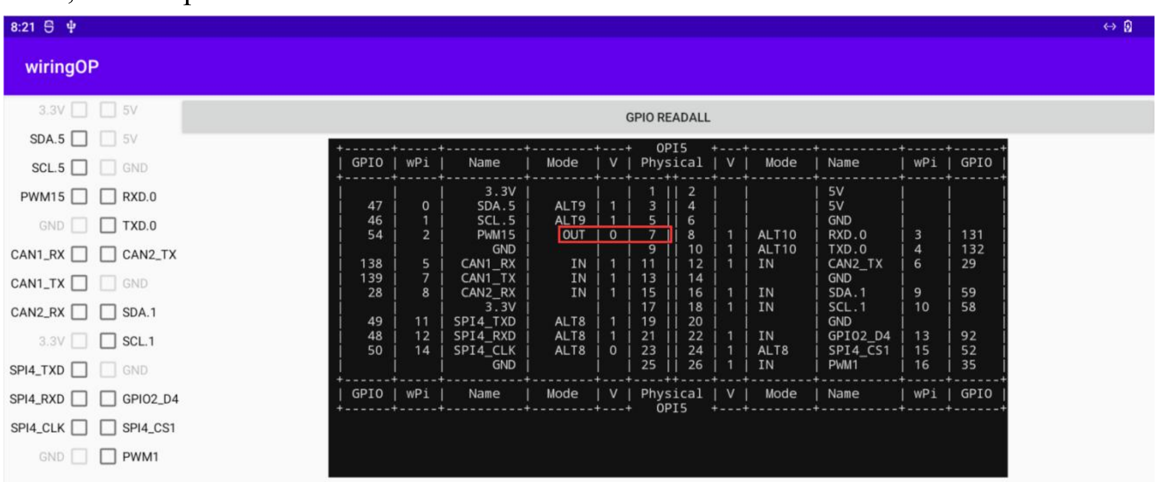
6) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is OUT, and the pin level is high



7) Click the **CheckBox** button in the figure below again to cancel the check status. Pin 7

will be set to low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level is set successfully.

8) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is OUT, and the pin level is low

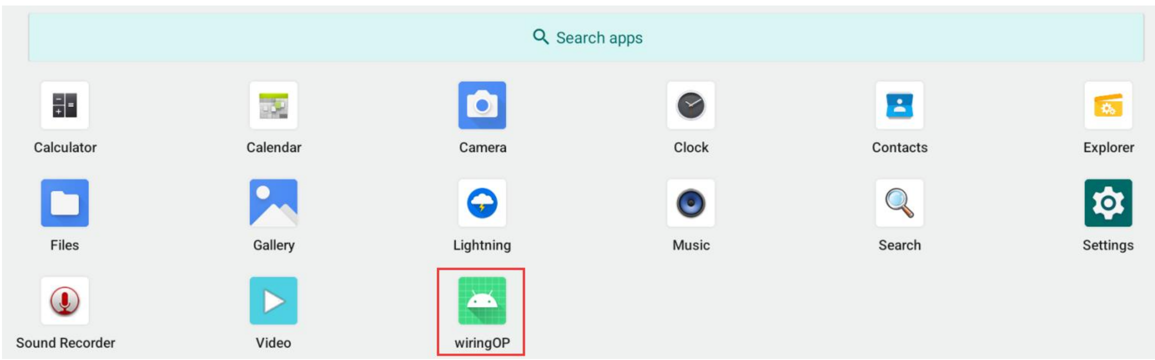


GPIO	wPi	Name	Mode	V	Physical	Physical	V	Mode	Name	wPi	GPIO
		3.3V			1	2			5V		
47	0	SDA.5	ALT9	1	3	4			5V		
46	1	SCL.5	ALT9	1	5	6			GND		
54	2	PWM15	OUT	0	7	8	1	ALT10	RXD.0	3	131
		GND			9	10	1	ALT10	TXD.0	4	132
138	5	CAN1_RX	IN	1	11	12	1	IN	CAN2_TX	6	29
139	7	CAN1_TX	IN	1	13	14			GND		
28	8	CAN2_RX	IN	1	15	16	1	IN	SDA.1	9	59
		3.3V			17	18	1	IN	SCL.1	10	58
49	11	SPI4_TXD	ALT8	1	19	20			GND		
48	12	SPI4_RXD	ALT8	1	21	22	1	IN	GPIO2_D4	13	92
50	14	SPI4_CLK	ALT8	0	23	24	1	ALT8	SPI4_CS1	15	52
		GND			25	26	1	IN	PWM1	16	35
GPIO	wPi	Name	Mode	V	Physical	Physical	V	Mode	Name	wPi	GPIO

8. 10. 2. 26pin UART test

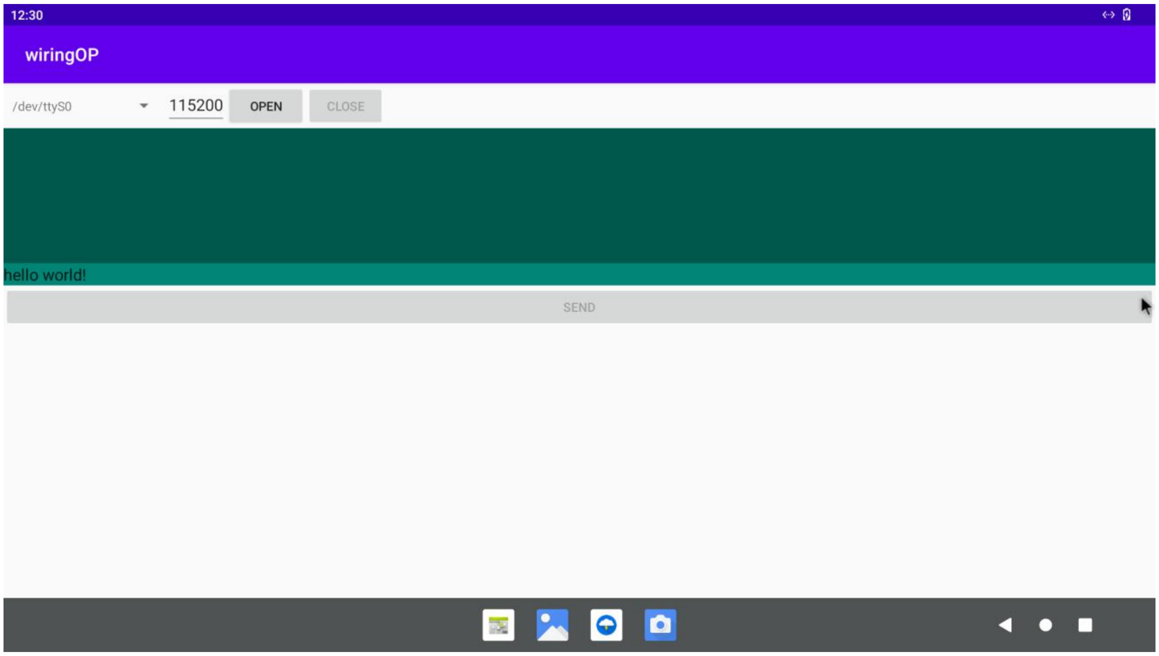
1) In Android, only one serial port of UART0 is opened by default. The position of UART0 at 26pin is shown in the figure below, and the corresponding device node is **/dev/ttyS0**

2) First click on the wiringOP icon to open the wiringOP APP

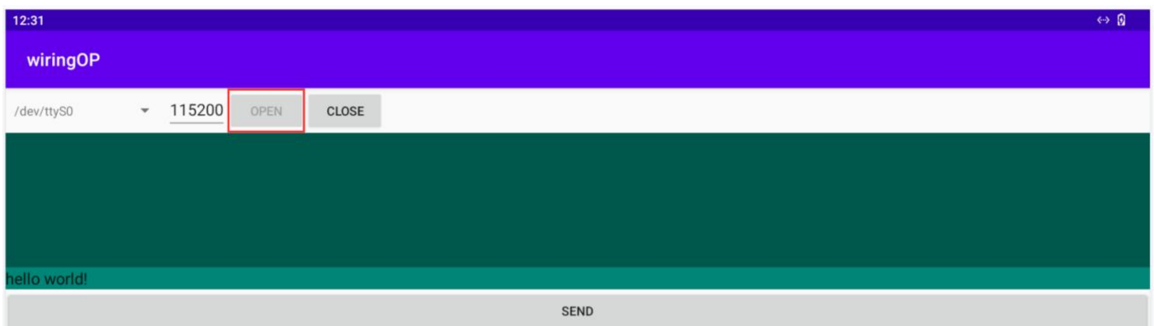


3) The main interface of wiringOP APP is displayed as shown in the figure below, and then click the **UART_TEST** button to open the UART test interface

4) The serial port test interface of the APP is shown in the figure below

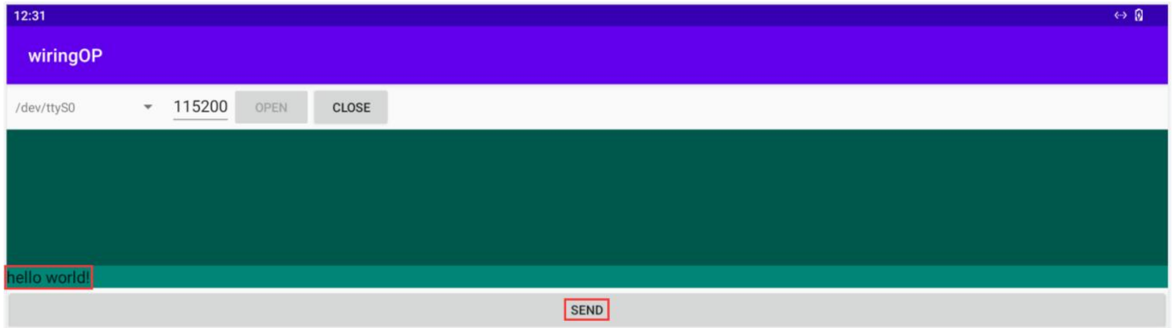


5) Then enter the baud rate you want to set in the edit box, and then click the **OPEN** button to open the /dev/ttyS0 node. After the opening is successful, the **OPEN** button becomes unselectable, and the **CLOSE** button and **SEND** button become selectable.

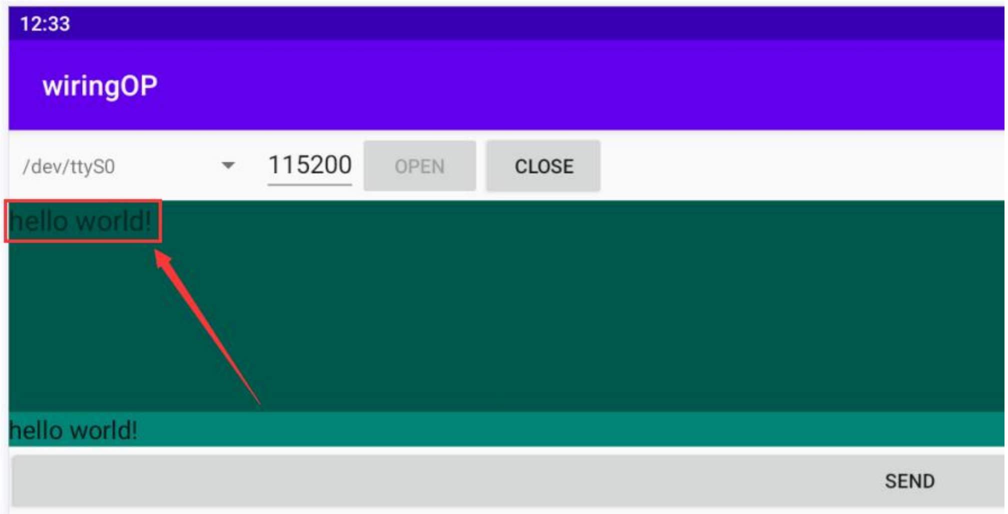


6) Then use Dupont wire to short the RXD and TXD pins of uart0

7) Then you can enter a character in the send edit box below, and click the **SEND** button to start sending

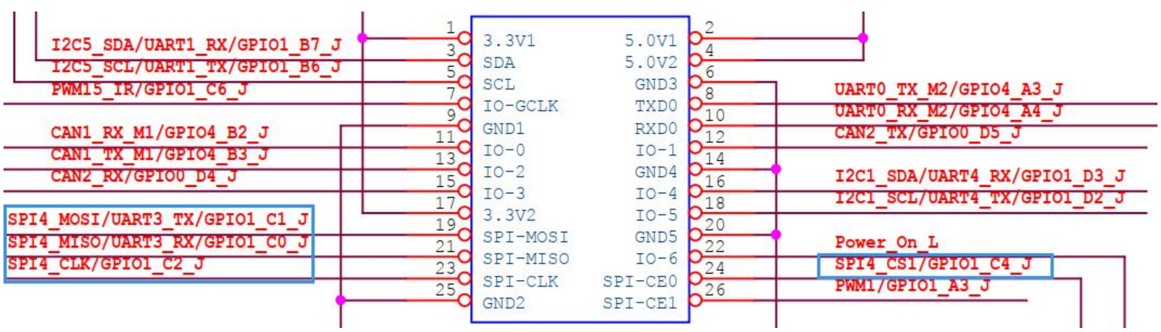


8) If everything is normal, the received string will be displayed in the receiving box



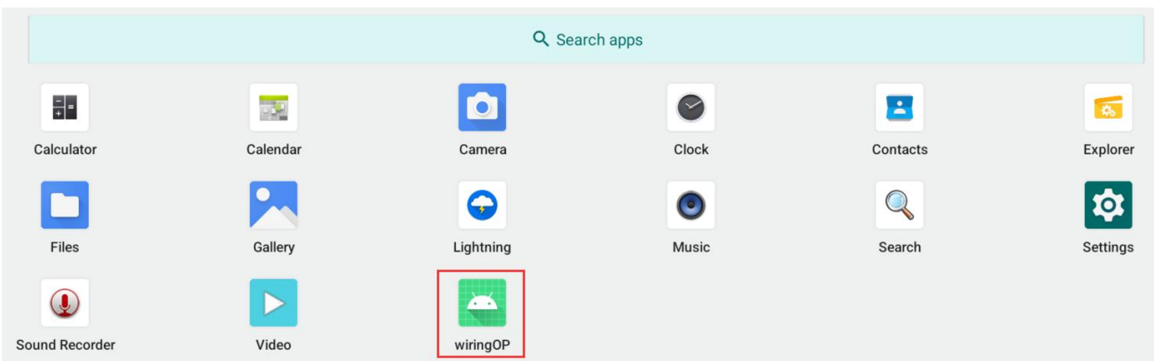
8. 10. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the spi available for Orange Pi 5B is spi4



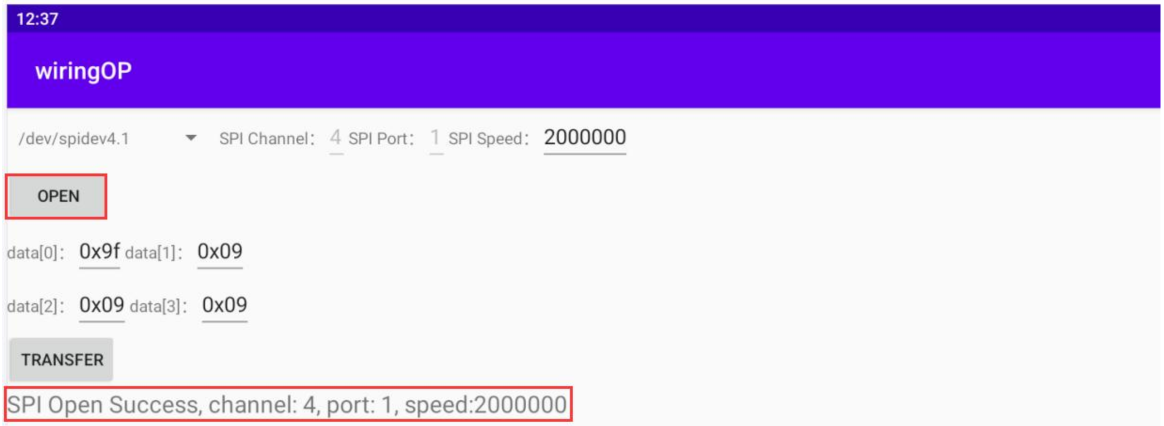
2) Here, the SPI interface is tested through the w25q64 module. First, the w25q64 device is connected to the SPI4 interface

3) Then click the wiringOP icon to open the wiringOP APP

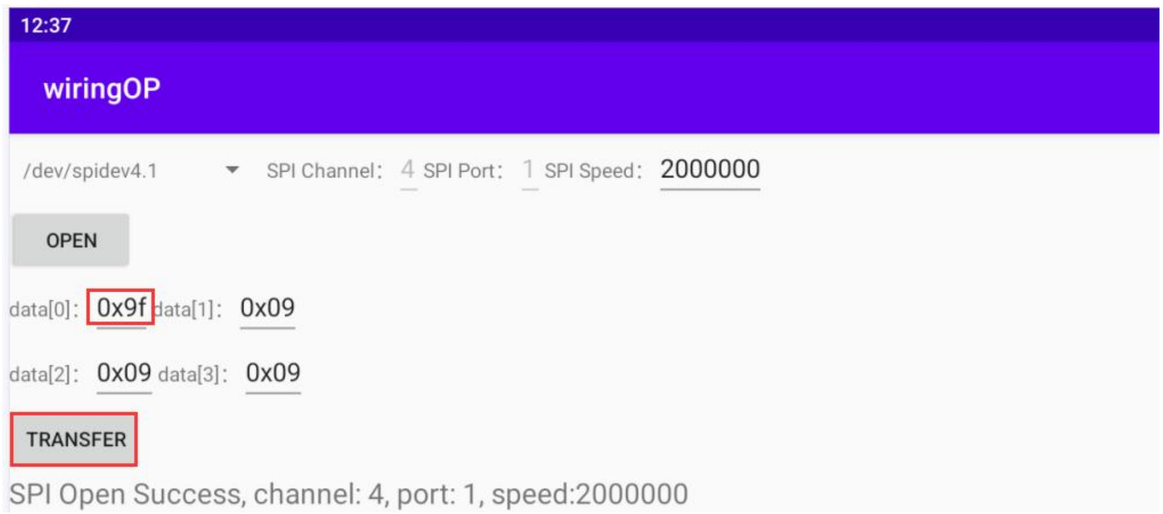


4) The main interface of wiringOP APP is displayed as shown in the figure below, click the SPI_TEST button to open the SPI test interface

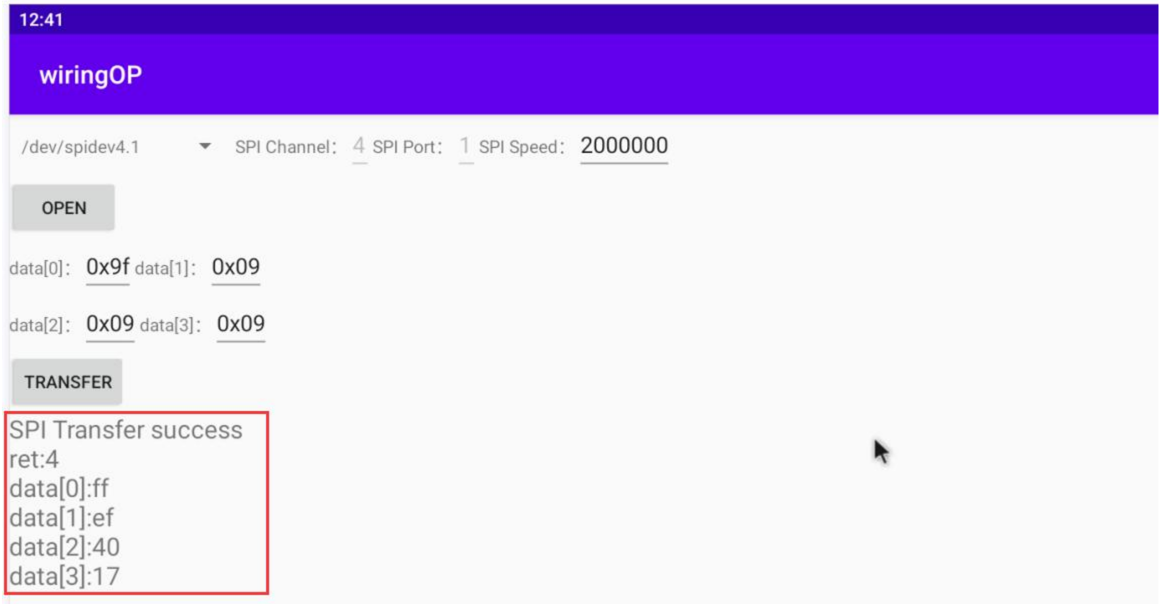
5) Then click the **OPEN** button to initialize the SPI



6) Then fill in the bytes that need to be sent, such as reading the ID information of w25q64, fill in the address 0x9f in data[0], and then click the **TRANSFER** button



7) Finally, the APP will display the read ID information

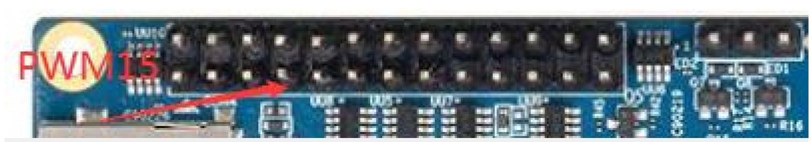


8) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, corresponding to the value read above (h stands for hexadecimal)

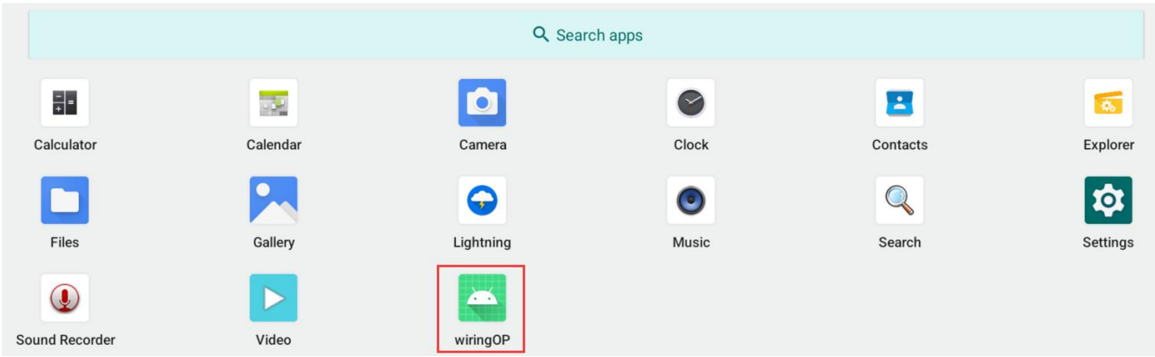
MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

8. 10. 4. 26pin PWM test

1) Android only enables **PWM15** by default, and the corresponding pin is located at 26pin as shown in the figure below

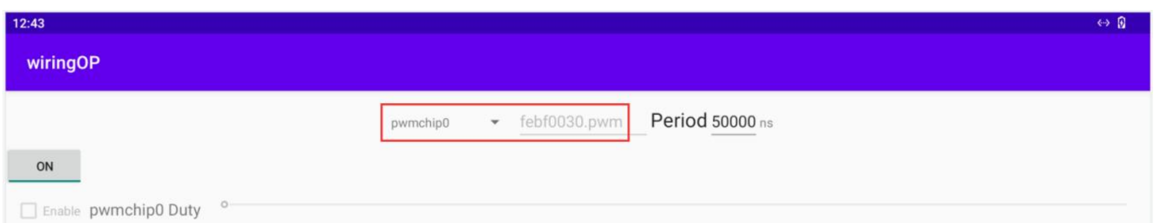


2) First click on the wiringOP icon to open the wiringOP APP



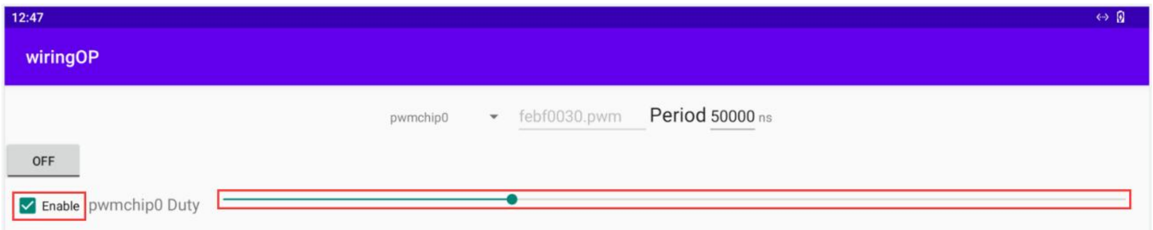
3) Then click the **PWM_TEST** button on the main interface of wiringOP to enter the PWM test interface

4) The base address corresponding to PWM15 is **febf0030**, and the right side of pwmchip0 is just **febf0030.pwm**, if the displayed base address is wrong, please click the drop-down option to select other pwmchips until **febf0030** is displayed on the right

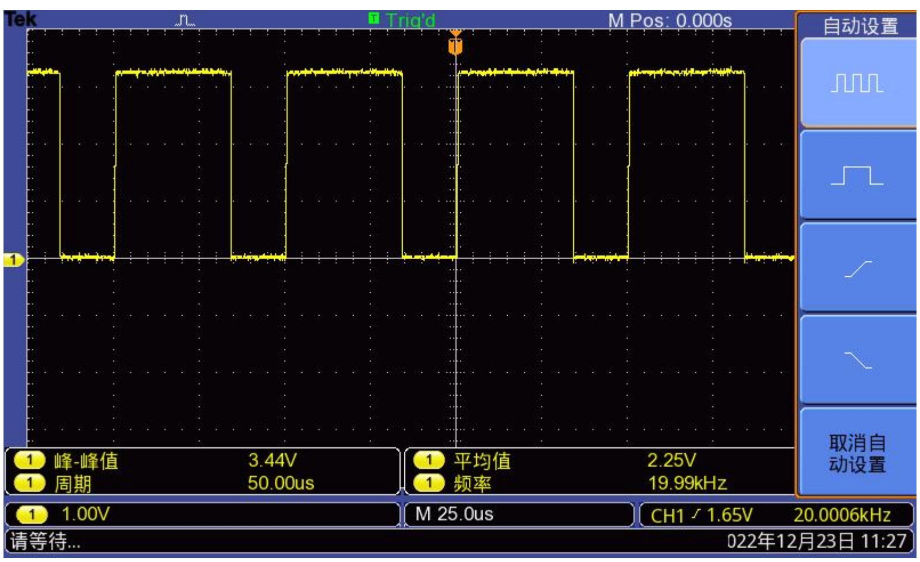


5) Then confirm the PWM cycle, the default configuration is **50000ns**, converted to PWM frequency is **20KHz**, you can modify it yourself, click the open button to export **PWM15**

6) Then drag the drag bar below to change the PWM duty cycle, and then check Enable to output the PWM waveform



7) Then use an oscilloscope to measure pin 7 in the 26pin of the development board, and you can see the following waveform



8. 11. How to use ADB

8. 11. 1. Use the data cable to connect to adb debugging

4) First prepare a good quality Type-C data cable

5) Then use the Type-C data cable to connect the development board to the USB interface of the computer (please use the Type-C power supply to power the development board at the same time)

6) Install adb tool on Ubuntu PC

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

7) You can view the identified ADB devices through the following command

```
test@test:~$ adb devices
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

8) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell
console:/ $
```

9) Execute the command to remount the Android system

```
test@test:~$ adb root
test@test:~$ adb remount
```

10) Then you can transfer files to the Android system

```
test@test:~$ adb push example.txt /system/
```

8. 11. 2. Use network connection adb debugging

Using the network adb does not require a USB Type C interface data cable to

connect the computer and the development board, but to communicate through the network, so first make sure that the wired or wireless network of the development board has been connected, and then obtain the IP address of the development board, and then to use.

1) Make sure that the **service.adb.tcp.port** of the Android system is set to port number 5555

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

4) Then connect network adb on Ubuntu PC

```
test@test:~$ adb connect 192.168.1.xxx     (The IP address needs to be changed to the IP address of the development board)
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xxx:5555

test@test:~$ adb devices
List of devices attached
192.168.1.xxx:5555          device
```

5) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell
console:/ #
```

8. 12. 2.4G USB remote control tested by Android Box

7) A 2.4G USB remote control that has been tested so far is shown in the figure below

a. Contains a remote control

b. A USB wireless receiver

8) The Android Box system does not require any configuration, it can be used after plugging it in

8. 13. How to use HDMI CEC function in Android Box system

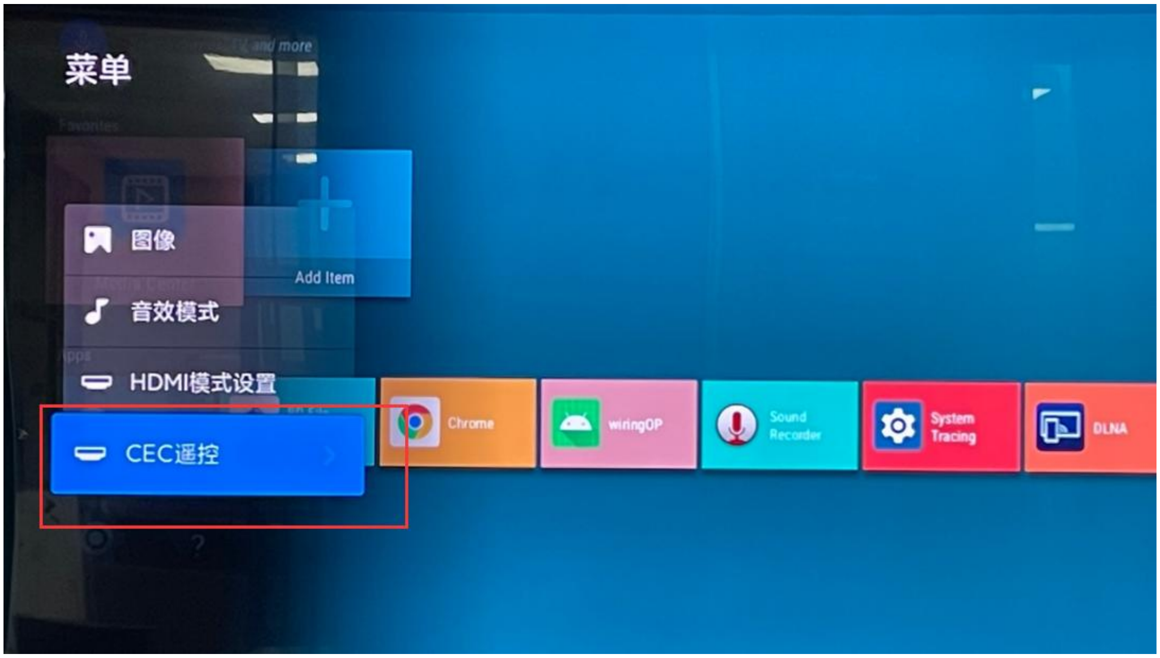
HDMI CEC allows users to control all connected devices through HDMI with only one remote control. Based on this function, we can control the development board with the remote control of the TV.

Before testing this function, please make sure your TV supports HDMI CEC.

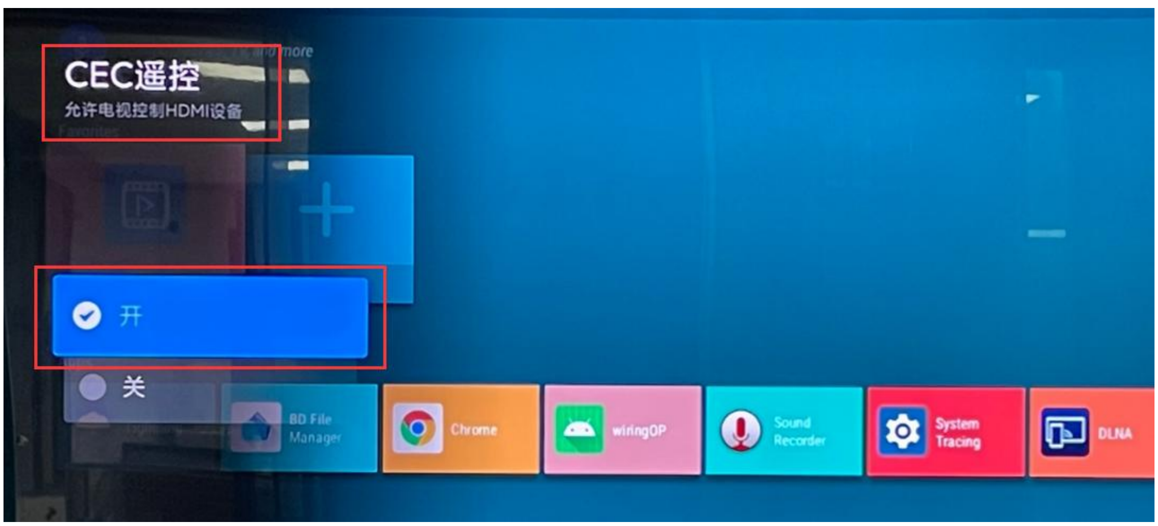
1) First connect the development board to the TV through the HDMI cable, then power on and start

2) Then turn on the HDMI CEC function in the TV settings. Different TVs may have

different ways to turn it on. Here we take Xiaomi TV as an example. Press the menu button on the remote control, then select CEC remote control and press the confirmation button



3) Then select **"On"** to open the HDMI CEC remote control



4) At this point, you can control the Android Box system of the development board through the remote control of the TV

9. How to compile Android 12 source code

9. 1. Download the source code of Android 12

1) First download the Android 12 source code sub-volume compressed package from Google Drive

a. Google Drive

2) After downloading the sub-volume compression package of the Android 12 source code, please check whether the MD5 checksum is correct, if not, please download the source code again

```
test@test:~$ md5sum -c Android_12.tar.gz.md5sum
Android_12.tar.gz00: OK
Android_12.tar.gz01: OK
```

```
Android_12.tar.gz02: OK
Android_12.tar.gz03: OK
Android_12.tar.gz04: OK
Android_12.tar.gz05: OK
Android_12.tar.gz06: OK
Android_12.tar.gz07: OK
```

3) Then you need to merge multiple compressed files into one, and then decompress

```
test@test:~$ cat Android_12.tar.gz0* > Android_12.tar.gz
test@test:~$ tar -xvf Android_12.tar.gz
```

9. 2. Compile the source code of Android 12

1) First install the software packages required to compile the Android12 source code

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y git gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip
test@test:~$ sudo apt-get install -y u-boot-tools
```

2) There is a build.sh compilation script in the source code, and the compilation parameters are as follows

c. **-U**: compile uboot
d. **-K**: compile kernel
e. **-A**: compile android
f. **-u**: Package and generate update.img and update_spi_nvme.img
g. **-o**: Compile the OTA package
h. **-d**: specify kernel dts

3) Compile uboot, kernel, android and package them into update.img

a. The command to compile and support HDMI 8K display mirroring (LCD off by default) is as follows:

```
test@test:~$ cd Android_12
test@test:~/ Android_12$ source build/envsetup.sh
```

```
test@test:~/ Android_12$ lunch rk3588s_s-userdebug
test@test:~/ Android_12$ ./build.sh -AUKu
```

b. The command to compile and support LCD display image is as follows：

```
test@test:~$ cd Android_12
test@test:~/ Android_12$ export DUAL_LCD=true
test@test:~/ Android_12$ source build/envsetup.sh
test@test:~/ Android_12$ lunch rk3588s_s-userdebug
test@test:~/ Android_12$ ./build.sh -AUKu
```

c. The command to compile and support sata boot image is as follows：

```
test@test:~$ cd Android_12
test@test:~/ Android_12$ export BOOT_DEVICE=spi-sata
test@test:~/ Android_12$ source build/envsetup.sh
test@test:~/ Android_12$ lunch rk3588s_s-userdebug
test@test:~/ Android_12$ ./build.sh -AUKu
```

4) After the compilation is complete, the following information will be printed

```
********rkImageMaker ver 2.1********
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
Making update.img OK.
Make update image ok!
```

5) The final image file will be placed in the **rockdev/Image-rk3588s_s** directory. Among them, **update.img** is the TF card boot image, **update_spi_nvme.img** is the NVME SSD boot image

```
test@test:~/Android_12$ cd rockdev/Image-rk3588s_s
test@test:~/Android_12/rockdev/Image-rk3588s_s $ ls update*
update.img update_spi_nvme.img
```

6) If you are compiling an image that supports sata startup, the image name is **update_spi_sata.img**

```
test@test:~/Android_12$ cd rockdev/Image-rk3588s_s
test@test:~/Android_12/rockdev/Image-rk3588s_s $ ls update*
update_spi_sata.img
```

10. OpenWRT system instructions

10. 1. OpenWRT version

OpenWRT version	kernel version
v22.03.4	Linux5.10.110

10. 2. OpenWRT Adaptation

Function	OpenWRT
USB2.0x2	OK
USB3.0x1	OK
USB Type-C 3.0	OK
3pin debugging serial port	OK
TF card start	OK
Gigabit Ethernet port	OK
Network port status light	OK
LED light	OK
USB to wired network port	OK
RTL8821CU USB network card	OK
RTL8723BU USB network card	OK
RTL8811 USB network card	OK

M.2 NVMe SSD boot	**OK**
M.2 SATA SSD boot	**OK**
AP6275P-WIFI	**NO**

10. 3. Expand the rootfs in the TF card before the first startup

1) When the TF card starts the OpenWRT system for the first time, the **resize-rootfs.sh** script will be executed to expand the rootfs, and it will automatically restart after the expansion is completed

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

```
root@OpenWrt:~# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                14.8G     14.7G     91.6M  99% /
tmpfs                   495.5M      6.1M    489.4M   1% /tmp
tmpfs                   512.0K         0    512.0K   0% /dev
/dev/root                14.8G     14.7G     91.6M  99% /opt/docker
```

10. 4. How to log in to the system

10. 4. 1. Login via serial port

1) First, to use the debugging serial port, please refer to the chapter on how to use the debugging serial port

2) The OpenWrt system will automatically log in as the **root** user by default, and the display interface is as follows

```
BusyBox v1.33.1 (2021-10-24 09:01:35 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 21.02.1, r16325-88151b8303
 -----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@OpenWrt:/#
```

10. 4. 2. Log in to the system via SSH

Please note that in the OpenWrt system of Orange Pi 5, the network port is configured as a LAN port by default, so the LAN port of the development board needs to be directly connected to the network port of the computer. If it is connected to a router, there is no way to obtain the IP.

1) First connect the LAN port of the board to the network port of the computer with a network cable, so that the network port of the computer can obtain an IP address through DHCP

2) The LAN port IP of the default board is set to **192.168.2.1**, so the computer can obtain the IP address starting with **192.168.2** at this time

3) If the computer is installed with an Ubuntu system, you can execute the following command to log in to the system through SSH. By default, you can log in directly without a password

```
test@ubuntu:~$ ssh root@192.168.2.1
```

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

```
orangepi:~$ ssh root@192.168.2.1


BusyBox v1.35.0 (2023-04-09 12:27:46 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 22.03.4, r20123-38ccc47687
 -----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@OpenWrt:~#
```

5) If the computer is installed with Windows system, you can log in by referring to the method of SSH remote login to the development board under Windows in the Linux system instruction manual

10. 4. 3. Log in to the LuCI management interface

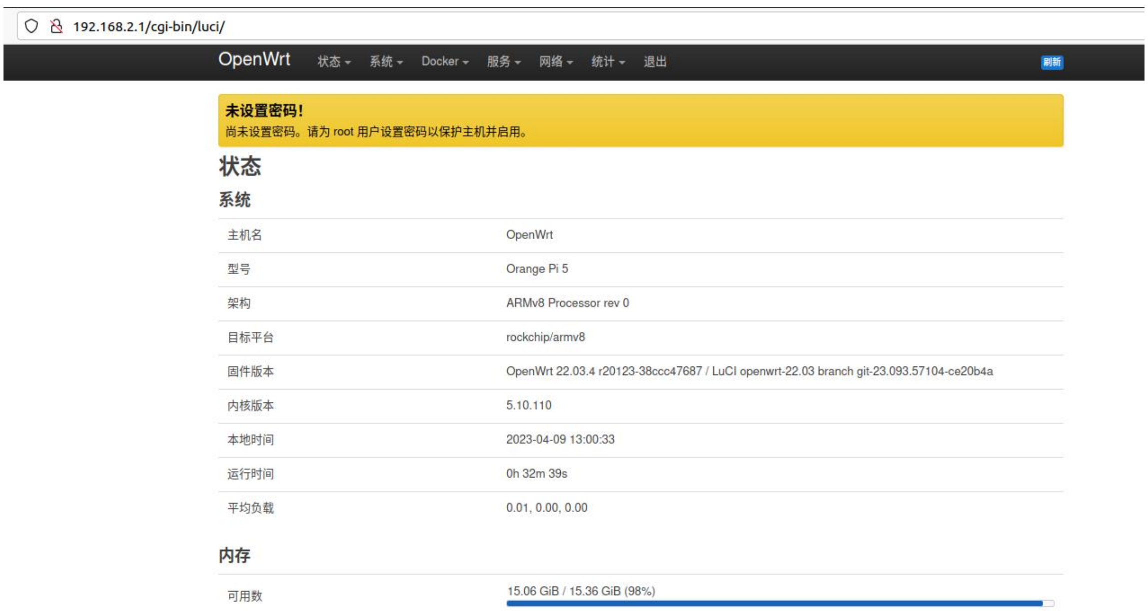
Please note that in the OpenWrt system of Orange Pi 5, the network port is configured as a LAN port by default, so the LAN port of the development board needs to be directly connected to the network port of the computer. If it is connected to a router, there is no way to obtain the IP.

1) First connect the LAN port of the board to the network port of the computer with a network cable, so that the network port of the computer can obtain an IP address through DHCP

2) The LAN port IP of the default board is set to **192.168.2.1**, so the computer can obtain the IP address starting with **192.168.2** at this time

3) Enter the IP address **192.168.2.1** in the browser on the computer to log in to the LuCI interface

4) The OpenWrt system does not set a password by default, so just click the **login** button. After successful login, the interface is displayed as shown in the figure below



10. 4. 4. Log in to the terminal through the LuCI management interface

Please note that in the OpenWrt system of Orange Pi 5, the network port is configured as a LAN port by default, so the LAN port of the development board needs to be directly connected to the network port of the computer. If it is connected to a router, there is no way to obtain the IP.

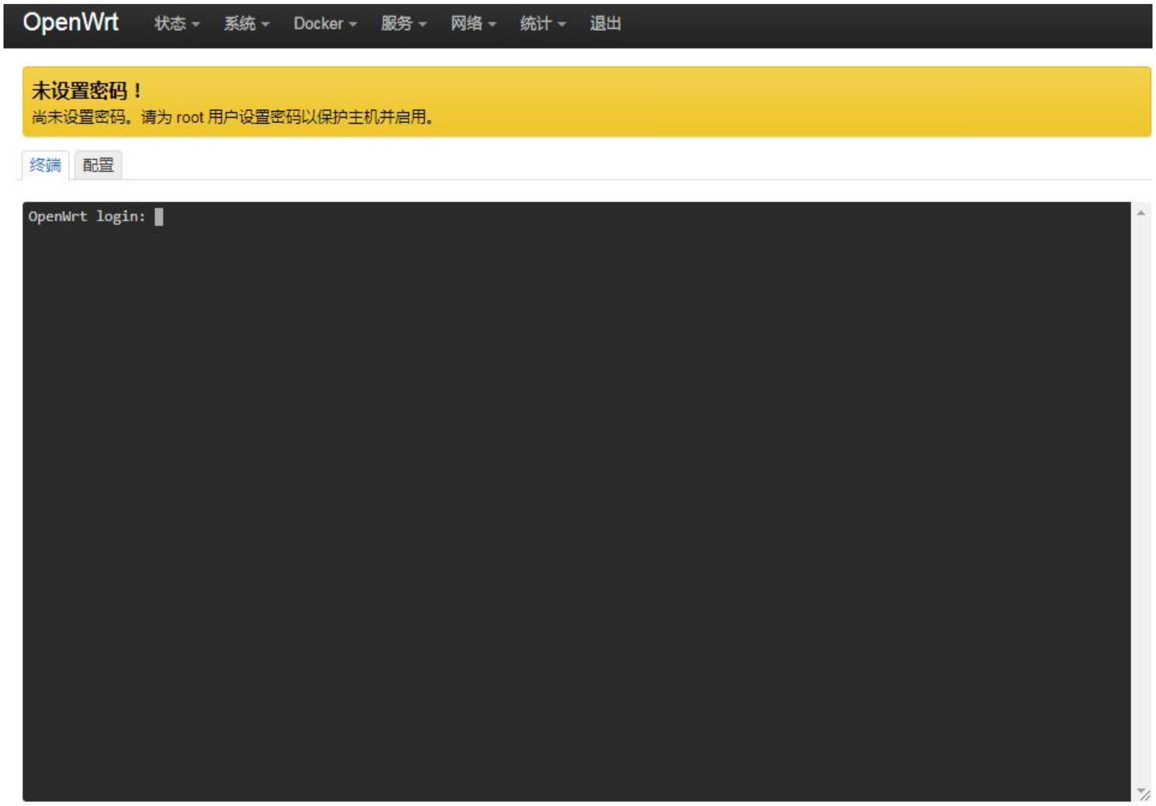
1) First connect the LAN port of the board to the network port of the computer with a network cable, so that the network port of the computer can obtain an IP address through DHCP

2) The LAN port IP of the default board is set to **192.168.2.1**, so the computer can obtain the IP address starting with **192.168.2** at this time

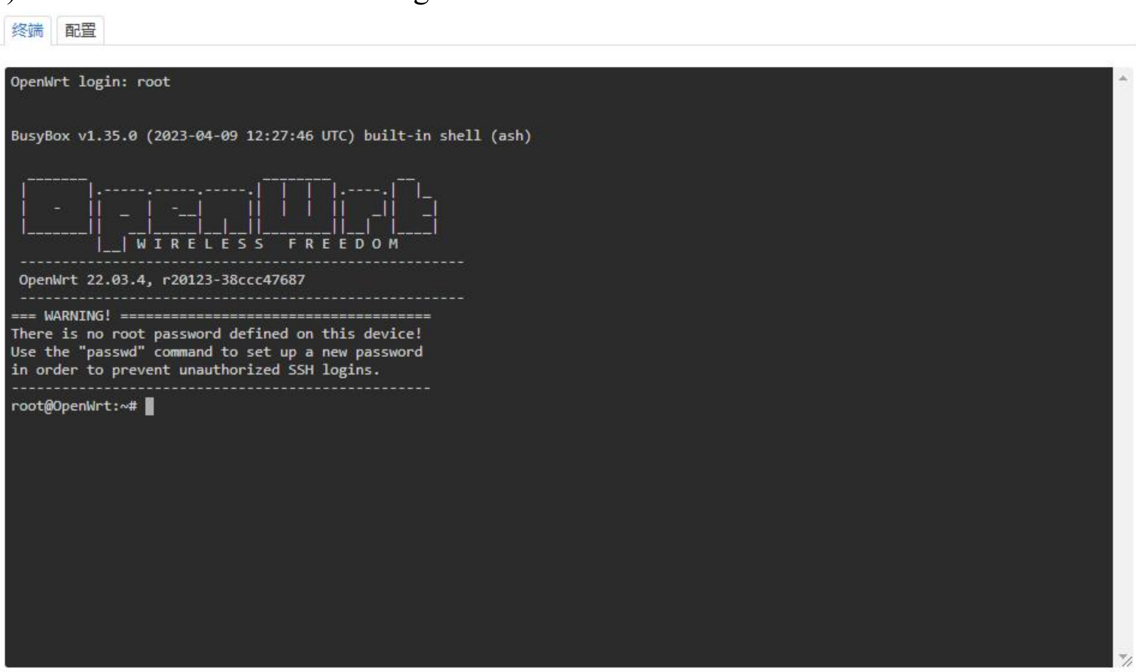
3) Enter the IP address **192.168.2.1** in the browser on the computer to log in to the LuCI interface

4) Select **"Terminal"** in the **"Service"** column of the navigation bar and click to enter

5) At this time, the terminal interface is as shown in the figure below



6) Enter the user name root to log in



10. 4. 5. Use IP address + port number to log in to the terminal

Please note that in the OpenWrt system of Orange Pi 5, the network port is configured as the LAN function by default, so the network port of the development board cannot be directly connected to the router through a network cable, but can only be directly connected to the network port of the computer through a network cable. At this time, the system starts Afterwards, an IP address will be assigned to the network port of the computer through the DHCP service.

1) First connect the LAN port of the board to the network port of the computer with a network cable, so that the network port of the computer can obtain an IP address through DHCP

2) The LAN port IP of the default board is set to **192.168.2.1**, so the computer can obtain the IP address starting with **192.168.2** at this time

3) Then enter **192.168.2.1:7681** in the browser to log in to the OpenWRT terminal

10. 5. How to modify the IP address of the LAN port through the command line

1) In the OpenWrt system, a command line tool uci is provided, which can easily modify, add, delete and read the content in the configuration file. For details, please refer to the

official document

2) First use the following command to obtain the network configuration, the corresponding configuration file is **/etc/config/network**, you can see that the value of **network.lan.ipaddr** is **192.168.2.1**

```
root@OpenWrt:~# uci show network
...
network.lan=interface
network.lan.device='br-lan'
network.lan.proto='static'
network.lan.ipaddr='192.168.2.1'
network.lan.netmask='255.255.255.0'
network.lan.ip6assign='60'
....
```

3) Then enter the following command to modify the item **network.lan.ipaddr**

```
root@OpenWrt:~# uci set network.lan.ipaddr='192.168.100.1'
```

4) Then enter the following command to complete the submission, that is, write to the configuration file

```
root@OpenWrt:~# uci commit
```

If the IP address in red font is consistent with the one to be set, it means that the modification is successful

```
root@OpenWrt:~# cat /etc/config/network
...
config interface 'lan'
        option device 'br-lan'
        option proto 'static'
        option netmask '255.255.255.0'
        option ip6assign '60'
        option ipaddr '192.168.100.1'
...
```

5) Restart the network through ubus, please refer to the official document for the usage

instructions of ubus

```
root@OpenWrt:~# ubus call network restart
```

6) At this point, enter the command and you can see that the IP of the LAN port is **192.168.100.1**

```
root@OpenWrt:~# ifconfig br-lan
br-lan      Link encap:Ethernet    HWaddr FE:55:13:A3:EF:E7
            inet addr:192.168.100.1    Bcast:192.168.100.255    Mask:255.255.255.0
            inet6 addr: fd60:c4cd:1033::1/60 Scope:Global
            UP BROADCAST MULTICAST    MTU:1500    Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)    TX bytes:370 (370.0 B)
```

10. 6. How to modify the root password

10. 6. 1. Modify via command line

1) First enter passwd root on the command line of the system, and the following prompt message will appear. At this time, you can enter the password you want to set, and press the Enter key to confirm

```
root@OpenWrt:/# passwd root
Enter new UNIX password:
```

2) Then you will be prompted to re-enter the password. At this time, enter the password again to confirm and press Enter

```
Retype password:
```

3) The display of successful modification is as follows

```
passwd: password for root changed by root
```

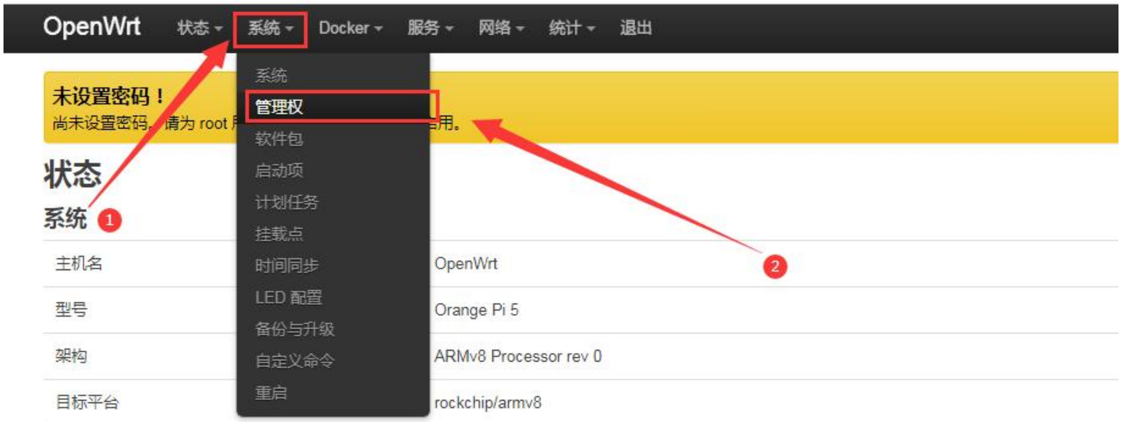
10. 6. 2. Modify through the LuCI management interface

1) First refer to the login LuCI management interface to enter the OpenWRT management interface

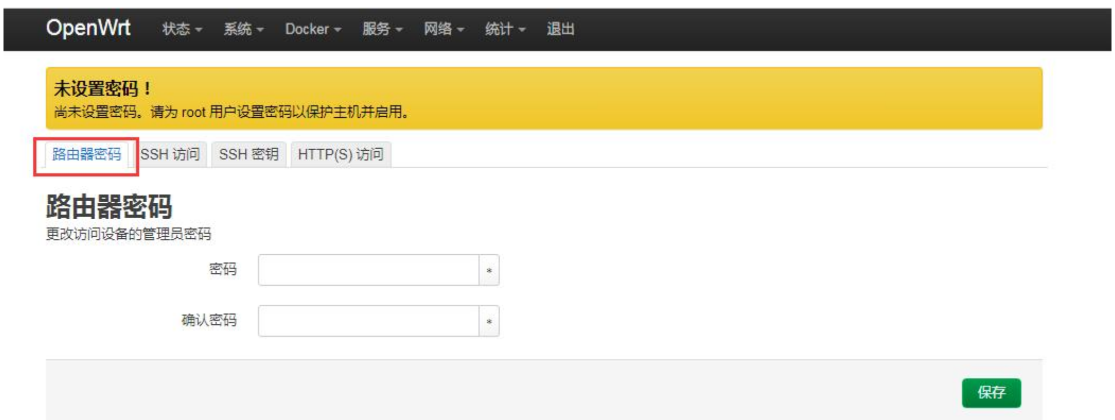
2) Then follow the steps below to change the password

a. Find the "**System**" option in the navigation bar and click

b. In the column options below the system, select "**Management Rights**" and click



c. Select the "**Router Password**" option on the Tab page



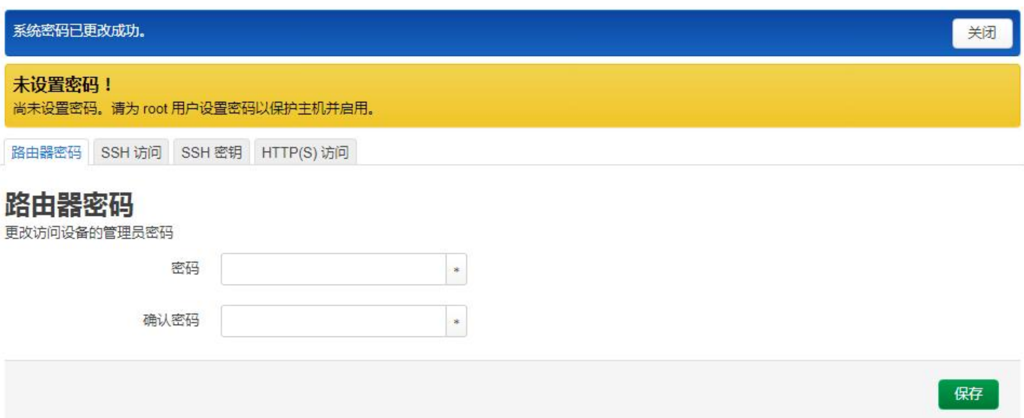
3) Modify and save the router password

a. Enter the password you set in the "**Password**" and "**Confirm Password**" dialog boxes (if you are not sure whether the password is entered correctly, you can click the "*" icon behind the dialog box to display the input characters)

b. Click "**Save**" to save the newly modified password

Note: In the "Password" and "Confirm Password" dialog boxes, the passwords entered twice must be consistent.

4) After the password is changed successfully, a pop-up box will pop up saying "**The system password has been changed successfully**". At this time, a password is required to log in to OpenWRT



10. 7. USB interface test

10. 7. 1. Mount the USB storage device under the command line

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

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

```
root@OpenWrt:~# cat /proc/partitions | grep "sd*"
major minor    #blocks    name
   8          0      15126528 sda
```

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

```
root@OpenWrt:~# mount /dev/sda /mnt/
root@OpenWrt:~# ls /mnt/
test.txt
```

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

```
root@OpenWrt:~# df -h | grep "sd"
/dev/sda              14.4G       187.2M         14.2G      1% /mnt
```

10. 7. 2. Mount the USB storage device on the LuCI management interface

1) First connect the U disk (or other storage device) to the development board via USB2.0

2) Then follow the login LuCI management interface to enter the LuCI management interface

3) Then in the LuCI management interface, click "System -> Mount Point" to enter the configuration interface of the mount point

4) Then follow the steps below to add a mount point

a. Find "**Mount Point**" at the bottom of the mount point **global setting interface**

b. Under the **mount point**, select the "Add" button and click Enter

c. Then the following pop-up interface will pop up

d. Then you can start to mount the storage device

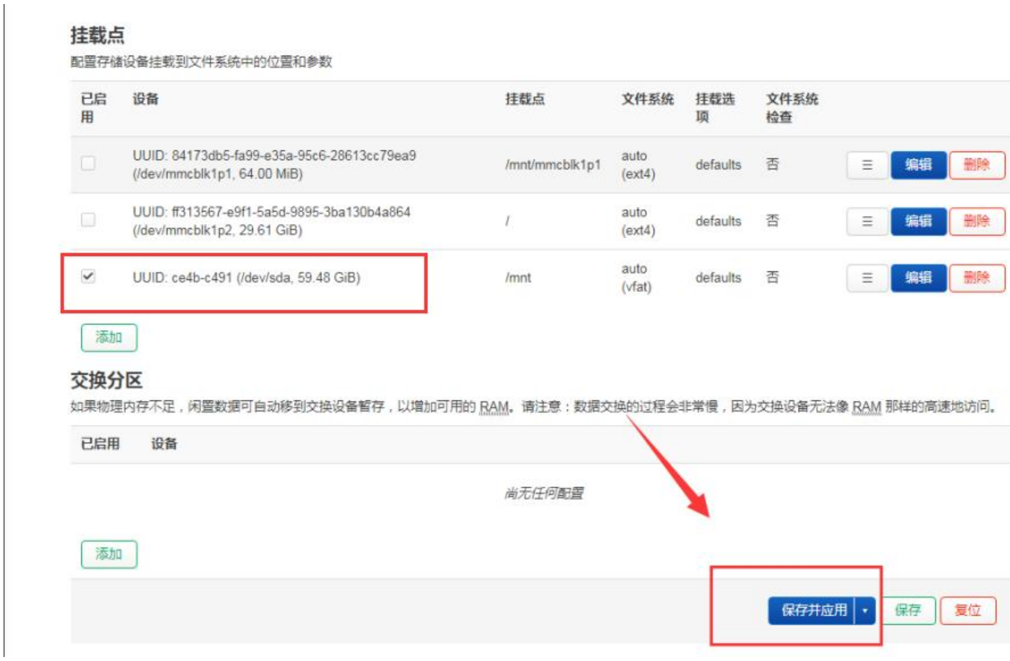
a) Check "**Enabled**"

b) Select the actual connected device /dev/sda in the UUID column of general settings (choose according to your own device)

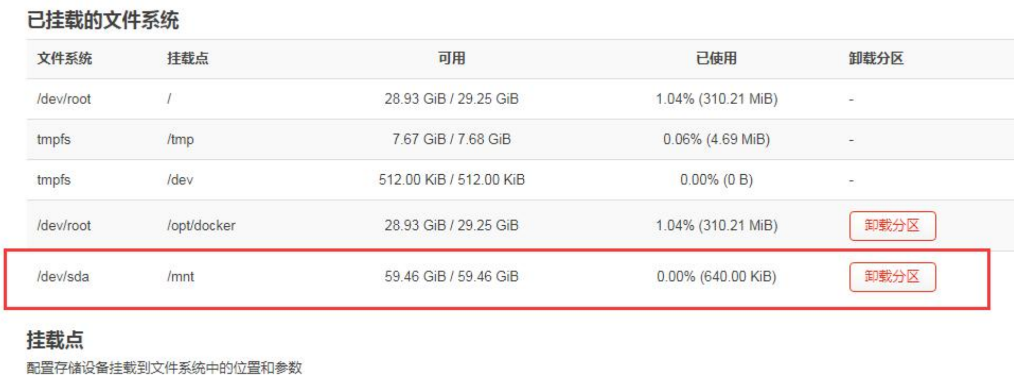
c) Select "**Custom**" in the mount point column, and fill in the target directory to be mounted. Here, take the **/mnt** directory as an example, and press **Enter** to confirm

d) Then click the "**Save**" button in the lower right corner

5) Then you will return to the mount point global settings page, click "Save and Apply" in the lower left corner of the page to make the mount point take effect



6) After saving, you can see the "**mounted file system**", the storage device has been mounted successfully



已挂载的文件系统

文件系统	挂载点	可用	已使用	卸载分区
/dev/root	/	28.93 GiB / 29.25 GiB	1.04% (310.21 MiB)	-
tmpfs	/tmp	7.67 GiB / 7.68 GiB	0.06% (4.69 MiB)	-
tmpfs	/dev	512.00 KiB / 512.00 KiB	0.00% (0 B)	-
/dev/root	/opt/docker	28.93 GiB / 29.25 GiB	1.04% (310.21 MiB)	卸载分区
/dev/sda	/mnt	59.46 GiB / 59.46 GiB	0.00% (640.00 KiB)	卸载分区

挂载点

配置存储设备挂载到文件系统中的位置和参数

10. 8. USB to network port test

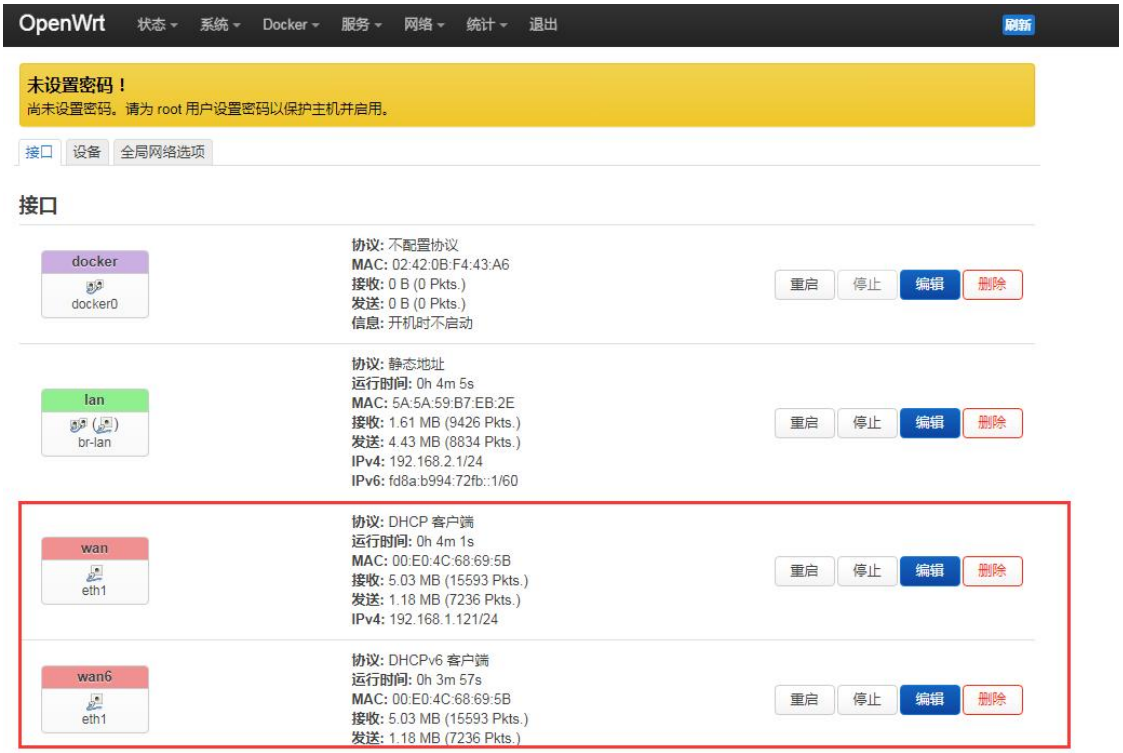
1）The usable USB port that has been tested so far is as follows

Chip model	VID&PID
RTL8153（Gigabit）	**0bda:8153**
RTL8152（100M）	**0bda:8152**

2）First insert the USB to network port module into the USB interface of the development board, and then power on and start the development board

3）Then enter the LuCI management interface according to the method of logging in to the LuCI management interface, and then click "Network -> Interface" to enter the wired network configuration interface

4）If you can see the “eth1” device as shown in the figure below in the configuration interface, it means that the USB port has been recognized. As can be seen from the figure below, “eth1” is configured as a WAN port by default



5）At this time, after the USB port is connected to the main router through the network cable, the IP address can be obtained automatically through DHCP, and then the development board and the computer connected to the LAN port of the development board can be connected to the Internet through the main router.

10. 9. USB wireless network card test

The usable USB wireless network cards that **have been tested** so far are as follows. Please test other types of USB wireless network cards by yourself. If they cannot be used, you need to transplant the corresponding USB wireless network card driver.

serial number	model	
1	RTL8723BU Support 2.4G WIFI	

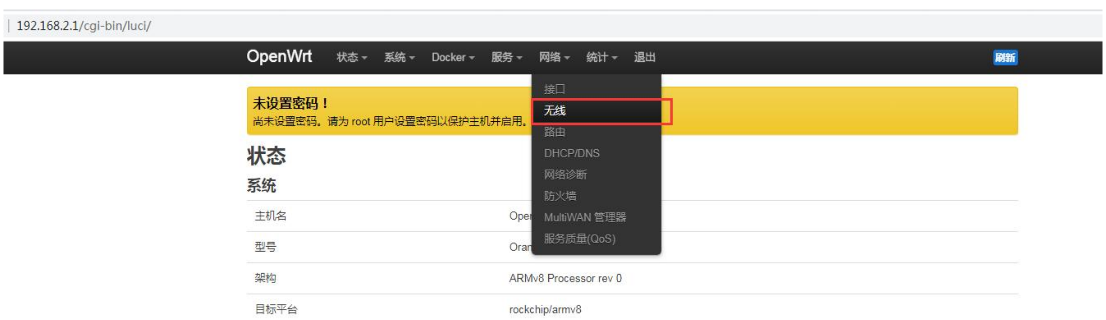
2	RTL8811 Support 2.4G +5G WIFI	
3	RTL8821CU Support 2.4G +5G WIFI	

10. 9. 1. How to create a WIFI hotspot using a USB wireless network card

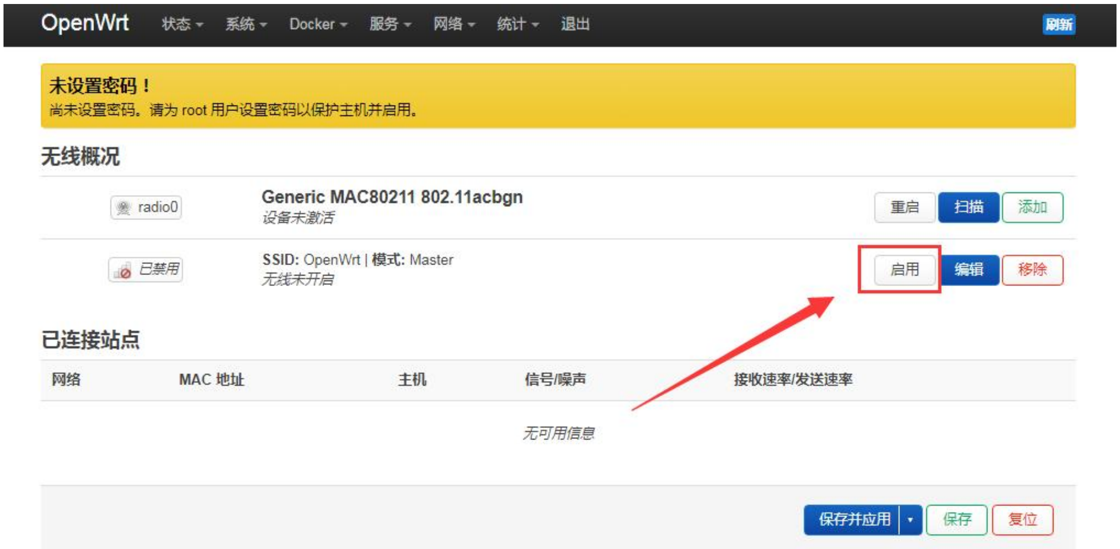
Note that the hotspot function is not supported on the RTL8723BU hardware, while the RTL8821CU and RTL8811 support the hotspot function.

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to the development board

2）After the system startup is complete, click "Network > Wireless" to enter the wireless WiFi configuration interface



3）The default wireless configuration of the OpenWRT system is the Master mode, here directly click **"Enable"**



4）The display interface of successfully creating a hotspot is shown in the figure below

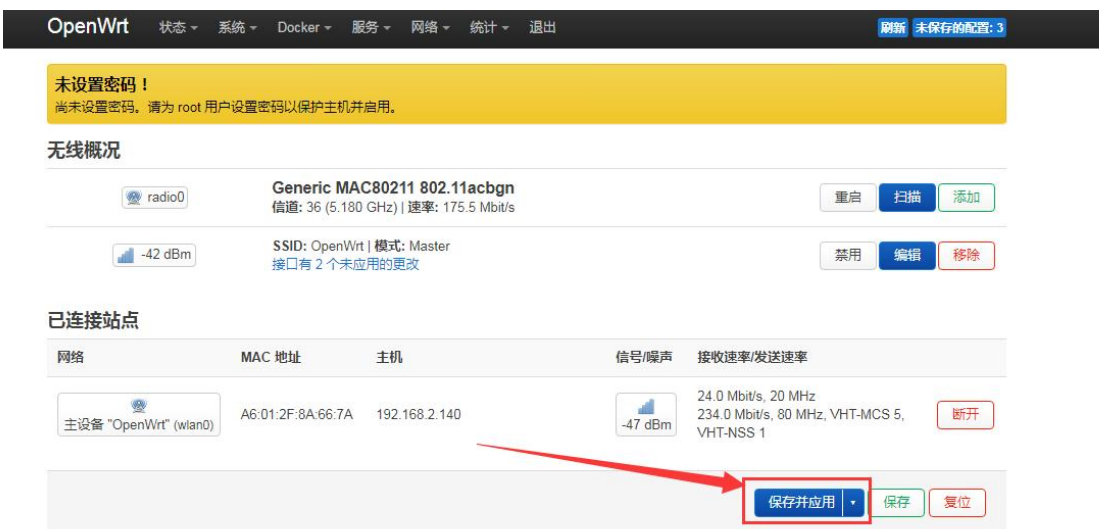
5）Then use the mobile phone or computer to search for the WiFi corresponding to the SSID to connect. After the connection is successful, as shown in the figure below

6）If you need to set a password for the created hotspot, click the "Edit" button below

7）Then click the drop-down box in the "Wireless Security" column and select the "WPA2-PSK" encryption method

8）Then enter the password you want to set in the "Key" column, and then click the "Save" button

9）Then click **"Save and Apply"** on the main interface of wireless settings



10）After the setting takes effect, you can see that the WiFi hotspot has been encrypted

10. 9. 2. How to use USB wireless network card to connect to WIFI hotspot

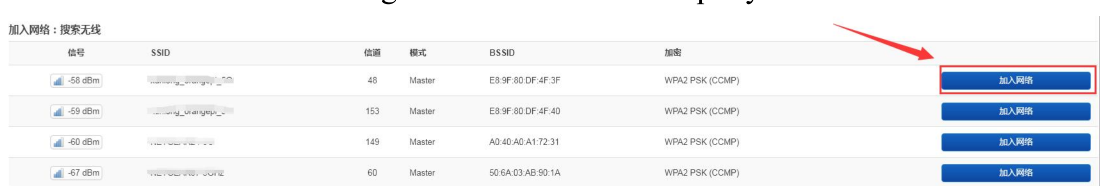
1) Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to the development board

2) After the system startup is complete, click "Network > Wireless" to enter the configuration wireless WiFi interface

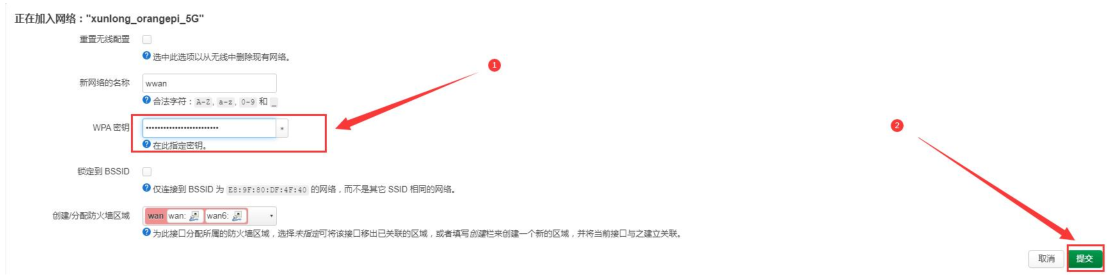
3) First, you need to remove the default wireless configuration, click the **"Remove"** button as shown in the figure below

4） Then click the "Scan" button to scan the surrounding WiFi hotspots

5） Then the following window will pop up to display available WiFi hotspots, click the "Join Network" button on the right side of the WiFi hotspot you want to connect to



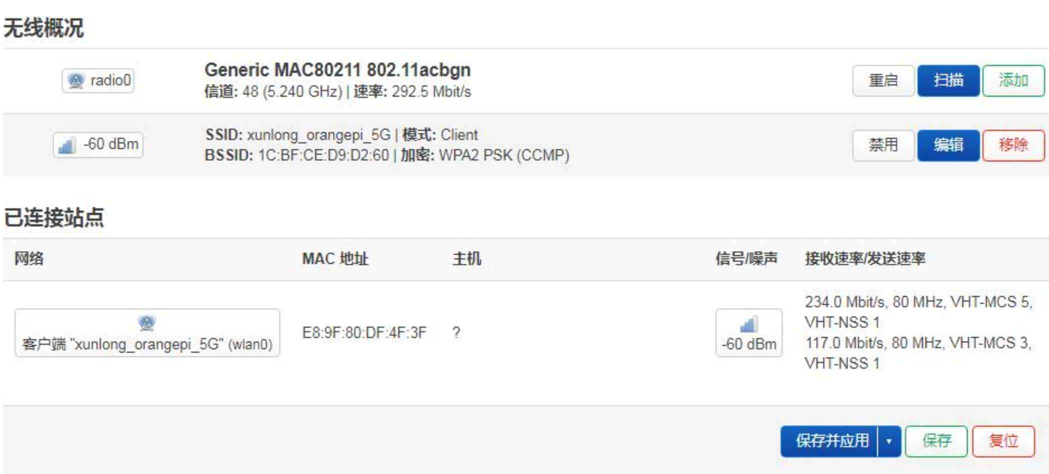
6） Then enter the password in the position shown in the figure below, and then click **"Submit"**



7）Then the following interface will pop up, just click Save

8）Finally, you will return to the main interface of wireless configuration, click "Save and Apply"

9）After successfully connecting to the WiFi hotspot, the interface is displayed as shown in the figure below



10. 10. Installing packages via the command line

10. 10. 1. Install via opkg in terminal

1) Update the list of available packages

```
root@OpenWrt:/# opkg update
```

2) Get the software list

```
root@OpenWrt:/# opkg list
```

3) Install the specified package

```
root@OpenWrt:/# opkg install <package name>
```

4) Check the installed software

```
root@OpenWrt:/# opkg list-installed
```

5) Uninstall the software

```
root@OpenWrt:/# opkg remove <package name>
```

10. 11. OpenWRT management interface installation software package

If you need to add new software packages, you can install them through the OpenWRT management interface

10. 11. 1. View the list of available software packages in the system

1) First enter the package management page

a. Find the **"System"** option in the navigation bar and click to enter

b. In the vertical column options below the system, select **"software package"** and click to enter

2) Then the main page of the software package will appear, as shown in the figure below, to obtain the list of available software

a. In the **"Operation"** option of the software package, click **"Update List"** to get the list of available software packages

b. On the Tab page, click **"Available"** to view the currently available software packages

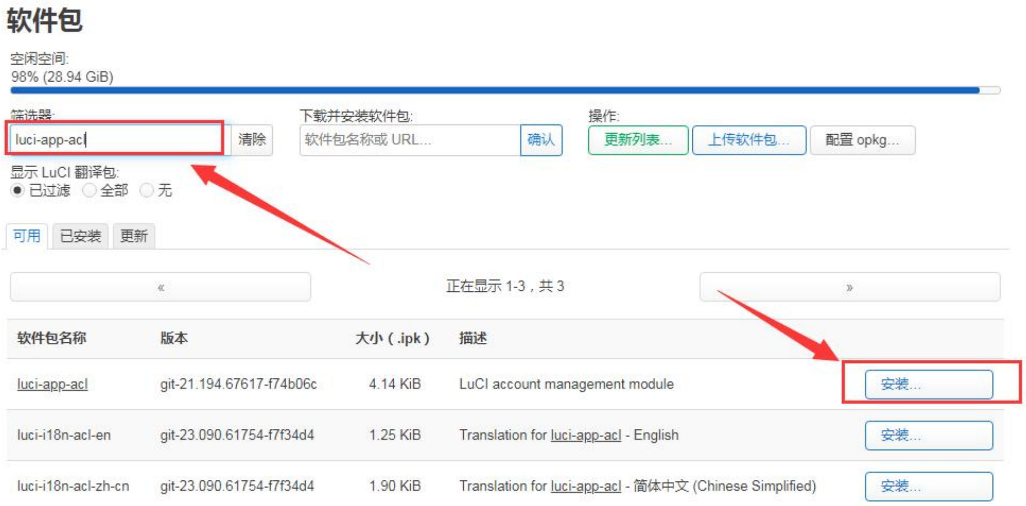
c. View the number of currently available packages

10. 11. 2. Example of installing software packages

1) Take the installation package “**luci-app-acl**” as an example

a. In the OpenWRT software package management interface, click the filter dialog box and enter"**luci-app-acl**"

b. In the list of software packages, you can see the version, package size and description information of the "**luci-app-acl**" software package, and then click the **"Install"** button



c. Then the following pop-up window will appear, click **"Install"**

d. Then wait for the installation to complete

e. The display after the installation is complete is as follows

正在执行软件包管理器

```
Installing luci-i18n-acl-en (git-23.090.61754-f7f34d4) to root...
Downloading
https://downloads.openwrt.org/releases/22.03.4/packages/aarch64_generic/luci/
luci-i18n-acl-en_git-23.090.61754-f7f34d4_all.ipk
Installing luci-app-acl (git-21.194.67617-f74b06c) to root...
Downloading
https://downloads.openwrt.org/releases/22.03.4/packages/aarch64_generic/luci/
luci-app-acl_git-21.194.67617-f74b06c_all.ipk
Installing luci-i18n-acl-zh-cn (git-23.090.61754-f7f34d4) to root...
Downloading
https://downloads.openwrt.org/releases/22.03.4/packages/aarch64_generic/luci/
luci-i18n-acl-zh-cn_git-23.090.61754-f7f34d4_all.ipk
Package luci-app-acl (git-21.194.67617-f74b06c) installed in root is up to
date.
Configuring luci-app-acl.
Configuring luci-i18n-acl-zh-cn.
Configuring luci-i18n-acl-en.
```

关闭

2) Check whether the software package is installed successfully

a. In the OpenWRT software package management interface, click the filter dialog box and enter “**luci-app-acl**”

b. Select and click **"Available"** on the Tab page

c. The **"luci-app-acl"** package will be displayed in the package list, and the update status will be **"installed"**

10. 11. 3. Remove package example

1) Take the removal of the package **"luci-app-acl"** as an example

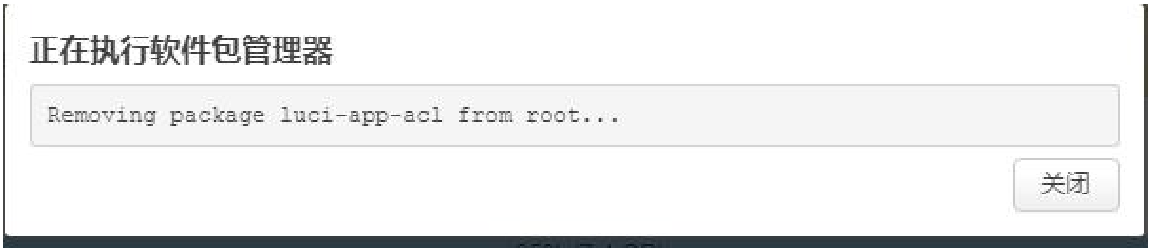
a. In the OpenWRT software package management interface, click the filter dialog box and enter **"luci-app-acl"**

b. Select **"Installed"** on the Tab page to display the list of installed software packages

c. Click **"Remove"** on the right to remove the corresponding software package

a. Then the following pop-up window will be displayed, click **"Remove"**

b. After the removal is successful, the display interface is as follows



2) Check whether the software package is removed successfully

a. In the OpenWRT software package management interface, click the filter dialog box and enter “**luci-app-acl**”

b. Select and click **"Installed"** on the Tab page

c. The “**luci-app-acl**” package will not be displayed in the package list, and the “**luci-app-acl**” package has been removed successfully

10. 12. Using Samba Network Shares

There are mainly two software options for OpenWRT LAN file sharing, Samba and NFS. The compatibility of the Samba system is better, while the performance of NFS is superior. For users who need to use Windows devices, it is recommended to choose Samba.

1) Enter the management page of the Samba network share

a. Find the **"Service"** option in the navigation bar and click to enter

b. In the vertical column options below the service, select **"network sharing"** and click to enter

2) Select the interface that the Samba service needs to monitor

a. Select **"General Settings"** in the navigation bar of network sharing and click to enter

b. The interface is specified according to actual needs. If you want to access through the "wan port", set it to **"wan"**

3) Set the shared directory of the network share

a. In the **"Shared Directory"** of the **"General Settings"** of the network share, click **"Add"** the shared directory address

b. Enter the name of the shared folder as **"mmt"** under the name

c. Under the path of the shared directory, choose to set the shared directory location **"/mnt"**

d. Check **"Browseable"** and **"Run anonymous user"**

e. Click **"Save and Apply"** to save the configuration

4) window10 starts network discovery and sharing

Note: To access Samba under the Windows 10 system, you need to confirm whether Windows 10 has enabled network discovery and sharing for sharing. If it is not enabled, perform the following settings first.

a. Enable Samba v1/v2 access

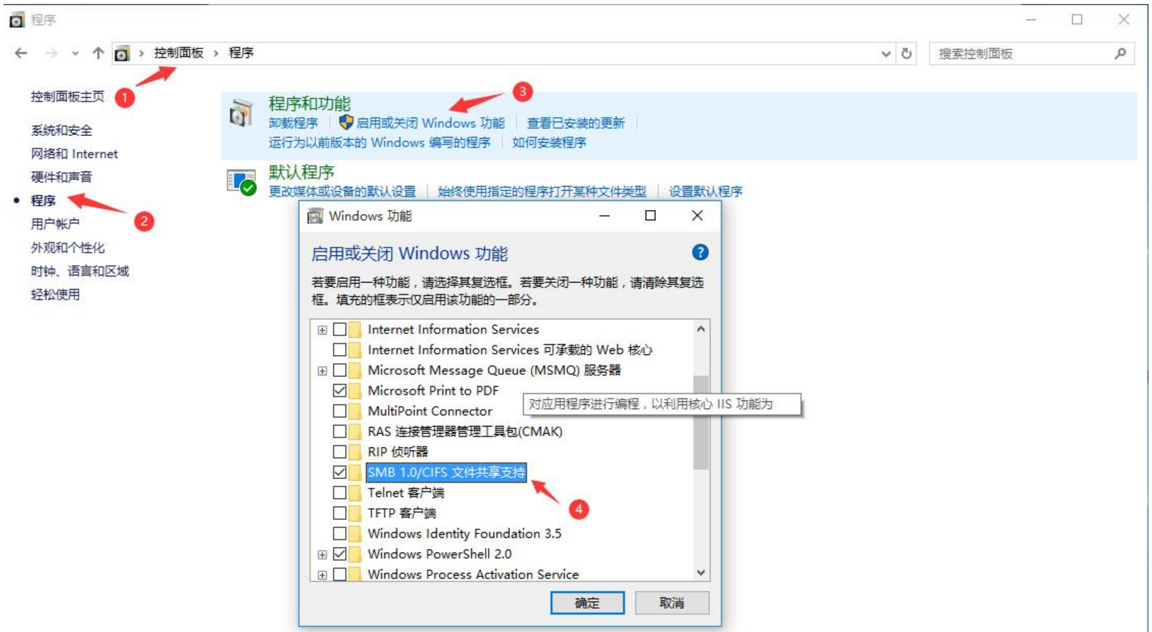
a) Enter the "Control Panel" of Windows 10

b) Click "Programs" on the left navigation bar of the control panel

c) Select "Turn Windows features on or off" in Programs and Features

d) Check "SMB 1.0/CIFS file sharing support" in the pop-up box of enabling or disabling Windows functions

e) Click "OK" to configure the application



b. Turn on the network discovery of Windows 10

a) Enter the "Control Panel" of Windows 10

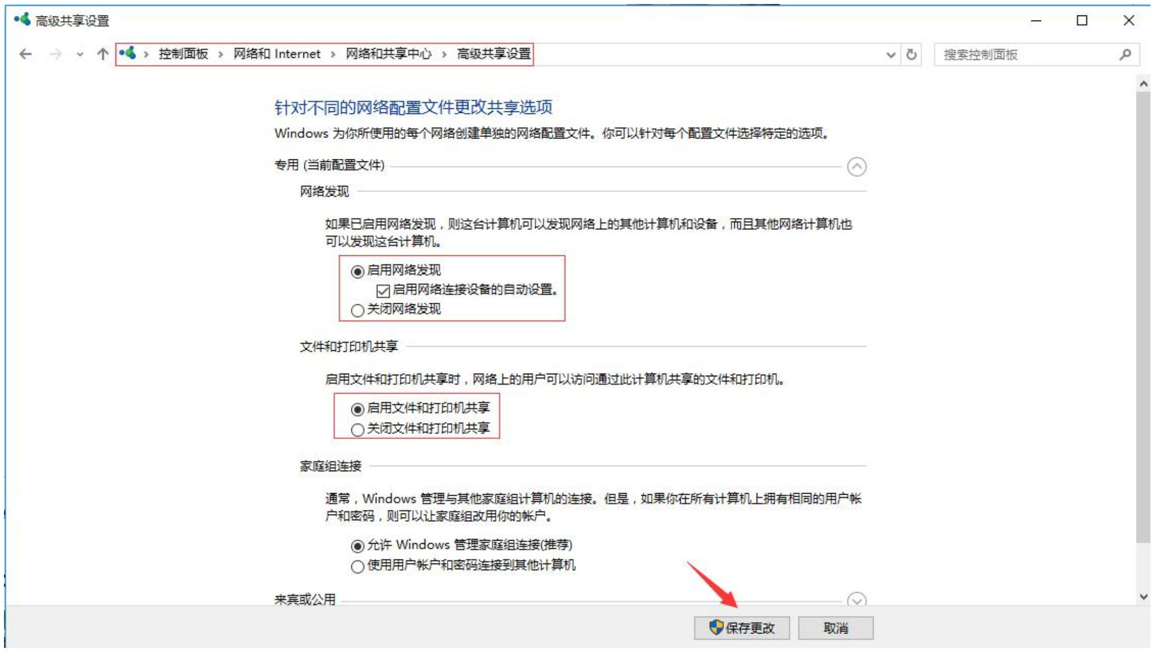
b) Select "Network and Internet" in the Control Panel

c) Then open "Network and Sharing Center"

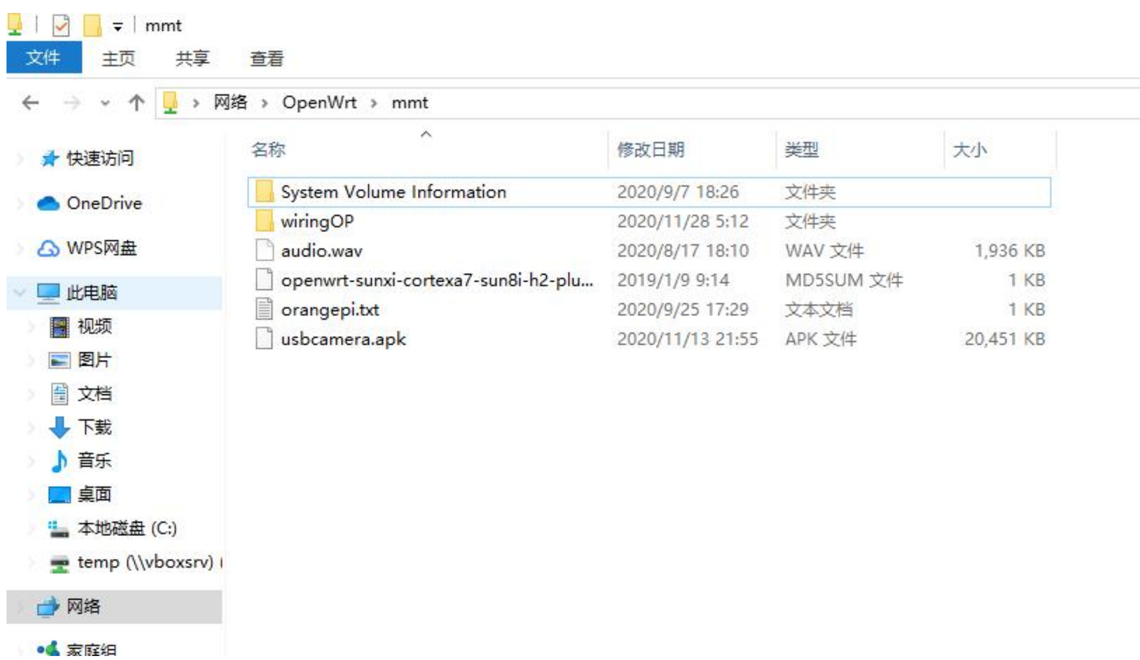
d) Click | "Advanced Sharing Settings"

e) Turn on **"Enable Network Discovery"** and **"Enable File and Printer Sharing"**

f) Click "Save Changes" to save the Windows 10 network discovery configuration



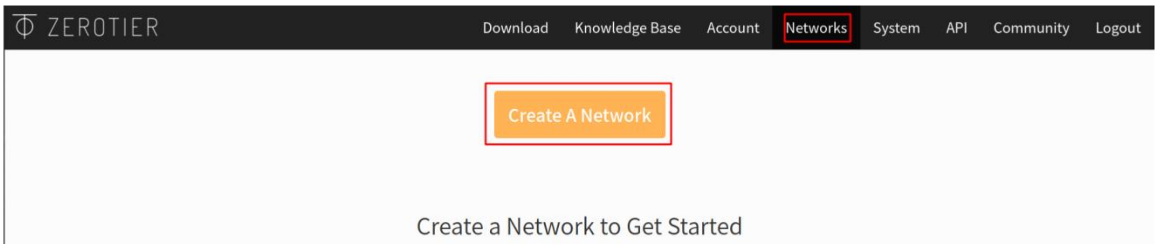
5) After the setting is completed, enter \\OpenWrt in the address bar of the resource manager to access the shared directory, the user name is root, and the password is the password set by the development board host

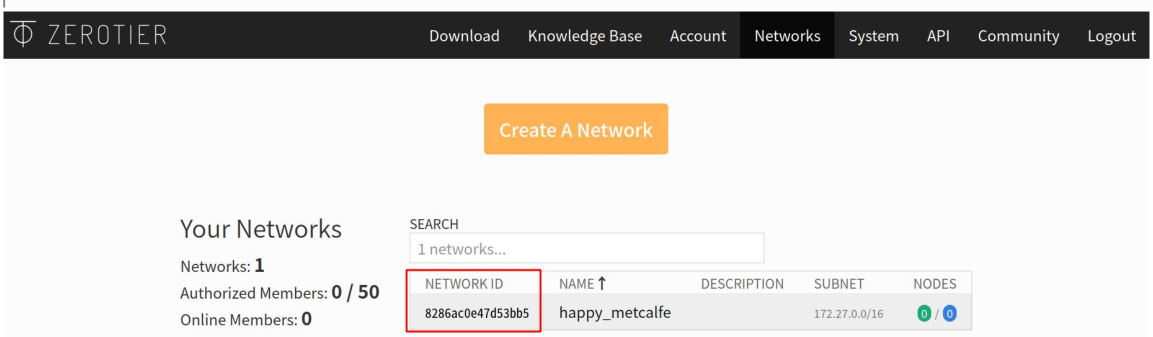


10. 13. Zerotier Instructions

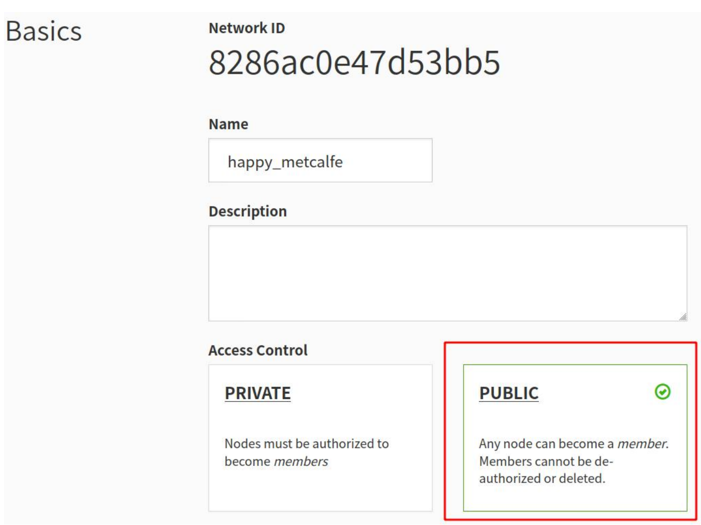
The OpenWRT system has pre-installed the zerotier client. After creating a virtual LAN on the zerotier official website, the client can directly join it through the Network ID. The specific operation is as shown below.

1) Log in to zerotier official website https://my.zerotier.com/network, register and log in and click Network->Create A Network to create a virtual local area network

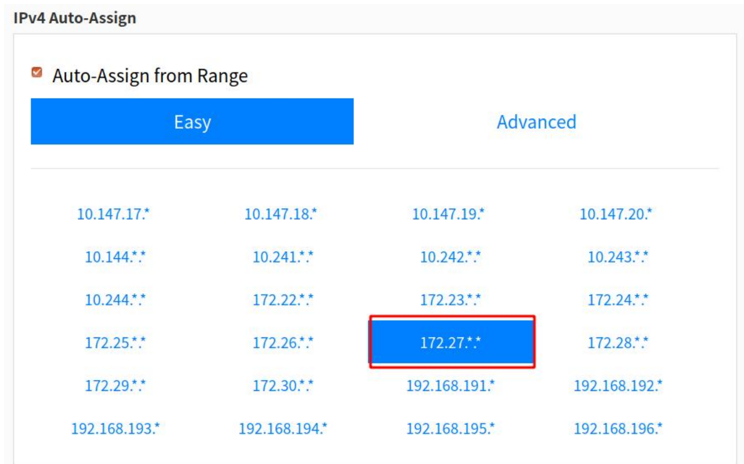




2) Click to enter the network console page, you can set the privacy option to public, so that the added network nodes do not need to be verified



3) The following automatically assigns the address Here you can choose the network segment yourself, here is 172.27.*.*



4) Enter the following command in the OpenWRT terminal to join the virtual LAN created above, **where 8286ac0e47d53bb5 is the Network ID of the virtual LAN created above**

```
root@OpenWrt:/# zerotier-one -d                         #Start the zerotier client
root@OpenWrt:/# zerotier-cli join 8286ac0e47d53bb5    #join the network
```

5) Enter ifconfig in the terminal and you can see that there is already a new **ztks54inm2** device with an IP address of **172.27.214.213**

```
root@OpenWrt:/# ifconfig
ztks54inm2 Link encap:Ethernet    HWaddr F6:4E:DE:BF:D8:52
           inet addr:172.27.214.213    Bcast:172.27.255.255    Mask:255.255.0.0
           inet6 addr: fe80::e82f:d0ff:fe5a:867e/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST    MTU:2800    Metric:1
           RX packets:18 errors:0 dropped:0 overruns:0 frame:0
           TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1720 (1.6 KiB)    TX byte81 (8.2 KiB)
```

6) Install the zerotier client on another device (Ubuntu18.04 is used as an example here), execute the following command to install, and restart the computer after the installation is complete

```
test@ubuntu:~$ curl -s https://install.zerotier.com | sudo bash
```

7) After restarting, join the virtual LAN according to the Network ID, and you can also see that the ip address assigned by zerotier has been obtained. At this time, the Ubuntu PC and OrangePi R1 Plus LTS are in the same LAN, and the two can communicate freely

```
test@ubuntu:~$ sudo zerotier-cli join 8286ac0e47d53bb5
test@ubuntu:~$ ifconfig
ztks54inm2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2800
        inet 172.27.47.214  netmask 255.255.0.0  broadcast 172.27.255.255
        inet6 fe80::5ce1:85ff:fe2b:6918  prefixlen 64  scopeid 0x20<link>
        ether f6:fd:87:68:12:cf  txqueuelen 1000  (ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 46  bytes 10006 (10.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

8) Test whether the two terminals can communicate

```
root@OpenWrt:/# ping 172.27.47.214 -I ztks54inm2
PING 172.27.47.214 (172.27.47.214): 56 data bytes
64 bytes from 172.27.47.214: seq=0 ttl=64 time=1.209 ms
64 bytes from 172.27.47.214: seq=1 ttl=64 time=1.136 ms
64 bytes from 172.27.47.214: seq=2 ttl=64 time=1.203 ms
64 bytes from 172.27.47.214: seq=3 ttl=64 time=1.235 ms
^C
--- 172.27.47.214 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.136/1.195/1.235 ms
```

9) other common commands of zerotier

```
root@OpenWrt:/# zerotier-one -d                      #Start the zerotier client
root@OpenWrt:/# zerotier-cli status                  #Get address and service status
root@OpenWrt:/# zerotier-cli join # Network ID       #join the network
root@OpenWrt:/# zerotier-cli leave # Network ID      #leave the network
root@OpenWrt:/# zerotier-cli listnetworks            #list networks
OPENWRT_DEVICE_REVISION="v0"
OPENWRT_RELEASE="OpenWrt 22.03.4 r20123-38ccc47687"
```

11. Compiling method of OpenWRT source code

11. 1. Download OpenWRT source code

1) First execute the following command to download the openwrt-22.03 branch code

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install -y git
test@test:~$ git clone https://github.com/orangepi-xunlong/openwrt.git -b openwrt-22.03
```

2) After the OpenWRT code is downloaded, the following files and folders will be included

```
test@test:~/openwrt$ ls
BSDmakefile   Config.in   include   Makefile   README.md   scripts   toolchain
Config   feeds.conf.default   LICENSE   package   rules.mk   target   tools
```

11. 2. Compile OpenWRT source code

1) First install the following dependent software (currently only tested to compile on Ubuntu 20.04 and need to install the following software, if compiling on other versions of the system, please install the dependent software by yourself according to the error message)

```
test@test:~/openwrt$ sudo apt update
test@test:~/openwrt$ sudo apt install -y ack antlr3 asciidoc autoconf   \
automake autopoint binutils bison build-essential   \
bzip2 ccache cmake cpio curl device-tree-compiler fastjar \
flex gawk gettext gcc-multilib g++-multilib git gperf haveged \
help2man intltool libc6-dev-i386 libelf-dev libglib2.0-dev \
libgmp3-dev libltdl-dev libmpc-dev libmpfr-dev \
libncurses5-dev \libncursesw5-dev libreadline-dev libssl-dev \
libtool lrzsz mkisofs msmtp nano ninja-build p7zip p7zip-full \
patch pkgconf python2.7 python3 python3-pyelftools \
libpython3-dev qemu-utils rsync scons squashfs-tools \
subversion swig texinfo uglifyjs upx-ucl unzip \
```

```
vim wget xmlto xxd zlib1g-dev
```

2) Then execute **./scripts/feeds update -a** and **./scripts/feeds install -a** to download dependent packages

```
test@test:~/openwrt$ ./scripts/feeds update -a
test@test:~/openwrt$ ./scripts/feeds install -a
```

3) Then choose to use the configuration file of OrangePi 5

```
test@test:~/openwrt$ cp configs/orangepi-5-rk3588_defconfig .config
test@test:~/openwrt$ make defconfig
```

4) Execute the following command to start compiling the openwrt source code

```
test@test:~/openwrt$ make V=s
```

5) After the compilation is complete, the path where the image is generated is:

```
test@test:~/openwrt$ tree -L 1 bin/targets/rockchip/armv8/
bin/targets/rockchip/armv8/
├── config.buildinfo
├── feeds.buildinfo
├── openwrt-rockchip-armv8-xunlong_orangepi-5-ext4-sysupgrade.img.gz
├── openwrt-rockchip-armv8-xunlong_orangepi-5.manifest
├── openwrt-rockchip-armv8-xunlong_orangepi-5-squashfs-sysupgrade.img.gz
├── packages
├── profiles.json
├── sha256sums
└── version.buildinfo

1 directory, 9 files
```

6) When compiling, you may encounter the error shown below

```
  OBJCOPY spl/u-boot-spl-nodtb.bin
  SYM     spl/u-boot-spl.sym
  CAT     spl/u-boot-spl-dtb.bin
  COPY    spl/u-boot-spl.bin
  BINMAN   .binman_stamp
```

```
Wrote map file './simple-bin.map' to show errors
binman: Node '/binman/simple-bin/fit': subnode 'images/@atf-SEQ': Failed to read ELF file: Python: No module named 'elftools'
make[3]: *** [Makefile:1108: .binman_stamp] Error 1
make[3]: Leaving directory
'/home/tangligang/openwrt/build_dir/target-aarch64_generic_musl/u-boot-orangepi-5-rk3588/u-boot-2023-04-16-24c50dca'
```

At this time, deleting the python3 compiled in the openwrt source code can solve this problem

```
test@test:~/openwrt$ rm staging_dir/hostpkg/bin/python3*
```

12. Appendix

12. 1. User Manual Update History

Version	Date	Release Notes
v0.1	2022-12-02	initial version
v0.2	2022-12-05	1. How to write Linux image to SPIFlash+NVMe SSD 2.Linux: How to upload files to the Linux system of the development board 3.Linux: How to download and install arm64 version balenaEtcher 4. How to burn Orange Pi OS (Droid) image to TF card 5. Burn Orange Pi OS (Droid) image to SPIFlash+NVMe SSD 6.Linux: How to log in to the desktop of the Linux system remotely
v0.3	2022-12-09	1. How to compile Android 12 source code 2. Linux: orangepi-build instructions 3. Linux: How to use adb
v0.4	2022-12-12	1.Linux: How to use SATA SSD 2. How to write Linux image to SPIFlash+SATA SSD 3. Linux: Test method of RTL8821CU USB WIFI module 4. Debian: How to set up Chinese environment and install Chinese input method
v0.5	2022-12-16	1.Linux: How to use AP6275P PCIe network card 2.Linux: How to install QT 3. How to install ROS 1 Noetic on Ubuntu 20.04 4. How to install ROS 2 Galactic on Ubuntu 20.04 5. How to install ROS 2 Humble on Ubuntu 22.04
v0.6	2022-12-23	1.Linux: Method of using commands to test recording 2.Linux: How to install kernel header files 3. Linux: How to use the 10.1-inch MIPI LCD screen 4.Ubuntu20.04: How to set Chinese and Chinese input methods in the system 5.Ubuntu22.04: How to set Chinese and Chinese input methods in the system

		6.Android12: How to burn Android image to SPIFlash+SATA SSD 7. Android12: How to use USB wireless network card 8.Android12: 26pin interface GPIO, UART, SPI and PWM test
v0.7	2023-01-06	1. How to burn Linux image to SPIFlash+USB storage device 2. Linux: How to install and use wiringOP-Python 3. Linux: Instructions for using the logo on and off 4.Linux: AP6275P PCIe network card creates WIFI hotspot method through create_ap 5. Ubuntu22.04: Instructions for using orangepi-build to compile the image on the development board
v0.8	2023-01-13	1. Android12: How to use the AP6275P PCIe network card 2. Android12: WIFI connection test method 3. Android12: How to use Wi-Fi hospot 4. Android12: Bluetooth test method 5. Android12: How to use 10.1-inch MIPI screen
v0.9	2023-02-17	1. How to burn Orange Pi OS (Droid) image to SPIFlash+SATA SSD 2. How to use RKDevTool to burn Linux image to TF card 3. Use RKDevTool to burn Linux image to SPIFlash+NVMe SSD 4.Linux: How to set the pull-up and pull-down resistance of 26pin GPIO port 5. Linux: How to use the CAN bus in 26pin 6. Android12: How to use ADB 7. Android12 Box: How to use the supported 2.4G USB remote control 8. Android12 Box: How to use the HDMI CEC function
v1.0	2023-03-03	1. Android12 source code compilation instructions: add the method of compiling LCD image and SATA image 2. The method of compiling the kernel source code separately in the linux system of the development board 3. Ubuntu22.04 Gnome Wayland Desktop System Instructions
v1.1	2023-03-08	1.Ubuntu22.04 Gnome: How to install ROS 2 Humble 2.Ubuntu22.04 Gnome: How to set up Chinese environment and install Chinese input method 3.Ubuntu22.04 Gnome: update Kodi instructions
v1.2	2023-03-17	1.Linux: OV13850 and OV13855 MIPI camera test method 2.Android: OV13850 and OV13855 MIPI camera test method

v1.3	2023-03-30	1. Fix the wrong picture of the camera wiring
v1.4	2023-04-14	1. Orange Pi OS Arch System Adaptation
v1.5	2023-04-21	1.OPi OS Arch: Update instructions for burning image 2. How to use SATA SSD in OPi OS Arch system 3. OPi OS Arch: How to use the 10.1-inch MIPI LCD screen 4. How to use OPi OS Arch: AP6275P PCIe WIFI6+Bluetooth module 5.OPi OS Arch: OV13850 and OV13855 MIPI camera test method 6.OPi OS Arch: How to install wiringOP 7. OPi OS Arch: 26pin GPIO, I2C, UART, SPI, CAN and PWM test 8.OPi OS Arch: How to set Chinese environment and install Chinese input method 9. Exchange the position instructions of LCD1 and LCD2 to be consistent with the silk screen on the development board 10. OpenWRT system instructions 11. Compiling method of OpenWRT source code
v1.6	2023-04-28	1. Updated method of burning OpenWRT image
v1.7	2023-06-06	1. Update the method of using RKDevTool to burn the image into SPIFlash+SSD 2. How to clear SPIFlash using RKDevTool 3. Delete the method of burning using the mtd tool of the OpenWRT system 4. Add a method to use the dd command to burn the OpenWRT system 5. Ubuntu/Debian：How to use ZFS file system 6. Ubuntu/Debian：How to turn off the green light by default at startup 7. Add usage instructions for Debian12 system (scattered in multiple sections) 8. Add method to view nvme ssd temperature
v1.8	2023-11-08	1. Method of using wiringOP hardware PWM 2. Ubuntu/Debian：How to use CasaOS
v1.9	2024-01-26	1. How to burn Orange Pi OS (OH) image to TF card
v2.0	2024-02-02	1. Linux6.1 system adaptation situation 2. Linux：How to use NPU 2. Linux：RK3588 How to use PaddlePaddle
v2.1	2024-07-03	1. Linux: How to run RKLLM large model on RK3588

12. 2. Image update history

Date	Release Notes
2022-12-02	Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.7z * initial version
2022-12-05	Orangepi5_1.0.2_debian_bullseye_desktop_xfce_linux5.10.110.7z * Pre-installed with balenaEtcherhe and Gparted * Pre-installed ffmpeg and mpv player * Add some scripts and configuration files
2022-12-09	Orangepi5_1.0.2_debian_bullseye_server_linux5.10.110.7z Orangepi5_1.0.2_ubuntu_jammy_server_linux5.10.110.7z Orangepi5_1.0.2_ubuntu_jammy_desktop_xfce_linux5.10.110.7z * initial version
2022-12-12	Orangepi5_1.0.4_debian_bullseye_server_linux5.10.110.7z Orangepi5_1.0.4_debian_bullseye_desktop_xfce_linux5.10.110.7z Orangepi5_1.0.4_ubuntu_jammy_server_linux5.10.110.7z Orangepi5_1.0.4_ubuntu_jammy_desktop_xfce_linux5.10.110.7z * Add rk3588-ssd-sata.dtbo * Add rkspi_loader_sata.img, used to start the linux system on sata ssd * Pre-installed usb-modeswitch package, test RLT8821CU WIFI module can be used normally
2022-12-16	Orangepi5_1.0.6_debian_bullseye_desktop_xfce_linux5.10.110.7z * Support ov13855 camera * Support open multiple mipi cameras at the same time * Test that qt can be installed and used normally * Some scripts are pre-installed Orangepi5_1.0.6_ubuntu_jammy_desktop_xfce_linux5.10.110.7z * Test that ros can be installed and used normally

	* Test that qt can be installed and used normally * Some scripts are pre-installed * Fix the bug that the fcitx5 configuration program cannot be opened Orangepi5_1.0.6_ubuntu_focal_server_linux5.10.110.7z Orangepi5_1.0.6_ubuntu_focal_desktop_xfce_linux5.10.110.7z * initial version
2022-12-23	Orangepi5_1.0.8_debian_bullseye_server_linux5.10.110.7z Orangepi5_1.0.8_ubuntu_jammy_server_linux5.10.110.7z Orangepi5_1.0.8_ubuntu_focal_server_linux5.10.110.7z Orangepi5_1.0.4_debian_bullseye_desktop_xfce_linux5.10.110.7z Orangepi5_1.0.4_ubuntu_jammy_desktop_xfce_linux5.10.110.7z * Some scripts are pre-installed * Solve the error problem of kernel header file deb package installation Orangepi5_1.0.6_ubuntu_focal_desktop_xfce_linux5.10.110.7z * Some scripts are pre-installed * Solve the error problem of kernel header file deb package installation * Solve the problem that the mipi camera cannot be used OrangePi5_RK3588S_Android12_v1.0.1.img OrangePi5_RK3588S_Android12_lcd_v1.0.1.img OrangePi5_RK3588S_Android12_spi-nvme_lcd_v1.0.1.img OrangePi5_RK3588S_Android12_spi-nvme_v1.0.1.img * Support OV13855 camera * Support RTL8211CU, RTL8822CU, RTL8723BU three USB wireless network cards * Enable UART0, I2C5, SPI4, PWM15 by default * Pre-installed WiringOP APP is used to operate GPIO, I2C, SPI and UART hardware resources

	OrangePi5_RK3588S_Android12_spi-sata_v1.0.1.img OrangePi5_RK3588S_Android12_spi-sata_lcd_v1.0.1.img * initial version
2023-01-06	Orangepi5_1.1.0_debian_bullseye_server_linux5.10.110.7z Orangepi5_1.1.0_ubuntu_focal_server_linux5.10.110.7z Orangepi5_1.1.0_ubuntu_jammy_server_linux5.10.110.7z * Pre-install create_ap, support AP6275P PCIe network card to open hotspot function * Support SPIFlash+USB storage device to start Linux system (only USB3.0 interface) * Open some kernel configuration Orangepi5_1.1.0_debian_bullseye_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.0_ubuntu_focal_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.0_ubuntu_jammy_desktop_xfce_linux5.10.110.7z * Add switch to display logo * Set VOP DCLK as dynamic allocation strategy * Pre-install create_ap, support AP6275P PCIe network card to open hotspot function * Support SPIFlash+USB storage device to start Linux system (only USB3.0 interface) * Optimize the set_lcd_rotate.sh script to solve the unusable problem in Debian11 * Open some kernel configuration
2023-01-13	OrangePi5_RK3588S_Android12_v1.0.2.img OrangePi5_RK3588S_Android12_lcd_v1.0.2.img OrangePi5_RK3588S_Android12_spi-nvme_v1.0.2.img OrangePi5_RK3588S_Android12_spi-nvme_lcd_v1.0.2.img OrangePi5_RK3588S_Android12_spi-sata_v1.0.2.img OrangePi5_RK3588S_Android12_spi-sata_lcd_v1.0.2.img * Support the Bluetooth function of RTL8821CU, RTL8723BU wireless

	network card * Support AP6275P PCIe network card * Solve the problem that some TF cards cannot enter the desktop when starting OrangePi-OS_Droid_orangepi5_v0.0.4_beta.img OrangePi-OS_Droid_orangepi5_spi-nvme_v0.0.4_beta.img OrangePi-OS_Droid_orangepi5_spi-sata_v0.0.4_beta.img The following functions have been updated compared to the original v0.0.2 version： * Solve the problem that the application cannot be restored to window mode after being maximized * Solve the problem that some applications cannot be closed after full screen * Solve the problem that there will be an instant gray background when opening the application * Solve the problem that some full-screen applications switch to the desktop and then switch back to the application to become a windowed display problem * Solve known issues related to volume adjustment in the control center * Solve the camera crash problem * Solve the game full screen open, click the taskbar to wake up the crash problem * Delete the screen saver option in the control center * Solve the problem of closing a single application in the task manager and clicking on other applications again * The control center has added a shutdown function and a function of long pressing the Bluetooth module to enter the Bluetooth setting * Support the Bluetooth function of RTL8821CU, RTL8723BU wireless network card OrangePi-OS_Droid_orangepi5_en_v0.0.4_beta.img OrangePi-OS_Droid_orangepi5_spi-nvme_en_v0.0.4_beta.img OrangePi-OS_Droid_orangepi5_spi-sata_en_v0.0.4_beta.img * initial version
2023-02-17	Orangepi5_1.1.2_debian_bullseye_server_linux5.10.110.7z

	Orangepi5_1.1.2_debian_bullseye_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.2_ubuntu_focal_server_linux5.10.110.7z Orangepi5_1.1.2_ubuntu_focal_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.2_ubuntu_jammy_server_linux5.10.110.7z * Support SPIFlash+ blue USB2.0 interface to start Linux system * wiringOP supports the function of setting GPIO pull-up and pull-down resistors Orangepi5_1.1.2_ubuntu_jammy_desktop_xfce_linux5.10.110.7z * Support ov13850/ov13855 camera * Support SPIFlash+ blue USB2.0 interface to start Linux system * wiringOP supports the function of setting GPIO pull-up and pull-down resistors OrangePi5_RK3588S_Android12-box_v1.0.0.img OrangePi5_RK3588S_Android12-box_spi-sata_v1.0.0.img OrangePi5_RK3588S_Android12-box_spi-nvme_v1.0.0.img * initial version
2023-02-24	OrangePi-OS_Droid_orangepi5_v0.0.5_beta.img OrangePi-OS_Droid_orangepi5_spi-nvme_v0.0.5_beta.img OrangePi-OS_Droid_orangepi5_spi-sata_v0.0.5_beta.img OrangePi-OS_Droid_orangepi5_en_v0.0.5_beta.img OrangePi-OS_Droid_orangepi5_spi-nvme_en_v0.0.5_beta.img OrangePi-OS_Droid_orangepi5_spi-sata_en_v0.0.5_beta.img Added notification reminder function * Removed the function of double-clicking the desktop to prompt the user whether to sleep * Removed the pop-up tool window when long pressing the desktop * Fixed the desktop moving icon function: the application will disappear when dragging and moving the application * Added the control center to click outside the window to automatically close

	the function * Fixed the problem that when closing a single application in the task manager, the task manager will be closed at the same time * Fixed the bottom recent application list, open new applications in order of priority * Added the control center to click the corresponding function, and the function is opened in windowed mode * Added a shutdown module at the bottom of the application list, click to choose sleep, shutdown and restart the device * Fixed the problem that the wallpaper item in the settings shows a lock screen * Fixed the problem that the close and maximize buttons in the hotspot setting interface of the control center are misplaced * Removed the function of user-defined application window size, the default is to open in window mode, and the user can adjust the window size by himself after opening the application * Fixed the problem that after uninstalling the application on the desktop, the application icon will not disappear, the moving icon will be black, and the system will freeze after restarting * Fixed the problem that PCIe SSD burning failed
2023-03-03	Orangepi5_1.1.2_ubuntu_jammy_desktop_gnome_linux5.10.110 * initial version
2023-03-09	Orangepi5_1.1.4_ubuntu_jammy_desktop_gnome_linux5.10.110 * Set HDMI as the default audio device * Optimize GPU performance * Turn on the hardware mouse to solve the problem of desktop flickering * Solve the problem of ros2 installation failure Orangepi5_1.1.4_debian_bullseye_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.4_ubuntu_focal_desktop_xfce_linux5.10.110.7z * Set HDMI as the default audio device * Turn on the hardware mouse * Support exfat file system

	* Set the default size of /boot partition to 1GB, which is convenient for kernel development Orangepi5_1.1.4_ubuntu_jammy_desktop_xfce_linux5.10.110.7z * Set HDMI as the default audio device * Turn on the hardware mouse * Support exfat file system * Set the default size of /boot partition to 1GB, which is convenient for kernel development * Pre-installed RK-adapted Chromium browser, supports h264/vp8/vp9 hardware solution to play video Orangepi5_1.1.4_ubuntu_focal_server_linux5.10.110.7z Orangepi5_1.1.4_debian_bullseye_server_linux5.10.110.7z Orangepi5_1.1.4_ubuntu_jammy_server_linux5.10.110.7z * Support exfat file system * Set the default size of /boot partition to 1GB, which is convenient for kernel development OrangePi5_RK3588S_Android12_spi-nvme_v1.0.2.img OrangePi5_RK3588S_Android12_spi-nvme_lcd_v1.0.2.img * Fixed the problem that burning Android image to empty nvme ssd failed
2023-03-30	Orangepi5_1.1.4_debian_bullseye_desktop_kde-plasma_linux5.10.110.7z * initial version
2023-04-07	OrangePi-OS_Droid_orangepi5_v0.0.6_beta.img OrangePi-OS_Droid_orangepi5_spi-nvme_v0.0.6_beta.img OrangePi-OS_Droid_orangepi5_spi-sata_v0.0.6_beta.img OrangePi-OS_Droid_orangepi5_en_v0.0.6_beta.img OrangePi-OS_Droid_orangepi5_spi-nvme_en_v0.0.6_beta.img OrangePi-OS_Droid_orangepi5_spi-sata_en_v0.0.6_beta.img

	* Fix the problem that wake-up from hibernation cannot be used normally * Added application minimization function, after the application is minimized, it can be opened in the bottom application list * Fix the problem that the Home button in the navigation bar does not work * Added the function of hiding the control center window when opening the application list * Solve the problem that the task manager does not display real-time thumbnails of recent tasks * Repair the uninstall function to delete the corresponding application in the desktop, application list, and recent tasks synchronously * The new full screen window hides the top title bar by default, slide the mouse to the top to display the title bar
2023-04-14	Opios-arch-aarch64-gnome-opi5-23.04-linux5.10.110.img.xz * initial version
2023-04-21	Opios-arch-aarch64-gnome-opi5-23.04.1-linux5.10.110.img.xz * Set the default value of loglevel to 2 to reduce the printing information of the serial port * Fix the problem that ssh or serial port login is slow due to network problems * Exchange the dtbo configuration of LCD1 and LCD2 to be consistent with the silkscreen on the development board openwrt-rockchip-armv8-xunlong_orangepi-5-ext4-sysupgrade.img.gz * initial version
2023-04-28	openwrt-aarch64-opi5-23.04-linux5.10.110-ext4.img.gz * Support SPIFlash + NVMe SSD boot * Support SPIFlash + USB boot openwrt-aarch64-opi5-23.04-linux5.10.110-ext4-sata.img.gz openwrt-rockchip-armv8-xunlong_orangepi-5-spi-squashfs-sysupgrade.bin * initial version

2023-06-06	Orangepi5_1.1.6_ubuntu_jammy_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.6_debian_bullseye_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.6_debian_bullseye_desktop_kde-plasma_linux5.10.110.7z * Update mpp package * Fix the problem that zfs file system cannot be installed * Fix the problem of stuck startup when rtl8821cu wifi module is plugged in * Added rk3588-disable-led.dtbo, used to turn off the green light * Rename rk3588-ssd-sata.dtbo to rk3588-ssd-sata0.dtbo Update the chromium browser to chromium-browser_110.0, which supports video hard decoding and playback in h264, h265, vp8, vp9 and av1 formats Orangepi5_1.1.6_ubuntu_focal_server_linux5.10.110.7z Orangepi5_1.1.6_debian_bullseye_server_linux5.10.110.7z Orangepi5_1.1.6_ubuntu_jammy_server_linux5.10.110.7z Orangepi5_1.1.6_ubuntu_focal_desktop_xfce_linux5.10.110.7z Orangepi5_1.1.6_ubuntu_jammy_desktop_gnome_linux5.10.110.7z * Fix the problem that zfs file system cannot be installed * Fix the problem of stuck startup when rtl8821cu wifi module is plugged in * Added rk3588-disable-led.dtbo, used to turn off the green light * Rename rk3588-ssd-sata.dtbo to rk3588-ssd-sata0.dtbo Orangepi5_1.1.6_debian_bookworm_server_linux5.10.110 Orangepi5_1.1.6_debian_bookworm_desktop_xfce_linux5.10.110 * initial version
2023-11-08	Orangepi5_1.1.8_ubuntu_focal_server_linux5.10.160.7z Orangepi5_1.1.8_ubuntu_jammy_server_linux5.10.160.7z Orangepi5_1.1.8_debian_bullseye_server_linux5.10.160.7z Orangepi5_1.1.8_debian_bookworm_server_linux5.10.160.7z Orangepi5_1.1.8_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi5_1.1.8_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi5_1.1.8_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi5_1.1.8_debian_bookworm_desktop_xfce_linux5.10.160.7z

	* Kernel version upgraded to 5.10.160 * Support PWM control through wiringOP Orangepi5_RK3588_Android12_v1.0.5.tar.gz Orangepi5_RK3588_Android12_lcd_v1.0.5.tar.gz Orangepi5_RK3588_Android12-box_v1.0.5.tar.gz Orangepi5_RK3588_Android12_spi-nvme_v1.0.5.tar.gz Orangepi5_RK3588_Android12-box_spi-nvme_v1.0.5.tar.gz Orangepi5_RK3588_Android12_lcd_spi-nvme_v1.0.5.tar.gz * Kernel version upgraded to 5.10.160
2024-02-02	Orangepi5_1.1.8_ubuntu_focal_server_linux6.1.43.7z Orangepi5_1.1.8_ubuntu_jammy_server_linux6.1.43.7z Orangepi5_1.1.8_debian_bullseye_server_linux6.1.43.7z Orangepi5_1.1.8_debian_bookworm_server_linux6.1.43.7z Orangepi5_1.1.8_ubuntu_focal_desktop_xfce_linux6.1.43.7z Orangepi5_1.1.8_ubuntu_jammy_desktop_xfce_linux6.1.43.7z Orangepi5_1.1.8_debian_bullseye_desktop_xfce_linux6.1.43.7z Orangepi5_1.1.8_debian_bookworm_desktop_xfce_linux6.1.43.7z * initial version
2024-07-03	Orangepi5_1.1.10_ubuntu_jammy_server_linux6.1.43.7z Orangepi5_1.1.10_debian_bullseye_server_linux6.1.43.7z Orangepi5_1.1.10_debian_bookworm_server_linux6.1.43.7z Orangepi5_1.1.10_ubuntu_jammy_desktop_xfce_linux6.1.43.7z Orangepi5_1.1.10_debian_bullseye_desktop_xfce_linux6.1.43.7z Orangepi5_1.1.10_debian_bookworm_desktop_xfce_linux6.1.43.7z Orangepi5_1.1.10_ubuntu_focal_server_linux5.10.160.7z Orangepi5_1.1.10_ubuntu_jammy_server_linux5.10.160.7z Orangepi5_1.1.10_debian_bullseye_server_linux5.10.160.7z Orangepi5_1.1.10_debian_bookworm_server_linux5.10.160.7z Orangepi5_1.1.10_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi5_1.1.10_ubuntu_jammy_desktop_xfce_linux5.10.160.7z

	Orangepi5_1.1.10_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi5_1.1.10_debian_bookworm_desktop_xfce_linux5.10.160.7z * The kernel's rknpu version is upgraded to 0.9.6