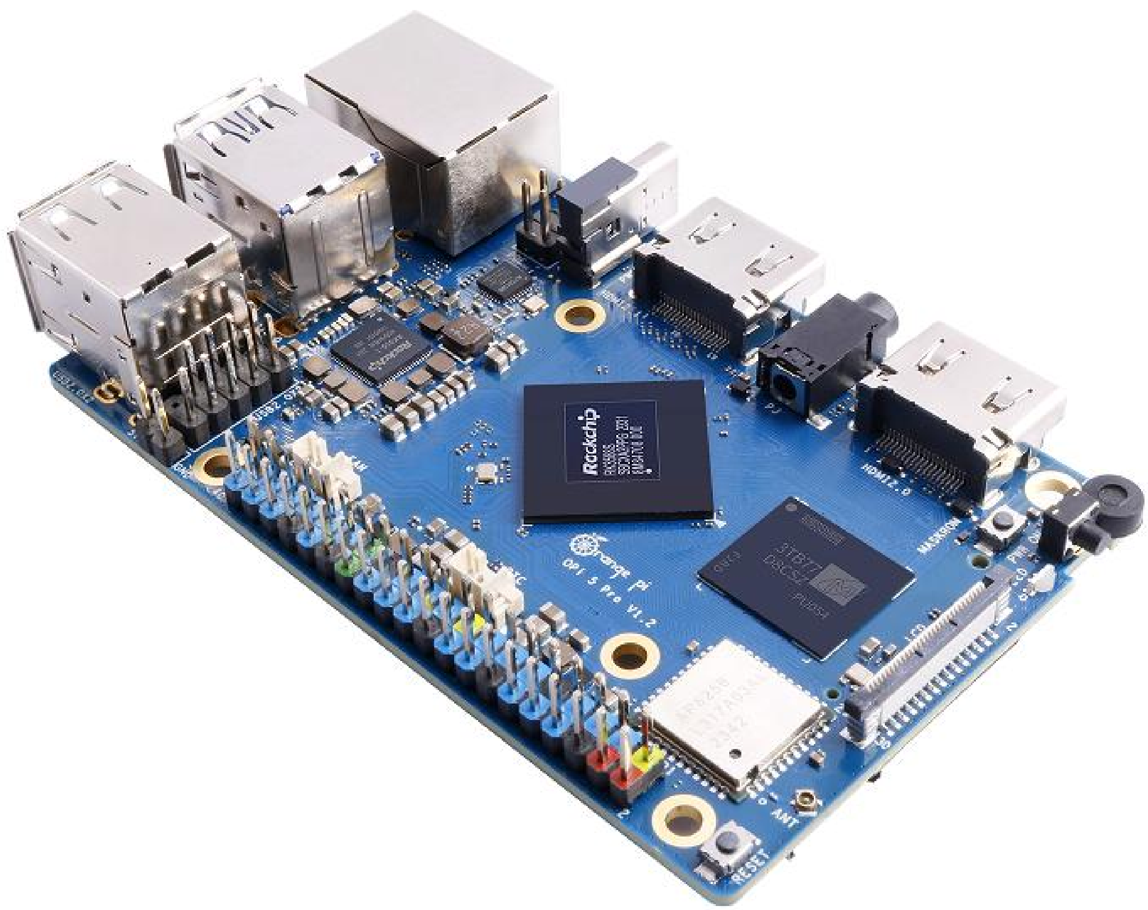
Orange Pi 5 Pro
User Manual



Catalogue

1. Basic features of Orange Pi 5 Pro

1.1. What is Orange Pi 5 Pro

The Orange Pi 5 Pro adopts the new generation of 8-core 64 bit ARM processors from the Ruixin Micro RK3588S, specifically the quad core A76 and quad core A55. It adopts the Samsung 8nm LP process technology, with a large core main frequency of up to 2.4GHz. It integrates the ARM Mali-G610 MP4 GPU, embedded with high-performance 3D and 2D image acceleration modules, and an AI accelerator NPU with up to 6 Tops of computing power. It has 4GB/8GB/16GB (LPDDR5) memory and up to 8K display processing capabilities.

The Orange Pi 5 Pro offers a wide range of interfaces, including HDMI output, Wi Fi 5, M.2 PCIe 2.0x1, Gigabit Ethernet ports, support for PoE+(requiring PoE+HAT), USB2.0, USB3.1 interfaces, and 40pin expansion pins. It can be widely used in high-end tablet, edge computing, artificial intelligence, cloud computing, AR/VR, intelligent security, smart home and other fields, covering all AIoT industries.

Orange Pi 5 Pro supports the official operating system Orange Pi OS developed by Orange Pi, as well as operating systems such as Android 12.1, Debian11, Debian12, Ubuntu 20.04, and Ubuntu 22.04.

1.2. Purpose of Orange Pi 5 Pro

We can use it to achieve:

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

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

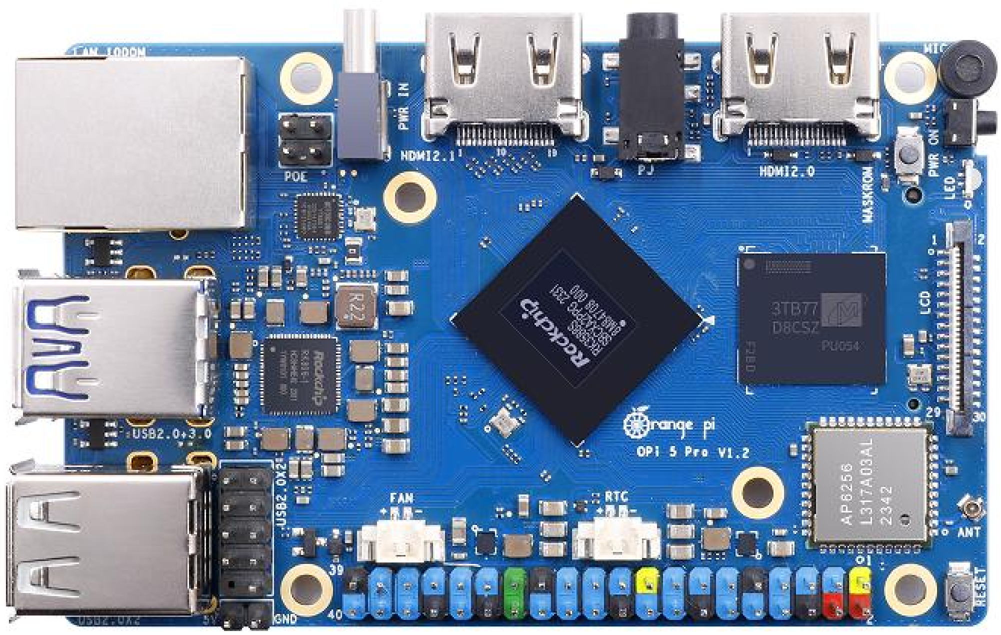
1. 3. Hardware features of Orange Pi 5 Pro

Introduction to Hardware Features	
CPU	• Rockchip RK3588S (8nm LP process） • 8-core 64 bit processor • 4-core Cortex-A76 and 4-core Cortex-A55 large and small core architectures • The maximum frequency of the large core is 2.4GHz, while the maximum frequency of the small core is 1.8GHz
GPU	• Integrated ARM Mali-G610 • OpenGL ES1.1/2.0/3.2、OpenCL 2.2 and Vulkan 1.2
NPU	• An AI accelerator NPU with built-in computing power of up to 6 Tops • Supports mixed operations of INT4/INT8/INT16
Video output	• HDMI 2.1，Maximum support 8K @60Hz • HDMI 2.0，Maximum support 4K@60Hz • 1* MIPI D-PHY TX 4Lane
Memory	4GB/8GB/16GB（LPDDR5）
camera	• 1 * MIPI CSI 4Lane • 1 * MIPI D-PHY RX 4Lane
PMU	RK806-1
Onboard storage	• EMMC socket, can be connected to external eMMC modules • 16MB QSPI No FLASH (default empty patch, reused with eMMC) • MicroSD (TF) Card slot • PCIe 2.0x1 M.2 M-KEY (SSD) slot
Ethernet	10/100/1000Mbps Ethernet, supports PoE+(requires PoE+HAT)
audio frequency	• 3.5mm headphone jack audio input/output • On board MIC input

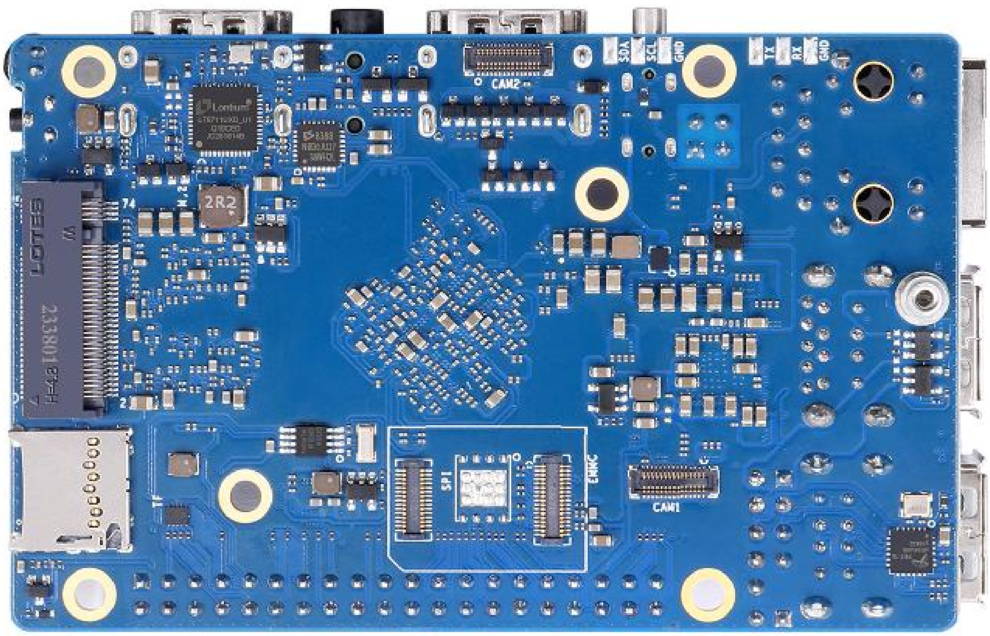
	• 2 * HDMI output
PCIe M.2 M-KEY	• Supports SSD
USB interface	1 * USB3.0 Supports Device or HOST mode 3 * USB2.0 HOST 2 * USB2.0 HOST (Lead out through 9-pin pin pin arrangement)
40pin Extended needle arrangement	Used to expand UART, PWM, I2C, SPI, CAN, and GPIO interfaces
Debug UART	Included in the 40PIN expansion port
LED lamp	RGB LED Three color indicator light
Keys	1 * MaskROM key，1 * RESET key，1 * On/Off key
power supply	Type-C interface power supply 5V/5A；
Supported operating systems	Orange Pi OS（Droid）、Orange Pi OS（Arch）、Android12.1、Debian11、Debian12、Ubuntu20.04 and Ubuntu22.04 etc
Introduction to appearance specifications	
Product size	89mm*56mm
weight	58g
range Pi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd	

1.4. Top and Bottom Views of Orange Pi 5 Pro

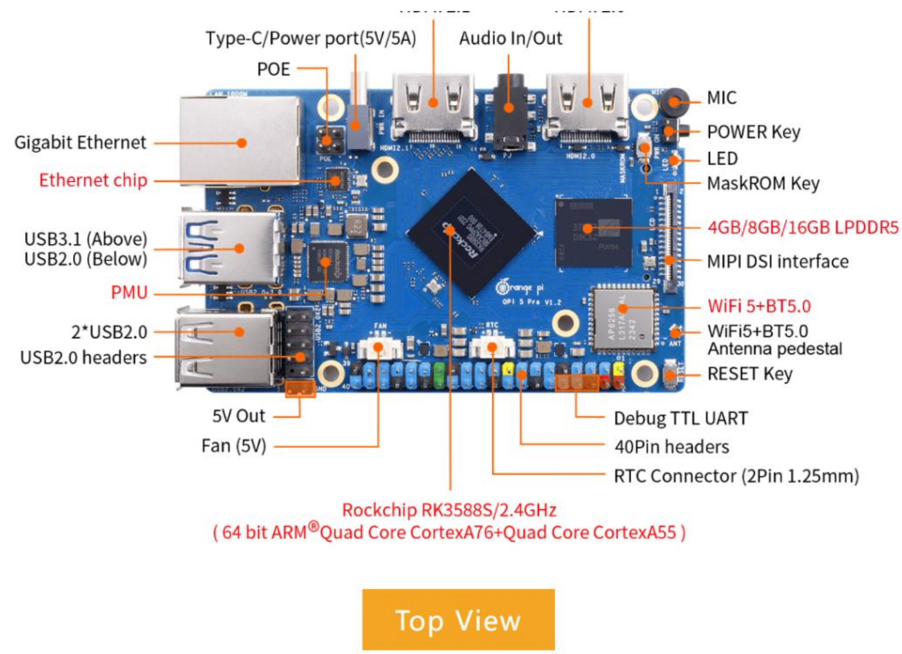
Top view:

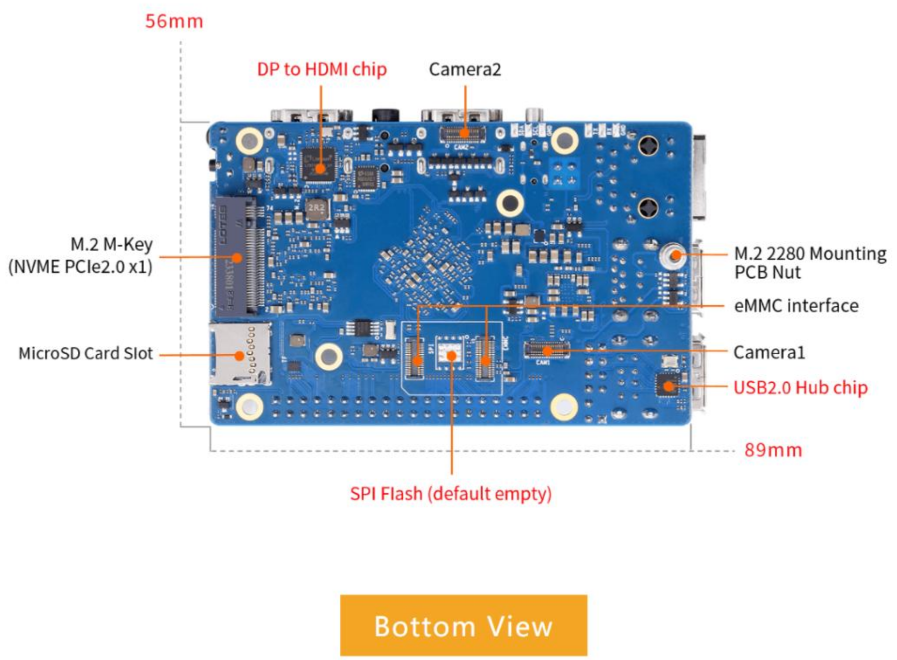


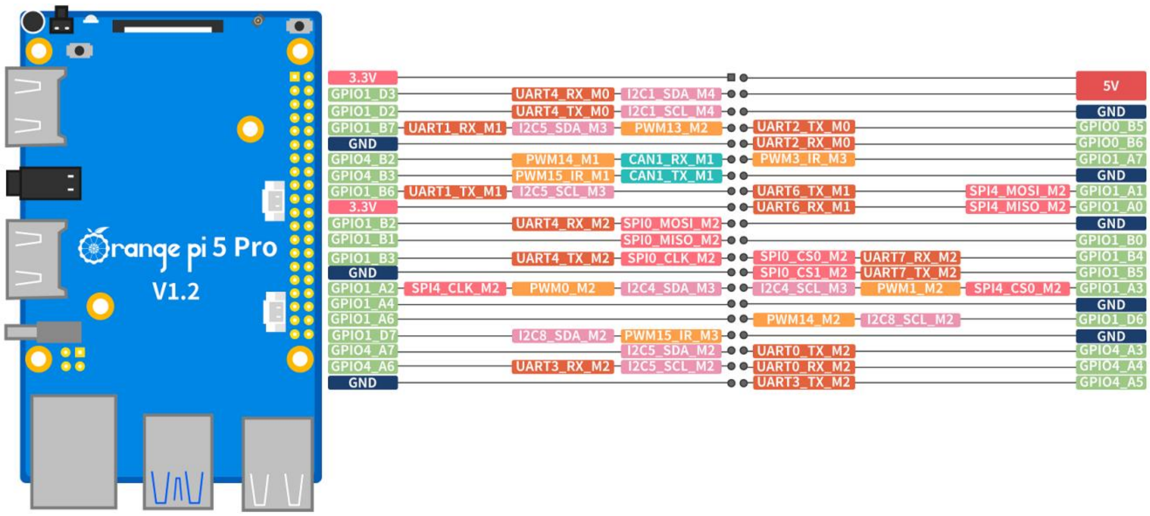
Bottom view:


CAM2
2R2
CAM1

1.5. 1.5. Interface Details of Orange Pi 5 Pro







The diameter of all four positioning holes is 3.0mm.

2. Introduction to using the development board

2. 1. Prepare the necessary accessories

1) TF card, a **class10** or above high-speed SanDisk card with a minimum capacity of 8GB (32GB or above is recommended)

2) TF card reader, used to burn images into TF cards

3) HDMI interface display

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

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

5) 10.1-inch MIPI screen, used to display the system interface of the development board (this screen includes adapter board and OPi5Plus/OPi5B/OPi5/OPi5Pro universal)

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

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

7) USB interface mouse and keyboard, as long as it is a standard USB interface mouse and keyboard, can be used to control the Orange Pi development board

8) USB camera

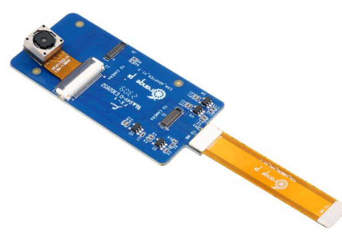
9) 5V cooling fan. As shown in the following figure, there are interfaces on the development board for connecting the cooling fan. The specifications of the interfaces are **2pin 1.25mm spacing**

The fan on the development board can be adjusted for speed and switch through PWM.

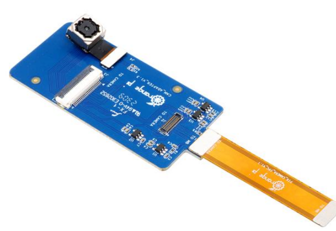
10) 100Mbps or Gigabit Ethernet cable, used to connect the development board to the Internet

11) USB 2.0 corporate to corporate data cable, used for burning images to eMMC, NVMe SSD, and other functions

12) OV13850 camera with 13 million MIPI interface



13) OV13855 camera with 13 million MIPI interface



14) When using the serial port debugging function, a **3.3V** USB to TTL module and DuPont cable are required to connect the development board and computer with the USB to TTL module and DuPont cable

15) Personal computers with Ubuntu and Windows operating systems installed

1	Ubuntu22.04 PC	Optional, used to compile Linux source code
2	Windows PC	Used to burn 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-Pro.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 referred to here specifically refers to Linux distribution images such as Debian, Ubuntu, OpenWRT, or OPi OS Arch downloaded from the Orange Pi download page.

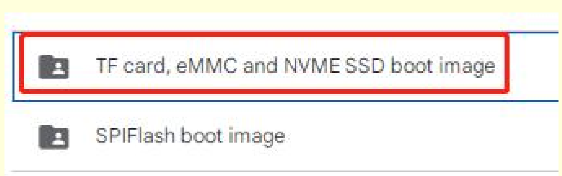
2. 3. 1. How to use balenaEtcher to burn Linux

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

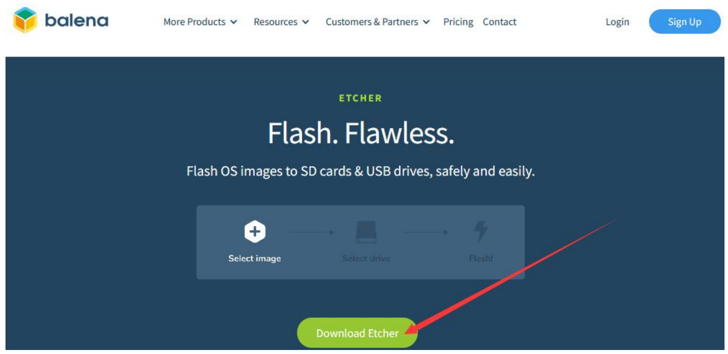
Note that if you are downloading an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the image file in the "TF Card, eMMC, and NVME SSD Boot Image" folder.



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

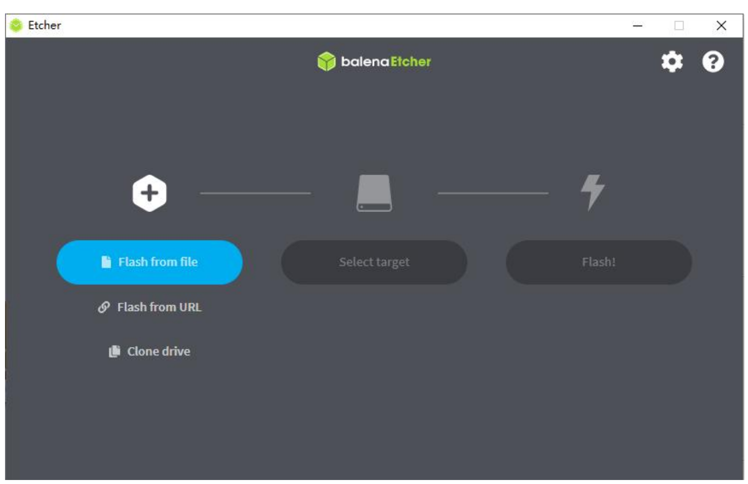
https://www.balena.io/etcher/

5) After entering the BalenaEtcher download page, clicking the green download button will redirect you to the software download location

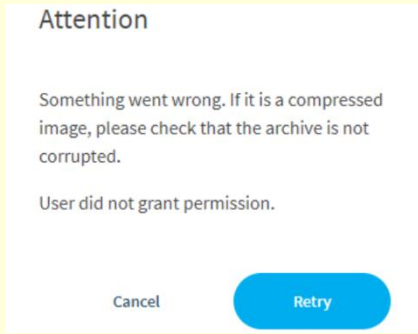


6) Then you can choose to download the Portable version of BalenaEtcher software. The Portable version does not need to be installed, and can be opened by double clicking to use it

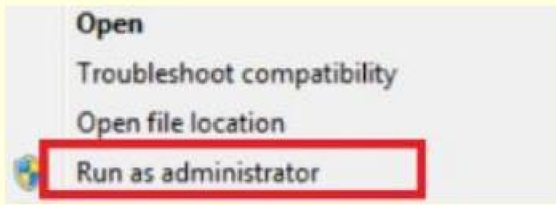
7) If you are downloading a version of BalenaEtcher that requires installation, please install it before using it. If you download the Portable version of balenaEtcher, simply double-click to open it. The interface of balenaEtcher after opening is shown in the following figure:



If the following error is prompted when opening balenaEtcher:



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



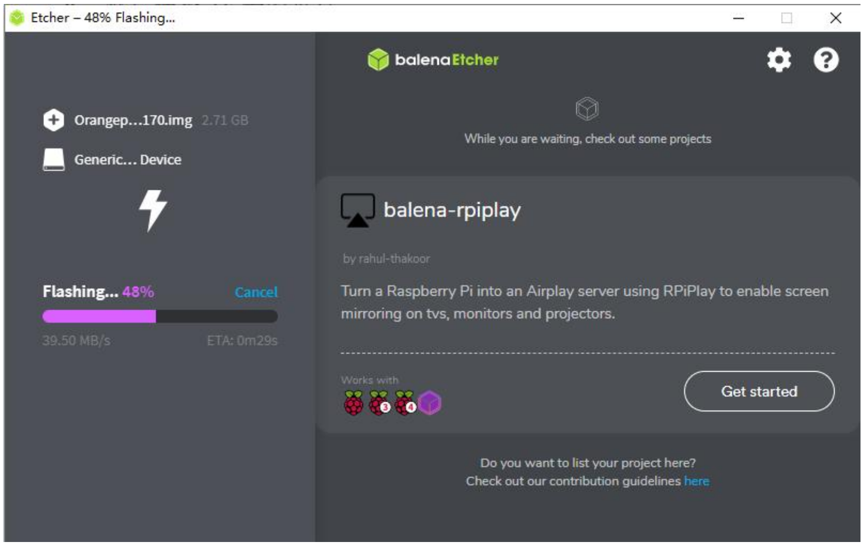
8) The specific steps to use balenaEtcher to burn the Linux image are aa follow

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

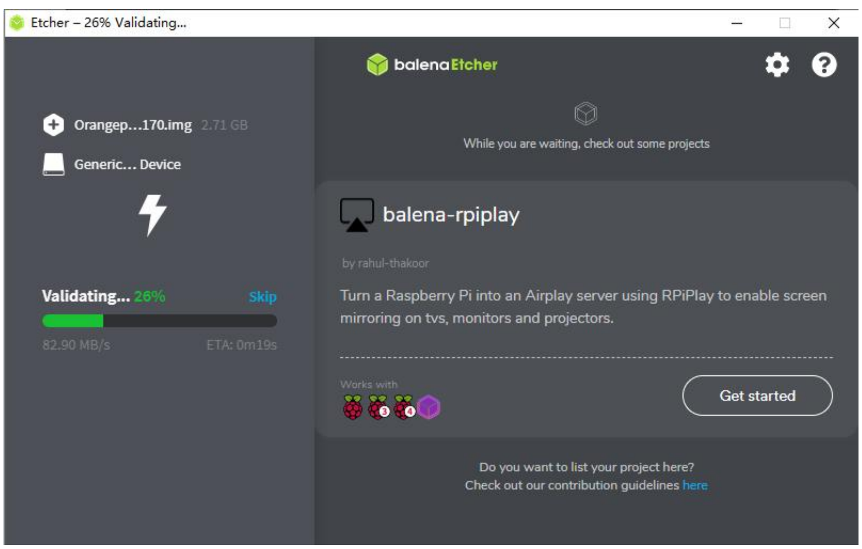
b. Then select the drive letter of the TF card

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

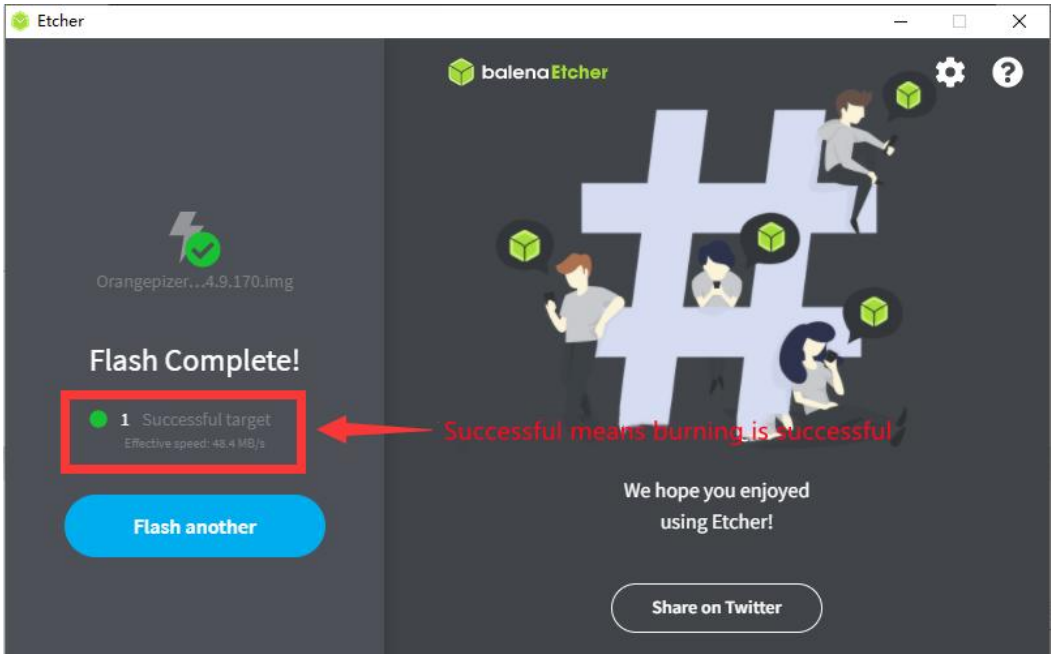
9) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card



10) 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



11) 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

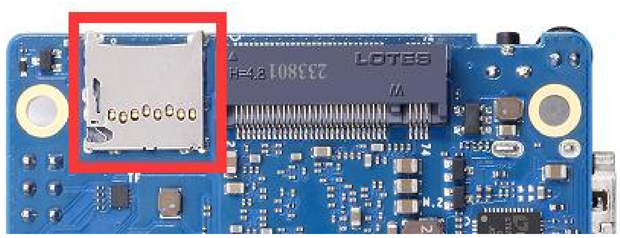


2. 3. 2. How to use RKDevTool to burn Linux image to TF card

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

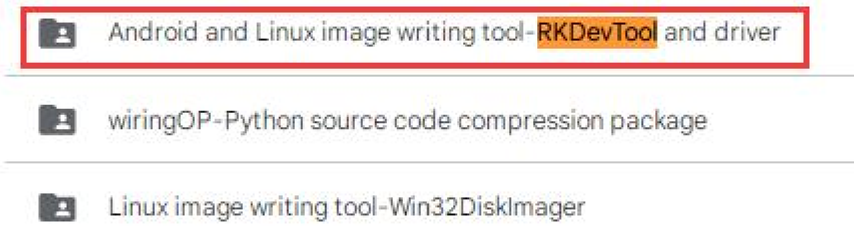
2) You also need to prepare a 16GB or larger TF card. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brands

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

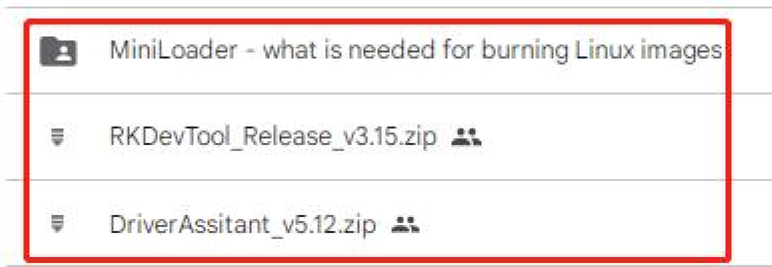


4) Then download the Ruixin micro driver **DriverAssitant_v5.12.zip**. zip and MiniLoader, as well as the burning tool **RKDevTool_Release_v3.15.zip**, from **Orange Pi's data download page**

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



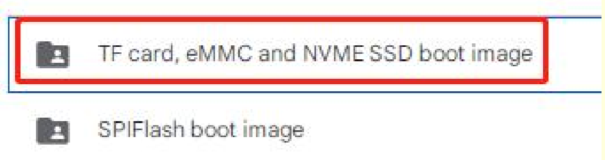
b. Then download all the files below



Note that the "MiniLoader - something needed to burn Linux images" folder 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

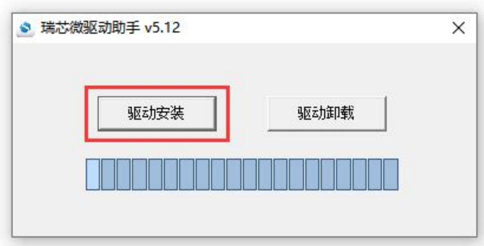
Note that if you are downloading an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the image file in the "TF Card, eMMC, and NVME SSD Boot Image" folder.



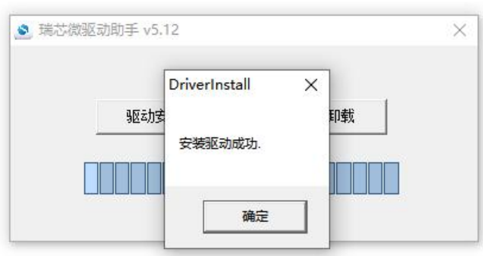
6) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**. zip, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

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

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



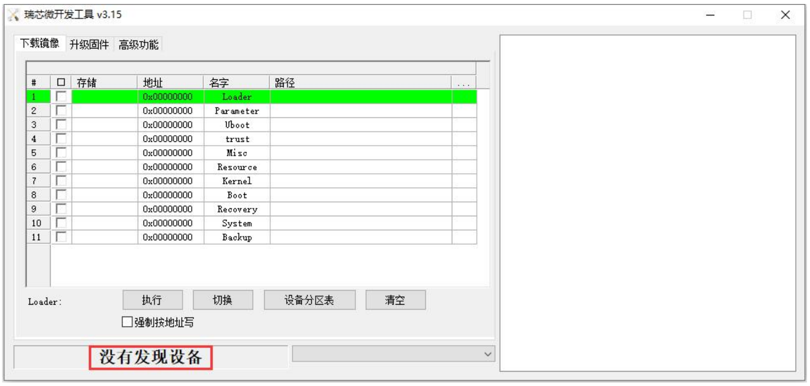
b. After waiting for a while, a pop-up window will prompt "**driver installed successfully**", and then click the "**OK**" button.



8) Then decompress **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

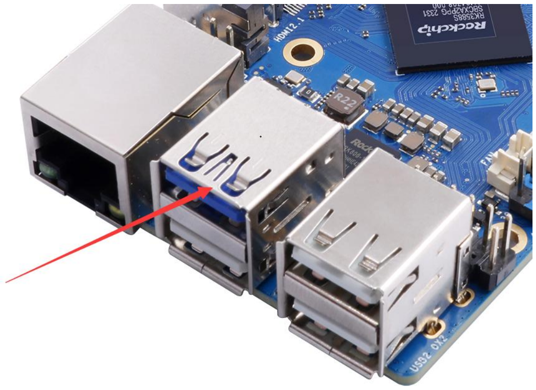
9) After opening the **RKDevTool** burning tool, because the computer has not been connected to the development board through the Type-C cable at this time, the lower left

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



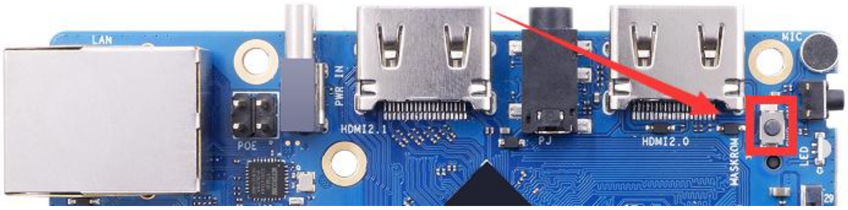
10) Then start burning the Linux image to the TF card

a. Firstly, connect the development board to the Windows computer through a USB male to female data cable. The location of the USB flash port on the development board is shown in the following figure

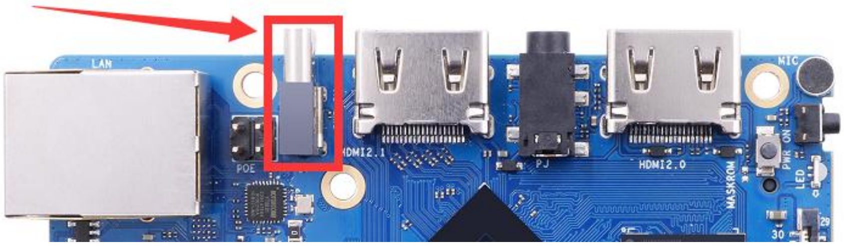


b. Then insert the TF card into the development board and ensure that the board is not connected to a power source

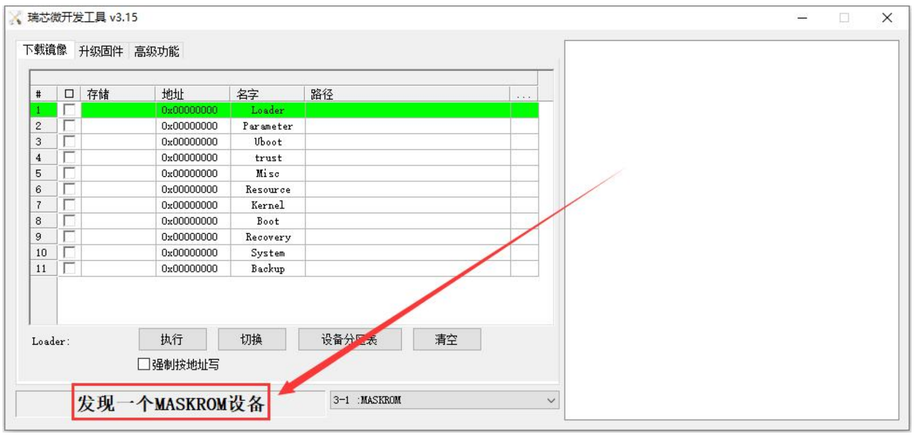
c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:



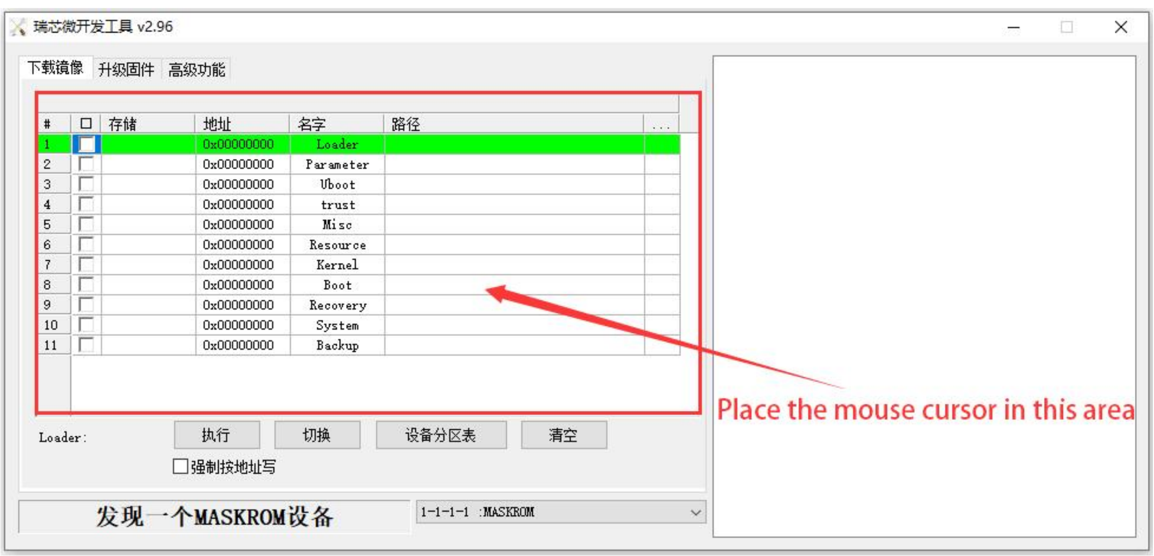
d. Then connect the Type-C interface power supply to the development board and power it on, then you can release the MaskROM button



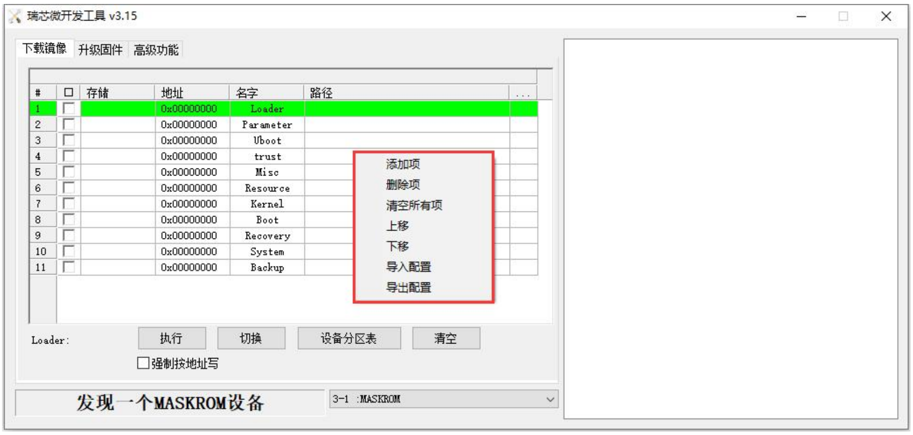
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



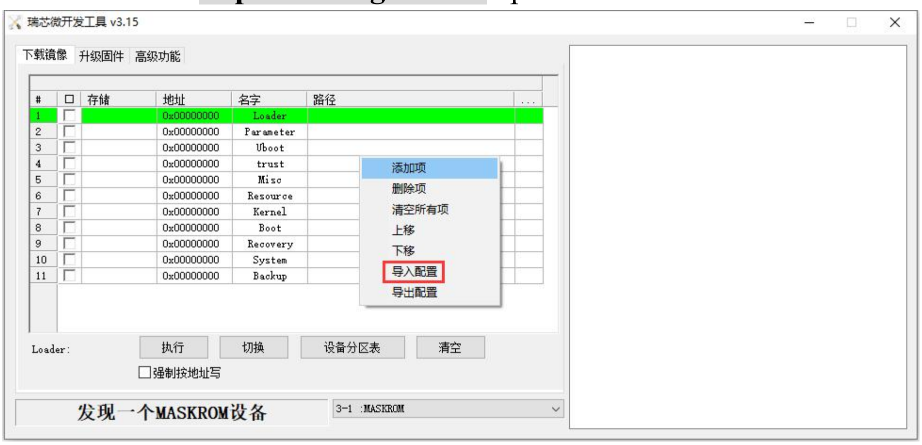
f. Then place the mouse cursor in the area below



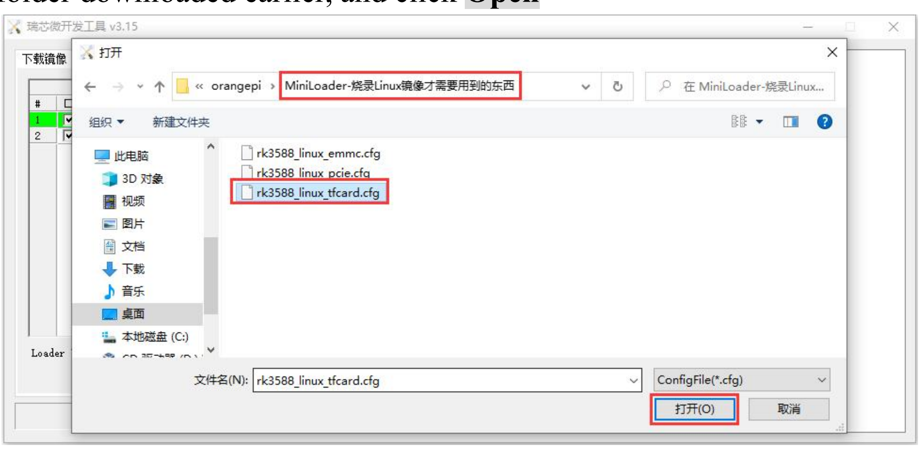
g. Then click the right mouse button and the selection interface shown in the figure below will pop up



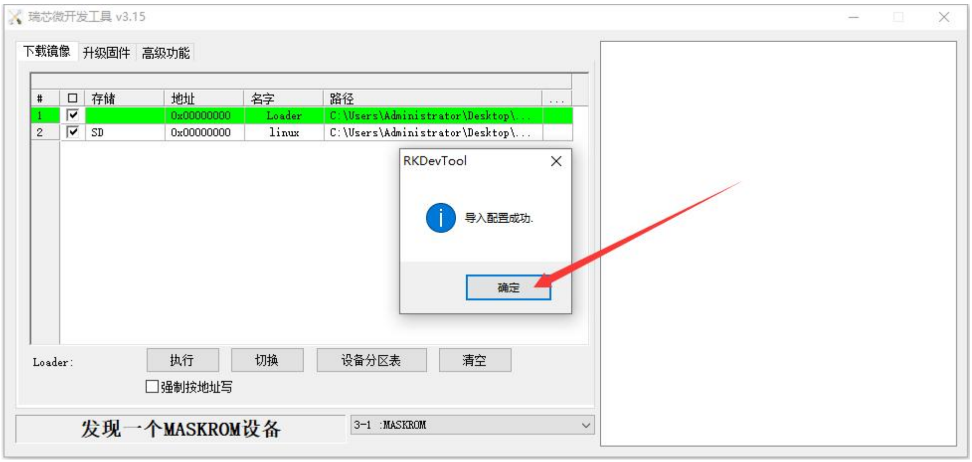
h. Then select the **import configuration** option



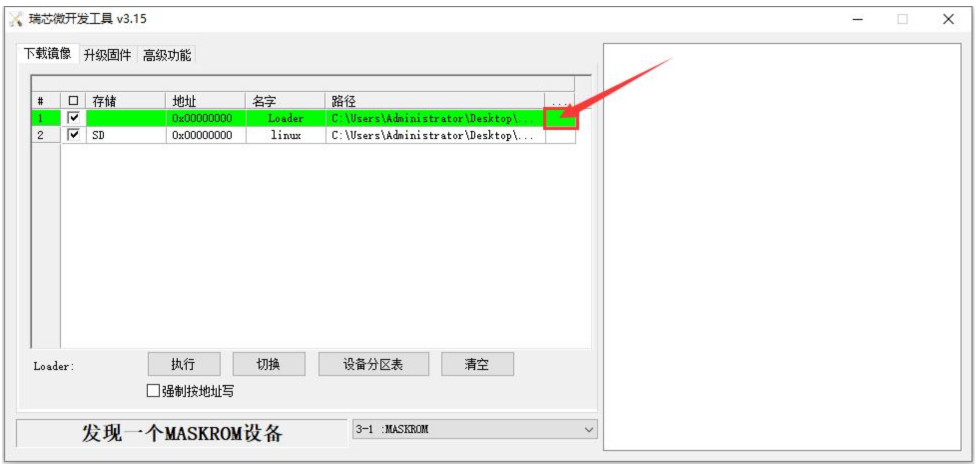
i. Then select the **rk3588_linux_tfcard.cfg** configuration file in the MiniLoader folder downloaded earlier, and click **Open**



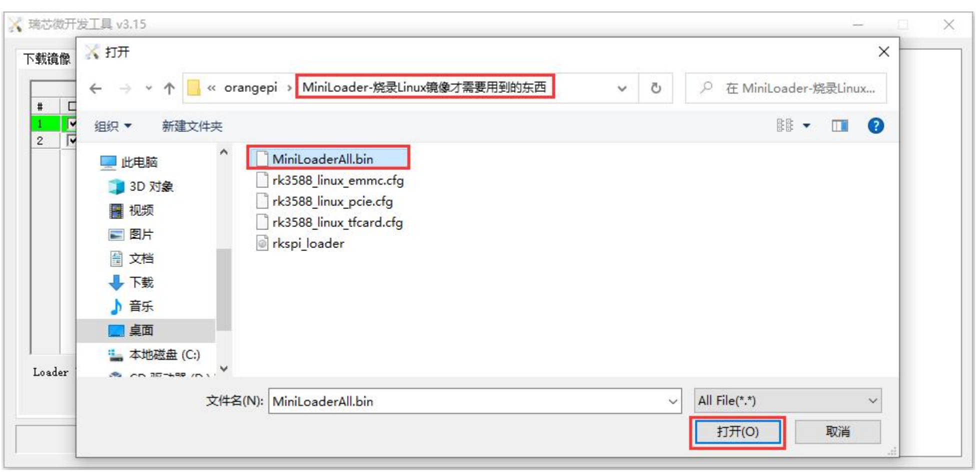
j. Then click **OK**



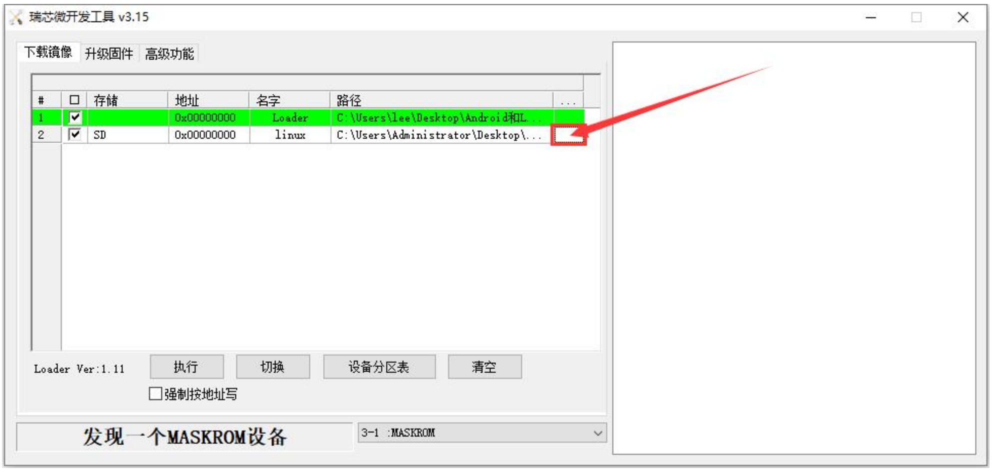
k. Then click the position shown in the figure below



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

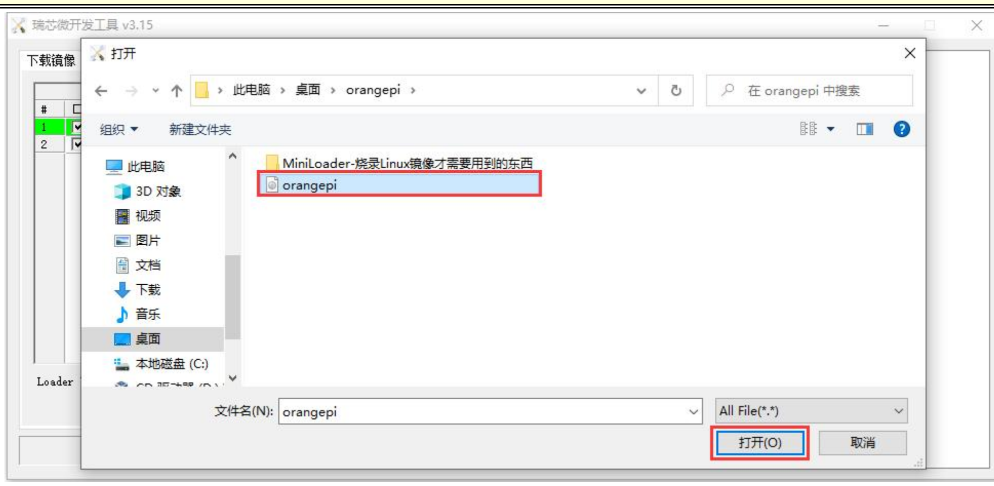


m. Then click on the location shown in the following image

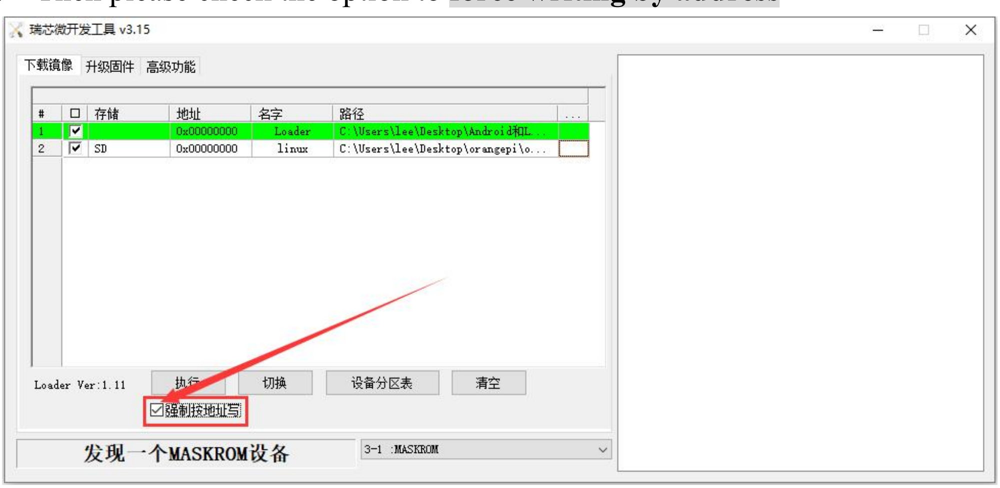


n. Then select the path to the Linux image you want to burn, and **click open**

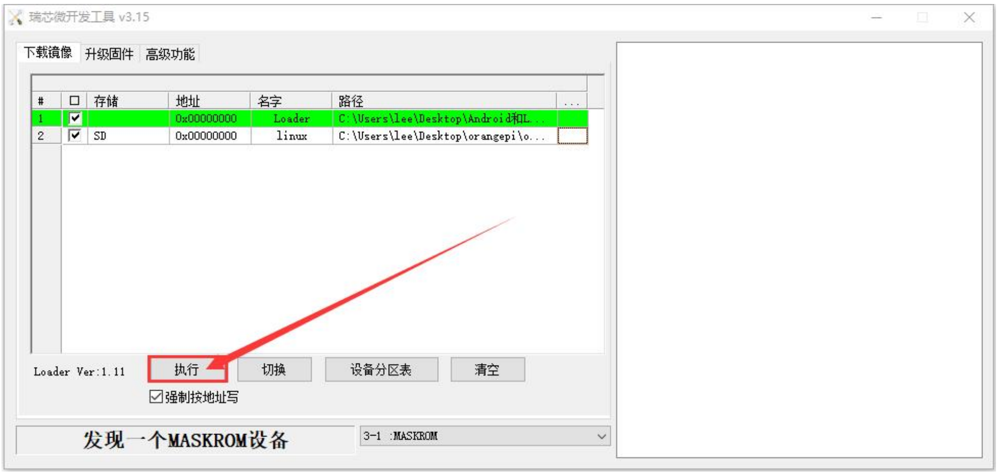
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



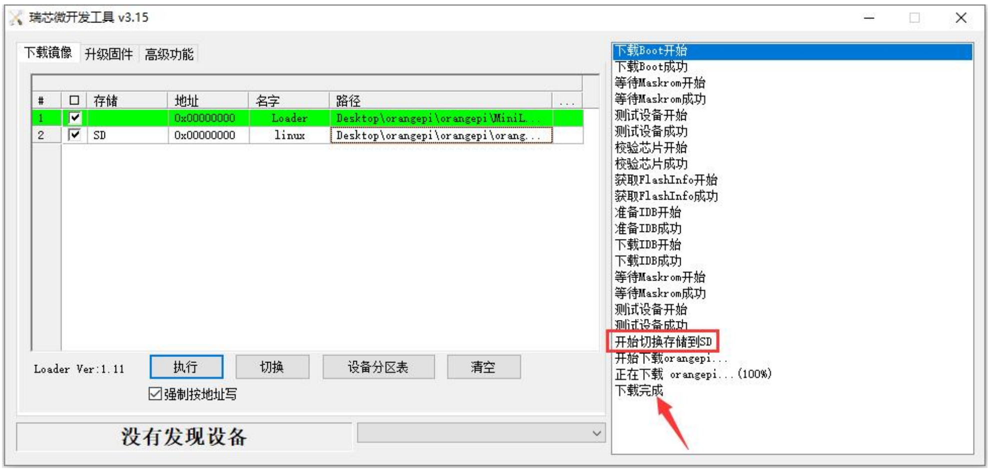
o. Then please check the option to **force writing by address**



p. Clicking the execute button again to start burning the Linux image to the TF card of the development board



q. The displayed log after burning the Linux image is shown in the following figure



r. After burning the Linux image to the TF card, the Linux system will automatically start.

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

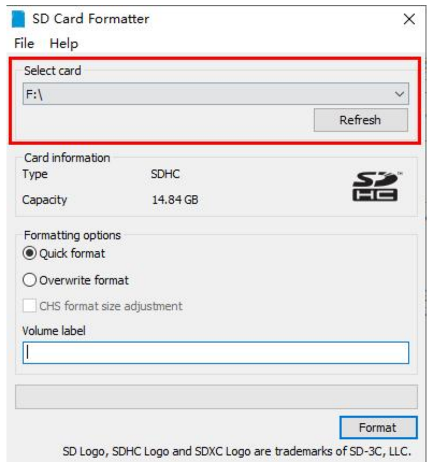
3) Then format the TF

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

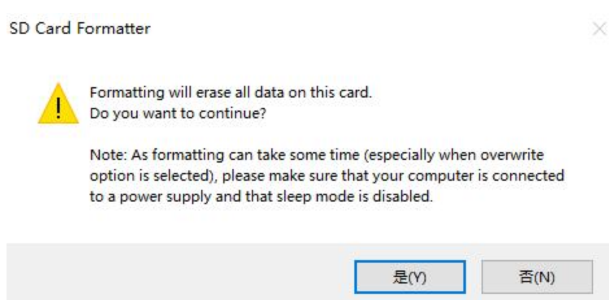
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

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

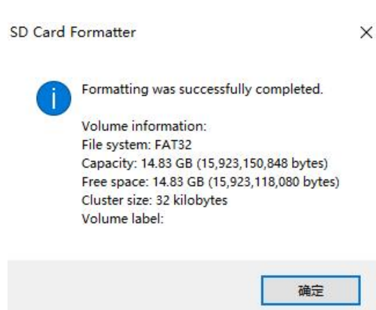
c. If only a TF card is inserted into the computer, the drive letter of the TF card will be displayed in the "**Select card**" column. If multiple USB storage devices are inserted into the computer, you can select the corresponding drive letter of the TF card through the drop-down box.



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

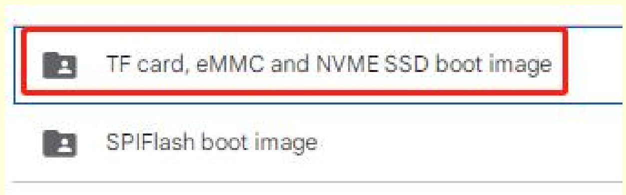


e. After formatting the TF card, the information shown in the figure below will pop up, click OK



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.



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

a. The download page of Win32Diskimager is

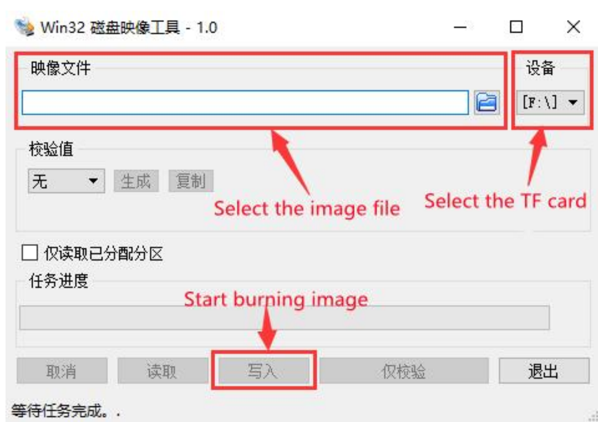
http://sourceforge.net/projects/win32diskimager/files/Archive/

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 referred to here specifically refers to Linux distribution images such as Debian, Ubuntu, OpenWRT, or OPi OS Arch downloaded from the Orange Pi download page. Ubuntu PC refers to a personal computer with the Ubuntu system installed.

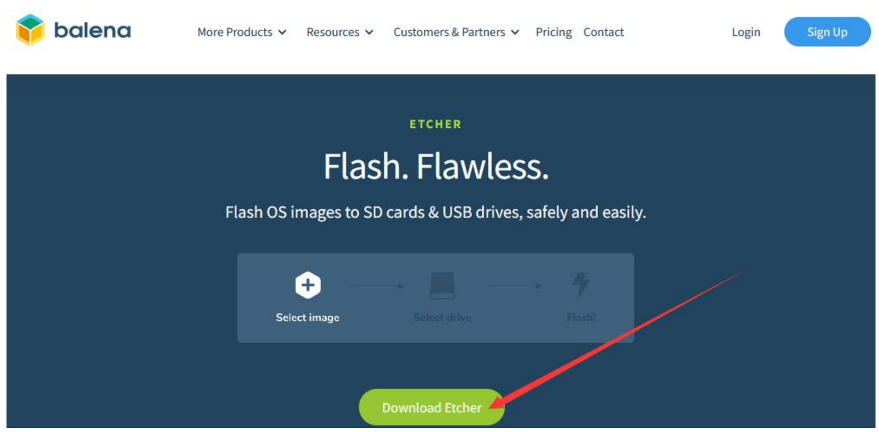
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brand

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

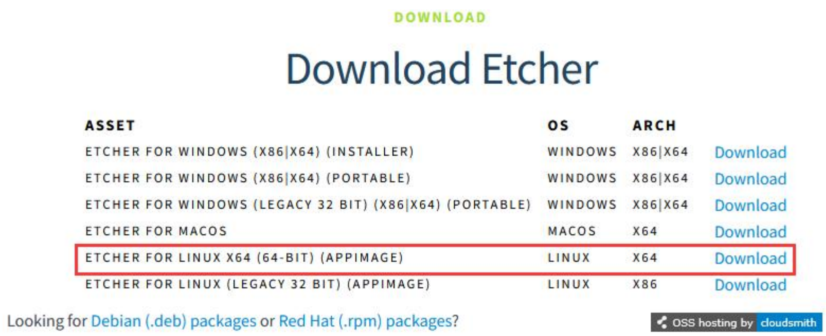
3) Download the balenaEtcher software, the download address is:

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

4) After entering the BalenaEtcher download page, clicking the green download button will redirect you to the software download location

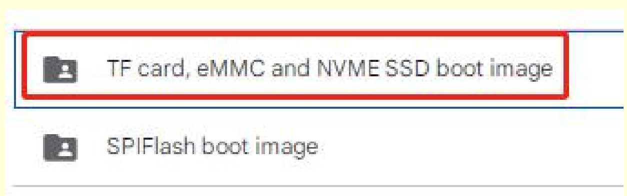


5) Then choose to download the Linux version of the software



6) Download the compressed file of the Linux operating system image that you want to burn from **Orange Pi's information download page**, and then use decompression software to extract it. In the extracted file, the file ending in "**.img**" is the operating system image file, which is generally over 2GB in size

Note that if you are downloading 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 "TF Card, eMMC, and NVME SSD Boot Image" folder.



The decompression command for the compressed file ending in 7z is as follows

```
test@test:~$ 7z x Orangepi5propro_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
test@test:~$ ls Orangepi5pro_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.*
Orangepi5pro_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
Orangepi5pro_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.sha       #Checksum file
Orangepi5pro_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img    #Image file
```

If you are downloading an OpenWRT image and the compressed file ends in gz, the decompression command is as follows

```
test@test:~$ gunzip openwrt-aarch64-opi5pro-24.03-linux-6.1.43-ext4.img.gz
test@test:~$ ls openwrt-aarch64-opi5pro-24.03-linux-6.1.43-ext4.img
openwrt-aarch64-opi5pro-24.03-linux-6.1.43-ext4.img     #Image file
```

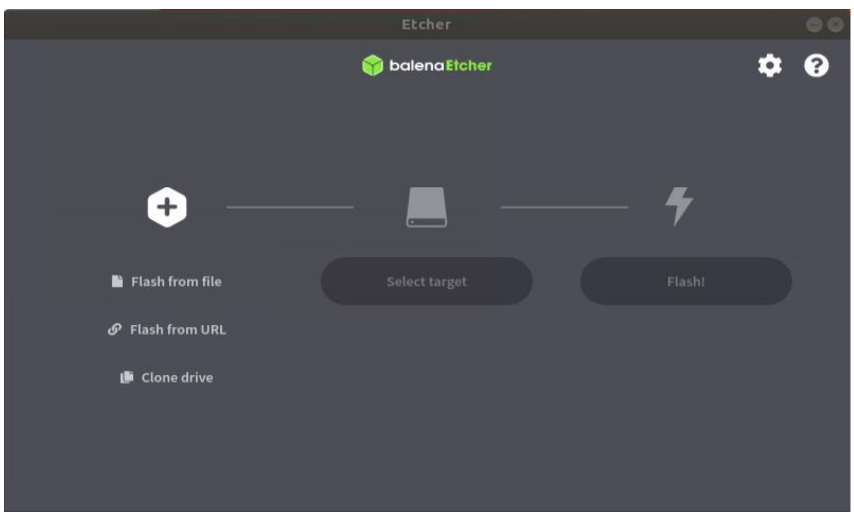
7) After decompressing the image, you can first use the **sha256sum -c *.sha** command to calculate whether the checksum is correct. If the prompt is **successful**, it indicates that the downloaded image is not incorrect. You can rest assured to burn it to the TF card. If the prompt is that the **checksum does not match**, it indicates that there is a problem with the downloaded image. Please try downloading again

```
test@test:~$ sha256sum -c *.sha
Orangepi5pro_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img: OK
```

If you are downloading an OpenWRT image, you need to verify the compressed file instead of decompressing it before verifying it

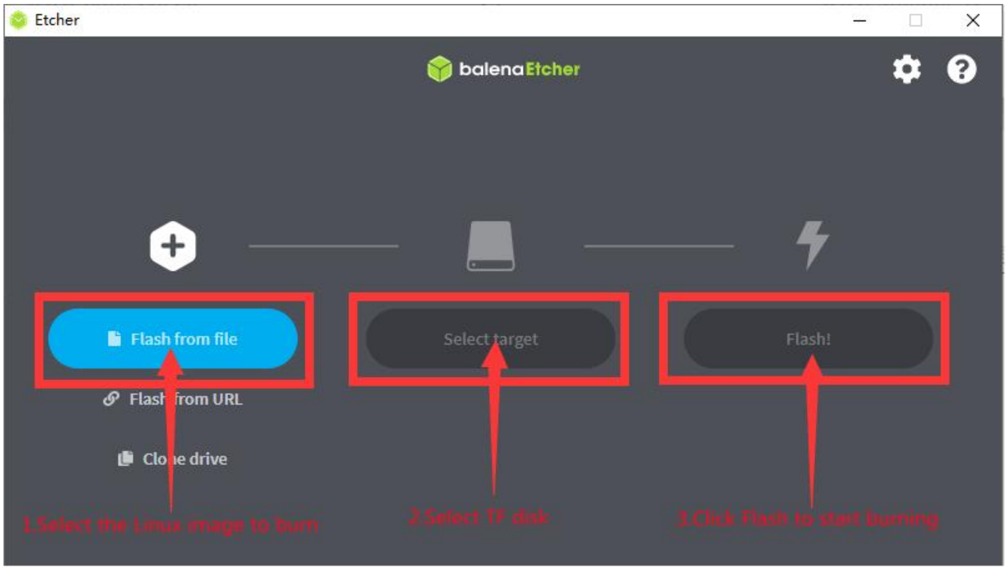
```
test@test:~$ sha256sum -c openwrt-aarch64-opi5pro-24.03-linux-6.1.43-ext4.img.gz.sha
openwrt-aarch64-opi5pro-24.03-linux-6.1.43-ext4.img.gz: OK
```

8) 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

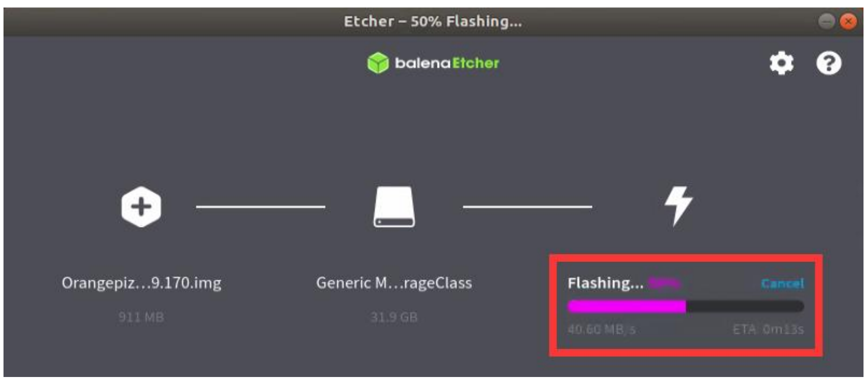


9) The specific steps to use balenaEtcher to burn the Linux image are as follows

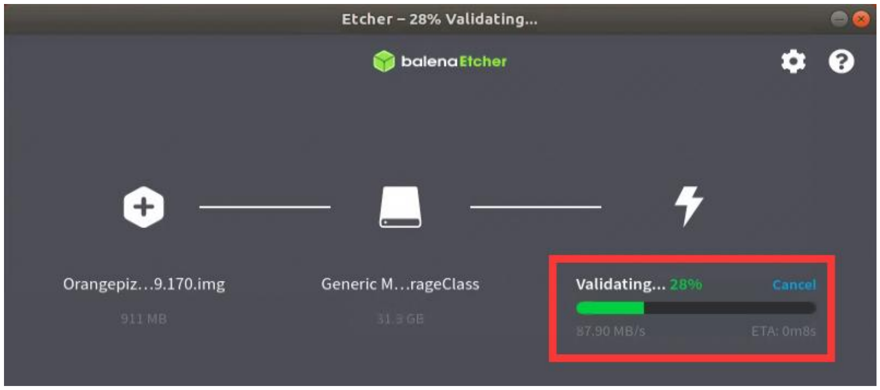
a. First select the path of the Linux image file to be burned
b. Then select the drive letter of the TF Card
c. Finally, click Flash to start burning the Linux image to the TF Card



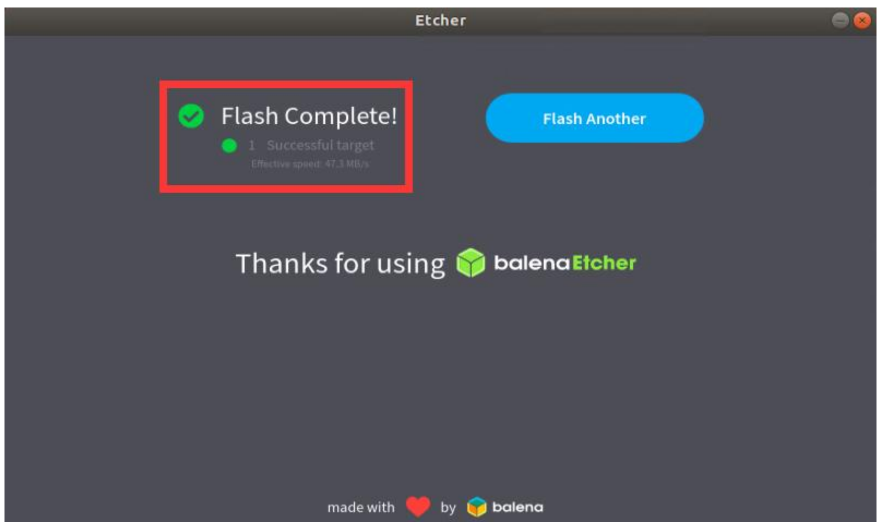
10) The interface displayed during the process of burning a Linux image with BalenaEtcher is shown in the following figure. In addition, the progress bar displays purple, indicating that the Linux image is being burned to the TF card



12) 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.



13) 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. The method of burning Linux images into eMMC

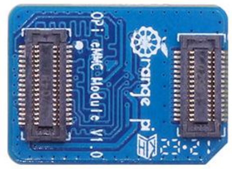
2. 5. 1. Method of burning Linux images into eMMC using RKDevTool

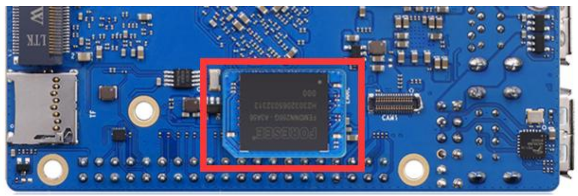
Note that all the operations below are performed on a Windows computer.

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

1) The development board has reserved an extension interface for the eMMC module.

Before burning the system to eMMC, it is necessary to first purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC module onto the development board. The eMMC module and the method of inserting the development board are as follows:

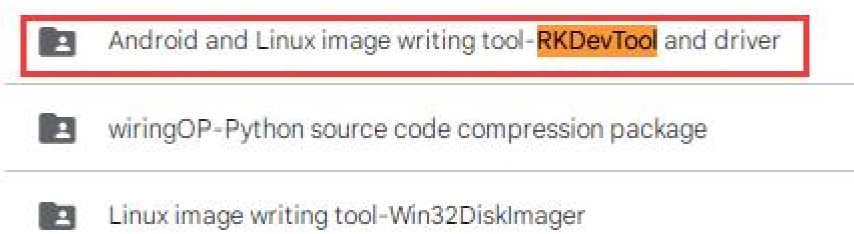




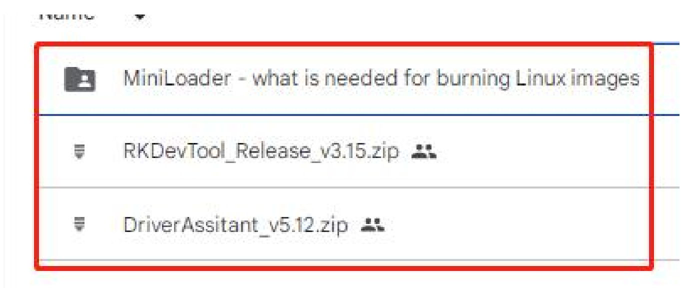
2) Also need to prepare a good quality USB2.0 public to public data cable

3) Then download the Ruixin micro driver **DriverAssitant_v5.12.zip** and **MiniLoader**, as well as the burning tool **RKDevTool_Release_v3.15.zip**, from **Orange Pi's data download page**

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



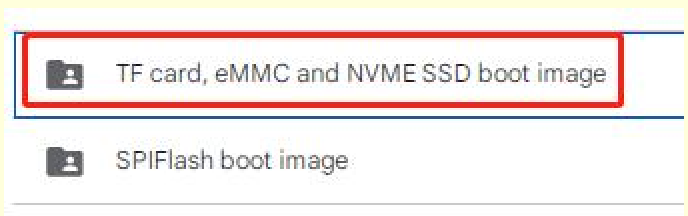
b. Then download all the files below



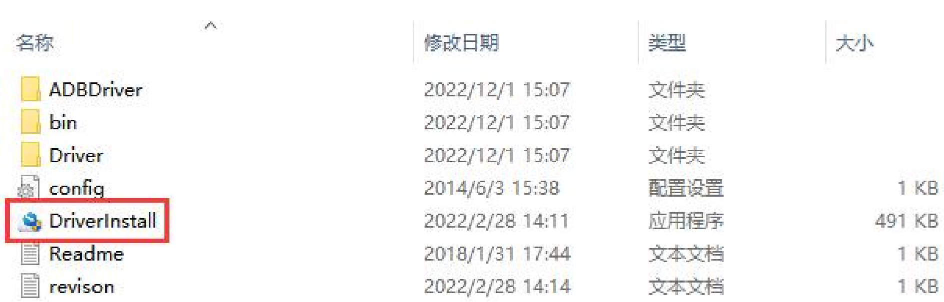
Note that the "MiniLoader - something needed to burn Linux images" folder is hereinafter referred to as the MiniLoader folder.

4) Then download the compressed file of the Linux operating system image that you want to burn from **Orange Pi's information download page**, and use decompression software to extract it. In the extracted file, the file ending in "**.img**" is the operating system image file, which is generally over 2GB in size

Note that if you are downloading an OpenWRT image, you will see the following two types of images in the download link of the OpenWRT image. Please download the image files from the "TF Card, eMMC, and NVME SSD Boot Images" folder.

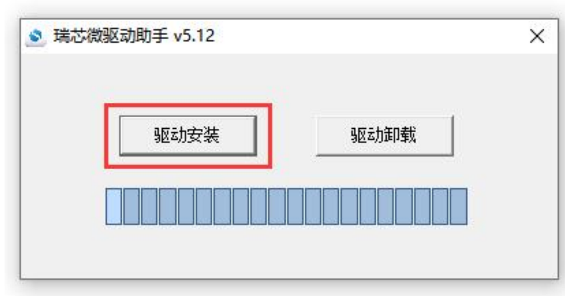


5) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

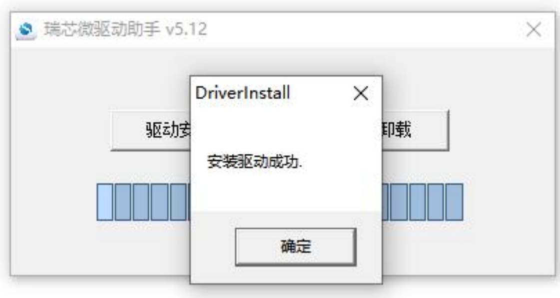


6) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

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

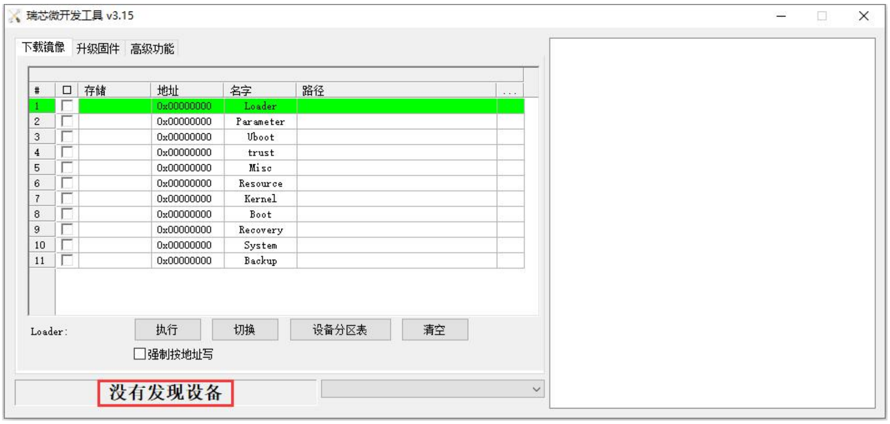


b. After waiting for a while, a pop-up window will prompt "**driver installed successfully**", and then click the "**OK**" button.



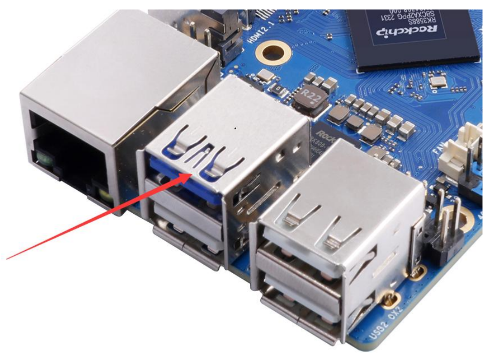
7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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, the lower left corner will prompt "**No device found**"



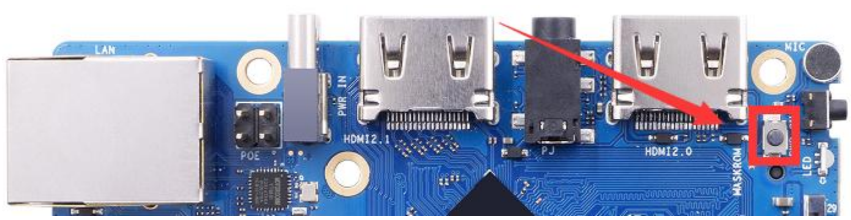
9) Then start burning Linux images into eMMC

a. Firstly, connect the development board to the Windows computer through the USB2.0 public-to-public data cable. The location of the USB burning port on the development board is shown in the figure below

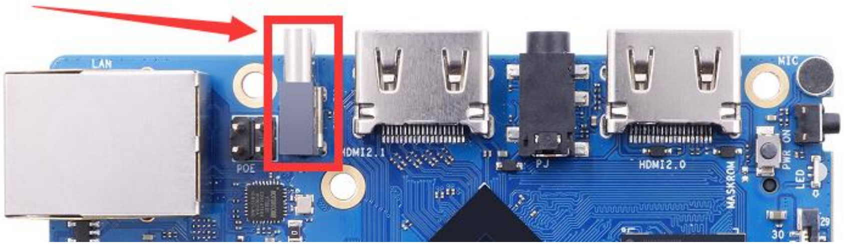


b. Ensure that the development board is not inserted with a TF card or connected to a power source

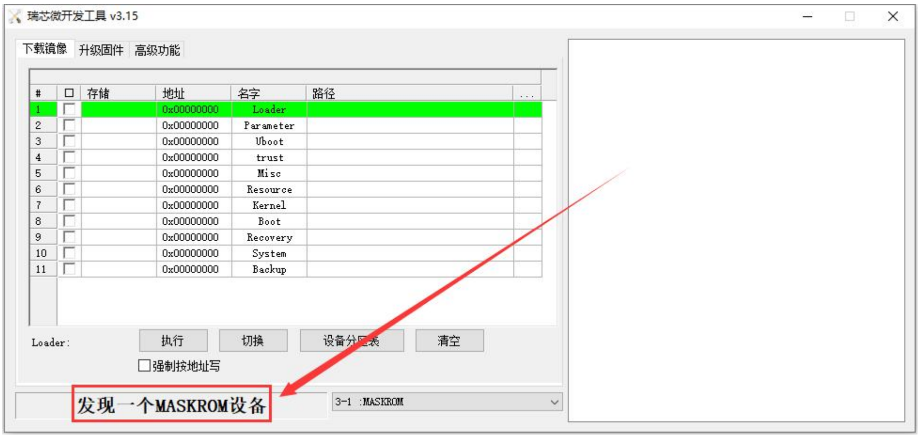
c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:



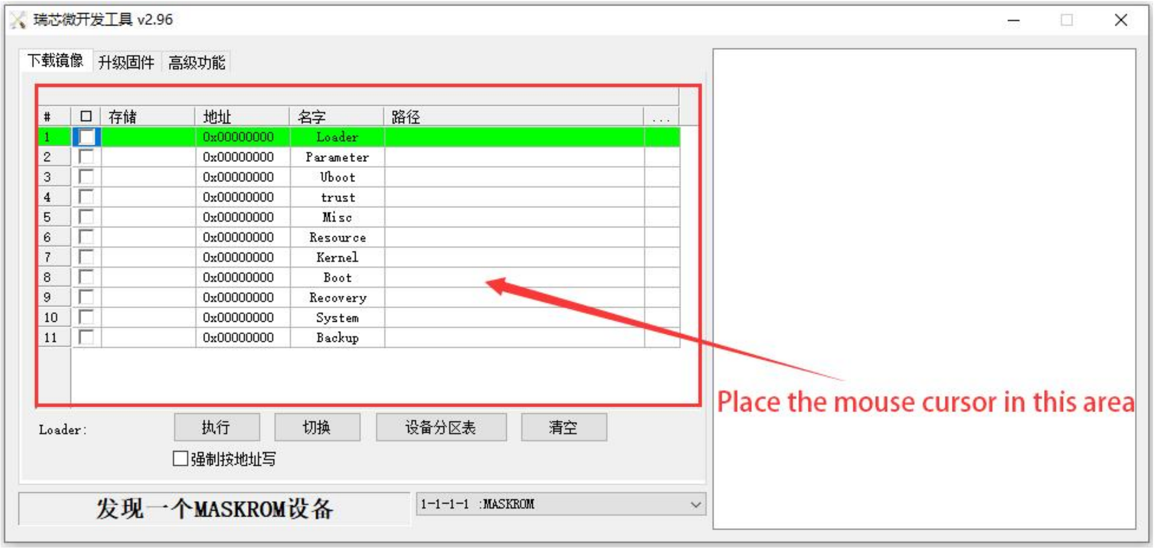
d. Then connect the Type-C interface power supply to the development board and power it on, then you can release the MaskROM button



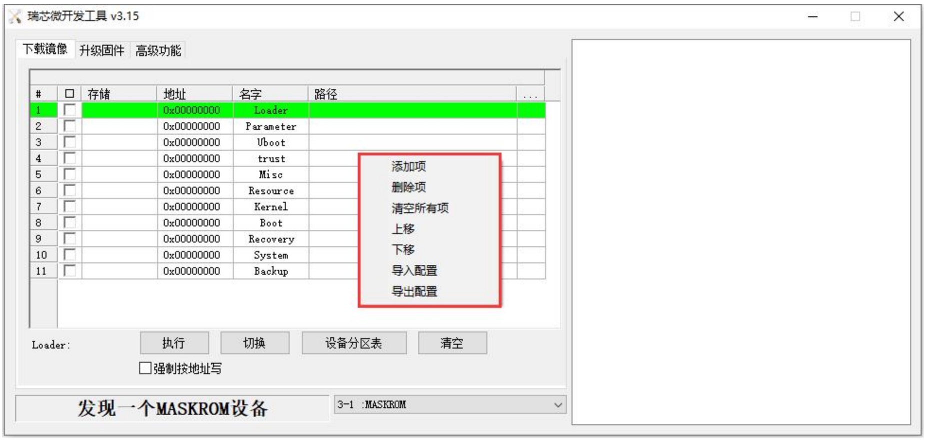
e. If the previous steps are successful, the development board will enter **MASKROM** mode and the interface of the burning tool will prompt "**Found a MASKROM device**"



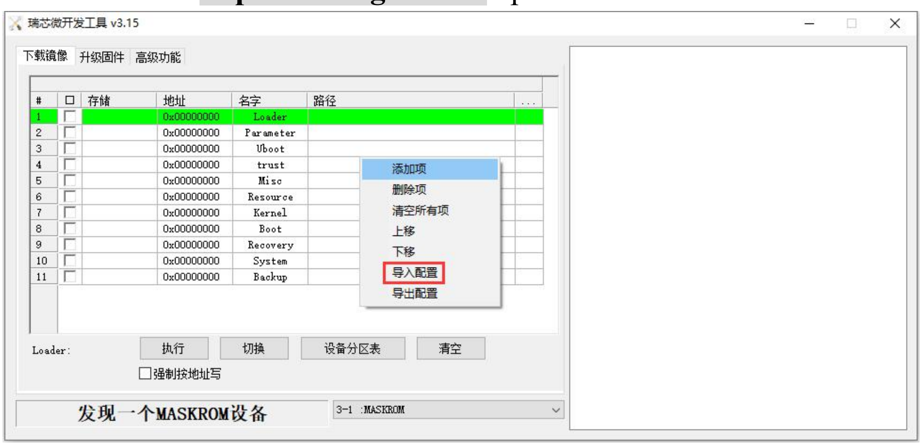
f. Then place the mouse cursor in the area below



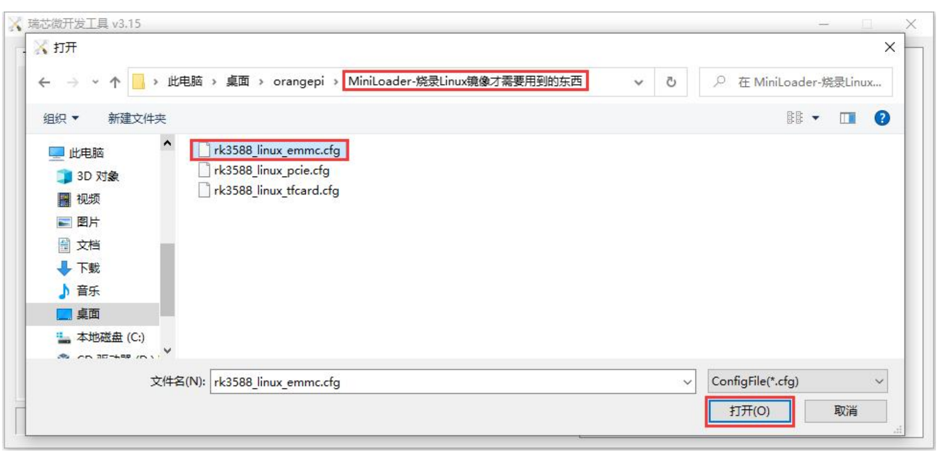
g. Then, clicking the right mouse button will pop up the selection interface shown in the following figure



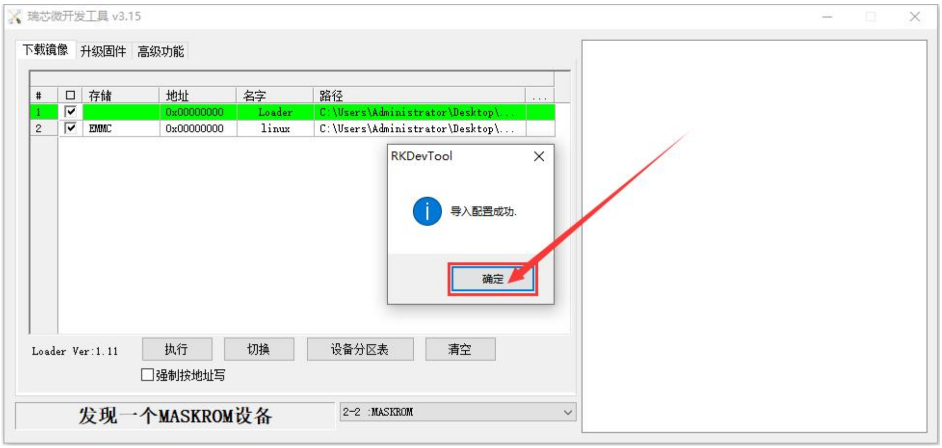
h. Then select the **import configuration** option



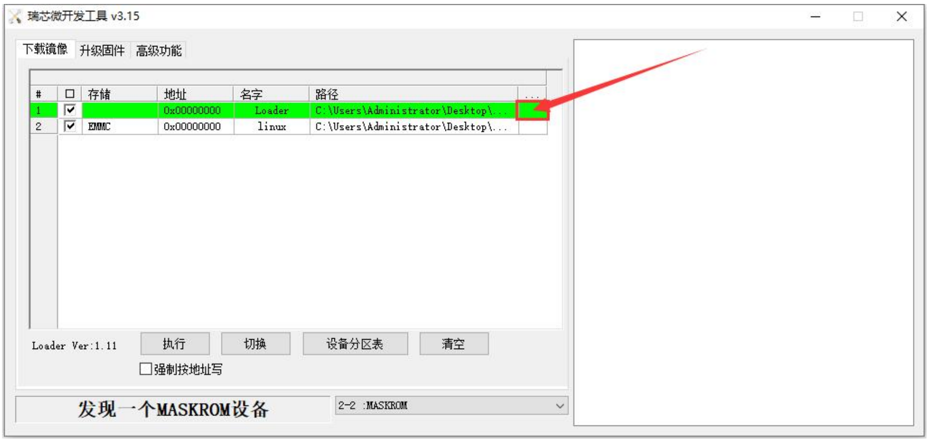
i. Then select the **rk3588_linux_emmc.cfg** configuration file from the **MiniLoader** folder downloaded earlier, and click to **open**



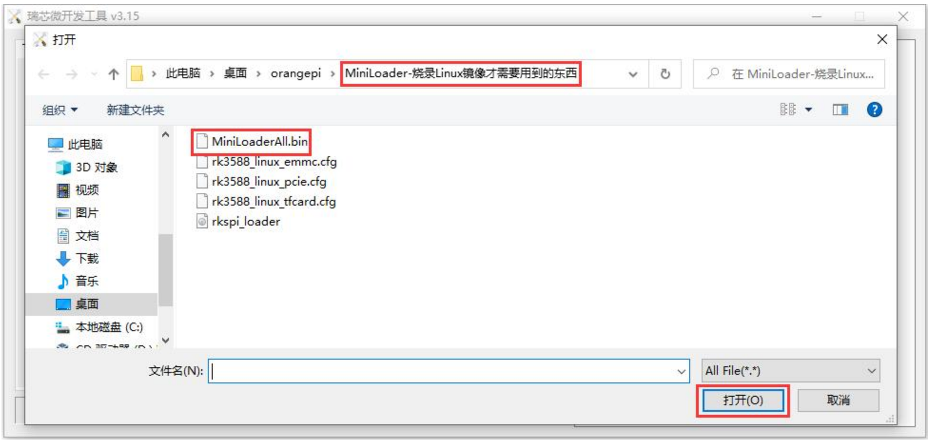
j. Then click **OK**



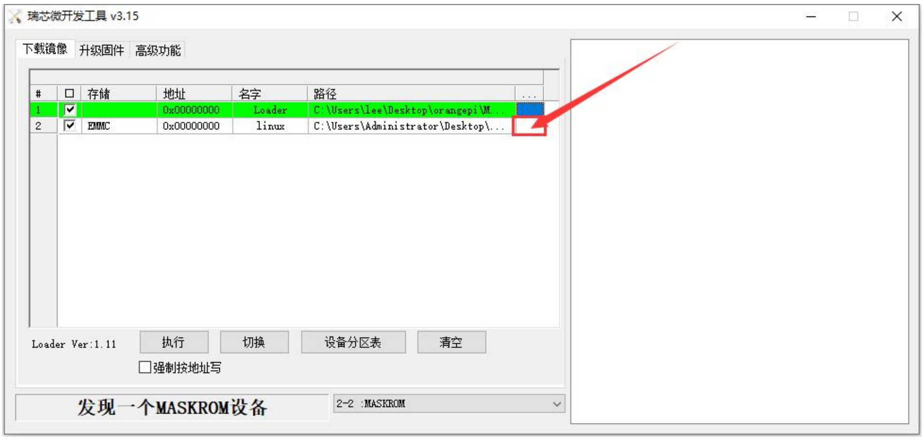
k. Then click on the location shown in the following image



l. Select **MiniLoaderAll.bin** from the **MiniLoader** folder downloaded earlier, and then click **open**

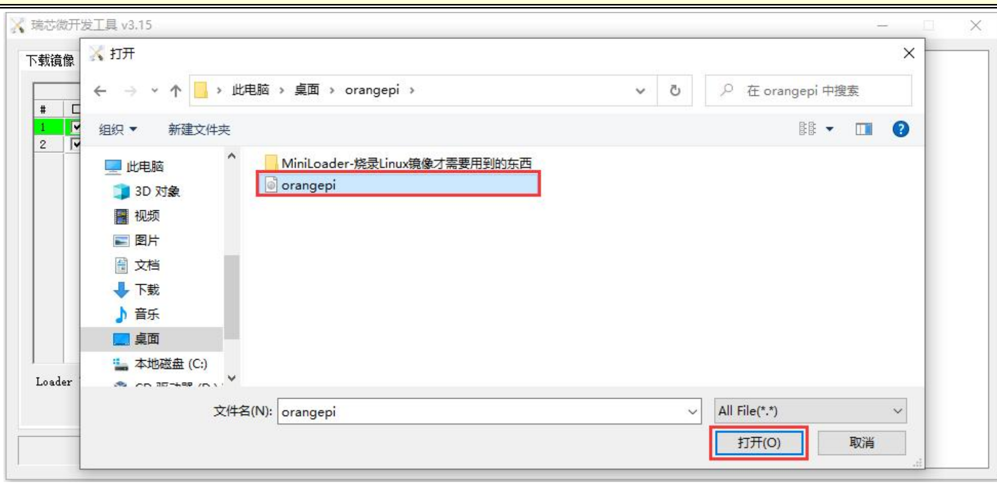


m. Then click on the location shown in the following image

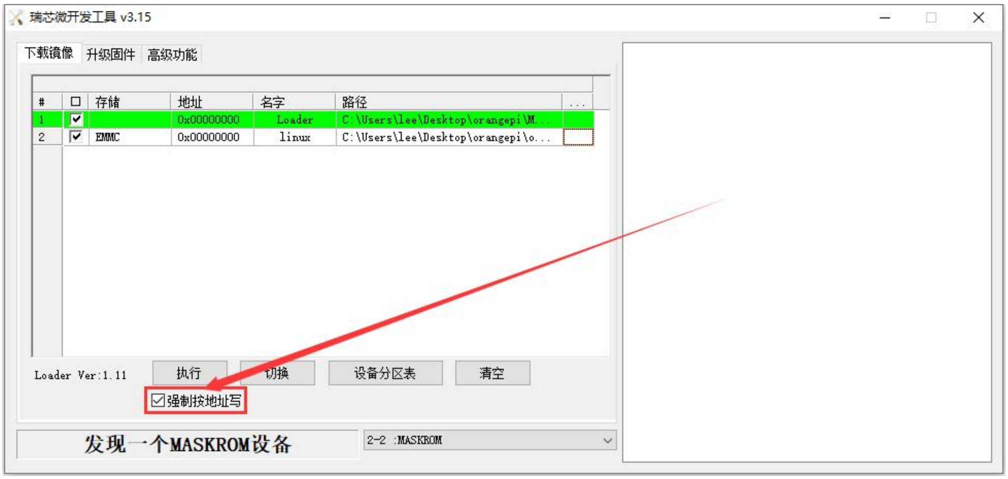


n. Then select the path to the Linux image you want to burn, and click **open**

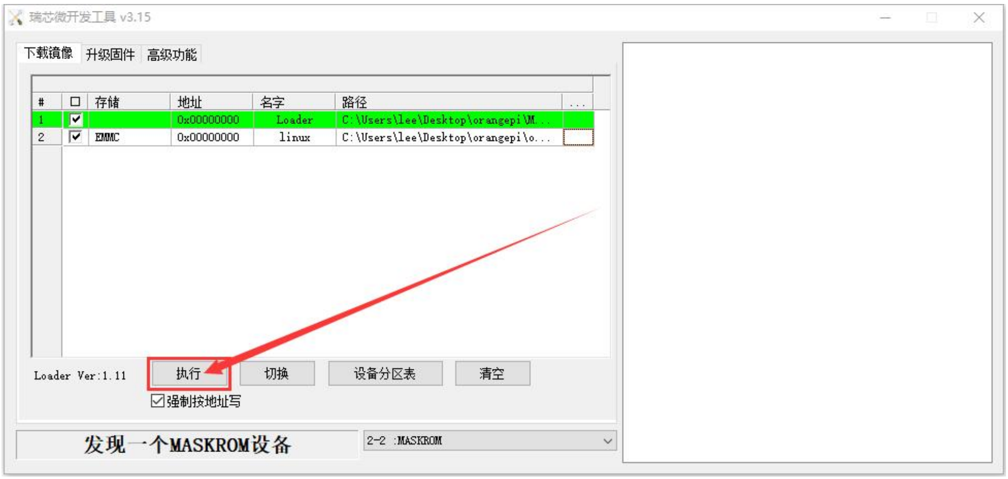
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



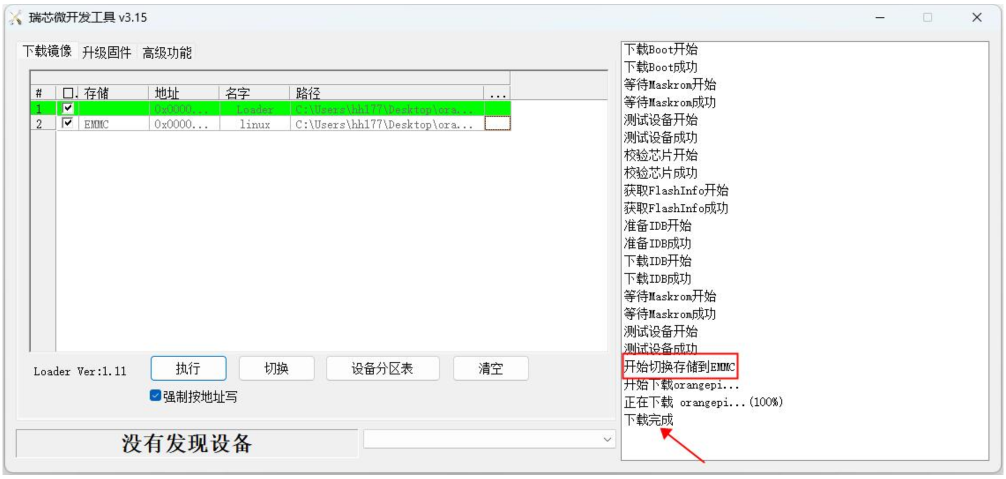
o. Then please check the option to **force writing by address**



p. Clicking the execute button again will start burning the Linux image to the eMMC of the development board



q. The displayed log after burning the Linux image is shown in the following figure



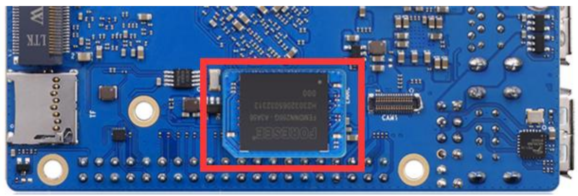
r. After burning the Linux image into eMMC, the Linux system will automatically start.

2. 5. 2. The method of burning Linux images into eMMC using the dd command

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

1) The development board reserves the expansion interface of the eMMC module. Before burning the system to the eMMC, you first need to purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC

module to the development board. The eMMC module and the method of plugging into the development board are as follows:



2) Using the dd command to burn the linux image to eMMC needs to be done with 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.**

3) After using the TF card to start the linux system, we first upload the decompressed linux image file (Debian, Ubuntu image or OPi Arch 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.

4) 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
Orangepi5pro_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

How to enter the command line of the Linux system on the development board?

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 a display screen such as HDMI or LCD is connected, you can open a command line terminal on the desktop.

5) Next, we first use the following command to confirm the device node of eMMC

```
orangepi@orangepi:~/Desktop$ ls /dev/mmcblk*boot0 | cut -c1-12
/dev/mmcblk1
```

6) Then we can use the dd command to clear the eMMC. Note that after the **of=** parameter, please fill in the output result of the above command

```
orangepi@orangepi:~/Desktop$ sudo dd bs=1M if=/dev/zero of=/dev/mmcblk1 count=1000 status=progress
orangepi@orangepi:~/Desktop$ sudo sync
```

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

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=Orangepi5pro_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/mmcblk1 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.

8) After successfully burning the linux image of the development board to the eMMC, 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 eMMC will be started.

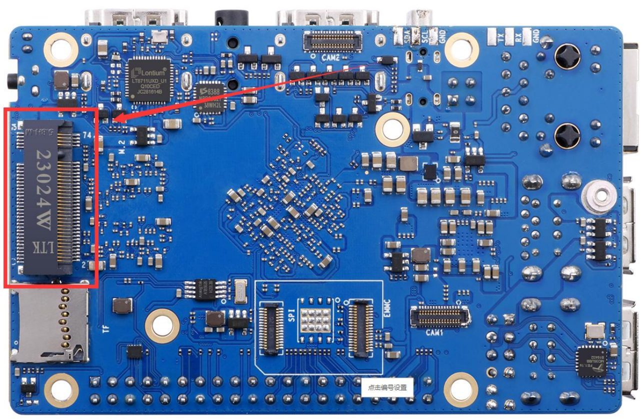
2. 6. Method of burning Linux image to TF card+NVMe SSD

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

The method introduced in this chapter requires burning the Bootloader file into a TF card. In this way, even if the development board does not have SPI Flash attached, the system in NVMe can still be booted through the TF card.

1) Firstly, it is necessary to prepare an NVMe SSD solid-state drive. The development board M.2 slot supports PCIe 2.0x1, with a theoretical maximum speed of 500MB/s. NVMe SSDs for PCIe 3.0 and PCIe 4.0 are also usable, but the maximum speed is only PCIe 2.0x1.

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



3) Burning a Linux image to a TF card+NVMe SSD requires the use of a TF card, so the

first step is to burn the Linux image to the TF card, and then use the TF card to launch the development board and enter the Linux system. The method of burning Linux images to TF cards can be found in the user manual's two sections: **the method of burning Linux images to TF cards based on Windows PC and the method of burning Linux images to TF cards based on Ubuntu PC**.

4) Then upload the Linux image file (Debian, Ubuntu, or OpenWRT image downloaded from the official website) to the TF card. For the method of uploading Linux image files to the development board, please refer to the instructions in the section on **uploading files to the development board Linux system** in the user manual.

5) After uploading the image to the Linux system of the development board, we can enter the storage path of the image file in the command line of the Linux system of the development board. For example, I stored the Linux image of the development board in the **/home/orangepi/Desktop** directory, and then enter the **/home/orangepi/Desktop** directory to see the uploaded image file.

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepi5pro_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

How to enter the command line of the Linux system on the development board?

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

2. For remote login to Linux system using SSH, please refer to the instructions in the SSH remote login development board section of the user manual.

3. If an HDMI, LCD or other display screen is connected, a command line terminal can be opened on the desktop.

6) Next, let's confirm that the NVMe SSD has been properly recognized by the linux development board. If the NVMe SSD recognizes it properly, using the **sudo fdisk -l** command can display information related to **nvme**

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

Using the **lspci** command, you can 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)
```

7) Then we can use the dd command to clear the NVMe SSD (optional)

```
orangepi@orangepi5pro:~/Desktop$ sudo dd bs=1M if=/dev/zero of=/dev/nvme0n1 count=2000 status=progress
orangepi@orangepi5pro:~/Desktop$ sudo sync
```

8) Then you can use the dd command to burn the Linux image of the development board to NVMe SSD

a. In the following command, after the **if=** parameter, you need to fill in the complete path where the Linux image is stored+the name of the Linux image (such as **/home/orangepi/Desktop/Linux image name**). Because we have already 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, replace it with the actual image name (as the version number of the image may be updated).

```
sudo dd bs=1M if=Orangepi5pro_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/nvme0n1 status=progress

sudo sync
```

Note that if you are uploading a . 7z,. xz, or. gz Linux image compressed file, please remember to decompress it before using the dd command to burn.

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

9) Then download **rkspi_loader.img** from **Orange Pi's data download page** and upload it to the development board

10) After successfully burning the Linux image of the development board to the NVMe SSD, you need to use the following command to burn **rkspi_loader.img** to the TF card. Note that after the burning is completed, the TF card will only boot with uboot, and the kernel and file system will be destroyed. At this time, it is impossible to start the system with only the TF card inserted.

```
sudo dd if=rkspi_loader.img of=$(findmnt -n -o SOURCE / | sed 's/..$//')
```

11) Then use the **poweroff** command to shut down the device, and briefly press the power button to turn it on. This will start the Linux system in TF Card+NVMe SSD.

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

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
```

2. 7. Method of burning Linux images to SPIFlash+NVMe SSD

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

Note that even without the SPI Flash chip attached, the TF card can still be used to boot the system in NVMe. For specific methods, please refer to **2.6. Method of burning Linux images to TF card+NVMe SSD**

The method introduced in this chapter requires burning the Bootloader file into SPI Flash. Before starting to burn the image, it is necessary to ensure that the development board has already attached the SPI Flash chip, because the development board does not have an SPI Flash chip attached at the factory, so it needs to be purchased and soldered on. We recommend using the SPI Flash chip model XM25QU128CWIQT08Q

Due to SPI Flash and eMMC reusing the same pins, eMMC cannot be used with SPI Flash attached

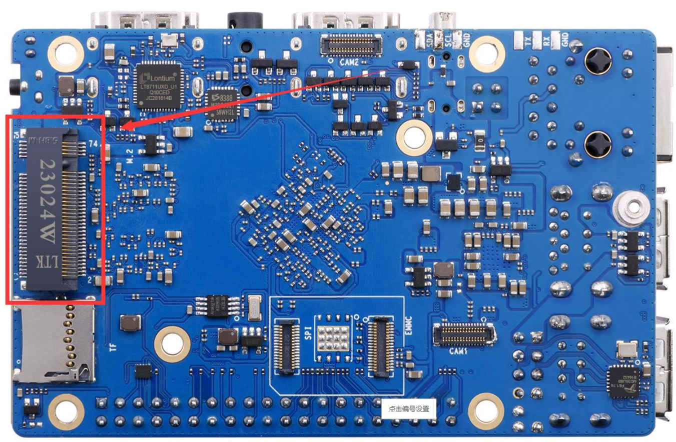
Note that all the following operations are performed on a Windows computer.

2. 7. 1. The method of burning using RKDevTool

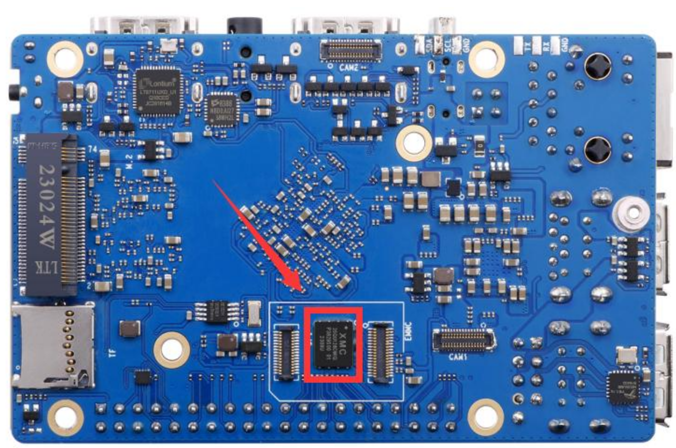
1) Firstly, it is necessary to prepare an NVMe SSD solid-state drive. The development

board M.2 slot supports PCIe 2.0x1, with a theoretical maximum speed of 500MB/s. NVMe SSDs for PCIe 3.0 and PCIe 4.0 are also usable, but the maximum speed is only PCIe 2.0x1.

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



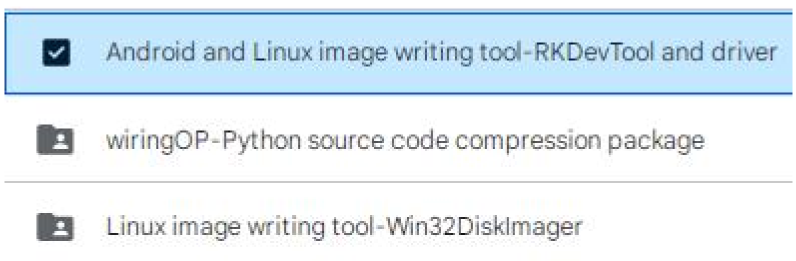
3) Please ensure that the development board has already been labeled with SPI Flash. The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting the burning process



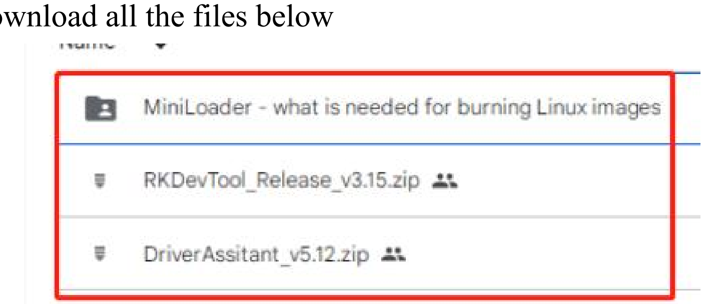
4) Then you need to prepare a high-quality USB 2.0 corporate data cable

5) Then download the Ruixin micro driver **DriverAssitant_v5.12.zip** and MiniLoader, as well as the burning tool **RKDevTool_Release_v3.15.zip**, from Orange Pi's 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

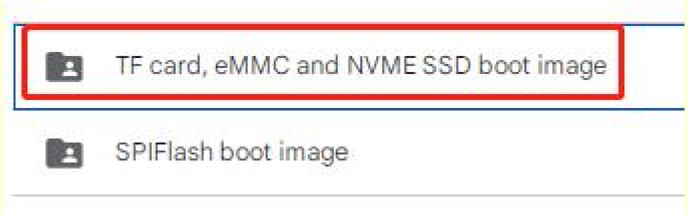


Note that the "MiniLoader - something needed to burn Linux images" folder is hereinafter referred to as the MiniLoader folder.

6) Then download the compressed file of the Linux operating system image that you want to burn from **Orange Pi's information download page**, and use decompression software to extract it. In the extracted file, the file ending in "**.img**" is the operating system image file, which is generally over 2GB in size

Note that if you are downloading an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the

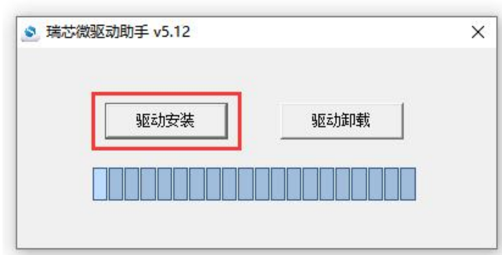
image file in the "TF Card, eMMC, and NVME SSD Boot Image" folder.



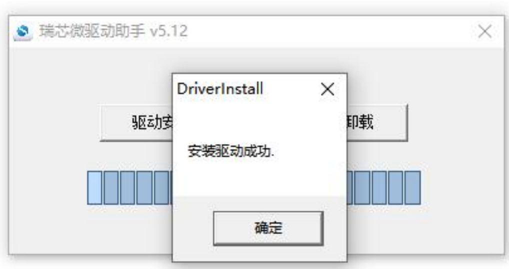
7) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

8) The steps to install the Ruixin micro driver after opening **DriverInstall.exe** are as follows

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

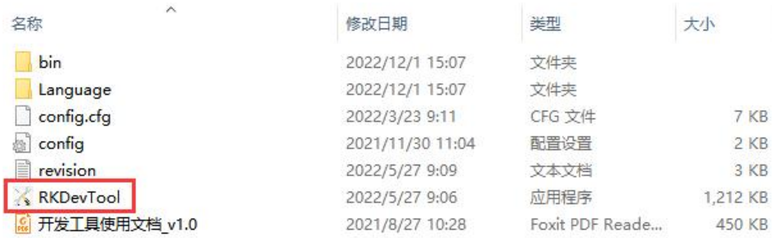


b. After waiting for a period of time, a pop-up window will prompt "**Driver installation successful**", and then click the "**OK**" button to proceed

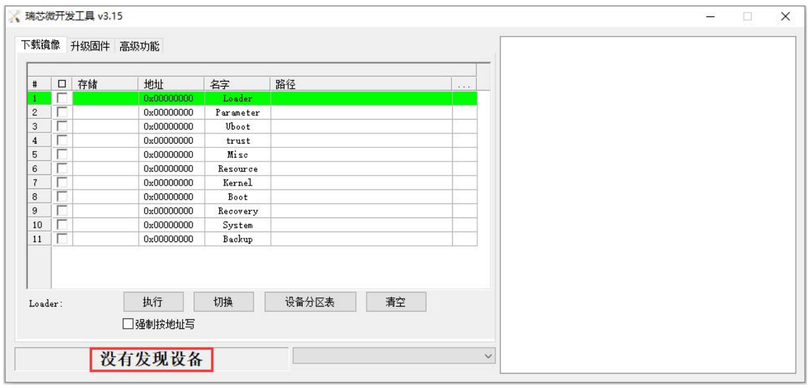


9) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be

installed. You can find **RKDevTool** in the unzipped folder and open it

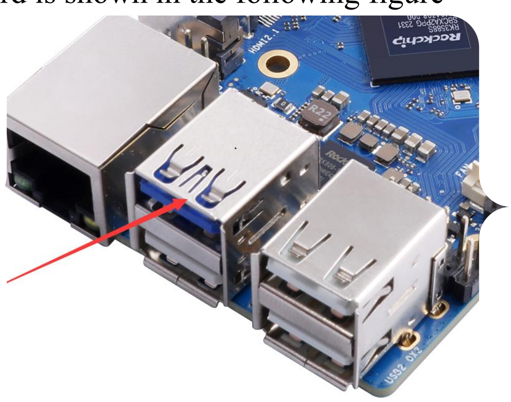


10) After opening the **RKDevTool** burning tool, because the computer has not yet been connected to the development board through a USB 2.0 male to female data cable, the bottom left corner will prompt "**No device found**"



11) Then start burning the Linux image to the SSD

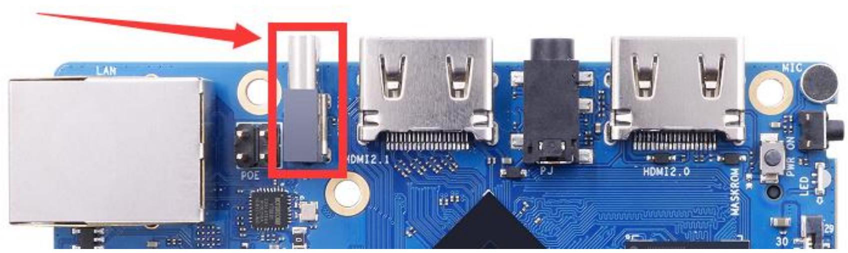
a. Firstly, connect the development board to the Windows computer through a USB 2.0 male to female data cable. The location of the USB flash port on the development board is shown in the following figure



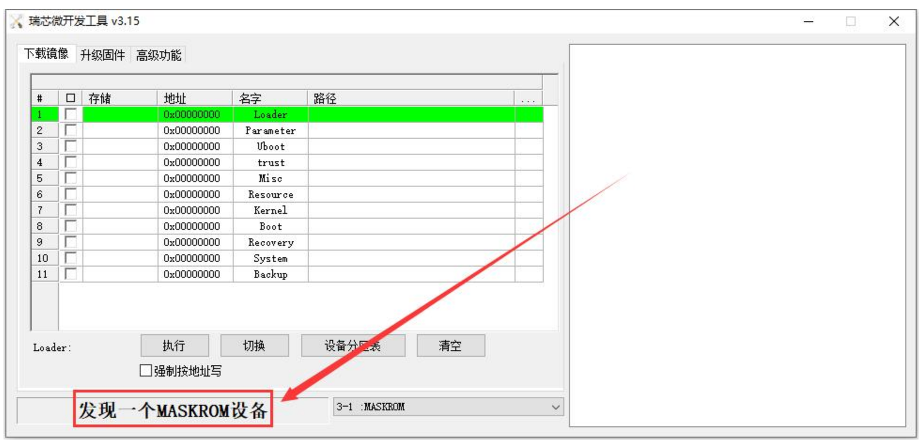
b. Ensure that the development board is not connected to a power source or inserted with a TF card

c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:

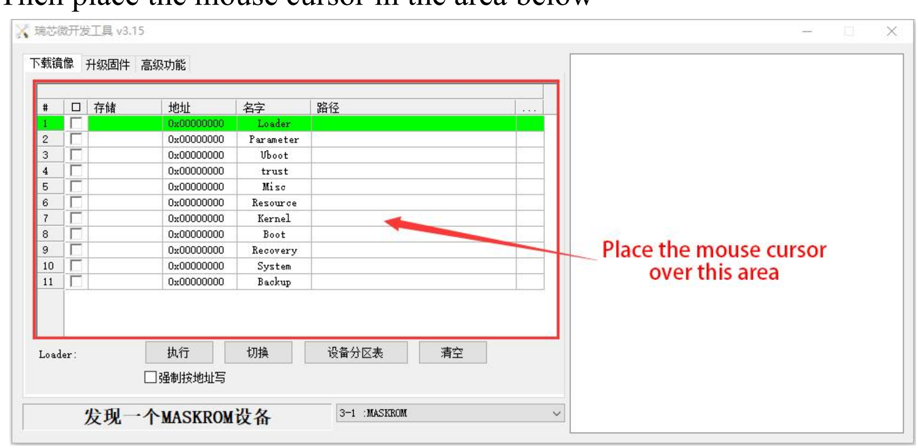
d. Then connect the Type-C interface power supply to the development board and power it on, then you can release the MaskROM button



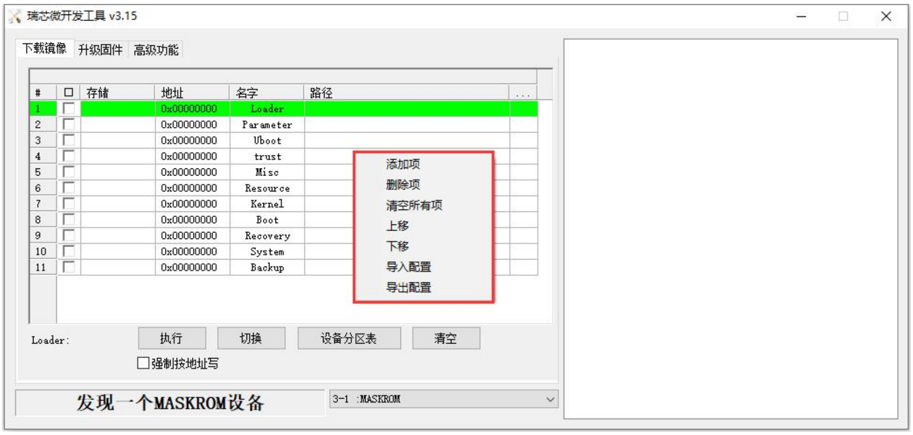
e. If the previous steps are successful, the development board will enter **MASKROM** mode and the interface of the burning tool will prompt "**Found a MASKROM device**"



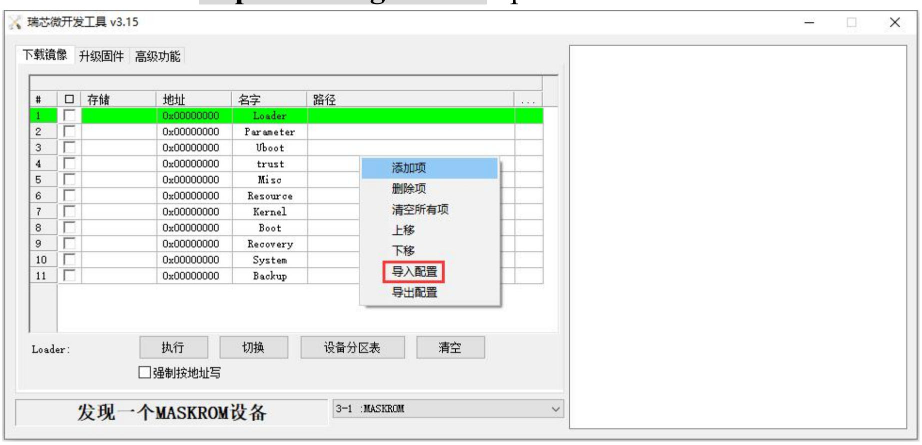
f. Then place the mouse cursor in the area below



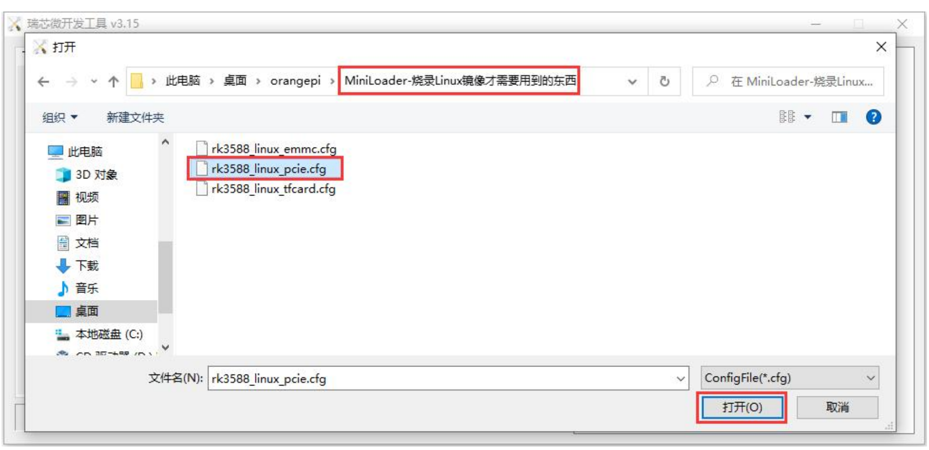
g. Then, clicking the right mouse button will pop up the selection interface shown in the following figure



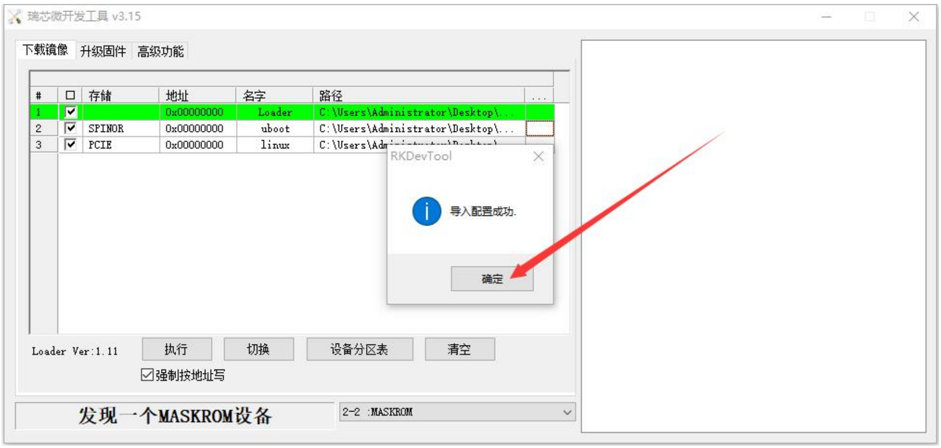
h. Then select the **import configuration** option



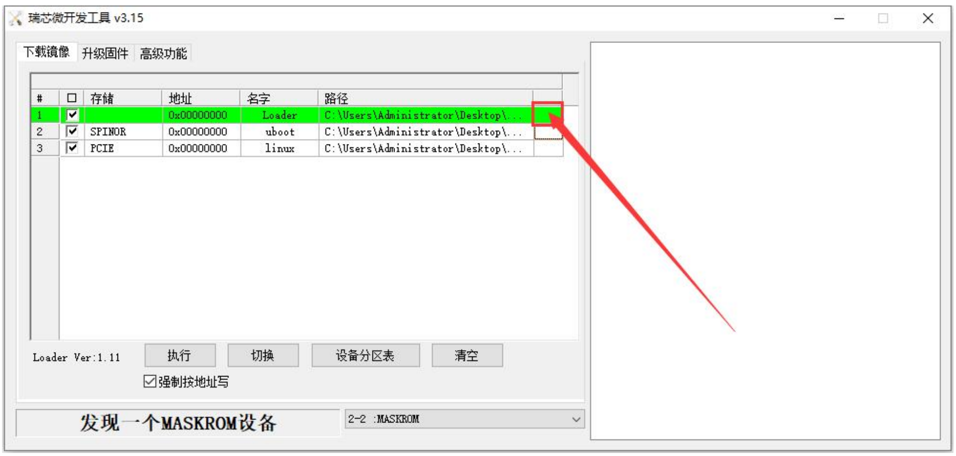
i. Then enter the MiniLoader folder downloaded earlier, select the **rk3588_linux_pcie.cfg** configuration file, and click **open**



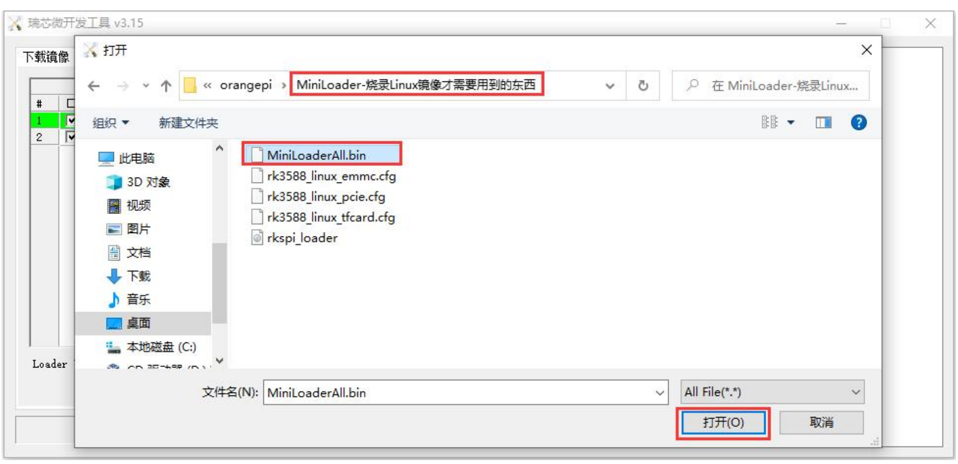
j. Then click **OK**



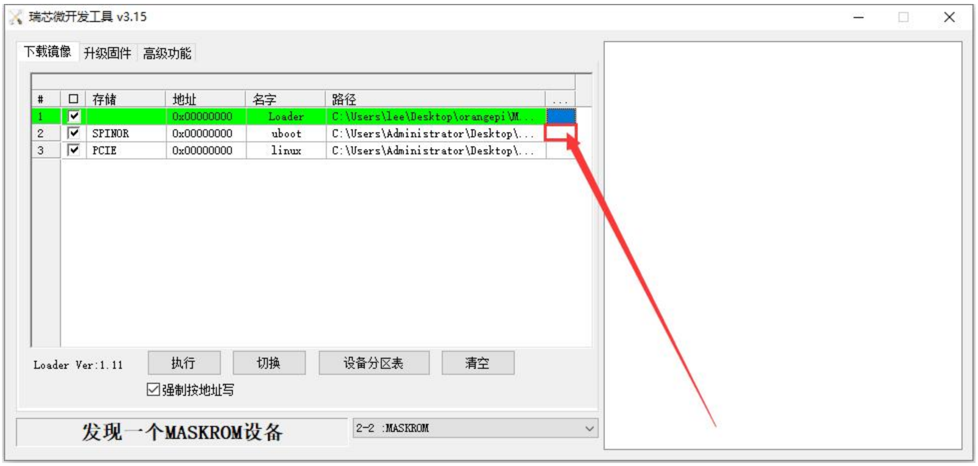
k. Then click on the location shown in the following image



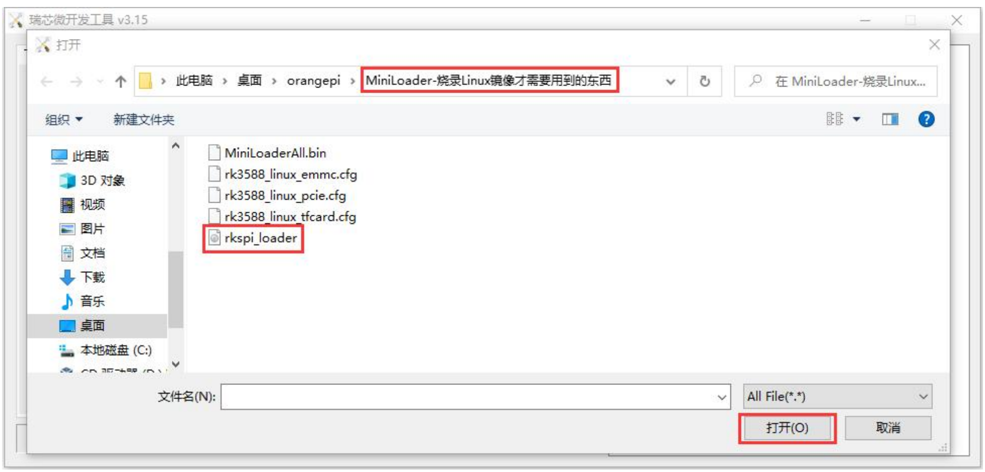
l. Select **MiniLoaderAll.bin** from the **MiniLoader** folder downloaded earlier, and then click **open**



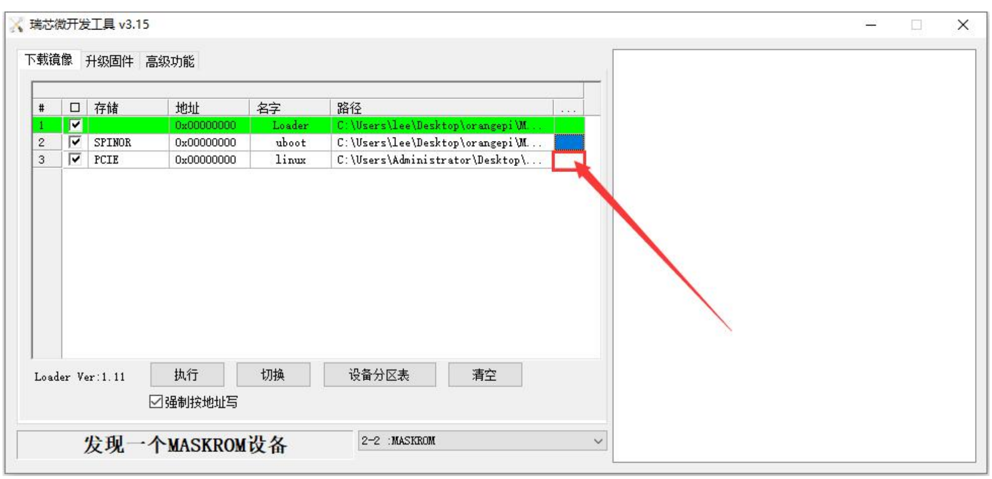
m. Then click on the location shown in the following image



n. Then enter the **MiniLoader** folder downloaded earlier, select **rkspi_loader.img**, and click **open**



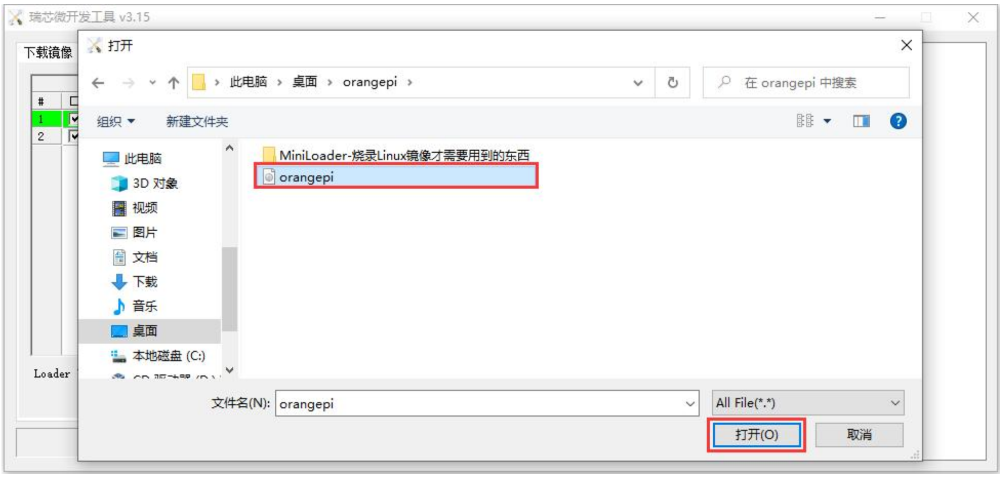
o. Then click on the location shown in the following image



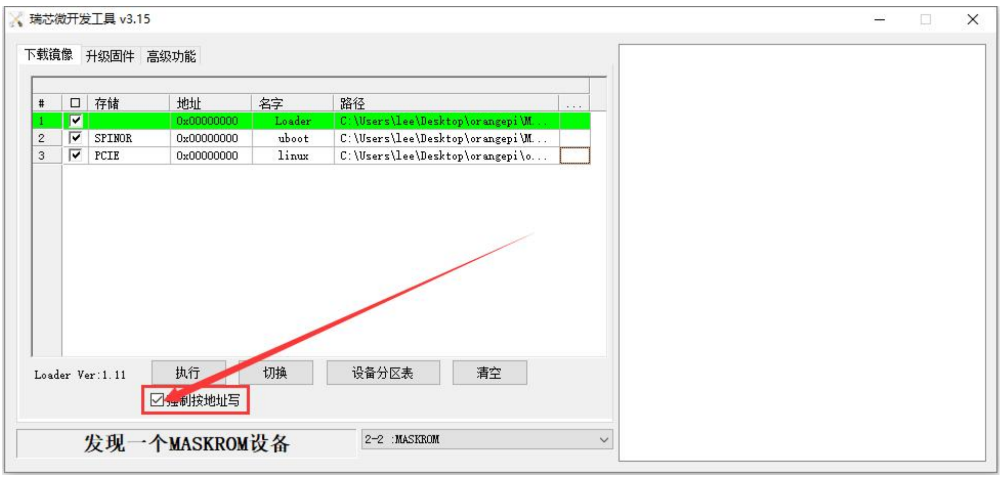
p. Then select the path to the Linux image you want to burn, and click **open**

Before burning the image, it is recommended to rename the Linux image to

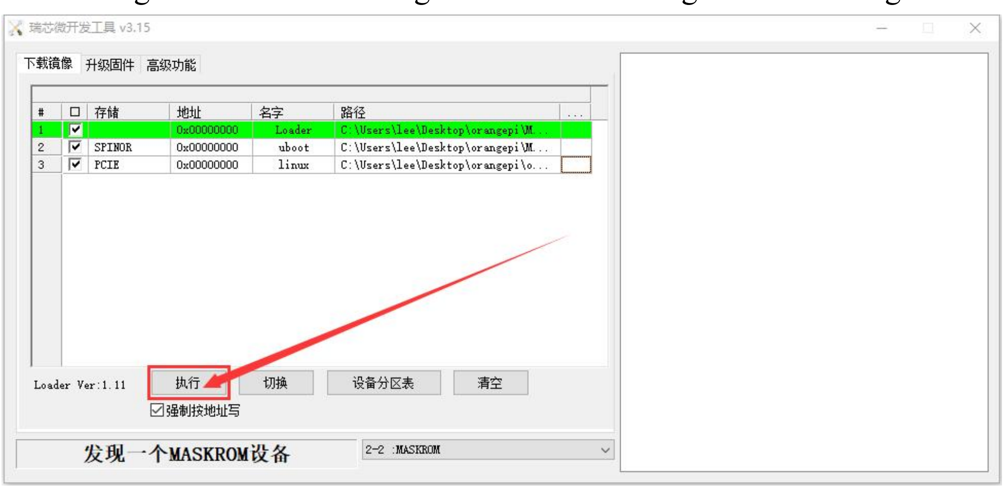
orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



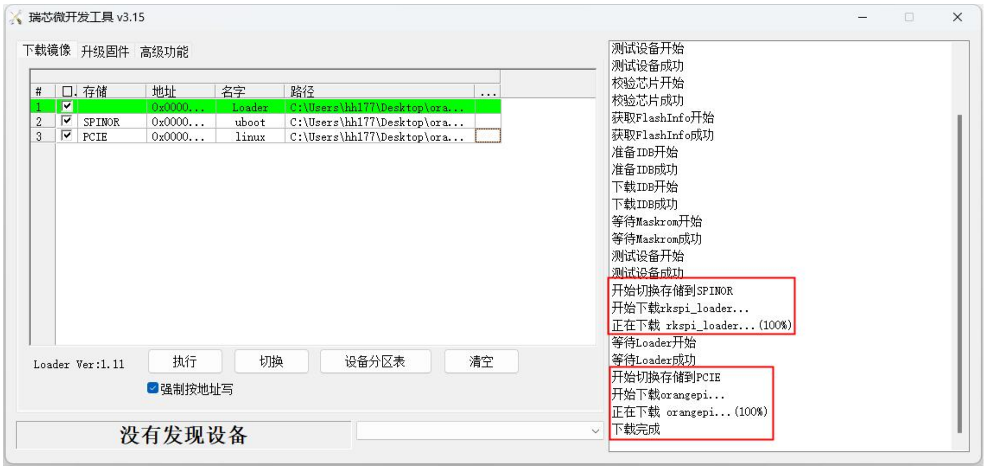
q. Then please check the option to **force writing by address**



r. Clicking the execute button again will start burning the Linux image to the SSD



s. The displayed log after burning the Linux image is shown in the following figure



If there is a problem with burning, please clear SPIFlash first and then try burning again. For the method of clearing SPIFlash, please refer to the instructions in the section on clearing SPIFlash using RKDevTool.

t. After the image is burned, the Linux system in SPIFlash+PCIe SSD will automatically start. If it does not start properly, please power on again and try again.

2. 7. 2. How to use the dd command to burn

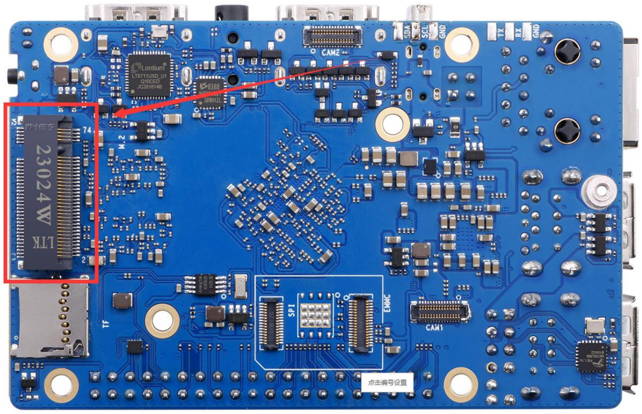
1) Firstly, it is necessary to prepare an NVMe SSD solid-state drive. The development board M.2 slot supports PCIe 2.0x1, with a theoretical maximum speed of 500MB/s. NVMe SSDs for PCIe 3.0 and PCIe 4.0 are also usable, but the maximum speed is only PCIe 2.0x1.

a. The M.2 2230 SSD is as follow

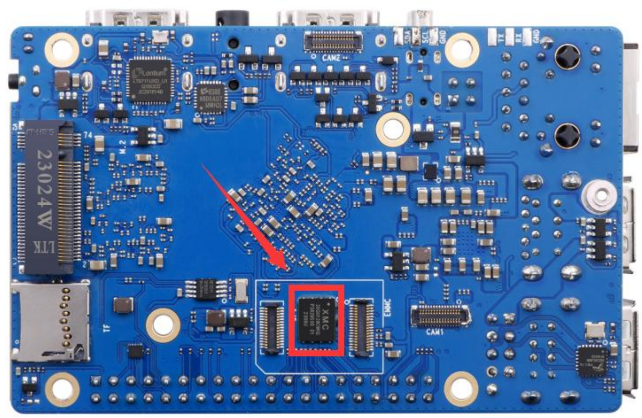
b. The M.2 2242 SSD is as follow

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

fix it



3) Please ensure that the development board has already been labeled with SPI Flash. The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting the burning process



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 Linux images to TF cards based on Windows PC** and **the method of burning Linux images to TF cards based on Ubuntu PC**.

5) After starting the Linux system with a TF card, we first burn the u-boot image to SPI Flash

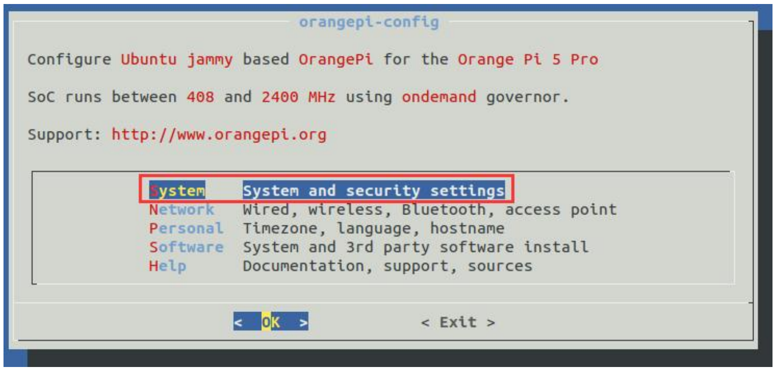
a. In Linux systems, the default SPI Flash feature is turned off and needs to be manually turned on to use. The detailed steps are as follows:

a) First, run orange pi configuration. **ordinary users remember to add sudo**

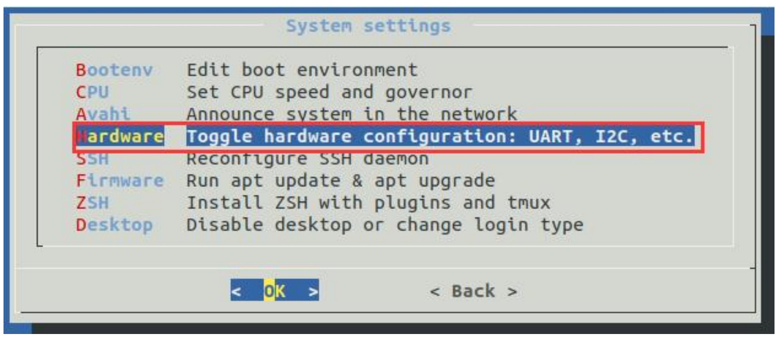
permission

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

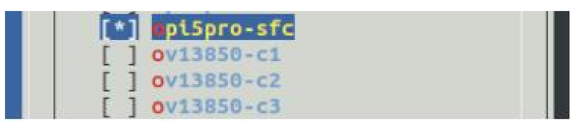
b) Then select **System**



c) Then select **Hardware**

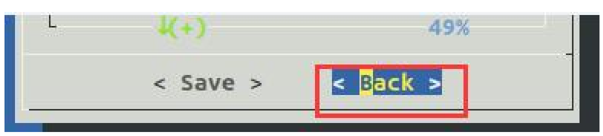


d) Then use the arrow keys on the keyboard to navigate to **opi5pro-sfc**, and use a space to select it

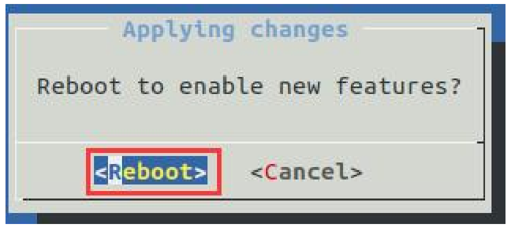


e) Then select**<Save>**to save

f) Then select**<Back>**



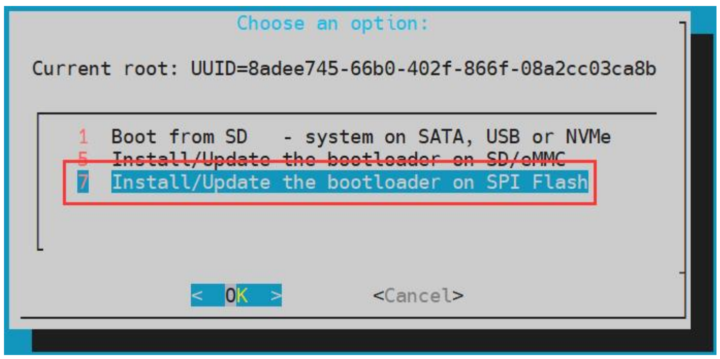
g) Then select**<Reboot>**to restart the system for the configuration to take effect



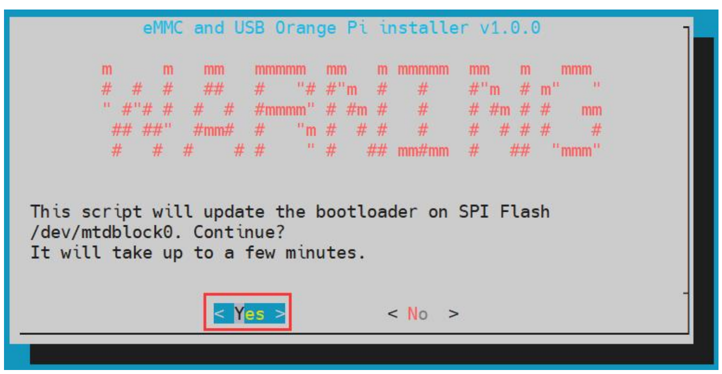
b. Then run **nand-sata-install**, **ordinary users remember to add sudo permission**

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

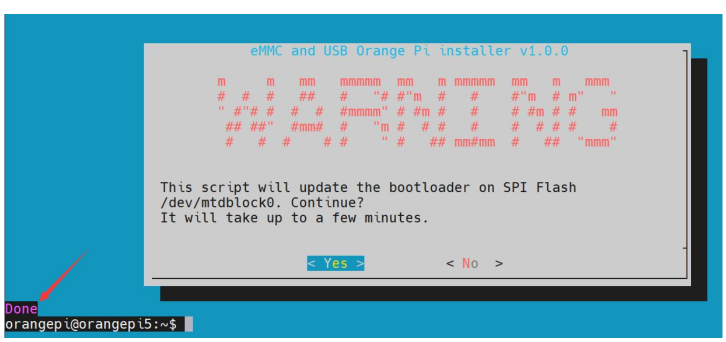
c. Then select **7 Install/Update ther bootloader on SPI Flash**



d. Then select **<Yes>**



e. Then please be patient and wait for the burning to complete. After the burning is completed, it will display as follows (a **Done** will be displayed in the bottom left corner):



The OPi OS Arch system does not have SPI Flash enabled by default and needs to be manually opened. Please add the following configuration in/boot/extLinux/extlinuxconf and restart the system to recognize SPI Flash

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf

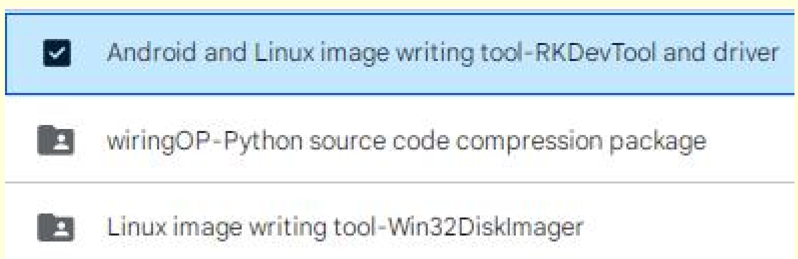
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-opi5pro-sfc.dtbo

The nand data install script is not available in the OPi OS Arch system. Please use the following command to mirror u-boot to SPI Flash:

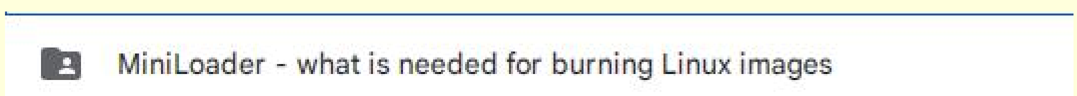
[orangepi@orangepi ~]$ **sudo dd if=/boot/rkspi_loader.img of=/dev/mtdblock0**

If you need to start the OpenWRT image, you need to download the latest version of the u-boot image from the official website, and then burn it to 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 select to enter the directory below



c. Then download rkspi_loader.img

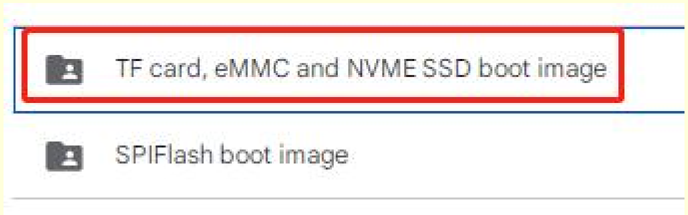
d. Then upload rkspi_loader.img to the Ubuntu, Debian, or OPi OS Arch system of the development board. Please refer to the instructions in the Methods section for uploading files to the Linux system of the development board.

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, Ubuntu, or OpenWRT image downloaded from the official website) to the TF card. For the method of uploading Linux image files to the development board, please refer to the instructions in **the Methods section of uploading files to the Linux system of the development board**.

Note that if you are downloading 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 "TF Card, eMMC, and NVME SSD Boot Image" folder.



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

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepi5pro_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@Orangepi5pro:~/Desktop$ sudo dd bs=1M if=/dev/zero of=/dev/nvme0n1 count=2000 status=progress
orangepi@Orangepi5pro:~/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. 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=Orangepi5pro_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. 7. 3. The method of burning using BalenaEtcher software

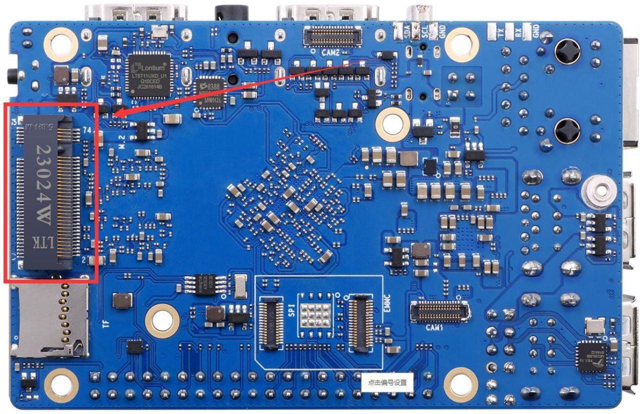
Please do not use this method for OPi OS Arch and OpenWRT systems.

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.

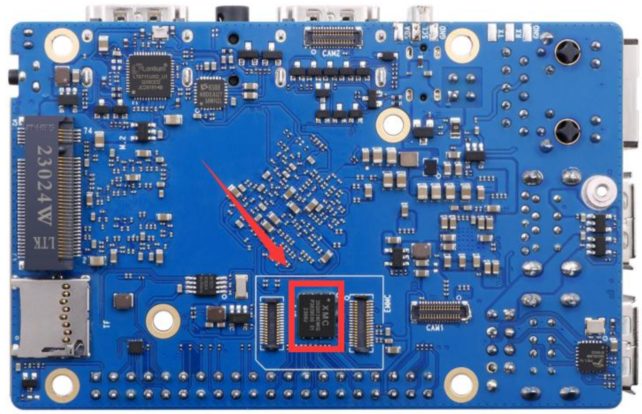
a. The M.2 2230 SSD is as follows

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) Please ensure that the development board has already been labeled with SPI Flash. The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting the burning process



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.

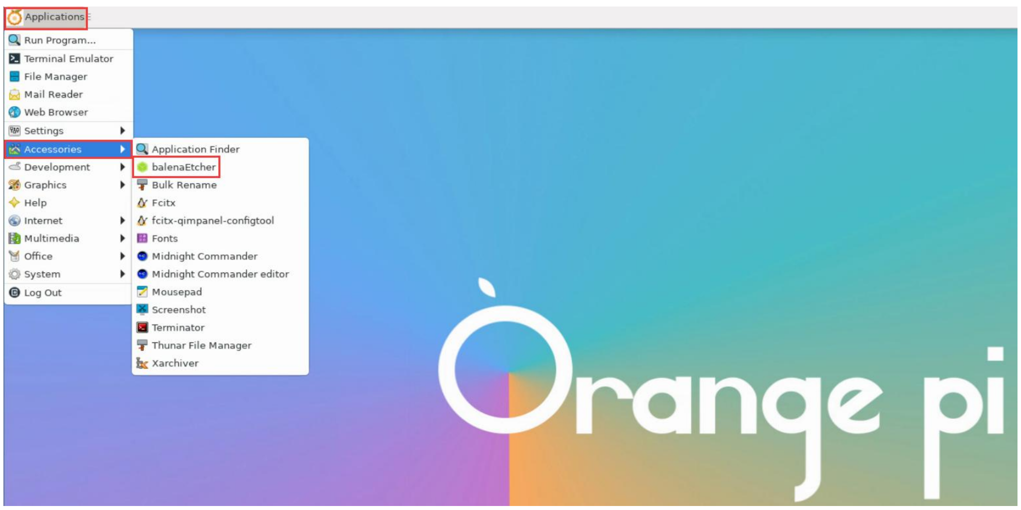
```
orangepi@orangepi:~/Desktop$ sudo fdisk -l | grep "nvme0n1"
```

```
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, please refer to the instructions in the section on downloading and installing the arm64 version of balenaEtcher.

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

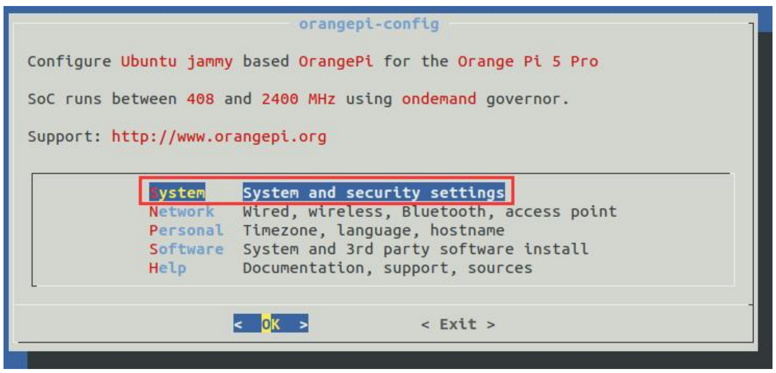
8) The method of burning u-boot into SPI Flash on the development board using balenaEtcher is as follows:

a. In Linux systems, the default SPI Flash feature is turned off and needs to be manually turned on to use. The detailed steps are as follows:

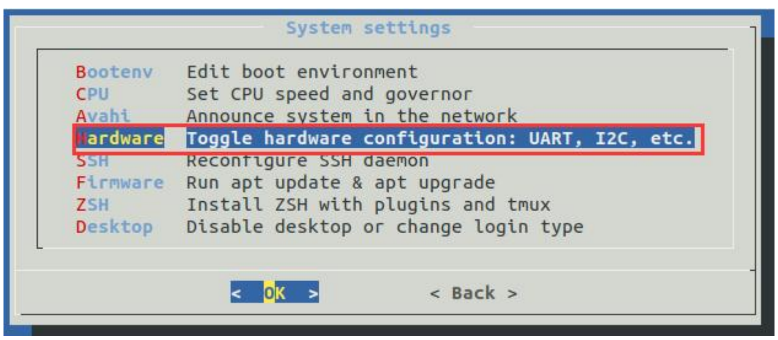
a) First, run orange pi configuration, **ordinary users remember to add sudo permissions**

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

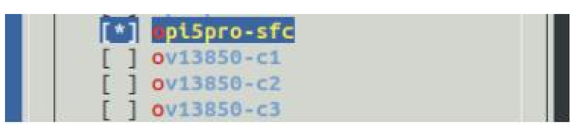
b) Then select **System**



c) Then select **Hardware**



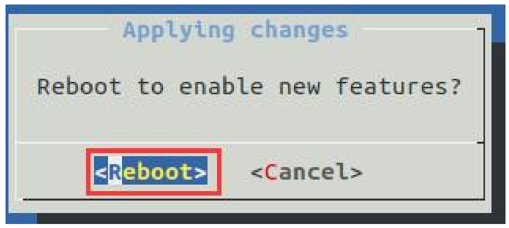
d) Then use the arrow keys on the keyboard to navigate to **opi5pro-sfc**, and use a space to select it



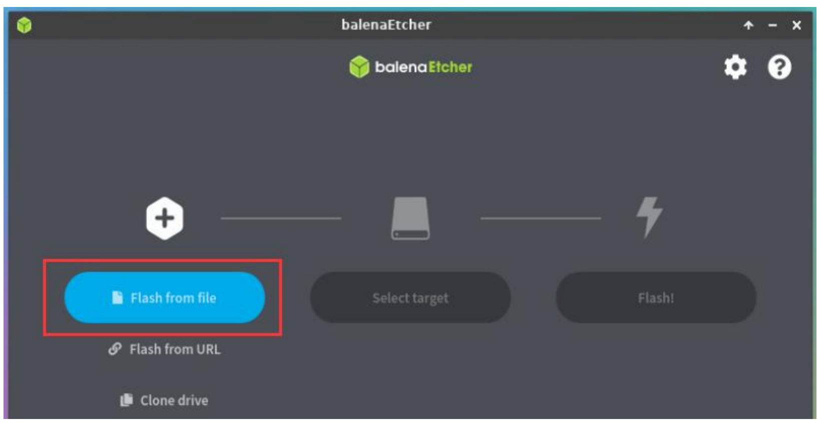
e) Then select **<Save>** to save

f) Then select **<Back>**

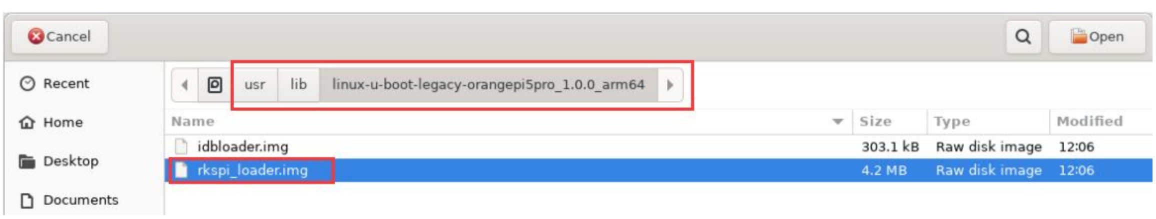
g) Then select**<Reboot>**to restart the system for the configuration to take effect



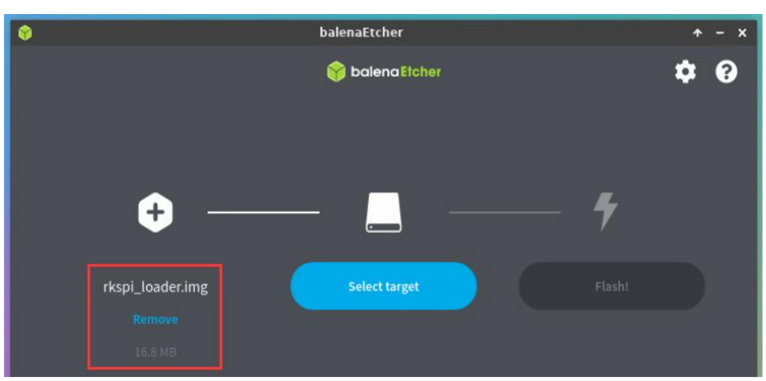
b. Then open the balenaEtcher software and click on **Flash from file**



c. Then enter the **/usr/lib/linux-u-boot-legacy-orangepi5pro_1.0.0_arm64/** directory, select **rkspi_loader.img**, and click **Open** to open it

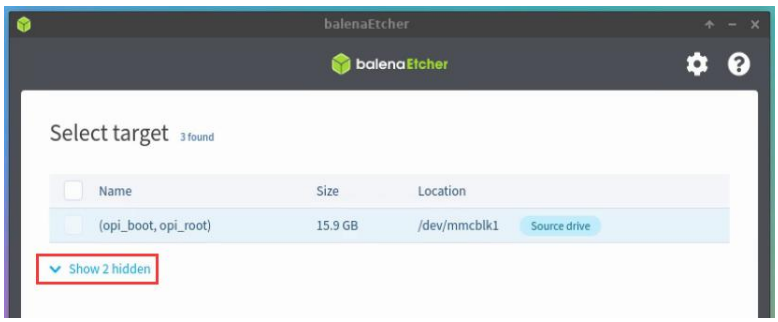


d. The interface after opening **rkspi_loader.img** is as follows:

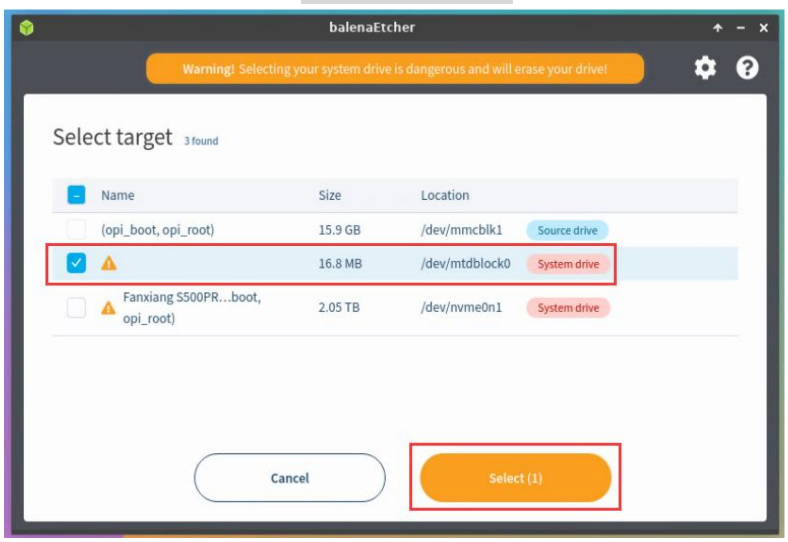


e. Then click **Select target**

f. Then click on **Show 2 hidden** to open more storage device options

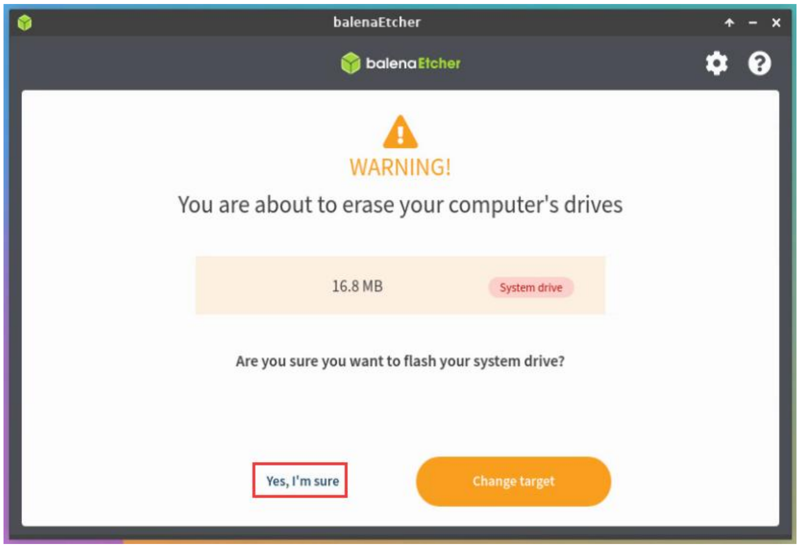


g. Then select the device name **/dev/mtdblock0** for SPI Flash and click **Select**

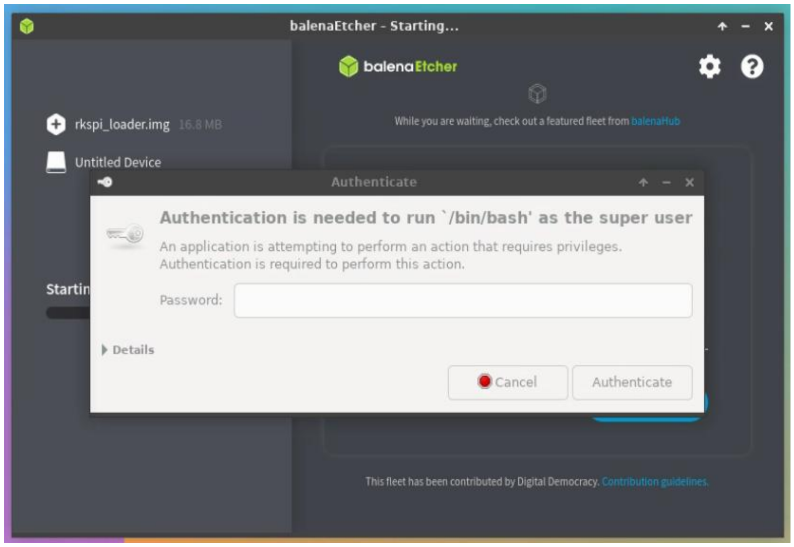


h. Then click **Flash**

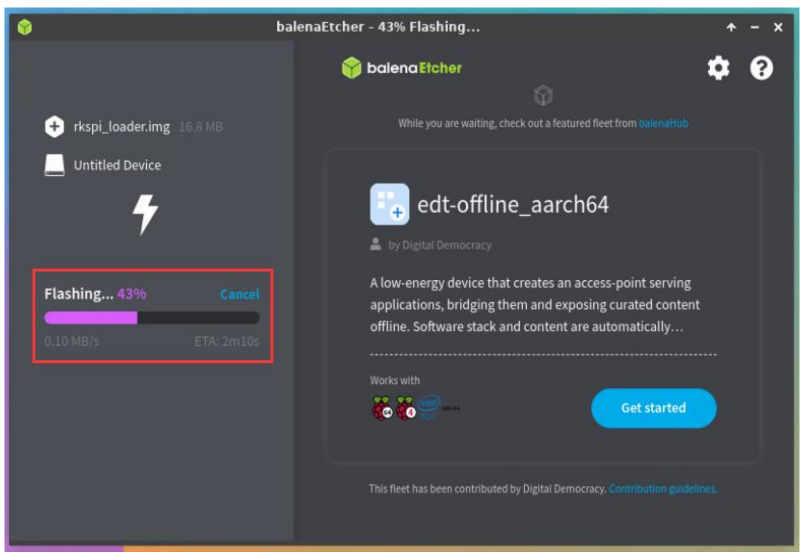
i. Then click **Yes, I'm sure**



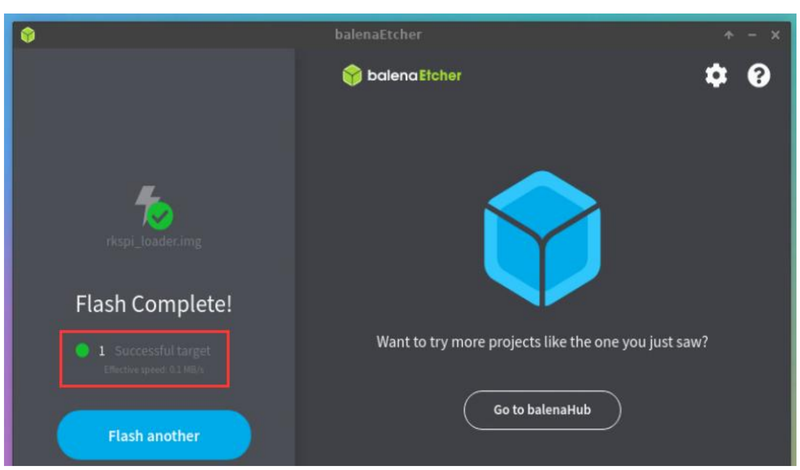
j. Then enter the password **orangepi** for the Linux system on the development board, and you will start burning the u-boot image to SPI Flash



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

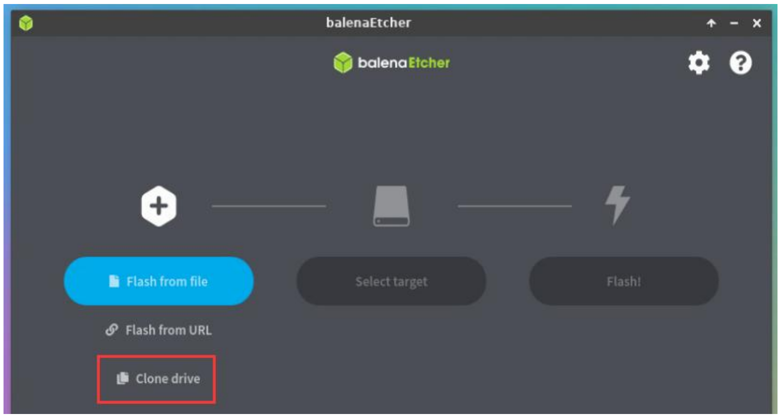


l. The display after burning is completed is as follows:

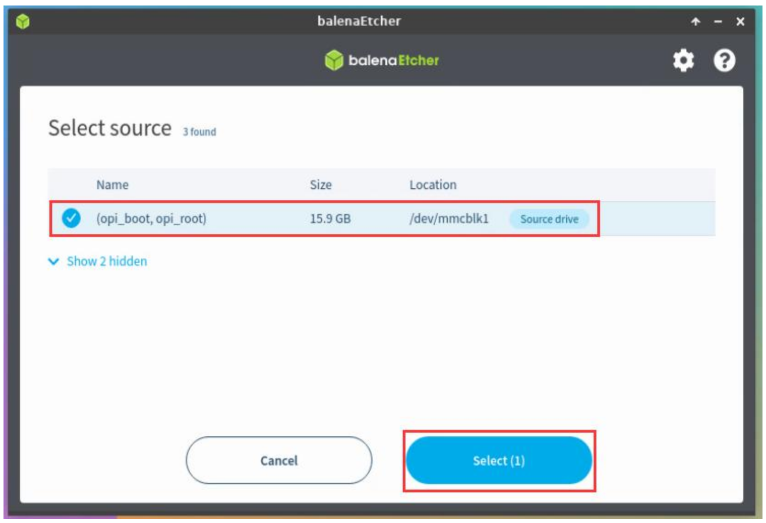


9) The method of burning Linux system from TF card to NVMe SSD (this method is equivalent to cloning the system from TF card to NVMe SSD)

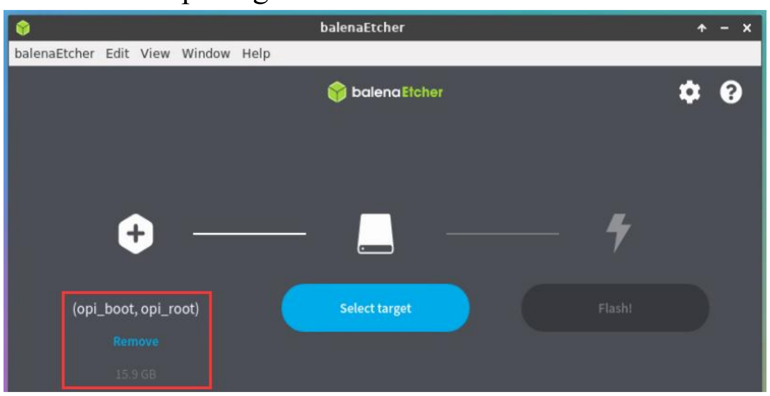
a. First click **Clone drive**



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

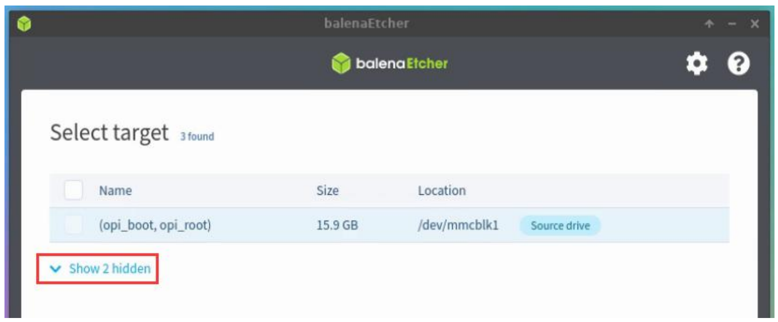


c. The interface after opening the TF card is as follows:

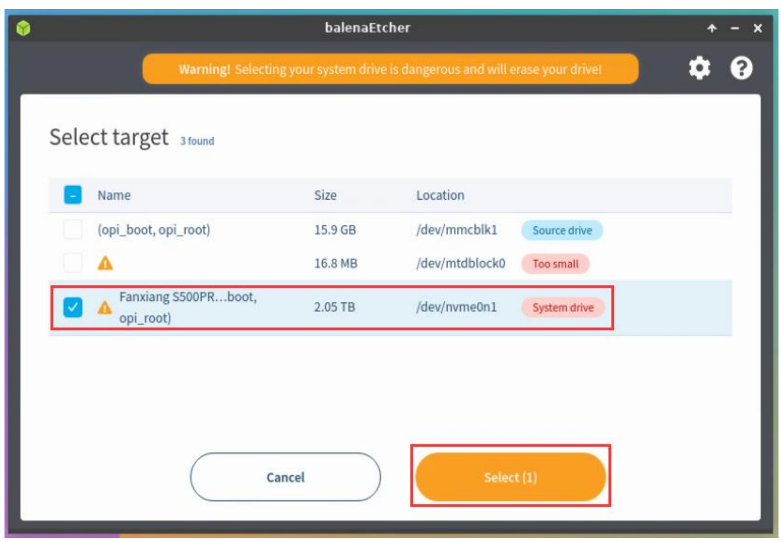


d. Then click **Select target**

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

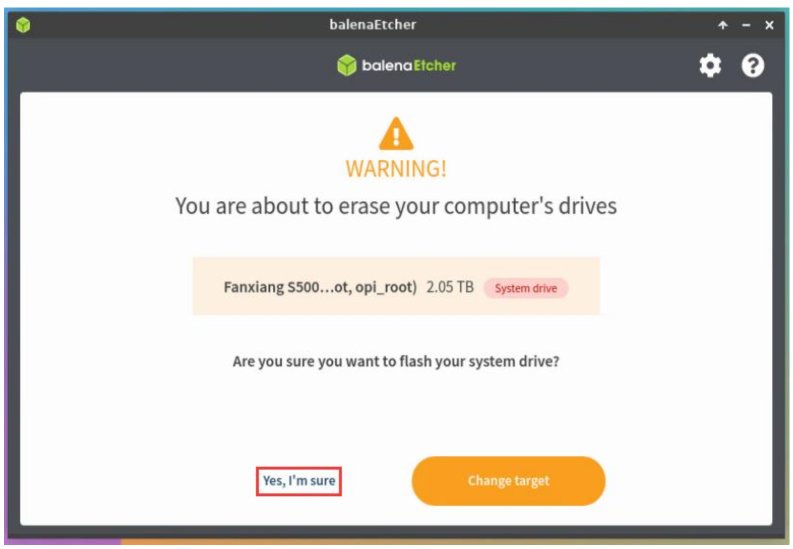


f. Then select the device name **/dev/nvme0n1** for the NVMe SSD and click **Select**

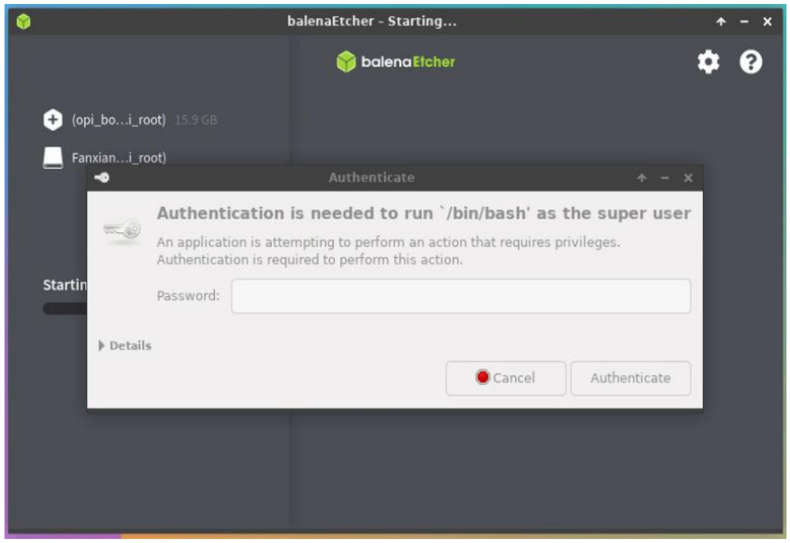


g. Then click **Flash**

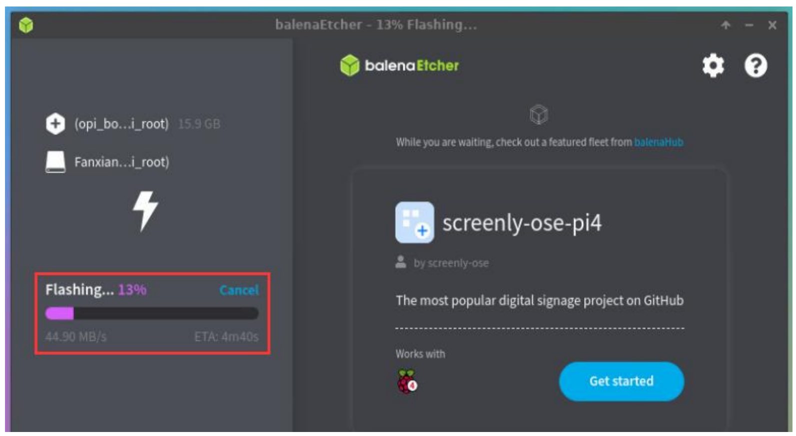
h. Then click **Yes, I'm sure**

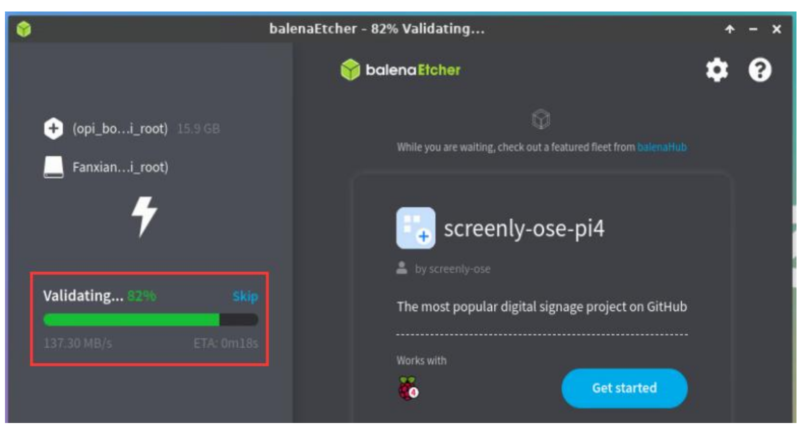


i. Then enter the password orange pi for the Linux system on the development board, and it will start burning the Linux image to the SSD

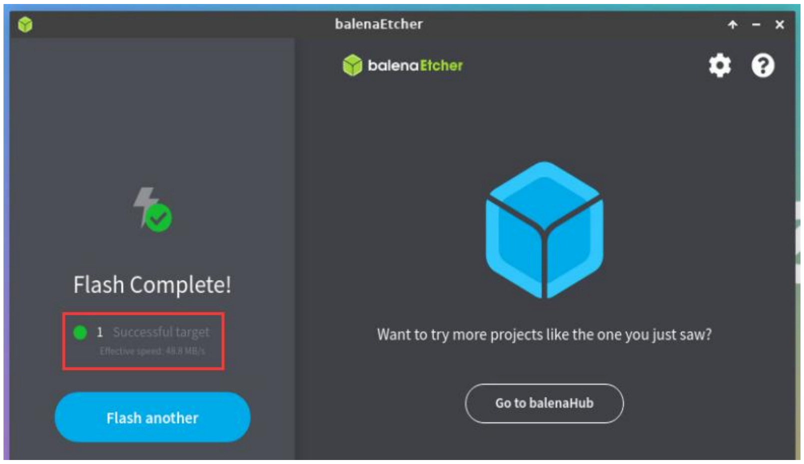


j. The display of the burning process is as follows:





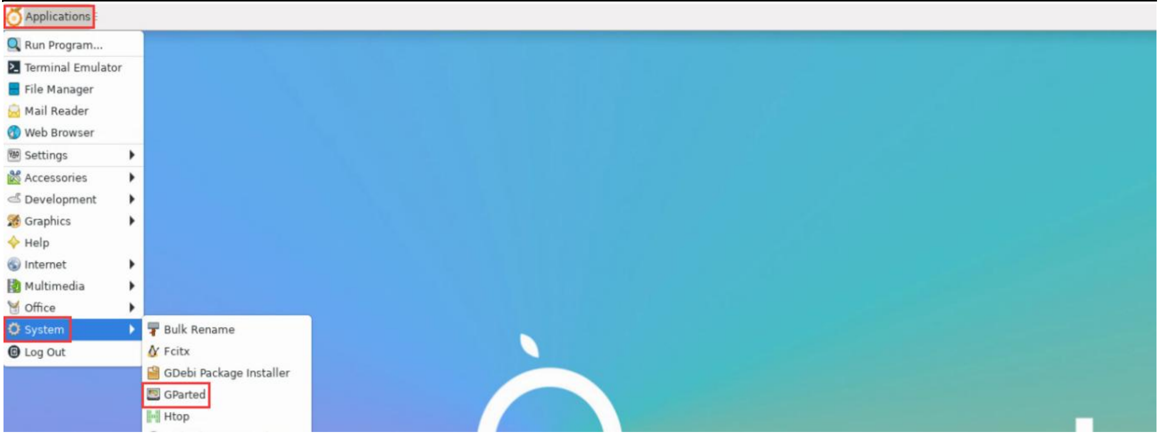
k. The display after burning is completed is as follows:



l. Then it is necessary to expand the capacity of the rootfs partition in the NVMe SSD, as follows:

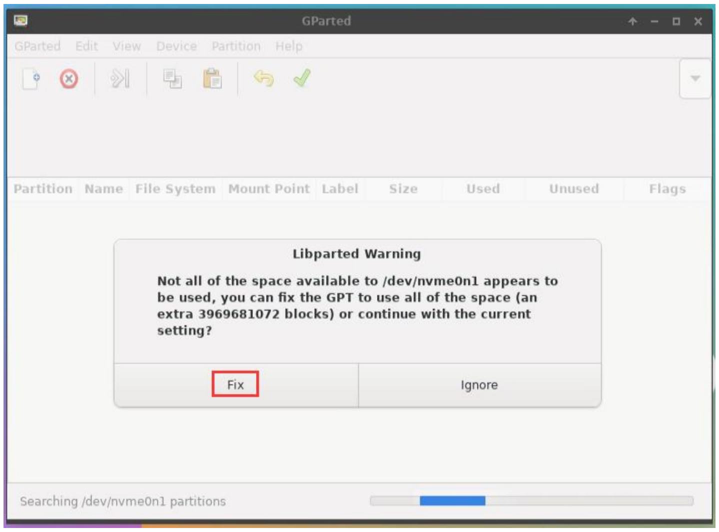
a) First, open **GParted**. If the system does not have Gparted pre installed, please use the apt command to install it

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

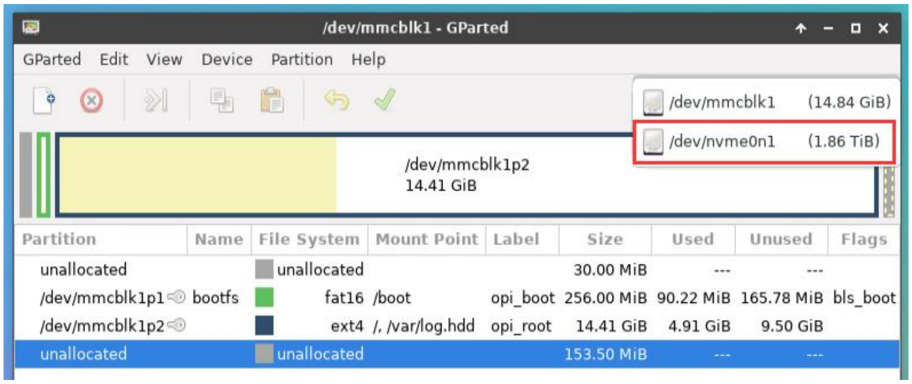


b) Then enter the password for the Linux system, orangepi, and click on **Authenticate**

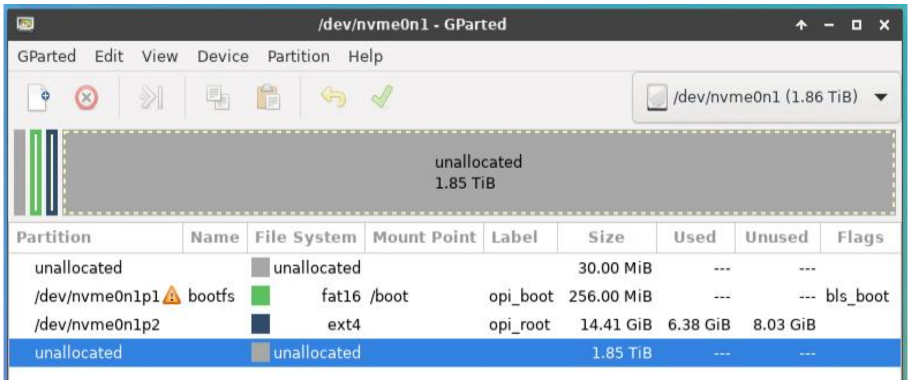
c) Then click **Fix**



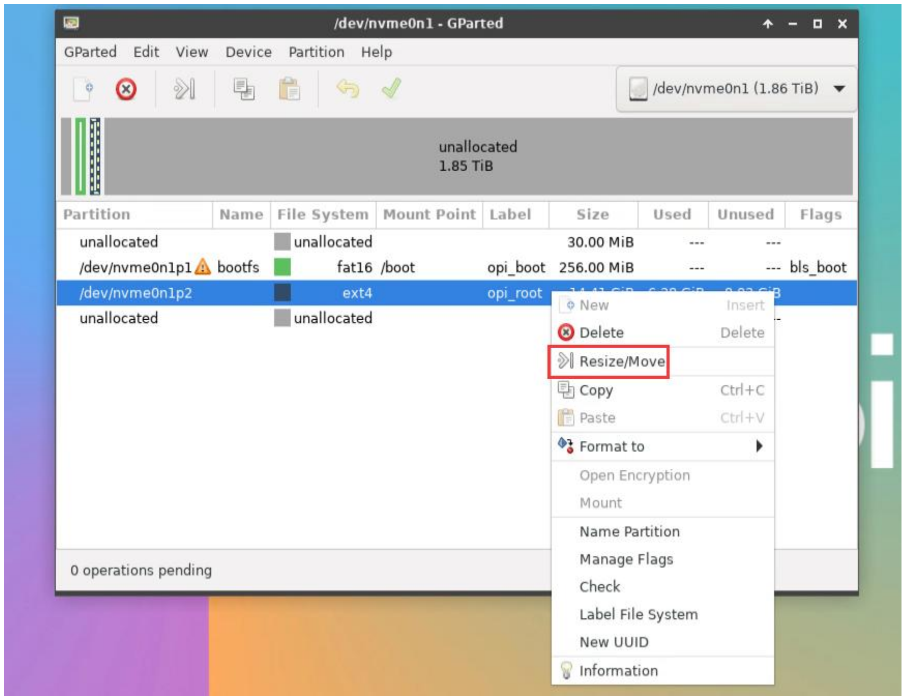
d) Then select NVMe SSD



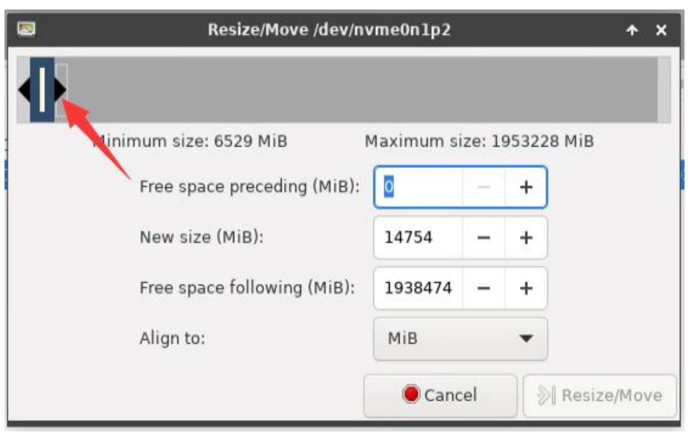
e) The display interface after selecting NVMe SSD is as follows:



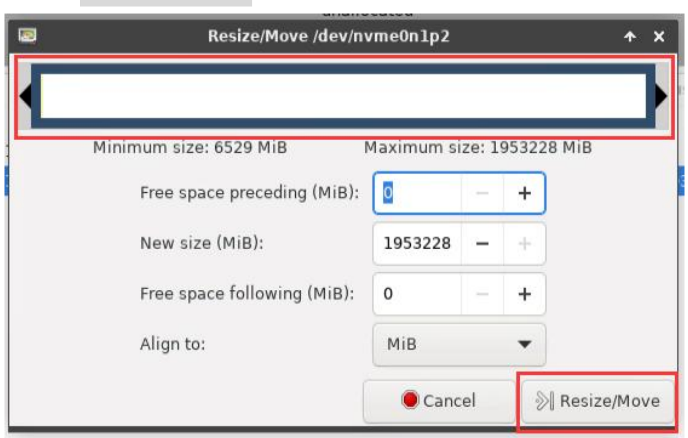
f) Then select the **/dev/nvme0n1p2** partition, right-click, and select **Resize/Move**



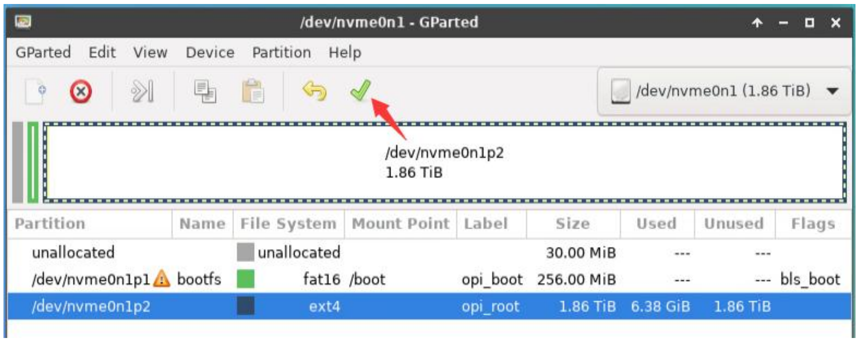
g) Then drag the capacity to its maximum position as shown in the following figure



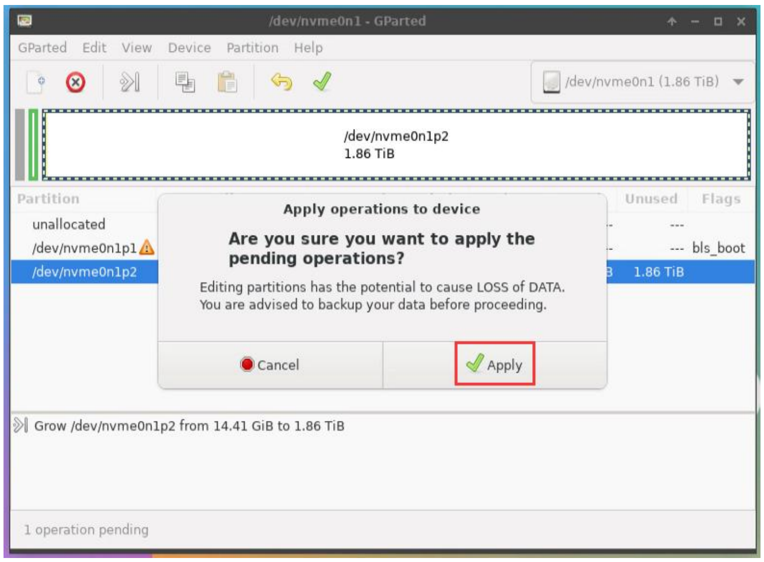
h) Then click **Resize/Move**



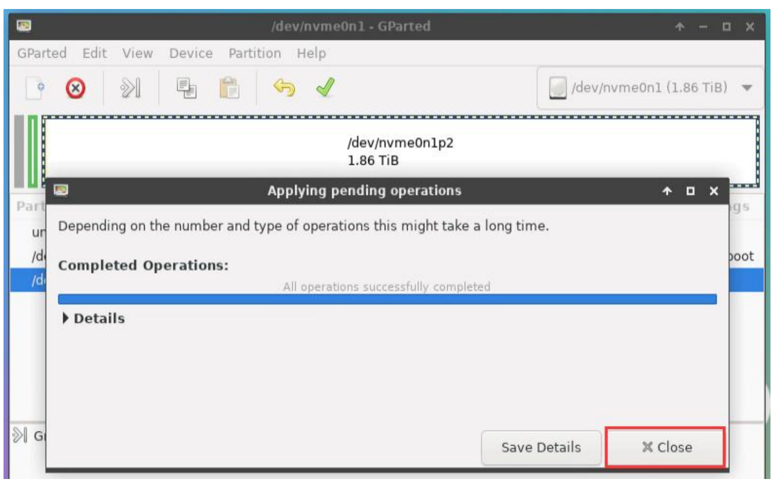
i) Then click on the green √ in the position shown below



j) Then click **Apply**



k) Then click **Close** to close it

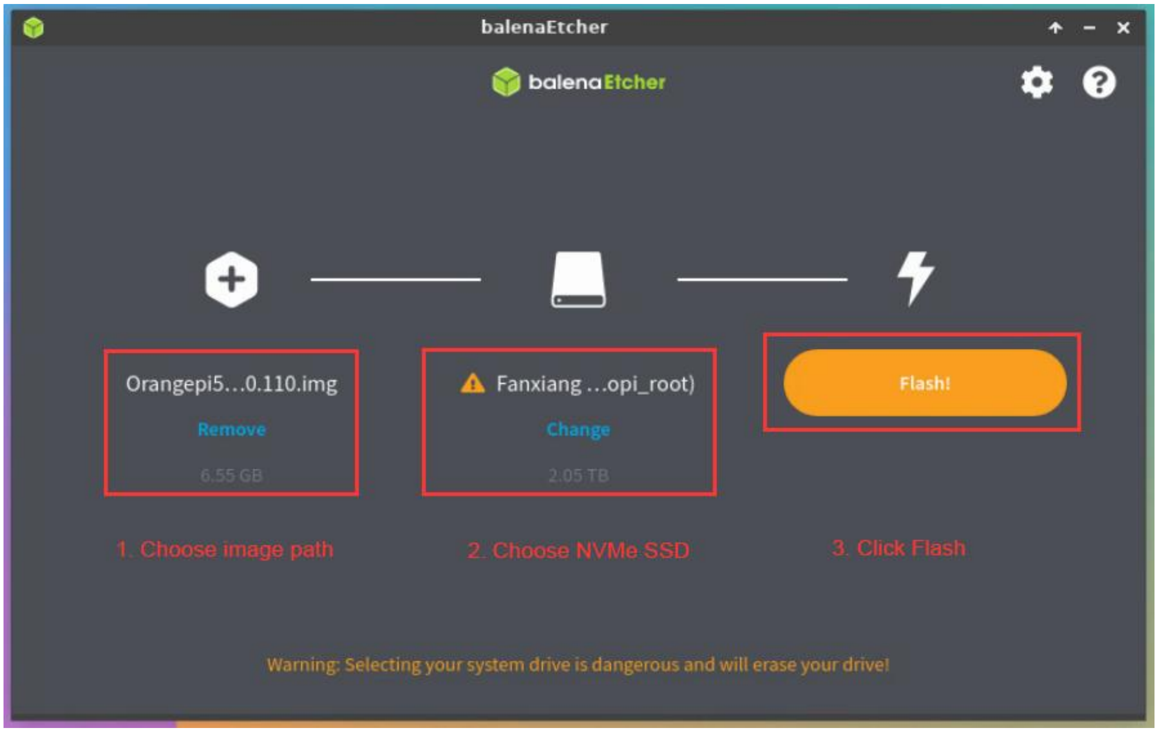


m. At this point, you can use the **sudo poweroff** command to shut down. Then please unplug the TF card and briefly press the power button to turn on the Linux system in SPIFlash+NVMe SSD.

10) Step **9)** is to clone the system from the TF card to the NMVe SSD, and we can also directly burn the Linux image file to the NVMe SSD. Here is an overview of the following steps:

a. Upload Linux image files to the Linux system of the development board

b. Then use balenaEtcher to burn it



c. **After burning the image using this method, there is no need to manually expand it. It will automatically expand on the first startup.**

2. 8. The method of burning Linux images to SPIFlash+SATA SSDs

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

Before starting to burn the image, it is necessary to ensure that the development board has already attached the SPI Flash chip, as the development board does not have an SPI Flash chip attached at the factory, so it needs to be purchased and soldered on by oneself. We recommend using the XM25QU128CWIQT08Q SPI

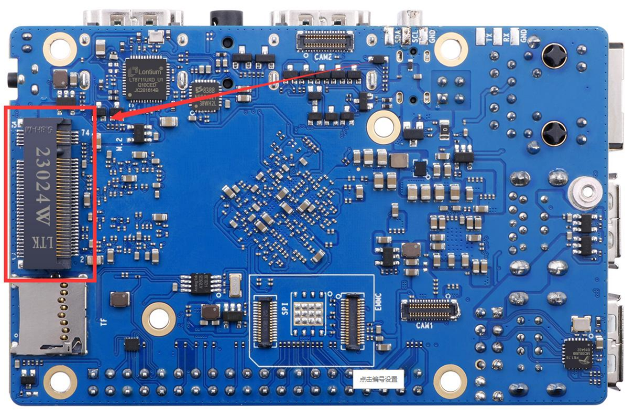
Flash chip model

Due to SPI Flash and eMMC reusing the same pins, eMMC cannot be used with SPI Flash attached

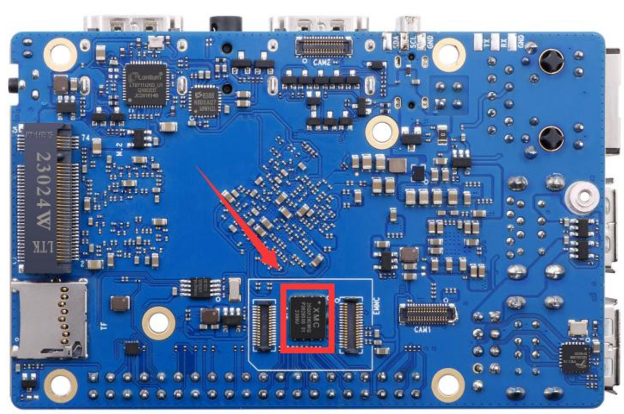
2.8.1. The method of burning using the dd command

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

2) Then insert the SSD into the M.2 interface of the development board and secure it in place



3) Please ensure that the development board has already been labeled with SPI Flash. The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting the burning process



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 **the method of burning the Linux image to the TF card based on the on the Ubuntu PC**.

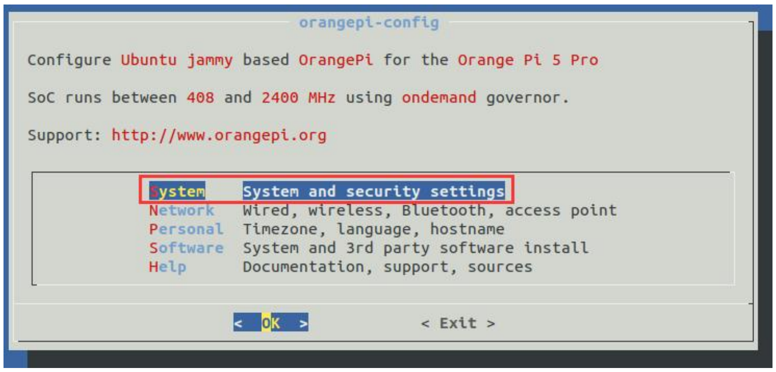
5) After booting the Linux system with a TF card, we first burn the u-boot image dedicated to Sata SSD to SPI Flash

a. In Linux systems, the default SPI Flash feature is turned off and needs to be manually turned on to use. The detailed steps are as follows:

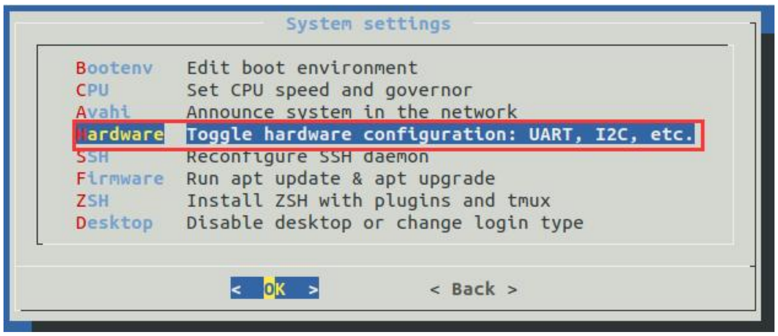
a) First run orangepi-config, **normal users remember to add sudo permission**

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

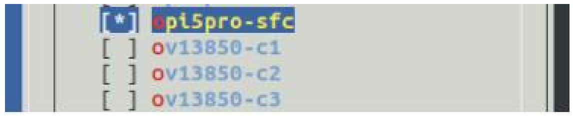
b) Then select **System**



c) Then select **Hardware**



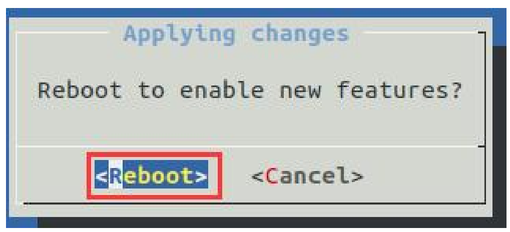
d) Then use the arrow keys on the keyboard to navigate to **opi5pro-sfc**, and use a space to select it



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

f) Then select **<Back>**

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



b. The dedicated u-boot image storage path for sata ssd startup is:

/usr/share/orangepi5pro/rkspi_loader_sata.img

c. 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/orangepi5pro/**
orangepi@orangepi:~$ **sudo dd if=rkspi_loader_sata.img of=/dev/mtdblock0**
orangepi@orangepi:~$ **sudo sync**

OPi OS Arch system does not have SPI Flash enabled by default and needs to be manually opened. Please add the following configuration in /boot/extLinux/extlinuxconf and restart the system to recognize SPI Flash

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf

FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-opi5pro-sfc.dtbo

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

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

/boot/rkspi_loader_sata.img

b. 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:

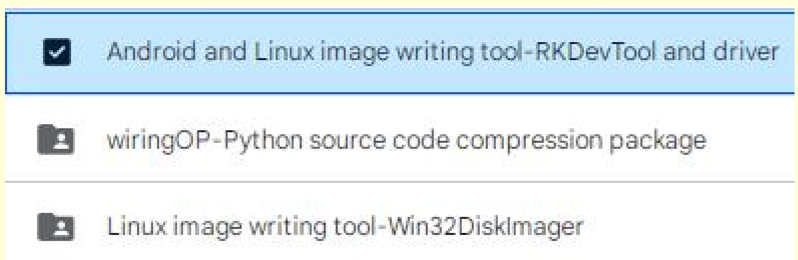
[orangepi@orangepi boot]$ **cd /boot/**

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

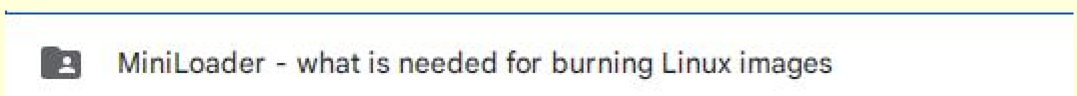
[orangepi@orangepi boot]$ **sudo sync**

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:

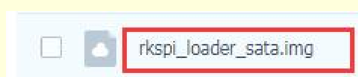
a. First enter the data download page of the development board, then select the official tool on the data download page, and then enter the folder below



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.

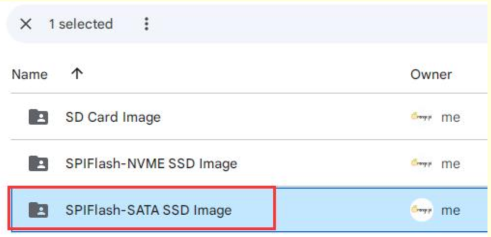
Finally, execute the following command to burn the u-boot image into SPI Flash (note that this command is executed in Ubuntu, Debian, or OPi OS Arch):

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, OPi OS Arch image, or OpenWRT image) to the TF card. For the method of uploading Linux image files to the development board, please refer to the instructions in the Methods section of **uploading files to the Linux system of the development board**.

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
Orangepi5pro_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) 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=Orangepi5pro_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:

Note that if it is a burned OpenWRT image, the following steps can be ignored, as the SATA image of OpenWRT has SATA SSD configuration enabled by default.

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-sata2
```

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-sata2.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.

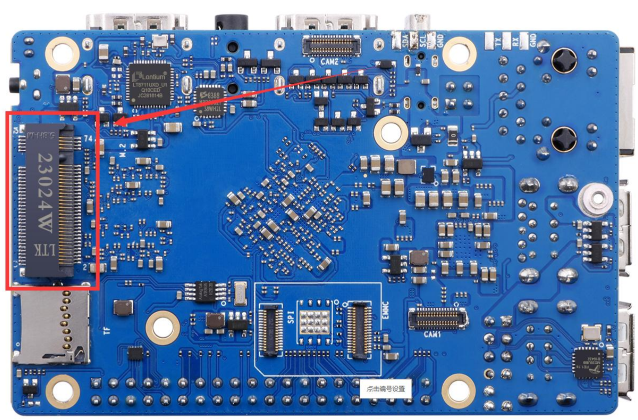
There is no fix_mmc_ssd.sh script in OPi OS Arch system.

2.8.2. The method of burning using BalenaEtcher software

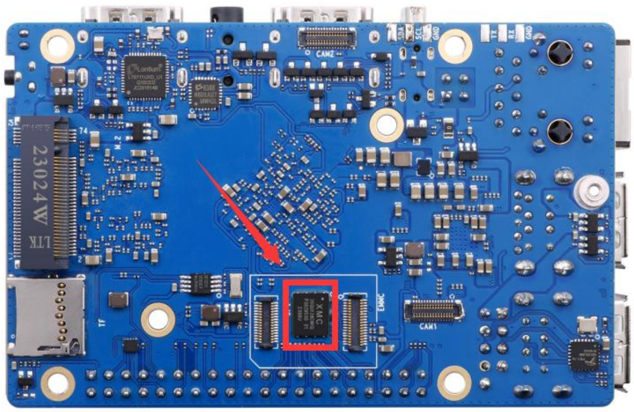
Please do not use this method for OPi OS Arch and OpenWRT systems.

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

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



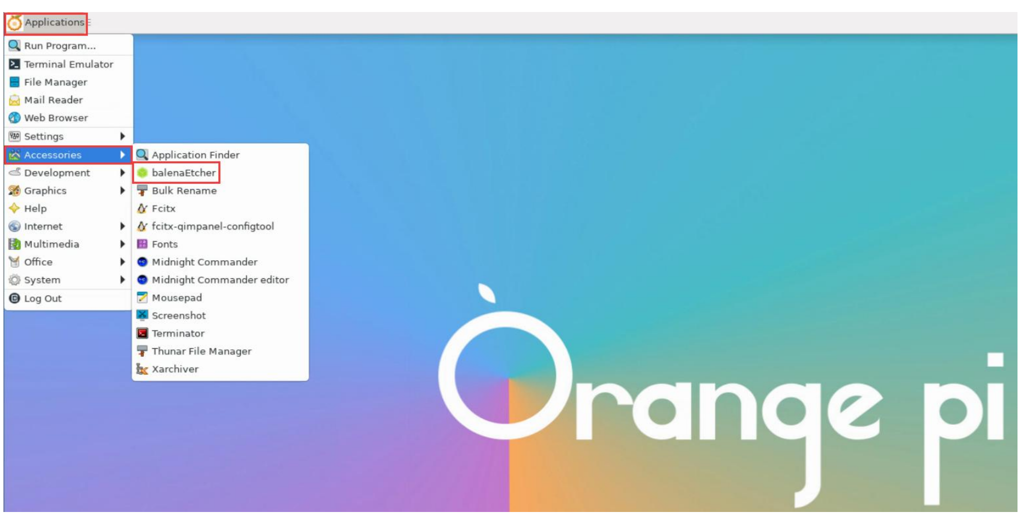
3) Please ensure that the development board has already been labeled with SPI Flash. The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting the burning process



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 **the method of burning the Linux image to the TF card based on the Ubuntu PC**.

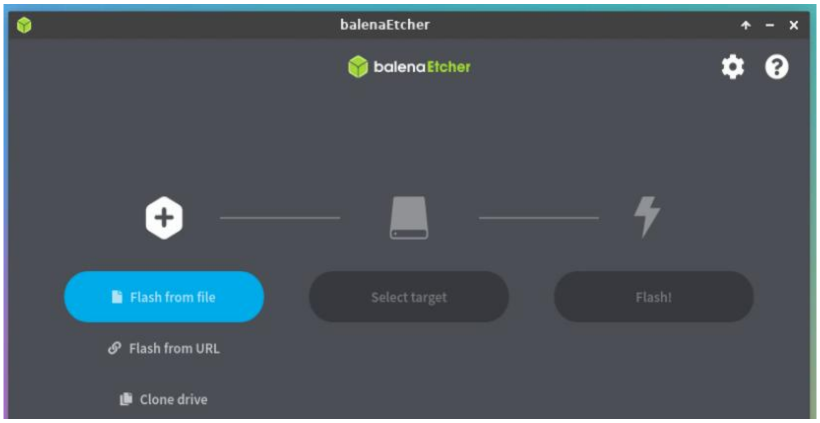
5) Then please refer to the instructions in the section of **the method of using SATA SSD** to open the sata ssd configuration to ensure that the system can recognize the ssd normally.

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:



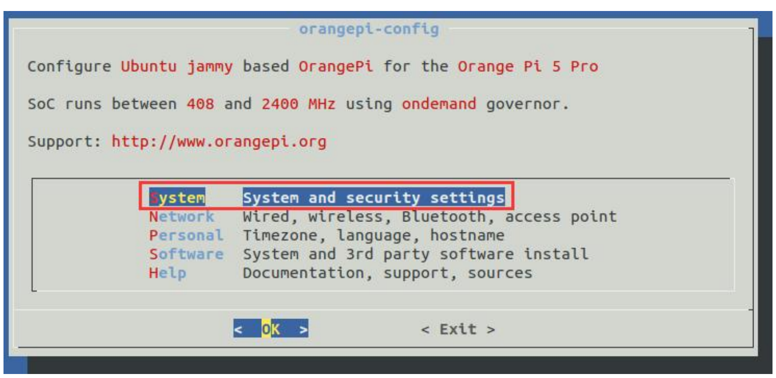
8) The method of burning u-boot into SPI Flash on the development board using balenaEtcher is as follows:

a. In Linux systems, the default SPI Flash feature is turned off and needs to be manually turned on to use. The detailed steps are as follows:

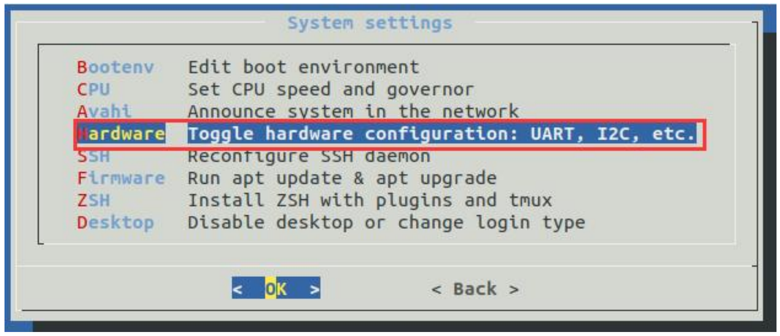
a) First, run orange pi configuration, **ordinary users remember to add sudo permissions**

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

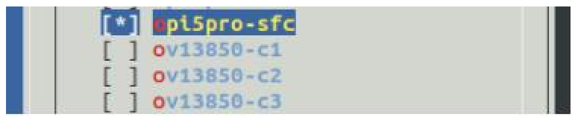
b) Then select **System**



c) Then select **Hardware**



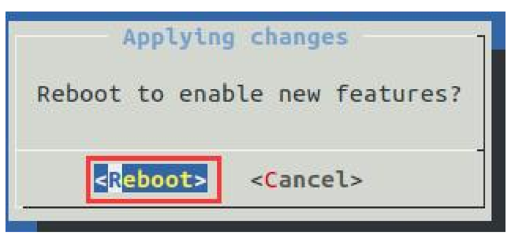
d) Then use the arrow keys on the keyboard to navigate to **opi5pro-sfc**, and use a space to select it



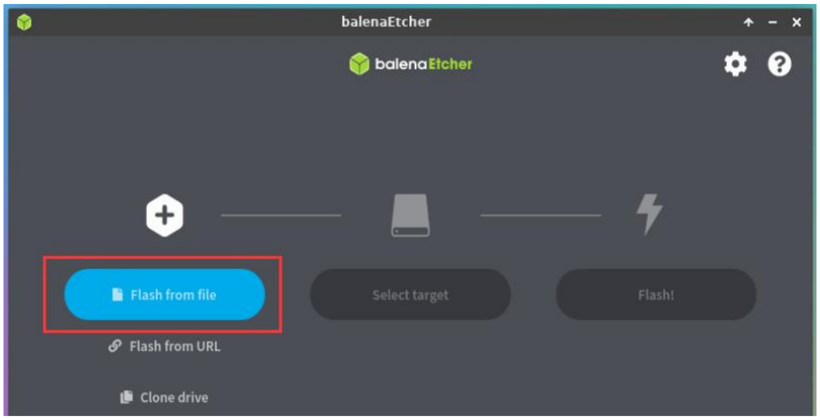
e) Then select **<Save>** to save

f) Then select **<Back>**

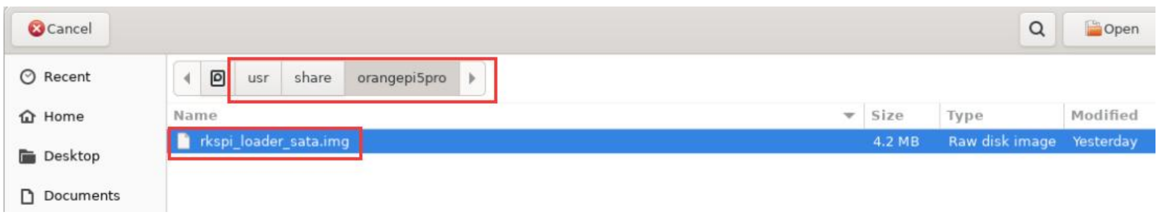
g) Then select **<Reboot>** to restart the system for the configuration to take effect



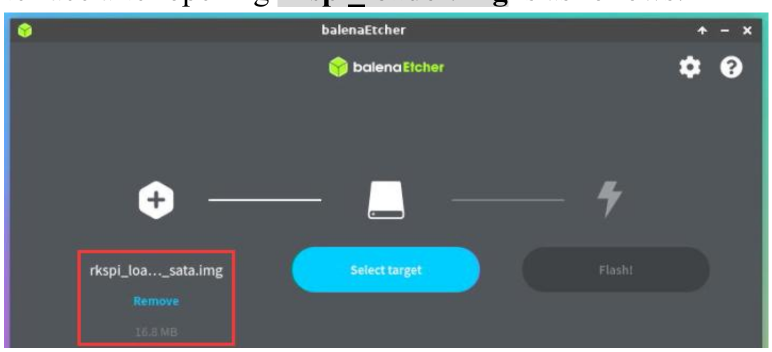
b. Then select **Flash from file**



c. Then enter the **/usr/share/orangepi5pro/** directory, select **rkspi_loader_sata.img**, and click **Open** to open it

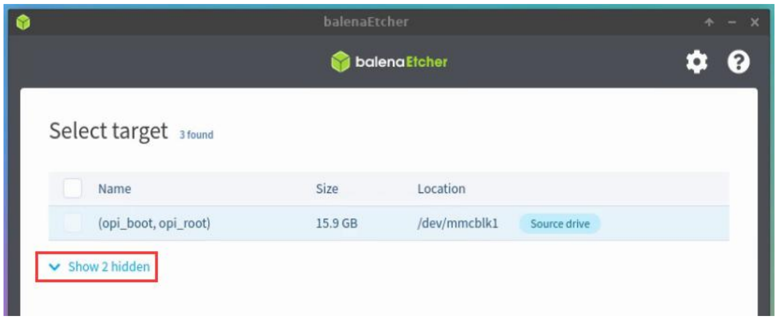


d. The interface after opening **rkspi_loader.img** is as follows:

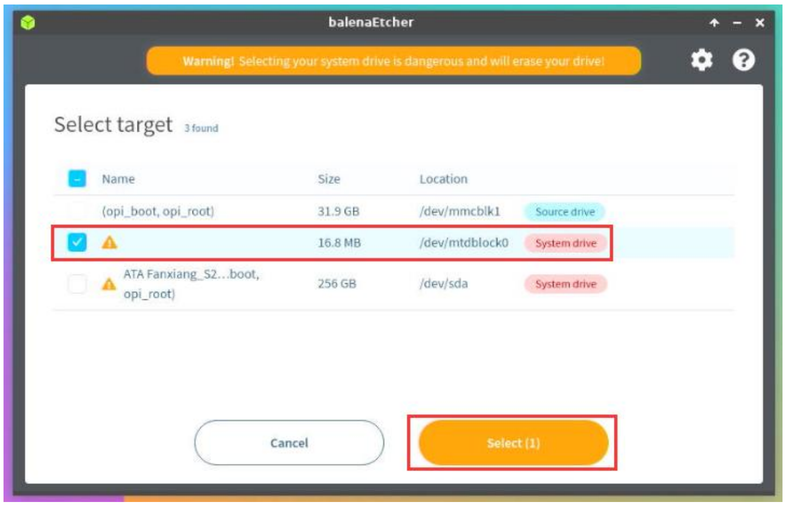


e. Then select **Select target**

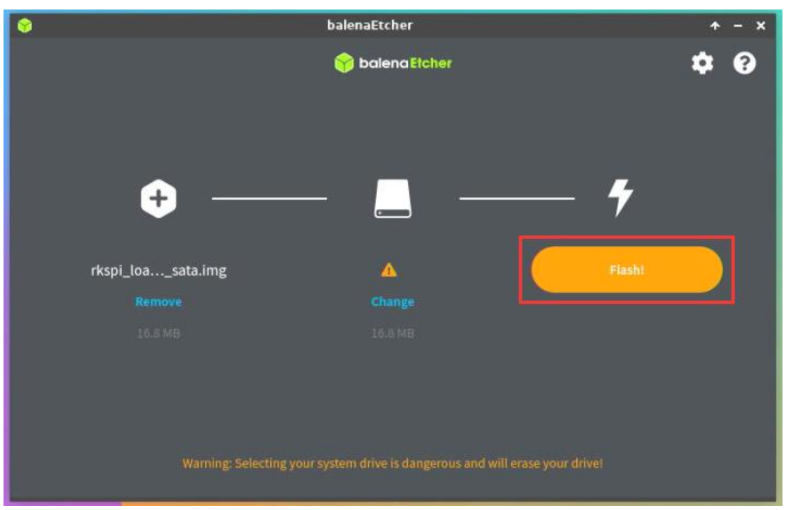
f. Then click on **Show 2 hidden** to open more storage device options



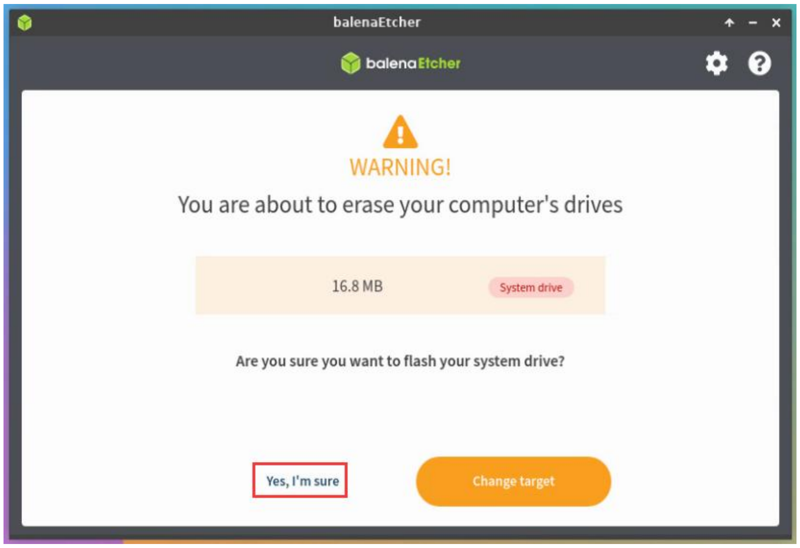
g. Then select the device name **/dev/mtdblock0** for SPI Flash and click **Select**



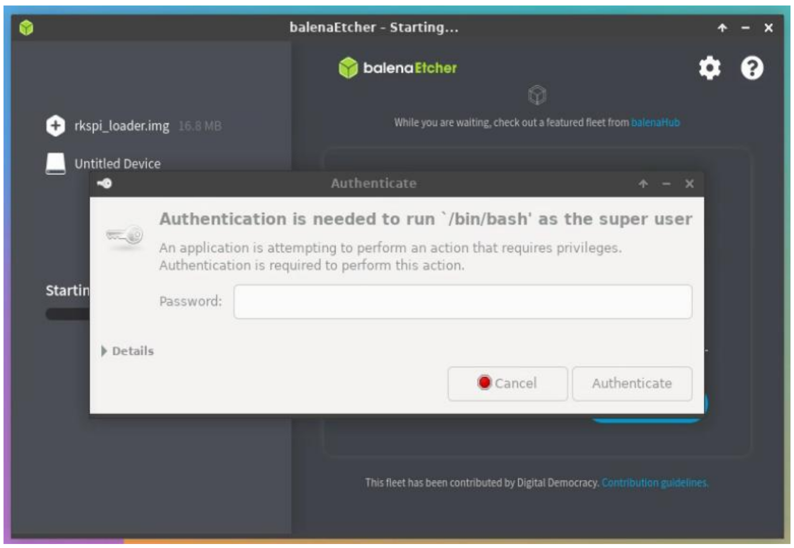
h. Then click **Flash**



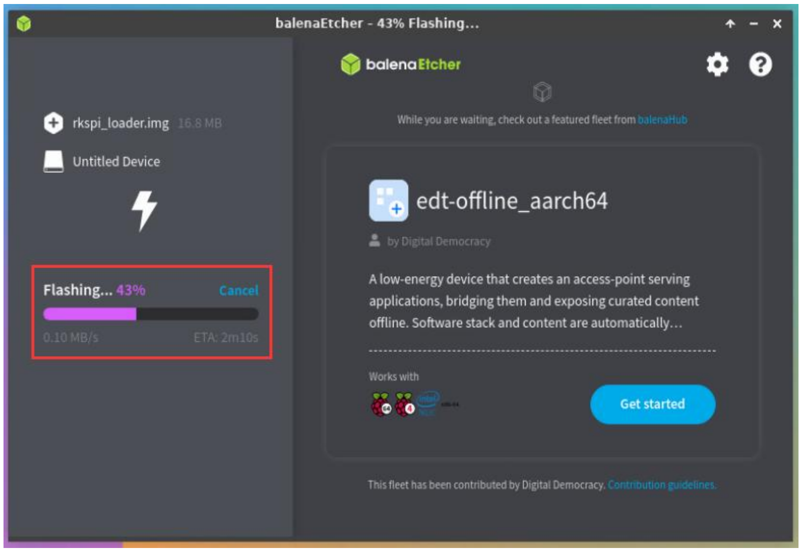
i. Then click **Yes, I'm sure**



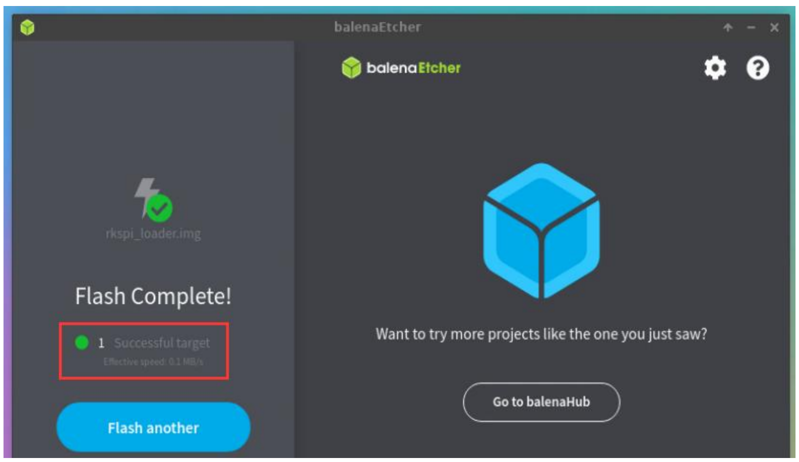
j. Then enter the password orange pi for the Linux system on the development board, and you will start burning the u-boot image to SPI Flash



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

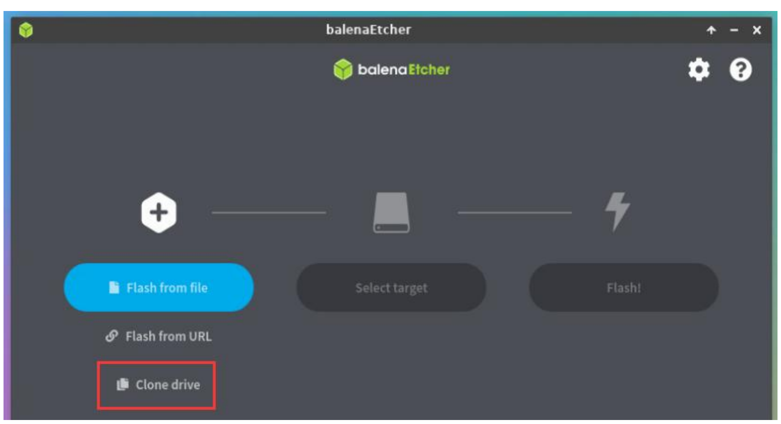


l. The display after burning is completed is as follows:

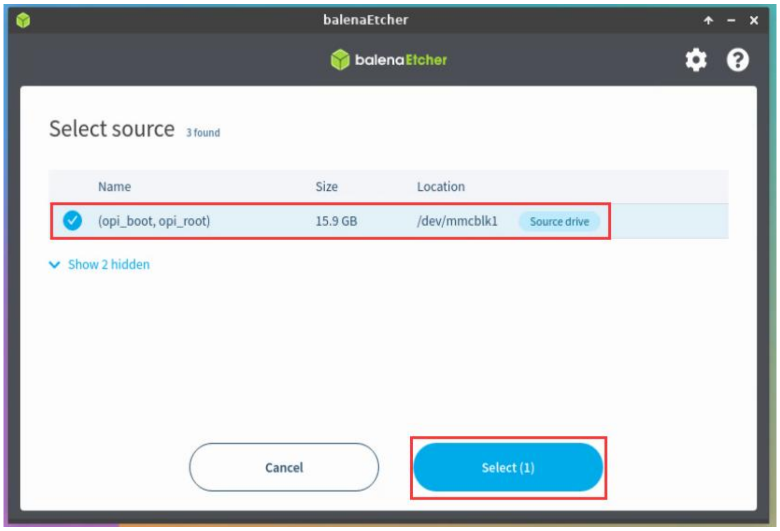


9) The method of burning Linux system from TF card to SSD (this method is equivalent to cloning the system from TF card to SSD)

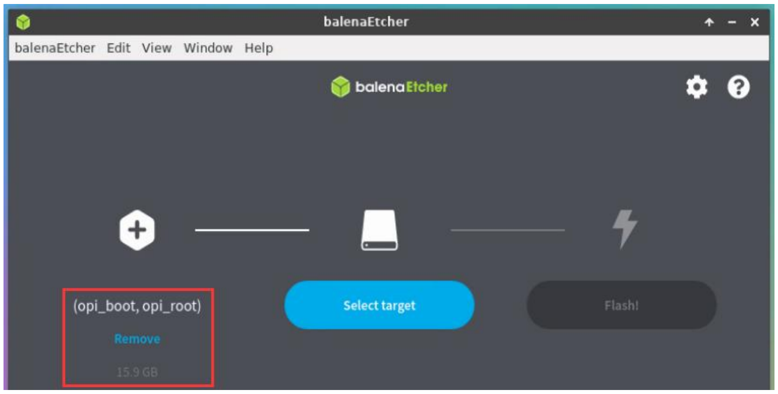
a. First click **Clone drive**



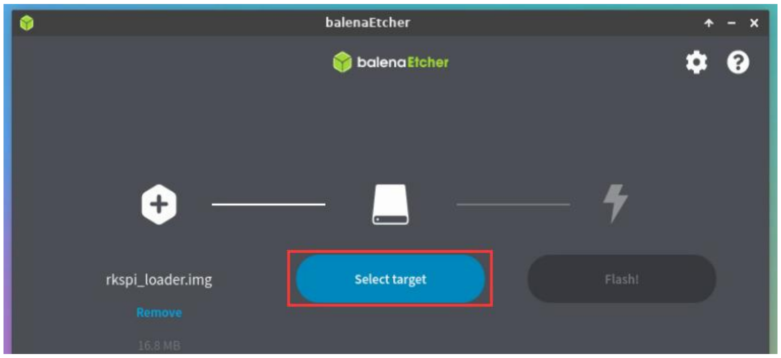
b. Then select the device name of the TF card **/dev/mmcblk1**



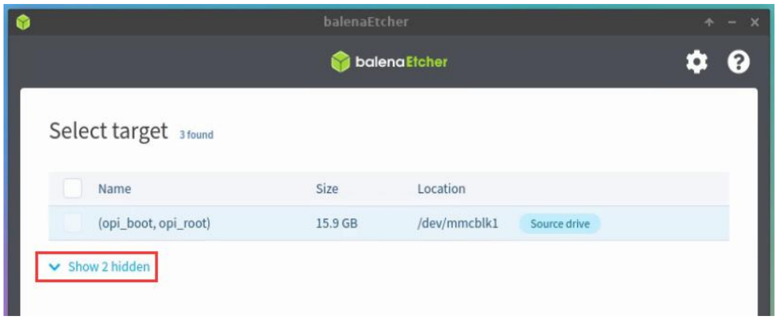
c. The interface after opening the TF card is as follows:



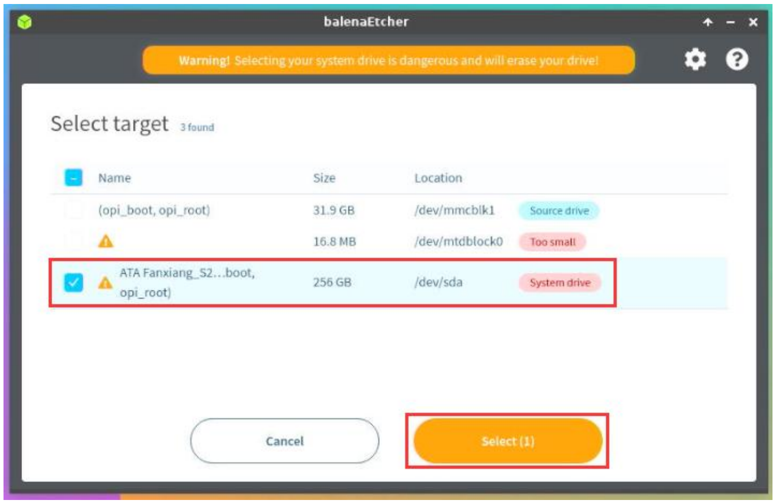
d. Then click **Select target**



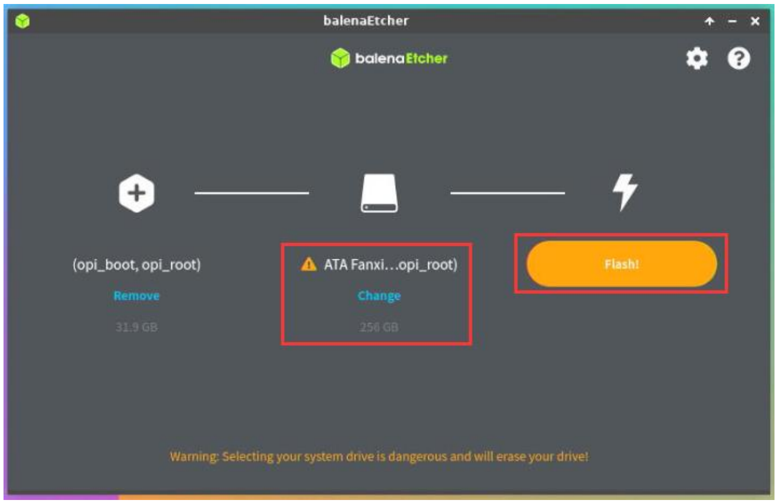
e. Then click on **Show 2 hidden** to open more storage device options



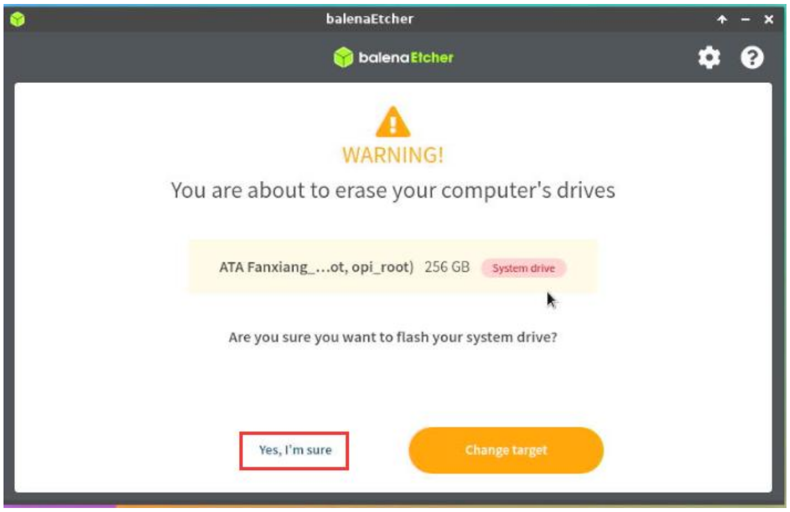
f. Then select the device name **/dev/sda** for the SSD, and click **Select**



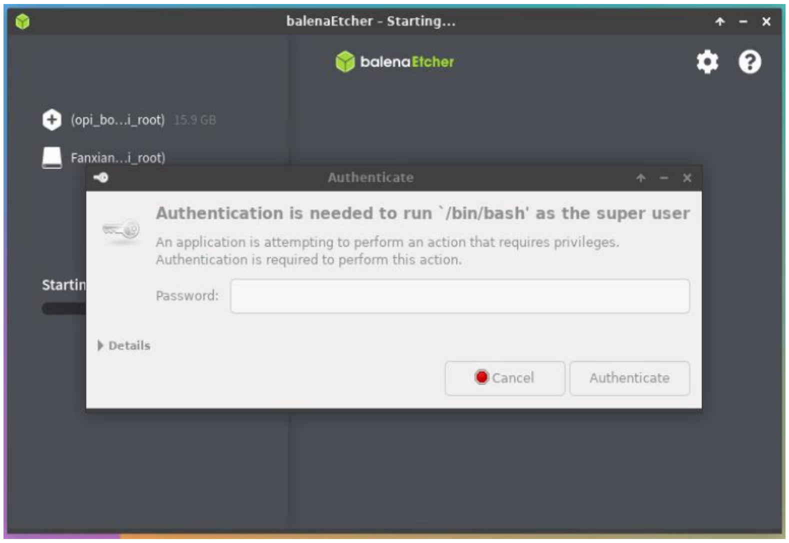
g. Then click **Flash**



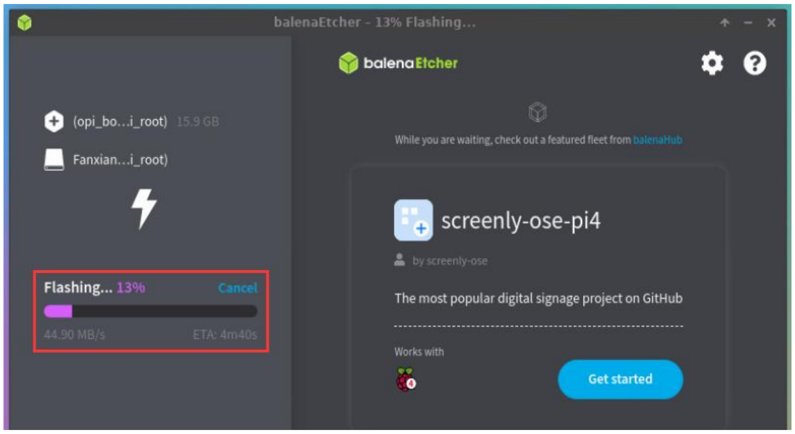
h. Then click **Yes, I'm sure**

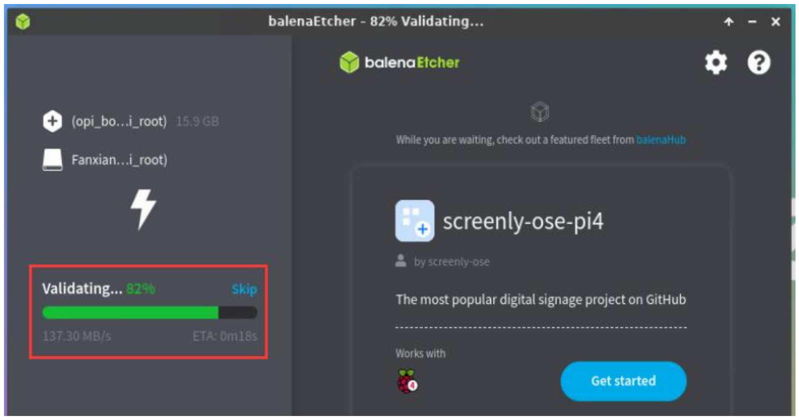


i. Then enter the password **orangepi** for the Linux system on the development board, and it will start burning the Linux image to the SSD

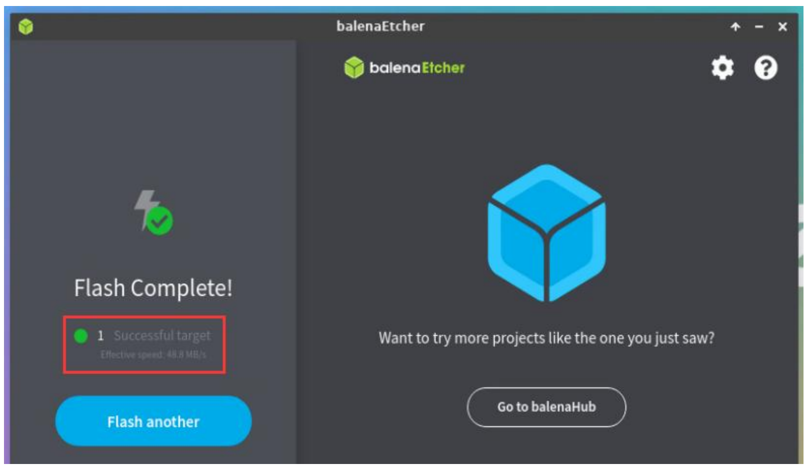


j. The display of the burning process is as follows:





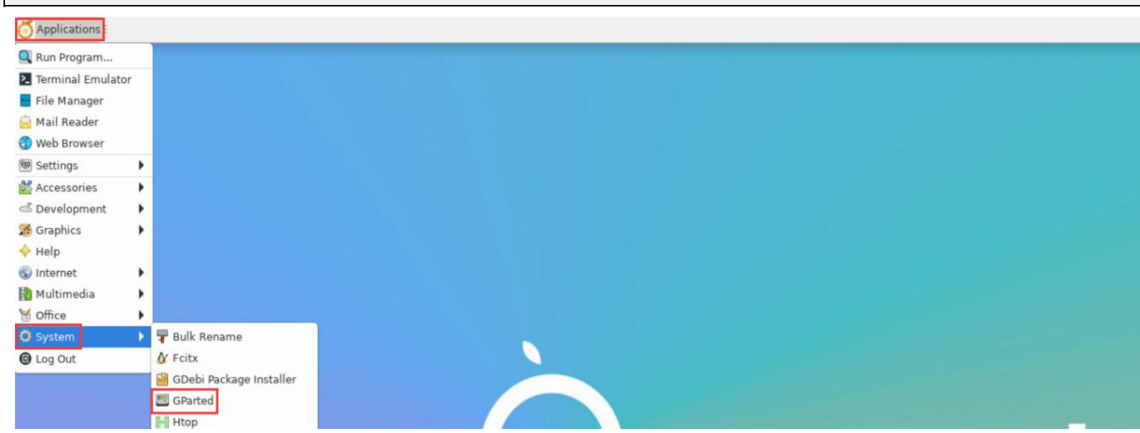
k. The display after burning is completed is as follows:



l. Then it is necessary to expand the capacity of the rootfs partition in the SSD, as follows:

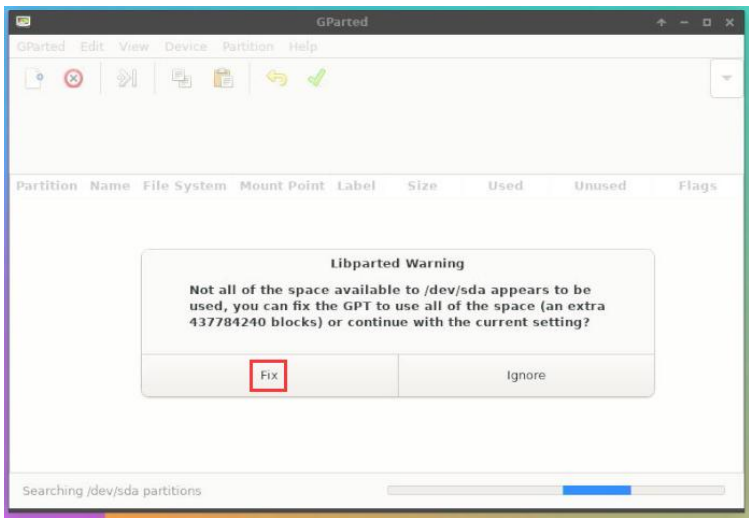
a) First, open **GParted**. If the system does not have Gparted pre installed, please use the apt. command to install it

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

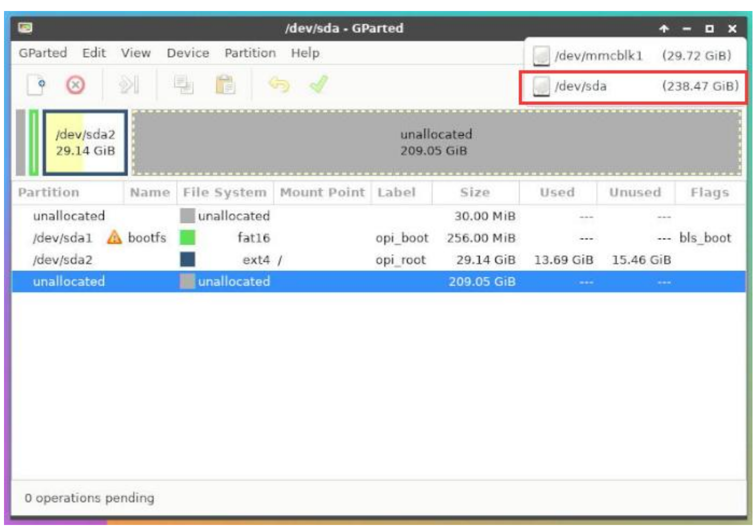


b) Then enter the password **orangepi** of the linux system, and click **Authenticate**

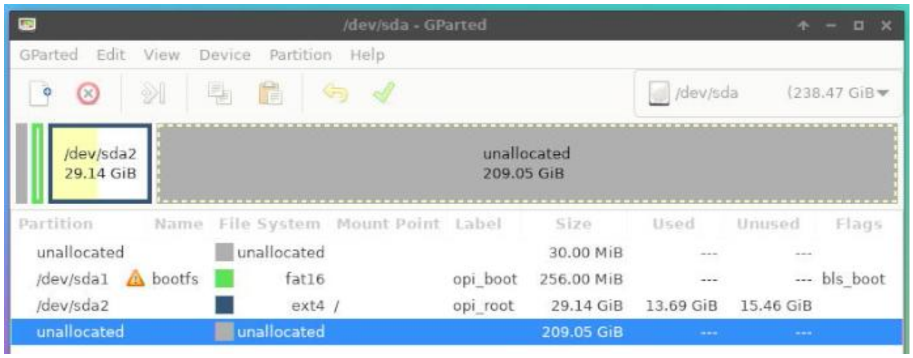
c) Then click **Fix**



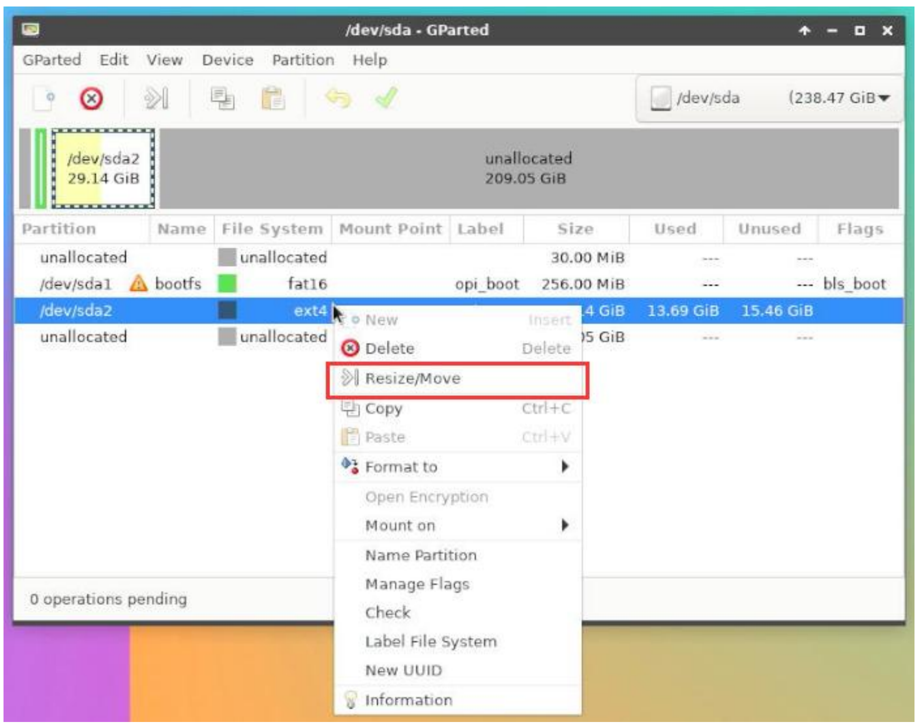
d) Then select SSD



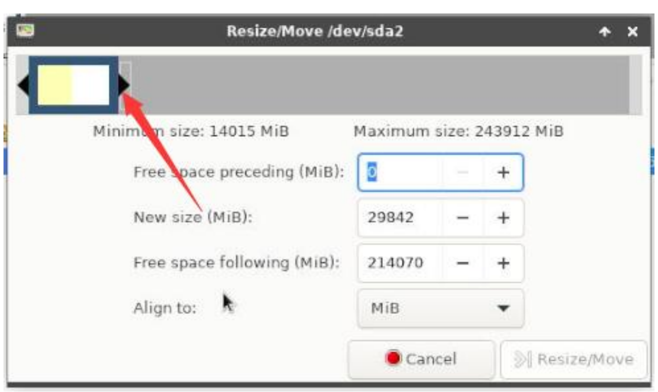
e) The display interface after selecting SSD is as follow：



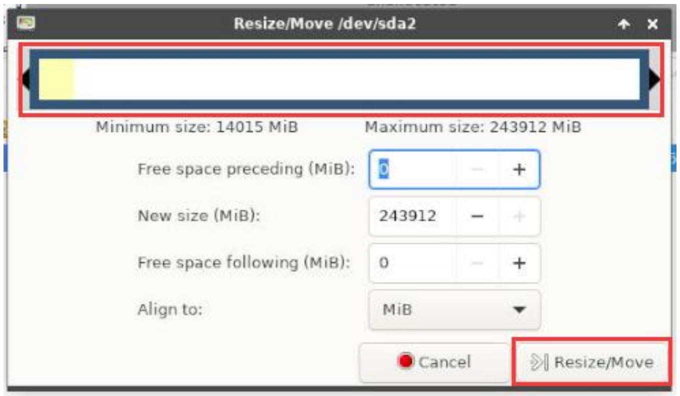
f) Then select the **/dev/sda2** partition, then right-click, and then select **Resize/Move**



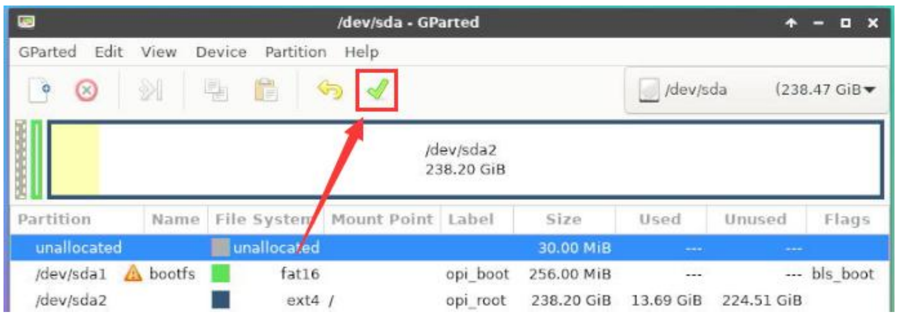
g) Then drag the capacity to the maximum at the position shown in the figure below



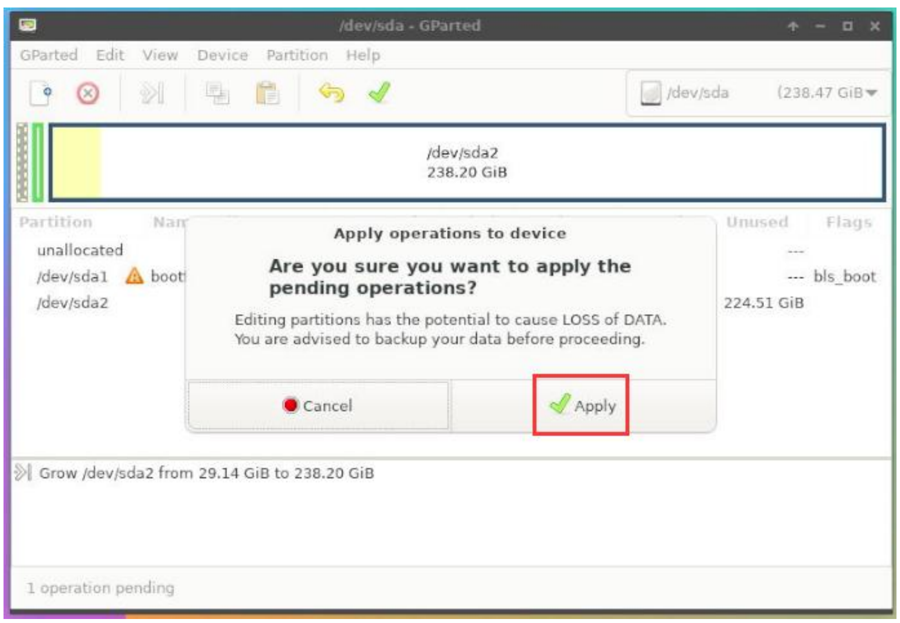
h) Then click **Resize/Move**



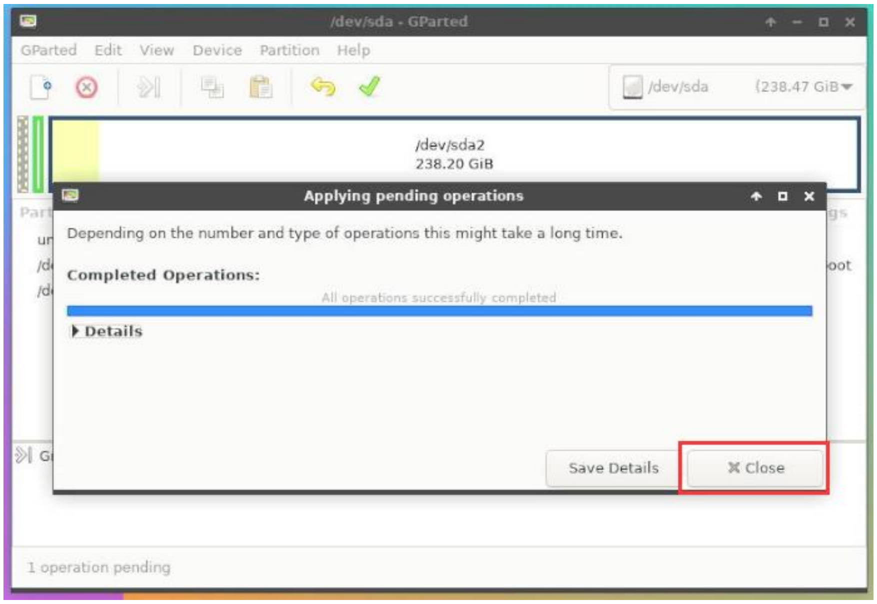
i) Then click the green √



j) Then click **Apply**



k) Then click **Close**

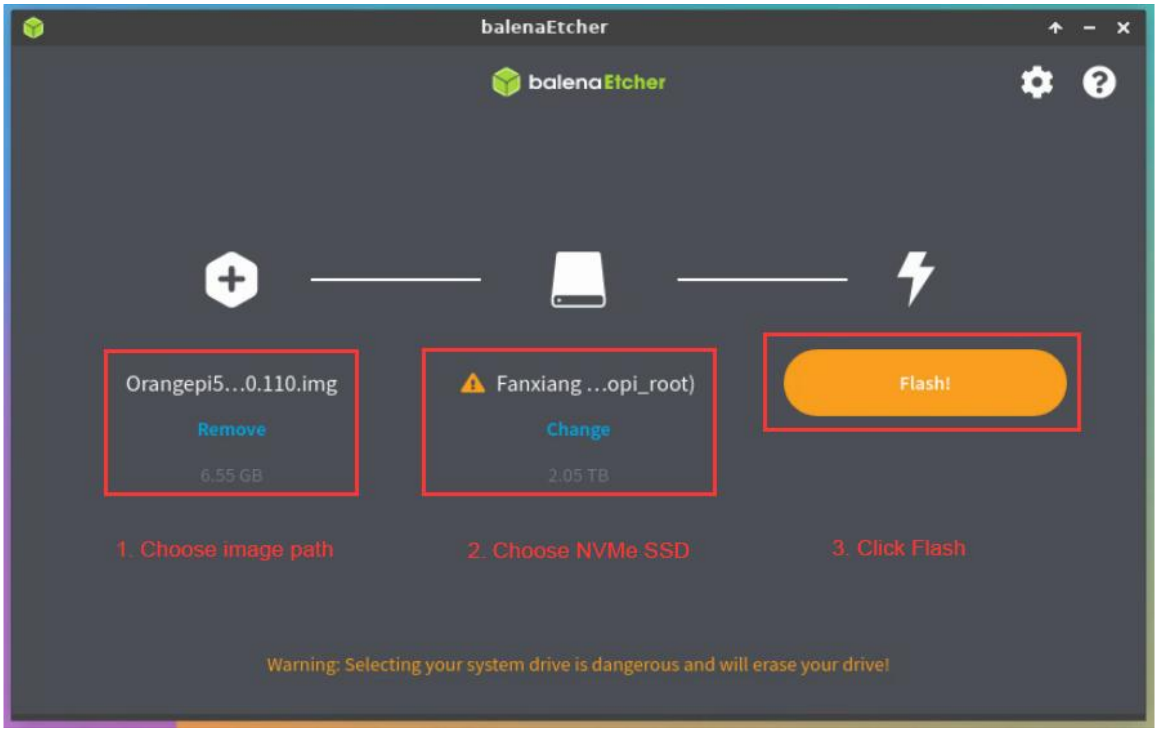


m. At this point, you can use the **sudo poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in SPIFlash+STAT SSD will be started.

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-sata2
```

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.

2.9. How to burn Linux image into SPI Flash+USB storage device

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

Before starting to burn the image, it is necessary to ensure that the development board has been equipped with an SPI Flash chip. The development board does not come with an SPI Flash chip attached from the factory, so it needs to be purchased and soldered on separately. We recommend using the XM25QU128CWIQT08Q model for the SPI Flash chip.

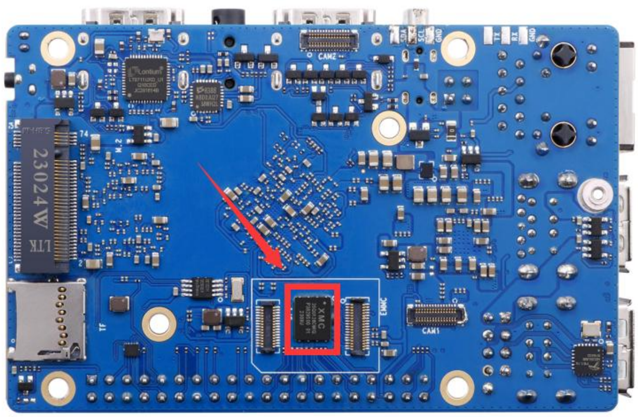
It should be noted that since the SPI Flash and eMMC share the same pins, attaching the SPI Flash will render the eMMC unusable.

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

2) Next, please refer to **the sections "Method for Burning Linux Image to TF Card Based on Windows PC"** and **"Method for Burning Linux Image to TF Card Based on Ubuntu PC"** for instructions on burning the Linux image to a USB storage device. There is no difference between burning the Linux image to a USB storage device and burning it to a TF card (when the TF card is inserted into a card reader, it functions like a USB flash drive).

3) Then insert the USB storage device with the burned Linux system into the USB port of the development board. **Note that only the three USB 2.0 ports shown in the figure below support booting the Linux system, while the blue USB 3.0 port does not support it.**

4) The location of PI Flash on the development board is shown in the figure below. No other settings are required before starting burning.



5) Burning the u-boot image to the SPIFlash requires the assistance of a TF card, so first, it is necessary to burn the Linux image to the TF card, and then use the TF card to boot the development board into the Linux system. Please refer to **the sections "Method for Burning Linux Image to TF Card Based on Windows PC" and "Method for Burning Linux Image to TF Card Based on Ubuntu PC"** for instructions on burning the Linux image to a TF card.

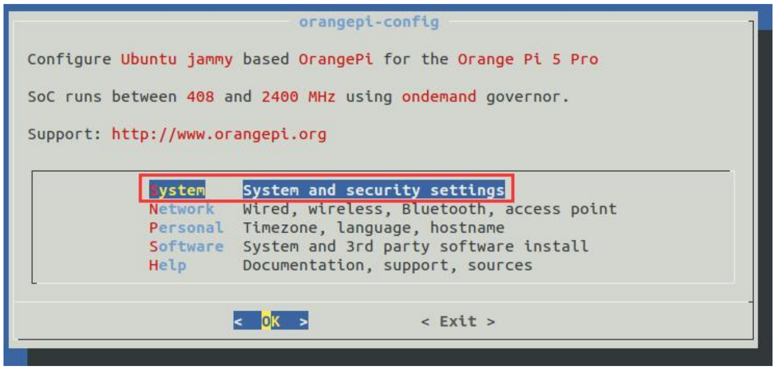
6) After using the TF card to start the Linux system, you can burn the u-boot image into SPI Flash.

a. In Linux systems, the SPI Flash function is turned off by default and needs to be turned on manually to use it. Detailed steps are as follows:

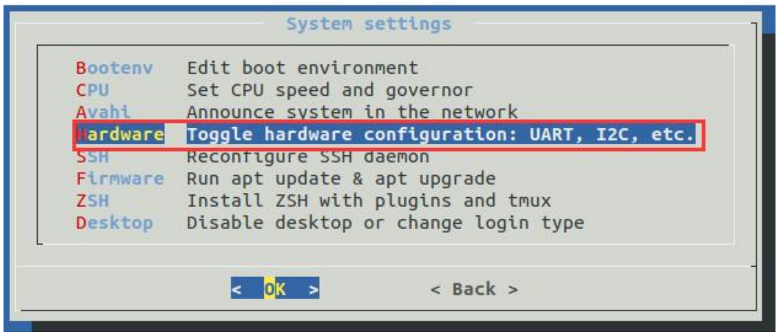
h) First run orangepi-config. **Ordinary users remember to add sudo permissions.**

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

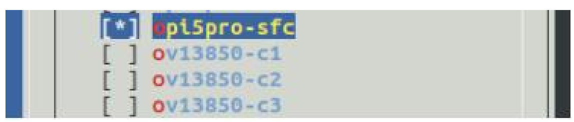
i) And then select **System**



j) And select **Hardware**

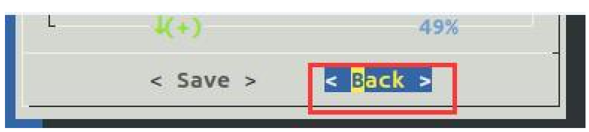


k) Then use the keyboard's arrow keys to locate **opi5pro-sfc**, and then use the space to select it.

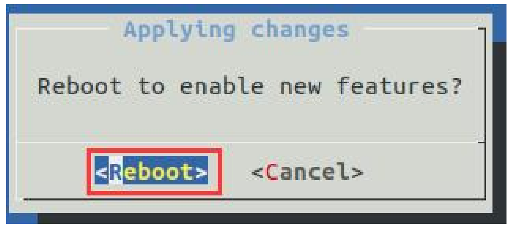


l) Then select**<Save>** to save

m) Select **<Back>**



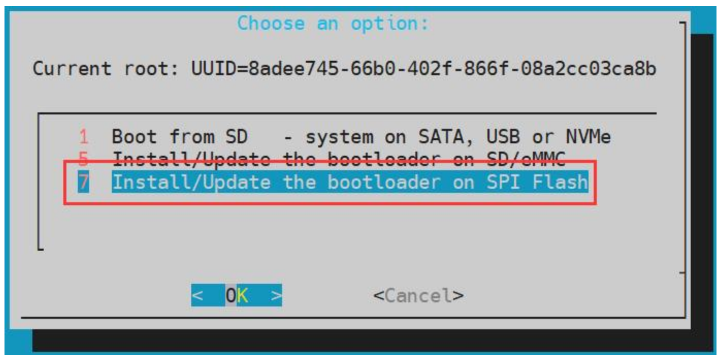
n) Then select **<Reboot>** to restart the system to make the configuration take effect.



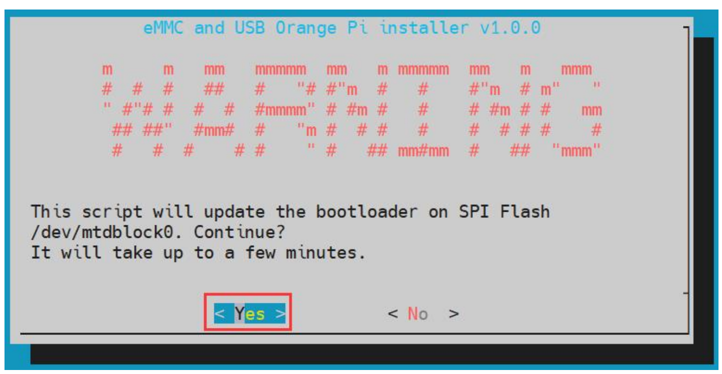
b. Then run **nand-sata-install.** **Ordinary users remember to add sudo permissions.**

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

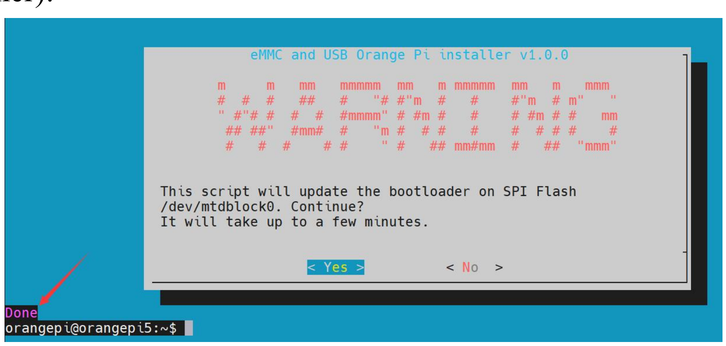
c. Select **7 Install/Update ther bootloader on SPI Flash**



d. And then select **<Yes>**



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



OPi OS Arch system does not have the configuration to open SPI Flash by default. It needs to be opened manually. Please add the following configuration to /boot/extlinux/extlinux.conf and restart the system to recognize SPI Flash.

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf

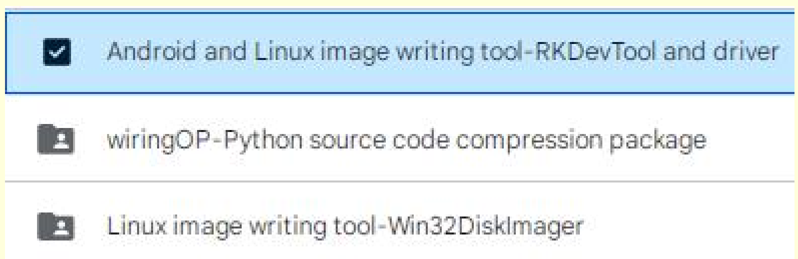
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-opi5pro-sfc.dtbo

There is no nand-sata-install script in OPi OS Arch system. Please use the following command to mirror u-boot to SPI Flash:

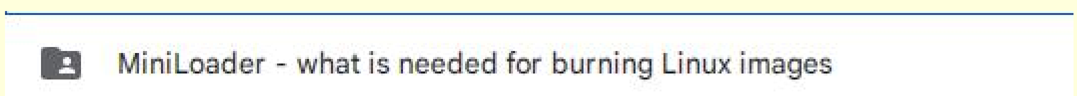
[orangepi@orangepi ~]$ **sudo dd if=/boot/rkspi_loader.img of=/dev/mtdblock0**

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:

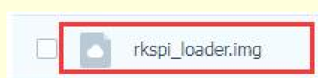
a. First enter the data download page of the development board, then select the official tool on the data download page, and then enter the folder below



b. Then choose to enter the directory below



c. And Download rkspi_loader.img



d. Then upload rkspi_loader.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.img of=/dev/mtdblock0**

7) 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 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. 10. Method of burning OpenWRT image into SPI FLASH

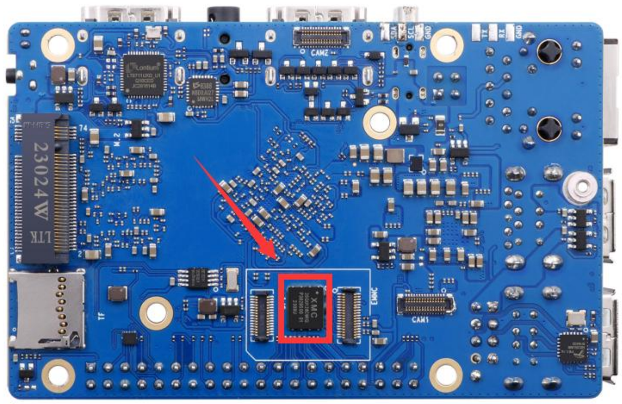
This section describes the method of burning the entire OpenWRT image into the SPI Flash, eliminating the need for an SSD or USB drive. This means that u-boot, kernel, and rootfs are all stored in the SPI Flash.

Before starting the image burning process, it is essential to ensure that the development board is equipped with an SPI Flash chip. Since the development board does not come with an SPI Flash chip attached from the factory, it needs to be purchased and soldered on separately. We recommend using the XM25QU128CWIQT08Q model for the SPI Flash chip.

It should be noted that since the SPI Flash and eMMC share the same pins, attaching the SPI Flash will render the eMMC unusable.

2. 10. 1. How to burn using RKDevTool

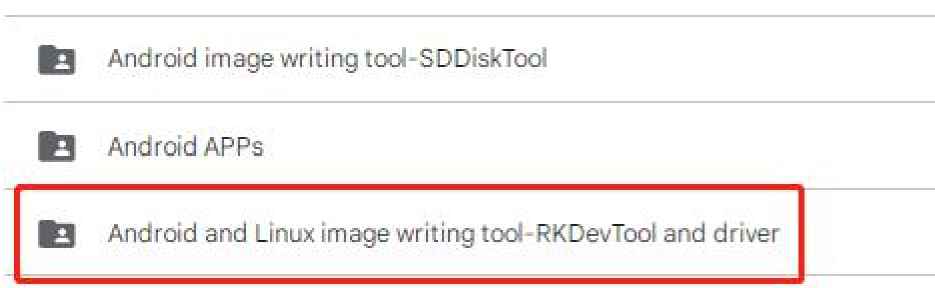
1) The location of SPI Flash on the development board is as shown in the picture below. No other settings are required before starting programming.



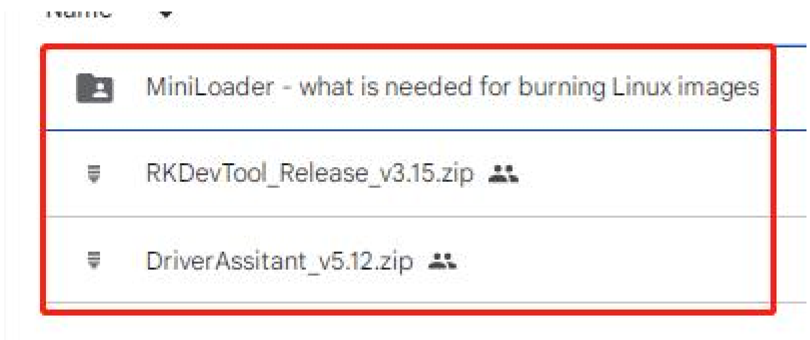
2) Then you need to prepare a good quality USB male-to-male data cable

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

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

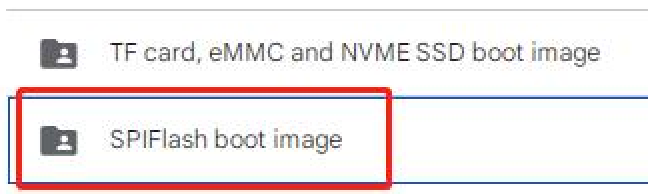


b. Then download all the files below

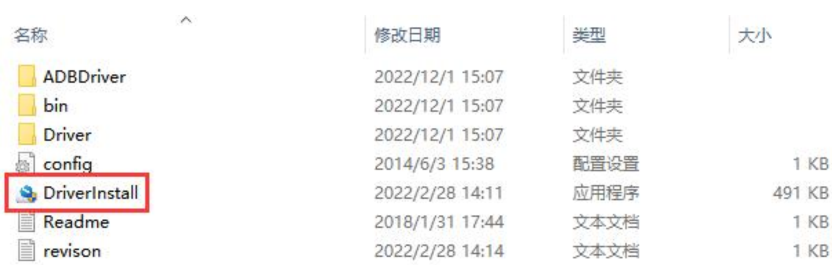


Note that the "MiniLoader-things needed to burn Linux images" folder will be referred to as the MiniLoader folder below.

4) Then 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**. 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.

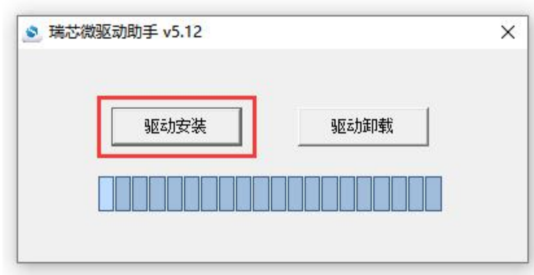


5) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

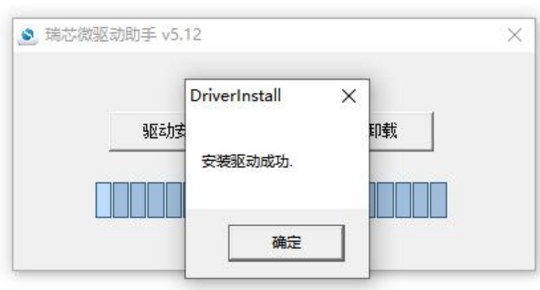


6) Open **DriverInstall.exe** and install the Rockchip microdriver as follows:

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.

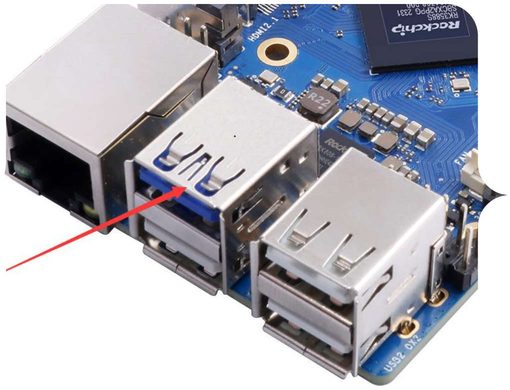


7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

8) After opening the **RKDevTool** burning tool, because the computer has not 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.

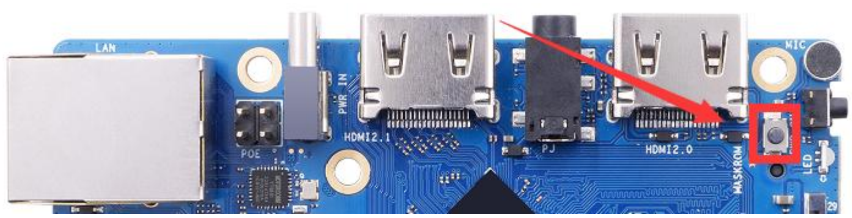
9) Then start burning the OpenWRT image into SPI FLASH

a. First, connect the development board to the Windows computer through a USB male-to-male data cable. The location of the USB burning port on the development board is as shown in the figure below.

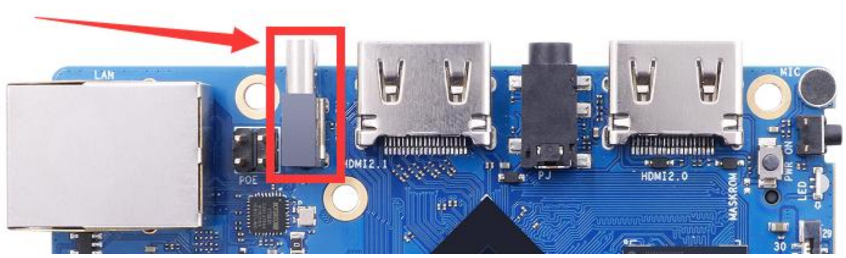


b. Make sure the development board is not connected to the power supply and the TF card is not inserted.

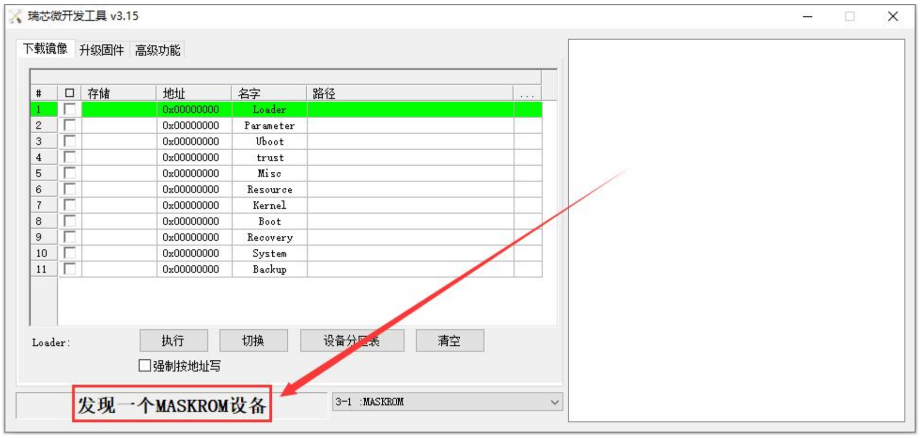
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



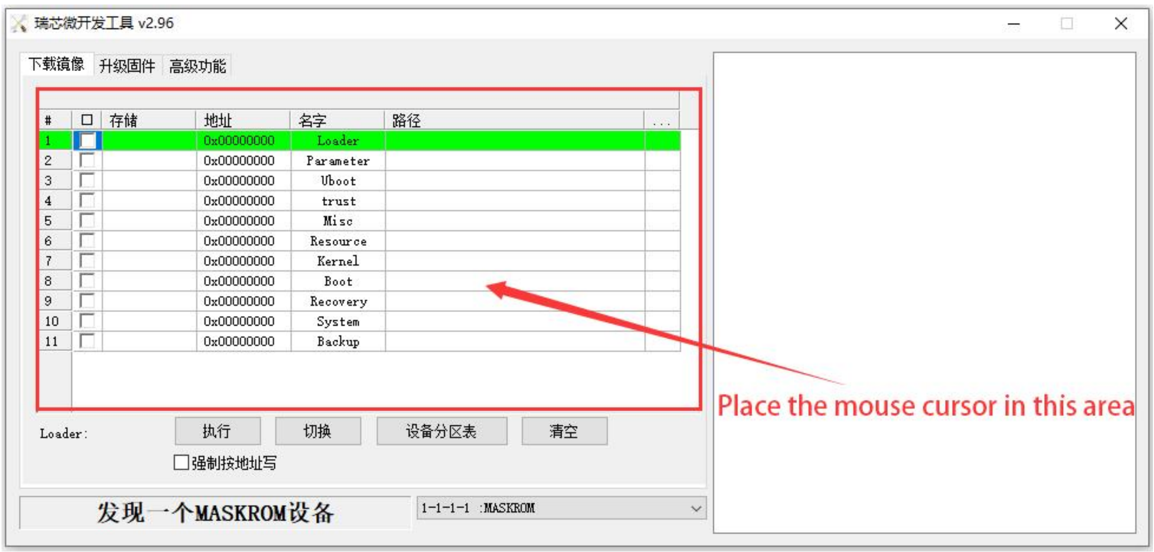
d. Then connect the development board to the power supply of the Type-C interface, power it on, and then release the MaskROM button.



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

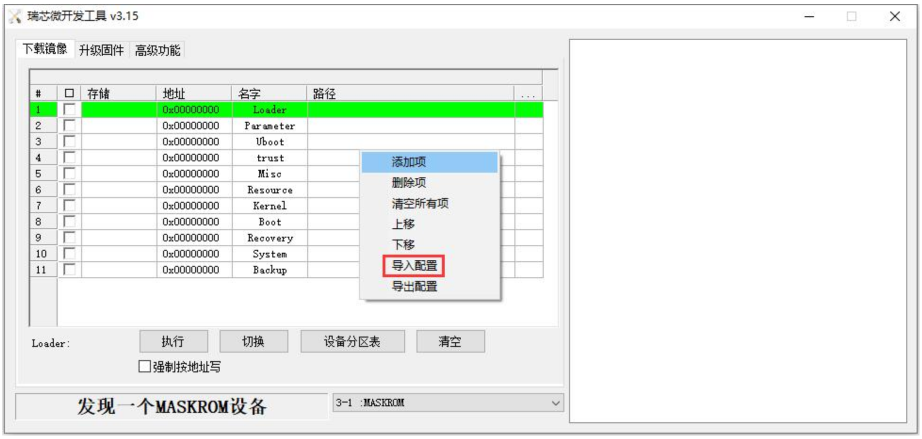


f. Then place the mouse cursor in the area below

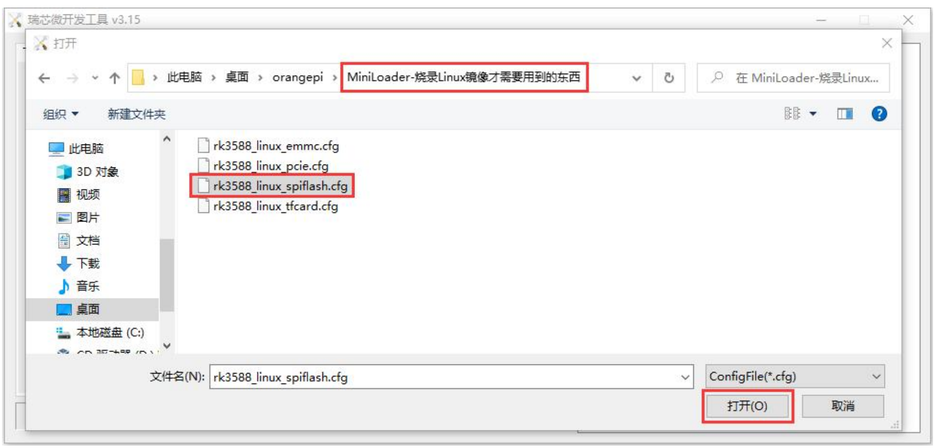


g. Then click the right button of the mouse and the selection interface shown below will pop up.

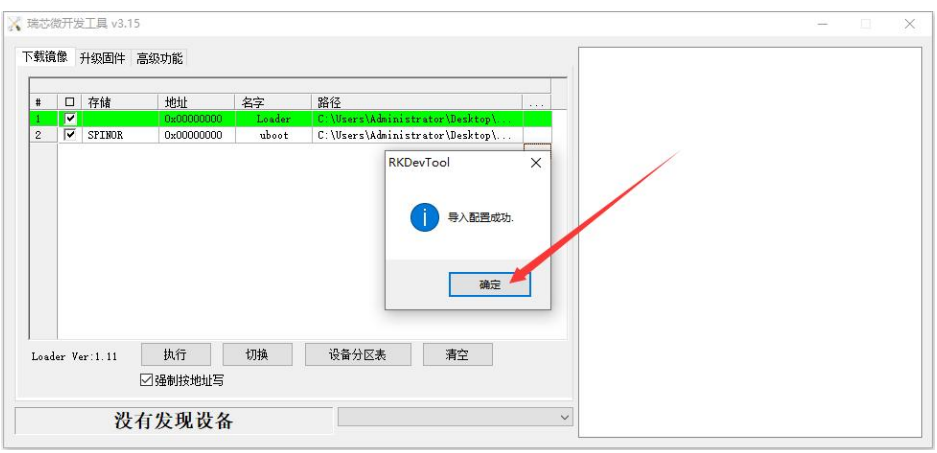
h. Then select the **Import Configuration** option



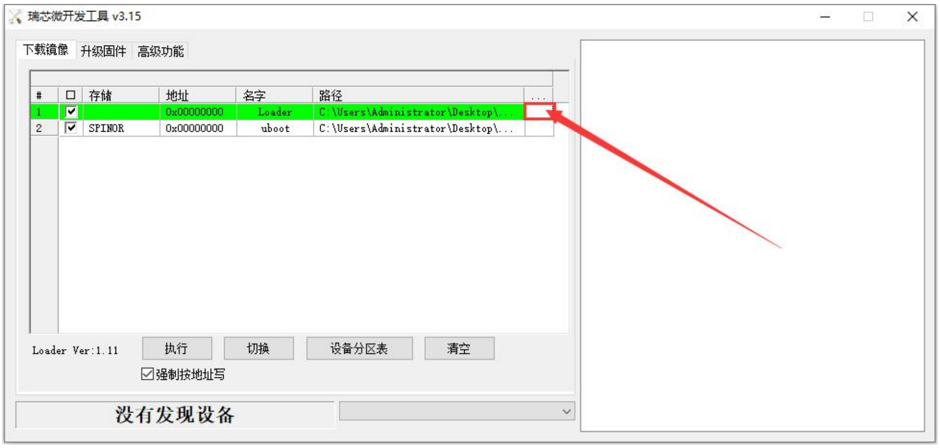
i. Then select the **rk3588_linux_spiflash.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and then click Open



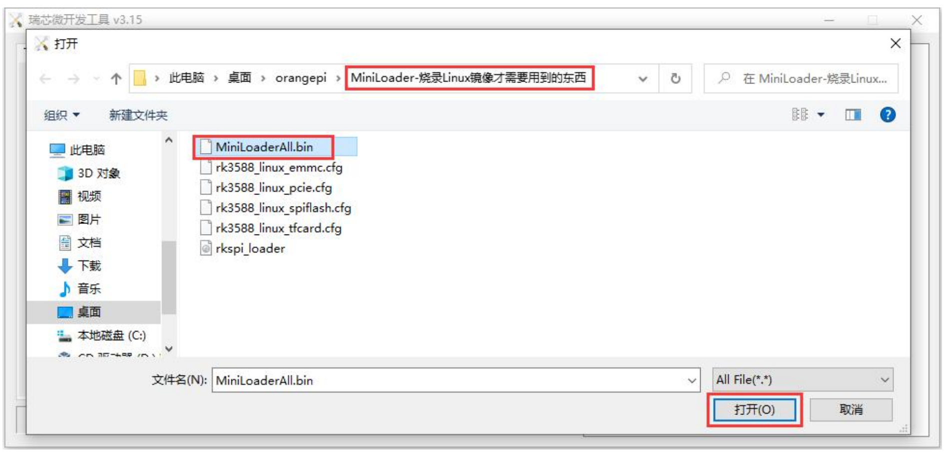
j. Click **YES**



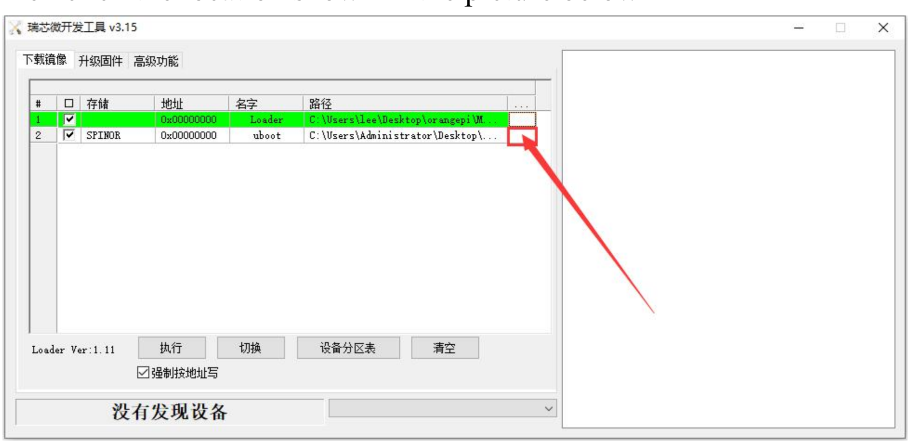
k. Then click the location shown in the picture below



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



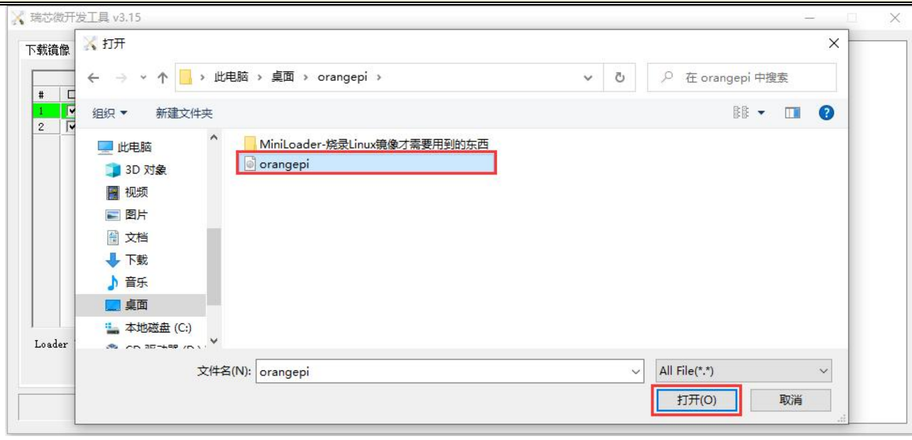
m. Then click the location shown in the picture below



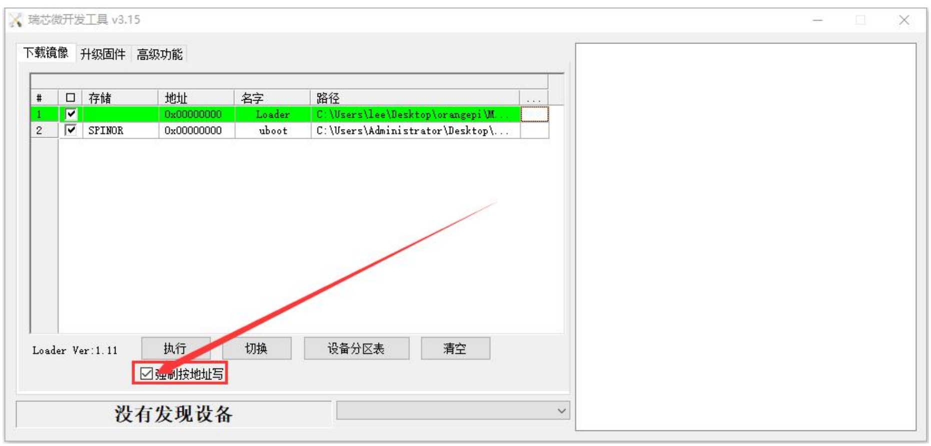
n. Then select the path of the OpenWRT image you want to burn, and then click Open

Before burning the image, it is recommended to rename the OpenWRT image to

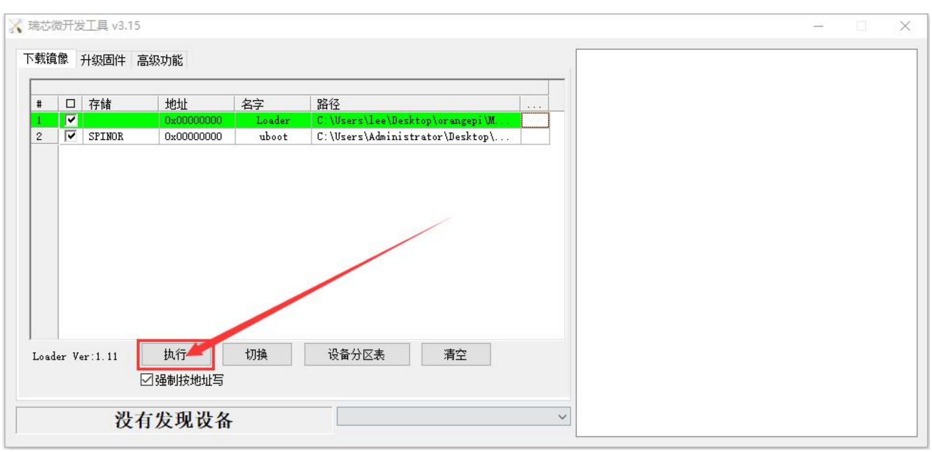
something short like "orangepi.img" or another concise name. This way, during the image burning process, you can see the percentage progress of the burn.



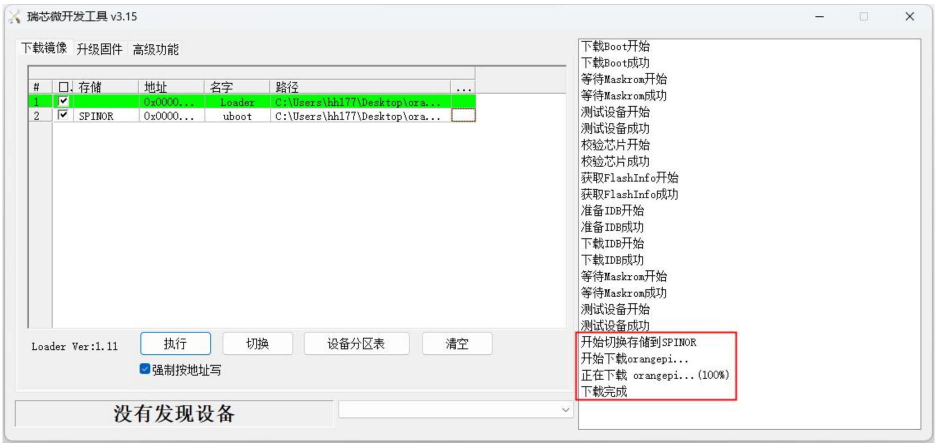
o. Then make sure the **Force writing by address** option is checked.



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



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



If there is a problem with burning, please clear SPIFlash first and then try burning again. For the method of clearing SPIFlash, please refer to the instructions in the section "How to clear SPIFlash using RKDevTool".

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

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

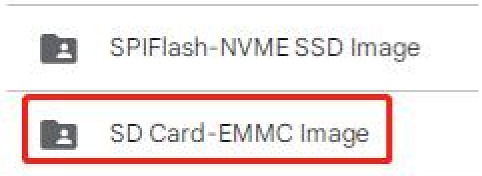
2. 11. 1. How to burn using RKDevTool

1) First you need to prepare a good quality USB2.0 male-to-male data cable

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

3) Then download the Android 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 the **TF card and eMMC boot image** folder to download.

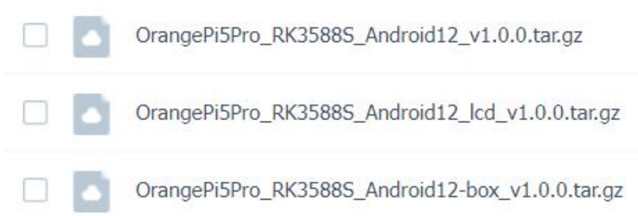


b. After entering the **TF card and eMMC boot image** folders, you can see the following three images. Their differences are:

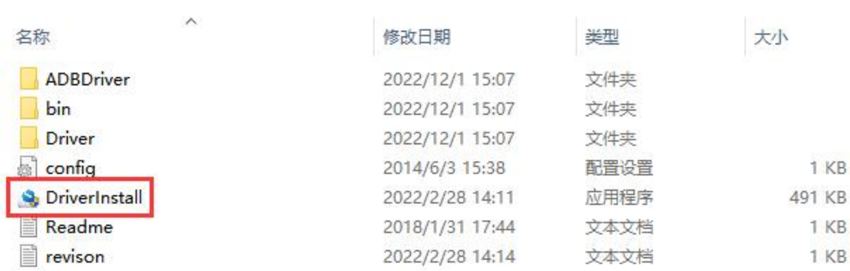
a) The first image is specially used for HDMI display and supports 8K display. If you do not use an LCD screen, please download the image without LCD.

b) If you want to use an LCD screen, please choose the mirror with LCD

c) The image with box is an image dedicated to TV box

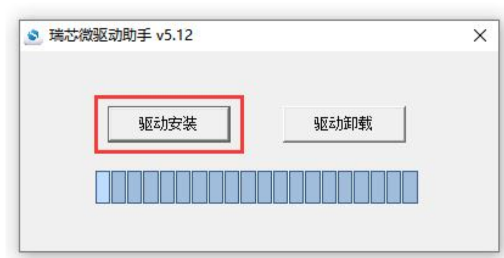


4) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.



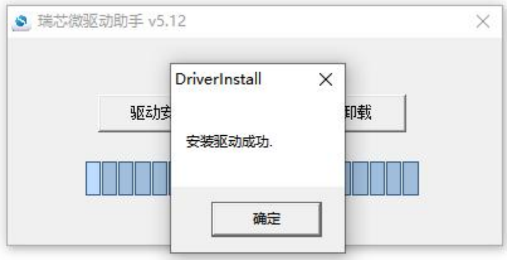
5) Open **DriverInstall.exe** and install the Rockchip microdriver as follows:

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.

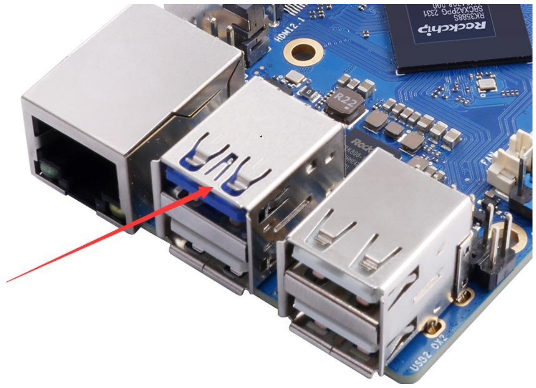


6) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

7) After opening the **RKDevTool**burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a message "**No device found**" will appear in the lower left corner.

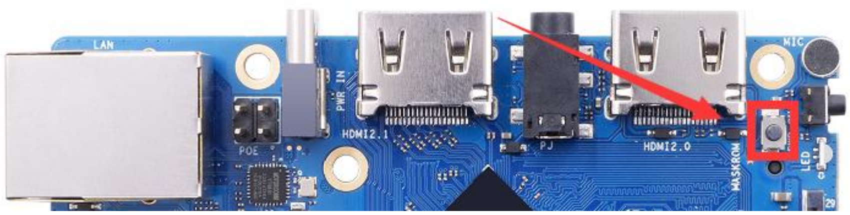
8) Then start burning the Android image to the TF card

a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the development board's USB programming port is as shown in the figure below.

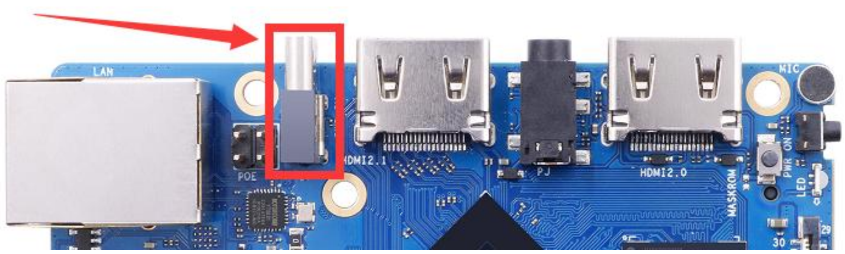


b. Then insert the TF card into the development board and make sure that the development board is not connected to the power supply.

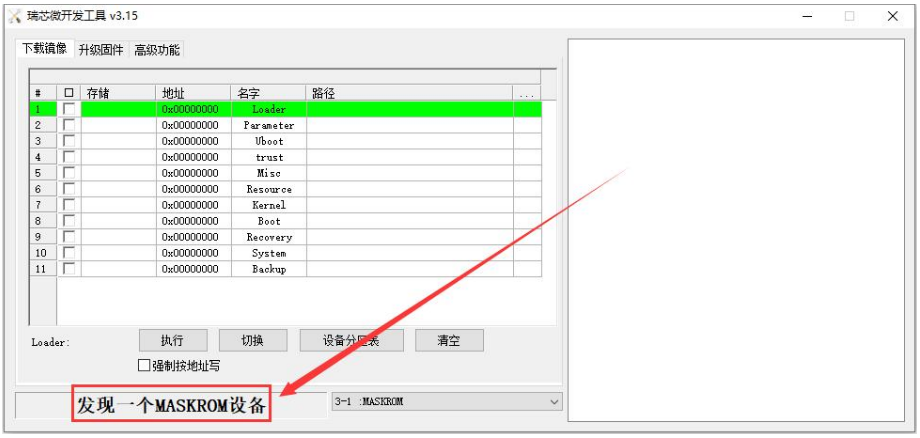
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



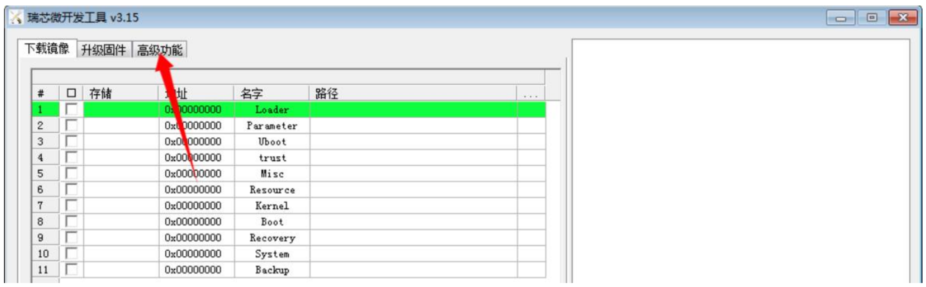
d. Then connect the development board to the power supply of the Type-C interface, power it on, and then release the MaskROM button.



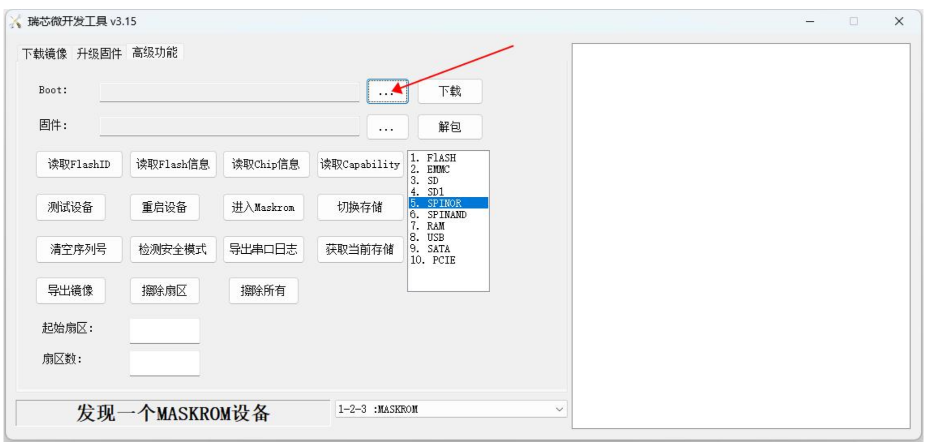
e. If the previous steps go well, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device was found**"



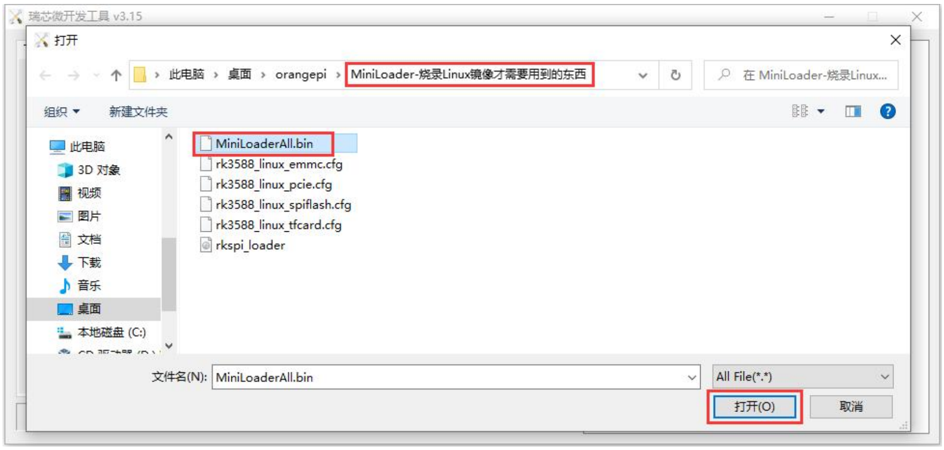
f. Then please select **Advanced Features**



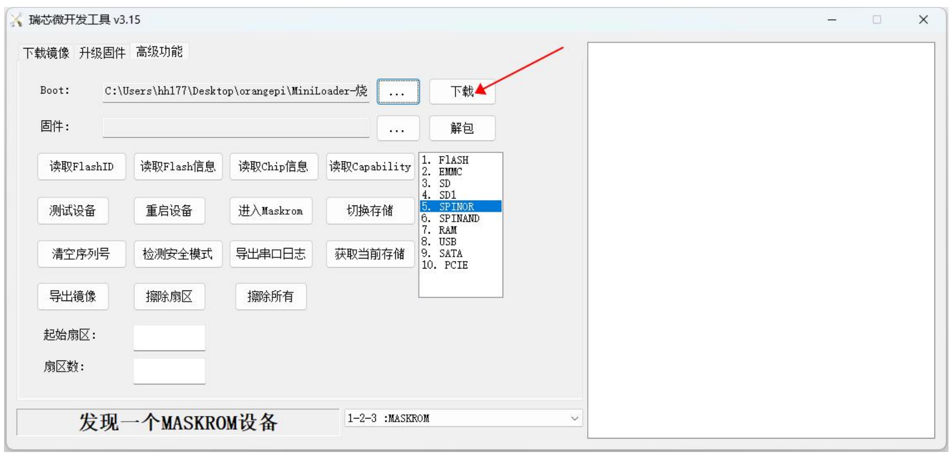
g. Then click the location shown in the picture below



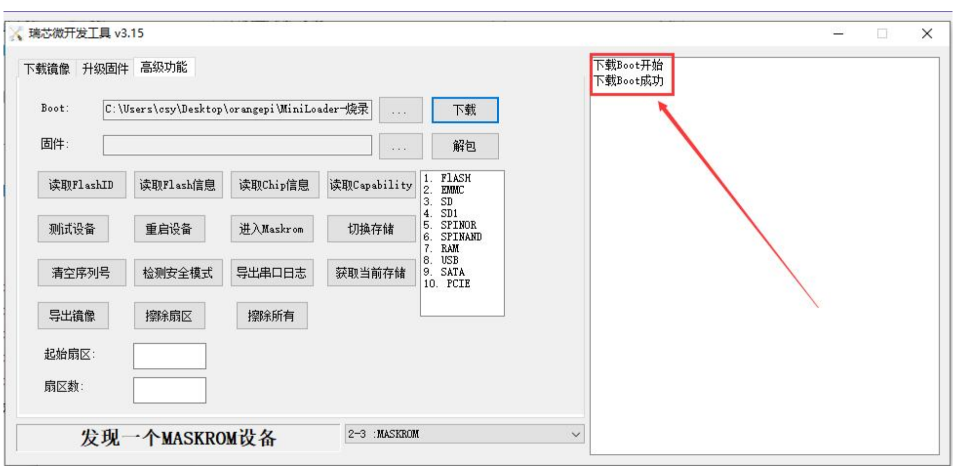
h. Then select **MiniLoaderAll.bin** in the **MiniLoader**r folder downloaded earlier, and then click Open



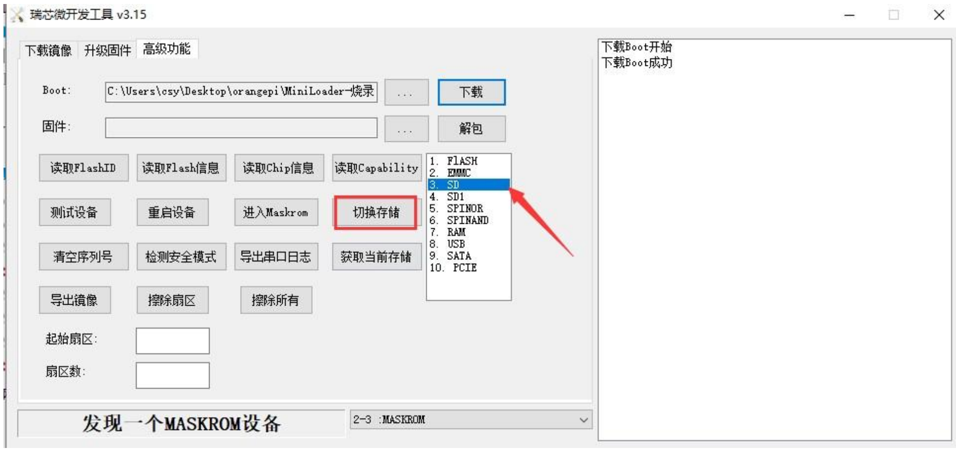
i. Then click **Download**



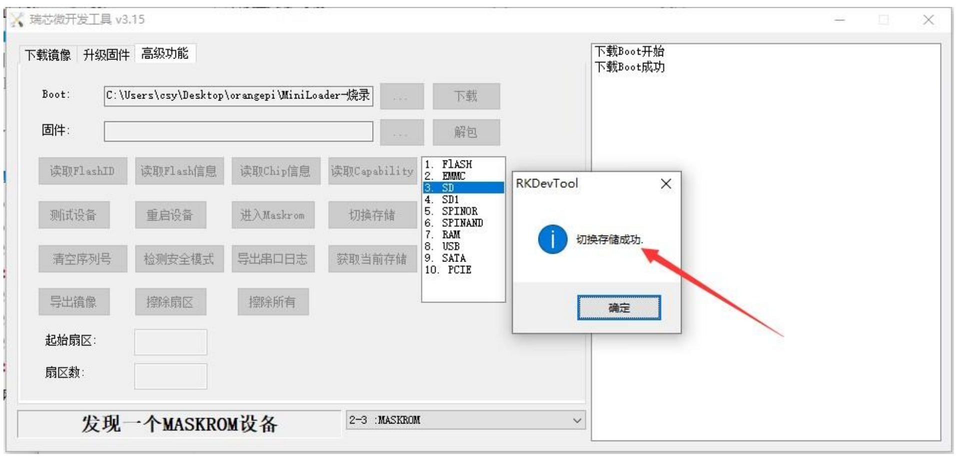
j. After downloading **MiniLoaderAll.bin**, the display is as shown below



k. Then select the storage device as **SD**, and then click to switch storage



l. The successful switching is displayed as shown below



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

n. Then click the "**Firmware**" button to select the path of the Android image that needs to be burned.

o. Finally, click the "**Upgrade**" button to start burning. The log during the burning process is as shown below. After the burning is completed, the Android system will automatically start.

2. 11. 2. How to burn using SDDiskTool tool

This method is not recommended. Some TF cards are prone to stuck on the boot logo and cannot be started.

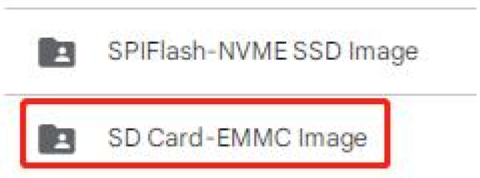
1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use TF cards from SanDisk and other brands.

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

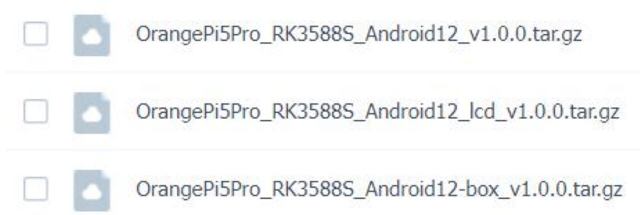
3) Then download the SDDiskTool programming tool from the Orange Pi download page. **Please ensure that the version of the SDDiskTool tool is the latest v1.72**

4) Then download the 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 the **TF card and eMMC boot image** folder to download.

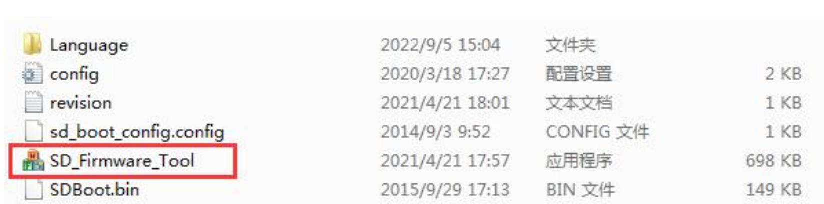


b. After entering the **TF card and eMMC boot image** folders, you can see the following two images. The difference between them is:
a) The first image is specially used for HDMI display and supports 8K display. If you do not use an LCD screen, please download the image without LCD.
b) If you want to use an LCD screen, please choose the mirror with lcd
c) The image with box is an image dedicated to TV box



5) Then use decompression software to decompress the compressed package of the downloaded Android image. In the decompressed file, the file ending with ".img" is the Android image file, 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.

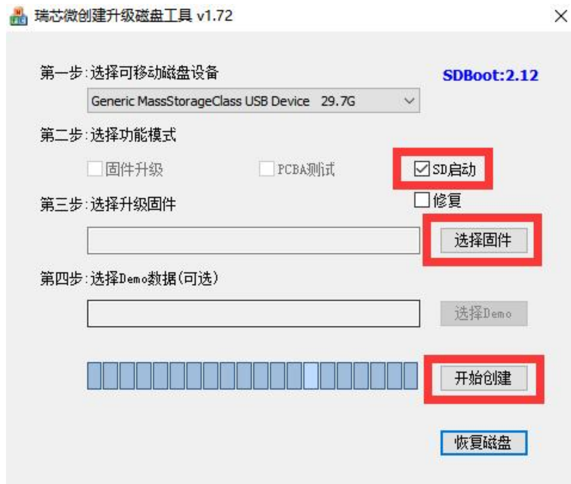
8) After confirming the drive letter, you can format the TF card first and click the **Recover Disk** button in **SDDiskTool**. You can also use the **SD Card Formatter** mentioned earlier to format the TF card.

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

a. First check "SD boot" in "Select function mode"

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

c. Finally, click the "Start Creating" button to start burning the Android 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.

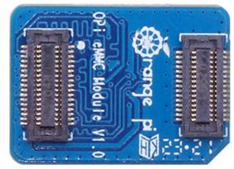
2. 12. Method of burning Android image into eMMC

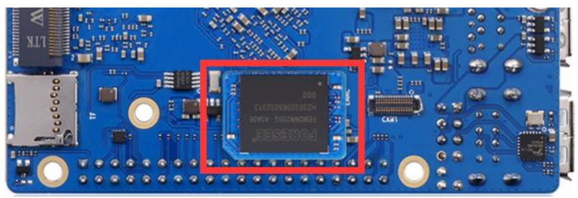
2. 12. 1. How to burn using RKDevTool

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

1) The development board has reserved eMMC expansion interface. Before burning the system to eMMC, you first need to purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC module to the development

board. The eMMC module and the method of inserting the development board are as follows:



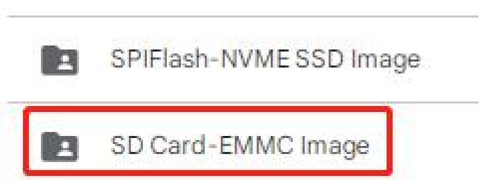


2) You also need to prepare a good quality USB2.0 male-to-male data cable

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

4) Then download the Android image from the Orange Pi download page.

a. After opening the download link of the Android image, you can see the following two types of Android images. Please select the image in the TF card and eMMC boot image folder to download.

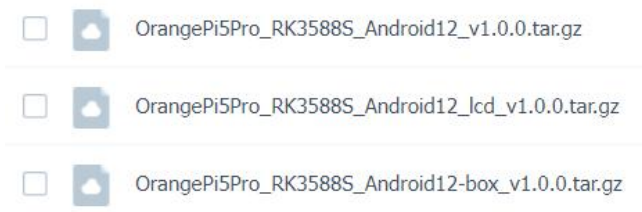


b. After entering the TF card and eMMC boot image folders, you can see the following three images. Their differences are:

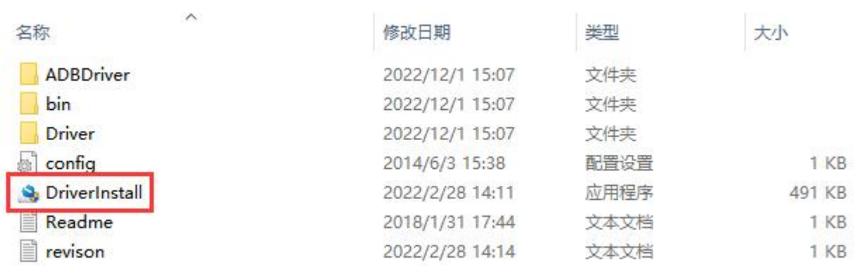
a) The first image is specially used for HDMI display and supports 8K display. If you do not use an LCD screen, please download the image without LCD.

b) If you want to use an LCD screen, please choose the mirror with LCD

c) The image with box is an image dedicated to TV box

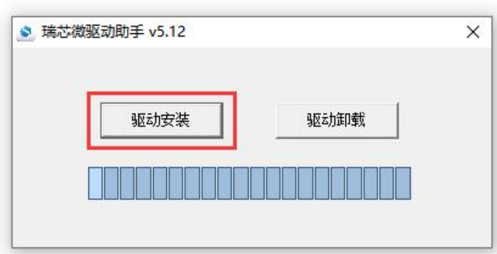


5) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

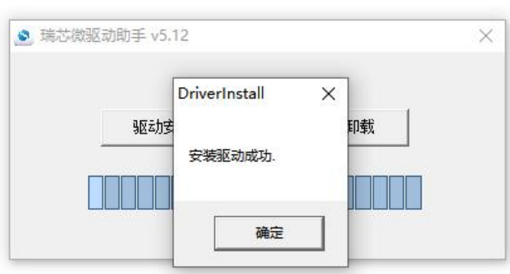


6) Open **DriverInstall.exe** and install the Rockchip microdriver as follows:

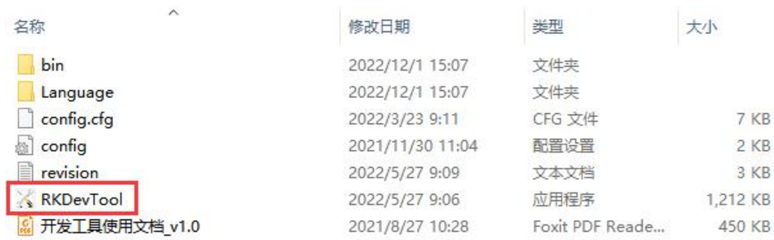
a. Click the "**Driver Installation**" button



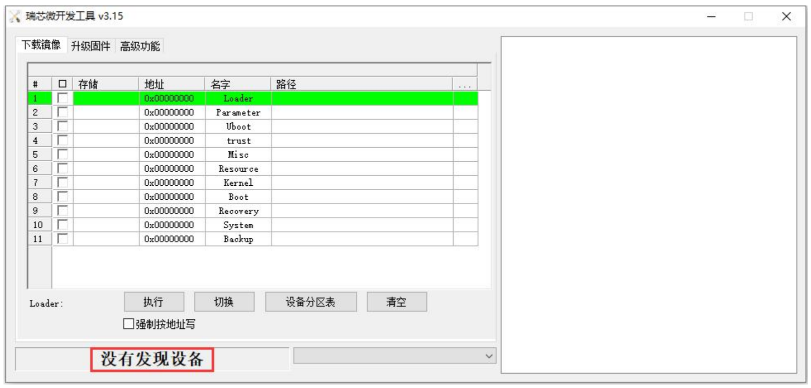
b. After waiting for a period of time, a window will pop up prompting "Driver installation successful", then click the "OK" button.



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

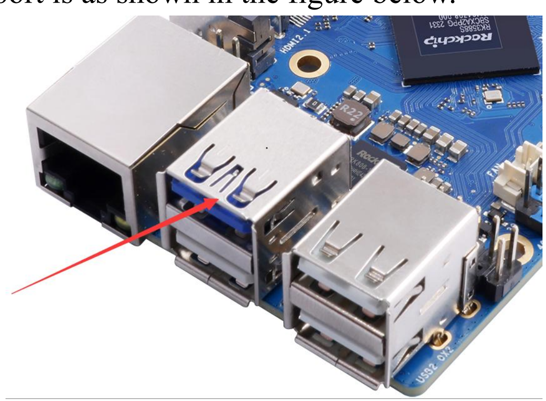


8) After opening the **RKDevTool** burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a message "**No device found**" will appear in the lower left corner.



9) Then start burning the Android image into eMMC

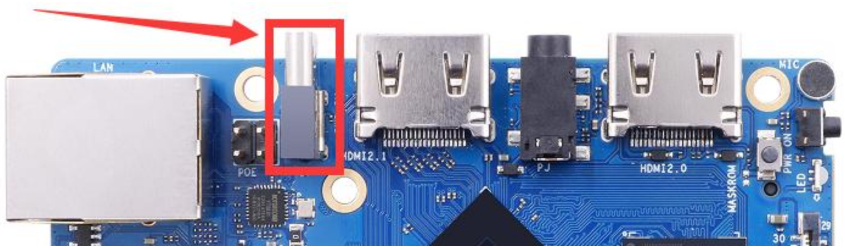
a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the development board's USB programming port is as shown in the figure below.



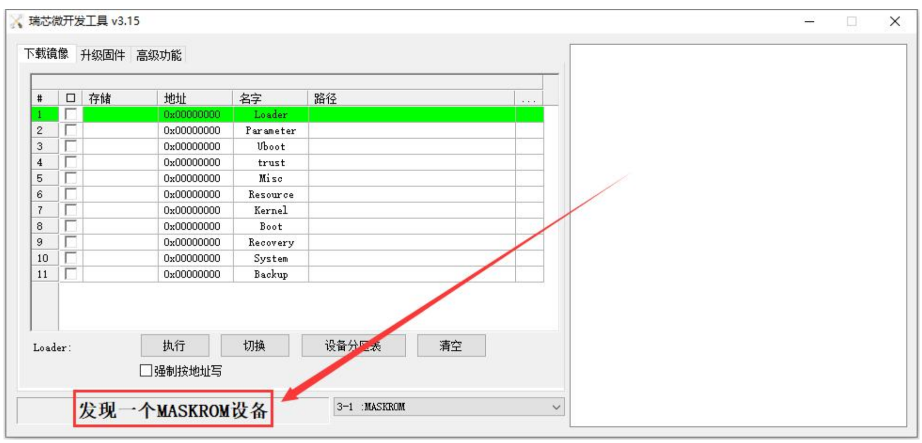
b. Make sure the development board is not connected to the power supply and the TF card is not inserted.

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

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



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



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

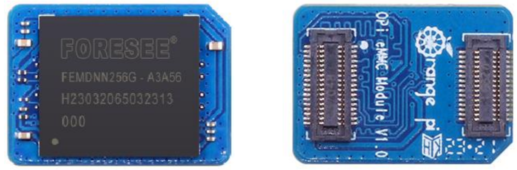
g. Then click the "**Firmware**" button to select the path of the Android image that needs to be burned.

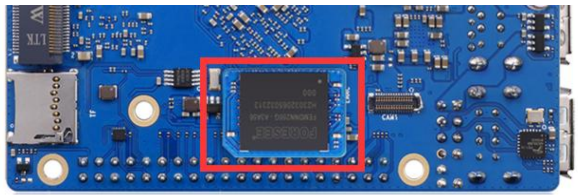
h. Finally, click the "**Upgrade**" button to start burning. The log during the burning process is as shown below. After the burning is completed, the Android system will automatically start.

2.12.2. How to burn using SDDiskTool tool

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

1) The development board has reserved eMMC expansion interface. Before burning the system to eMMC, you first need to purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC module to the development board. The eMMC module and the method of inserting the development board are as follows:





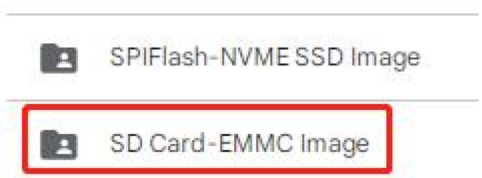
2) You also need to prepare a TF card with a capacity of 8GB or larger. The transmission speed of the TF card must be class10 or above. It is recommended to use TF cards from SanDisk and other brands.

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

4) Then download the SDDiskTool burning 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 Android image **from the Orange Pi download page.**

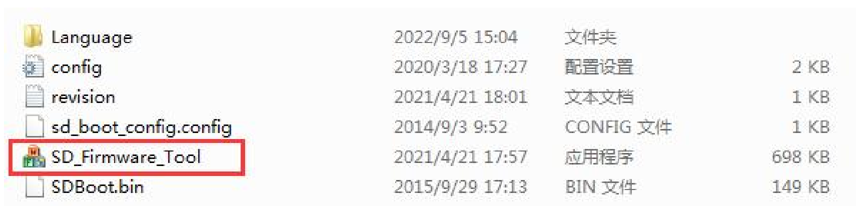
a. After opening the download link of the Android image, you can see the following two types of Android images. Please select the image in **the TF card and eMMC boot image** folder to download.



b. After entering the **TF card and eMMC boot image** folders, you can see the following three images. Their differences are:
a) The first image is specially used for HDMI display and supports 8K display. If you do not use an LCD screen, please download the image without LCD.
b) If you want to use an LCD screen, please choose the mirror with LCD
c) The image with box is an image dedicated to TV box

6) Then use decompression software to decompress the compressed package of the downloaded Android image. In the decompressed file, the file ending with ".img" is the Android image file, with a size of 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.** Yes, if there is no display, you can try to remove the TF card.

9) 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.

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

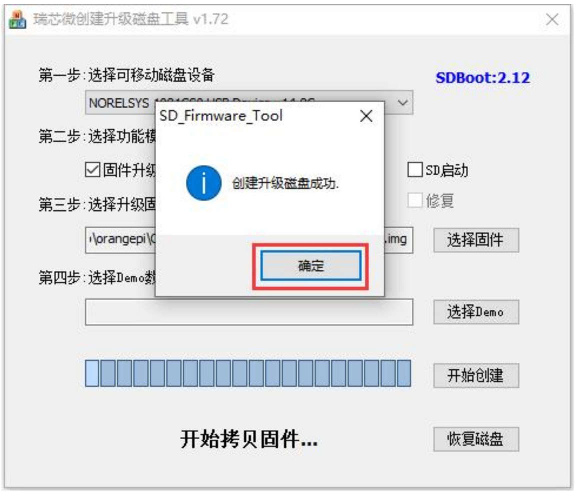
a. First, confirm that the drive letter displayed under "**Select Removable Disk Device**" is the drive letter corresponding to the TF card.

b. Then select "**Firmware Upgrade**" in "**Select Function Mode**"

c. Then select the path to the Android firmware in the "**Select upgrade firmware**" column

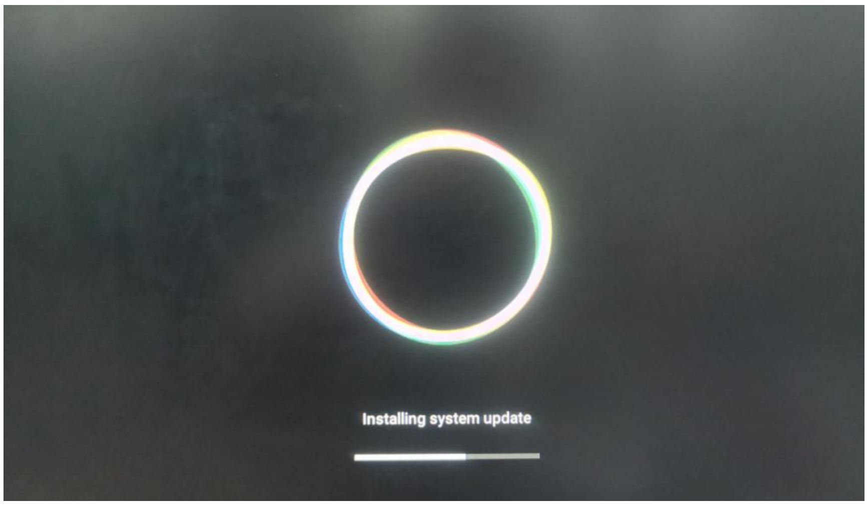
d. Finally, click the "**Start Creating**" button to start burning.

11) After the burning is completed, the display is as shown below, and then you can exit SDDiskTool

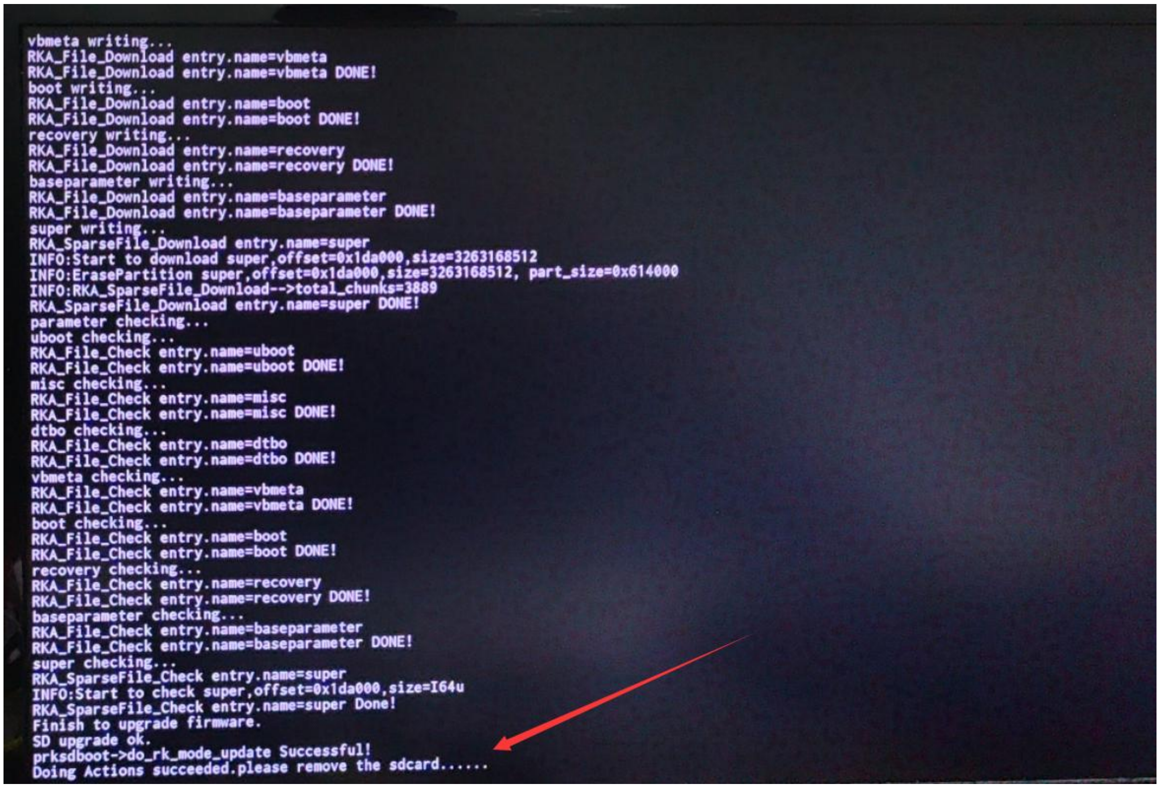


12) Then pull out the TF card from the computer and insert it into the development board. After the development board is powered on, it will automatically start burning the Android image in the TF card into the eMMC of the development board.

13) If the development board is connected to an HDMI display, you can also see the progress bar of burning the Android image to the eMMC from the HDMI display.



14) When the HDMI display shows the following information, it indicates that the burning of the Android image to the eMMC has been completed. At this point, you can remove the TF card, and the Android system in the eMMC will begin to boot.



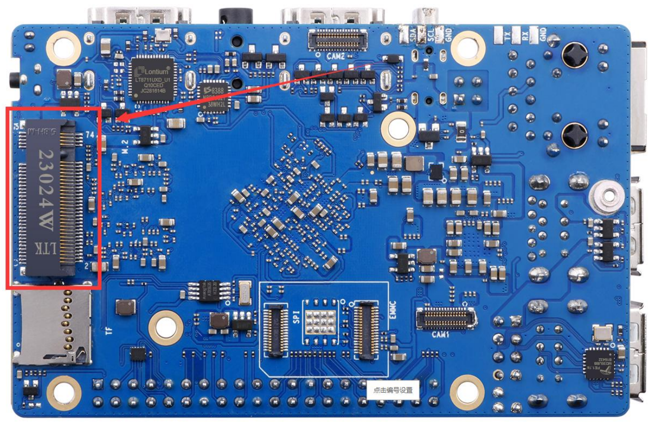
2.13. Method of burning Android image into SPIFlash+NVMe SSD

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

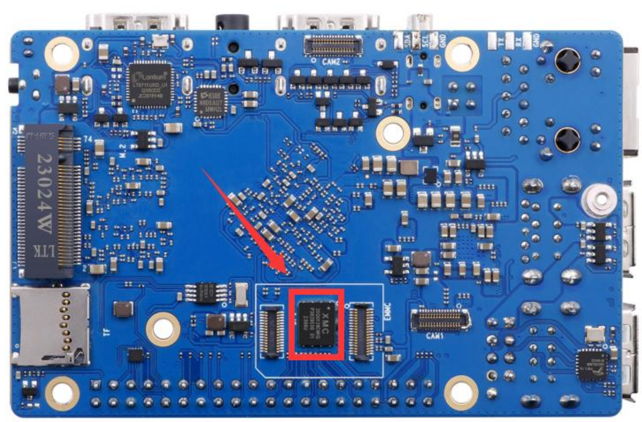
Before starting to burn the image, you must ensure that the development board has been attached with the SPI Flash chip. Because the development board does not have the SPI Flash chip attached when it leaves the factory, you need to purchase and solder it on by yourself. The SPI Flash chip model we recommend is XM25QU128CWIQT08Q. **Since SPI Flash and eMMC reuse the same pins, eMMC cannot be used if SPI Flash is attached. XM25QU128CWIQT08Q**

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

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



3) Please make sure that the development board has been attached with SPI Flash. The location of SPI Flash on the development board is as shown in the picture below. No other settings are required before starting burning.

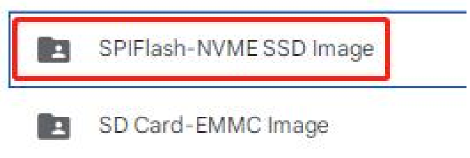


4) You also need to prepare a good quality USB2.0 male-to-male data cable

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

6) Then download the image of Android12

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.

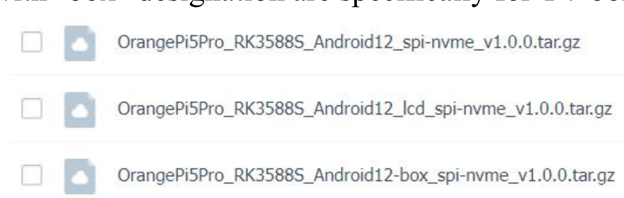


b. After entering the **SPIFlash-NVME SSD** folder, you can see the following three images. Their differences are:

a) The first image is specifically for HDMI display, supporting 8K display. If you're not using an LCD screen, please download the image without LCD support.

b) If you intend to use an LCD screen, please select the image with LCD support.

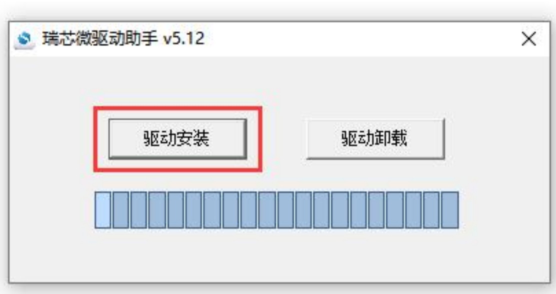
c) Images with "box" designation are specifically for TV boxes.



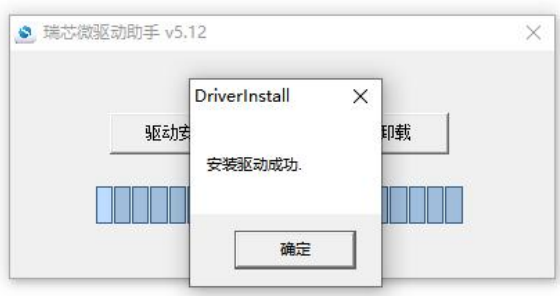
7) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

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

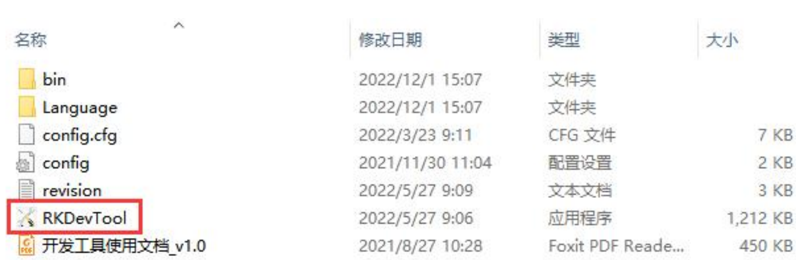
a. Click the "**Driver Installation**" button



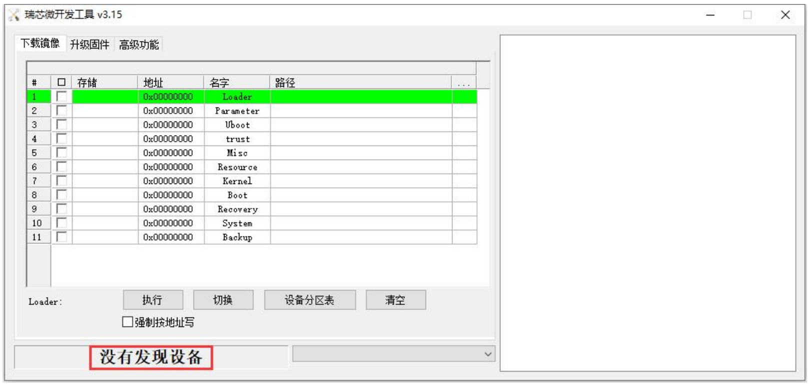
b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



9) 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.

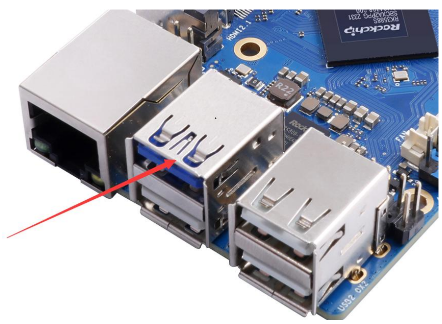


10) After opening the **RKDevTool** burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a message "**No device found**" will be displayed in the lower left corner.



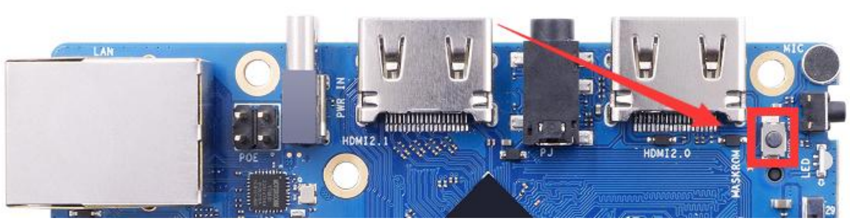
11) Then start burning the Android image to SPIFlash+NVMe SSD

a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the development board's USB programming port is as shown in the figure below.

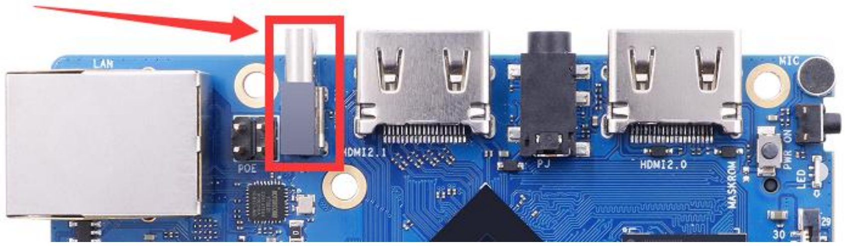


b. Make sure that the TF card is not inserted into the development

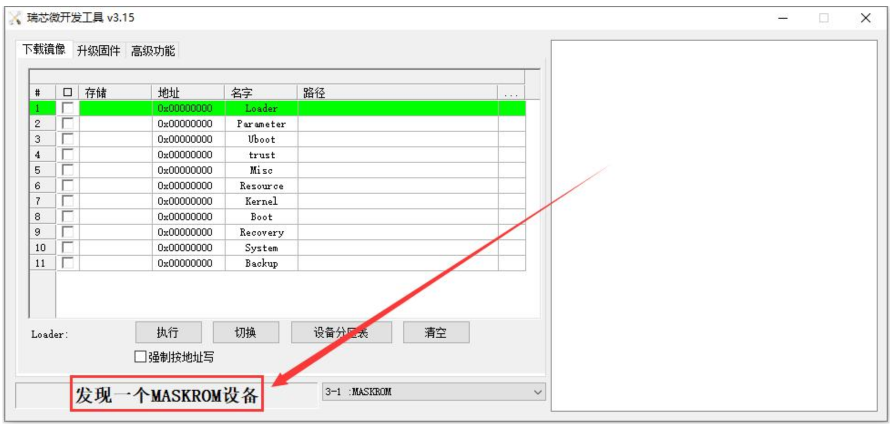
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



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



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



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

g. Then click the "**Firmware**" button to select the Android image that needs to be burned.

h. 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 the firmware will be burned into PCIE. After the burning is completed, the Android system will automatically start.

2. 14. Method of burning Android image into SPIFlash+SATA SSD

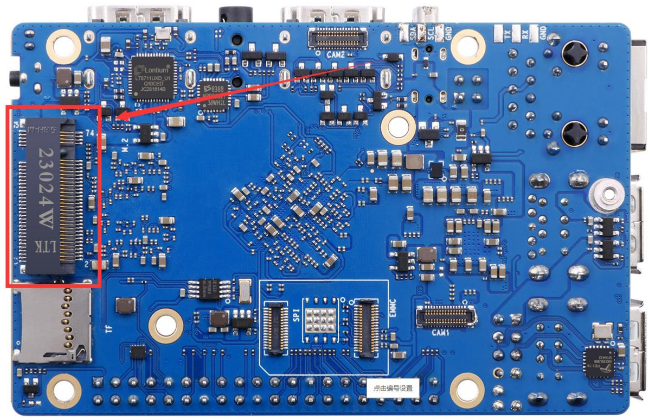
Note that all operations below are performed on a Windows computer.

Before starting to burn the image, you must ensure that the development board has been attached with the SPI Flash chip. Because the development board does not have the SPI Flash chip attached when it leaves the factory, you need to purchase and solder it on by yourself. The SPI Flash chip model we recommend is XM25QU128CWIQT08Q.

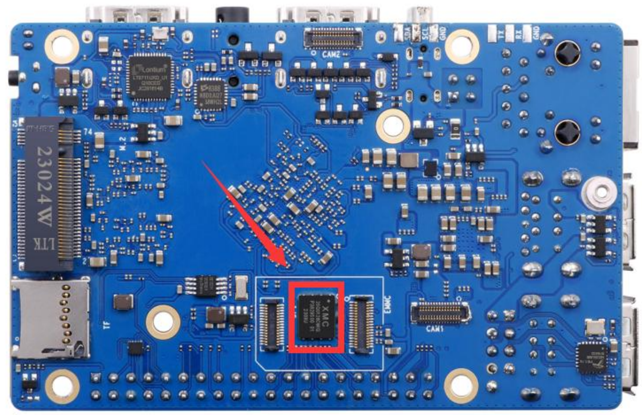
Since SPI Flash and eMMC reuse the same pins, eMMC cannot be used if SPI Flash is attached.

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

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



3) Please make sure that the development board has been attached with SPI Flash. The location of SPI Flash on the development board is as shown in the picture below. No other settings are required before starting burning.

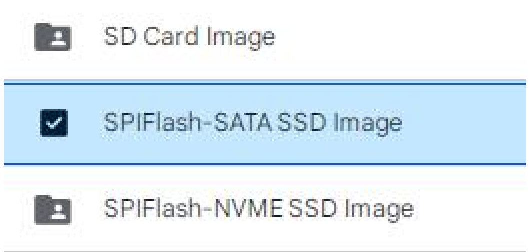


4) You also need to prepare a good quality USB2.0 male-to-male data cable

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

6) Then download the image of Android12

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.

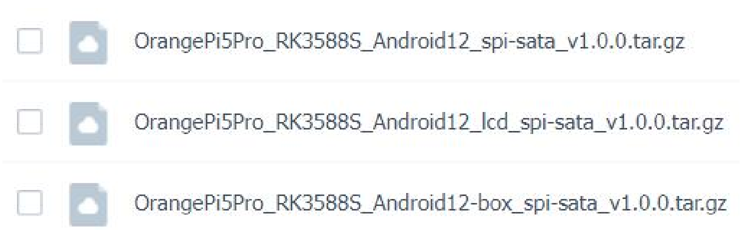


b. After entering the **SPIFlash-SATA SSD** folder, you can see the following two images. The difference between them is:

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

b) If you want to use an LCD screen, please choose the mirror with lcd

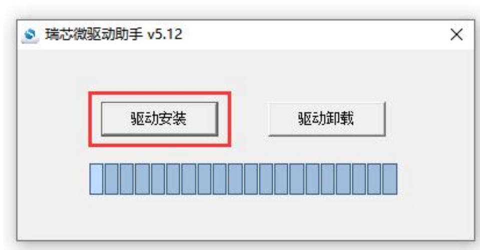
c) The image with box is an image dedicated to TV box



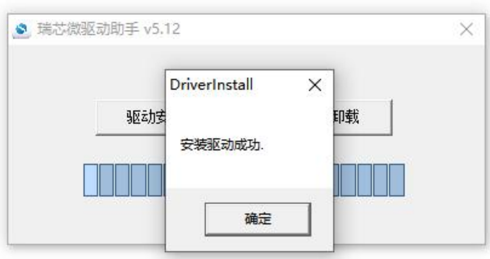
7) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

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

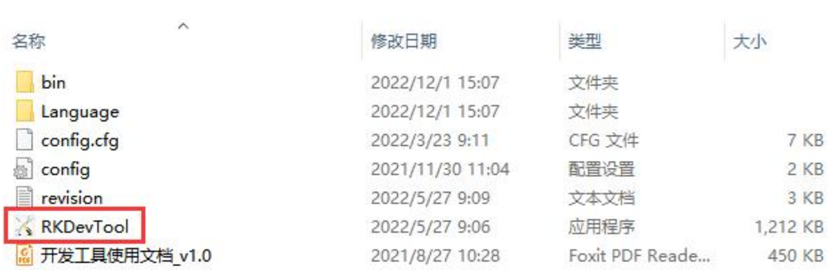
a. Click the "**Driver Installation**" button



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

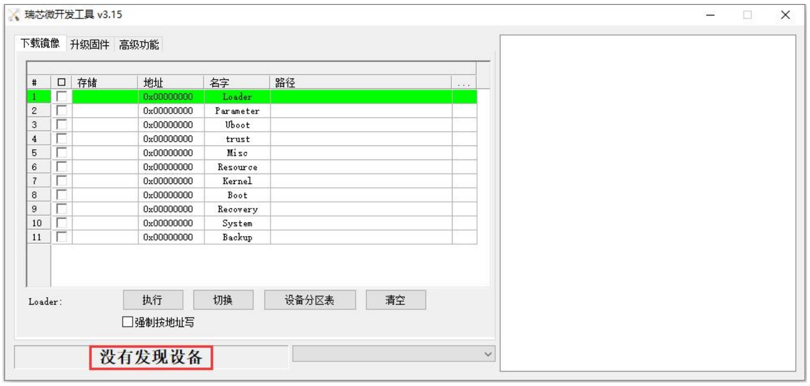


9) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool**in the unzipped folder and open it.



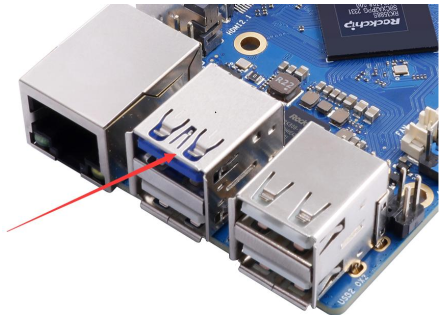
10) After opening the **RKDevTool**burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a

message "**No device found**" will be displayed in the lower left corner.



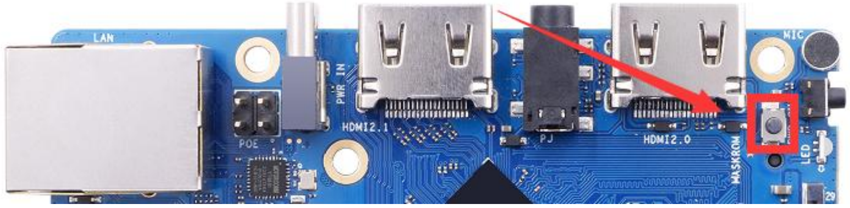
11) Then start burning the Android image to SPIFlash+SATA SSD

a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the development board's USB programming port is as shown in the figure below.

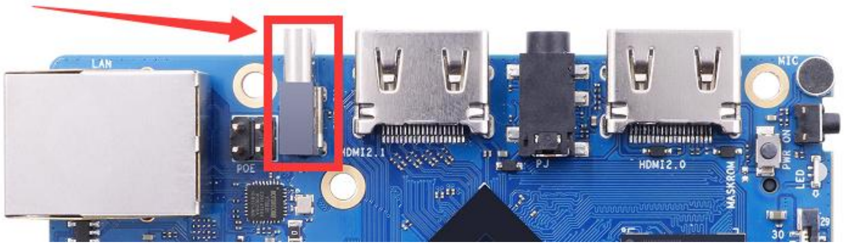


b. Ensure that the development board does not have a TF card inserted and is not connected to power.

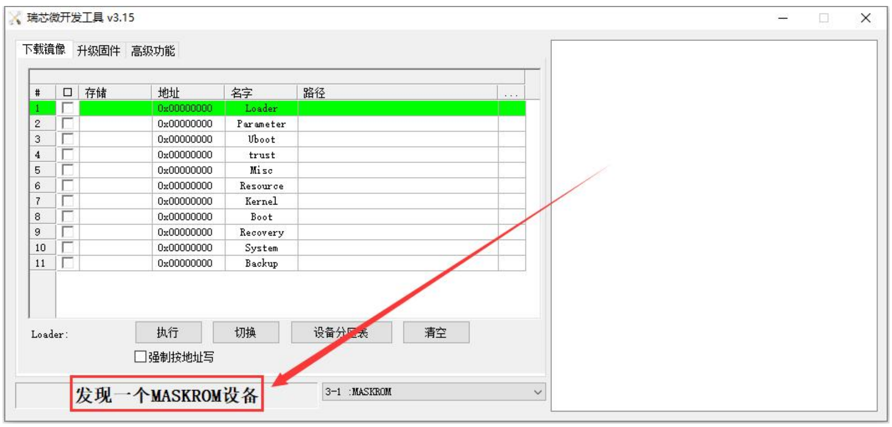
c. Then, press and hold the MaskROM button on the development board. The location of the MaskROM button on the development board is as shown in the diagram below:



d. Next, connect the development board to the Type-C power interface, and power it on. Then, you can release the MaskROM button.



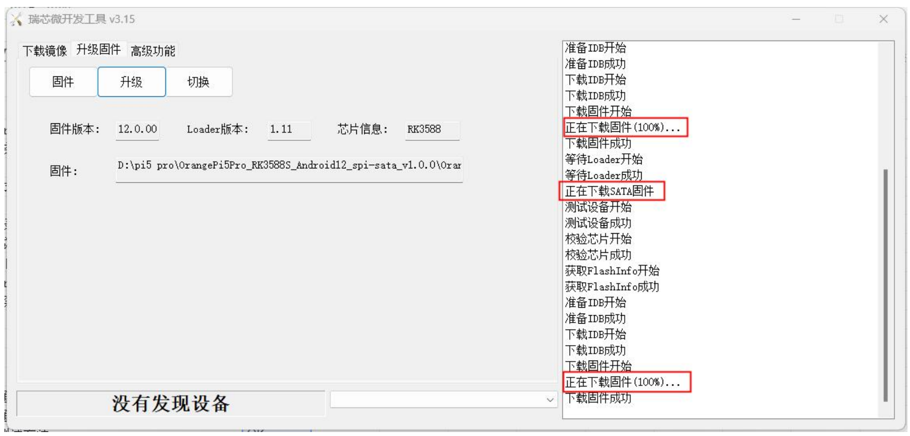
e. If the previous steps are successful, at this point, the development board will enter MASKROM mode. The burning tool's interface will prompt "Found a MASKROM device."



f. Then, click on the "**Upgrade Firmware**" option in the burning tool.

g. Next, click on the "**Firmware**" button to select the Android image that needs to be burned.

h. Finally, click on the "**Upgrade**" button to start the burning process. The burning process will proceed as shown in the diagram below. You can see that it will first burn the firmware to the SPIFlash, then to the SATA SSD. After burning is complete, the Android system will automatically start.



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

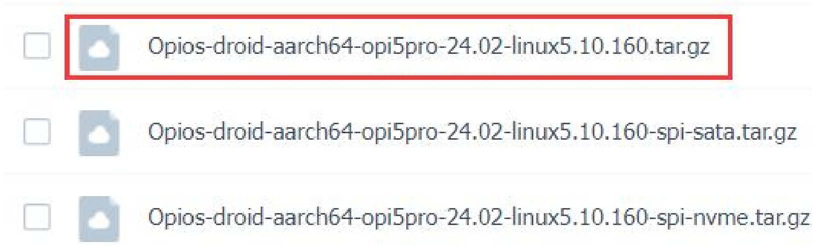
Note that all operations below are performed on a Windows computer.

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

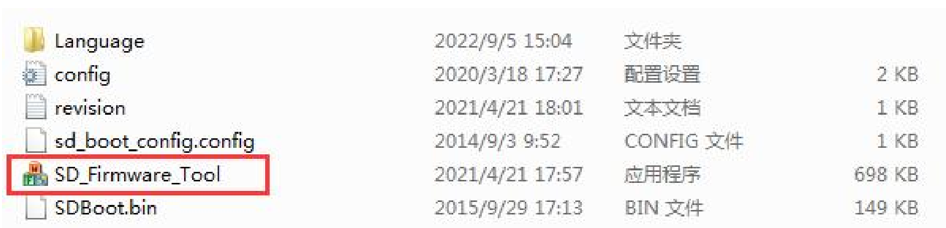
3) Then download the SDDiskTool programming tool from the **Orange Pi download page**. **Please ensure that the version of the SDDiskTool tool is the latest v1.72**

4) Then download the Orange Pi OS (Droid) image from the **Orange Pi information download page**. After opening the download link of the Orange Pi OS (Droid) image, you can see the following three types of images. Please select the image below.

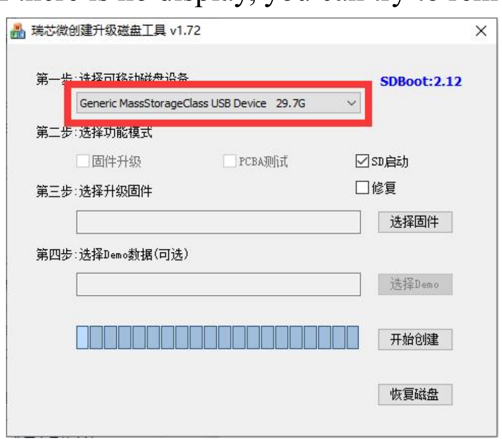


5) Then use decompression software to decompress the compressed package of the downloaded Orange Pi OS (Droid) image. In the decompressed file, the file ending with ".img" is the Orange Pi OS (Droid) image file, with a size of more than 1GB.

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.



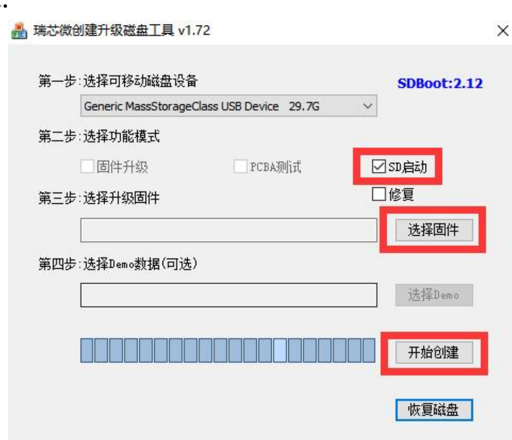
8) After confirming the drive letter, you can format the TF card first and click the **Recover Disk** button in SDDiskTool. You can also use the**SD Card Formatter** mentioned earlier to format the TF card.

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

c. First check "**SD boot**" in "**Select function mode**"

d. Then select the path to the Orange Pi OS (Droid) image in the "**Select Upgrade Firmware**" column

e. Finally, click the "**Start Creating**" 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. 16. Burn Orange Pi OS (Droid) image to SPIFlash+NVMe SSD

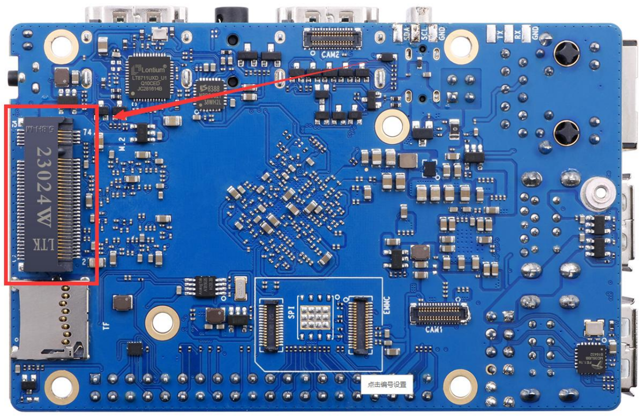
Note that all operations below are performed on a Windows computer.

Before starting to burn the image, you must ensure that the development board has been attached with the SPI Flash chip. Because the development board does not have the SPI Flash chip attached when it leaves the factory, you need to purchase and solder it on by yourself. The SPI Flash chip model we recommend is XM25QU128CWIQT08Q.

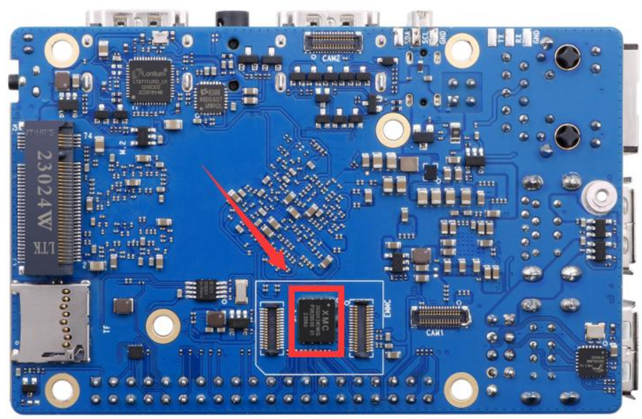
Since SPI Flash and eMMC reuse the same pins, eMMC cannot be used if SPI Flash is attached.

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

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



3) Please make sure that the development board has been attached with SPI Flash. The location of SPI Flash on the development board is as shown in the picture below. No other settings are required before starting burning.

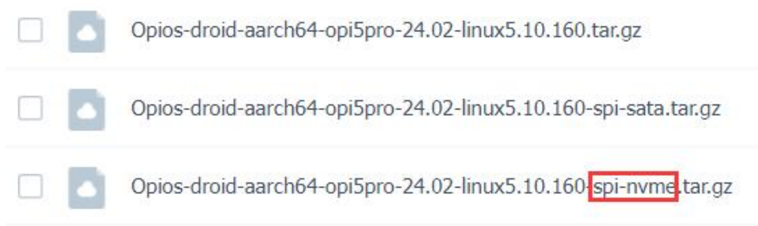


4) You also need to prepare a good quality USB2.0 male-to-male data cable

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

6) Then download the Orange Pi OS (Droid) image. After opening the download link of the Orange Pi OS (Droid) image, you can see the following three types of images. Please

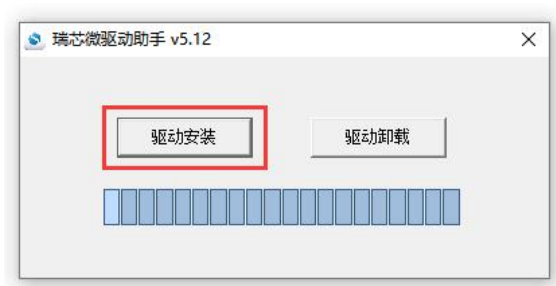
select the image with **spi-nvme** to download.



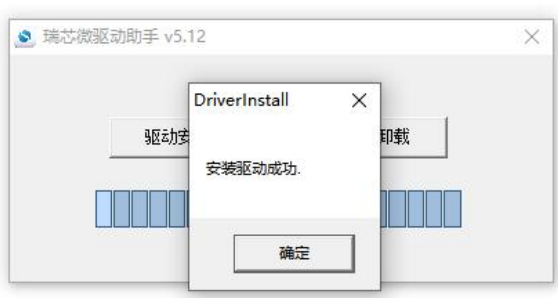
7) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

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

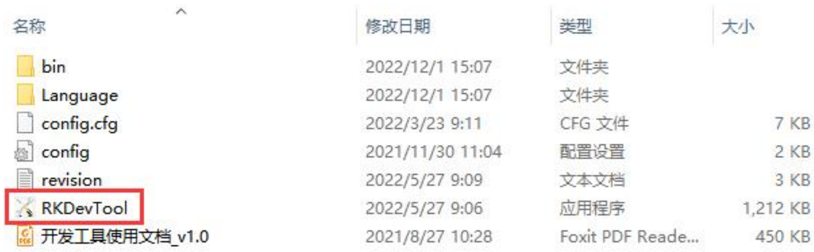
a. Click the "**Driver Installation**" button



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



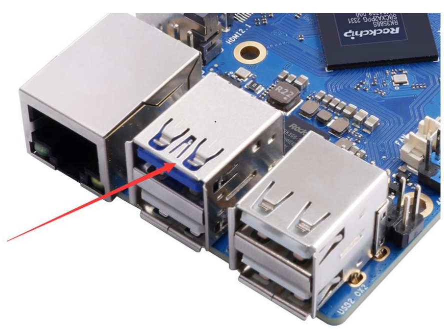
9) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.



10) After opening the **RKDevTool**burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a message "**No device found**" will be displayed in the lower left corner.

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

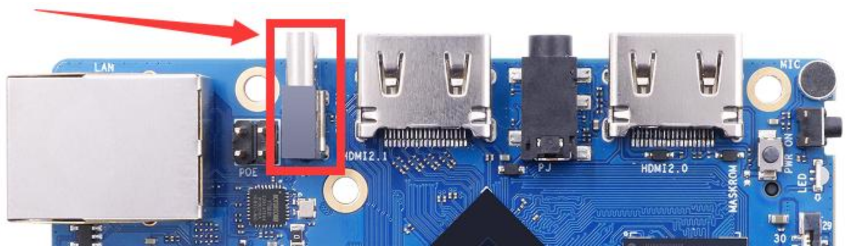
a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the USB2.0 programming port on the development board is as shown in the figure below.



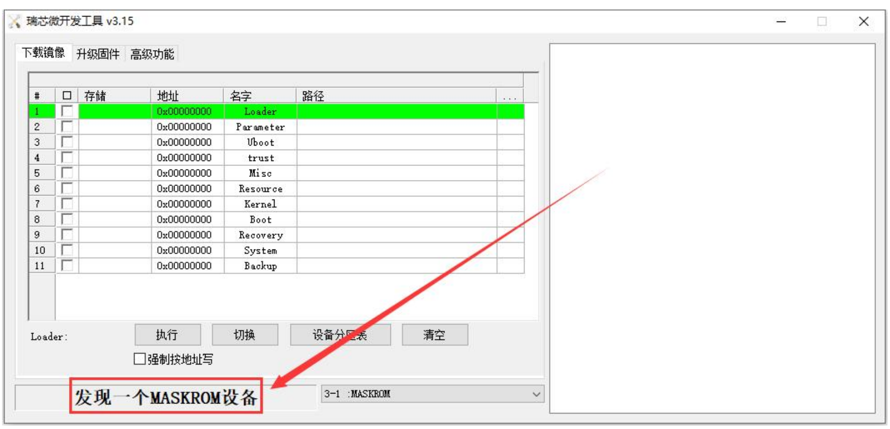
b. Make sure that the TF card is not inserted into the development board and the power supply is not connected.

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

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



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



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

g. Then click the "Firmware" button to select the Orange Pi OS (Droid) image that needs to be burned.

h. 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 the firmware will be burned into PCIE. After the burning is completed, the Orange Pi OS (Droid) system will automatically start.

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

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

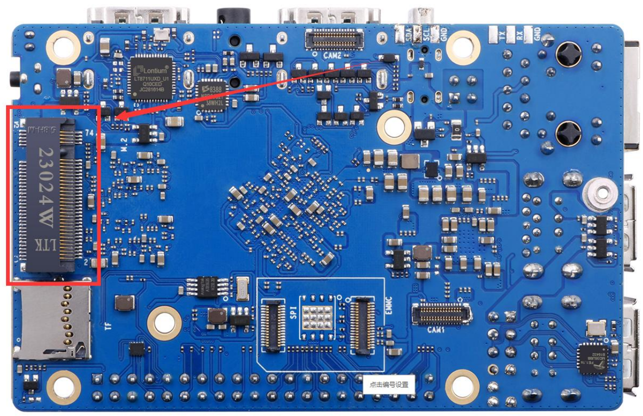
Before starting to burn the image, you must ensure that the development board has been attached with the SPI Flash chip. Because the development board does not have the SPI Flash chip attached when it leaves the factory, you need to purchase and solder it on by yourself. The SPI Flash chip model we recommend is XM25QU128CWIQT08Q.

Since SPI Flash and eMMC reuse the same pins, eMMC cannot be used if SPI

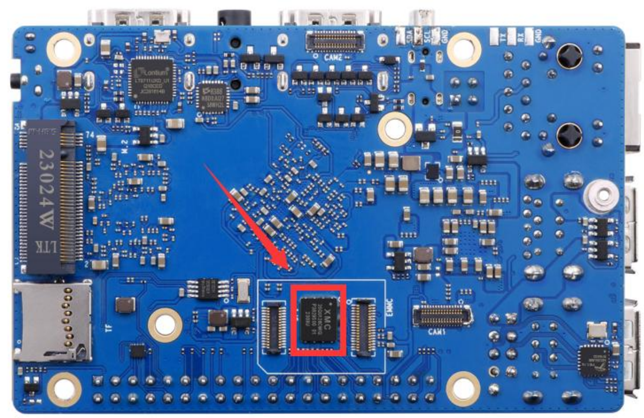
Flash is attached.

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

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



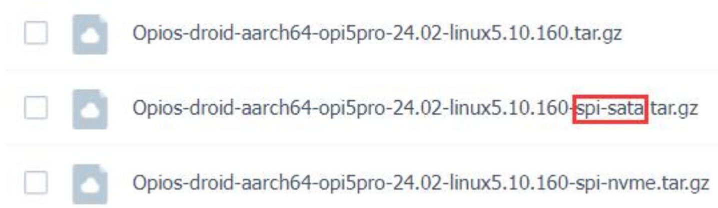
3) Please make sure that the development board has been attached with SPI Flash. The location of SPI Flash on the development board is as shown in the picture below. No other settings are required before starting burning.



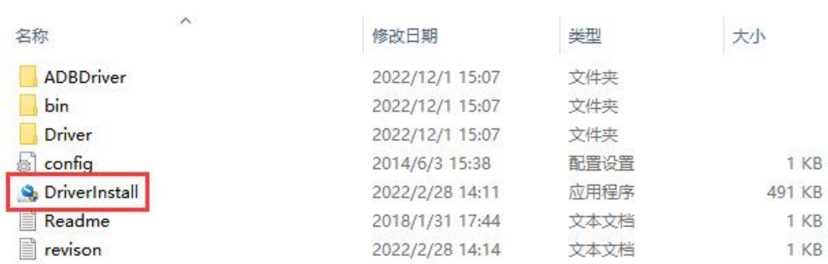
4) You also need to prepare a good quality USB2.0 male-to-male data cable

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

6) Then download the Orange Pi OS (Droid) image. After opening the download link of the Orange Pi OS (Droid) image, you can see the following three types of images. Please select the image with **spi-sata** to download.

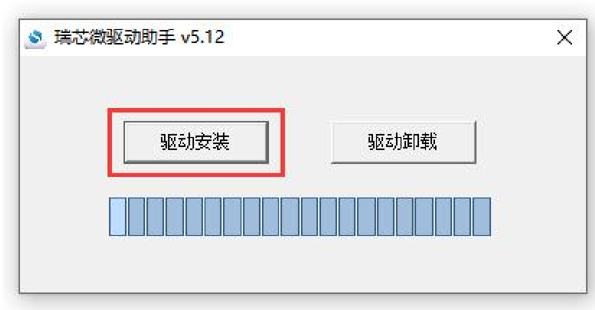


7) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

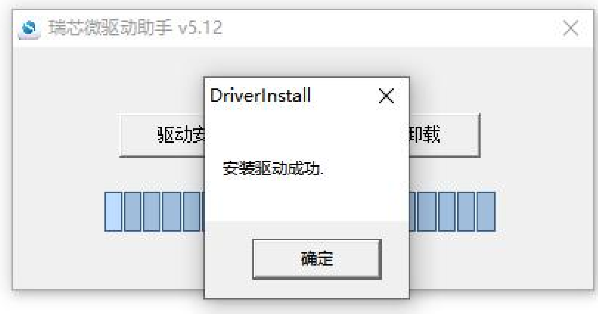


8) After opening **DriverInstall.exe**, the steps to install the Rockchip microdriver are as follows:

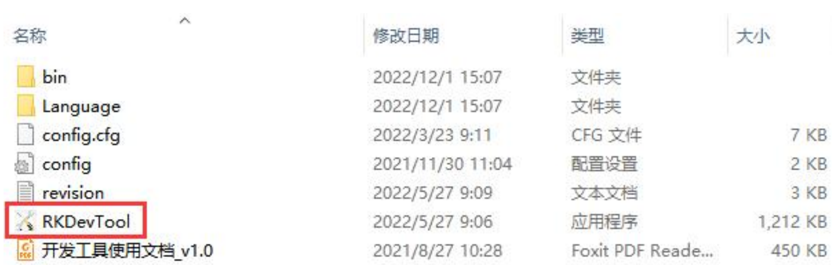
a. Click the "**Driver Installation**" button



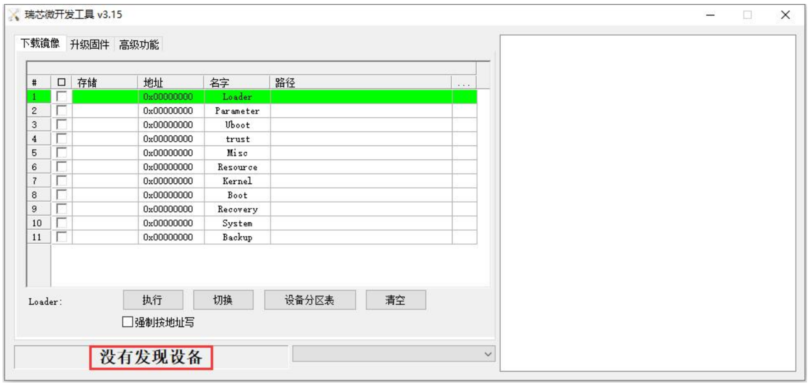
b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



9) 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.

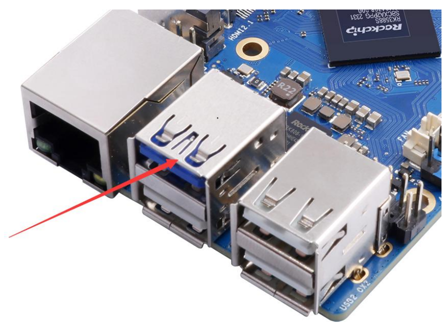


10) After opening the **RKDevTool** burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a message "**No device found**" will be displayed in the lower left corner.



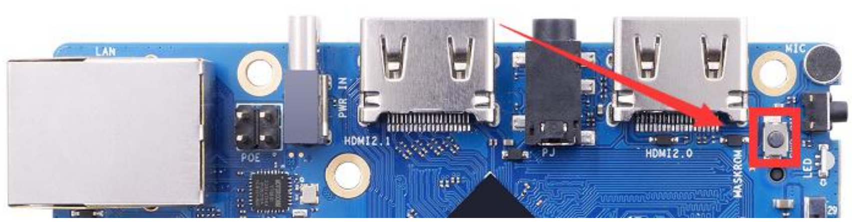
11) Then start burning the Orange Pi OS (Droid) image to SPIFlash+SATA SSD

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.

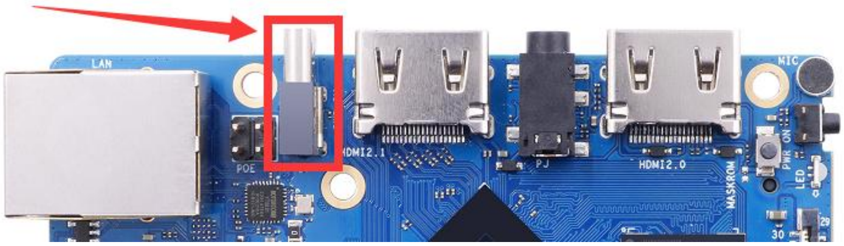


b. Make sure that the TF card is not inserted into the development board and the power supply is not connected.

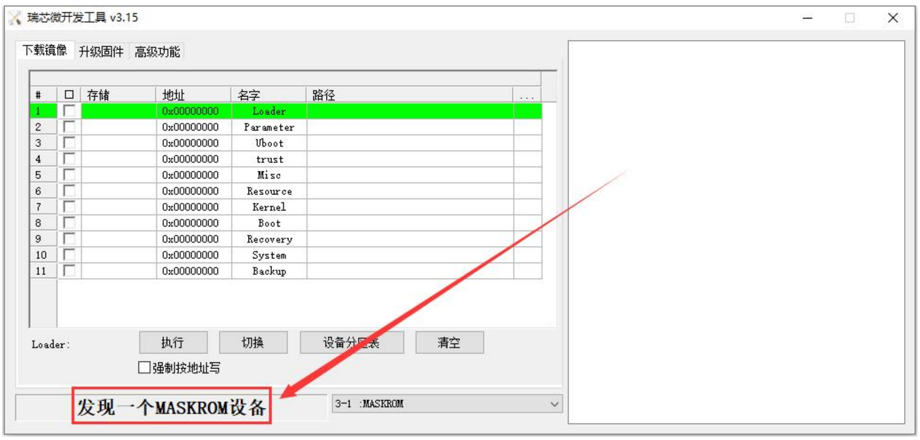
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



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



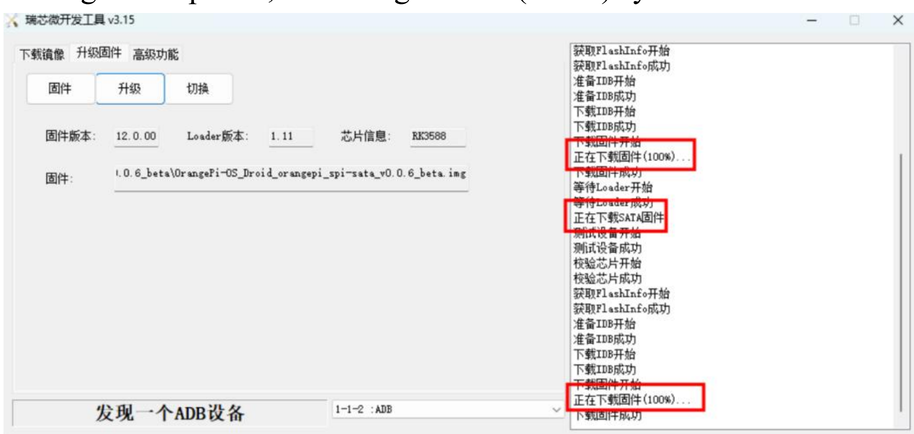
e. If the previous steps go well, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device was found**"



f. Then click the "Upgrade Firmware" column of the burning tool

g. Then click the "Firmware" button to select the Orange Pi OS (Droid) image that needs to be burned.

h. 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 into SPIFlash first, and then the firmware will be burned into SATA SSD. After the burning is completed, the Orange Pi OS (Droid) system will automatically start.

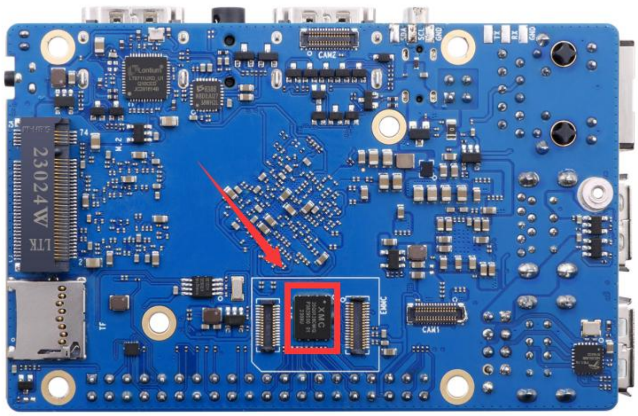


2. 18. How to clear SPIFlash using RKDevTool

The development board does not have an SPI Flash chip attached when it leaves the factory, so you need to attach it yourself. The SPI Flash chip model we recommend is XM25QU128CWIQT08Q.

Since SPI Flash and eMMC reuse the same pins, eMMC cannot be used if SPI Flash is attached.

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

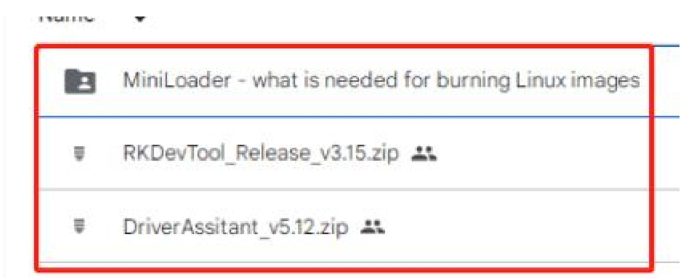


2) First you need to prepare a good quality USB2.0 male-to-male data cable

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 download page

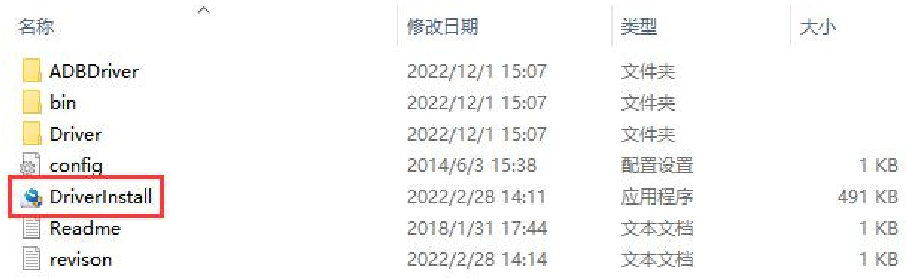
a. On the Orange Pi data download page, first select the official tool, and then enter the folder below

b. Then download all the files below



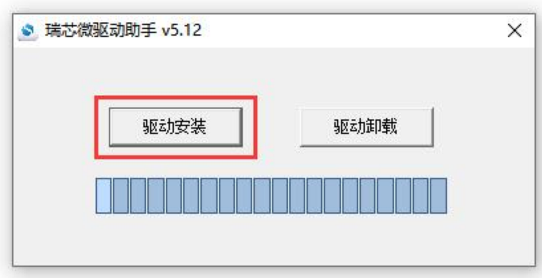
Note that the "MiniLoader-things needed to burn Linux images" folder will be referred to as the MiniLoader folder below.

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.

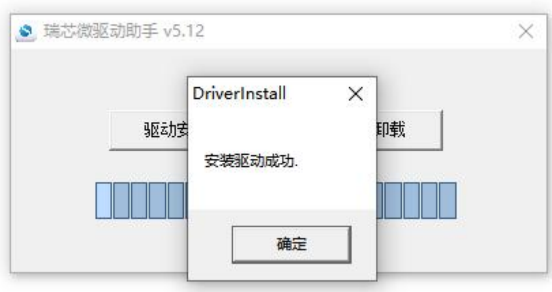


5) Open **DriverInstall.exe**and install the Rockchip microdriver as follows:

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.

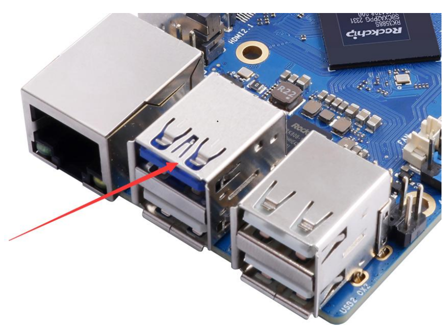


6) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

7) After opening the **RKDevTool**burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a message "No device found" will appear in the lower left corner.

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

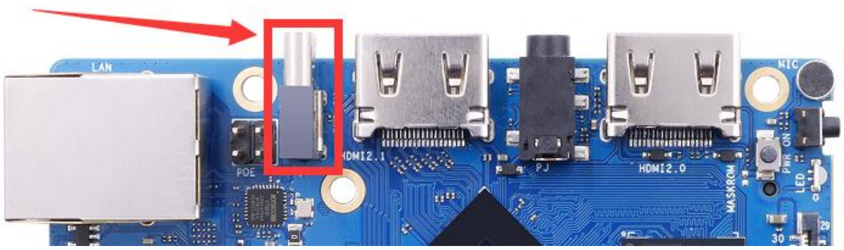
a. irst, connect the development board and Windows computer through the Type-C data cable. The location of the Type-C interface of the development board is as shown in the figure below.



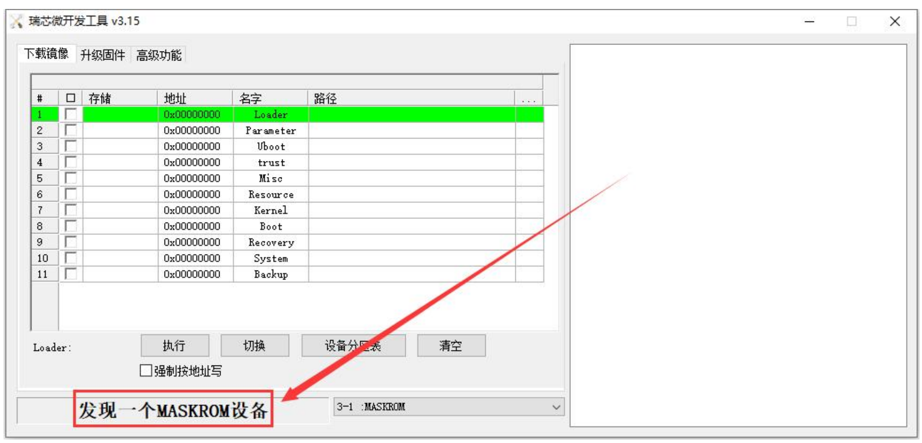
b. Make sure that the TF card is not inserted into the development board and the power supply is not connected.

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

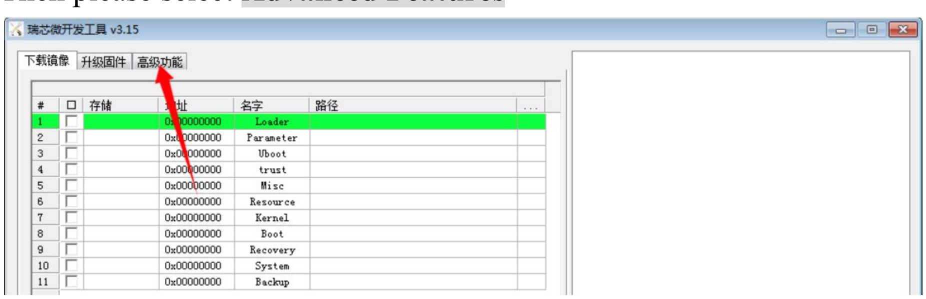
d. Then connect the development board to the power supply of the Type-C interface, power it on, and then release the MaskROM button.



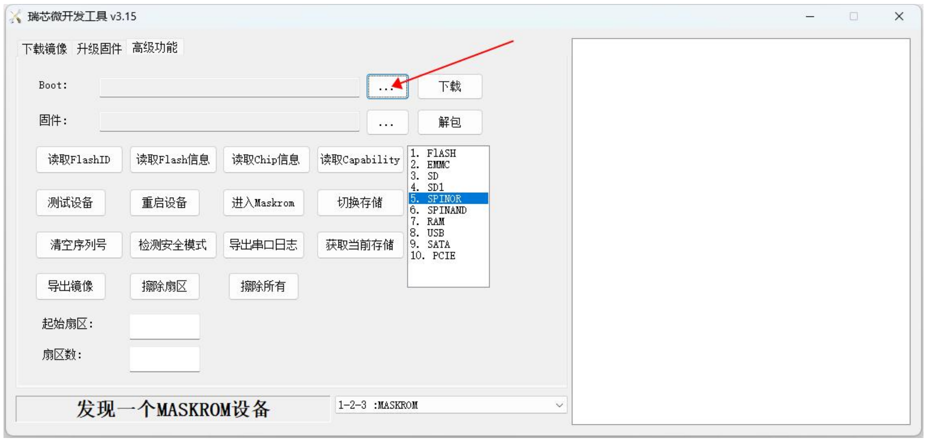
e. If the previous steps go well, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device was found**"



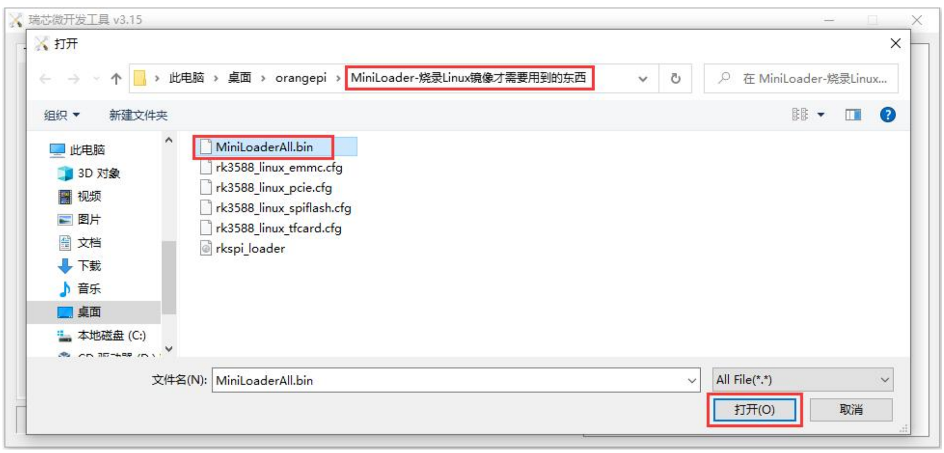
f. Then please select **Advanced Features**



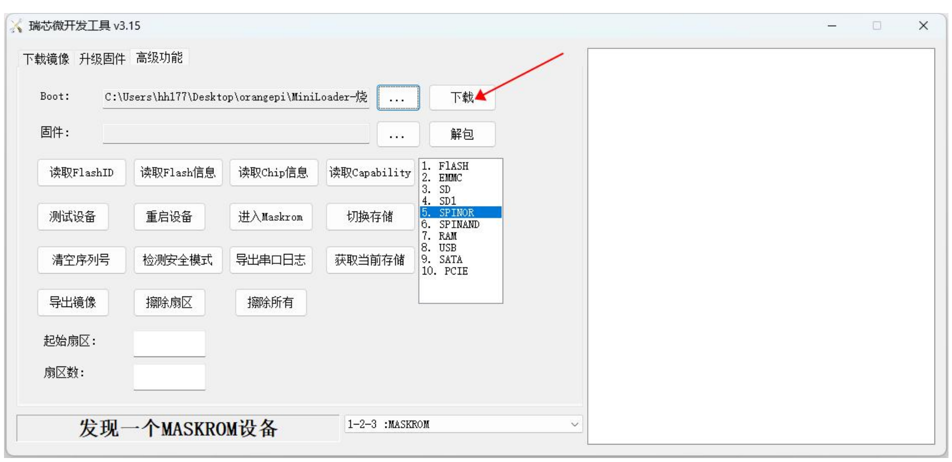
g. Then click the location shown in the picture below



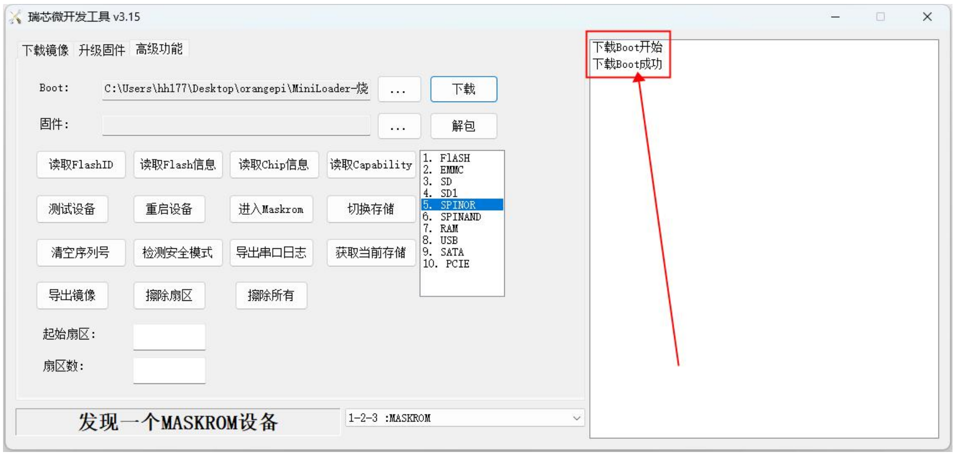
h. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click Open



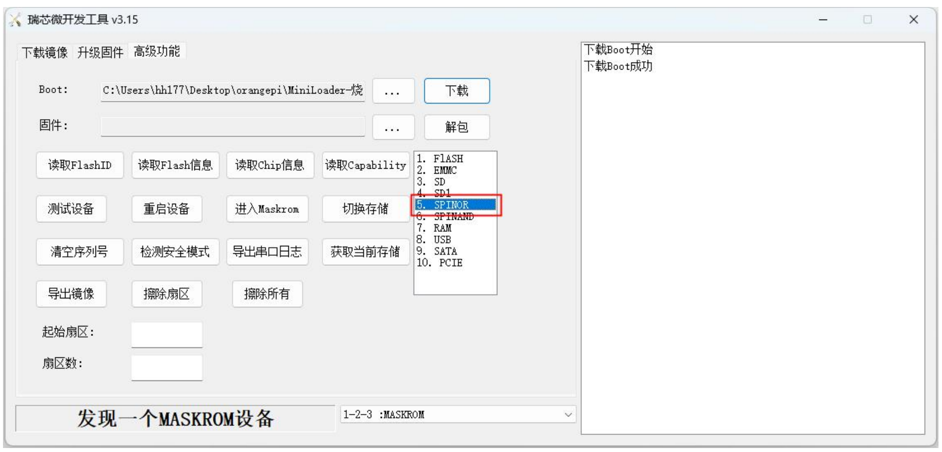
i. Then click Download



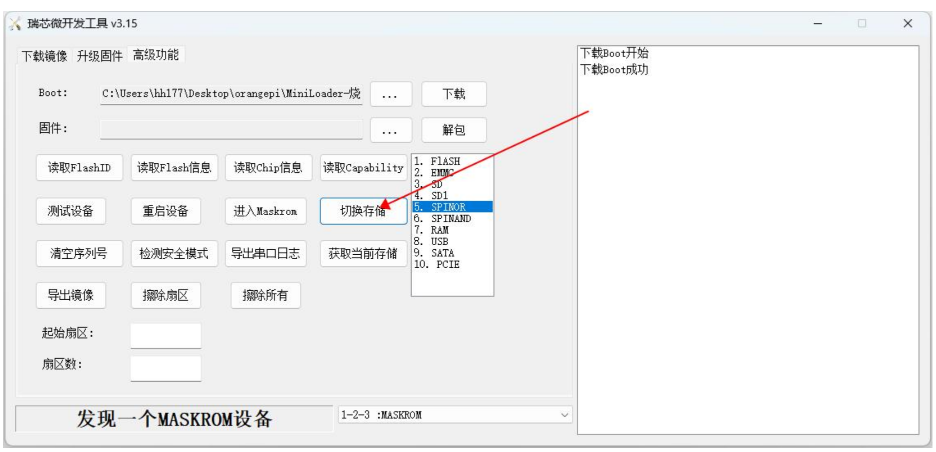
j. After downloading **MiniLoaderAll.bin** the display is as shown below



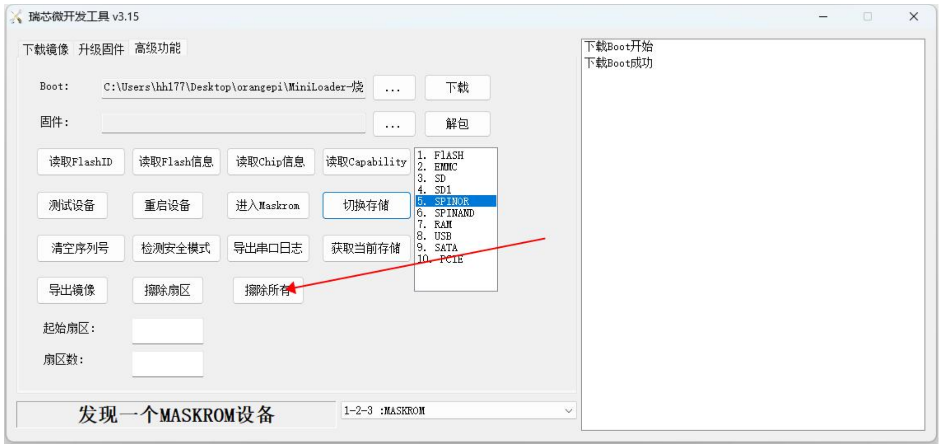
k. Then select the storage device as **SPINOR**



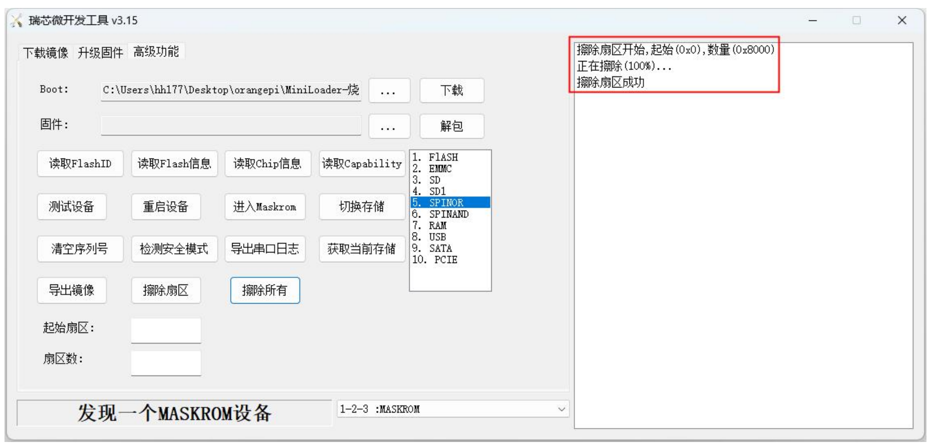
l. Then click **Switch Storage**



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



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



2. 19. 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 burned, there is no need to insert the 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 you can connect the development board to a TV or HDMI monitor through an HDMI-to-HDMI cable. If you purchase an LCD screen, you can also use the LCD screen to display the system interface of the development board.

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

4) The development board has an Ethernet port, which can be used to connect to the Internet by plugging in a network cable.

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

Please remember not to use a power adapter with an output voltage greater than 5V, as it can damage the development board.

Many unstable phenomena during the system's power-on startup process are mainly caused by power supply issues, so a reliable power adapter is crucial. If you encounter continuous rebooting during the startup process, please try changing the power supply or the Type-C data cable.

Please note that the Type-C power interface does not support PD negotiation.

Also, please do not use a USB port on a computer to power the development board.

6) Then turn on the power adapter. If everything is normal, you can see the system startup screen 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 port cable to connect the development board to the computer. For the serial port connection method, please refer to **the section on how to use the debugging serial port.**

2. 20. How to use the debugging serial port

2. 20. 1. Debug serial port connection instructions

1) First, you need to prepare a 3.3V USB to TTL module, and then insert one end of the USB interface 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. Please do not use CP2102 and PL2303 type USB to TTL modules.

Before purchasing a USB to TTL module, please confirm that the module supports a baud rate of 1,500,000.

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

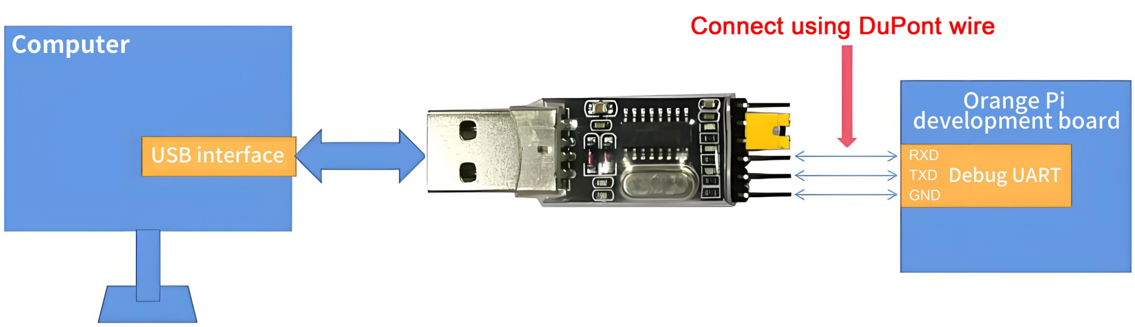
3) The GND, TXD and RXD pins of the USB to TTL module need to be connected to the debugging serial port of the development board through Dupont lines.

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

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

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

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



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

The TX and RX pins of the serial port need to be cross-connected. If you prefer

not to carefully distinguish the order of TX and RX, you can initially connect the TX and RX of the serial port randomly. If there is no output during testing, then swap the TX and RX pins' order. This way, there will always be one correct order.

2. 20. 2. How to use the debugging serial port on 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 interface of the Ubuntu computer. If the USB to TTL module is connected and recognized normally, you can see the corresponding device node name under **/dev** of the Ubuntu PC. Remember this node name and set the serial port later. software will be used

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

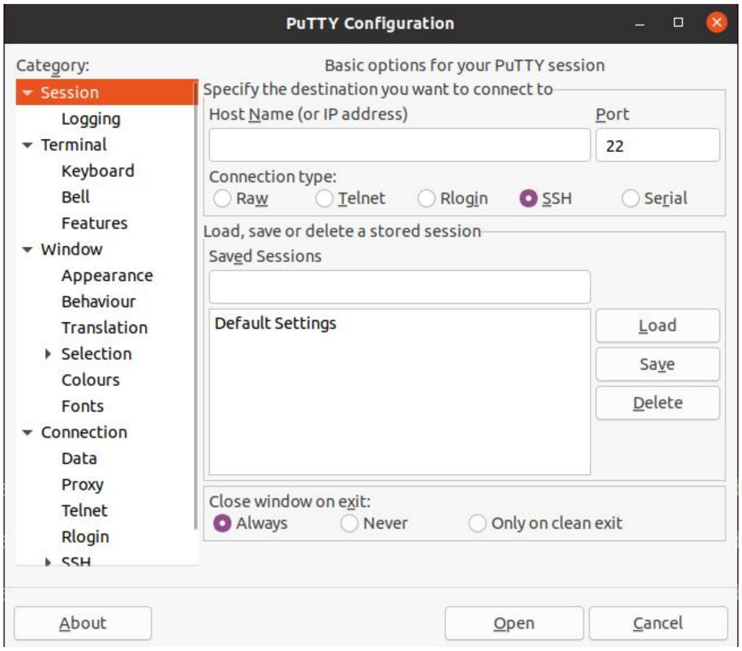
2) Then use the following command to install putty on Ubuntu PC

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

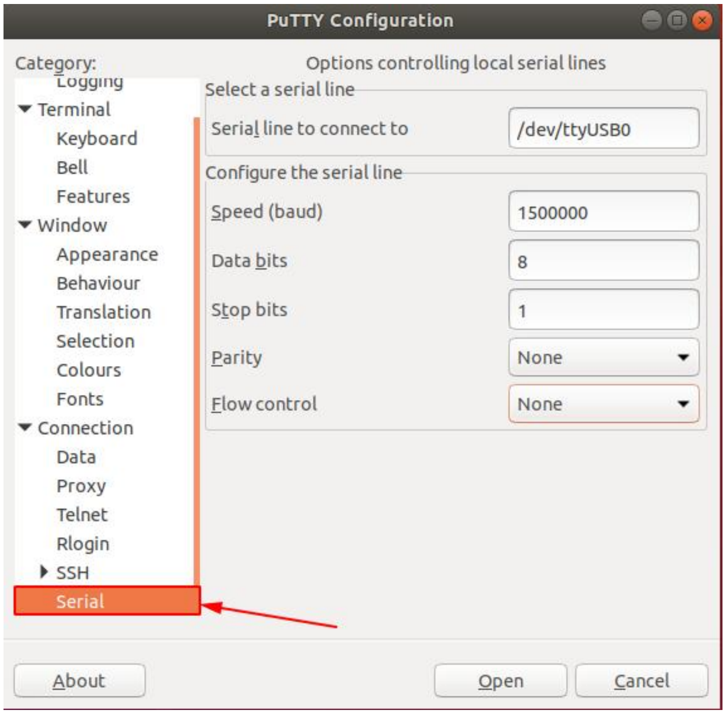
3) Then run putty, remember to **add sudo permissions**

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

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



5) First select the serial port setting interface

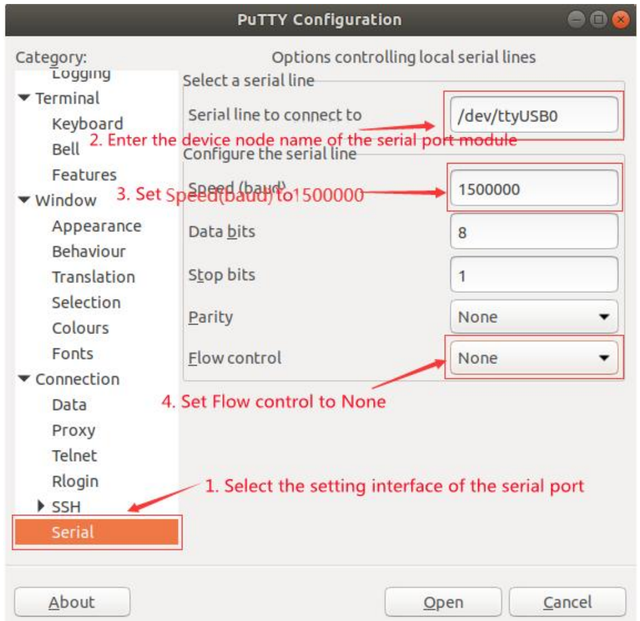


6) Then set the parameters of the serial port

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

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

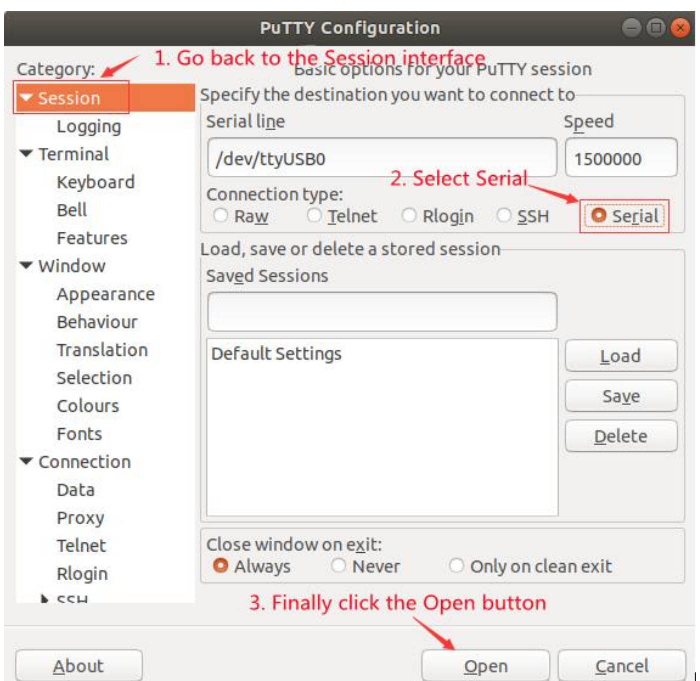
c. Set Flow control to None



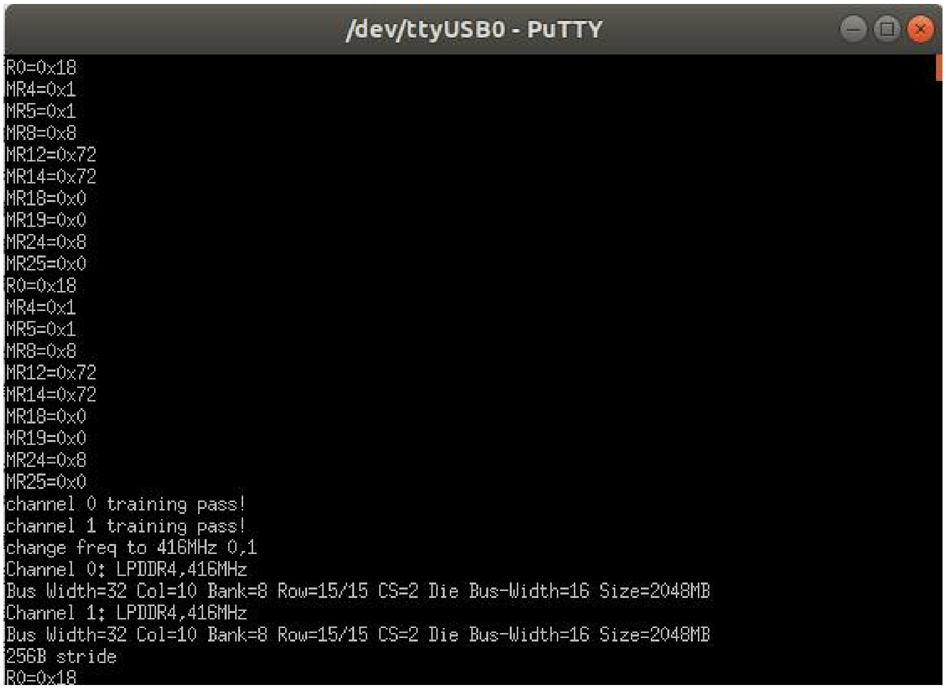
7) After completing the settings in the serial port setting interface, return to the Session interface.

a. Serial First select 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 open serial terminal.



2. 20. 3. How to use the debugging serial port on Windows platform

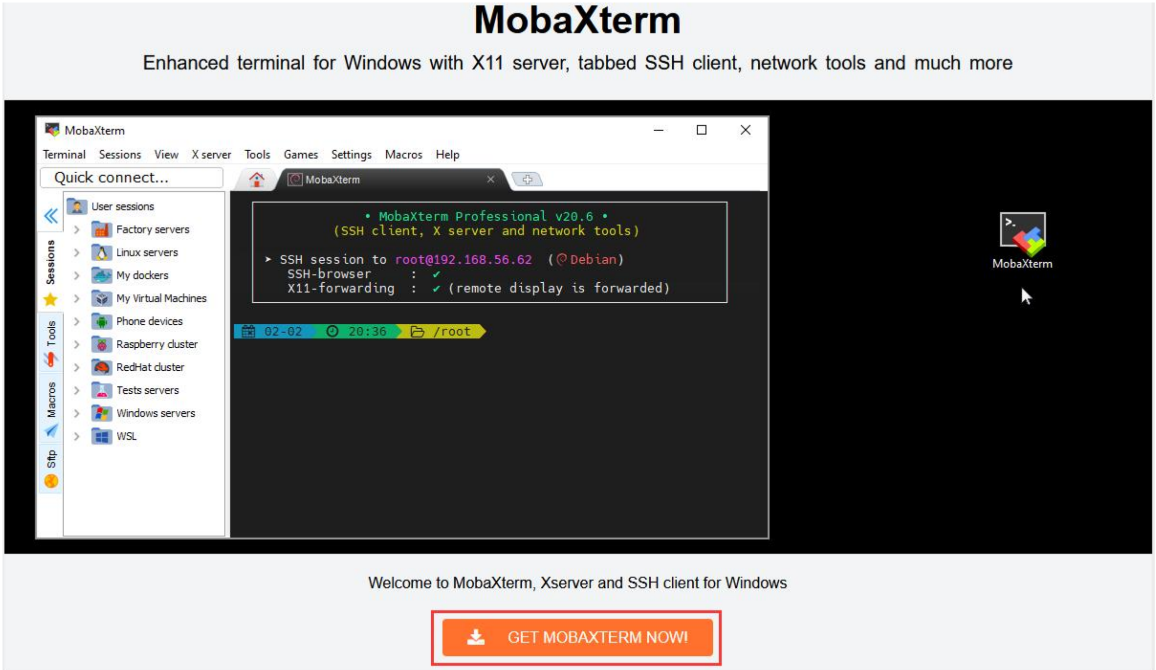
There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without purchasing a serial number.

1) Download MobaXterm

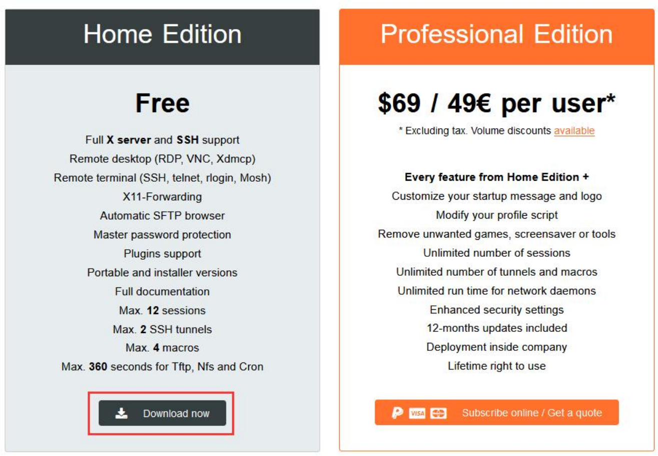
a. The download URL of MobaXterm is as follows

https://mobaxterm.mobatek.net

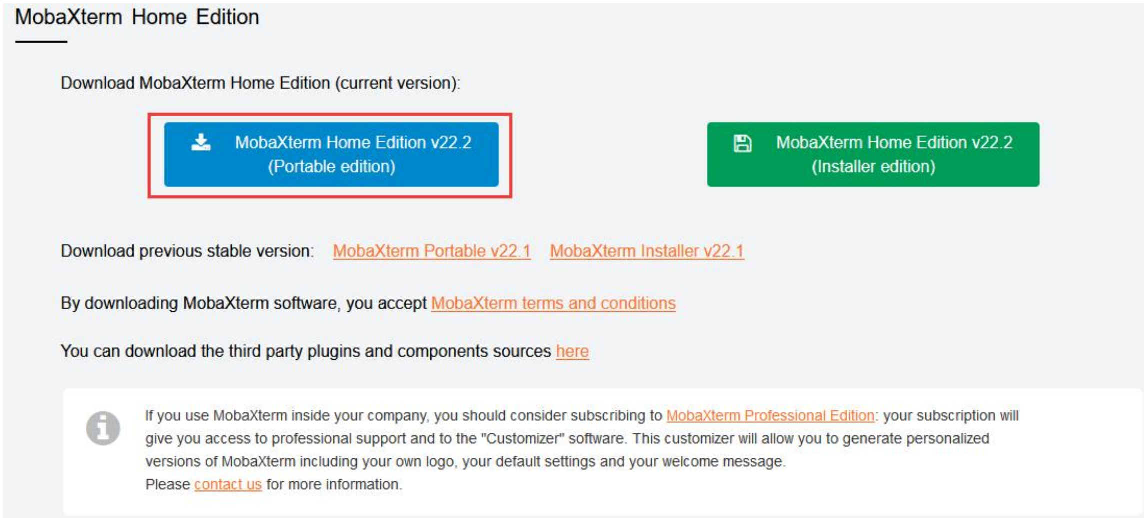
b. Enter the MobaXterm download webpage and click **GET XOBATERM NOW!**



c. Then choose to download the Home version



d. Then select the Portable version. There is no need to install it after downloading. You can open it directly and use it.



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

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

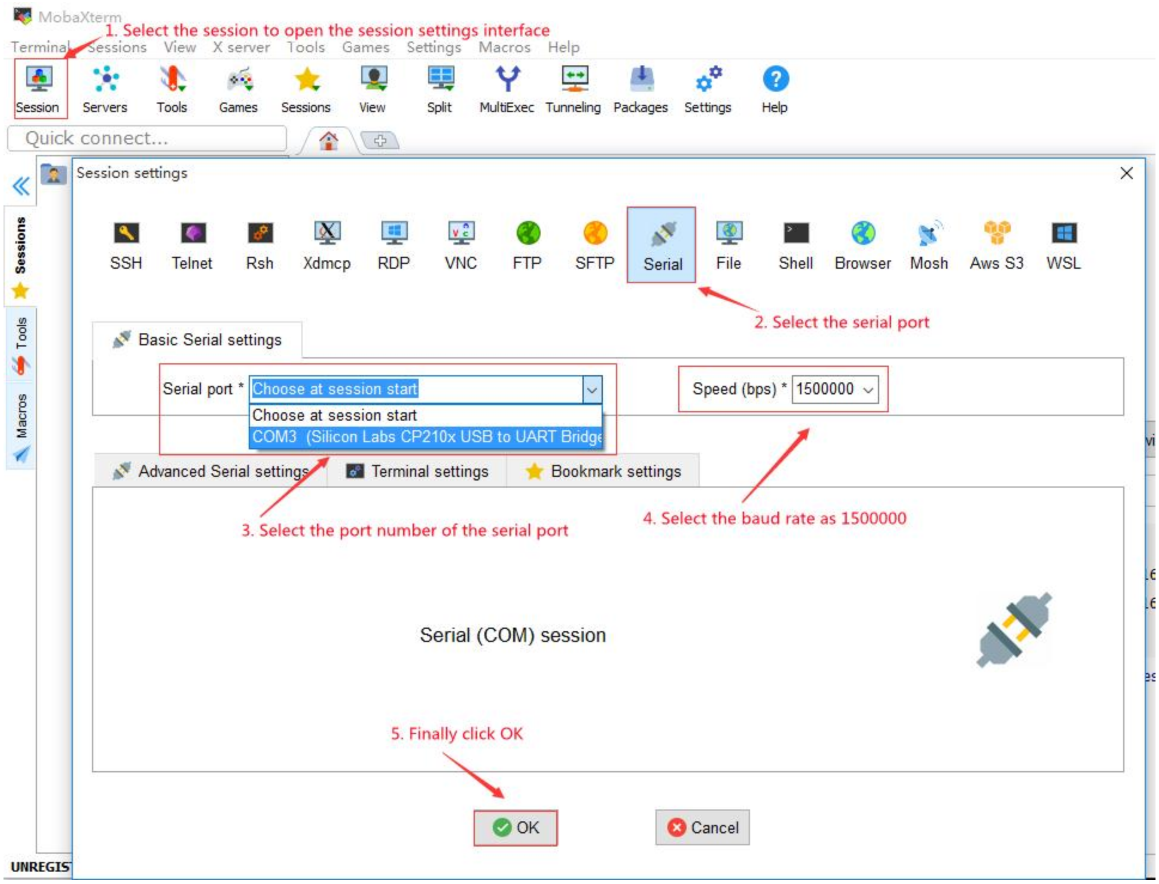
a. Open the settings interface for the session.

b. Select the type of serial port.

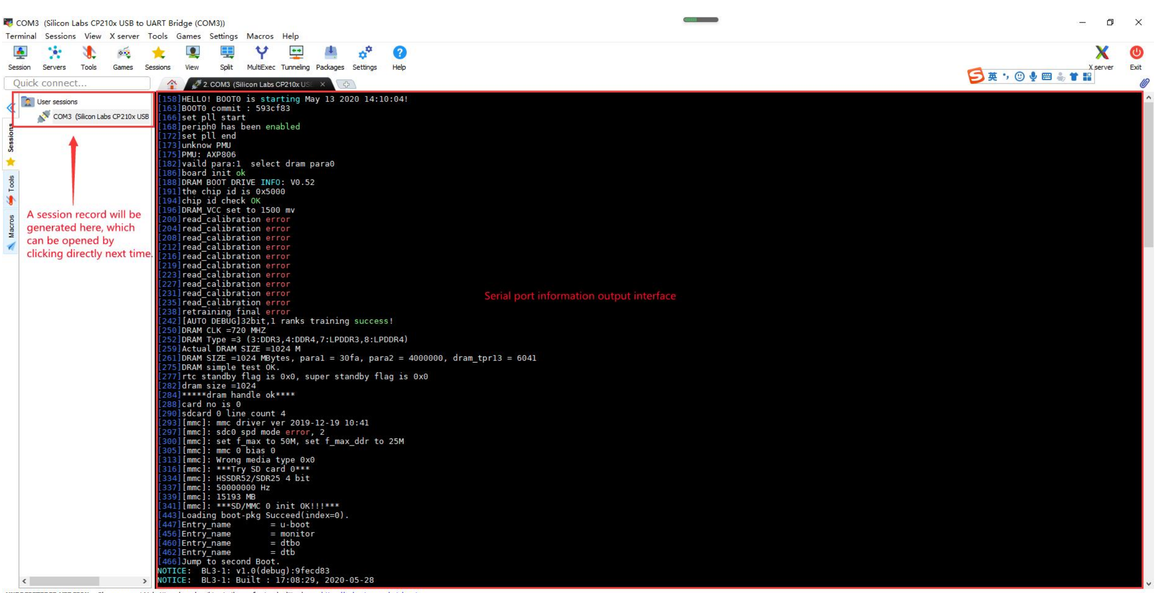
c. Choose the port number for the serial port (select the appropriate port number based on the actual situation). If you cannot see the port number, please use a driver scanning tool like "360 Driver Master" to scan and install the driver for the USB to TTL serial port chip.

d. Set the baud rate of the serial port to 1500000.

e. Finally, click the "OK" button to complete the settings.



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.



2.21. Instructions for using the 5v pin in the 40pin interface of the development board for power supply

The recommended power supply method for our development board is to use a 5V/4A Type-C power cable plugged into the Type-C power interface of the development board. If you need to power the development board using the 5V pin in the 40-pin interface, please ensure that the power cable and adapter you are using can meet the power supply requirements of the development board. If there are any stability issues, please switch back to using the Type-C power supply.

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

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

2) To power the development board using the 5V pin in the 40-pin interface, follow these steps for connecting the power cable:

a. The USB A end of the power cable, as shown in the diagram, needs to be plugged into a 5V/4A power adapter (do not plug it into a USB prt on a computer).

b. The red DuPont wire needs to be inserted into the 5V pin of the development board's 40-pin interface.

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

d. The positions of the 5V and GND pins in the 40-pin interface on the development board are as shown in the diagram. Be sure not to connect them incorrectly.

3. Ubuntu/Debian Server and Xfce Desktop System Instructions

The content of this chapter is based on the Linux server version image and the xfce desktop version image.

If you are using the OPi OS Arch image, please check the Orange Pi OS Arch System Instructions chapter.

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
Debian 12 - Bookworm	Linux5.10	support	support
Ubuntu 20.04 - Focal	Linux5.10	support	support
Ubuntu 22.04 - Jammy	Linux5.10	support	support

3. 2. Linux system adaptation situation

Function	Debian11	Debian12	Ubuntu20.04	Ubuntu22.04
HDMI 2.1 Video	OK	OK	OK	OK
HDMI 2.1 Audio	OK	OK	OK	OK
HDMI 2.0 Video	OK	OK	OK	OK
HDMI 2.0 Audio	OK	OK	OK	OK

USB2.0x3	**OK**	**OK**	**OK**	**OK**
USB3.0x1	**OK**	**OK**	**OK**	**OK**
Gigabit Network	**OK**	**OK**	**OK**	**OK**
Network port status light	**OK**	**OK**	**OK**	**OK**
WIFI	**OK**	**OK**	**OK**	**OK**
Bluetooth	**OK**	**OK**	**OK**	**OK**
Debug serial port	**OK**	**OK**	**OK**	**OK**
RTC Chip	**OK**	**OK**	**OK**	**OK**
FAN Interface	**OK**	**OK**	**OK**	**OK**
EMMC Slot	**OK**	**OK**	**OK**	**OK**
GPIO（40pin）	**OK**	**OK**	**OK**	**OK**
UART（40pin）	**OK**	**OK**	**OK**	**OK**
SPI（40pin）	**OK**	**OK**	**OK**	**OK**
I2C（40pin）	**OK**	**OK**	**OK**	**OK**
CAN（40pin）	**OK**	**OK**	**OK**	**OK**
OV13850 Carema	**OK**	**OK**	**OK**	**OK**
OV13855 Carema	**OK**	**OK**	**OK**	**OK**
SPI+NVME Start	**OK**	**OK**	**OK**	**OK**
SPI+SATA Start	**OK**	**OK**	**OK**	**OK**
LCD	**OK**	**OK**	**OK**	**OK**
MIC	**OK**	**OK**	**OK**	**OK**
Headphone playback	**OK**	**OK**	**OK**	**OK**
Headphone recording	**OK**	**OK**	**OK**	**OK**
Tri-color LED light	**OK**	**OK**	**OK**	**OK**
GPU	**OK**	**OK**	**OK**	**OK**
NPU	**OK**	**OK**	**OK**	**OK**
VPU	**OK**	**OK**	**OK**	**OK**
On/off button	**OK**	**OK**	**OK**	**OK**
watchdog test	**OK**	**OK**	**OK**	**OK**
Chromium Hard solution video	**OK**	**OK**	**OK**	**OK**

3. 3. Linux command format description in this manual

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

```

```

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

```

```

2) Description of the prompt type before the command

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

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

b. **root@orangepi:~$** The prompt indicates that this command is entered in the Linux system of the development board. The **$** at the end of the prompt indicates that the current user of the system is 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**. 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 to execute.

d. **test@test:~$** The prompt indicates that the command was entered in the Ubuntu PC or Ubuntu virtual machine, not the Linux system of the development board. The **$** at the end of the prompt indicates that the current user of the system is an ordinary user. When executing a privileged command, you need to add **sudo**

e. **root@test:~#** The prompt indicates that the command was entered in the Ubuntu PC or Ubuntu virtual machine, not 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 to execute.

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

a. As shown below, the bold black part is the command that needs to be entered, and the content below the command is the output content (some commands have output, some may not 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 that cannot be written in one line will be placed on the next line. As long as the black and bold parts are the commands that need to be entered. When these commands are entered into one line, the "\" at the end of each line needs to be removed. This is not part of the command. In addition, different parts of the command have spaces, please don't miss them.

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

3. 4. Linux system login instructions

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

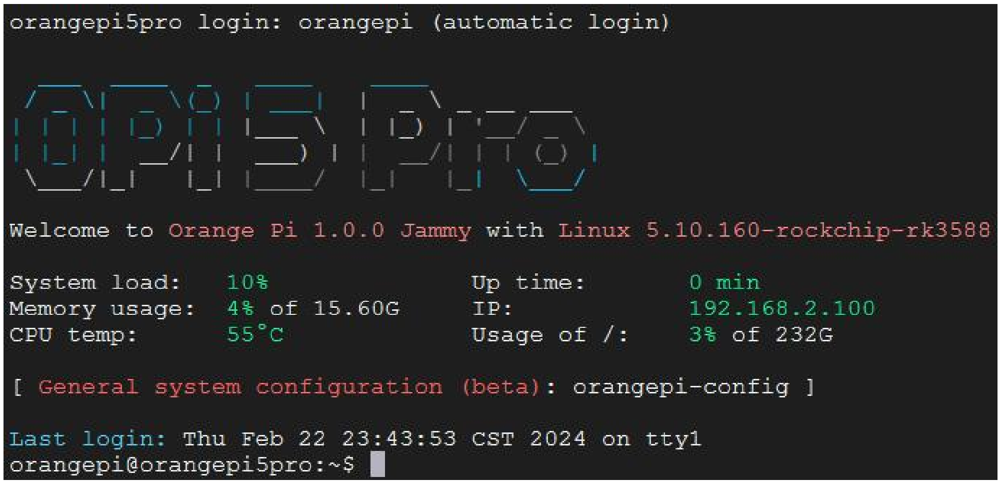
Account	Password
root	**orangepi**
orangepi	**orangepi**

Please note that when entering the password, the specific content of the password will not be displayed on the screen. Do not mistake this for a malfunction. Simply input the password and press Enter directly.

When encountering password-related errors or issues with SSH connections, please note that as long as you are using the Linux image provided by Orange Pi, do not doubt the correctness of the password. Instead, explore other potential reasons for the error or connection problem.

3. 4. 2. How to set up automatic login of Linux system terminal

1) The Linux system automatically logs in to the terminal by default. 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 to the terminal

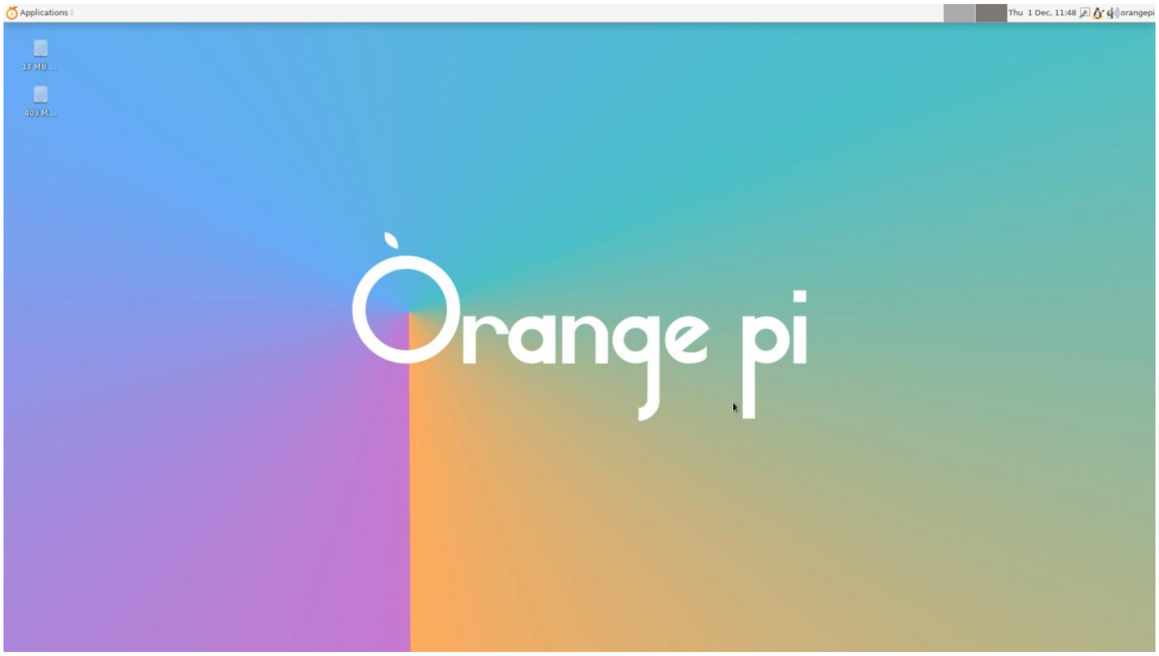
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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. Linux desktop version system automatic login instructions

1) The desktop version system will automatically log in to the desktop after startup, without entering a password.



2) Run the following command to prevent 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 this time, you need to enter a password to enter the system.

3. 4. 4. Setting method for automatic login of root user in Linux desktop system

1) Execute the following command to set the desktop version of the system to automatically log in using the root user

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

2) Then restart the system and you will automatically log in to the desktop as the root user.

> **Please note that if you log in as the root user on the desktop system, you will be unable to use the pulseaudio tool in the top right corner to manage audio devices.**
>
> **Additionally, please be aware that this is not a bug, as pulseaudio inherently does not allow running under the root user.**

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

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

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

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

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

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

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

3) The steps to reopen the desktop are as follows

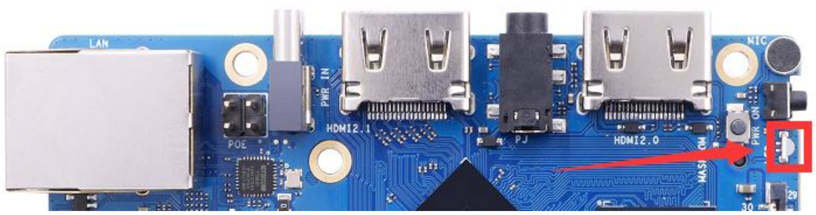
a. 首 First enter the following command on the command line, **please remember to add sudo permissions**

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

b. After selecting, the monitor will display the desktop.

3. 5. Onboard LED light test instructions

1) There is a red, green and blue light on the development board, and its location is as shown in the picture below:



2) As long as the development board is powered on, the red LED light will remain constantly illuminated. This behavior is hardware-controlled and cannot be turned off by software. The constant illumination of the red LED light serves as an indicator that the development board's power supply is functioning properly.

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

4) The method of setting the green light to turn on and off and flash is as follows

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

a. First enter the settings directory of the green light

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

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

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

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

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

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

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

5) The method of using commands to set the blue 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 settings directory of the blue light

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

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

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

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

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

d. The command to set the blue light flashing is as follows:

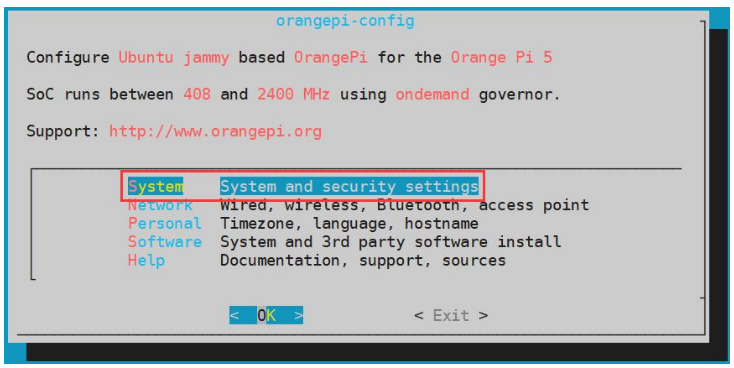
```
root@orangepi:/sys/class/leds/blue_led# echo heartbeat > trigger
```

6) If you don't need the LED light to flash after powering on, you can use the following method to turn off the green and blue lights.

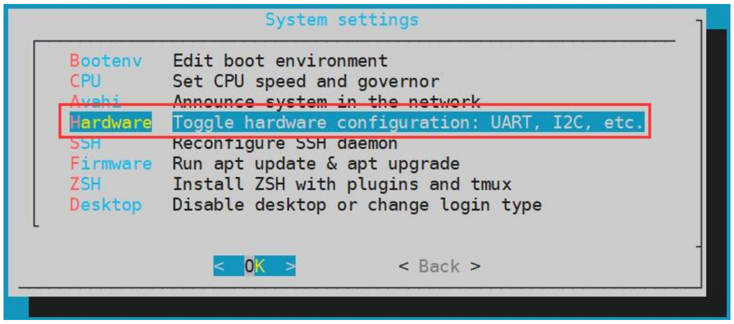
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions.

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

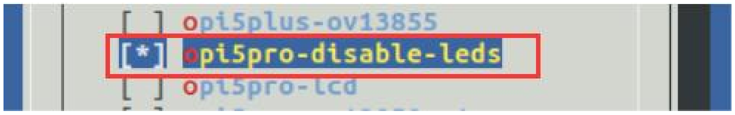
b. Select **System**



c. Select **Hardware**

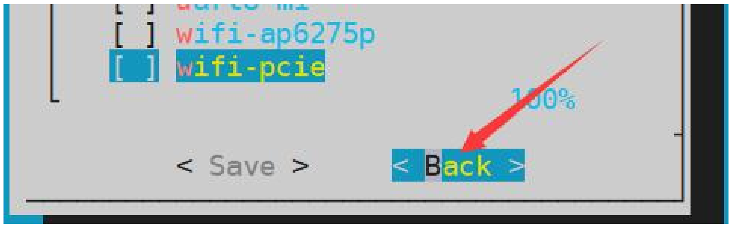


d. Then use the keyboard's arrow keys to locate the position shown in the picture below, and then use the space to select the **opi5pro-disable-leds** configuration

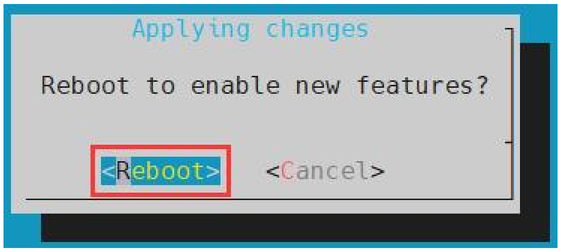


e. Select **<Save>**

f. Select **<Back>**



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



h. After restarting, you can see that only the red light on the development board is always on, and the green and blue lights will no longer flash.

3. 6. Network connection test

3. 6. 1. Ethernet port test

1) First, plug one end of the network cable into the Ethernet interface of the development board, and the other end of the network cable into the router, and make sure the network is open.

2) After the system starts, it will automatically assign an IP address to the Ethernet card through **DHCP**, and **no other configuration is required.**

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

```
orangepi@orangepi:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enP4p65s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:e0:4c:68:00:0f brd ff:ff:ff:ff:ff:ff
    inet 10.31.2.249/16 brd 10.31.255.255 scope global dynamic noprefixroute enP4p65s0
       valid_lft 42670sec preferred_lft 42670sec
    inet6 fe80::d5aa:9a6:cd41:942e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state DORMANT
group default qlen 1000
    link/ether 50:41:1c:f1:0f:7e brd ff:ff:ff:ff:ff:ff
```

When using ifconfig to check the IP address, if the following information is prompted, it is caused by not adding sudo. 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
```

1. Connect an HDMI display and log in to the system to use the ip addr show command to view the IP address.

2. Use the ip addr show command in a debugging serial terminal to check the IP address.

3. If there is no debugging serial port or HDMI display available, you can still check the IP address through the router's management interface. However, this method often results in people being unable to view the development board's IP address properly. If this occurs, follow these debugging steps:

A) First, check if the Linux system has started correctly. If the green LED on the development board is flashing, it generally indicates that the system has started correctly. If only the red LED is lit, it means the system has not started correctly.

B) Check if the Ethernet cable is securely plugged in or try using a different cable.

C) Try using a different router. Router issues are common and may include problems like the router not assigning IP addresses properly or the assigned IP address not being visible in the router's settings.

D) If changing the router is not an option, the only way to view the IP address is by connecting an HDMI display or using a debugging serial port.

Additionally, it's important to note that the development board's DHCP automatic IP address assignment requires no additional configuration.

4) The **ping** command can be interrupted by pressing the **Ctrl+C** shortcut key.

```
orangepi@orangepi:~$ ping www.baidu.com -I enP4p65s0
PING www.a.shifen.com (183.2.172.185) from 10.31.2.249 enP4p65s0: 56(84) bytes of data.
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=1 ttl=53 time=39.5 ms
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=2 ttl=53 time=33.1 ms
```

```
64 bytes from 183.2.172.185 (183.2.172.185): icmp_seq=3 ttl=53 time=32.4 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

Please refrain from connecting to Wi-Fi networks by modifying the /etc/network/interfaces configuration file, as this method may lead to issues with connecting to Wi-Fi networks.

3. 6. 2. 1. Connect the server version image to WIFI through commands

When the development board is not connected to Ethernet and HDMI display, and only the serial port is connected, it is recommended to use the commands demonstrated in this section to connect to Wi-Fi networks. This is because `nmtui` may not display graphical interfaces properly in some serial port software like Minicom, only showing characters. However, if the development board is connected to Ethernet or an HDMI display, you can still use the commands demonstrated in this section to connect to Wi-Fi networks.

1) First log in to the Linux system, there are three ways:

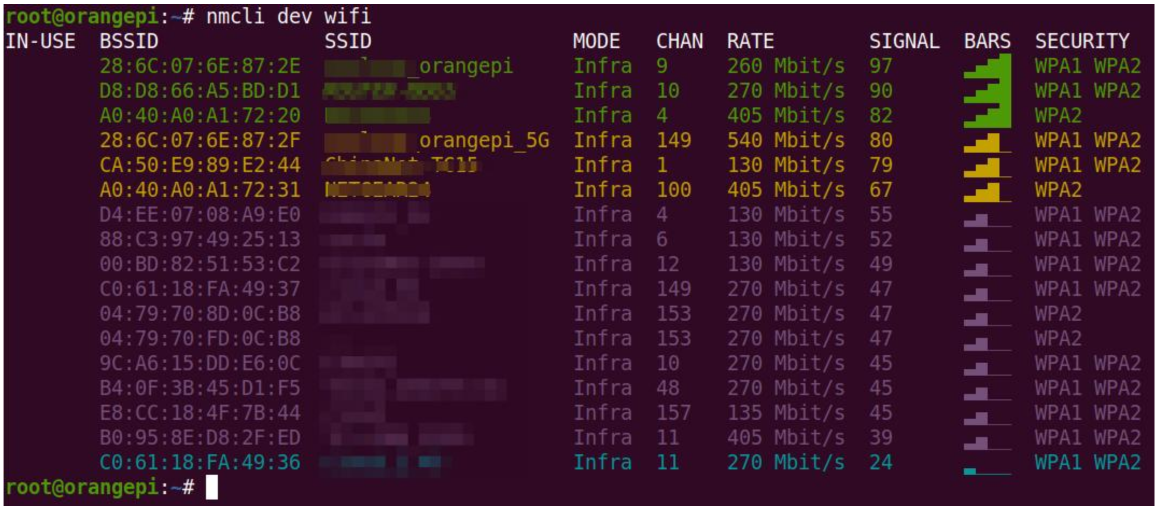
   a. If the development board is connected to a network cable, you can remotely log in to the Linux system through ssh.

   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.

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

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

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



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

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

b. **wifi_passwd** needs to be replaced with the password of the WIFI hotspot you want to connect to.

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

3) You can check the IP address of the 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 23:8c:d6:ae:76:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
        valid_lft 259192sec preferred_lft 259192sec
    inet6 240e:3b7:3240:c3a0:c401:a445:5002:ccdd/64 scope global dynamic noprefixroute
        valid_lft 259192sec preferred_lft 172792sec
    inet6 fe80::42f1:6019:a80e:4c31/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

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

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

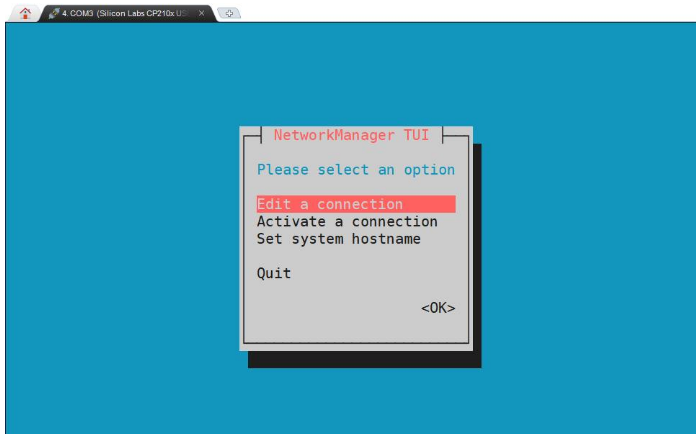
3.6.2.2. Server version image connects to WIFI graphically

1) First log in to the Linux system, there are three ways:
   a. If the development board is connected to 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, the graphical interface cannot be displayed using minicom)
   c. If the development board is connected to an HDMI display, you can log in to the Linux system through the HDMI display terminal.

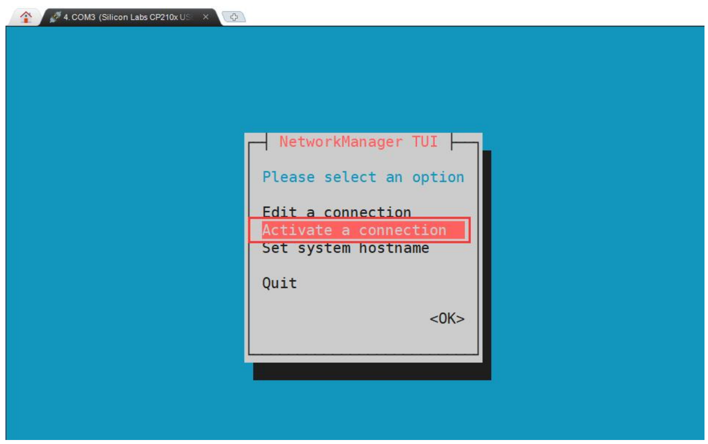
2) Then enter the nmtui command in the command line to open the wifi connection interface

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

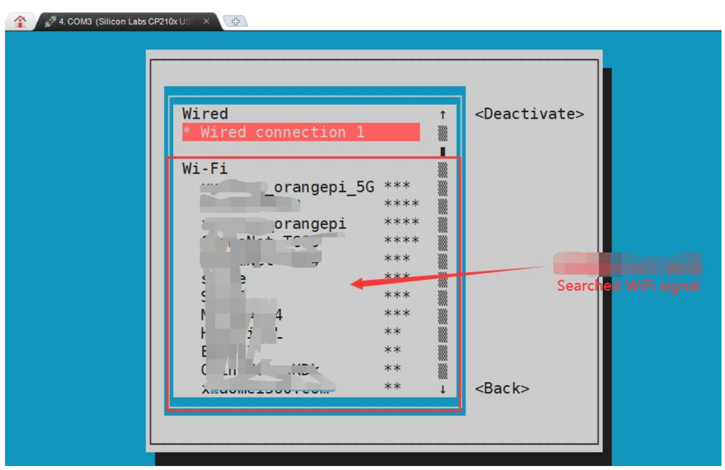
3) Enter the nmtui command to open the interface as shown below



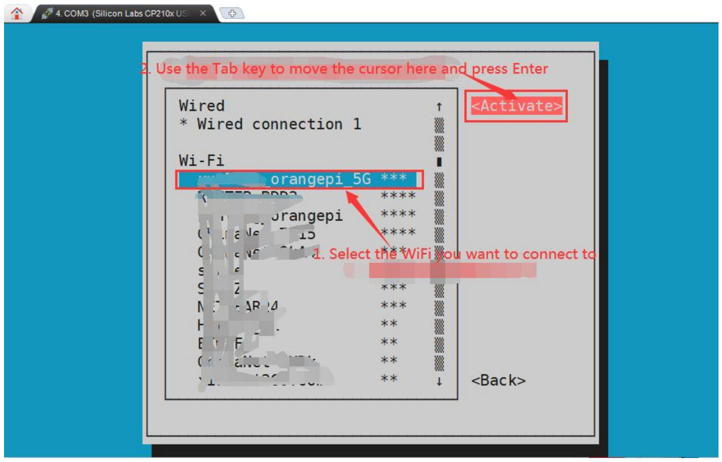
4) Select **Activate a connect** and press Enter



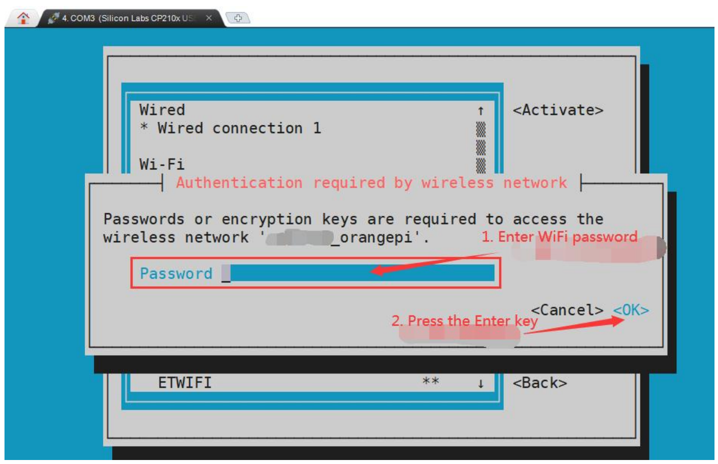
5) Then you can see all the searched WIFI hotspots



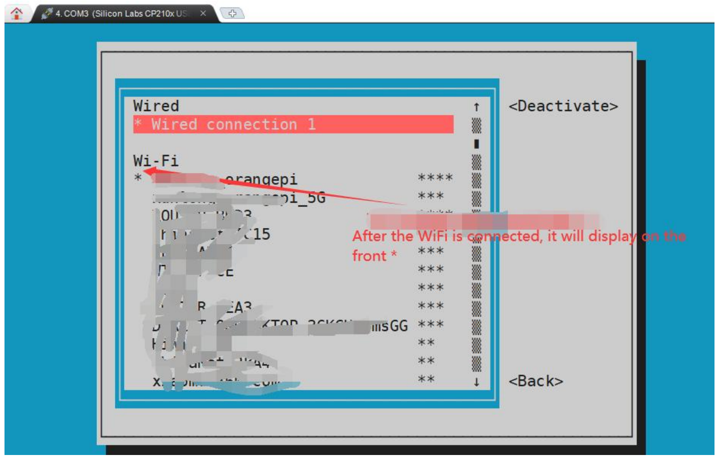
6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the cursor on **Activate** and press Enter.



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



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



9) You can check the IP address of the 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

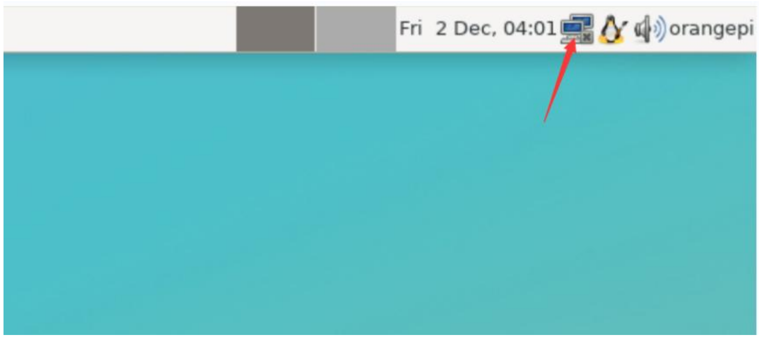
10) Use the **ping** command to test the connectivity of the wifi network. The **ping** command can be interrupted by pressing the **Ctrl+C** shortcut key.

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

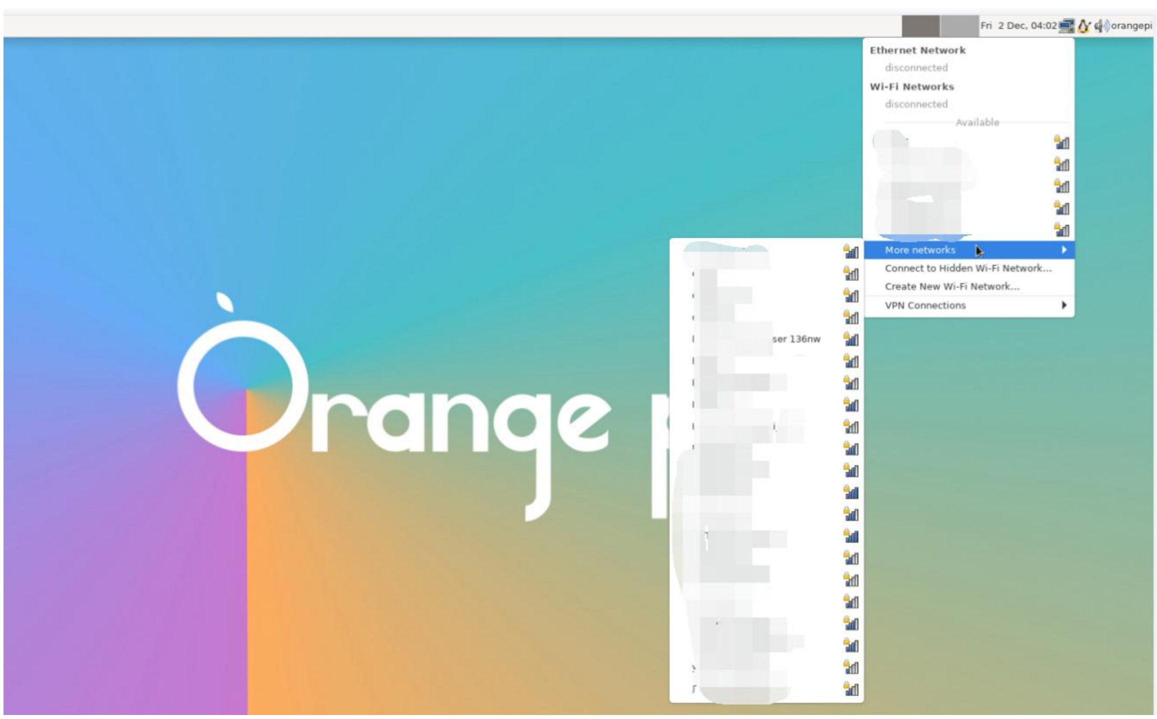
```
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

3. 6. 2. 3. Testing 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)

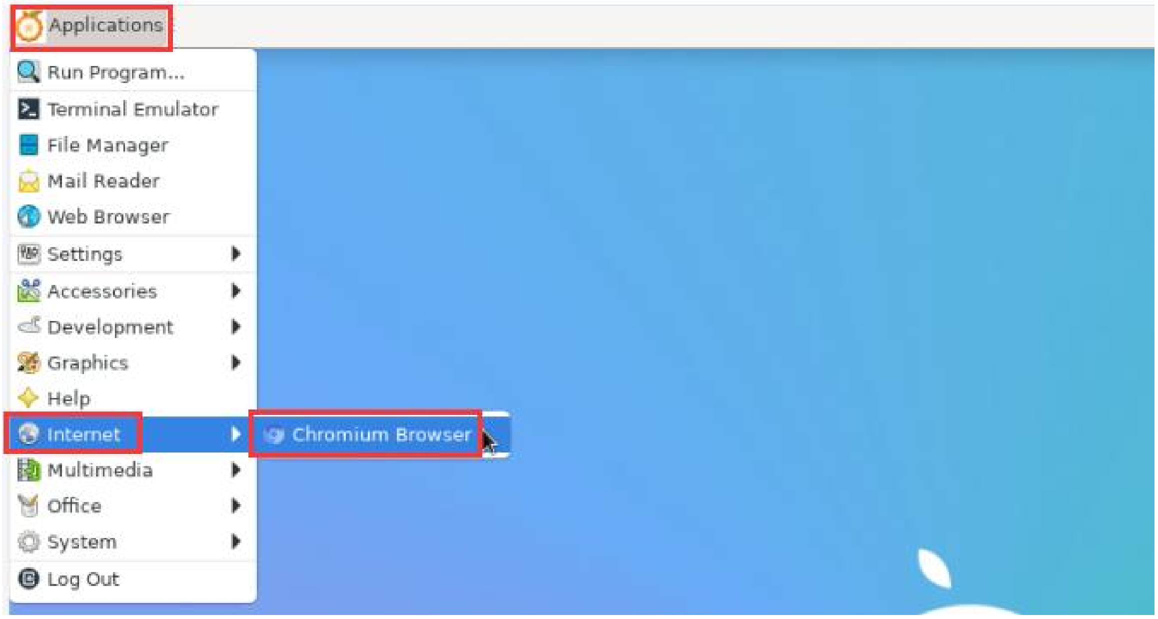


2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to.

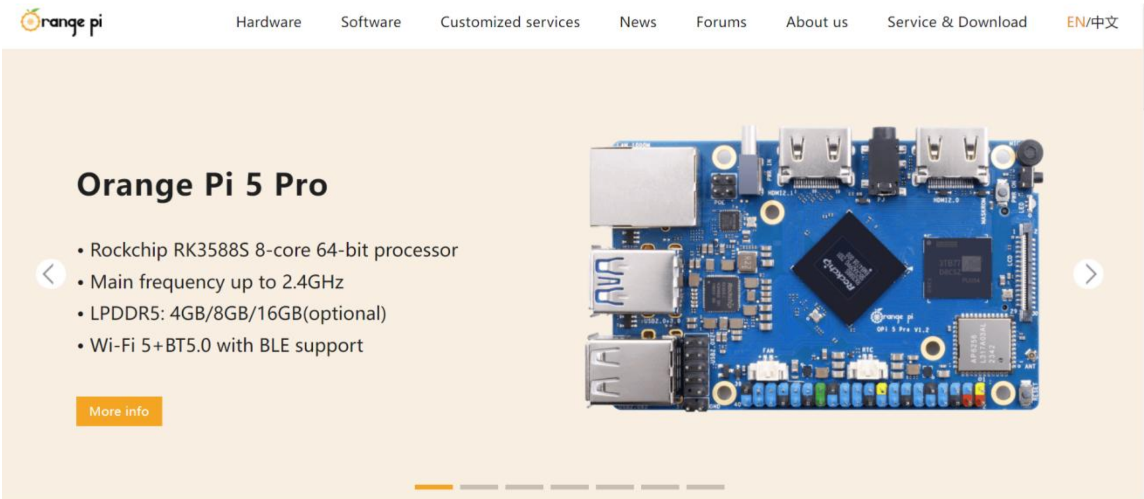


3) Then enter the password of the WIFI hotspot and click **Connect** to start connecting to WIFI

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



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



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

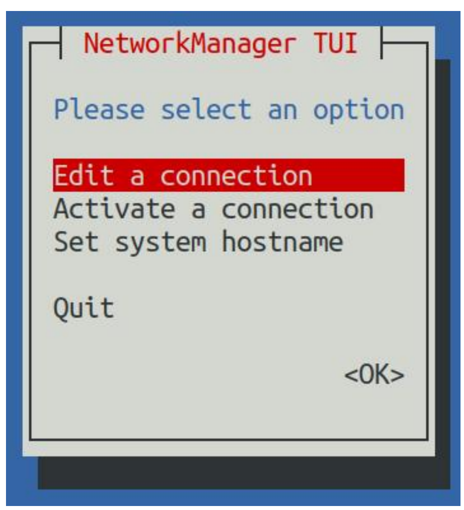
Please do not set a static IP address by modifying the/etc/network/interfaces configuration file.

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

1) First run the **nmtui** command

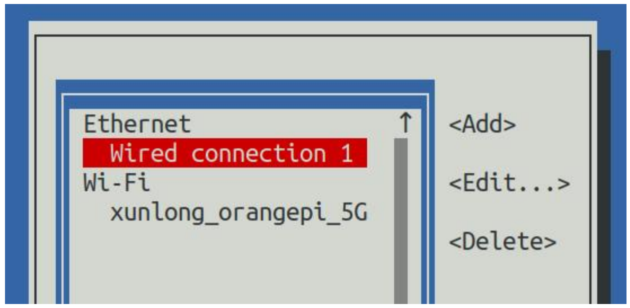
```
orangepi@orangepi:~$ sudo nmtui
```

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

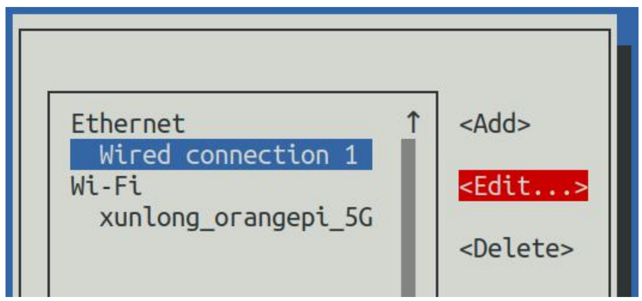


3) Then select the network interface for which a static IP address needs to be set. For

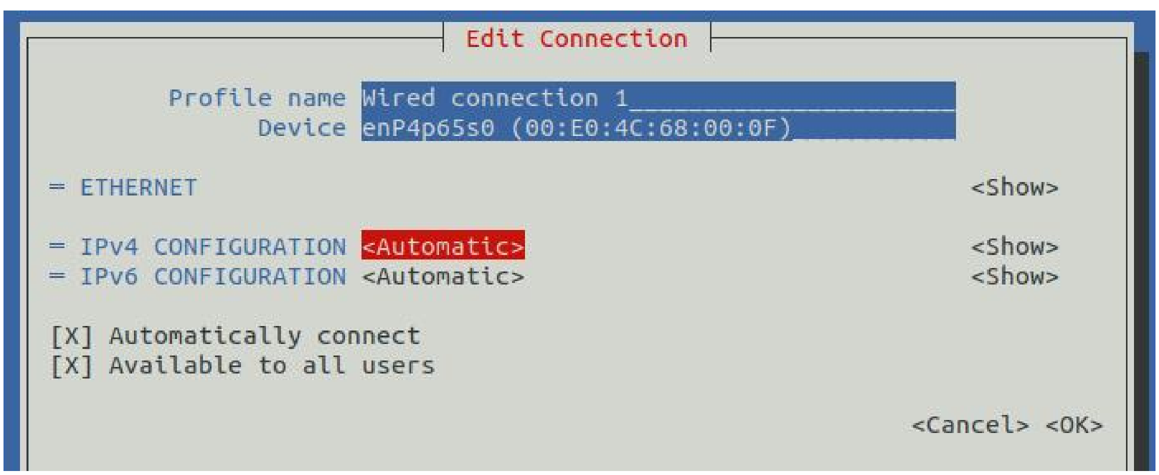
example, to set the static IP address of the **Ethernet**interface, select **Wired connection 1**



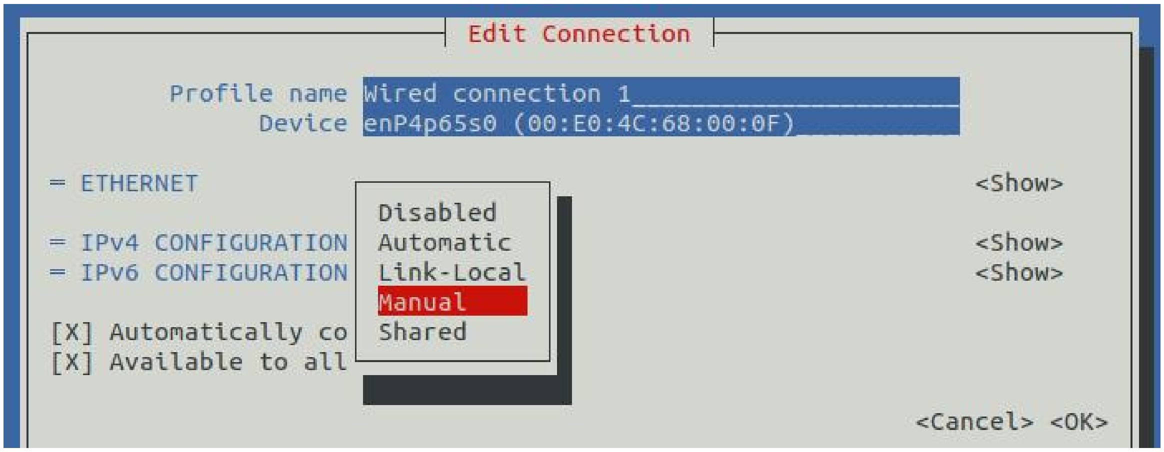
4) Then select **Edit** via the **Tab** key and press Enter.



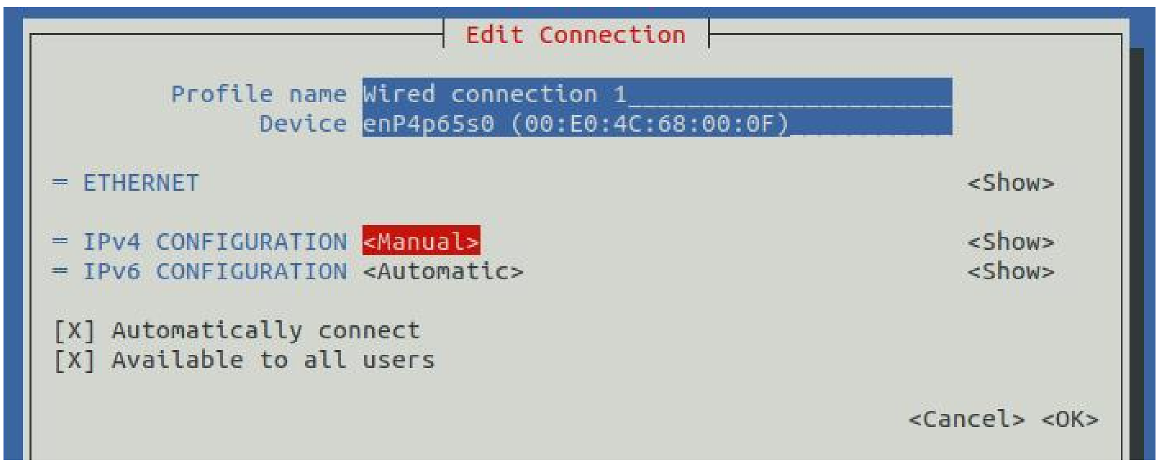
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4



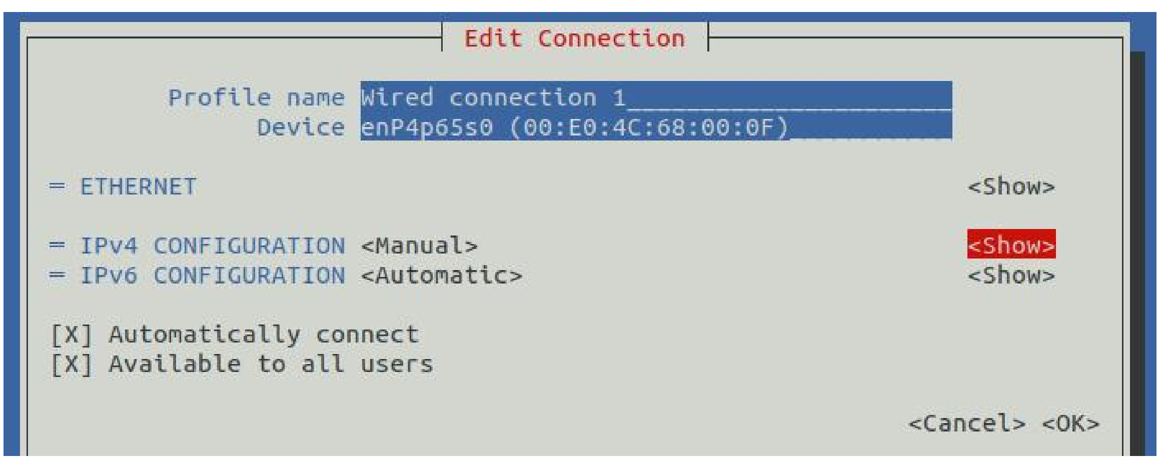
6) Then press Enter, select **Manual**through the up and down arrow keys, and then press Enter to confirm.



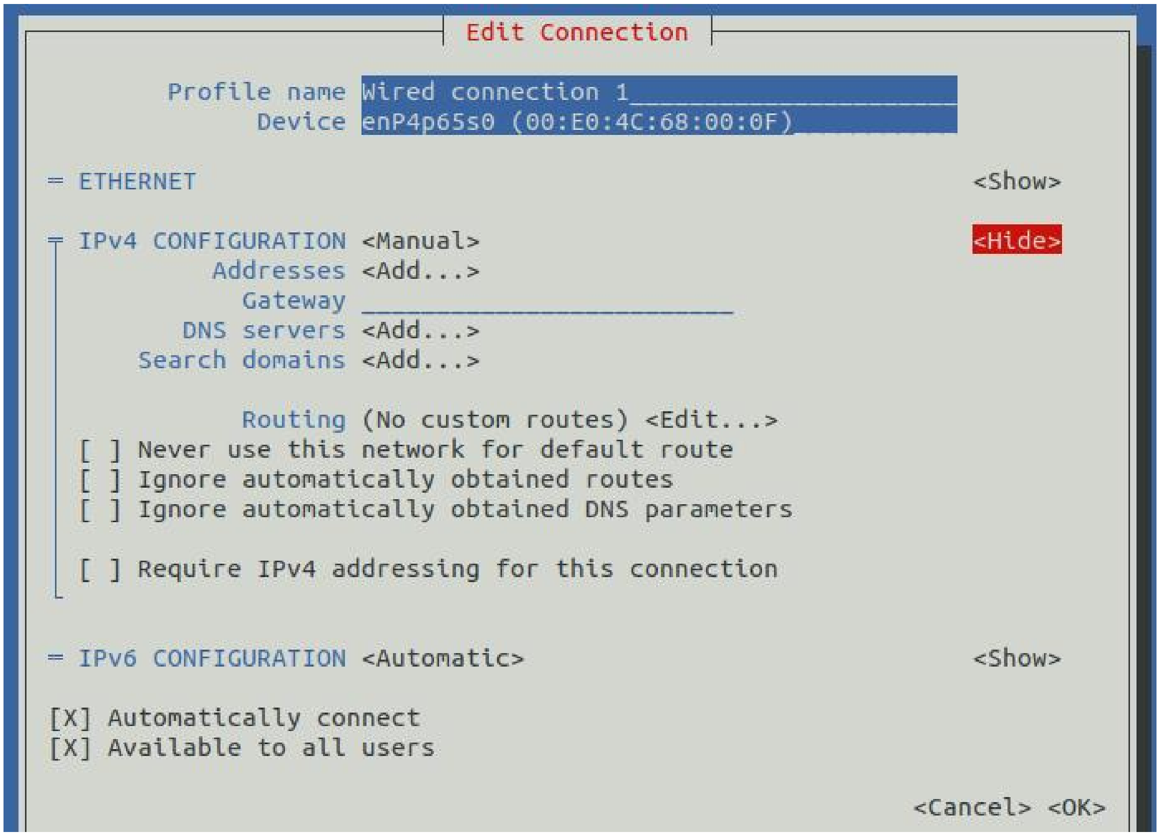
7) The display after selection is as shown below



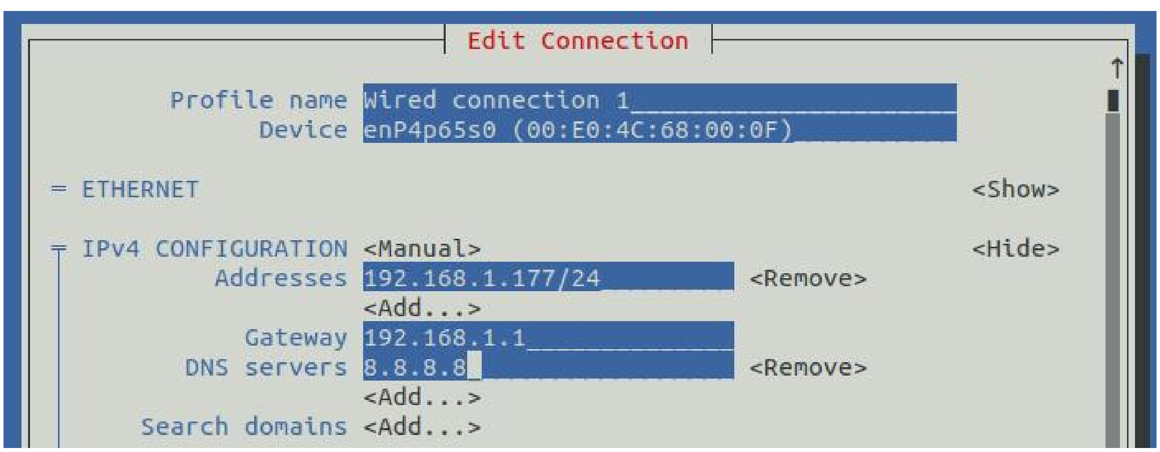
8) Then move the cursor to **<Show>**



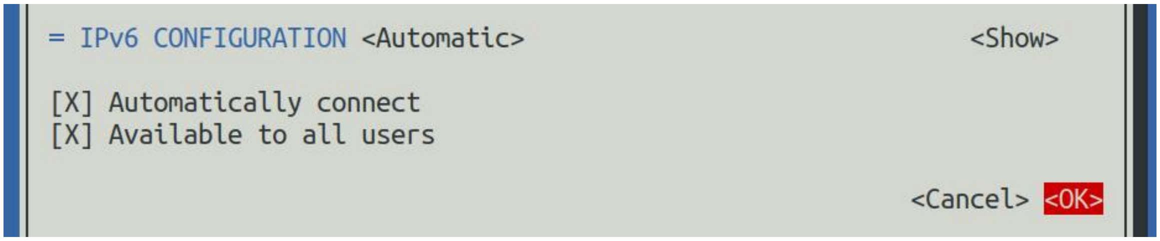
9) Then press Enter. After pressing Enter, the following setting interface will pop up.



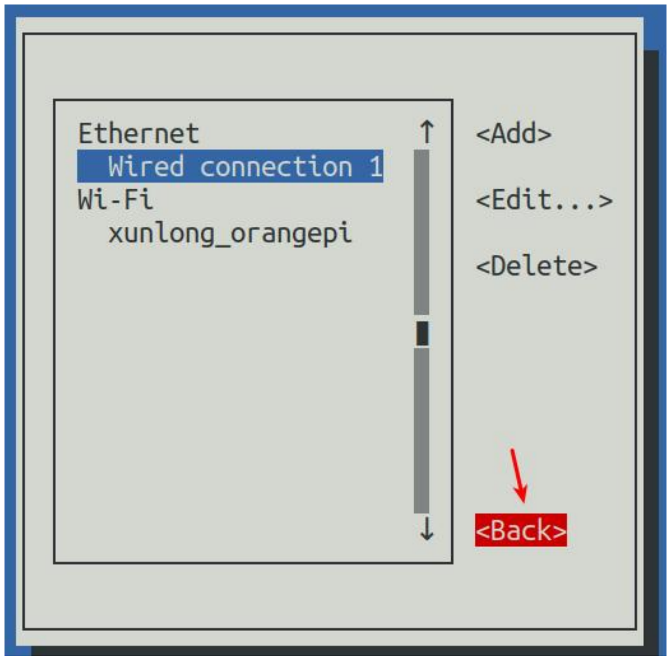
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address as shown in the figure below (there are many other setting options, please explore by yourself), **please set according to your specific needs. The values set in the image below are just an example**



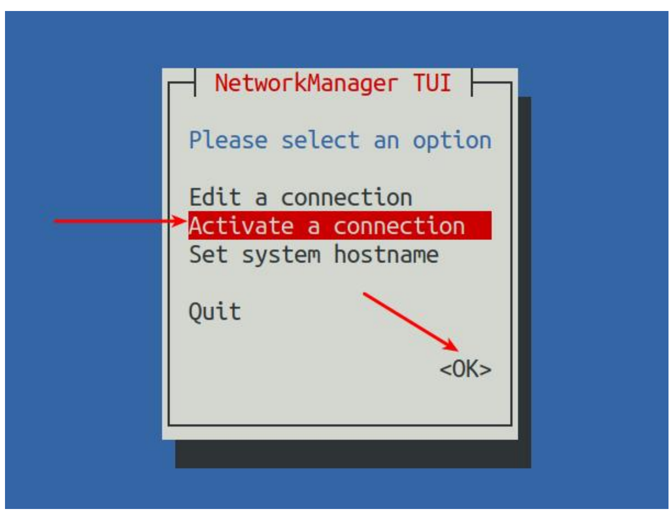
11) After setting, move the cursor to **<OK>** in the lower right corner, and then press Enter to confirm.



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

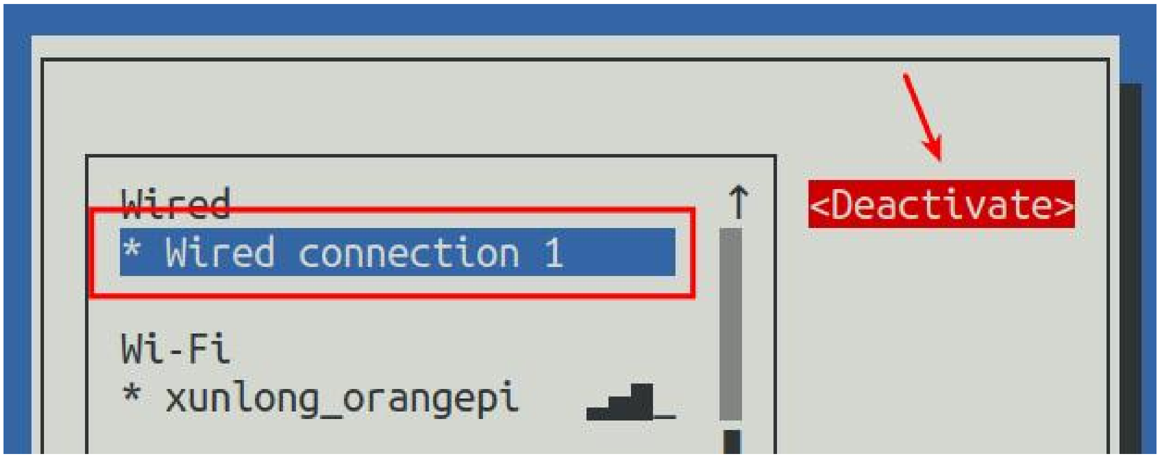


13) Then select **Activate a connection**, then move the cursor to**<OK>**, and finally click Enter

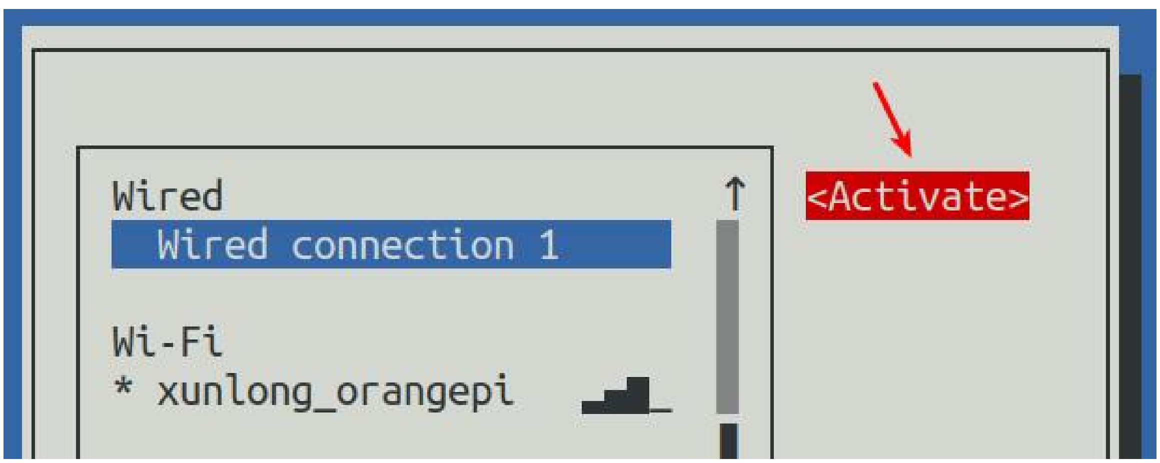


14) Then select the network interface that needs to be set, such as **<Deactivate>**, then

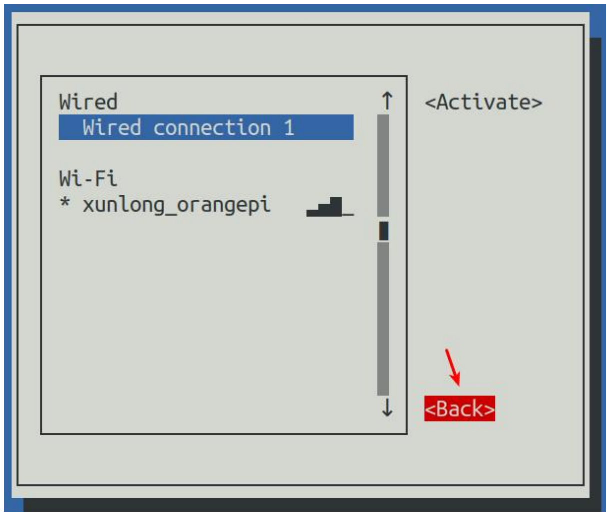
move the cursor to **<Deactivate>**, and then press the Enter key to disable **Wired connection 1**

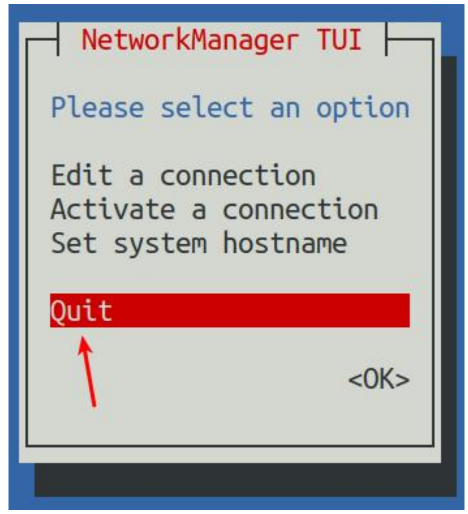


15) Then please do not move the cursor, and then press the Enter key to re-enable **Wired connection 1**, so that the static IP address set previously will take effect.



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





17) Then use **ip addr show enP4p65s0** to see that the IP address of the network port has become the static IP address set previously.

```
orangepi@orangepi:~$ ip addr show enP4p65s0
2: enP4p65s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:e0:4c:68:00:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.177/24 brd 192.168.1.255 scope global noprefixroute enP4p65s0
       valid_lft forever preferred_lft forever
    inet6 fe80::d5aa:9a6:cd41:942e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

18) Then you can test the network connectivity to check whether the IP address is configured OK. The **ping** command can be interrupted by using the **Ctrl+C** shortcut key.

```
orangepi@orangepi:~$ ping 192.168.1.47 -I enP4p65s0
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 nmcli command to set static IP address

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

2) Then 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 names may not be 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

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

b. **ipv4.addresses** is 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 enP4p65s0
3: enP4p65s0: <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. Method to create 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 users from making complicated configurations. 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. You can use the create_ap command to create a WIFI hotspot. The basic command format of create_ap is as follows:

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

*** options: You can use this parameter to specify the encryption method, frequency band of WIFI hotspot, bandwidth mode, 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, usually eth0**

*** access-point-name: hotspot name**

*** passphrase: Password of hotspot**

3. 6. 4. 1. create_ap method to create WIFI hotspot in NAT mode

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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP4p65s0 orangepi orangepi
```

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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP4p65s0 orangepi orangepi
Config dir: /tmp/create_ap.wlan0.conf.Ks6HobEw
PID: 5405
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.Ks6HobEw/hostapd_ctrl
ap0: interface state UNINITIALIZED->ENABLED
ap0: AP-ENABLED
```

3）At this time, take out your mobile phone and find the WIFI hotspot named **orangepi** created by the development board in the searched WIFI list. Then you can click **orangepi** to connect to the hotspot. The password is **orangepi** the one set above.

4）The display after successful connection is as shown below

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

```
orangepi@orangepi:~$ ifconfig enP4p65s0
enP4p65s0: 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 the 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. For example, use the -g parameter to specify the network segment of the access point AP as 192.168.2.1.

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP4p65s0 orangepi orangepi -g 192.168.2.1
```

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

7）Without specifying the **--freq-band** parameter, 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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP4p65s0 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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP4p65s0 orangepi orangepi --hidden
```

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

3.6.4.2. create_ap method to create WIFI hotspot in bridge mode

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP4p65s0 orangepi orangepi
```

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP4p65s0 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）At this time, take out your mobile phone and find the WIFI hotspot named **orangepi** created by the development board in the searched WIFI list. Then you can click **orangepi** to connect to the hotspot. The password is **orangepi** set above.

4）The display after successful connection is as shown below

5）In 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 to which the development board is connected). For example, the IP of the development board here is **192.168.1.X**

```
orangepi@orangepi:~$ ifconfig enP4p65s0
enP4p65s0: 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. Too **192.168.1.X**。

6）Without specifying the **--freq-band** parameter, 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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP4p65s0 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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP4p65s0 orangepi orangepi --hidden
```

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

3. 7. SSH Remote Login to Development Board

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

3. 7. 1. SSH Remote Login to Development Board on Ubunt

1) Obtain the IP address of the development board

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

```
test@test:~$ ssh 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 your password here, the default password is orangepi）
```

Note that when entering a password, the specific content of the entered password will not be displayed on the screen. Please do not think that there is any malfunction. Just press Enter after entering the password.

If you are prompted to refuse the connection, as long as you are using the image provided by Orange Pi, please do not doubt whether the orangepi password is incorrect, but look for other reasons.

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

If ssh cannot log in to the Linux system normally, first check whether the IP address of the development board can be pinged. If there is no problem with pinging, you can log in to the Linux system through the serial port or HDMI display and enter the following command on the development board before trying again. Can it be connected:

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

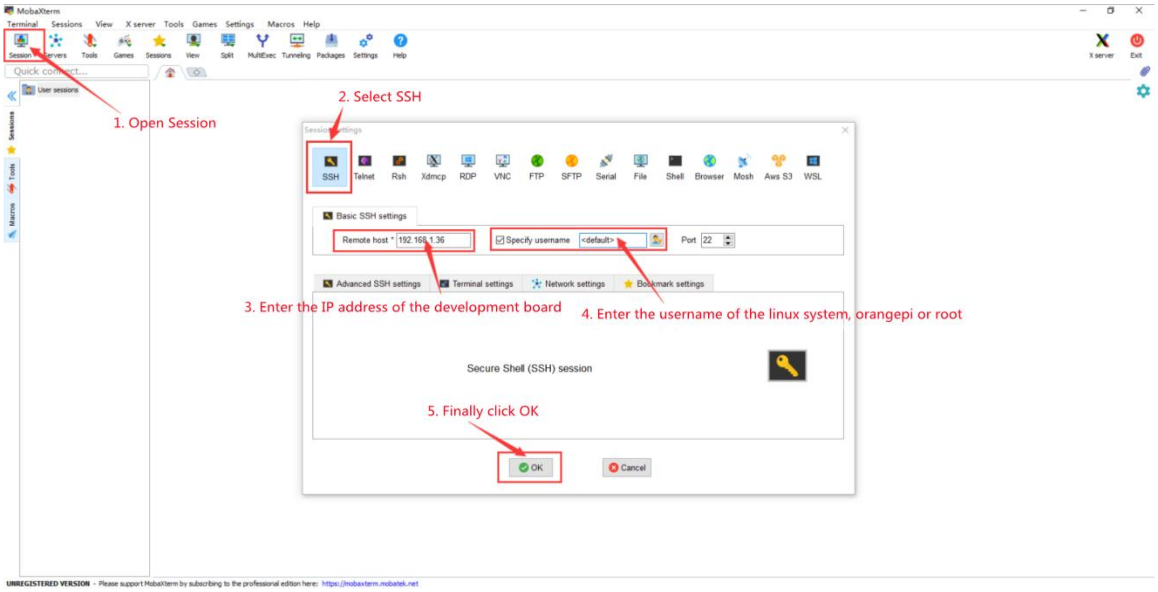
If it still doesn't work, please try restarting the system.

3. 7. 2. SSH remote login development board under Windows

1) First get the IP address of the development board

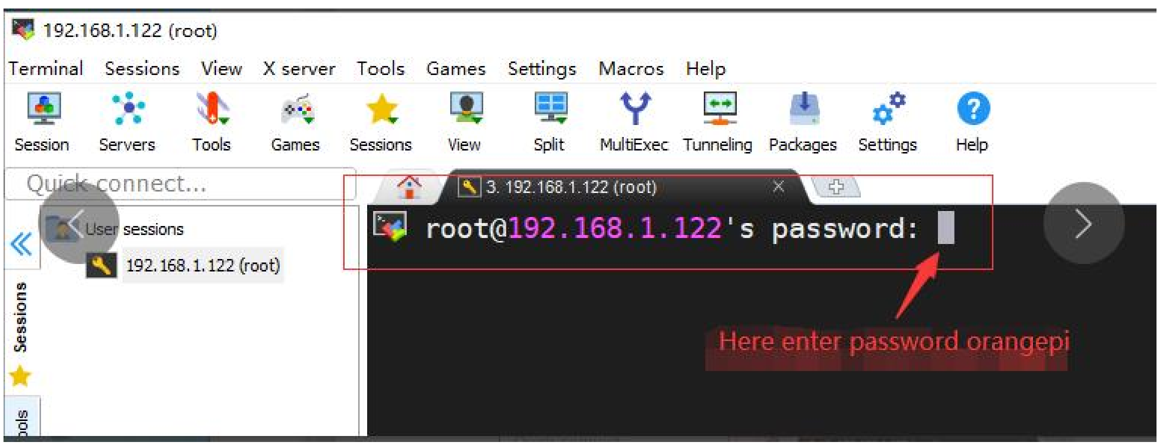
2) You can use MobaXterm to remotely log in to the development board under Windows. First create a new ssh session

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

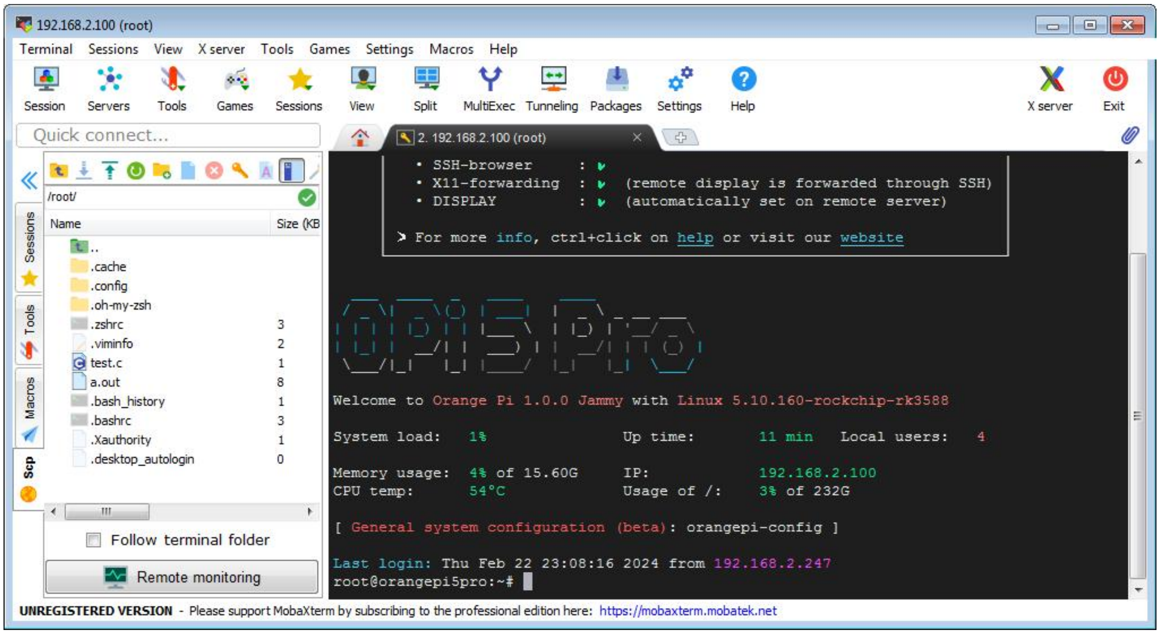


3) You will then be prompted to enter a password. The default passwords for both root and orangepi users are orangepi

Note that when entering a password, the specific content of the entered password will not be displayed on the screen. Please do not think that there is any malfunction. Just press Enter after entering the password.



4) After successfully logging into the system, the display is as shown 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 adb tool on 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        #Please replace the IP address 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@orangepi5pro:/#      <--- After seeing this prompt, it means you have successfully logged in to the development board
```

6) The command to use adb to upload files to the development board is as follows

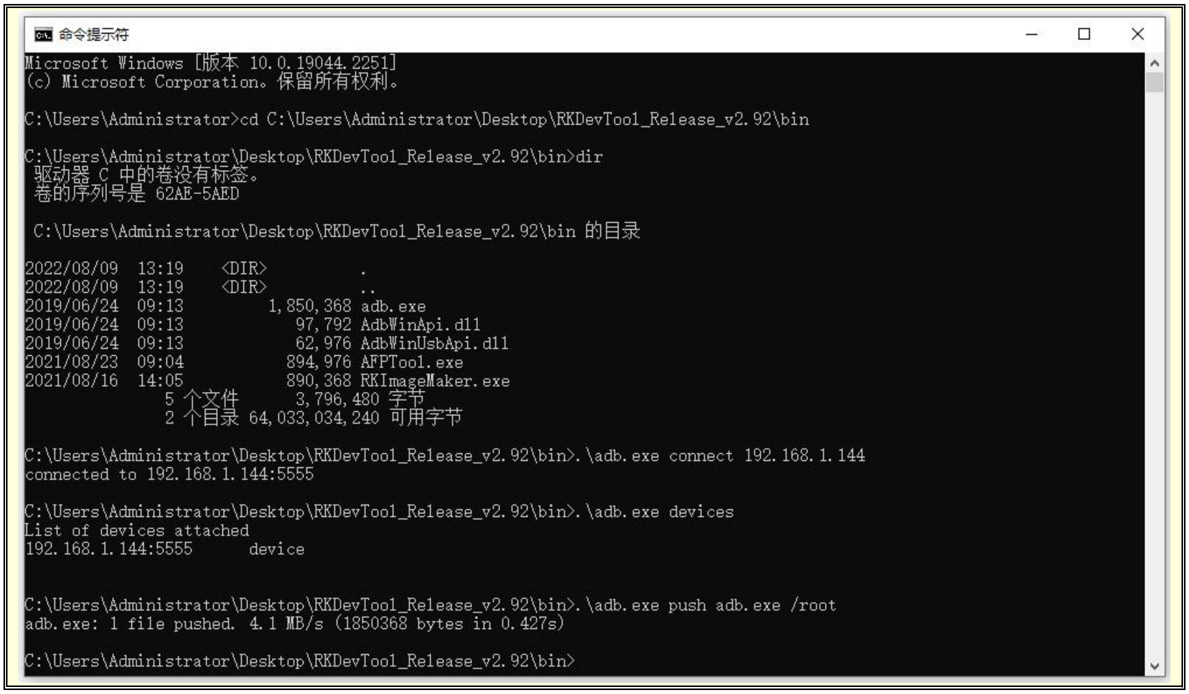
```
test@test:~$ adb push filename /root
filename: 1 file pushed. 3.7 MB/s (1075091 bytes in 0.277s)
```

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 method of burning the Android image to SPIFlash+NVMe SSD)

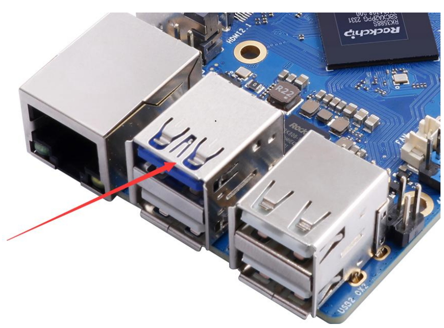
An example of using adb in Windows looks like this:



3.8.2. Use USB2.0 male-to-male data cable to connect ad

1) First prepare a good quality USB2.0 male-to-male data cable

2) Then connect the development board and Ubuntu PC through the USB2.0 male-to-male data cable. The location of the USB2.0 interface of the development board that supports the device function is as shown in the figure below:



3) Then run the following command to set the USB2.0 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
```

4) **Then 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
```

5) Then install the adb tool on Ubuntu PC

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

6) Then use the following command to check whether 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

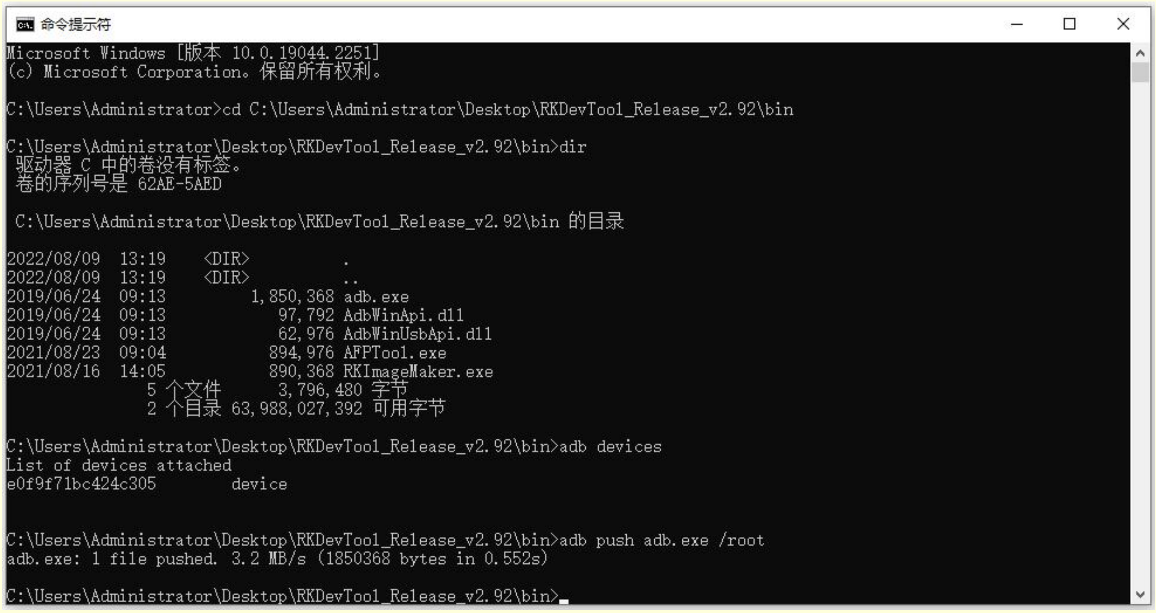
```
test@test:~$ adb shell
root@orangepi5pro:/#     <--- Upon seeing this prompt, it indicates that the development board has been successfully logged in
```

9) The command to use adb to upload files to the development board 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 method of burning the Android image to SPIFlash+NVMe SSD)

An example of using adb in Windows looks like this：



3.9. Method of uploading files to the development board Linux system

3.9.1. Method to upload files to the development board Linux system in Ubuntu PC

3.9.1.1. How to upload files using scp command

1) Use the scp command to upload files to the Linux system of the development board in Ubuntu PC. The specific command is as follows

a. **file_path：** Needs to be replaced with the path of the file to be uploaded

b. **orangepi：** This is the user name of the development board's Linux system. It can also be replaced with something else, such as root

c. **192.168.xx.xx:** This is the IP address of the development board. Please modify it according to the actual situation

d. **/home/orangepi:** The path in the development board Linux system 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

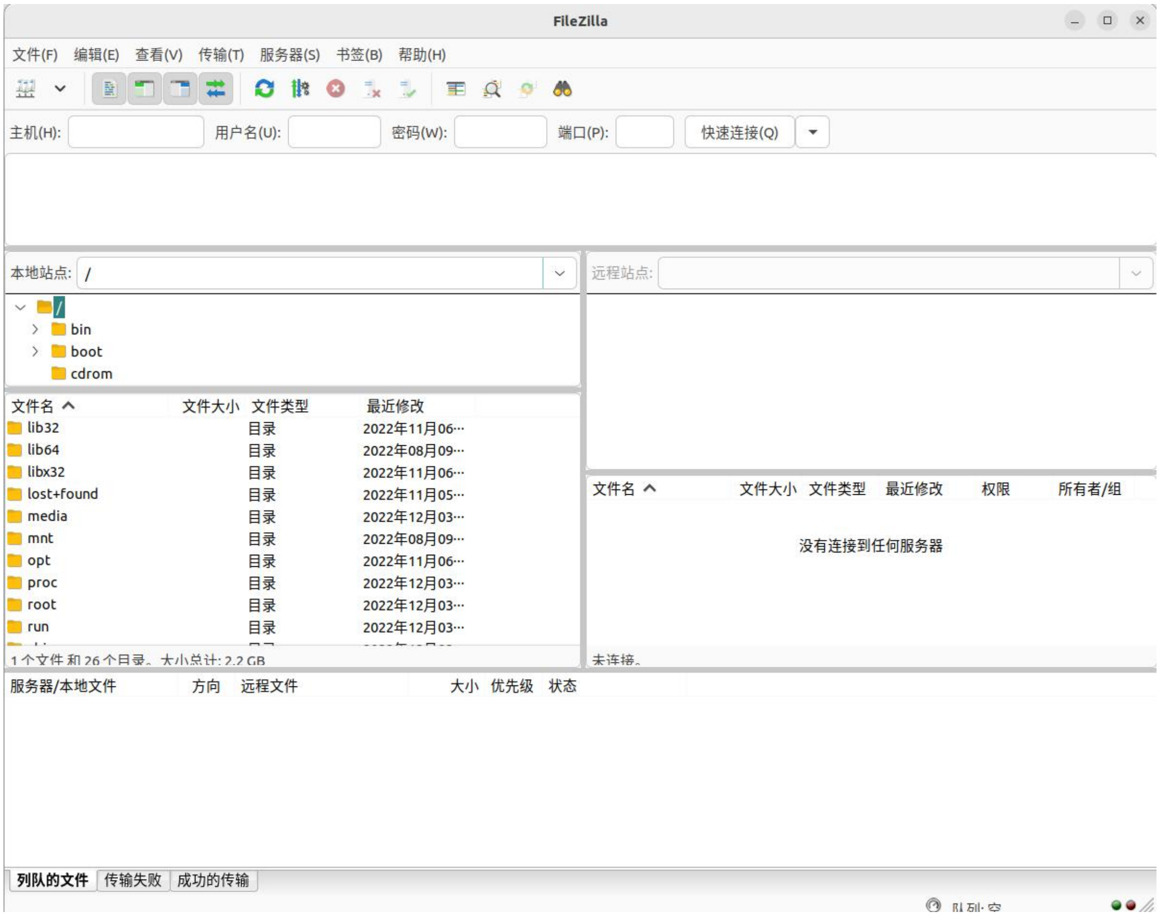
1) First install filezilla in Ubuntu PC

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

2) Then use the following command to open filezilla

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

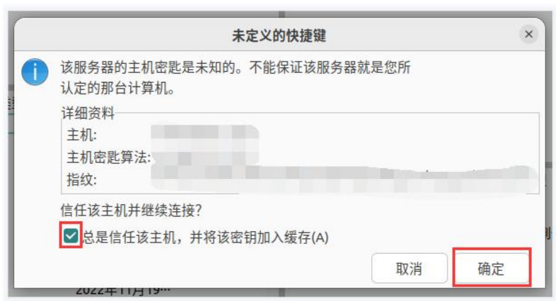
3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty



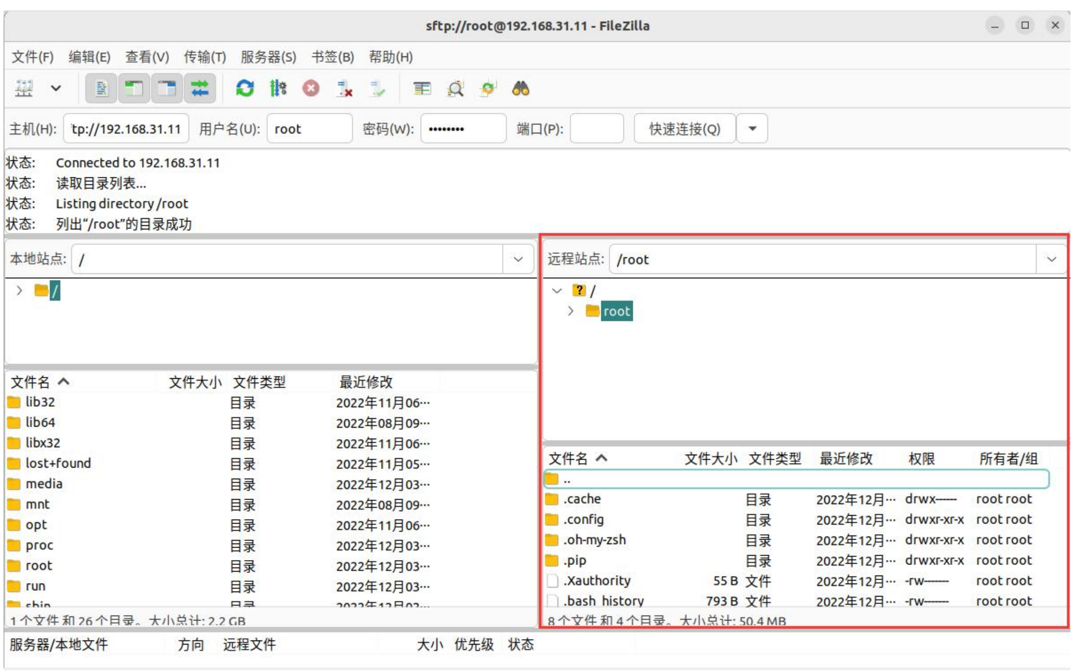
4) The method of connecting the development board is as shown in the figure below

5) Then choose to **save the password** and click **OK**

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

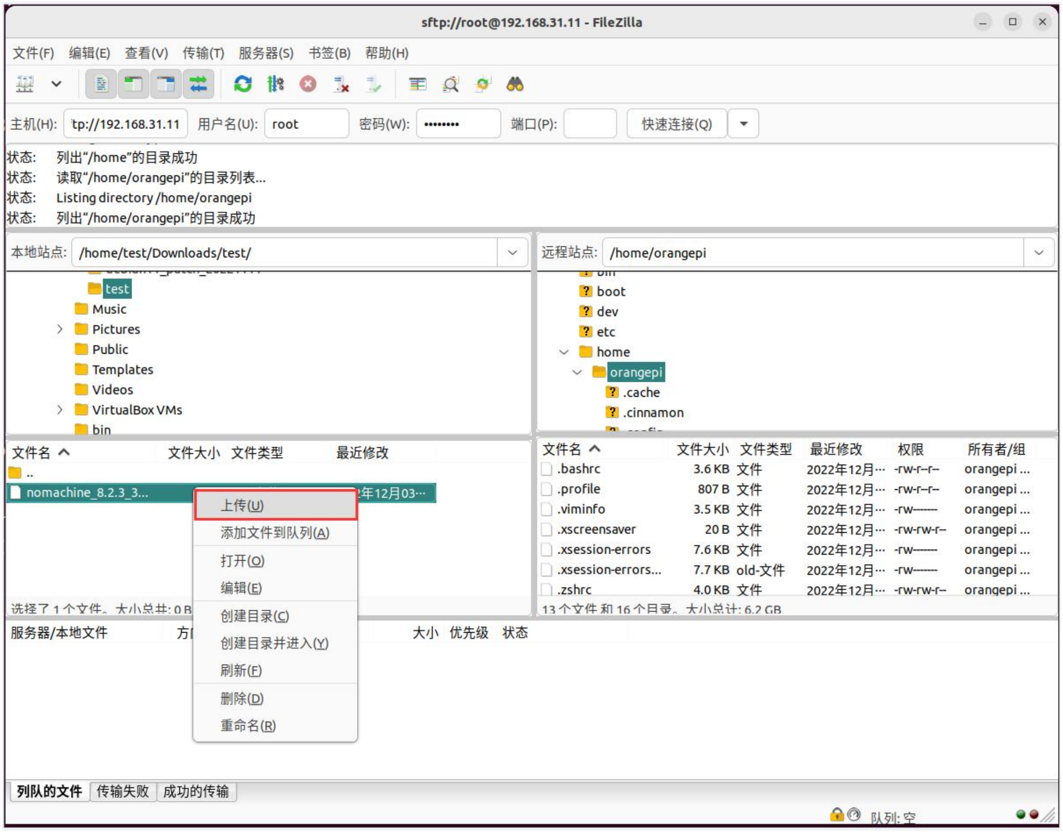


7) After the connection is successful, you can see the directory structure of the development board's Linux file system on the right side of the filezilla software



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

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



9) After the upload is completed, you can go to the corresponding path in the development board Linux system to view the uploaded file

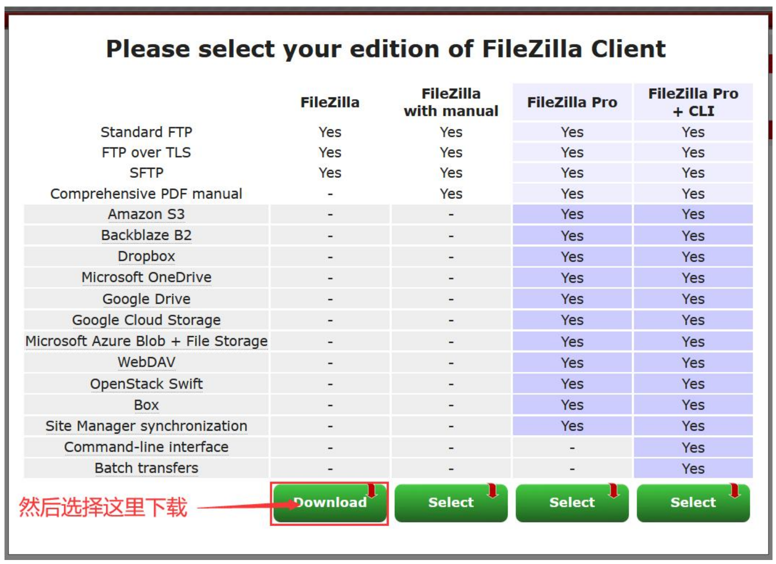
10) The method of uploading a folder is the same as the method of uploading a file, so I won't go into details here

3. 9. 2. How to upload files from Windows PC to development board in Linux system

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?type=client

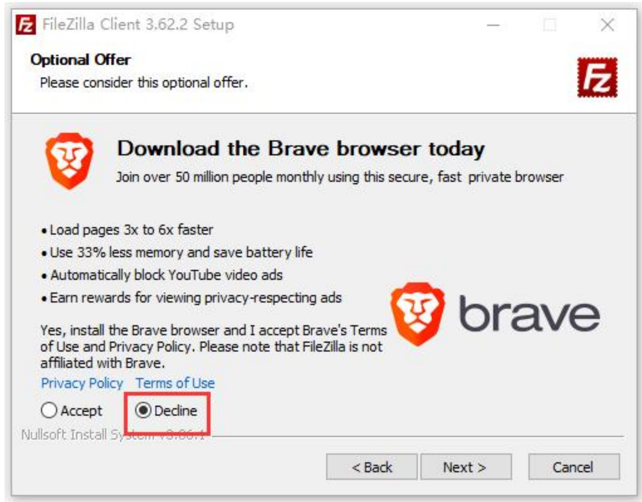

Please select your edition of FileZilla Client

	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
然后选择这里下载	Download	Select	Select	Select

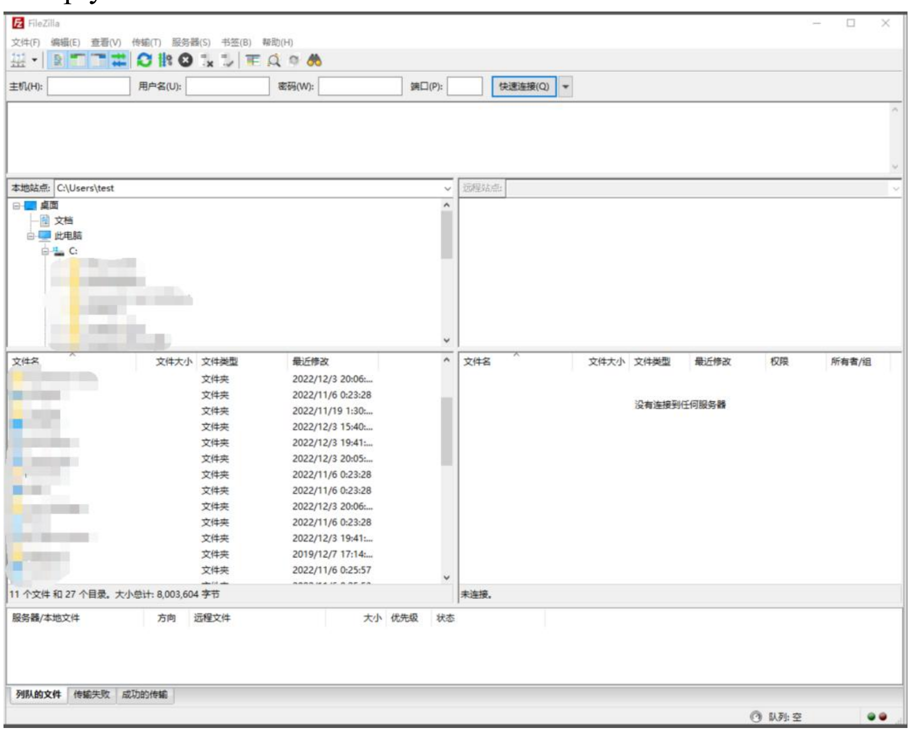
2) The downloaded installation package is as shown below, then double-click to install it directly

FileZilla_Server_1.5.1_win64-setup.exe

During the installation process, please select **Decline** on the following installation interface, and then select **Next>**

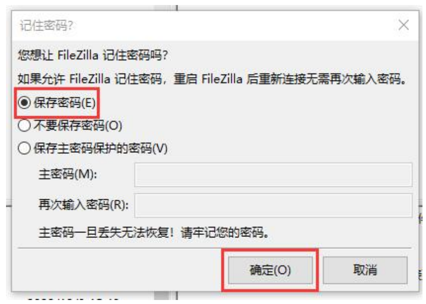


3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty

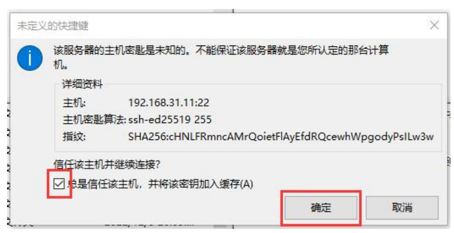


4) The method of connecting the development board is as shown in the figure below：

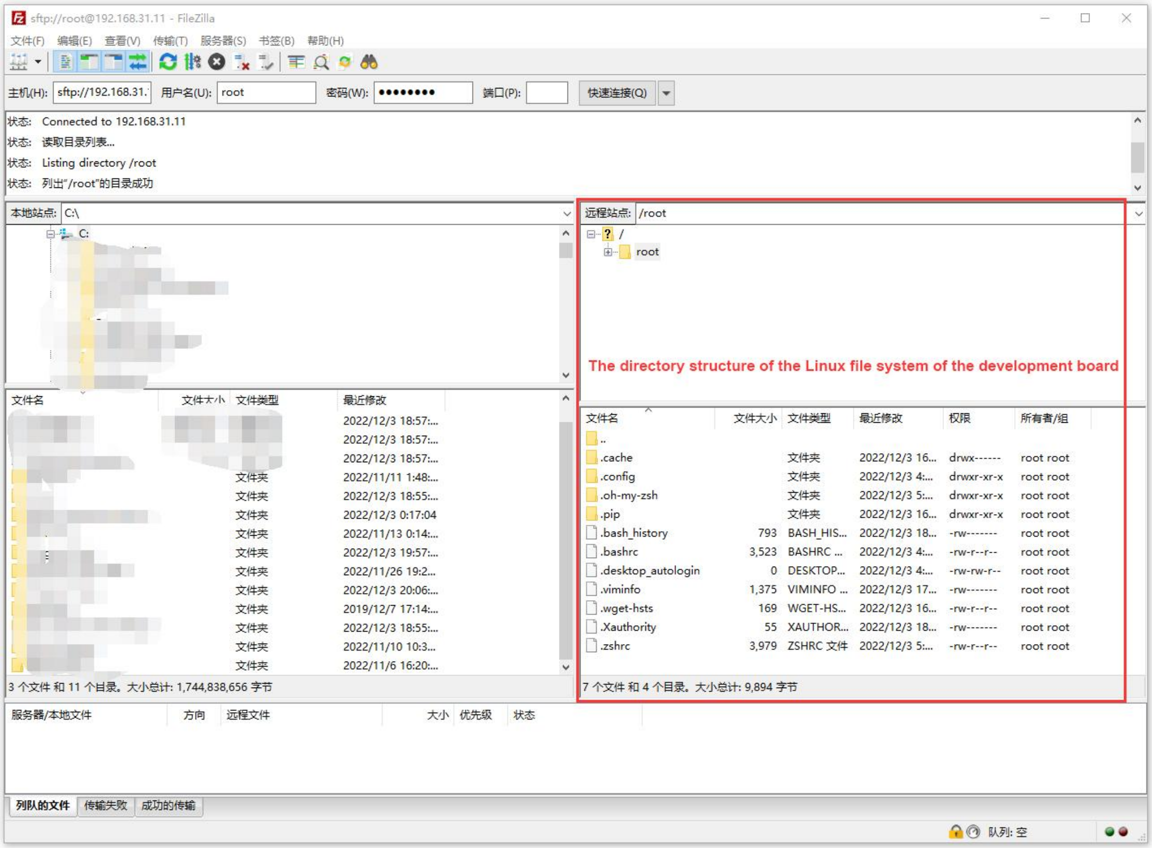
5) Then choose to **save the password** and click **OK**



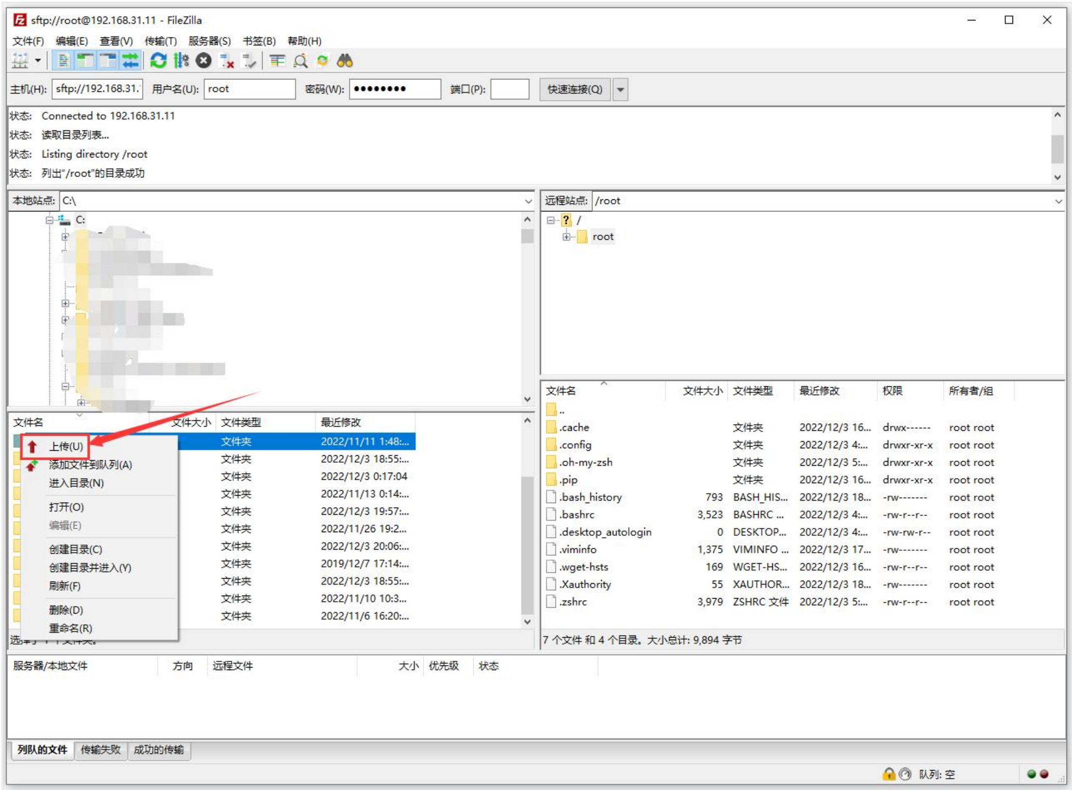
6) Then select **Always trust this host** and click **OK**



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



8) Then select the path to be uploaded to the development board on the right side of the filezilla software, select the file to be uploaded in the Windows PC on the left side of the filezilla software, right-click the mouse, and then click the upload option to start uploading the file to the development board



9) After the upload is completed, you can go to the corresponding path in the development board Linux system to view the uploaded file

10) The method of uploading a folder is the same as the method of uploading a file, so I won't go into details here

3. 10. HDMI Test

3. 10. 1. HDMI display test

2) Use HDMI to HDMI cable to connect Orange Pi development board and HDMI display

3) After starting the Linux system, if there is image output on the HDMI display, it means that the HDMI interface is working normally

Note that although many laptops have HDMI interfaces, the HDMI interface of the laptop generally only has the output function and does not have the HDMI in function, which means that the HDMI output of other devices cannot be displayed on the laptop screen.

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

When HDMI does not display, please first check whether the HDMI cable is plugged in tightly. After confirming that the wiring is OK, you can try a different screen to see if there is a display.

3. 10. 2. HDMI to VGA display test

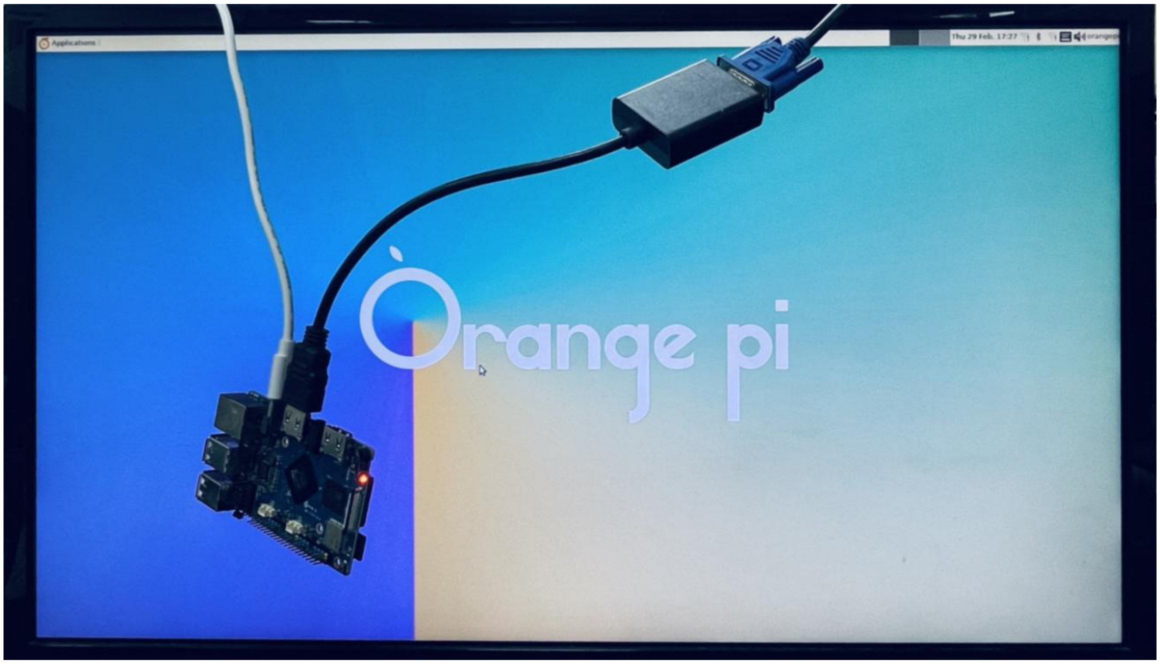
1) First you need to prepare the following accessories

a. HDMI to VGA converter

b. A VGA Cable

c. A monitor or TV that supports VGA interface

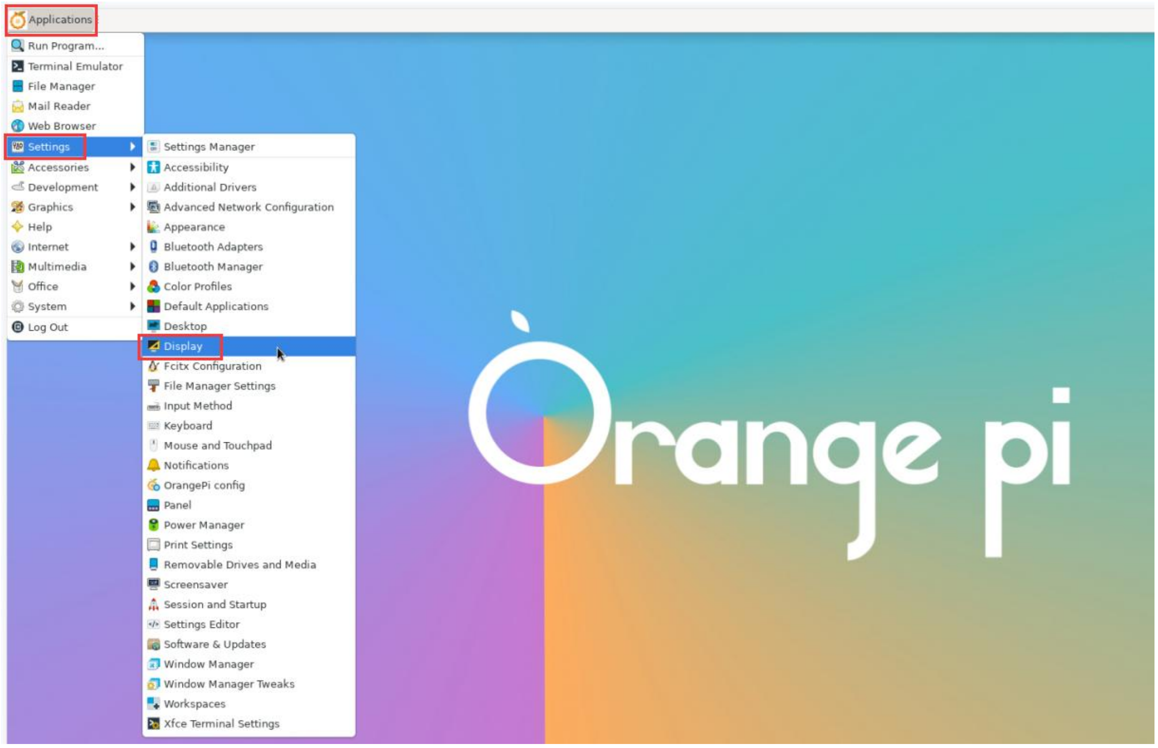
2) HDMI to VGA display test is as follows



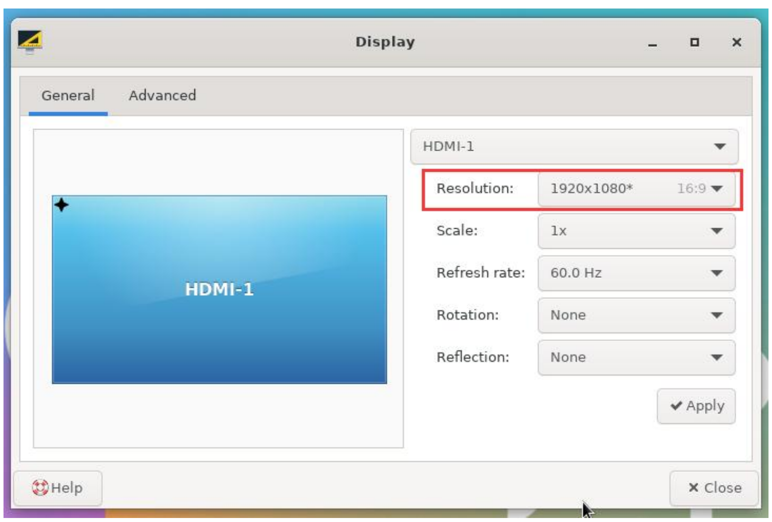
When using HDMI to VGA display, the development board and the Linux system of the development board do not need to make any settings. You only need the HDMI interface of the development board to display normally. So if there is a problem with the test, please check whether there is a problem with the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

3. 10. 3. HDMI resolution setting method

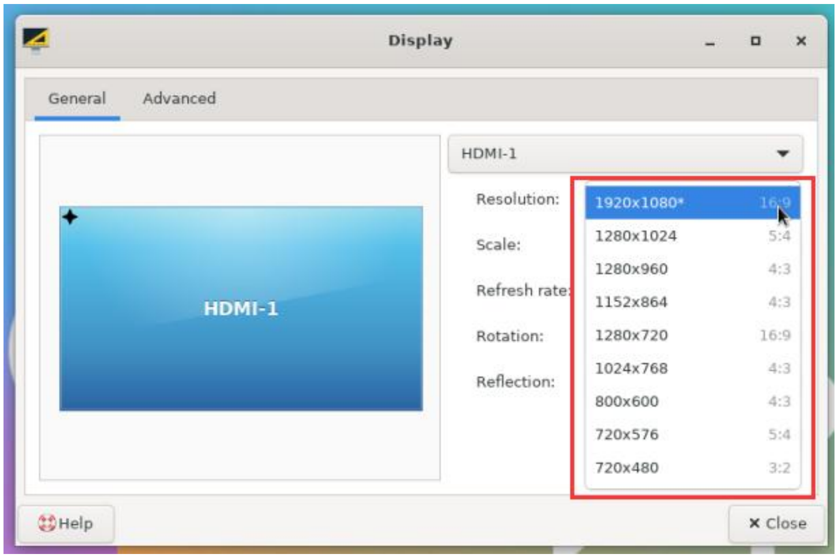
1) First open **Display** in **Settings**



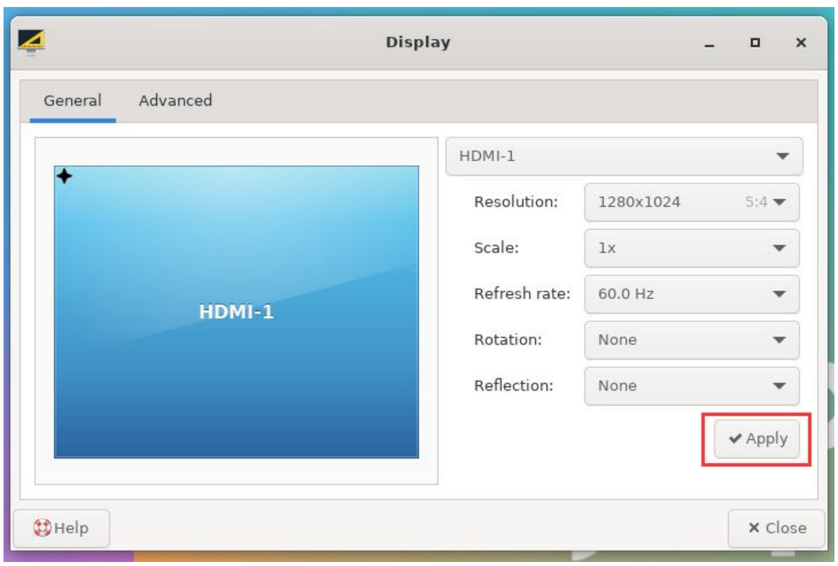
2) Then you can see the current resolution of the system



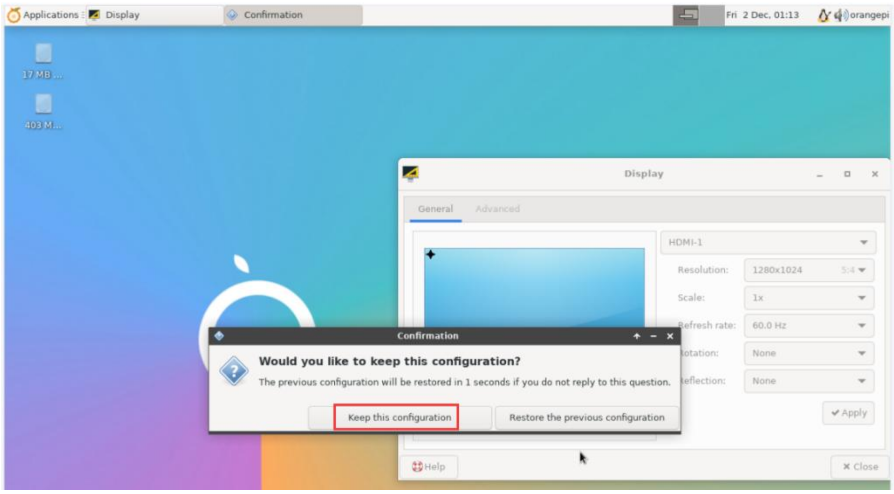
3) Click the drop-down box of Resolution to see all the resolutions currently supported by the monitor



4) Then select the resolution you want to set and click Apply



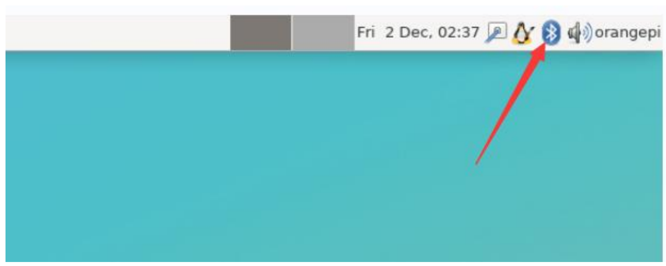
5) Wait until the new resolution is set and then select **Keep the configuration**



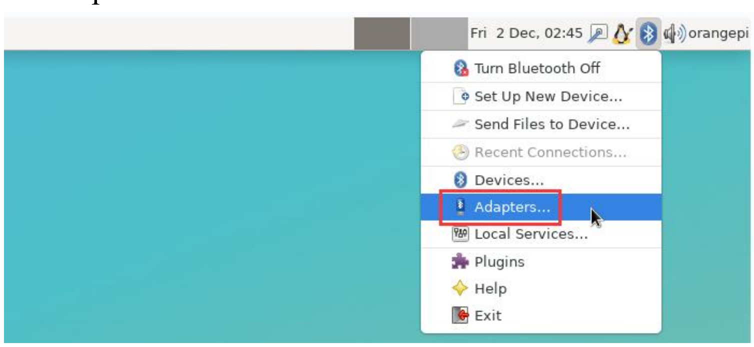
3. 11. How to use Bluetooth

3. 11. 1. Testing method of desktop version image

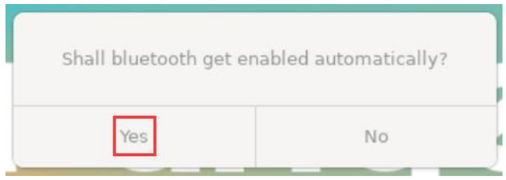
1) Click the Bluetooth icon in the upper right corner of the desktop



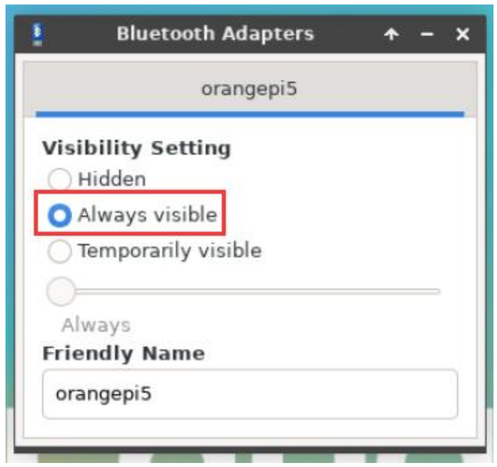
2) Then select the adapter



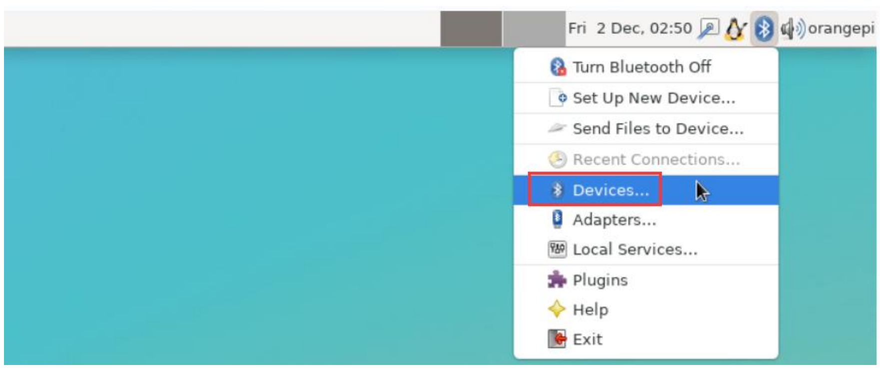
3) If prompted with the following interface, please select **Yes**



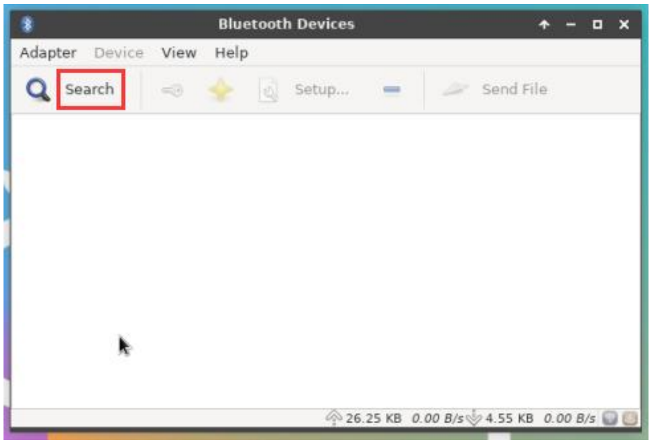
4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter settings interface, and then turn it off



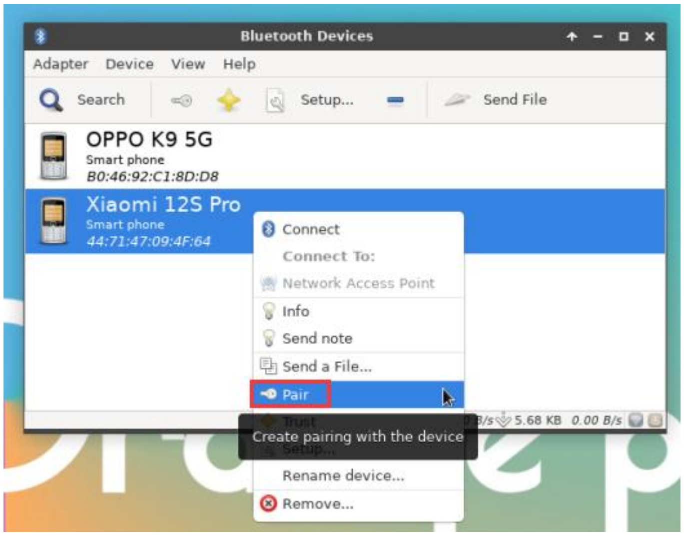
5) Then open the configuration interface of the Bluetooth device



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

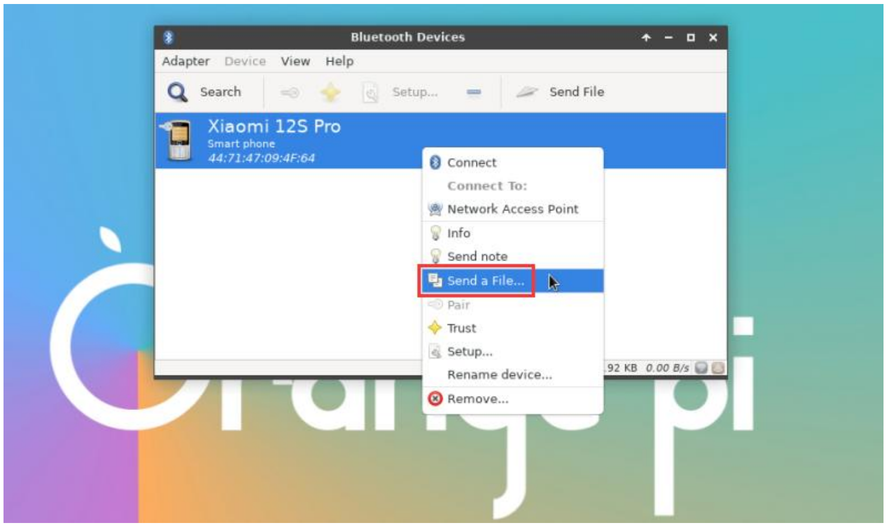


6) Then select the Bluetooth device you want to connect to, and then right-click the mouse. The operation interface for the Bluetooth device will pop up. Select **Pair** to start pairing. The demonstration here is pairing with an Android phone

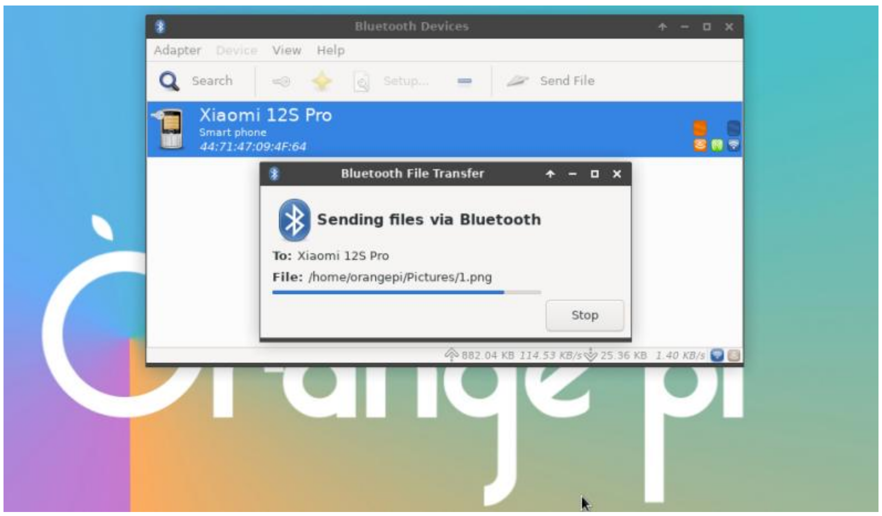


7) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop. Just select **Confirm** to confirm. At this time, you also need to confirm on the mobile phone

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 to test

1) Plug the USB keyboard into the USB port of the Orange Pi development board

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

3) If the mouse or keyboard can operate the operating system 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 for testing

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 the USB disk is successfully recognized

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

3) Use the mount command to mount the 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 check 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. Please test other models of USB wireless network cards by yourself. If it 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) You can see the loading information of the RTL8723BU module through the dmesg command

```
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 you can see the RTL8723BU WIFI device node through the **sudo ifconfig** command. For WIFI connection and test methods, please refer to the **WIFI connection test** section, which will not be described 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)
```

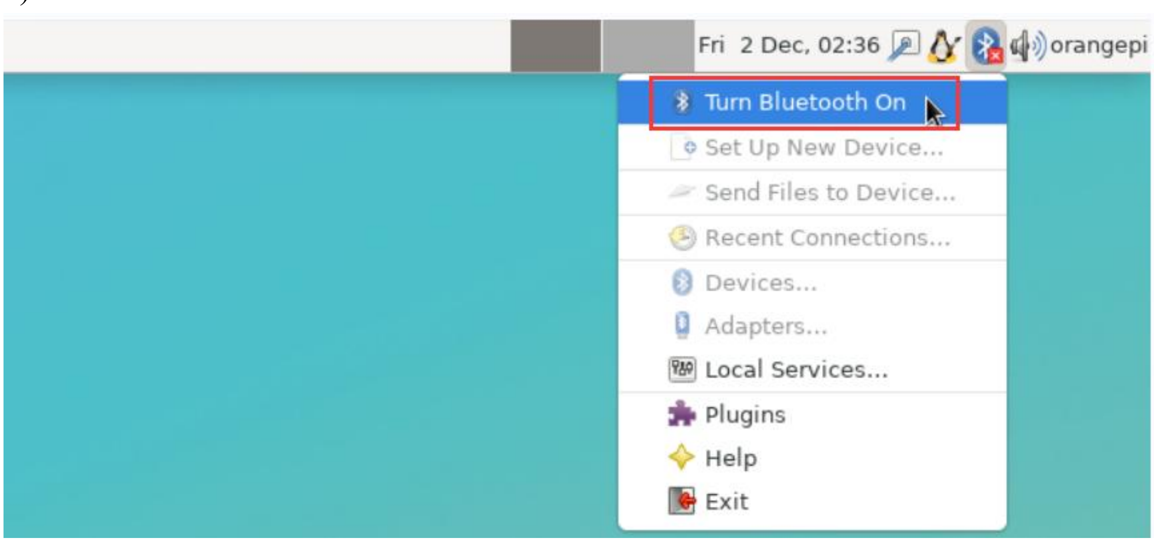
```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5) Then you can see the USB Bluetooth device through the **hciconfig** command

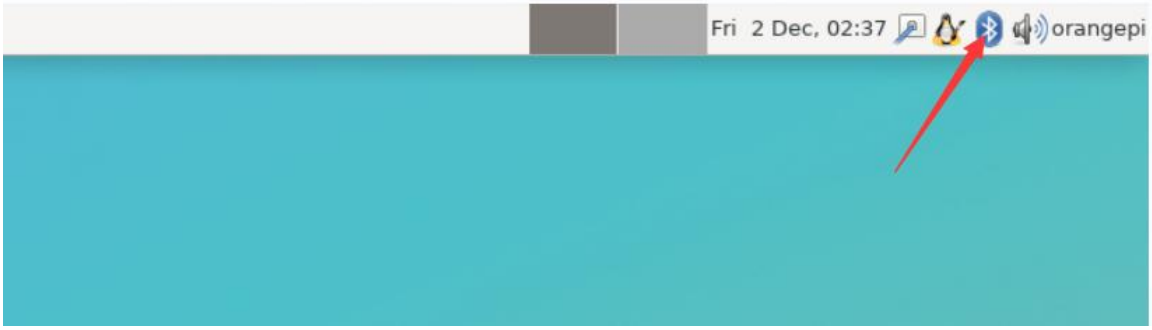
```
orangepi@orangepi:~$ sudo apt update && sudo apt install bluez
orangepi@orangepi:~$ hciconfig
hci0:    Type: Primary  Bus: USB
         BD Address: 00:13:EF:F4:58:AE  ACL MTU: 820:8  SCO MTU: 255:16
         DOWN
         RX bytes:1252 acl:0 sco:0 events:125 errors:0
         TX bytes:23307 acl:0 sco:0 commands:125 errors:0
```

6) You can also see the Bluetooth icon appearing on the desktop. Bluetooth is not turned on at this time, 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 Bluetooth test method, please refer to the section on **Bluetooth usage** and will not be repeated 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 modules related to RTL8811 WIFI. 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
```

3) You can see the loading information of the RTL8811 module through the dmesg command

```
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 WIFI connection and testing methods, please refer to the **WIFI connection test** section. I will not 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 ensure 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 and unplug the USB WIFI module again. If it still doesn't work, please manually execute the following command to switch the 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) You can see the loading information of the rtl8821cu module through the dmesg command

```
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
```

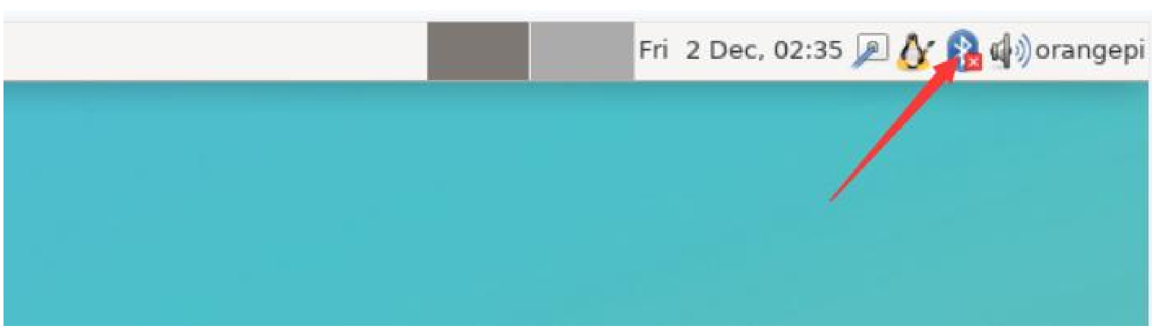
5) Then you can see the device node of rtl8821cu wifi through the **sudo ifconfig** command. For wifi connection and test methods, please refer to the **WIFI connection test** section, which will not be repeated here

```
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
```

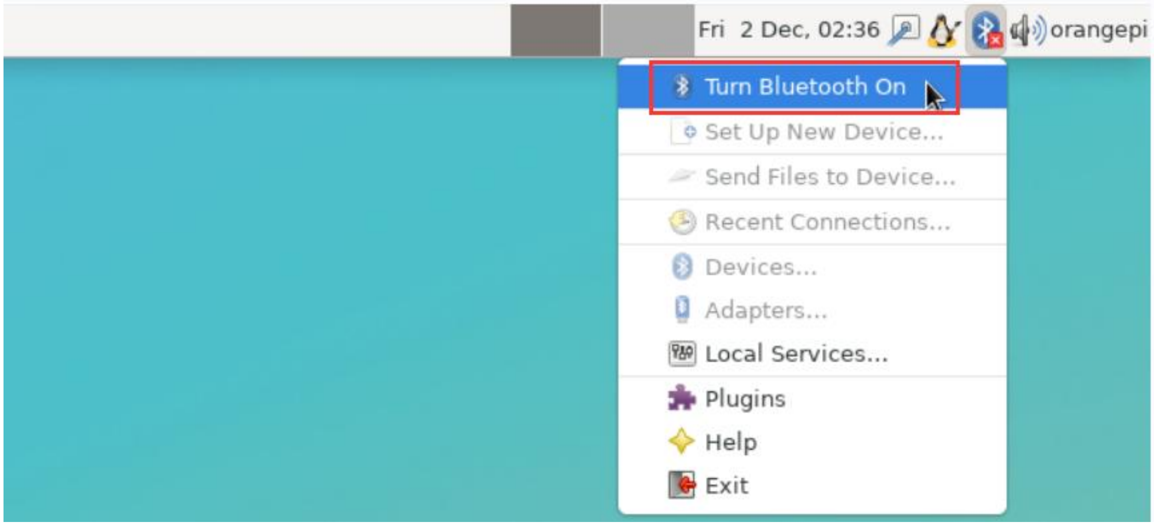
6) Then you can see the USB Bluetooth device through the **hciconfig** command

```
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
```

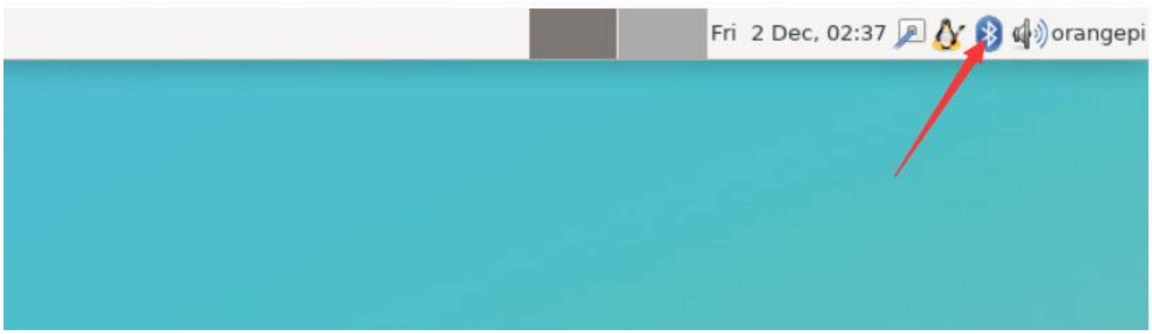
7) You can also see the Bluetooth icon appearing on the desktop. Bluetooth is not turned on at this time, so a red **x** will be displayed



8) Click **Turn Bluetooth On** to turn on Bluetooth



9) The display after turning on Bluetooth is as follows



10) For the Bluetooth test method, please refer to the **Bluetooth usage section** and will not be repeated here

3. 12. 4. USB camera test

1) First, you need to prepare a USB camera as shown in the picture below or similar that supports UVC protocol, and then insert the USB camera into the USB interface of the Orange Pi development board

2) Through the v4l2-ctl command, you can see that the device node information of the

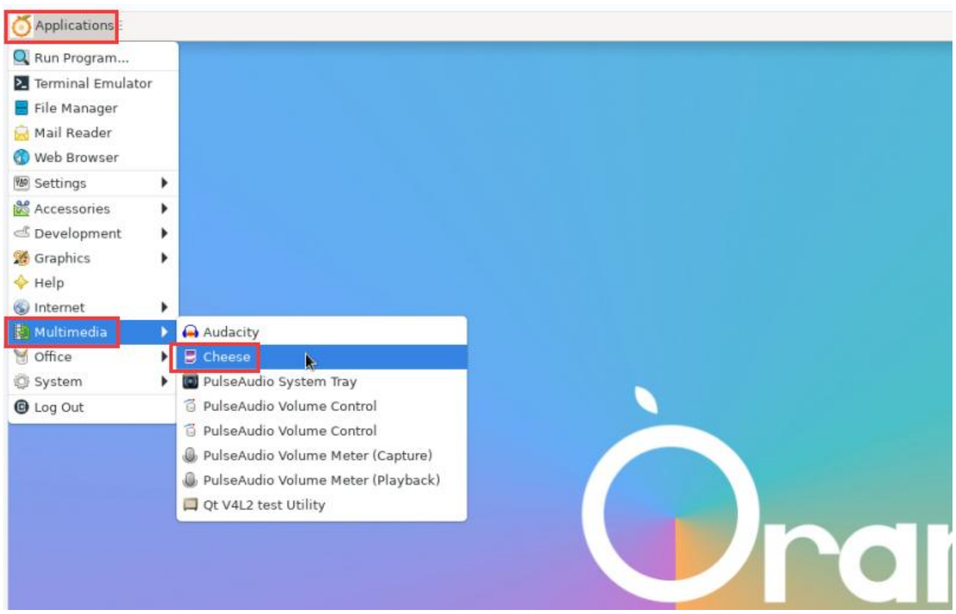
USB camera is /dev/video0

```
orangepi@orangepi:~$ v4l2-ctl --list-devices
Q8 HD Webcam: Q8 HD Webcam (usb-fc880000.usb-1):
        /dev/video0
        /dev/video1
        /dev/media0
```

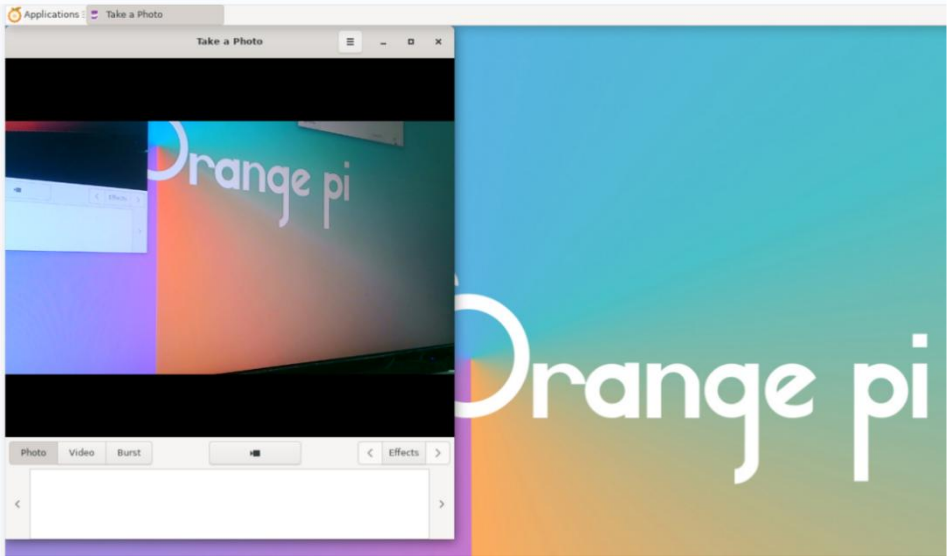
Note that the l in v4l2 is a lowercase letter l, not the number 1

In addition, the serial number of the video may not always be video0, please refer to what you actually see

3) In the desktop system, you can use Cheese to directly open the USB camera. The method of opening Cheese is as shown in the figure below:



The interface after Cheese turns on the USB camera is as shown below:



4) How to use 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) The -d option is used to specify the device node of the USB camera

b) --no-banneris used to remove watermarks from photos

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

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

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

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

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

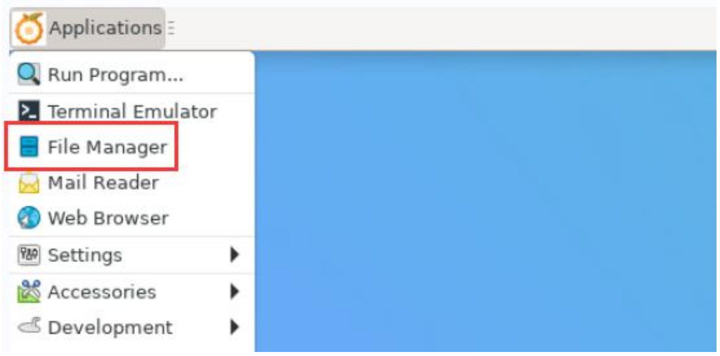
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test  （Modify the IP address and path according to the actual situation）
```

d. In the desktop version of Linux system, the captured pictures can be viewed directly through the HDMI display

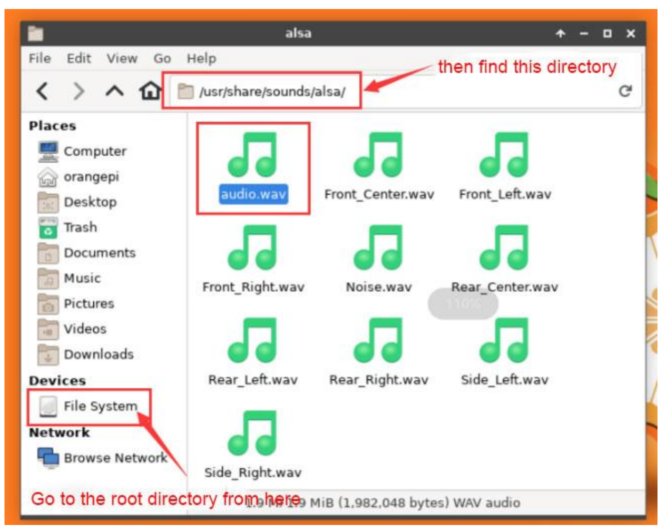
3. 13. Audio test

3. 13. 1. Test audio methods on desktop systems

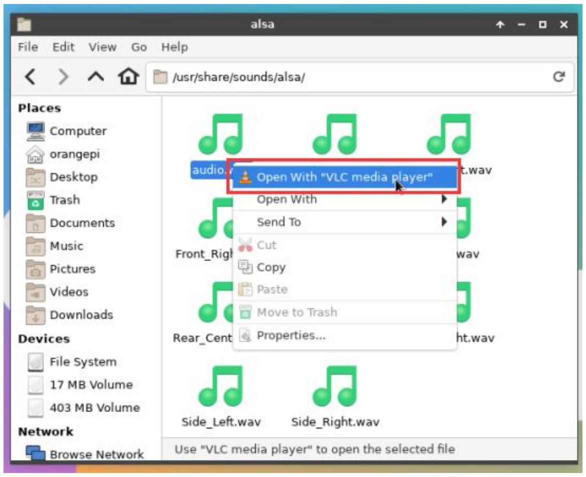
1) First open the file manager



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

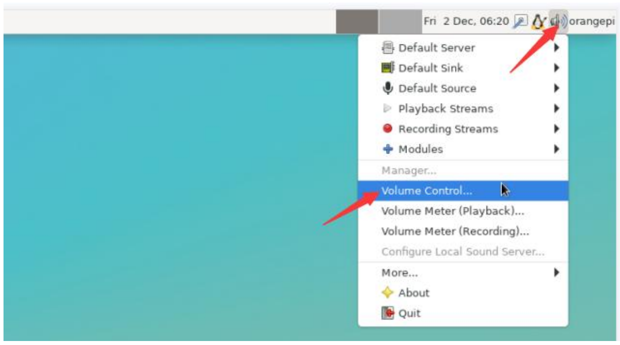


3) Then select the audio.wav file, right-click and select open with vlc to start playing

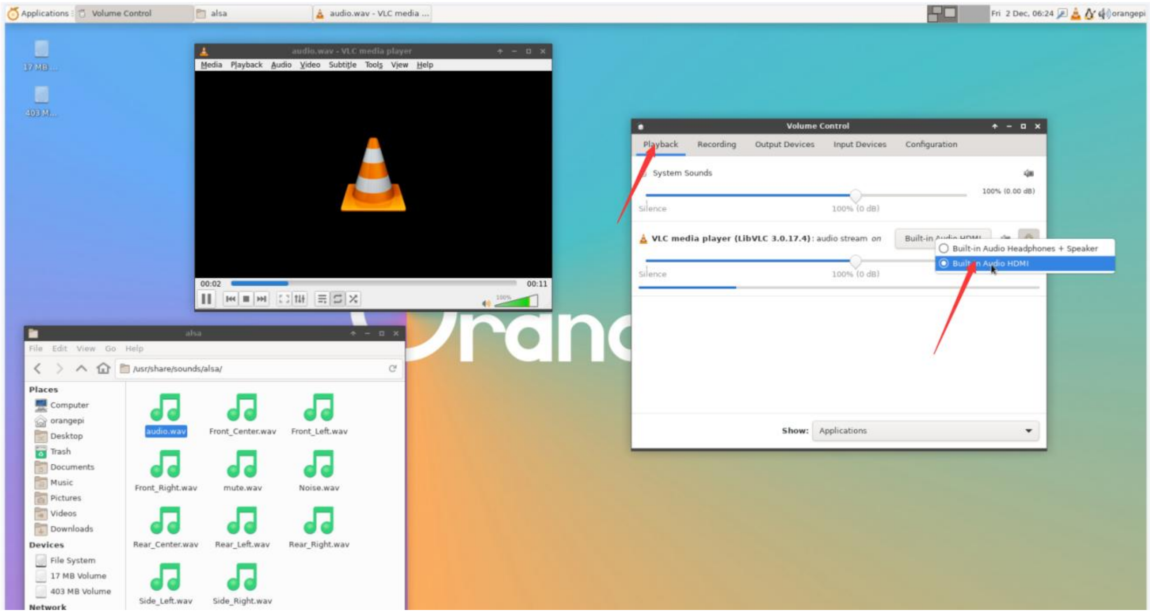


4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First open the volume control interface



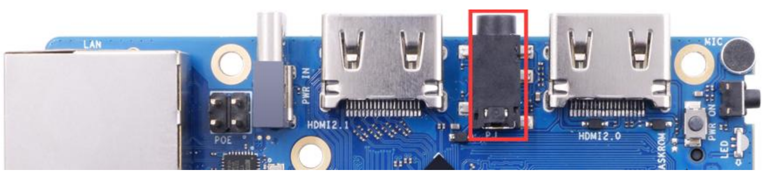
b. When playing audio, the audio device options that can be used by the **playback** software will be displayed in Playback, as shown in the figure below. Here you can set which audio device needs to be played



3. 13. 2. How to use commands to play audio

3. 13. 2. 1. Headphone interface playback audio test

1) First, insert the headphones into the headphone jack of the development board



2) Then you can use the **aplay -l** command to check the sound card devices supported by the Linux system. From the output below, it can be seen that **card 2** is the sound card device of es8388, which is the sound card device of the headset

```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchipdp0 [rockchip,dp0], device 0: rockchip,dp0 spdif-hifi-0 [rockchip,dp0 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip,es8388], device 0: dailink-multicodecs ES8323.3-0010-0 [dailink-multicodecs ES8323.3-0010-0]
```

```
    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 headset can hear the sound, it means 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 an HDMI to HDMI cable to connect the Orange Pi development board to the TV (other HDMI displays need to ensure that they can play audio)

2) Then check the HDMI sound card serial number. From the output below, you can know that the HDMI sound card is **card 0**

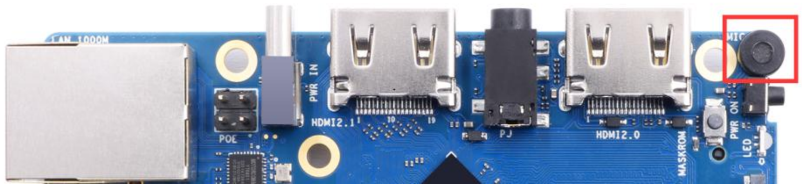
```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
card 1: rockchipdp0 [rockchip,dp0], device 0: rockchip,dp0 spdif-hifi-0 [rockchip,dp0 spdif-hifi-0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip,es8388], device 0: dailink-multicodecs ES8323.3-0010-0 [dailink-multicodecs ES8323.3-0010-0]
    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 sound can be heard on the HDMI display or TV, it means that the hardware can be used normally.

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

3. 13. 3. How to test recording using commands

1) There is an onboard MIC on the development board, and its 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

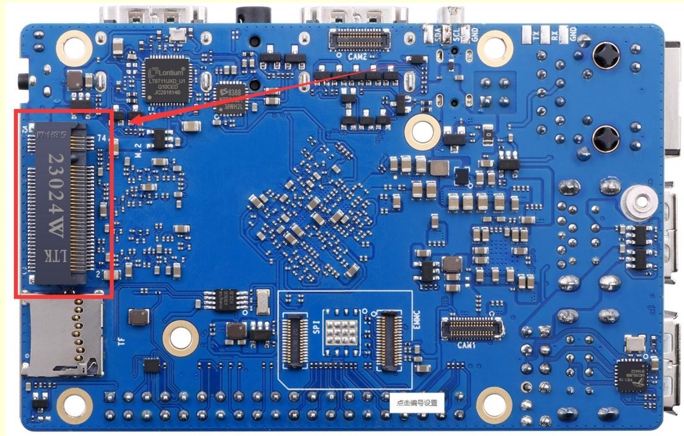
```
orangepi@orangepi:~$ test_record.sh main
Start recording: /tmp/test.wav
Recording WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Start playing
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3) HDMI 和 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, running the **test_record.sh headset** command will record an audio through the headset and then play it to HDMI and 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
```

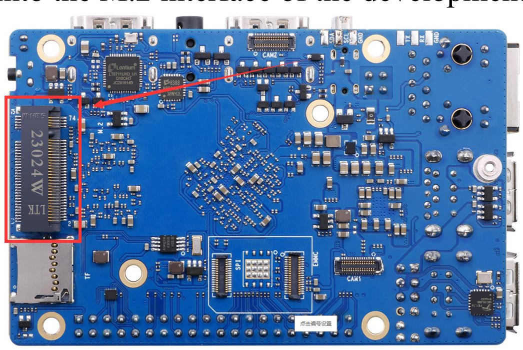
3. 14. How to use SATA SSD

The m.2 interface shown in the figure below can use either nvme ssd or sata ssd. Since the pcie2.0 controller and the sata controller are optional, only one of the configurations can be opened 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

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



3) There are two main ways to use sata ssd:

a. The Linux system is placed in the TF card, and then the SATA SSD is inserted 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 usage, please see **the instructions in the method of burning Linux image into 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/orangepi5pro/u-boot-sata.itb
```

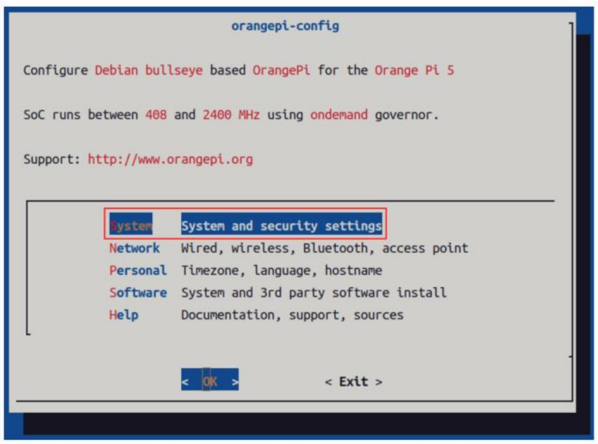
b. Make sure that **u-boot-sata.itb** exists in the Linux system, and then use the following command to burn it to the TF card or eMMC of the development board

```
orangepi@orangepi:~$ cd /usr/share/orangepi5pro/
orangepi@orangepi:~$ sudo dd if=u-boot-sata.itb of=$(findmnt -n -o SOURCE / | sed 's/..$//') seek=16384
orangepi@orangepi:~$ sudo sync
```

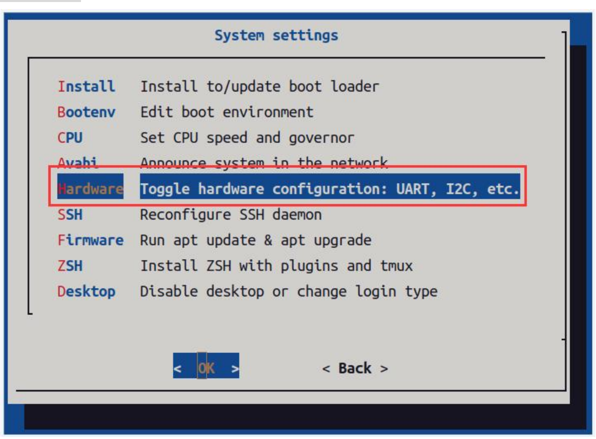
5) Then run **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

6) Then Choose **System**



7) Then Choose **Hardware**

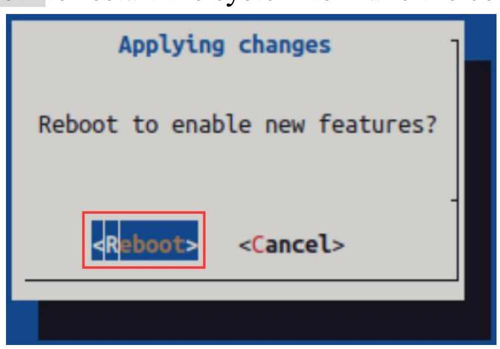


8) Then use the keyboard arrow keys to locate **ssd-sata2**, and then use the **space** to select it

9) Then select **<Save>** to save

10) Then Choose **<Back>**

11) Then Choose **<Reboot>**to restart the system to make the configuration take effect



The above settings will eventually add the overlays=ssd-sata2 line configuration to /boot/orangepiEnv.txt. You can check it first after setting it up. If this line of configuration does not exist, there is something wrong with the settings

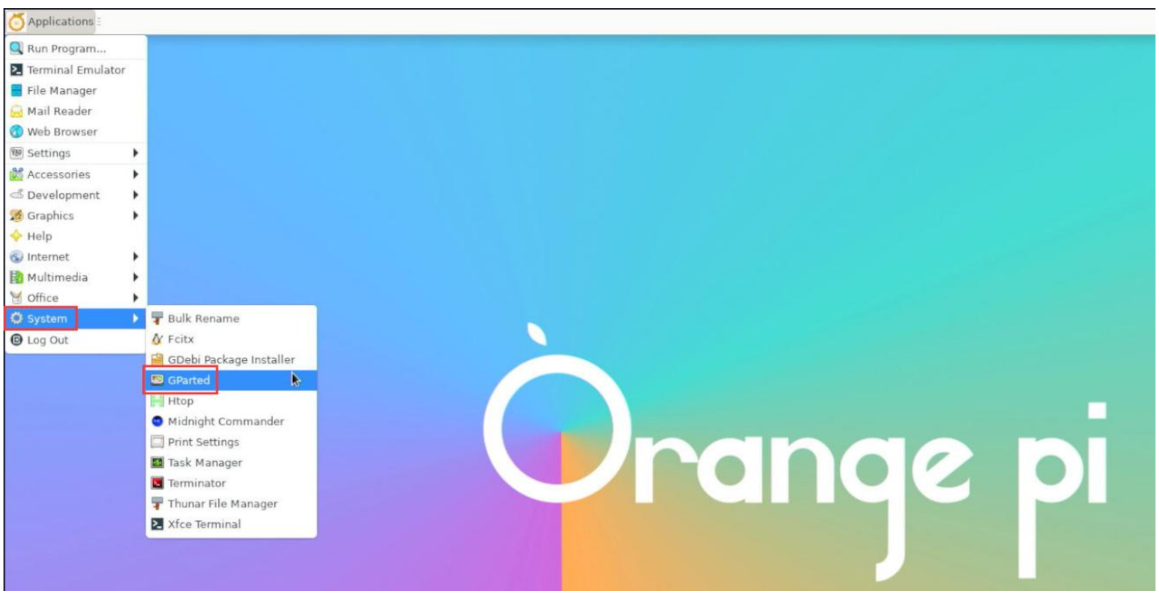
If you find it troublesome to use orangepi-config, you can also open /boot/orangepiEnv.txt and add the line configuration overlays=ssd-sata2

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep ssd**
overlays=ssd-sata2

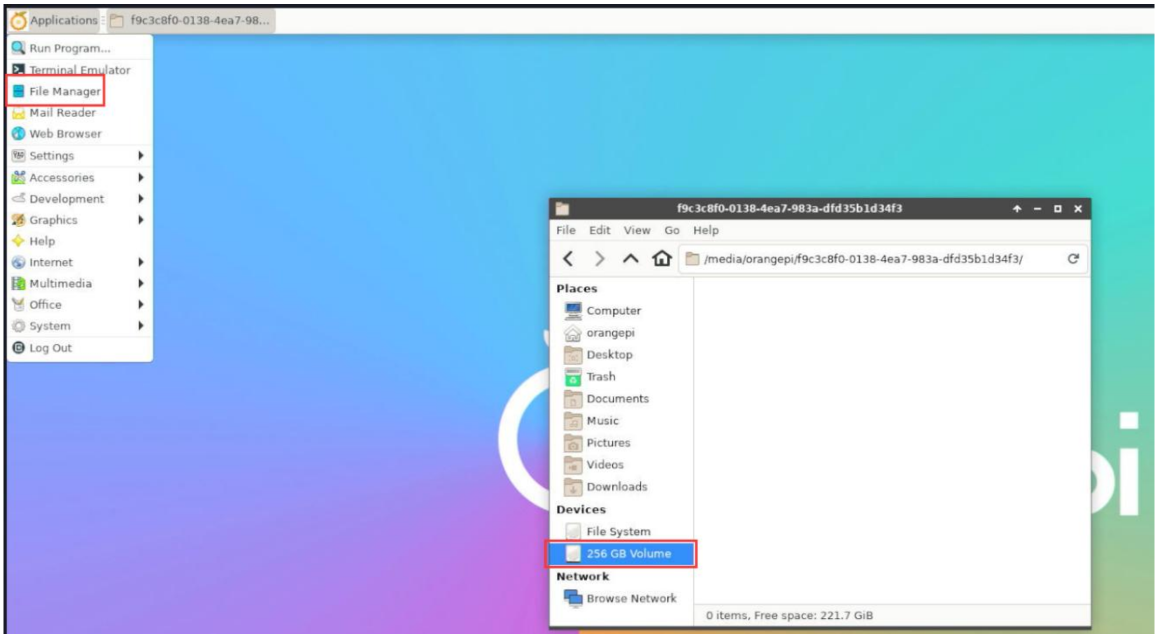
12) If everything is normal, use the **sudo fdisk -l** command after the system restarts to see the sata ssd information

```
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 the sata ssd



14) Then you can see the sata ssd device in the file management



15) In the server version of the 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

1) 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)
```

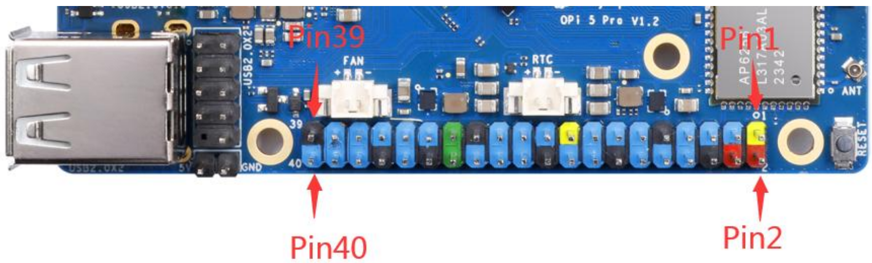
2) The command to check the current temperature of nvme ssd solid state drive is：

```
orangepi@orangepi:~$ sudo smartctl -a /dev/nvme0 | grep "Temperature:"
```

Temperature:	**40 Celsius**

3. 16. 40 Pin interface pin description

1) Please refer to the picture below for the order of the 40-pin interface pins of the Orange Pi 5 Pro development board



2) The functions of the 40-pin interface pins of the Orange Pi 5 Pro development board are as shown in the table below

a. Below is the complete pin diagram of 40pin

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

b. The table below is a 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
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7
			GND		9
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15
			3.3V		17
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19
		SPI0_MISO_M2	GPIO1_B1	41	21
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23
			GND		25
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27
			GPIO1_A4	36	29
			GPIO1_A6	38	31
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33
		I2C5_SDA_M2	GPIO4_A7	135	35
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37
			GND		39

c. The table below is a 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	13	GPIO0_B5	UART2_TX_M0		
10	14	GPIO0_B6	UART2_RX_M0		
12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
14		GND			
16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
20		GND			
22	40	GPIO1_B0			
24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
30		GND			
32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
34		GND			
36	131	GPIO4_A3	UART0_TX_M2		
38	132	GPIO4_A4	UART0_RX_M2		
40	133	GPIO4_A5		UART3_TX_M2	

In the above table, pwm 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 40pin header.。

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

3. 17. How to install wiringOP

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

The storage path of the compiled wiringOP deb package in orangepi-build is:

orangepi-build/external/cache/debs/arm64/wiringpi_x.xx.deb

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

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   46 |   8 | GPIO1_B6 |     IN | 1 | 15 || 16 | 0 | IN     | TXD.6    | 9   | 33   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | RXD.6    | 10  | 32   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | GPIO1_B0 | 13  | 40   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 |   34 |  17 |    SDA.4 |     IN | 0 | 27 || 28 | 0 | IN     | SCL.4    | 18  | 35   |
 |   36 |  19 | GPIO1_A4 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |   38 |  20 | GPIO1_A6 |     IN | 0 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |   63 |  22 |    PWM15 |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  135 |  23 | GPIO4_A7 |     IN | 0 | 35 || 36 | 0 | IN     | TXD.0    | 24  | 131  |
 |  134 |  25 | GPIO4_A6 |     IN | 0 | 37 || 38 | 0 | IN     | RXD.0    | 26  | 132  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO4_A5 | 27  | 133  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
orangepi@orangepi5pro:~$
```

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 Pro 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) The output of testing the gpio readall command is as follows

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   46 |   8 | GPIO1_B6 |     IN | 1 | 15 || 16 | 0 | IN     | TXD.6    | 9   | 33   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | RXD.6    | 10  | 32   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | GPIO1_B0 | 13  | 40   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 |   34 |  17 |    SDA.4 |     IN | 0 | 27 || 28 | 0 | IN     | SCL.4    | 18  | 35   |
 |   36 |  19 | GPIO1_A4 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |   38 |  20 | GPIO1_A6 |     IN | 0 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |   63 |  22 |    PWM15 |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  135 |  23 | GPIO4_A7 |     IN | 0 | 35 || 36 | 0 | IN     | TXD.0    | 24  | 131  |
 |  134 |  25 | GPIO4_A6 |     IN | 0 | 37 || 38 | 0 | IN     | RXD.0    | 26  | 132  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO4_A5 | 27  | 133  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
orangepi@orangepi5pro:~$
```

3.18. 40pin interface GPIO, I2C, UART, SPI, CAN and PWM testing

3.18.1. 40pin GPIO port test

The Linux system released by Orange Pi comes pre-installed with a blink_all_gpio program. This program will set all 28 GPIO ports in 40pin to

continuously switch between high and low levels.

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 continuously switch between 0 and 3.3v. Using this program we can test whether the GPIO port can work normally

The way to run the blink_all_gpio program is as follows:

orangepi@orangepi5pro:~$ **sudo blink_all_gpio** #Remember to add sudo permissions
[sudo] password for orangepi: #You need to enter your password here

1) There are a total of 28 GPIO ports in the 40-pin development board that can be used. The following uses pin No. 7 - the corresponding GPIO is GPIO1_B7 - the corresponding wPi serial number is 2 - as an example to demonstrate how to set the high and low levels of the GPIO port

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

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

root@orangepi:~/wiringOP# **gpio mode 2 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 the low level is set successfully

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

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

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |    OUT | 0 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

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 the setting of the high level is successful.

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

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

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |    OUT | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

5) The setting method for other pins is similar. You only need to modify the serial number of wPi to the serial number corresponding to the pin

3. 18. 2. How to set the pull-down resistor of the 40pin GPIO port

Note that the four GPIO pins below Orange Pi 5 Pro have external 3.3V pull-ups, so the pull-down settings are invalid

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   46 |   8 | GPIO1_B6 |     IN | 1 | 15 || 16 | 0 | IN     | TXD.6    | 9   | 33   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | RXD.6    | 10  | 32   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | GPIO1_B0 | 13  | 40   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 |   34 |  17 |    SDA.4 |     IN | 0 | 27 || 28 | 0 | IN     | SCL.4    | 18  | 35   |
 |   36 |  19 | GPIO1_A4 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |   38 |  20 | GPIO1_A6 |     IN | 0 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |   63 |  22 |    PWM15 |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  135 |  23 | GPIO4_A7 |     IN | 0 | 35 || 36 | 0 | IN     | TXD.0    | 24  | 131  |
 |  134 |  25 | GPIO4_A6 |     IN | 0 | 37 || 38 | 0 | IN     | RXD.0    | 26  | 132  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO4_A5 | 27  | 133  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
orangepi@orangepi5pro:~$
```

1) The following uses pin No. 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 resistors of the GPIO port

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |    OUT | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |    OUT | 0 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |    OUT | 0 |  7 || 8  | 0 | OUT    | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OUT    | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |    OUT | 0 | 11 || 12 | 0 | OUT    | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |    OUT | 0 | 13 || 14 |   |        | GND      |     |      |
```

2) First, you need to set the GPIO port to 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, it means that the pull-down mode is set successfully

```
root@orangepi:~/wiringOP# gpio read 5
0
```

3. 18. 3. 40pin SPI Test

1) As can be seen from the picture below, the spi available for Orange Pi 5 Pro is spi0 and spi4

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of SPI0 and SPI4 in 40pin are as shown in the table below

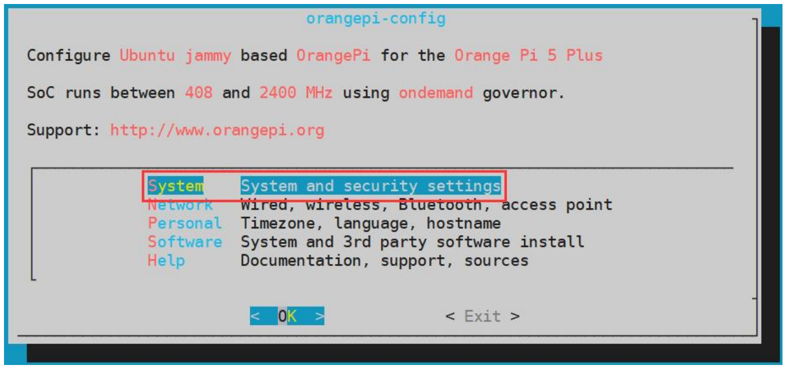
	SPI0_M2 correspond 40pin	**SPI4_M2 correspond 40pin**
MOSI	**Pin 19**	**Pin 16**
MISO	**Pin 21**	**Pin 18**
CLK	**Pin 23**	**Pin 27**
CS0	**Pin 24**	**Pin 28**
CS1	**Pin 26**	**None**
Dtbo Configuration	**spi0-m2-cs0-spidev** **spi0-m2-cs1-spidev** **spi0-m2-cs0-cs1-spidev**	**spi4-m2-cs0-spidev**

3) In Linux systems, the SPI in 40 pin is turned off by default and needs to be turned on manually to use it. Detailed steps are as follows：

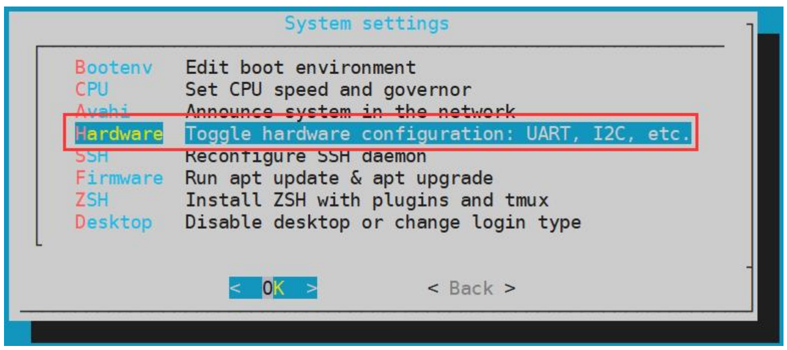
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

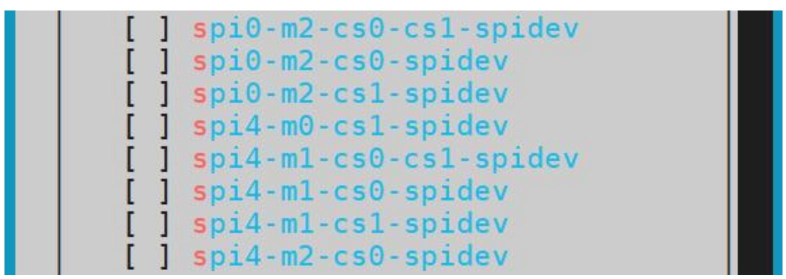
b. Then Select **System**



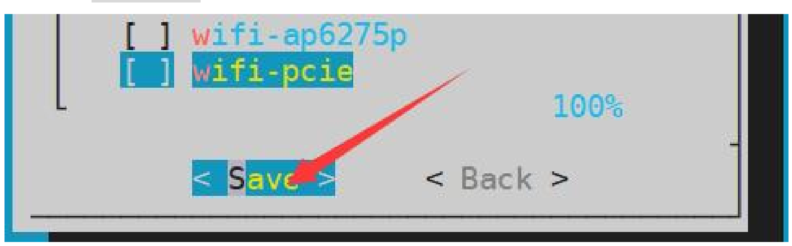
c. Then Select **Hardware**



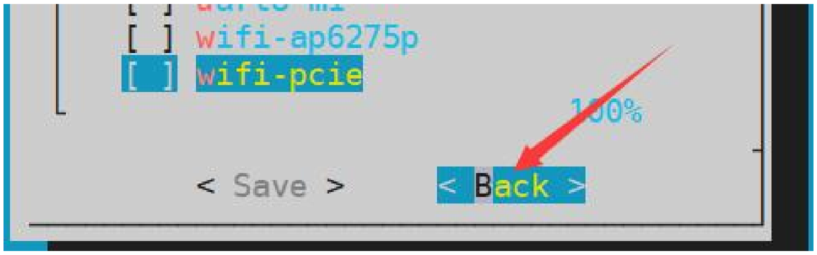
d. Then use the keyboard' s arrow keys to locate the position shown in the picture below, and then use the **space** to select the SPI configuration you want to open



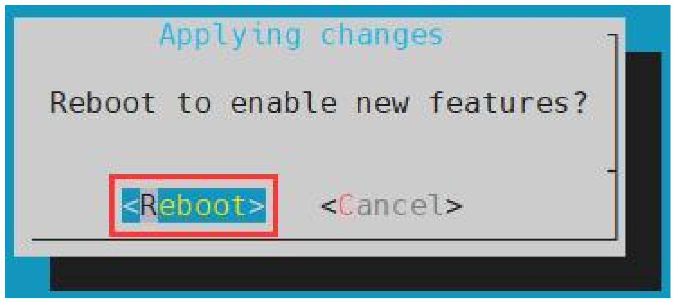
e. Then Select **<Save>** to save



f. Then select **<Back>**



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



4) After restarting, enter the system and check whether the device node of **spidevx.x** exists in the Linux system. If it exists, it means that the SPI has been set up and can be used directly

```
orangepi@orangepi:~$ ls /dev/spidev*
/dev/spidev0.0    /dev/spidev0.1    /dev/spidev4.0
```

The above is the result displayed after opening spi0-m2-cs0-cs1-spidev and spi4-m2-cs0-spidev

5) Without shorting the mosi and miso pins of SPI0 or 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/spidev4.0
Or
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev0.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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   | ................................
```

6) Then short-circuit the mosi and miso pins of SPI0 or SPI4 and then run spidev_test. The output is as follows. You can see that the data sent and received are the same.

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev4.0
Or
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev0.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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. 40pin I2C test

1) As can be seen from the table below, the i2c available for Orange Pi 5 Pro is i2c1, i2c4, i2c5 and i2c8, a total of four groups 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			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of the 4 sets of I2C buses in 40pin are as shown in the table below. I2C5_M2 and I2C5_M3 can only use one of them at the same time, not at the same time. They are the same I2C5, just connected to different pins. Please do not think that they are two different sets of I2C5 buses

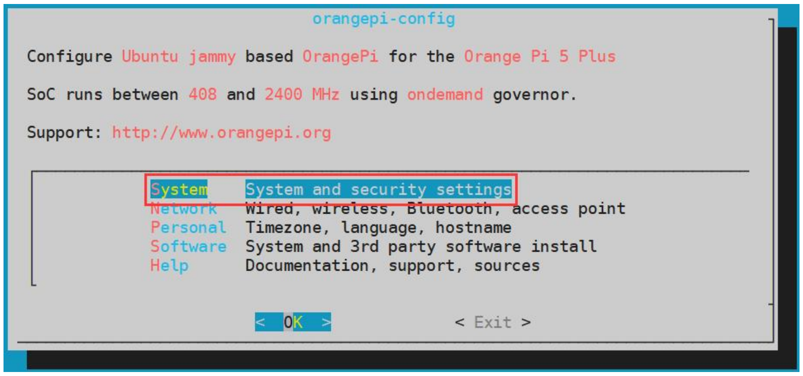
I2C bus	SDA correspond 40pin	SCL correspond 40pin	DtboCorresponding configuration
I2C1_M4	**Pin 3**	**Pin 5**	**i2c1-m4**
I2C4_M3	**Pin 27**	**Pin 28**	**i2c4-m3**
I2C5_M2	**Pin 35**	**Pin 37**	**i2c5-m2**
I2C5_M3	**Pin 7**	**Pin 15**	**i2c5-m3**
I2C8_M2	**Pin 33**	**Pin 32**	**i2c8-m2**

3) In Linux systems, the I2C bus in 40 pin is closed by default and needs to be manually opened before it can be used. Detailed steps are as follows:

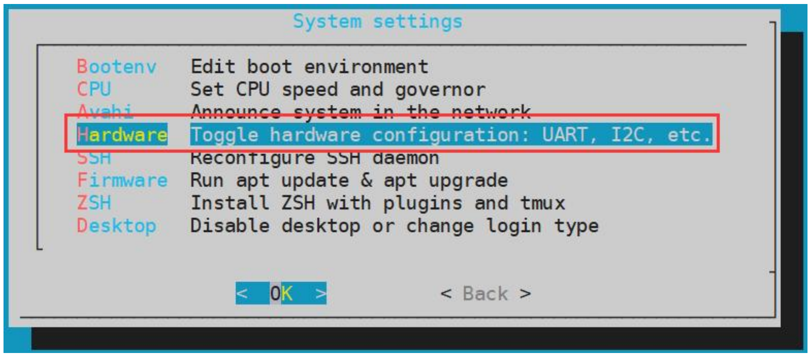
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

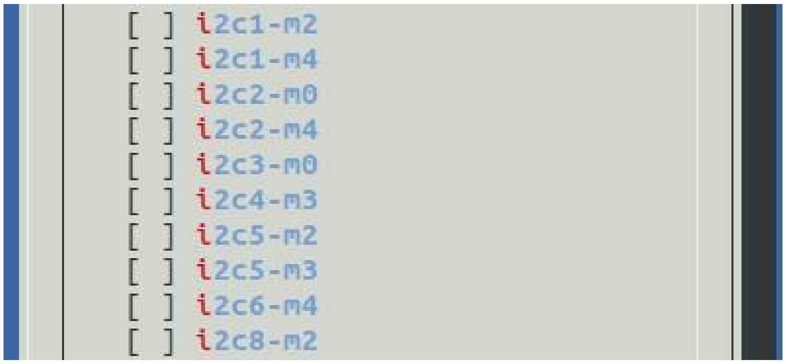
b. Then Select **System**



c. Then Selct **Hardware**

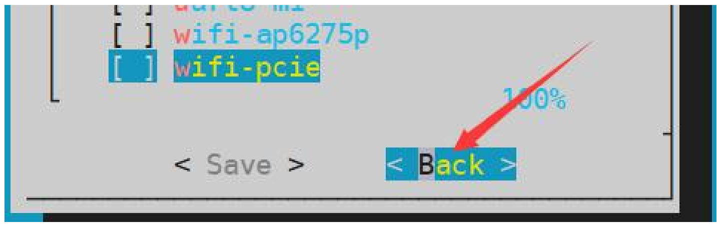


d. Then use the keyboard's arrow keys to locate the position shown in the picture below, and then use the space to select the I2C configuration you want to open

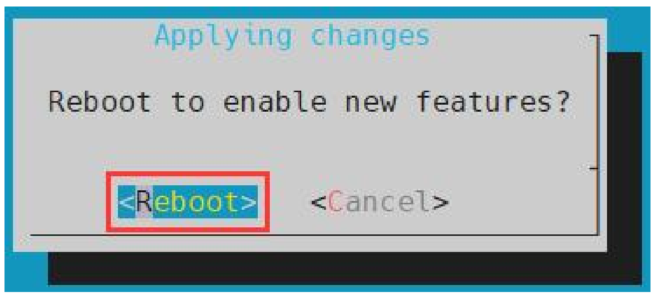


e. Then Select **<Save>** to save

f. Then Select **<Back>**



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



4) After starting the Linux system, first confirm that there are device nodes that need to use I2C under **/dev**

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

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

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

```
orangepi@orangepi:~$ sudo i2cdetect -y 1       #i2c1 command
orangepi@orangepi:~$ sudo i2cdetect -y 4       #i2c4 commandc
orangepi@orangepi:~$ sudo i2cdetect -y 5       #i2c5 command
orangepi@orangepi:~$ sudo i2cdetect -y 8       #i2c8 command
```

```
root@orangepi5pro:~# i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@orangepi5pro:~#
```

3. 18. 5. 40pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5 Pro are uart0, uart3, uart4, uart6 and uart7, a total of five groups 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			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of the five sets of UART buses in 40pin are as shown in the table below. UART4_M0 and UART4_M2 can only use one of them at the same time, not at the same time. They are the same UART4, just connected to different pins. Please do not think that they are two different sets of UART4 buses

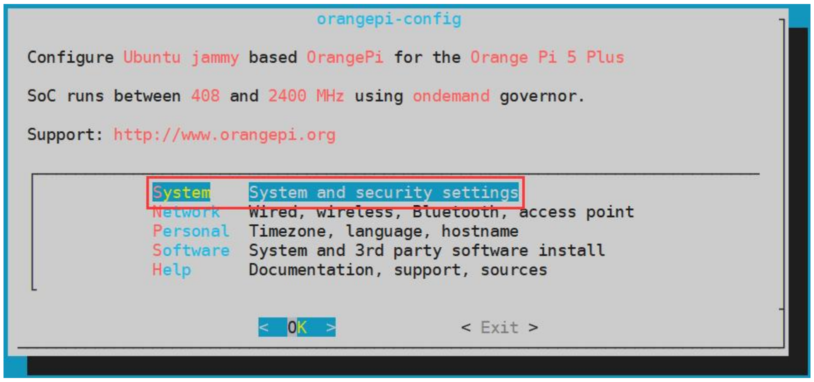
UART Bus	RXcorrespond40pin	TX correspond40pin	dtbo Corresponding configuration
UART0_M2	**Pin 38**	**Pin 36**	**uart0-m2**
UART3_M2	**Pin 37**	**Pin 40**	**uart3-m2**
UART4_M0	**Pin 3**	**Pin 5**	**uart4-m0**
UART4_M2	**Pin 19**	**Pin 23**	**uart4-m2**
UART6_M1	**Pin 18**	**Pin 16**	**uart6-m1**
UART7_M2	**Pin 24**	**Pin 26**	**uart7-m2**

3) In Linux systems, the UART in 40 pin is turned off by default and needs to be turned on manually before it can be used. Detailed steps are as follows:

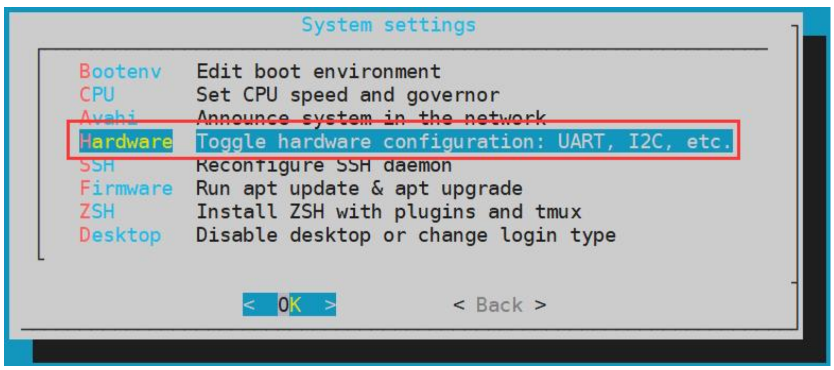
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

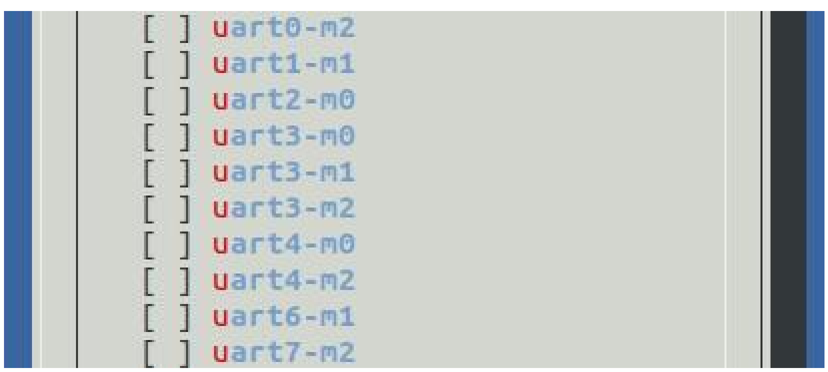
b. Then Select **System**



c. Then Select **Hardware**

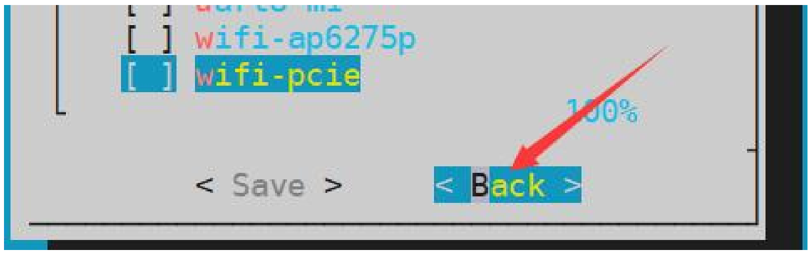


d. Then use the keyboard’ s arrow keys to locate the position shown in the picture below, and then use the **space** to select the UART configuration you want to open

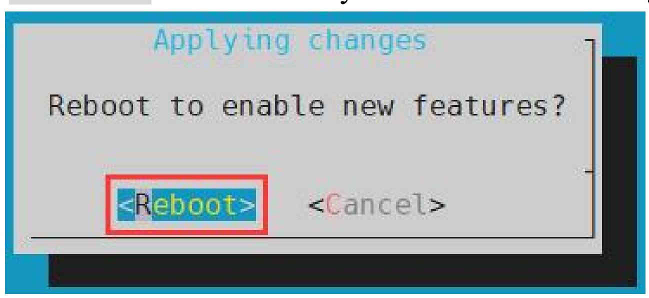


e. Then Select **<Save>** to save

f. Then Select **<Back>**



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



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

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

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

6) Use the **gpio serial** command to test the loopback function of the serial port as shown below. If you can see the following print, it means the serial port communication is normal (ttySX needs to be replaced with the node name corresponding to the uart, please do not copy it)

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttySX
[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 pwms for Orange Pi 5 Pro include pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15, a total of six pwms

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of PWM in 40pin are as shown in the table below. PWM14_M1 and PWM14_M2, PWM15_M1 and PWM15_M3 can only use one of them at the same time, not at the same time. They are all the same PWM, just connected to different pins. Please do not think that they are two different PWM buses

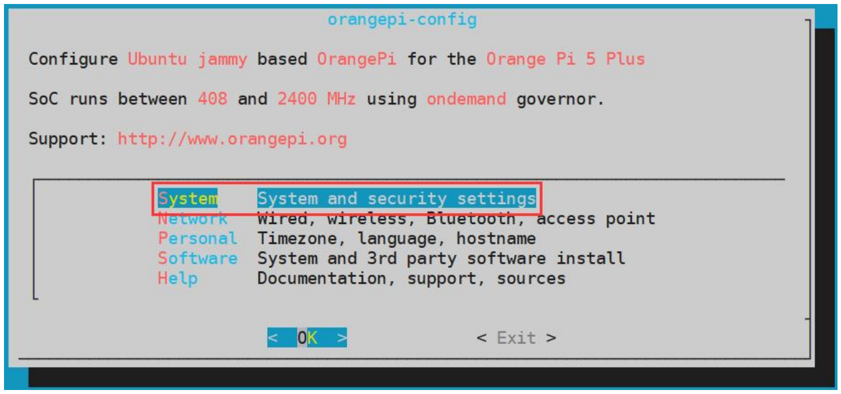
PWM Bus	correspond 40pin	DtboCorresponding configuration
PWM0_M2	**Pin 27**	**pwm0-m2**
PWM1_M2	**Pin 28**	**pwm1-m2**
PWM3_M3	**Pin 12**	**pwm3-m3**
PWM13_M2	**Pin 7**	**pwm13-m2**
PWM14_M1	**Pin 11**	**pwm14-m1**
PWM14_M2	**Pin 32**	**pwm14-m2**
PWM15_M1	**Pin 13**	**pwm15-m1**
PWM15_M3	**Pin 33**	**pwm15-m3**

3) In Linux systems, the PWM in 40 pin is turned off by default and needs to be turned on manually to use it. Detailed steps are as follows:

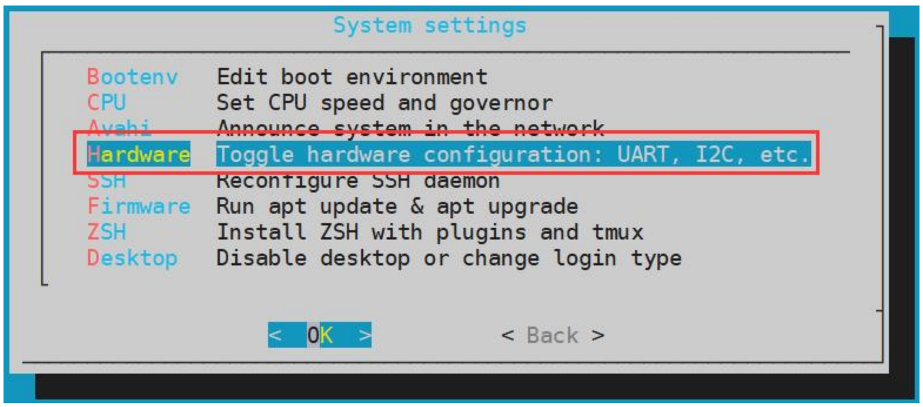
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions.

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

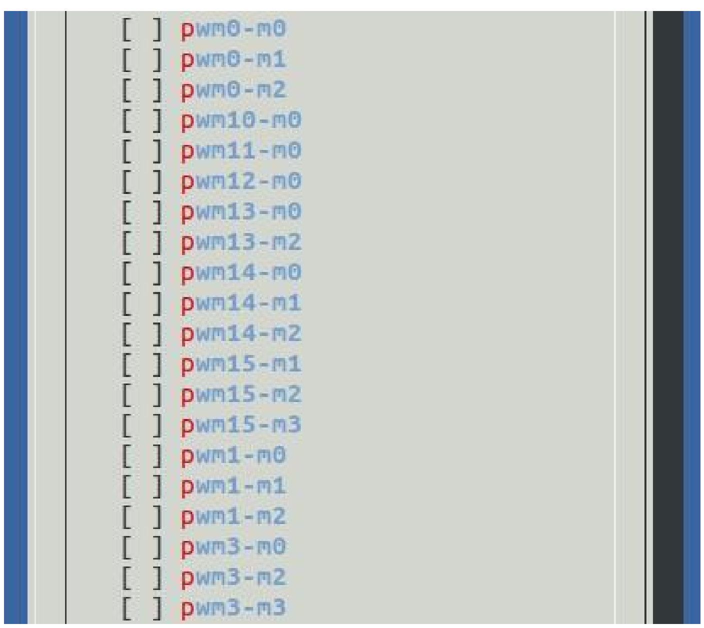
b. Then Select **System**



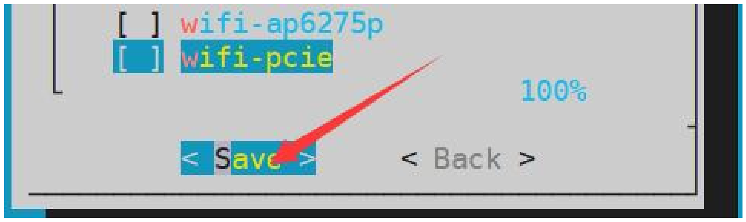
c. Then Select **Hardware**



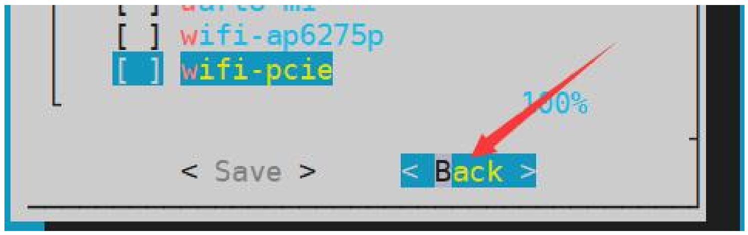
d. Then use the keyboard’ s arrow keys to locate the position shown in the picture below, and then use the **space** to select the configuration of the PWM you want to open



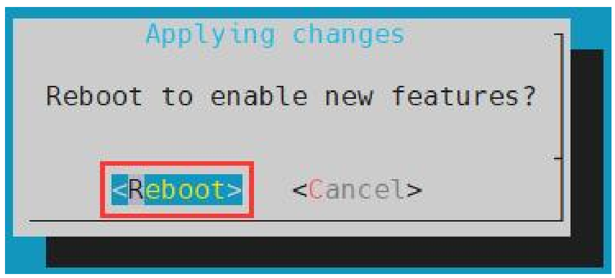
e. Then Select **<Save>** to save



f. Then Select **<Back>**



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



4) When a pwm is opened, there will be an additional pwmchipX (X is a specific number) in **/sys/class/pwm/**. For example, after opening pwm13, the pwmchipX under **/sys/class/pwm/** will be Two became three

```
orangepi@orangepi:~$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

5) Which pwmchip above corresponds to pwm13? Let’s first check the output of the **ls /sys/class/pwm/ -l** command, as shown below:

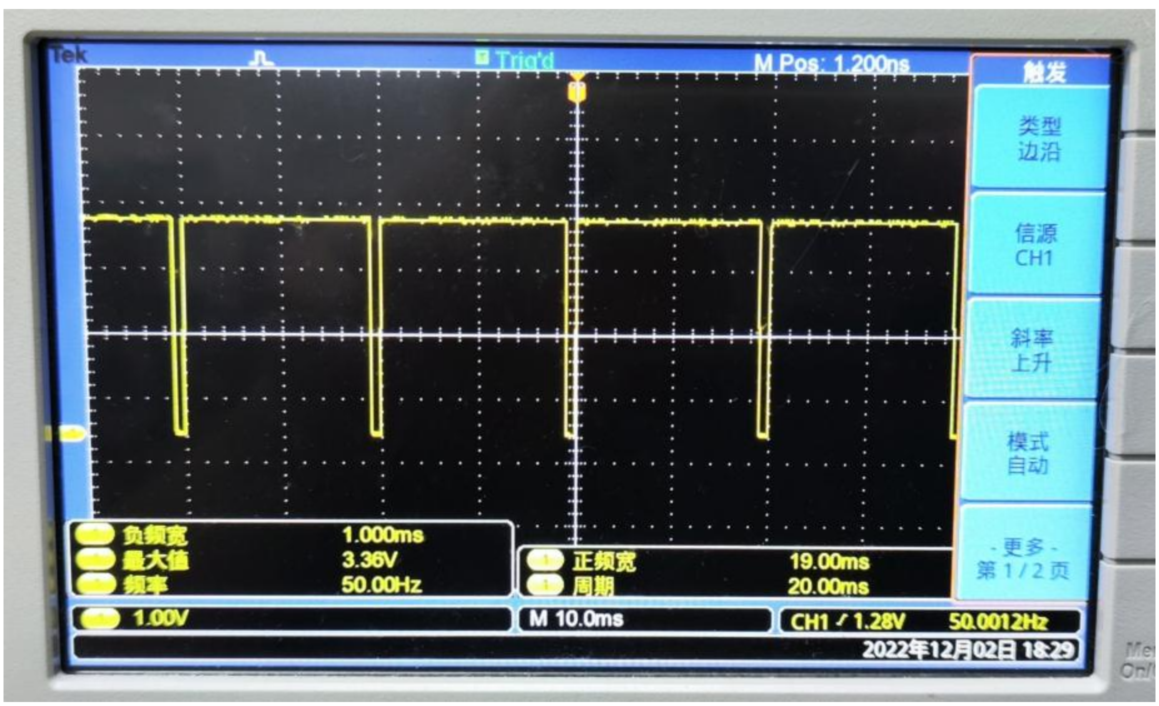
```
orangepi@orangepi5pro:~$ ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Dec 22 18:12 pwmchip0 -> ../../devices/platform/fd8b0020.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Dec 22 18:12 pwmchip1 -> ../../devices/platform/febd0020.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Dec 22 18:12 pwmchip2 -> ../../devices/platform/febf0010.pwm/pwm/pwmchip2
orangepi@orangepi5pro:~$ _
```

6) From the following table, we can see that the base address of the pwm13 register is febf0010. Looking at the output of the **ls /sys/class/pwm/ -l** command, we can see that pwmchip2 is linked to febf0010.pwm, so the corresponding pwmchip of pwm13 is pwmchip2

复用功能	复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
				3.3V		1
		UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3
		UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5
	UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7
				GND		9
I2C7_SCL_M3		PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11
I2C7_SDA_M3		PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13
	UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15
				3.3V		17
		UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19
			SPI0_MISO_M2	GPIO1_B1	41	21
		UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23
				GND		25
UART6_RTSN_M1	SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27
				GPIO1_A4	36	29
				GPIO1_A6	38	31
UART1_CTSN_M1		I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33
SPI2_CS0_M1			I2C5_SDA_M2	GPIO4_A7	135	35
SPI2_CLK_M1		UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37
				GND		39

7) Then use the following command to make pwm13 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
```



8) The test method of pwm13 demonstrated above is similar to other pwm test methods.

3. 18. 7. CAN test method

3. 18. 7. 1. How to open CAN

1) As can be seen from the table below, the CAN bus available for Orange Pi 5 Pro is CAN1

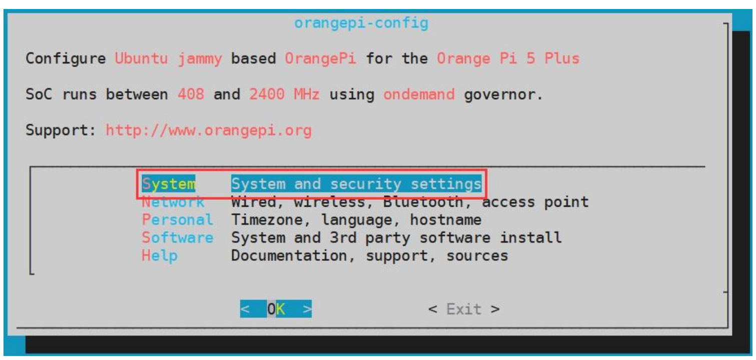
复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) In the Linux system, the CAN in 40 pin is closed by default and needs to be manually opened before it can be used. Detailed steps are as follows:

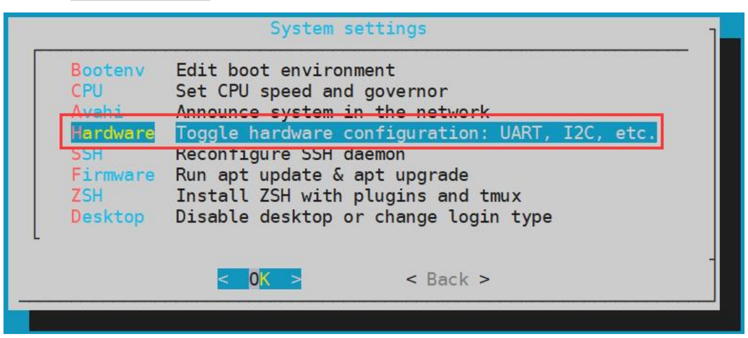
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions.

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

b. Then Select **System**

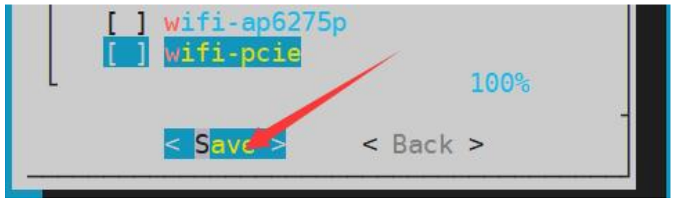


c. Then Select **Hardware**

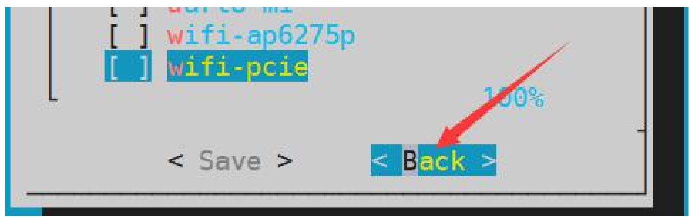


d. Then use the keyboard's arrow keys to locate the position shown in the picture below, and then use the space to select the configuration of CAN1

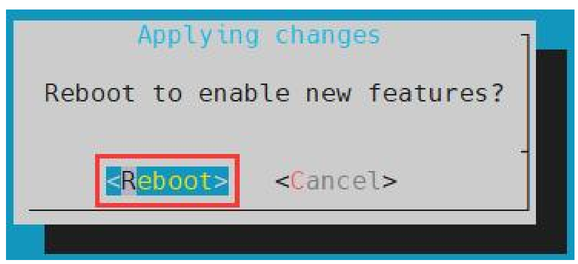
e. Then select **<Save>** to save



f. Then select **<Back>**



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



3) After entering the Linux system, use the **sudo ifconfig -a** command. If you can see the CAN device, it means that 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
```

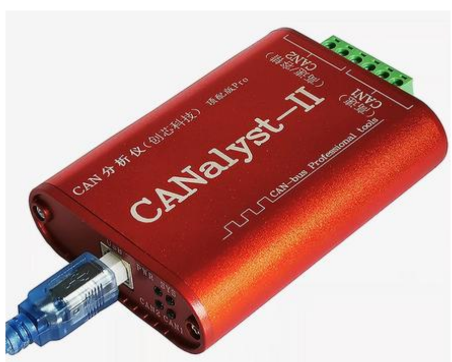
```
can1: flags=128<NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 92
```

4) The corresponding pin of CAN1 is

	CAN1
TX Pin	**Correspond Pin 13**
RX Pin	**Correspond Pin 11**

3.18.7.2. Use 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 USBCANToolSetup software

4) The shortcut after installation of USBCANToolSetup is

5) You also need to install the USB driver

6) 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 also 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 40pin development board
b. The GND pin of the CAN transceiver needs to be connected to the GND pin in the 40pin development board
c. The GND pin of the CAN transceiver needs to be connected to the GND pin in

the 40pin development board

d. The CAN RX pin of the CAN transceiver needs to be connected to the RX pin of the CAN bus in the 40pin development board

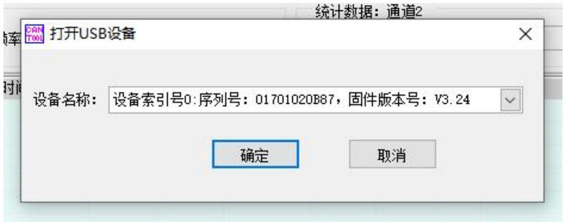
e. The CANH pin of the CAN transceiver needs to be connected to the H interface of the analyzer

f. The CANL pin of the CAN transceiver needs to be connected to the L interface of the analyzer

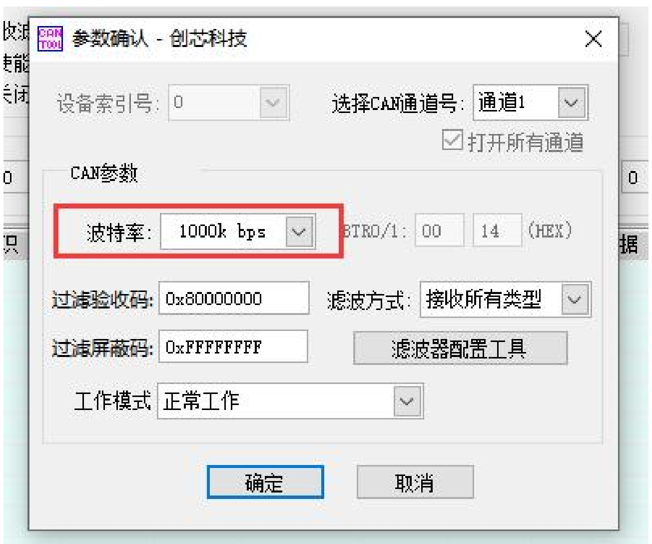
8) Then you can open the USB-CAN software

9) Then click Start 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

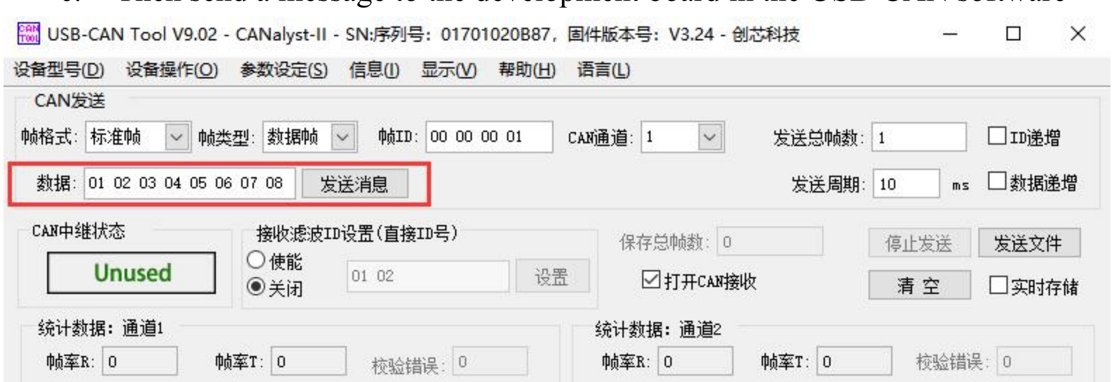
a. First set the baud rate of the CAN bus to **1000kbps** in the Linux system of the development board

```
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 development board can receive messages sent by the analyzer, it means that the CAN bus can be used normally

```
orangepi@orangepi5pro:~$ sudo candump can0
  can0  001   [8]  01 02 03 04 05 06 07 08
```

14) Development board sends CAN message test

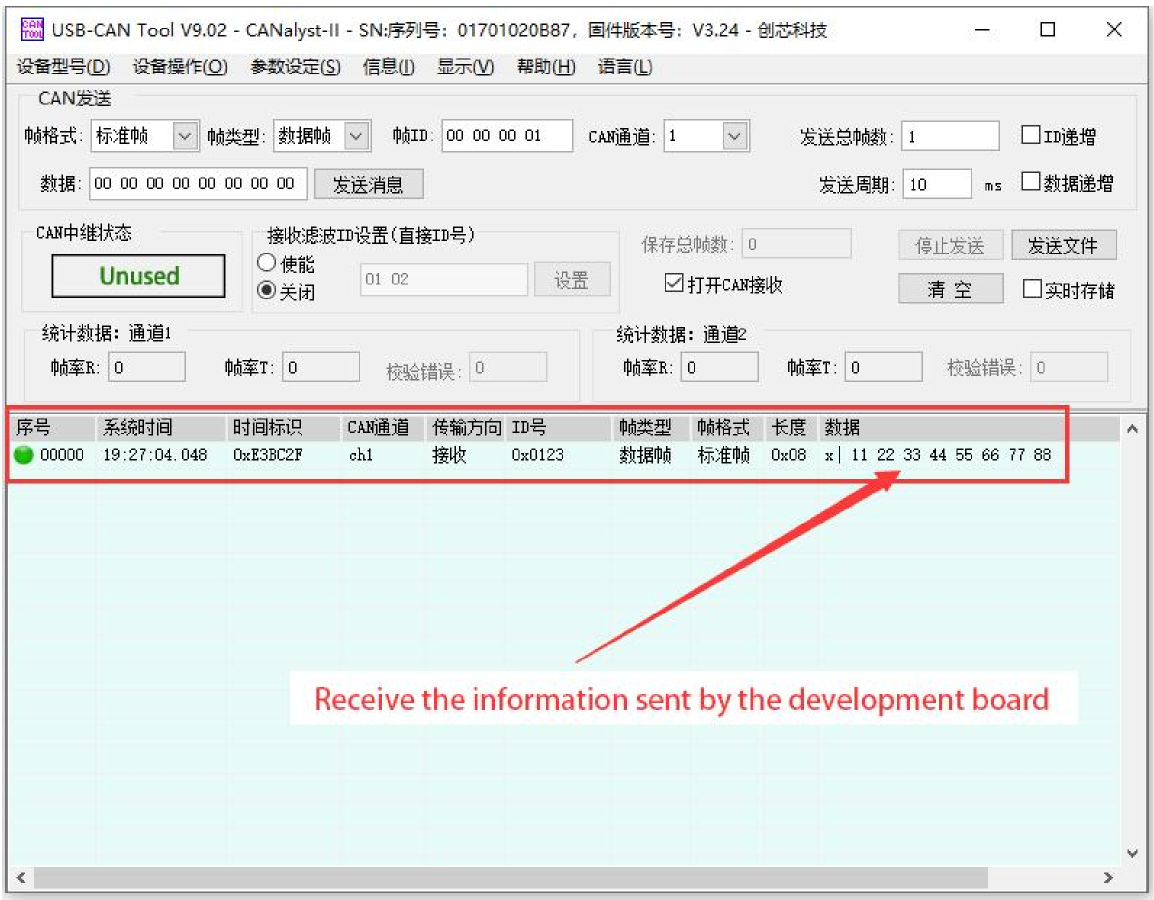
a. First set the CAN baud rate to **1000kbps** in the Linux system

```
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 in the development board to 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.

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 can be seen from the following table, there are six pwm channels available on the development board: pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15. PWM14_M1 and PWM14_M2, PWM15_M1 and PWM15_M3 can only use one of them at the same time, can not be used at the same time, they are the same PWM, but connected to different pins, please do not think that they are two different PWM buses.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

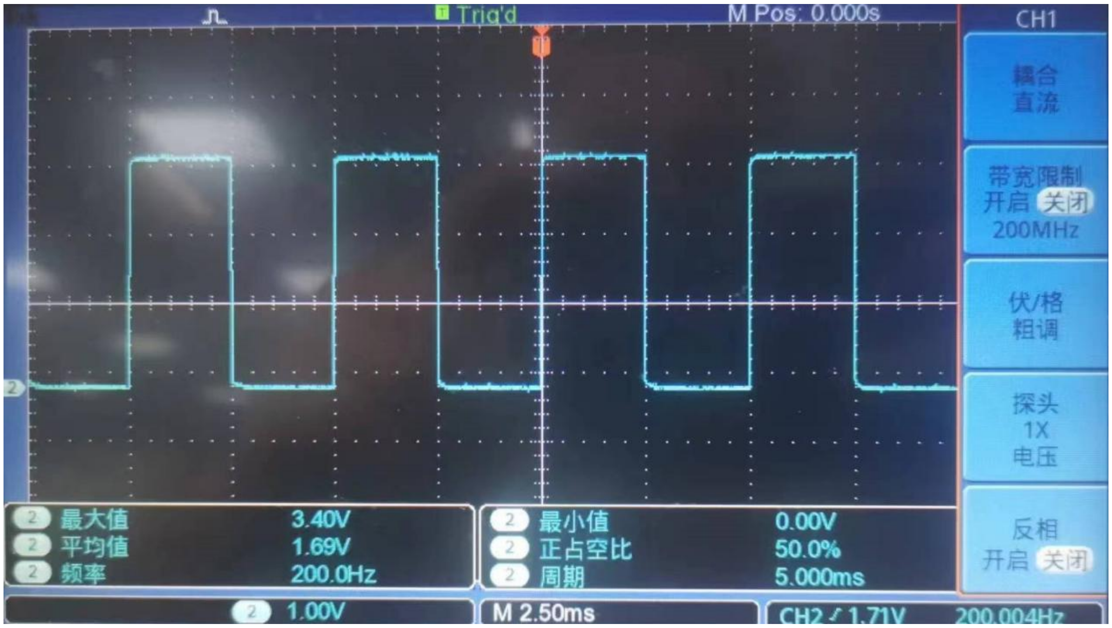
2) The wPi serial number corresponding to the PWM pin is as follows:

PWM Pin	wPi Serial number	Pin Serial number	GPIO Serial number
PWM0_M2	17	27	34
PWM1_M2	18	28	35
PWM3_M3	6	12	39
PWM13_M2	2	7	47
PWM14_M1	5	11	138
PWM14_M2	21	32	62
PWM15_M1	7	13	139
PWM15_M3	22	33	63

3) The command to set the pin to PWM mode is as follows, take PWM0_M2 as an example, where the third parameter needs to enter the wPi number corresponding to the PWM0_M2 pin.

```
orangepi@orangepi:~$ gpio mode 17 pwm
```

4) After the pin is set 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 use an oscilloscope to measure the corresponding PWM pin, and the following waveform can be seen.



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 the frequency division factor

1) We can use the following command to set the frequency division factor of PWM0_M2 pin to 4.

```
orangepi@orangepi:~$ gpio pwmc 17 4
```

2) According to the above formula, the calculated value of PWM frequency is 6000Hz, and it can be observed that the measured value of PWM frequency is 6010Hz through the oscilloscope, and the error can be ignored.

3.19.1.2.2. Method of setting PWM frequency directly

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_M2 pin to 500Hz.

```
orangepi@orangepi:~$ gpio pwmTone 17 500
```

When setting the PWM frequency, it is necessary to ensure that:

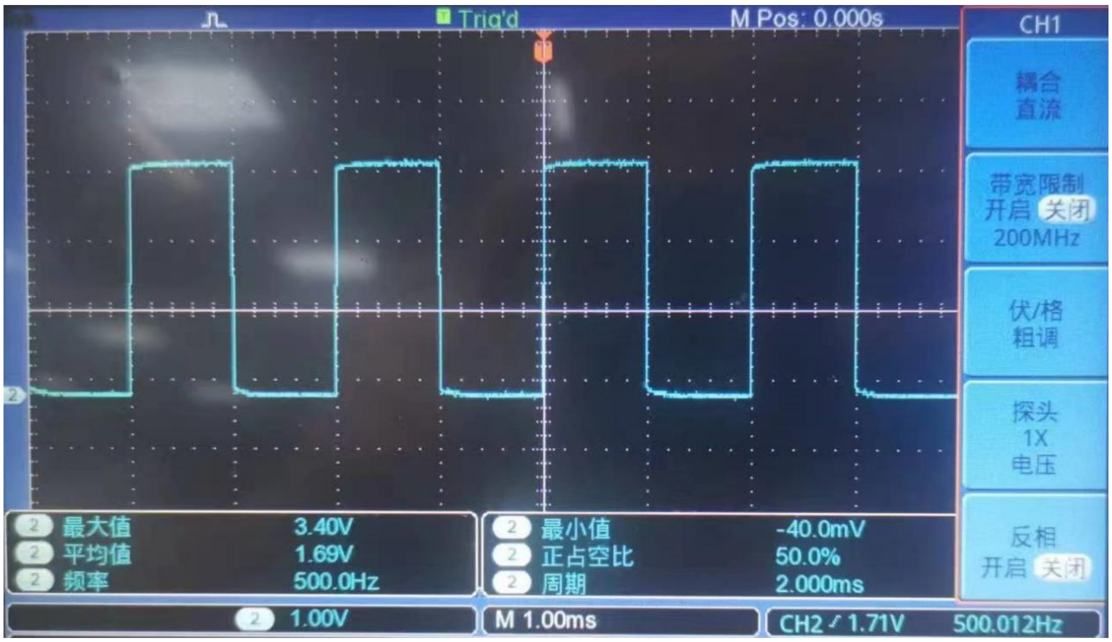
Set frequency value < 24000000 / (division factor * 2)。

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_M2 pin to 2000.

```
orangepi@orangepi:~$ gpio pwmr 17 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_M2 pin to 1000.

```
orangepi@orangepi:~$ gpio pwm 17 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_M2 pin:

```
orangepi@orangepi:/usr/src/wiringOP/examples$ sudo ./pwm 17
```

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. 40pin 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.

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   46 |   8 | GPIO1_B6 |     IN | 1 | 15 || 16 | 0 | IN     | TXD.6    | 9   | 33   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | RXD.6    | 10  | 32   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | GPIO1_B0 | 13  | 40   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 |   34 |  17 |    SDA.4 |     IN | 0 | 27 || 28 | 0 | IN     | SCL.4    | 18  | 35   |
 |   36 |  19 | GPIO1_A4 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |   38 |  20 | GPIO1_A6 |     IN | 0 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |   63 |  22 |    PWM15 |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  135 |  23 | GPIO4_A7 |     IN | 0 | 35 || 36 | 0 | IN     | TXD.0    | 24  | 131  |
 |  134 |  25 | GPIO4_A6 |     IN | 0 | 37 || 38 | 0 | IN     | RXD.0    | 26  | 132  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO4_A5 | 27  | 133  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
orangepi@orangepi5pro:~$
```

1) The following pin 7 —— corresponding to GPIO is GPIO1_C6 —— corresponding to wPi serial number is 2 —— as an example to demonstrate how to set the high and low level of GPIO port

```
orangepi@orangepi5pro:~$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

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. 40pin SPI test

1) As can be seen from the following figure, the available spi of Orange Pi 5 Pro are spi0 and spi4

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of SPI0 and SPI4 in 40pin are shown in the table below.

	SPI0_M2 corresponds to 40pin	SPI4_M2 corresponds to 40pin
MOSI	**Pin 19**	**Pin 16**

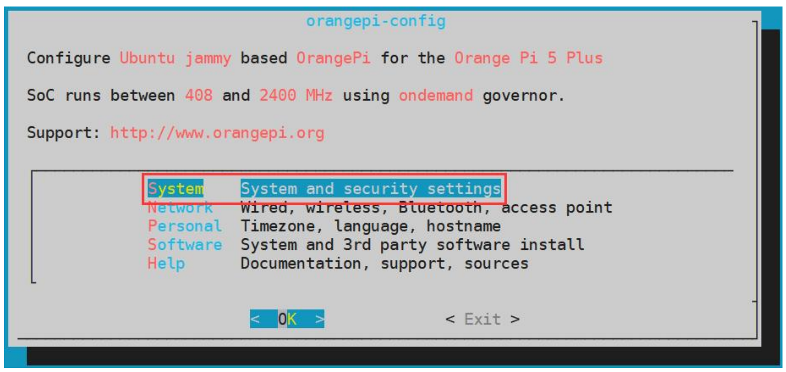
MISO	**Pin 21**	**Pin 18**
CLK	**Pin 23**	**Pin 27**
CS0	**Pin 24**	**Pin 28**
CS1	**Pin 26**	无
Dtbo configuration	**spi0-m2-cs0-spidev** **spi0-m2-cs1-spidev** **spi0-m2-cs0-cs1-spidev**	**spi4-m2-cs0-spidev**

3) In the linux system, the SPI in 40 pin is turned off by default and needs to be opened manually. The detailed steps are as follows:

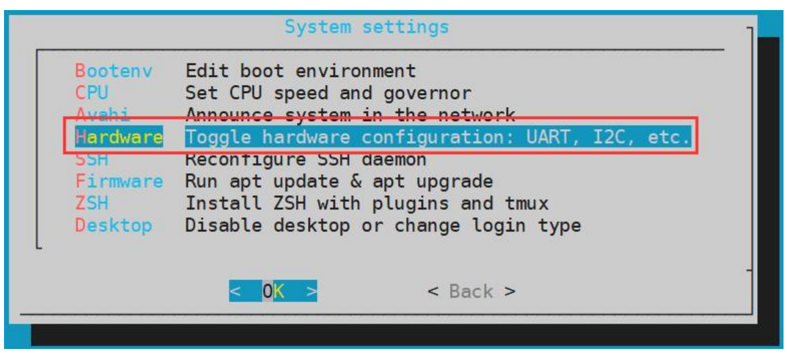
a. First run **orangepi-config**, ordinary users remember to add **sudo** permission

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

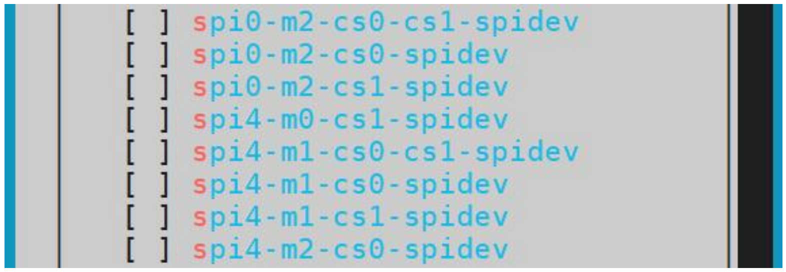
b. Then choose **System**



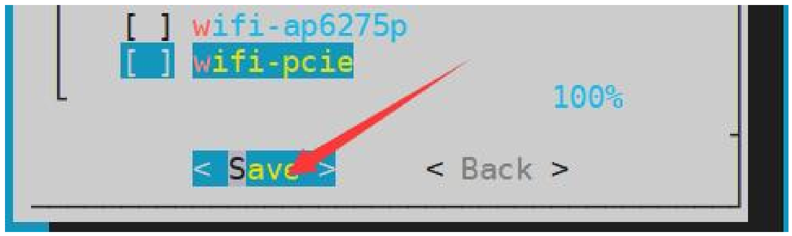
c. Then choose **Hardware**



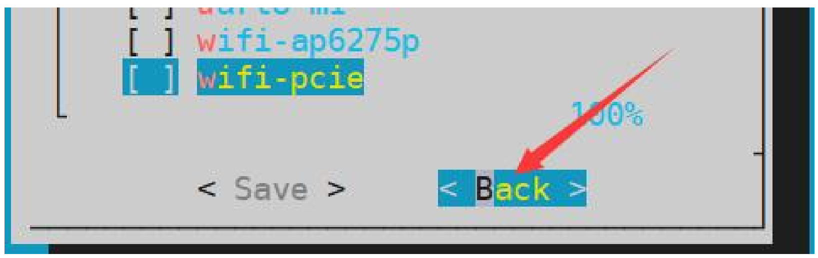
d. Then use the arrow keys on the keyboard to locate the position shown in the following figure, and then use the **space** to select the SPI configuration that you want to open



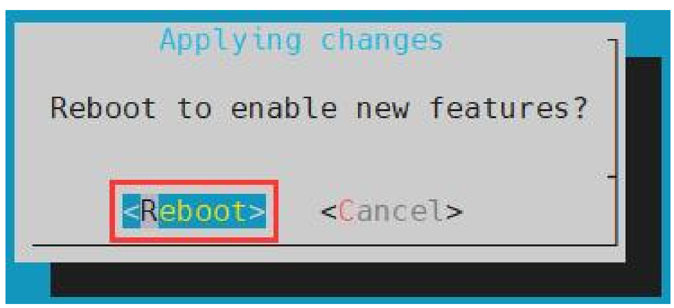
e. Then select **<Save>** to save



f. Then select **<Back>**



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



4) After the restart, go to the system and check whether **spidevx.x** device nodes exist in the linux system. If nodes exist, the SPI is set and can be used directly.

```
orangepi@orangepi:~$ ls /dev/spidev*
/dev/spidev0.0    /dev/spidev0.1    /dev/spidev4.0
```

The above is the result displayed after opening spi0-m2-cs0-cs1-spidev and spi4-m2-cs0-spidev.

5) 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

6) 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

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

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel x --port x
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev0.0
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  |................|
```

7) Then use the Dupont wire to short connect the two pins of SPI txd (pin 19 in the 40pin interface) and rxd (pin 21 in the 40pin interface) and run spidev_test.py as follows. You can see that the sent and received data are the same, indicating that the SPI loop test is normal

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

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel x --port x
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev0.0
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. 40pin I2C 测试

1) As can be seen from the following table, there are four i2c buses available for Orange Pi 5 Pro: i2c1, i2c4, i2c5 and i2c8

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The pins corresponding to the four groups of I2C buses in the 40pin are shown in the following table. I2C5_M2 and I2C5_M3 can only use one group at the same time, can not be used at the same time, they are the same I2C5, just received a different pin up, please do not think that they are two different groups of I2C5 bus.

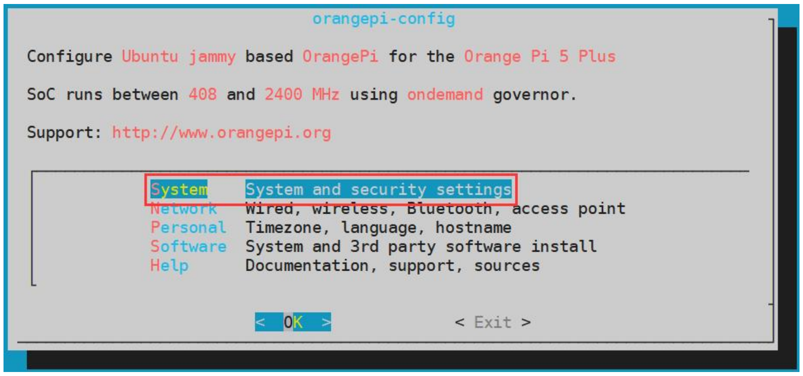
I2C Bus	SDA corresponds to 40pin	SCL corresponds to 40pin	Dtbo corresponding configuration
I2C1_M4	**Pin 3**	**Pin 5**	**i2c1-m4**
I2C4_M3	**Pin 27**	**Pin 28**	**i2c4-m3**
I2C5_M2	**Pin 35**	**Pin 37**	**i2c5-m2**
I2C5_M3	**Pin 7**	**Pin 15**	**i2c5-m3**
I2C8_M2	**Pin 33**	**Pin 32**	**i2c8-m2**

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

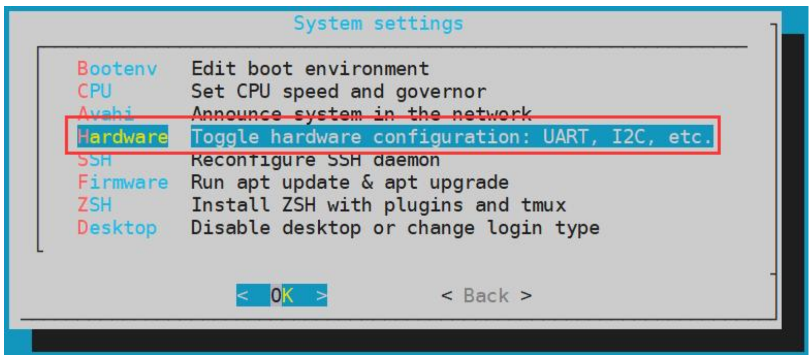
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

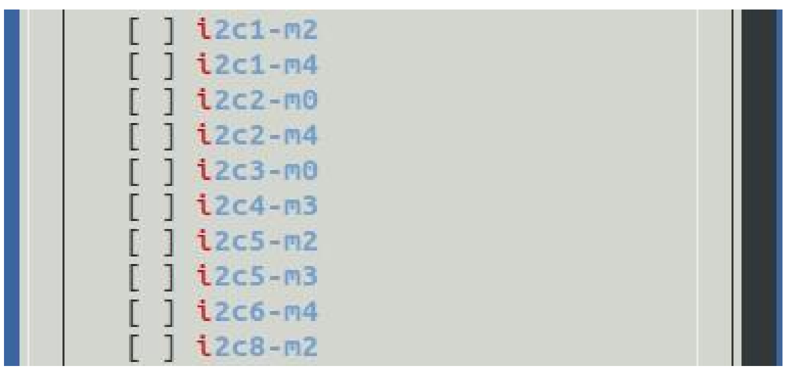
b. Then choose **System**



c. Then choose **Hardware**

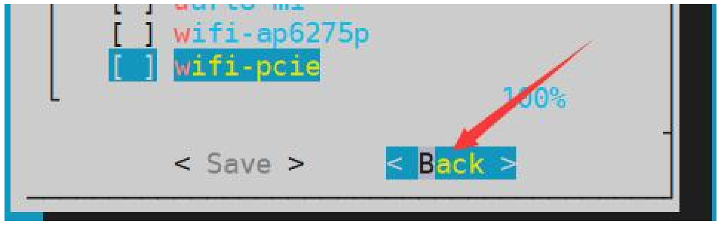


d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the I2C configuration you want to open

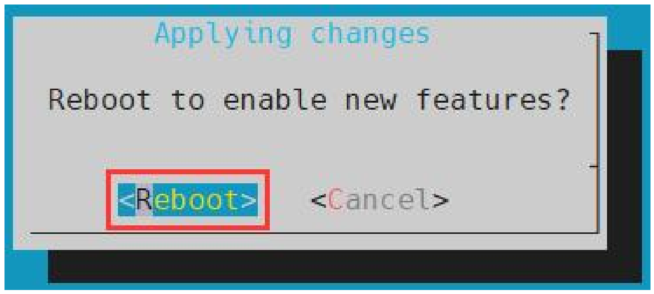


a) Then select **<Save>** to save

e. Then select **<Back>**



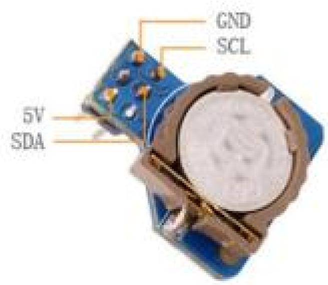
f. Then select **<Reboot>** to restart the system for the configuration to take effect



4) After starting the Linux system, first confirm the existence of i2c device nodes in /dev

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

5) Then connect an i2c device to the i2c pin of the 40pin connector, here we take the ds1307 RTC module as an example



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

```
orangepi@orangepi:~$ sudo i2cdetect -y 1      #i2c1 command
orangepi@orangepi:~$ sudo i2cdetect -y 4      #i2c4 command
orangepi@orangepi:~$ sudo i2cdetect -y 5      #i2c5 command
orangepi@orangepi:~$ sudo i2cdetect -y 8      #i2c8 command
```

```
root@orangepi5pro:~# i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@orangepi5pro:~#
```

7) Then you can run the **ds1307.py** test program in the **examples** to read the RTC time

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device \
"/dev/i2c-1"
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. 40pin UART test

1) As can be seen from the following table, there are five uart buses available for Orange Pi 5 Pro: uart0, uart3, uart4, uart6 and uart7

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of the five UART buses in the 40 pin configuration are shown in the table below

UART Bus	RX corresponds to 40pin	TX corresponds to 40pin	Dtbo corresponding configuration
UART0_M2	**Pin 38**	**Pin 36**	**uart0-m2**
UART3_M2	**Pin 37**	**Pin 40**	**uart3-m2**

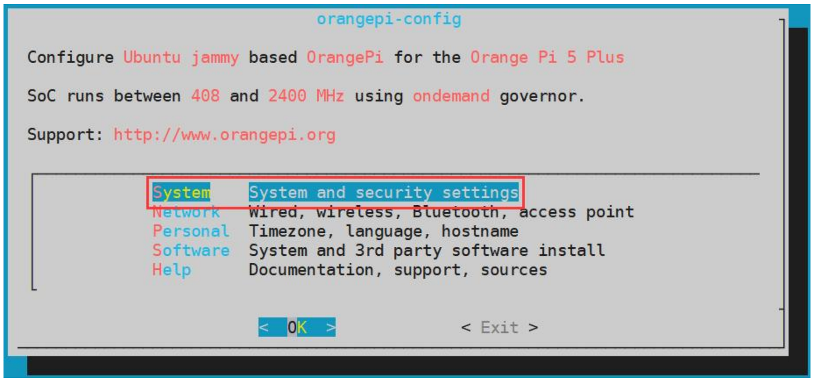
UART4_M0	**Pin 3**	**Pin 5**	**uart4-m0**
UART4_M2	**Pin 19**	**Pin 23**	**uart4-m2**
UART6_M1	**Pin 18**	**Pin 16**	**uart6-m1**
UART7_M2	**Pin 24**	**Pin 26**	**uart7-m2**

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

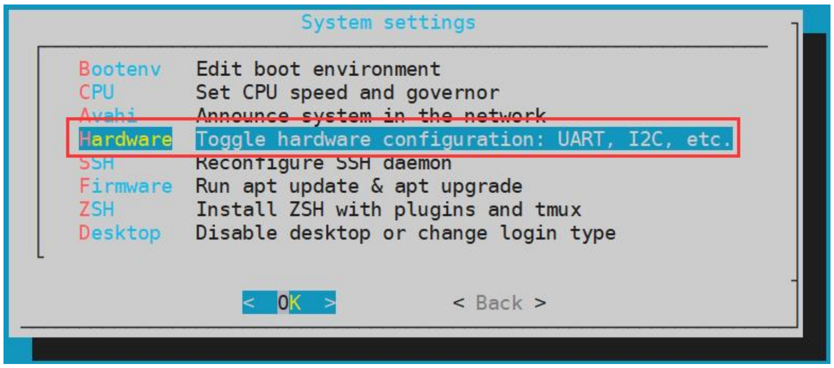
a. First, run **orangepi-config**. Regular users should remember to add **sudo** privileges

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

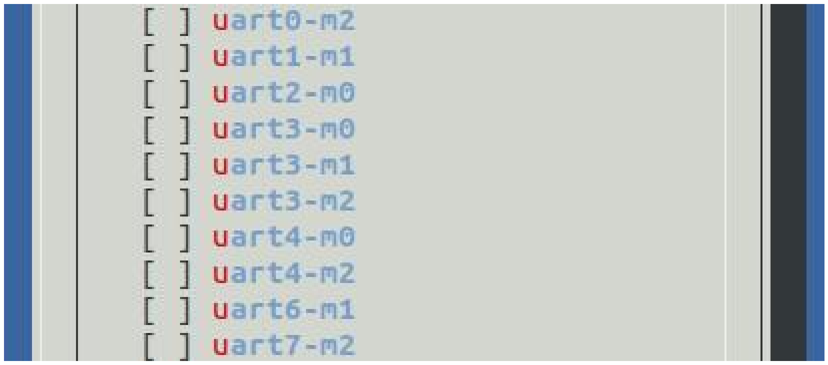
b. Then choose **System**



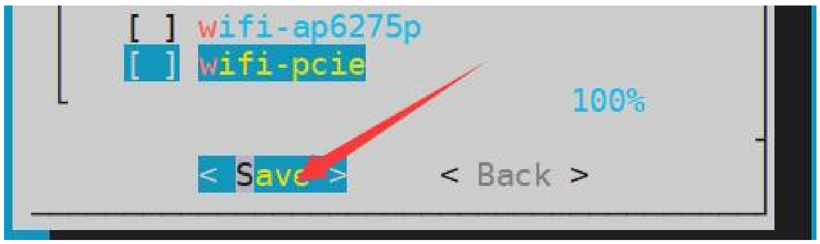
c. Then choose **Hardware**



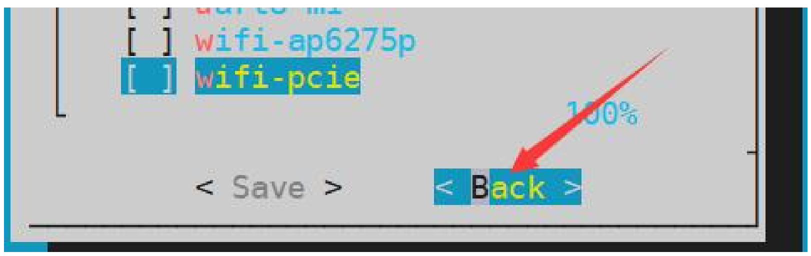
d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use the **space** to select the UART configuration you want to turn on



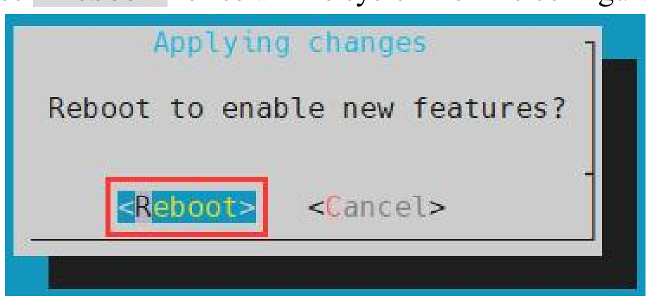
e. Then choose **<Save>** to save



f. Then choose **<Back>**



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



4) After entering the linux system, first confirm whether there is a device node corresponding to uart in /dev

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

5) Then start testing the uart interface, first use the DuPont wire to short-circuit the rx and tx of the uart interface to be tested

6) Test the loopback function of the serial port using the **serialTest.py** program in the examples as shown below. If you can see the print below, it indicates that the serial communication is normal

/dev/ttySX needs to be replaced with the serial number of the specific uart device node.

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttySX"

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 and can be tested directly

The method to run the watchdog_test program is as follows:

a. The second parameter 10 represents 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 feeding is successful.

```
orangepi@orangepi:~$ sudo watchdog_test 10
open success
options is 33152,identity is sunxi-wdt
put_usr return,if 0,success:0
The old reset time is: 16
return ENOTTY,if -1,success:0
return ENOTTY,if -1,success:0
put_user return,if 0,success:0
put_usr return,if 0,success:0
keep alive
```

```
keep alive
keep alive
```

3. 22. Check 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 Docker pre-installed, but the Docker service is not turned on 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 the next time you start the system.

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

3) Then you can use the following command to test docker. If hello-world can be run, 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

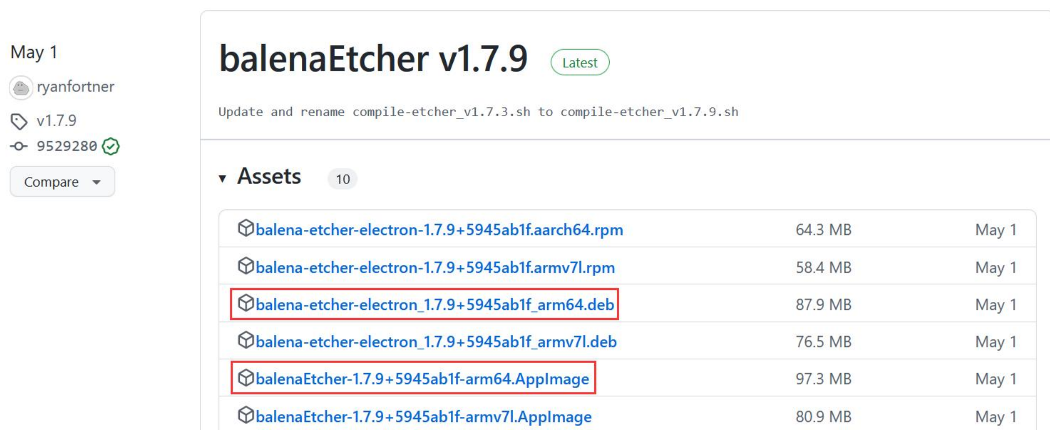
1) The download address of balenaEtcher arm64 version is

a. The download address of the a.deb installation package is as follows. It 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 require installation is as follows:

```
https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balenaEtcher-1.7.9+5945ab1f-arm64.AppImage
```

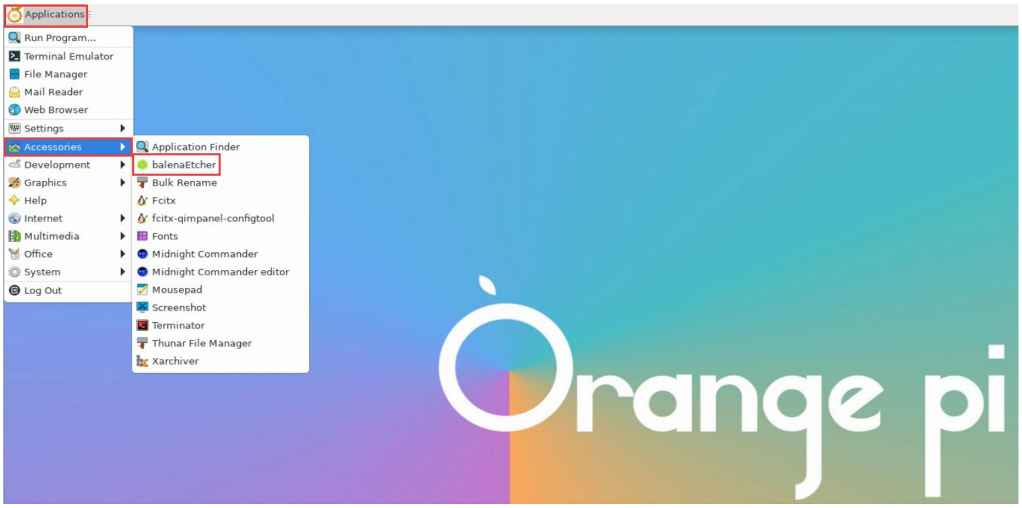


2) How to install and use deb version balenaEtcher:

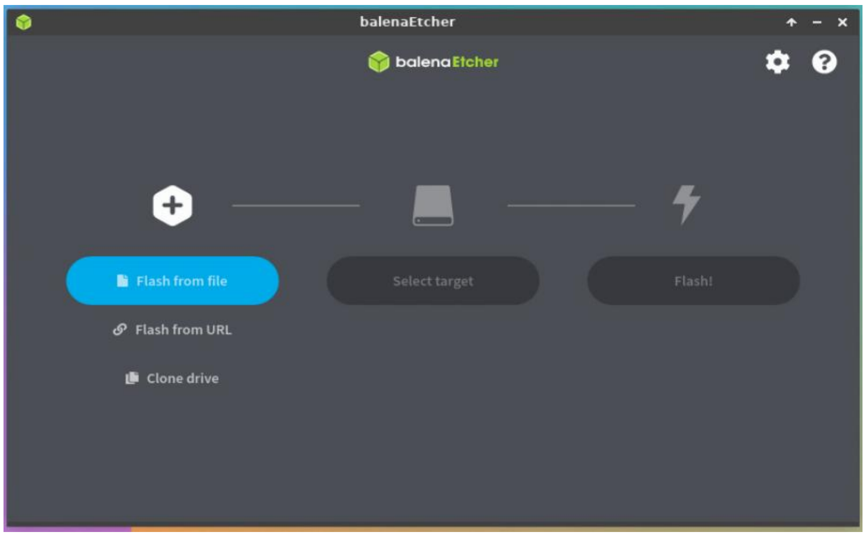
a. The installation command for the a.deb version of balenaEtcher is as follows:

```
orangepi@orangepi:~$ sudo apt install -y \
--fix-broken ./balena-etcher-electron_1.7.9+5945ab1f_arm64.deb
```

b. After the b.deb version of balenaEtcher is installed, it can be opened in Application



c. The interface after opening balenaEtcher is as follows:

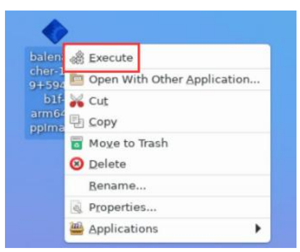


3) How to use the AppImage version of balenaEtcher:

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, right-click the mouse, and click Execute to open balenaEtcher



3. 25. How to install the Pagoda Linux panel

Pagoda Linux panel is a server management software that improves operation and maintenance efficiency. It supports more than 100 server management functions such as one-click LAMP/LNMP/cluster/monitoring/website/FTP/database/JAVA (excerpted from Pagoda official website)

1) The recommended order for Pagoda Linux system compatibility is

```
Debian11 > Ubuntu 22.04 > Debian12
```

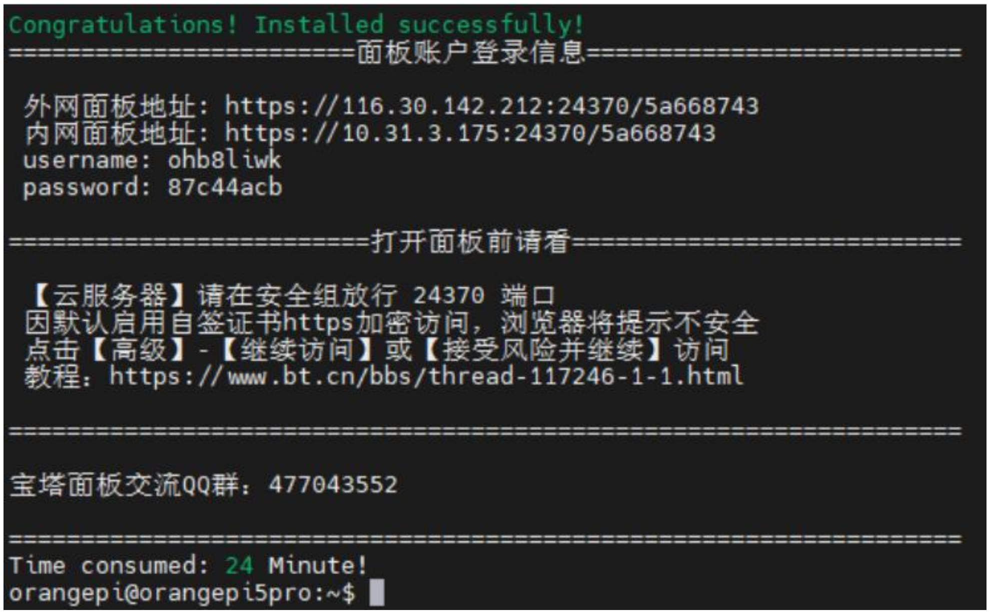
2) Then enter the following command in the linux system to start the installation of the pagoda

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

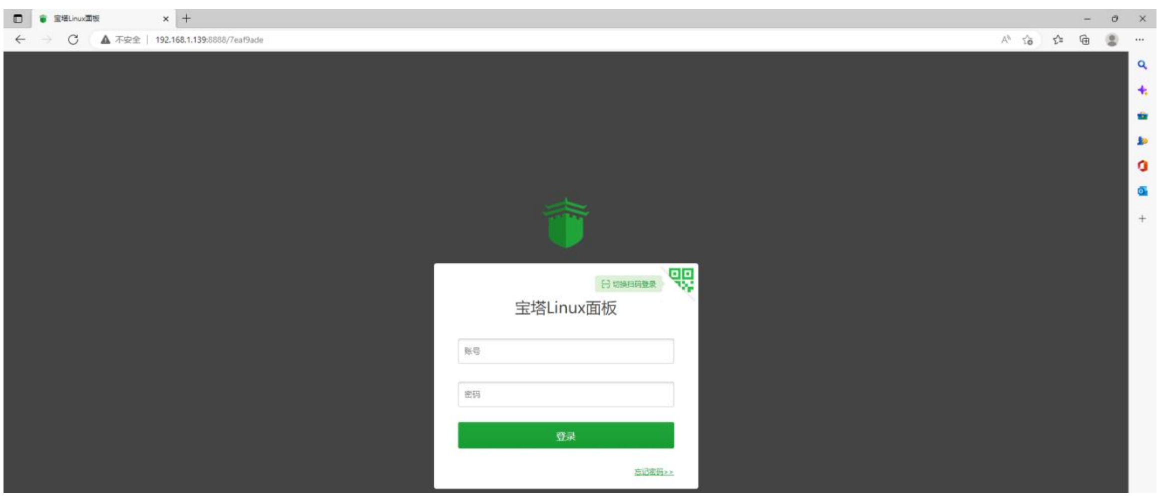
3) Then the Pagoda installation program will prompt whether to install **Bt-Panel** to the **/www** folder. At this time, enter y

```
+----------------------------------------------------------------------
| 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 you see the terminal outputting the following print information, it means that the Pagoda has been installed. The entire installation process takes about 12 minutes. There may be some differences depending on the network speed.

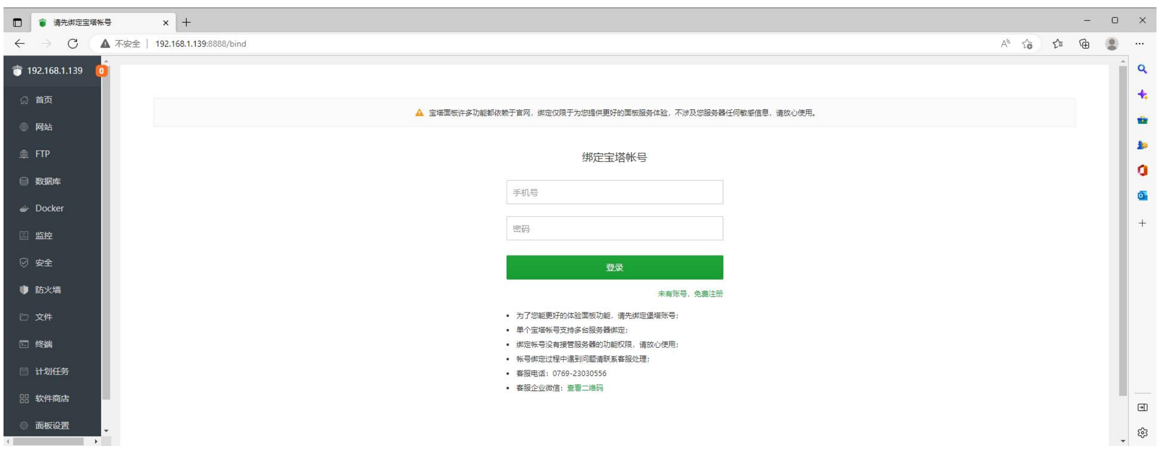


5) At this time, enter the **panel address** shown above in the browser to open the login interface of the Pagoda Linux panel, and then enter the **username** and **password** shown above in the corresponding positions to log in to the Pagoda

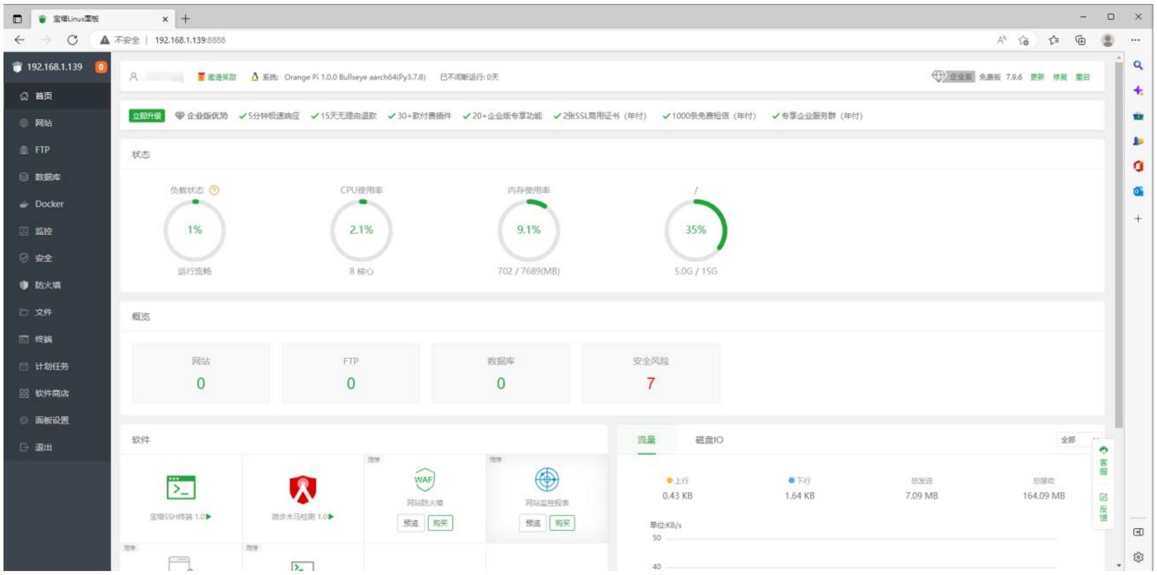


6) After successfully logging into the Pagoda, the following welcome interface will pop up. First, please drag the user instructions in the middle to the bottom after reading them. Then you can select "I have agreed and read the User Agreement", and then click "Enter Panel" You can enter the pagoda

7) After entering the pagoda, you will first be prompted to bind an account on the pagoda's official website. If you don't have an account, you can go to the pagoda's official website (**https://www.bt.cn**) to register one.

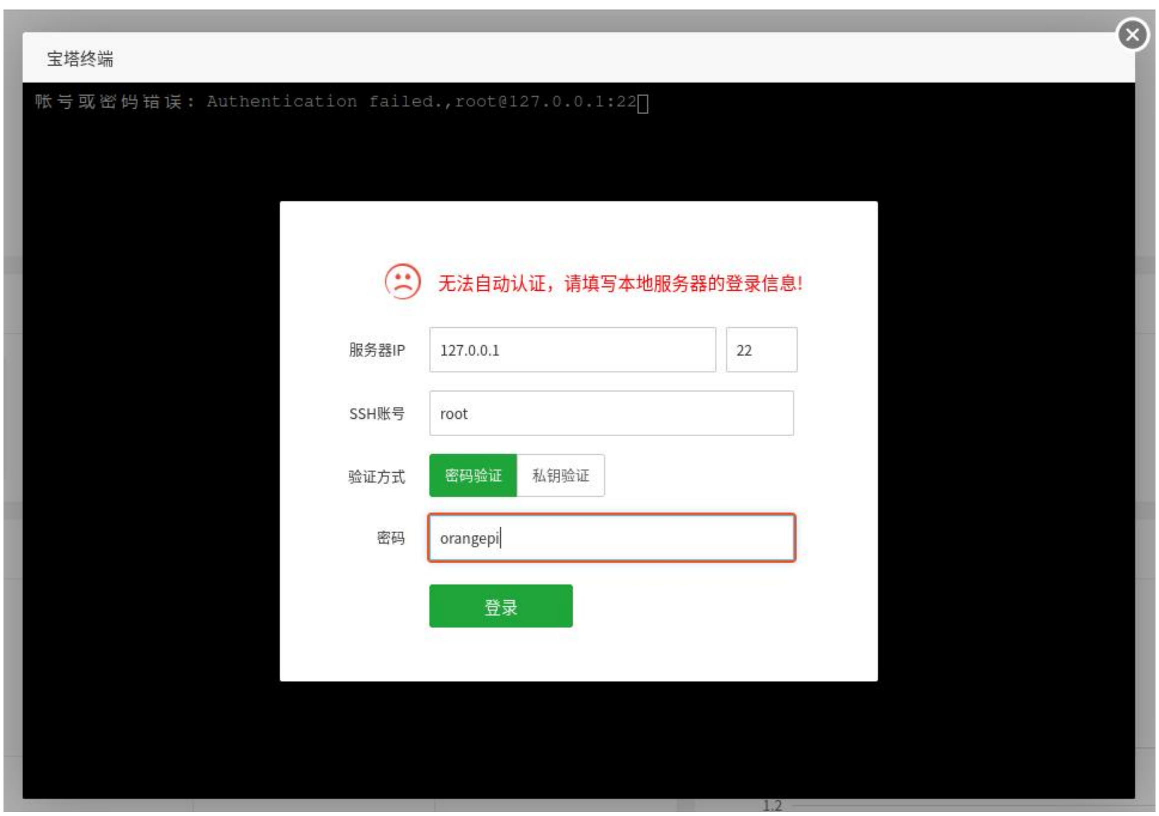


8) The final displayed interface is as shown in the figure below. You can intuitively see some status information of the development board's Linux system, such as load status, CPU usage, memory usage, storage space usage, etc

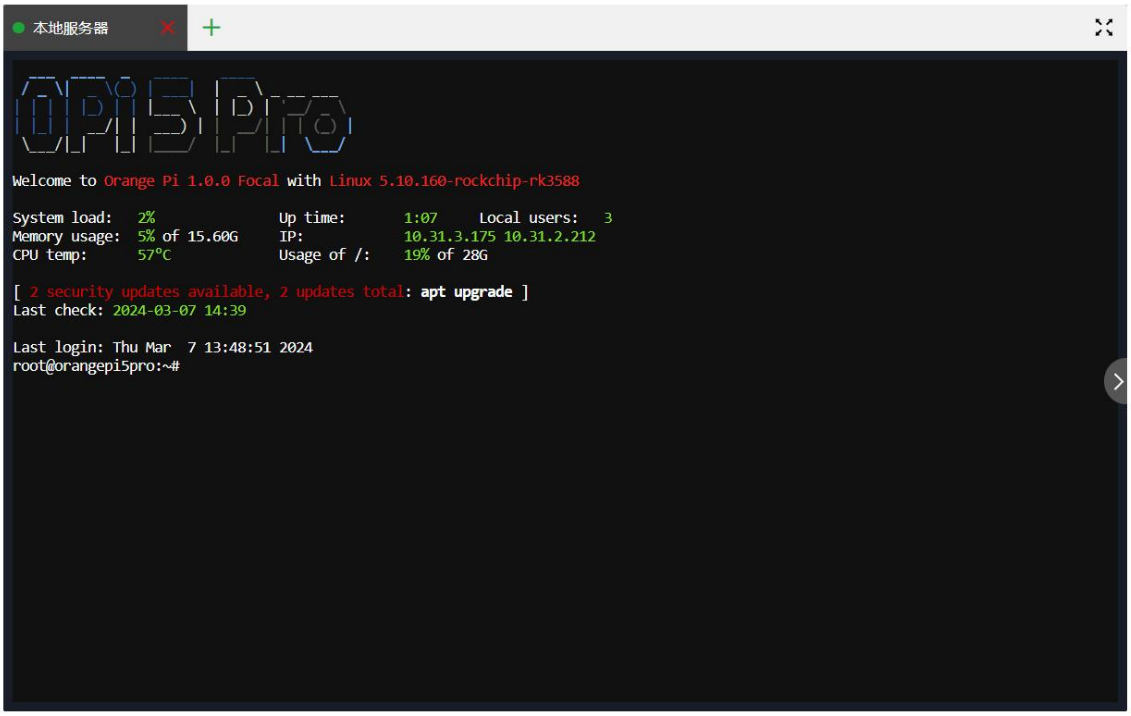


9) Test Pagoda's SSH terminal login

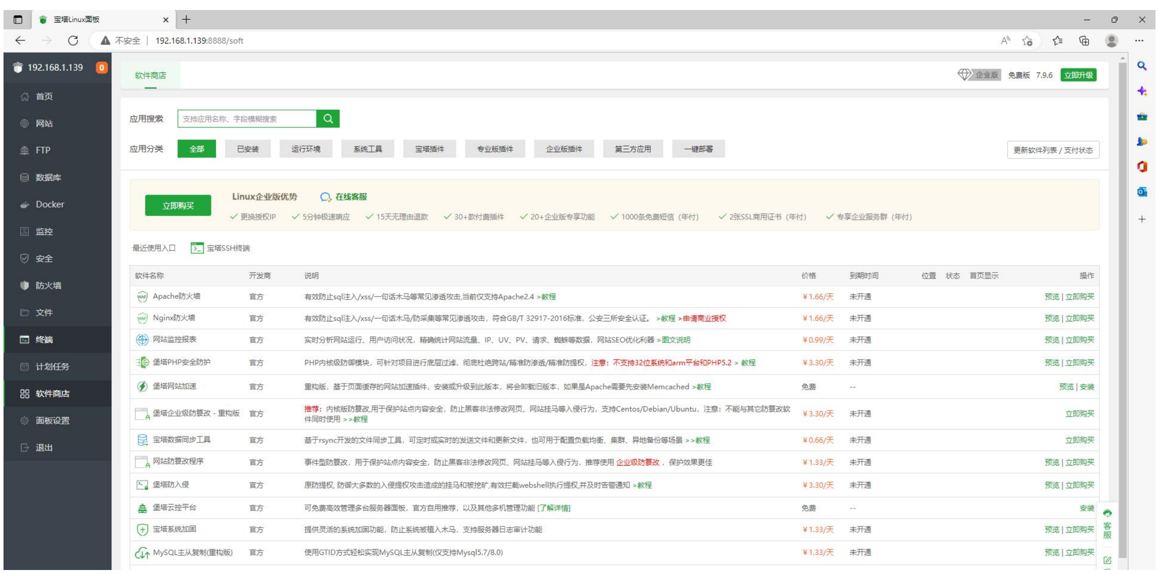
a. After opening the SSH terminal of Pagoda, you will first be prompted to enter the password of the development board system. At this time, enter **orangepi** (the default password, if there is any modification, please fill in the modified one) in the password box.



b. The display after successful login is as shown below



10) In Pagoda' s software store, you can install software such as Apache, MySQL, and PHP, 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@orangepi5pro:~$ sudo bt
==================================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                    (11) 设置是否开启IP + User-Agent验证         |
(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                    |
(26) 关闭面板ssl                    (19) 关闭面板登录地区限制                    |
(28) 修改面板安全入口               (29) 取消访问设备验证                        |
(0) 取消                                                                         |
==================================================================================
请输入命令编号：14
===============================================
正在执行(14)...
===============================================
==================================================================
BT-Panel default info!
==================================================================
外网面板地址: https://116.30.142.212:24370/5a668743
内网面板地址: https://10.31.3.175:24370/5a668743
username: ohb8liwk
password: ********
Warning:
If you cannot access the panel,
release the following port (8888|888|80|443|20|21) in the security group
注意：初始密码仅在首次登录面板前能正确获取，其它时间请通过 bt 5 命令修改密码
==================================================================
orangepi@orangepi5pro:~$
```

12) You can refer to the following information to explore more functions of the pagoda

Manual：**http://docs.bt.cn**

Forum address：**https://www.bt.cn/bbs**

GitHub Link：**https://github.com/aaPanel/BaoTa**

3.26. Set up the Chinese environment and install the Chinese input method

Note, before installing the Chinese input method, please make sure that the Linux system used by the development board is the desktop version.

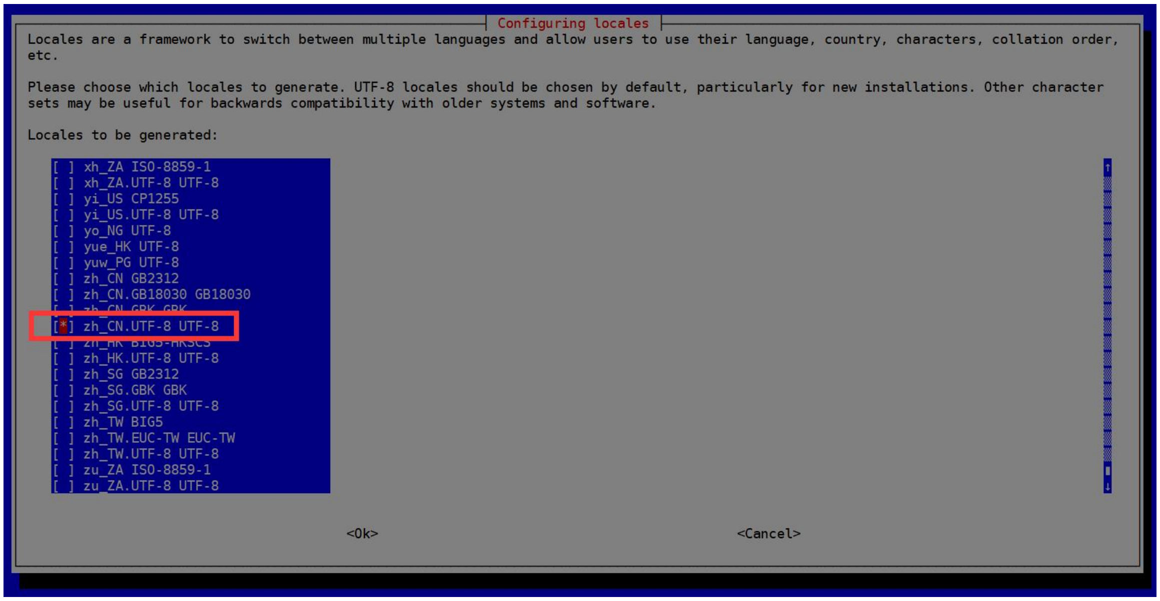
3.26.1. Debian system installation method

1) First set the default **locale** to Chinese

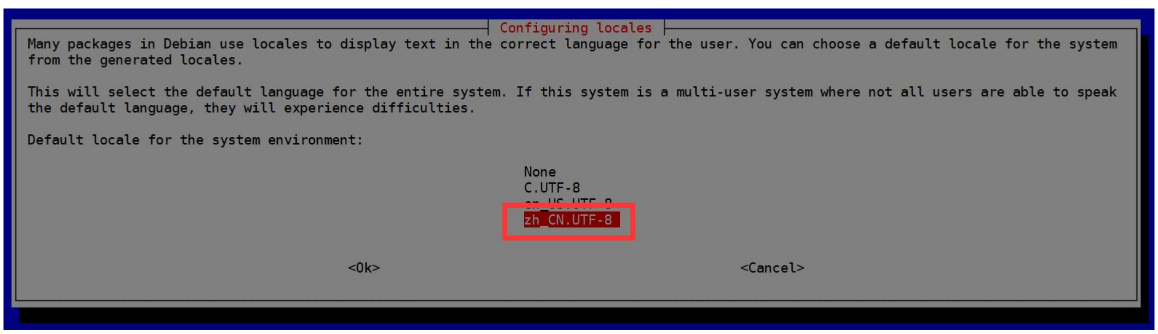
a. Enter the following command to start configuring **locale**

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
```

b. Then select **zh_CN.UTF-8 UTF-8** in the pop-up interface (use the up and down keys on the keyboard to move up and down, use the space bar to select, and finally use the Tab key to move the cursor to **<OK>**, and then return Car is enough)



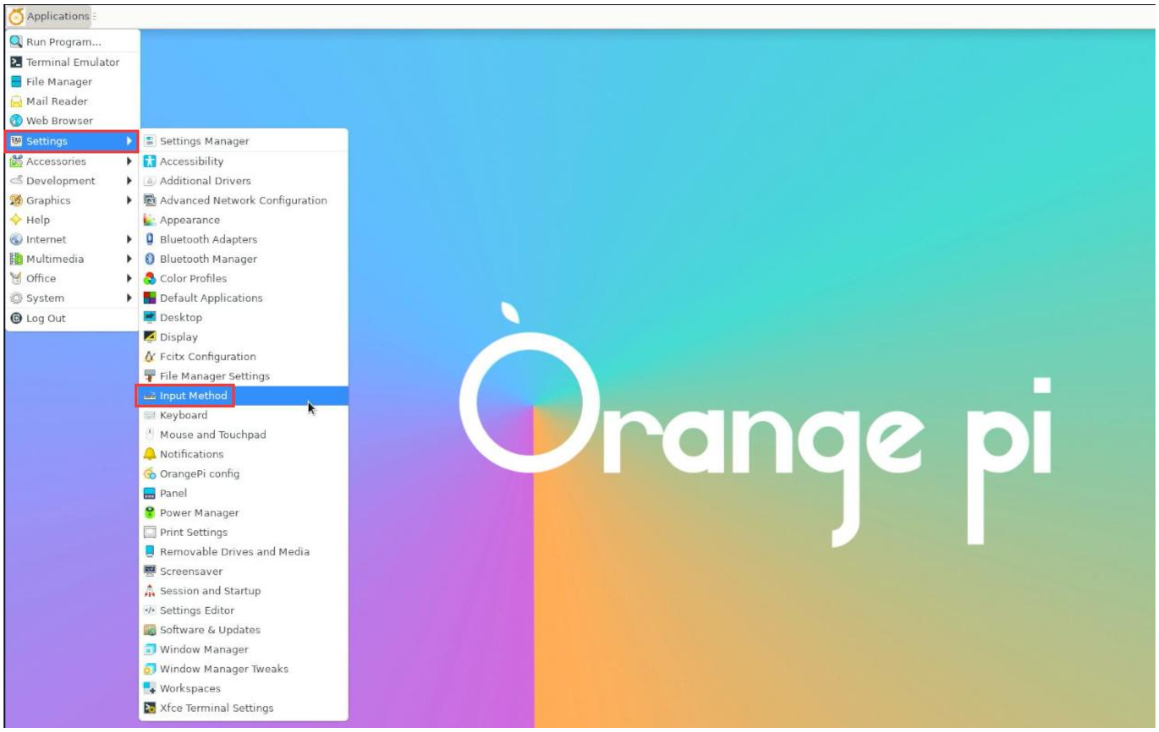
c. Then set the default **locale** to **zh_CN.UTF-8**



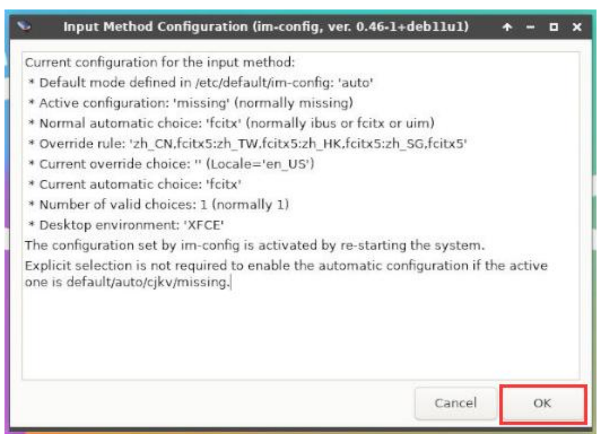
d. After exiting the interface, the **locale** setting will begin. The output displayed on the command line is as follows:

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
  en_US.UTF-8... done
  zh_CN.UTF-8... done
Generation complete.
```

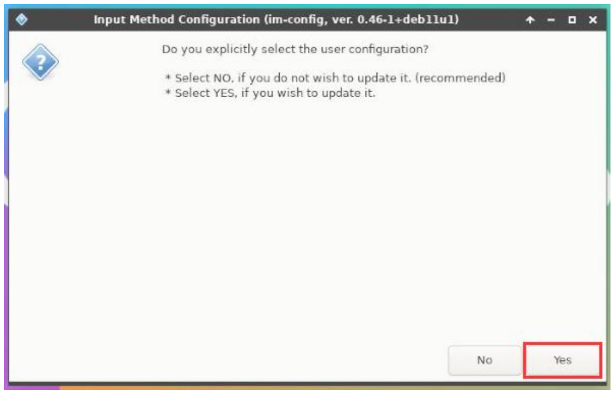
2) Then open **Input Method**



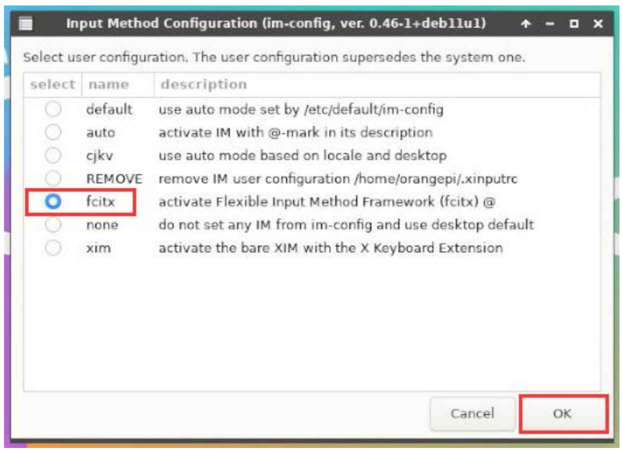
3) Then select **OK**



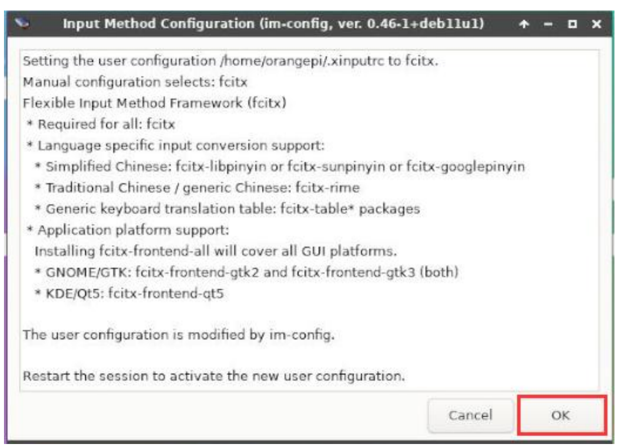
4) Then select **Yes**



5) Then select **fcitx**

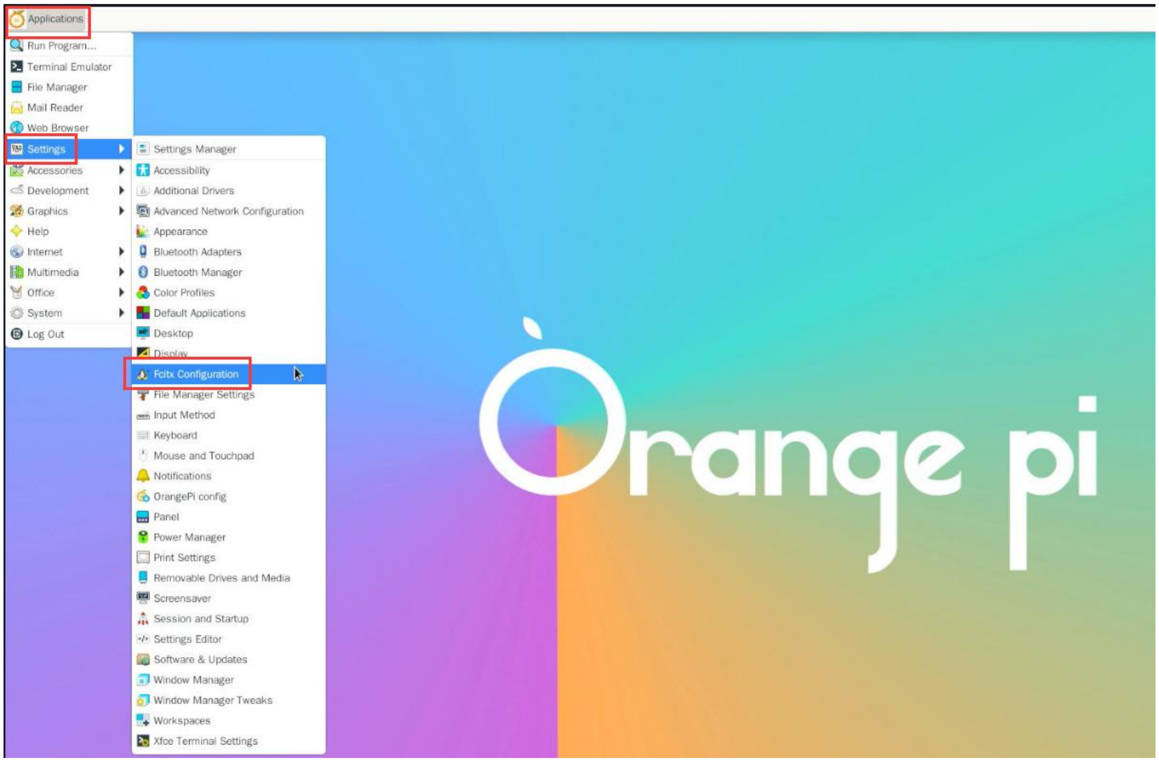


6) Then select **OK**

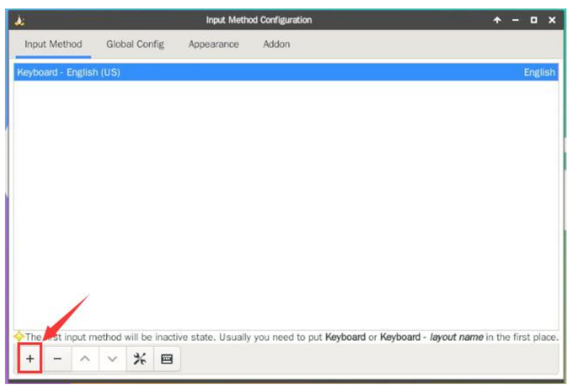


7) **Then restart the Linux system to make the configuration take effect.**

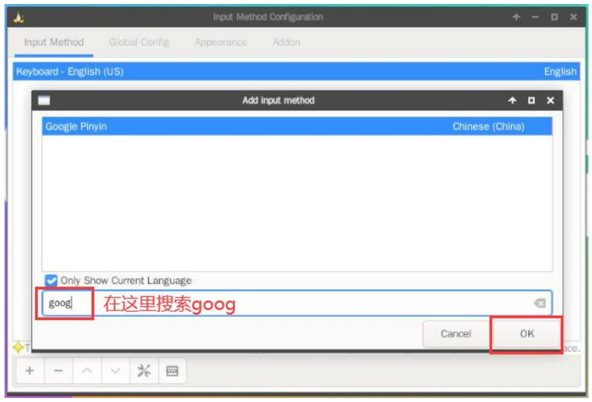
8) Then ope **Fcitx configuration**



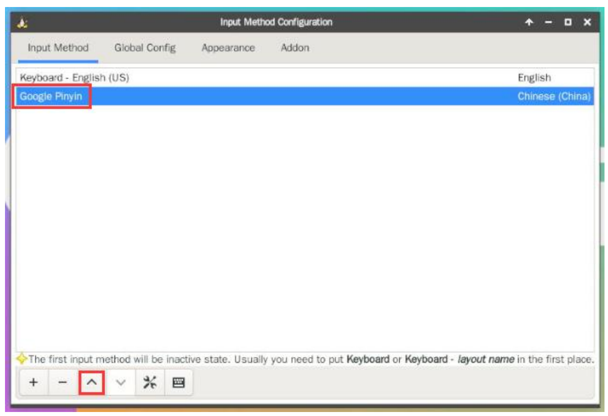
9) Then click the + sign as shown in the picture below

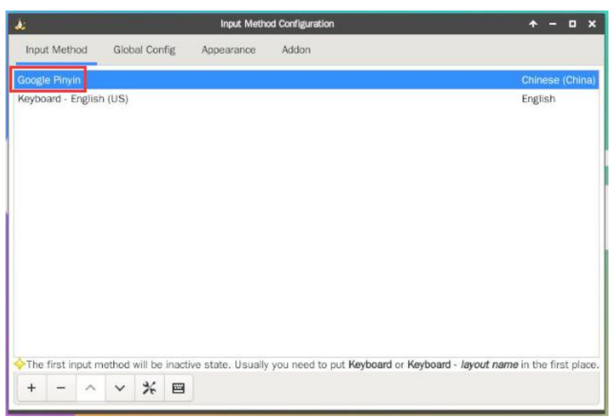


10) Then search **Google Pinyin** and click OK



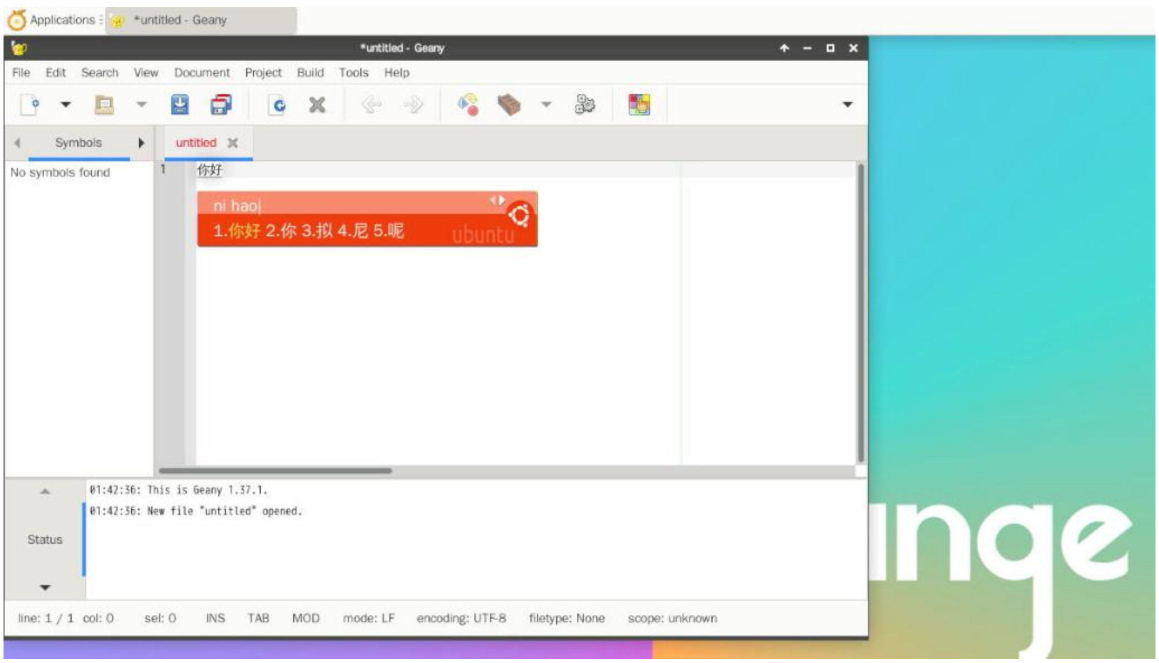
11) Then put **Google Pinyin** on top





12) Then open the **Geany** editor to test the Chinese input method

13) The Chinese input method test is as follows



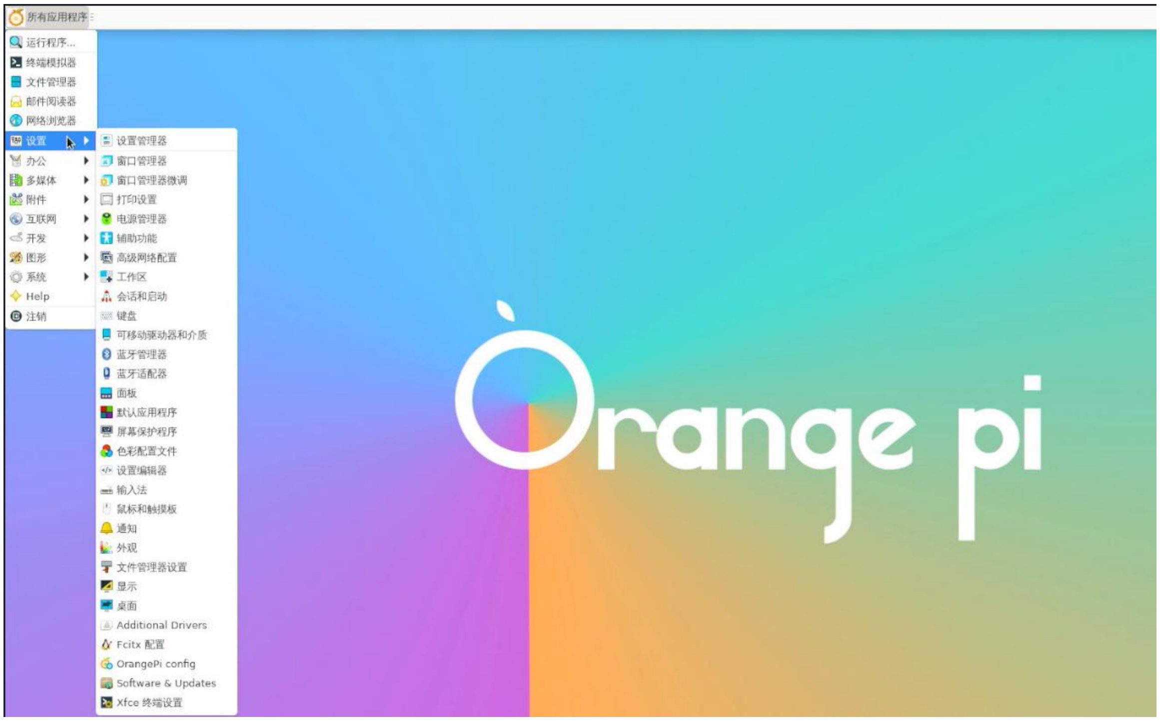
14) You can switch between Chinese and English input methods through the **Ctrl+Space** shortcut key

15) If you need the entire system to be displayed in Chinese, you can set all variables in **/etc/default/locale** to **zh_CN.UTF-8**

```
orangepi@orangepi:~$ sudo vim /etc/default/locale
#   File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
```

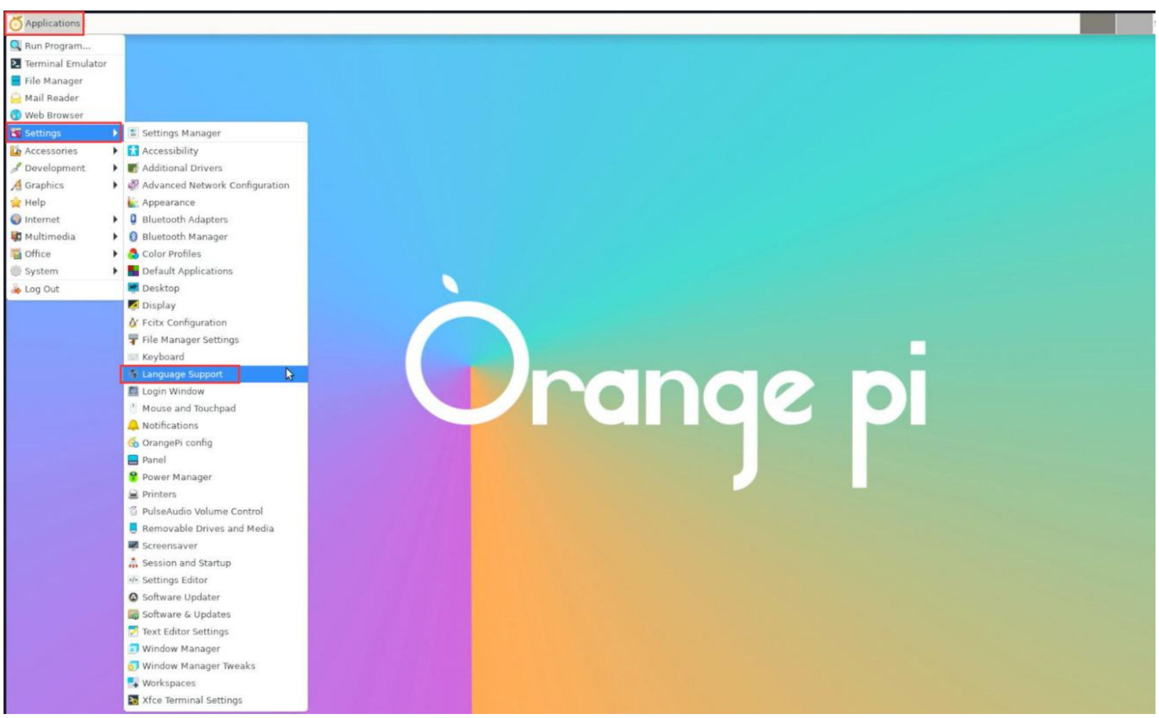
```
LANGUAGE=zh_CN.UTF-8
```

16) Then restart the system and you will see that the system is displayed in Chinese.

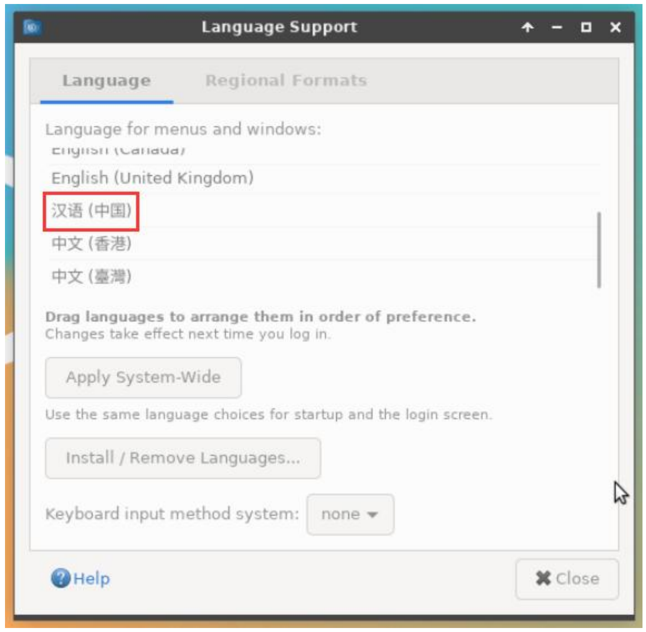


3. 26. 2. Installation method of Ubuntu 20.04 system

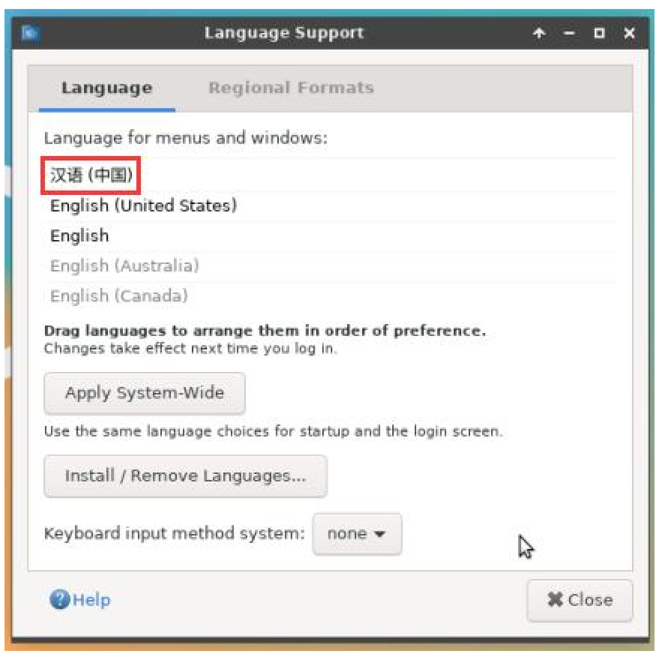
1) First open **Language Support**



2) Then find the **Chinese (China)** option

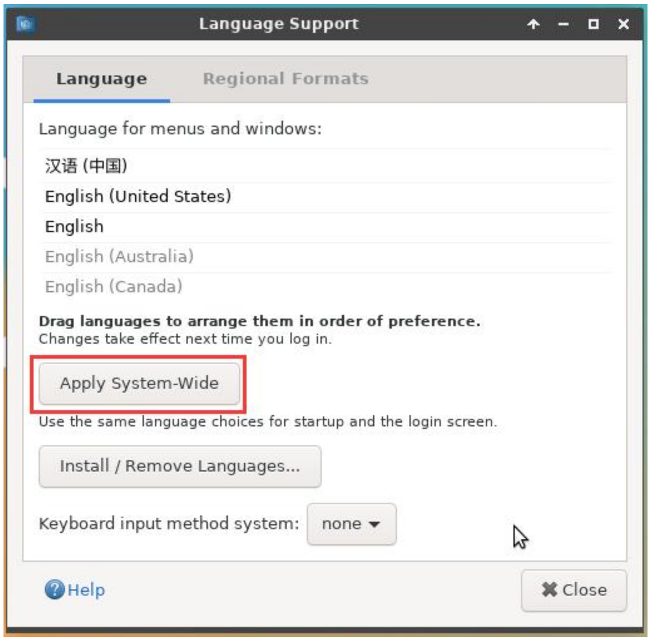


3) Then please use the left button of the mouse to select **Chinese (China)** and hold it down, then drag it up to the starting position. After dragging, the display will be as shown below：

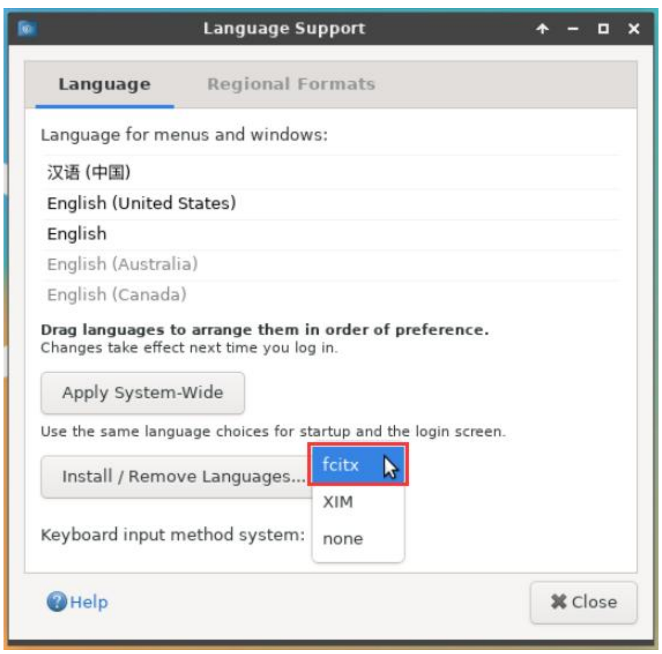


Note that this step is not easy to drag, please be patient and try it a few times.

4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



5) Then set the **Keyboard input method system** to **fcitx**

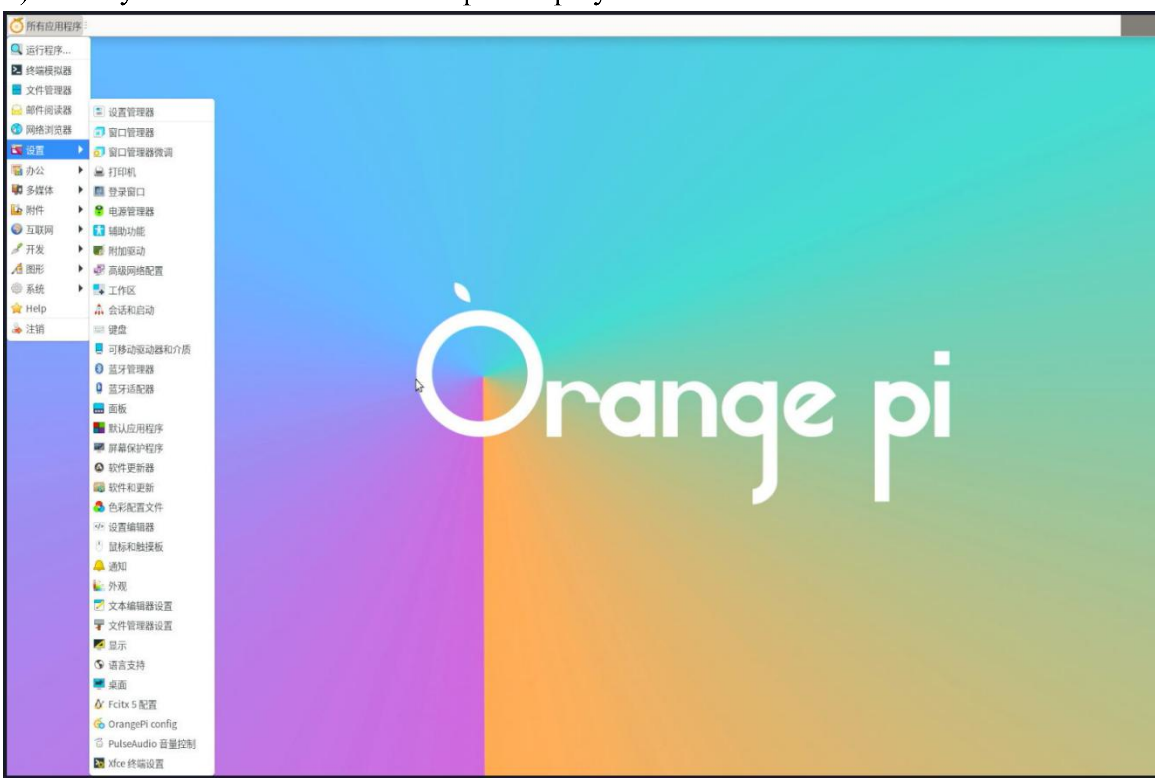


6) **Then restart the Linux system to make the configuration take effect**

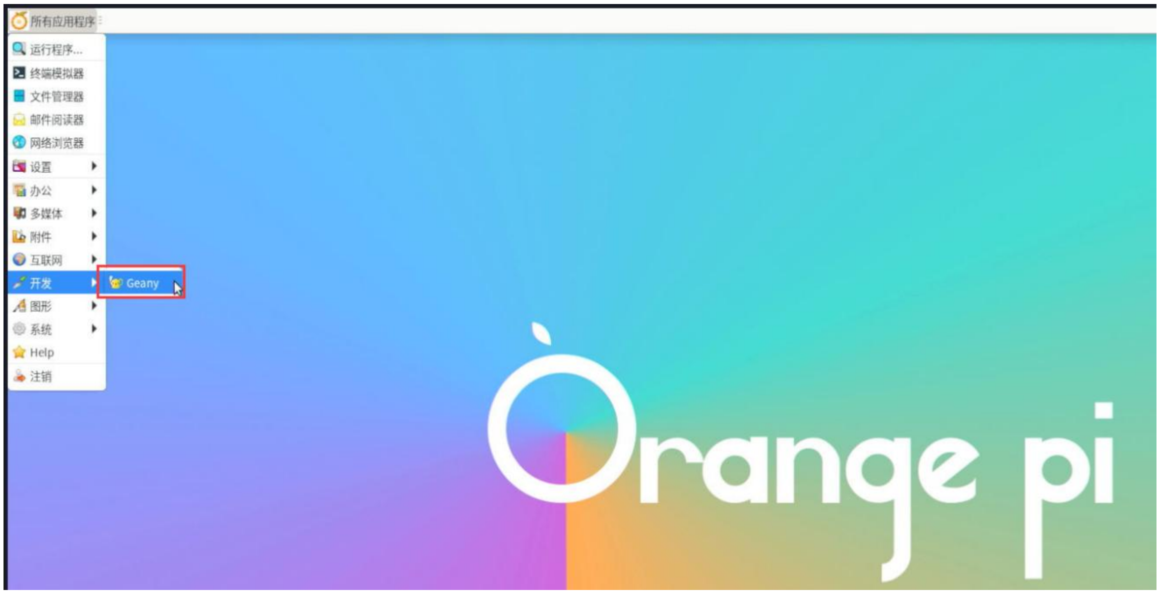
7) After re-entering the system, please select **Do not ask me again** in the following interface, and then please decide according to your own preferences whether the standard

folder should also be updated to Chinese

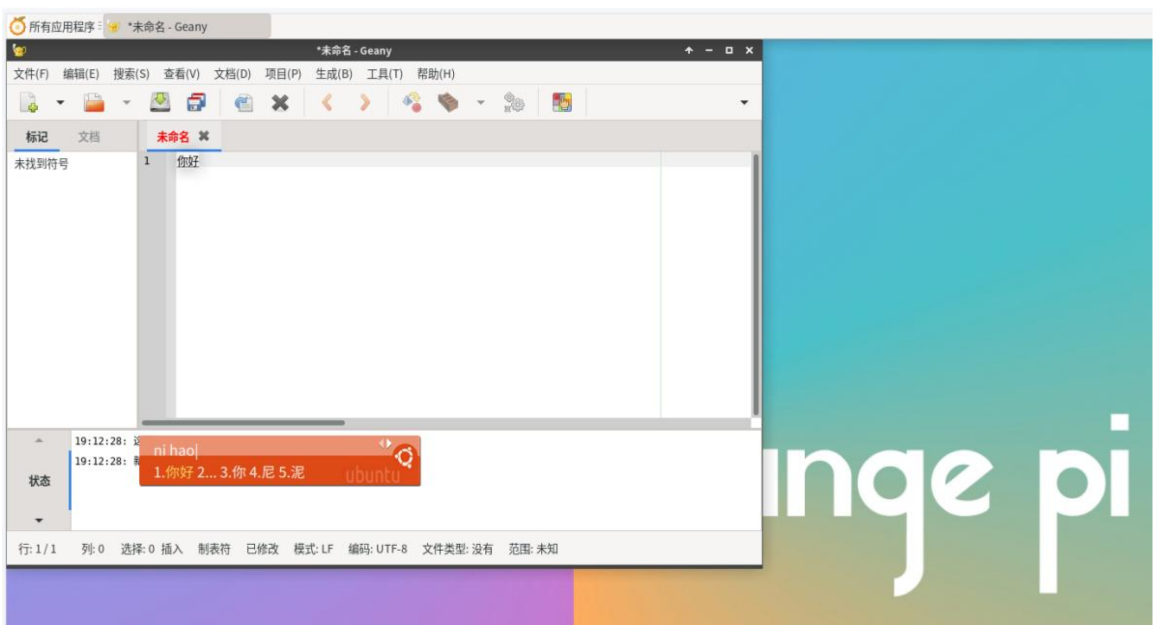
8) Then you can see that the desktop is displayed in Chinese



9) Then we can open **Geany** to test the Chinese input method. The opening method is as shown in the figure below

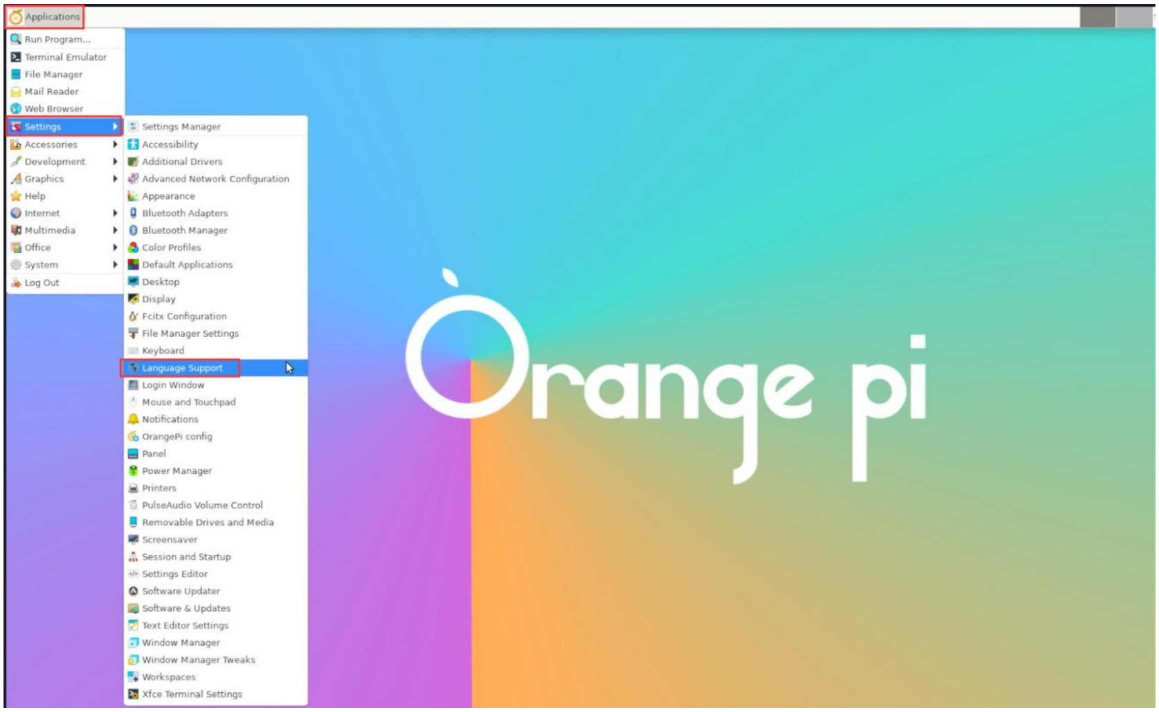


10) After opening **Geany**, the English input method is still the default. We can switch to the Chinese input method through the **Ctrl+Space** shortcut key, and then we can input Chinese.

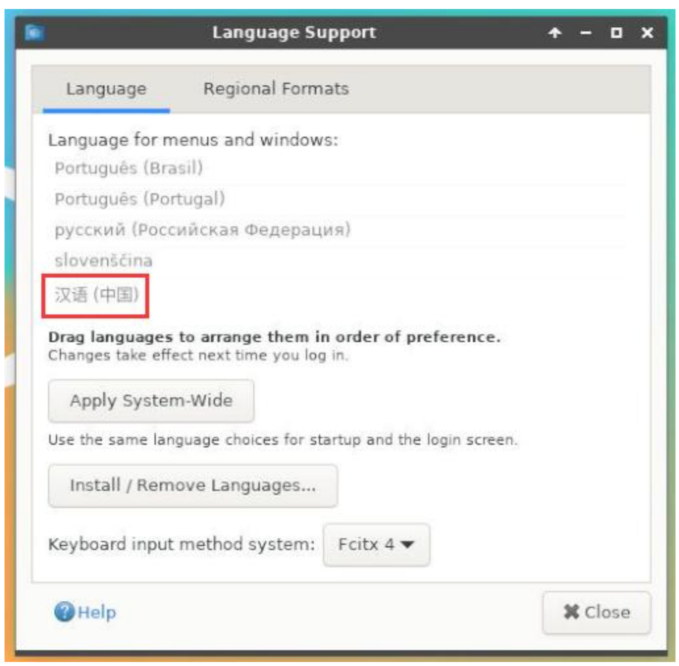


3. 26. 3. Installation method of Ubuntu 22.04 system

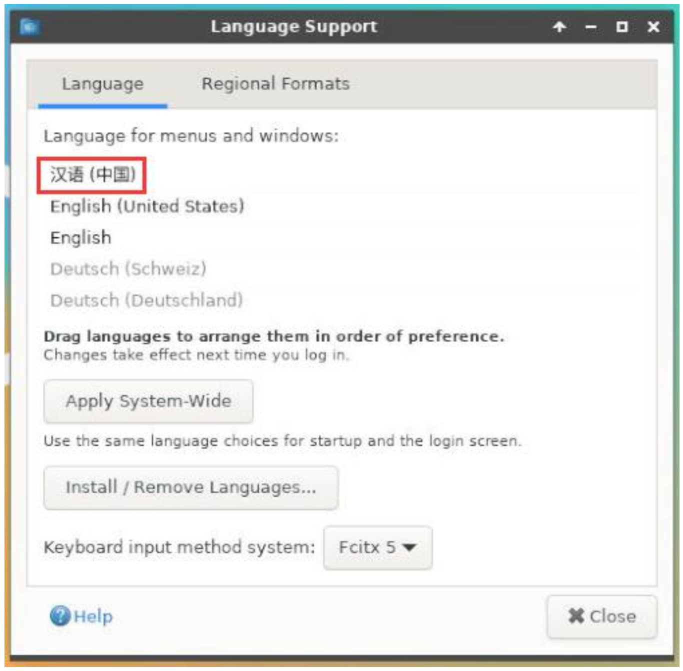
1) First open **Language Support**



2) Then find the **Chinese (China)** option

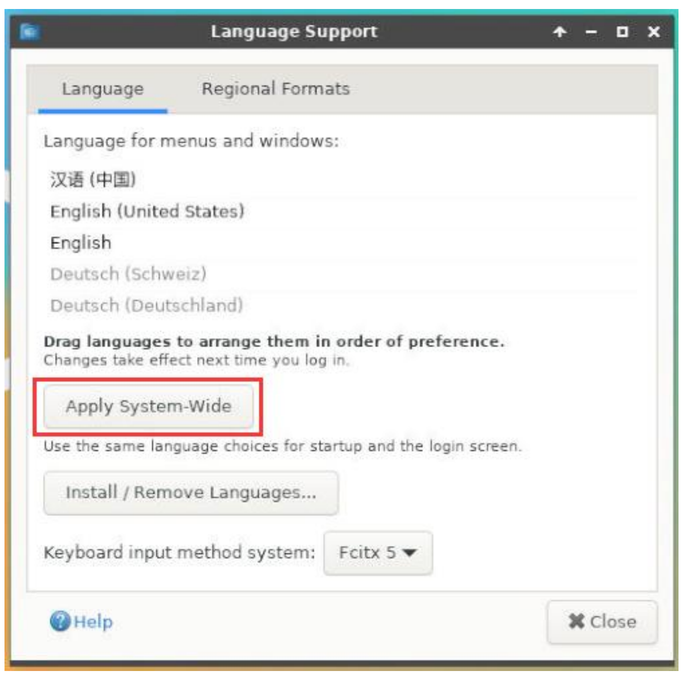


3) Then please use the left button of the mouse to select **Chinese (China)** and hold it down, then drag it up to the starting position. After dragging, the display will be as shown below:



Note that this step is not easy to drag, please be patient and try it a few times.

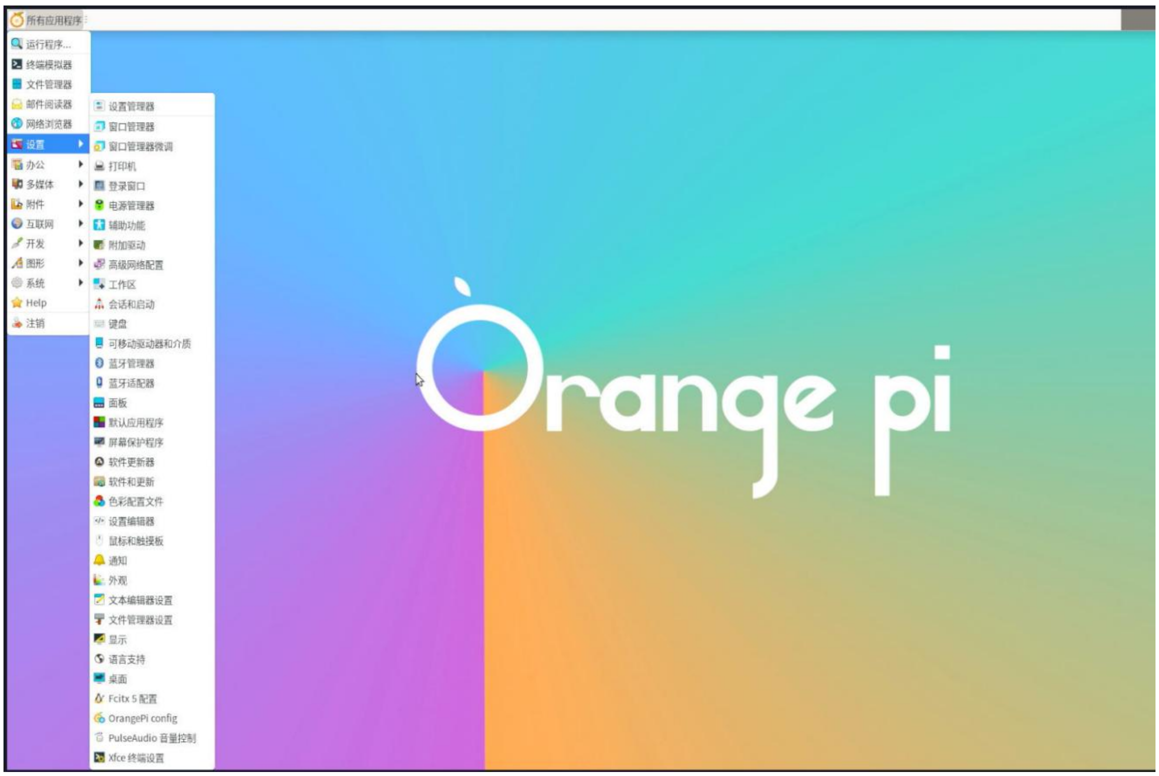
4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



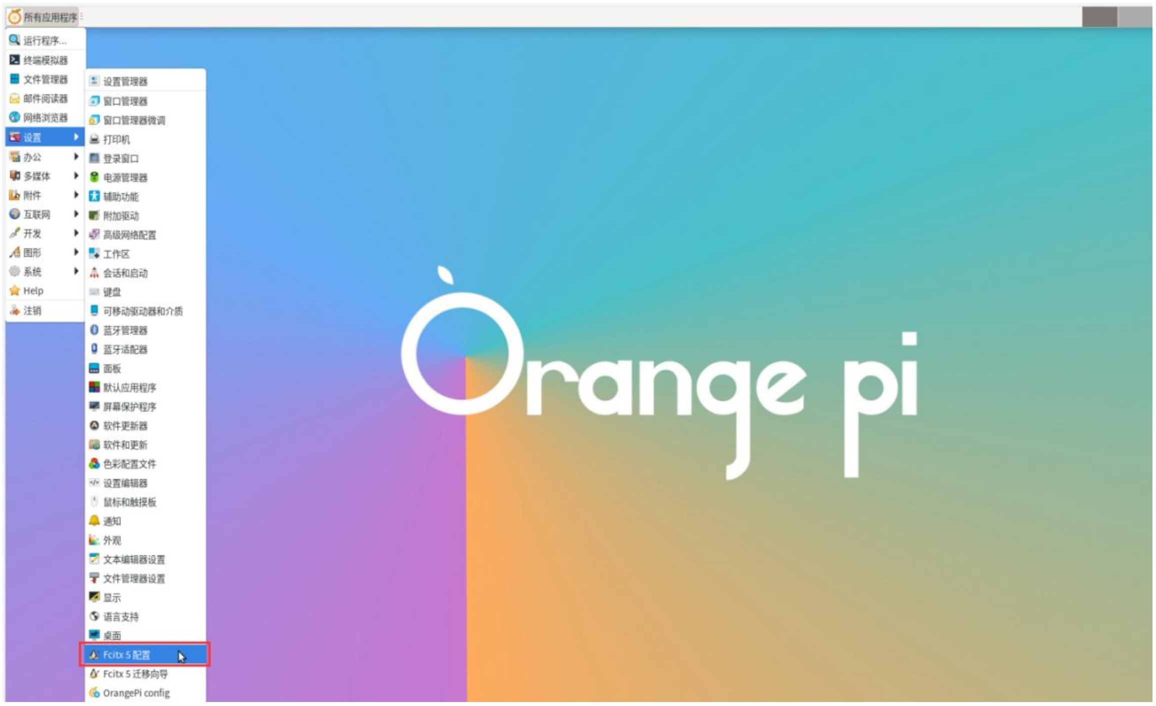
5) **Then restart the Linux system to make the configuration take effect**

6) After re-entering the system, please select **Do not ask me again** on the following interface, and then please decide whether the standard folder should also be updated to Chinese according to your own preferences.

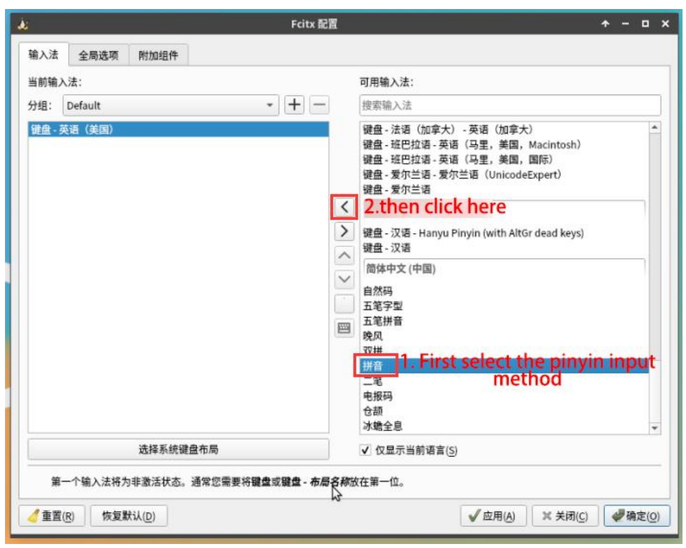
7) Then you can see that the desktop is displayed in Chinese



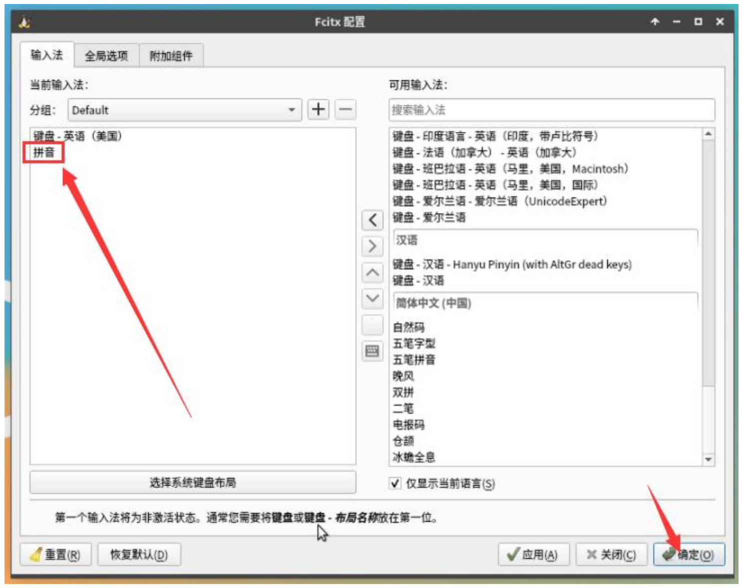
8) Then open the Fcitx5 configuration program



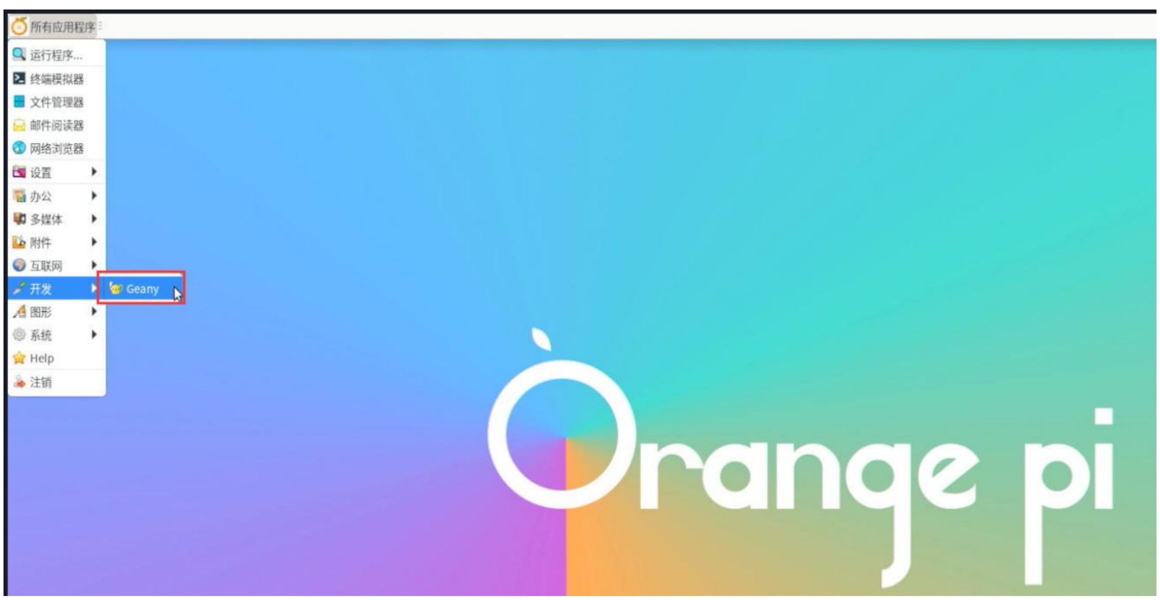
9) Then choose to use Pinyin input method



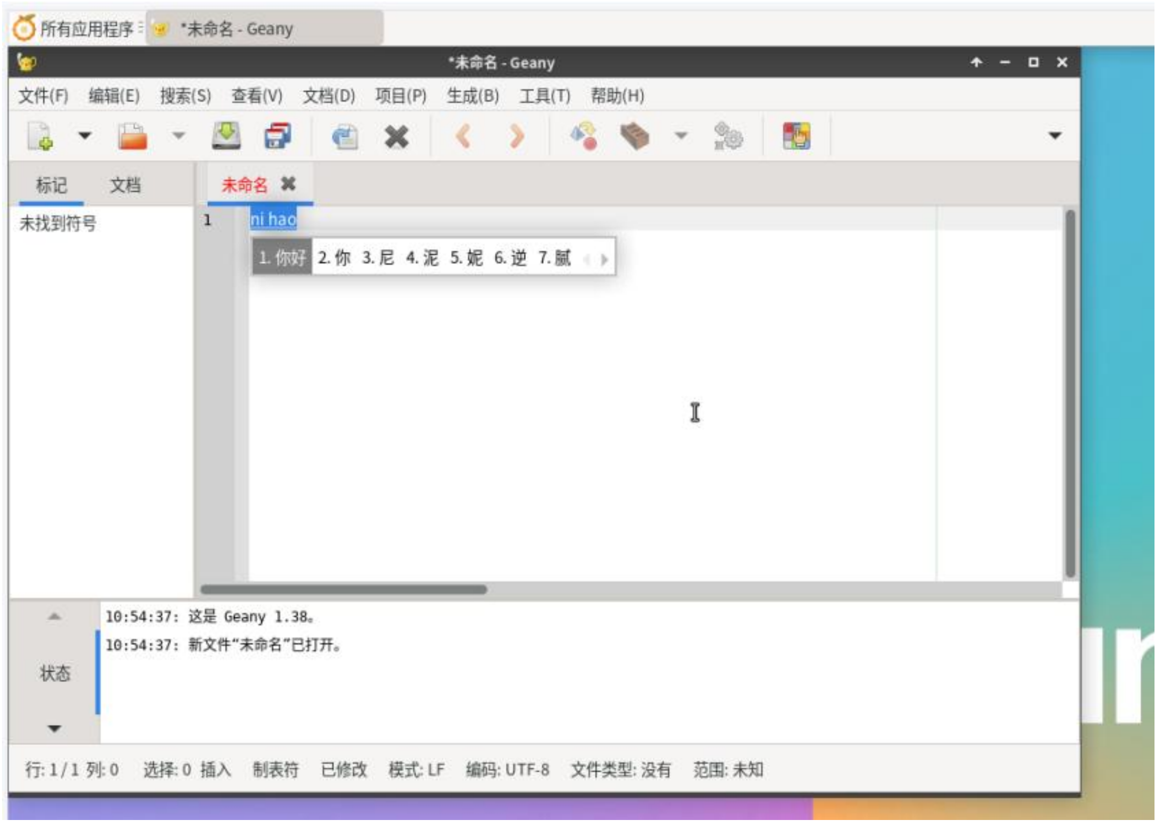
10) The interface after selection is as shown below, then click OK



11) Then we can open **Geany** to test the Chinese input method. The opening method is as shown in the figure below



12) After opening **Geany**, the English input method is still the default. We can switch to the Chinese input method through the **Ctrl+Space** shortcut key, and then we can enter Chinese.



3. 27. How to remotely log in to the Linux system desktop

The Ubuntu Gnome Wayland image does not support remote login to the desktop using Nomachine and VNC as described here.

3. 27. 1. Remote login using NoMachine

Please ensure that the Ubuntu or Debian system installed on the development board is a desktop version of the system. In addition, NoMachine also provides detailed usage documentation. It is strongly recommended to read this document thoroughly to become familiar 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 NoMachine' s official documentation.

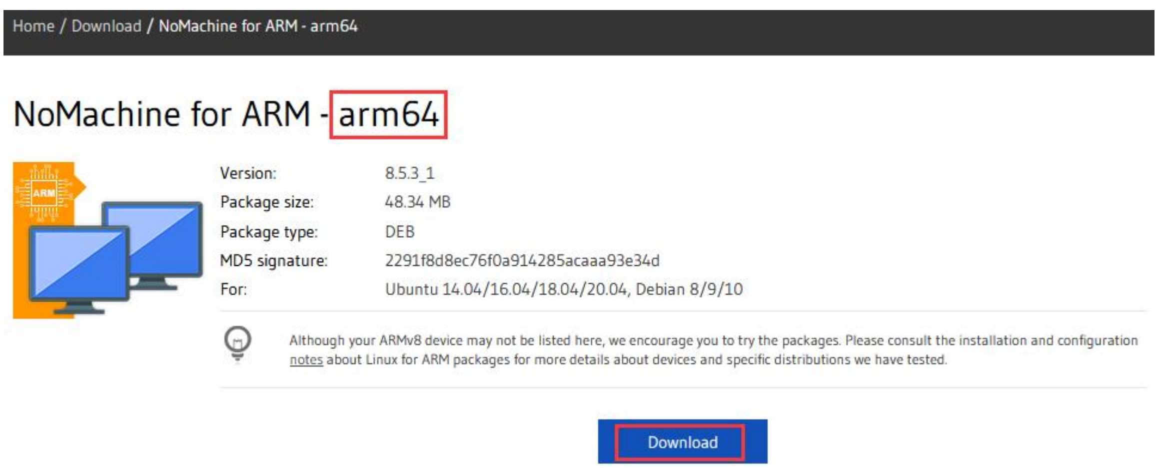
Before operating, please make sure that the Windwos computer and the development board are in the same LAN, and that you can log in to the Ubuntu or Debian system of the development board through ssh normally.

1) First download the installation package of the NoMachine software Linux **arm64** deb version, and then install it into the Linux system of the development board

a. Since RK3588S is an ARMv8 architecture SOC, the system we use is Ubuntu or Debian, so 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 Armv8/Arm64 version of the deb package.

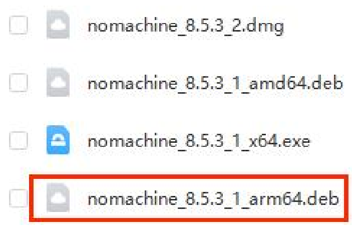
https://downloads.nomachine.com/download/?id=114&distro=ARM



b. In addition, you can also download the **NoMachine** installation package from the official tool.

First enter the **remote login software-NoMachine** folder

Then download the arm64 version of the deb installation package



c. Then upload the downloaded **nomachine_x.x.x_x_arm64.deb** to the Linux system of the development board

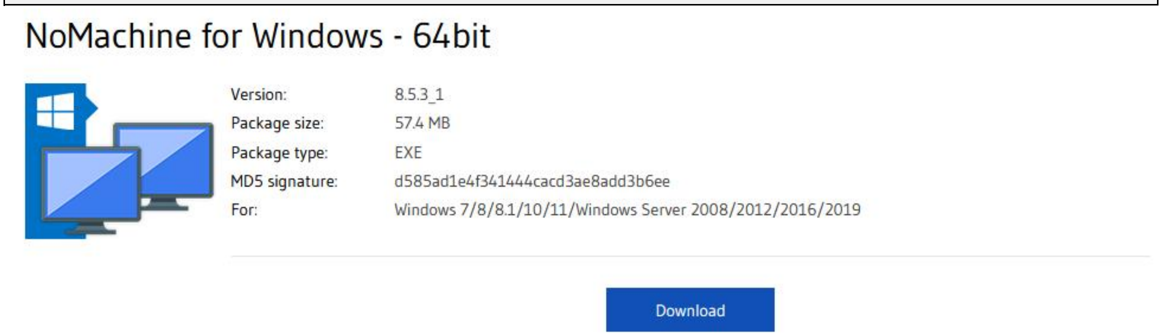
d. Linux 系 Then use the following command to install **NoMachine** in the Linux system of the development board

```
orangepi@orangepi:~$ sudo dpkg -i nomachine_x.x.x_x_arm64_arm64.deb
```

2) Then download the installation package of the Windows version of the NoMachine software. The download address is as follows

Note that this download link may change.

https://downloads.nomachine.com/download/?id=9

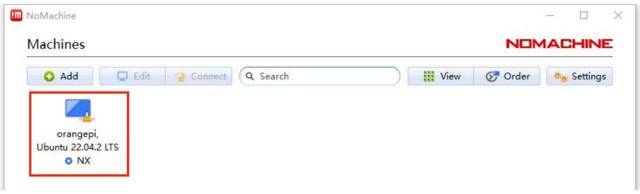


3) Then install NoMachine in Windows. **Please restart your computer after installation**

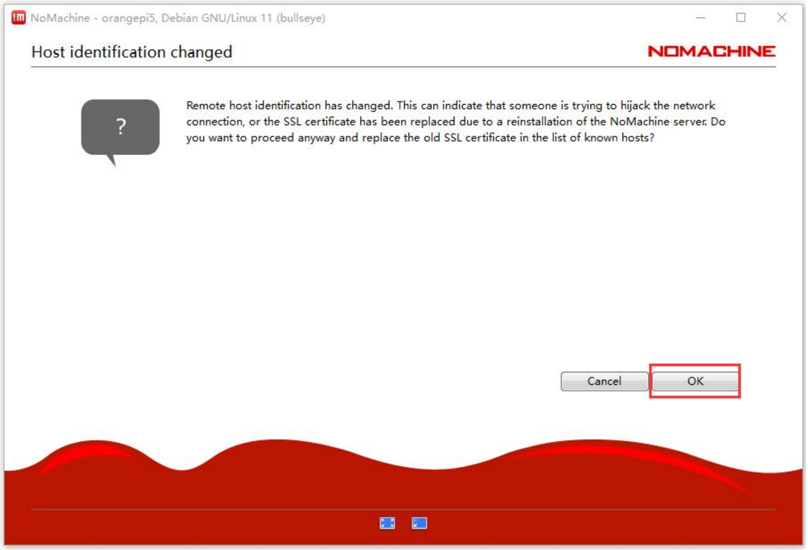
4) Then open **NoMachine** in Window

5) After NoMachine is started, it will automatically scan other devices with NoMachine

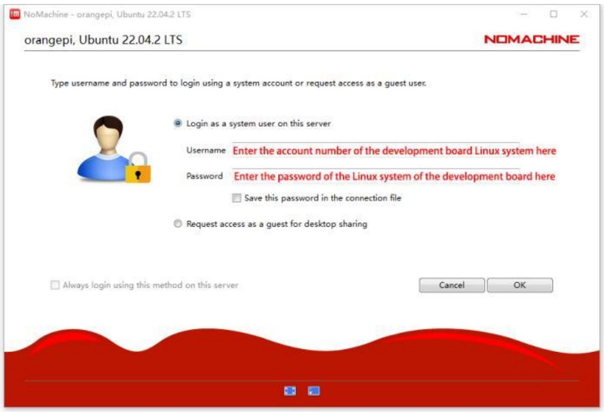
installed on the LAN. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices, and then click on the location shown in the red box in the picture below You can now log in to the Linux system desktop of the development board.



6) Then click **OK**

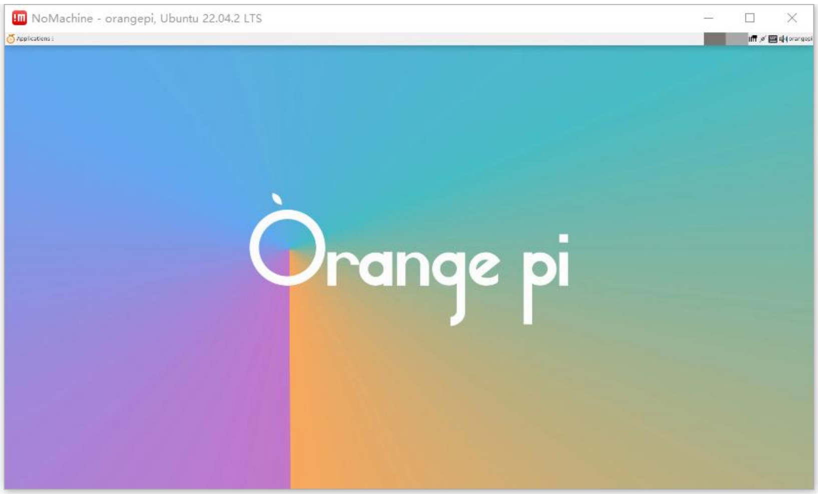


7) Then enter the username and password of the development board Linux system in the corresponding positions in the figure below, and then click **OK** to start logging in.



8) Then click OK in the next interface.

9) Finally you can see the desktop of the development board Linux system



3. 27. 2. Remote login using VNC

Before operating, please make sure that the Windwos computer and the development board are in the same LAN, and that you can log in to the Ubuntu or Debian system of the development board through ssh normally.

There are many problems with VNC testing in Ubuntu20.04, please do not use this method.

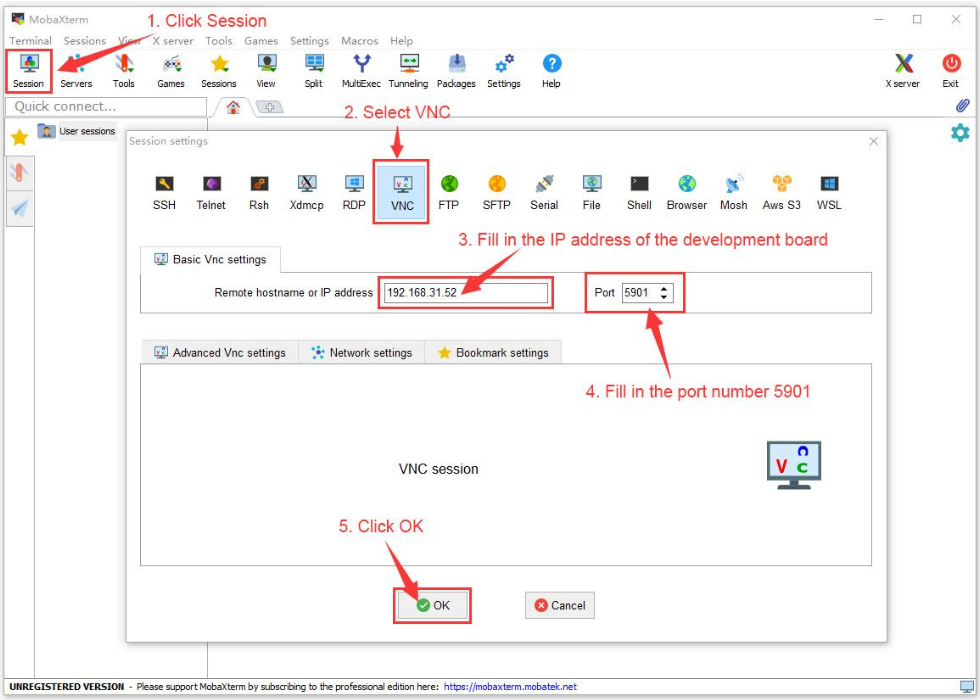
1) First run the **set_vnc.sh** script to set up vnc, **remember to add sudo permissions**

```
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 orangepi5pro:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/Orangepi5pro:1.log
```

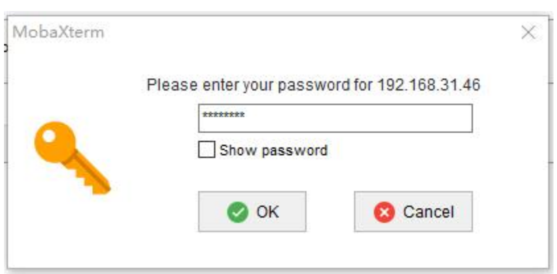
```
Killing Xtightvnc process ID 3047

New 'X' desktop is orangepi5pro:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi5pro:1.log
```

2) The steps to use MobaXterm software to connect to the development board Linux system desktop are as follows::

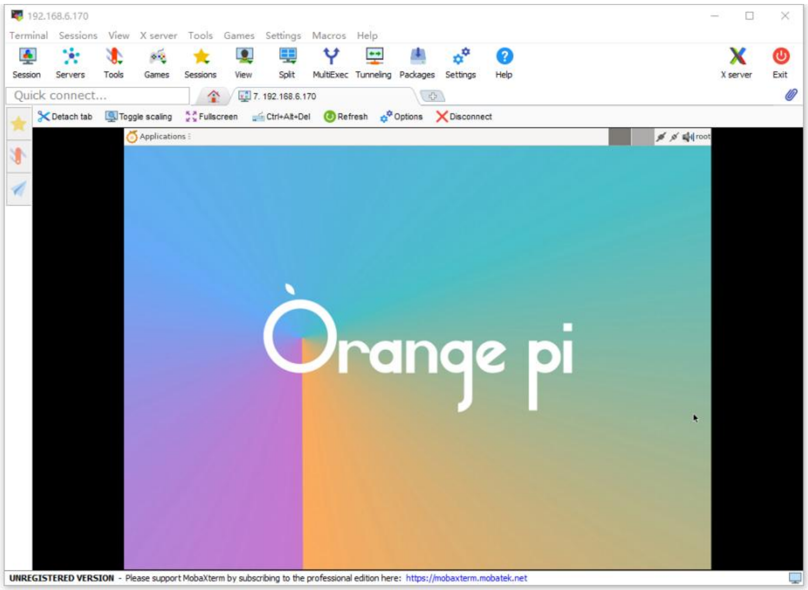
a. First click Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm.



b. Then enter the VNC password set earlier



c. After successful login, the interface is displayed as shown below, and then you can remotely operate the desktop of the development board Linux system.



3. 28. Testing of some programming languages supported by Linux system

3. 28. 1. Debian Bullseye system

1) Debian Bullseye 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 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 Java version

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

c. Write a **hello_world.java** of Java version

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
```

```
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 28. 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 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 a Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 28. 3. Ubuntu Focal system

1) Ubuntu Focal 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 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 have Java compilation tools and running environment installed

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 Java version.

```
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 a Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 28. 4. Ubuntu Jammy system

4) Ubuntu Jammy 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 gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>
```

```
int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

5) Ubuntu Jammy has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.12 (main, Sep 11 2024, 15:47:36) [GCC 11.4.0] 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!
```

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 Java version.

```
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 a Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
```

```
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 29. QT installation method

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

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

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

a. The qt version that comes with Ubuntu20.04 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 Ubuntu22.04 is **5.15.3**

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

c. The QT version that comes with Debian11 is **5.15.2**

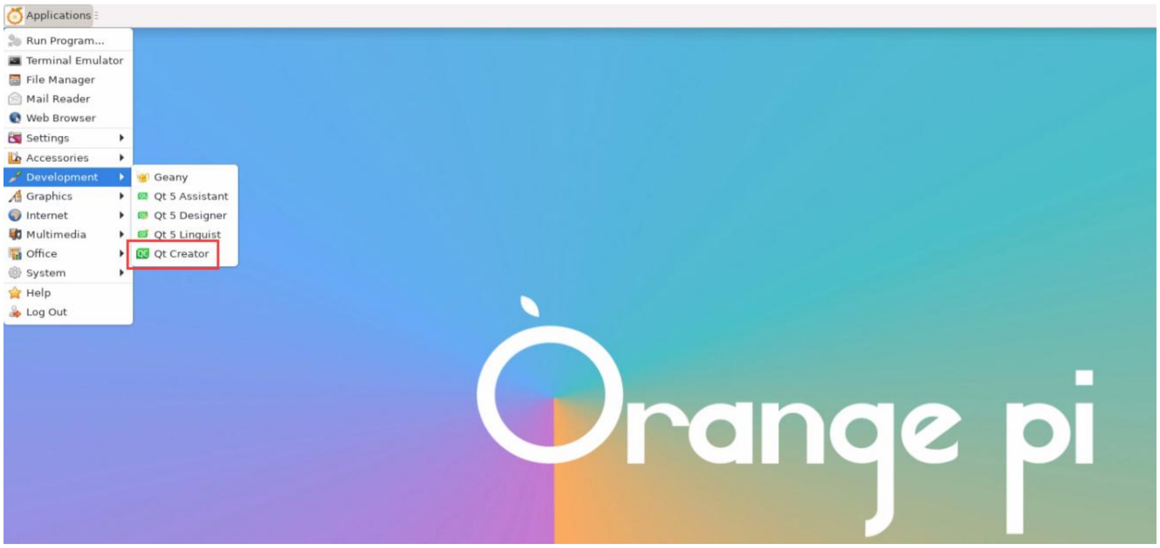
```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

d. The QT version that comes with Debian12 is **5.15.8**

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

```
......
QMake version 3.1
Using Qt version 5.15.8 in /usr/lib/aarch64-linux-gnu
```

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



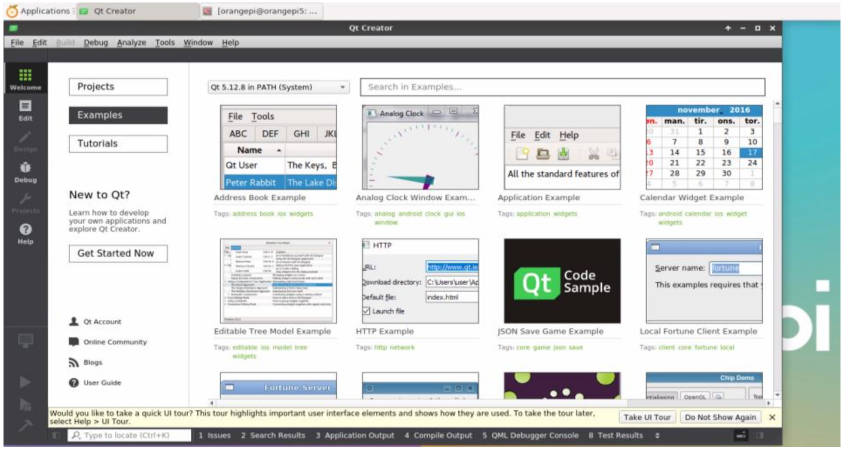
You can also use the following command to open QT Creator

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

During the startup process of QT and QT applications, if the following error is prompted, please ignore it. This error will not affect the operation of the application.

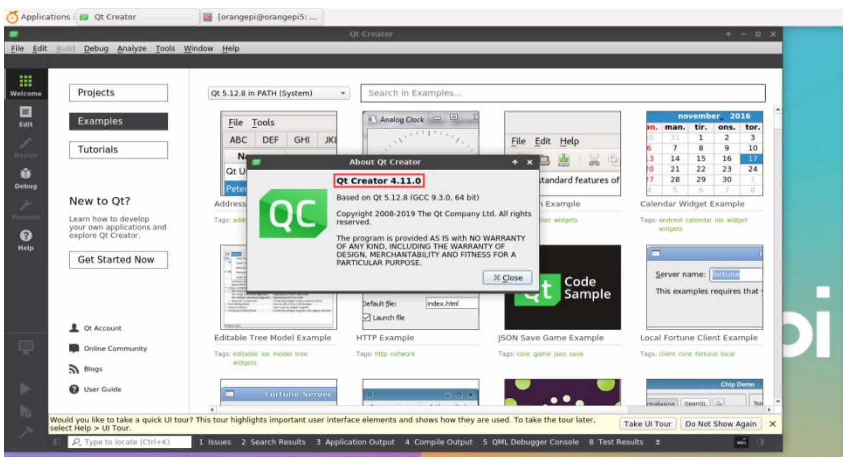
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip

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

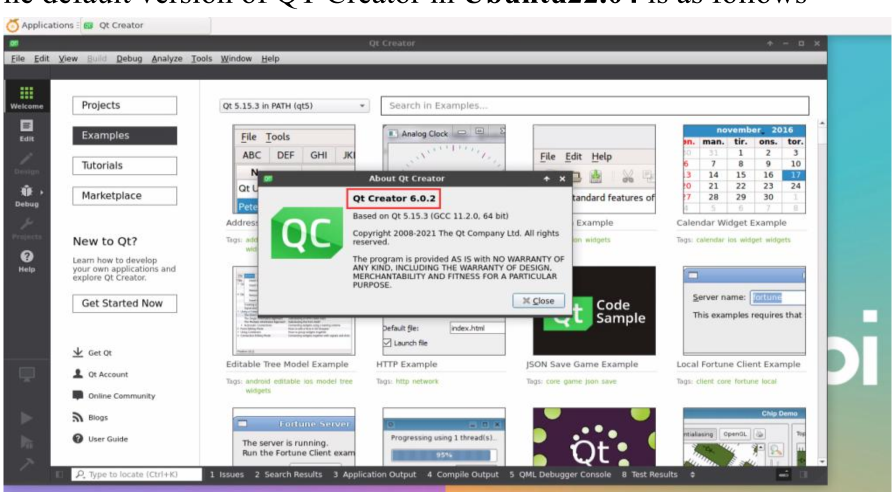


5) The version of QT Creator is as follows

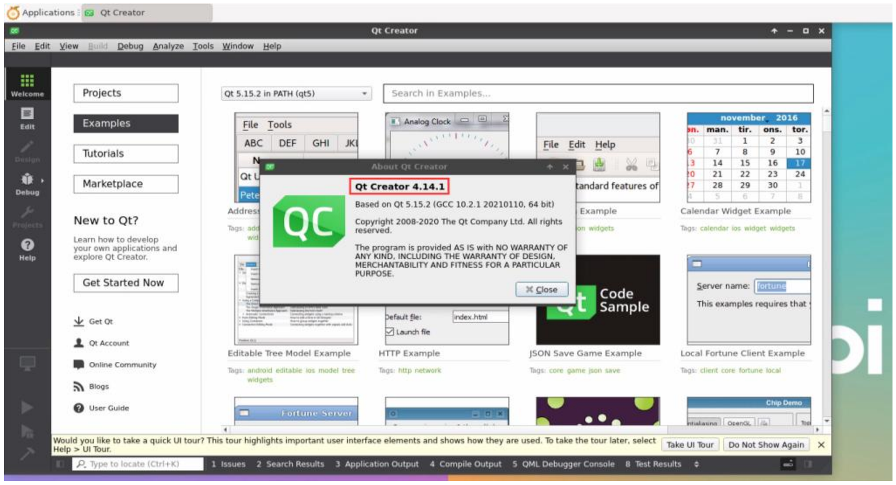
a. The default version of QT Creator in **Ubuntu20.04** is as follows



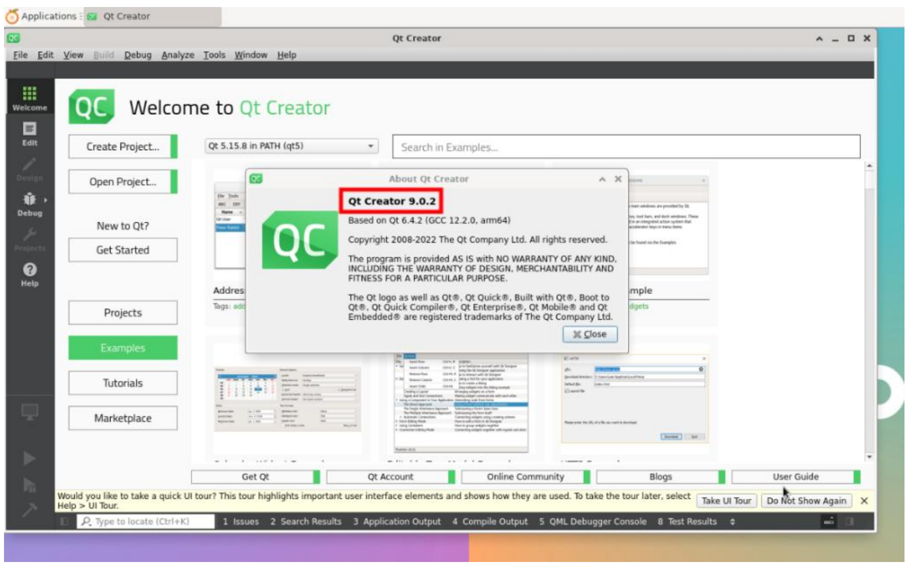
b. The default version of QT Creator in **Ubuntu22.04** is as follows



c. The default version of QT Creator in **Debian11** is as follows

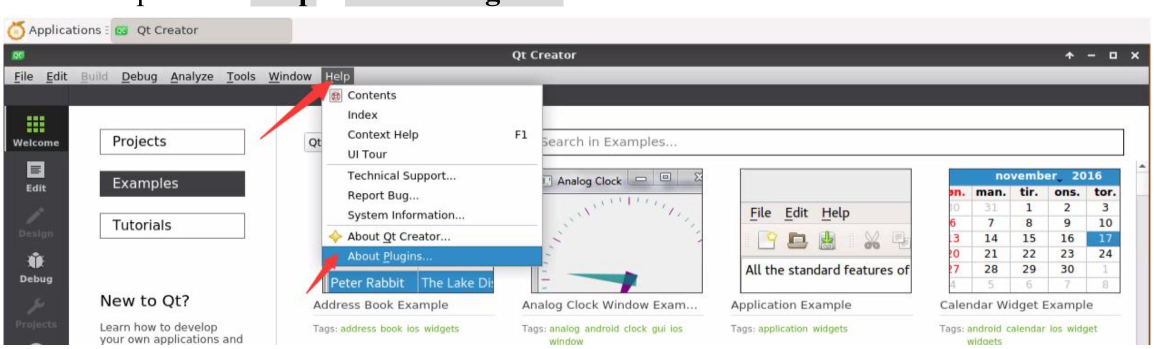


d. The default version of QT Creator in **Debian12** is as follows

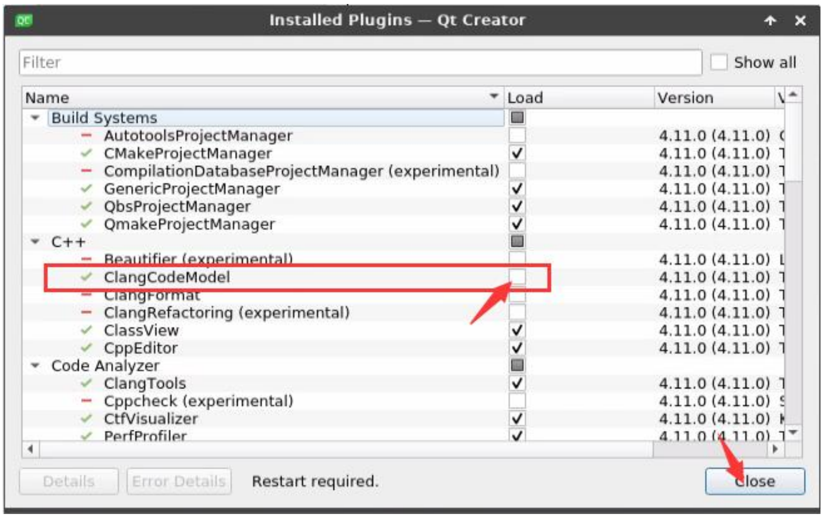


6) Then set up QT

a. Open first **Help**->**About Plugins...**



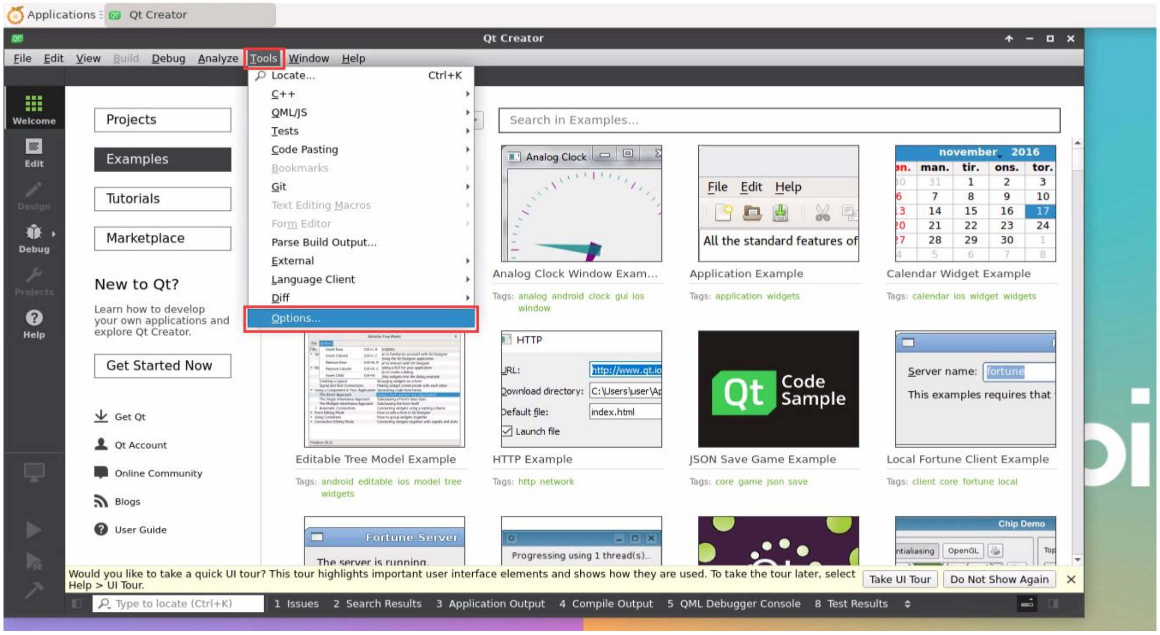
b. Then remove the check mark of **ClangCodeModel**

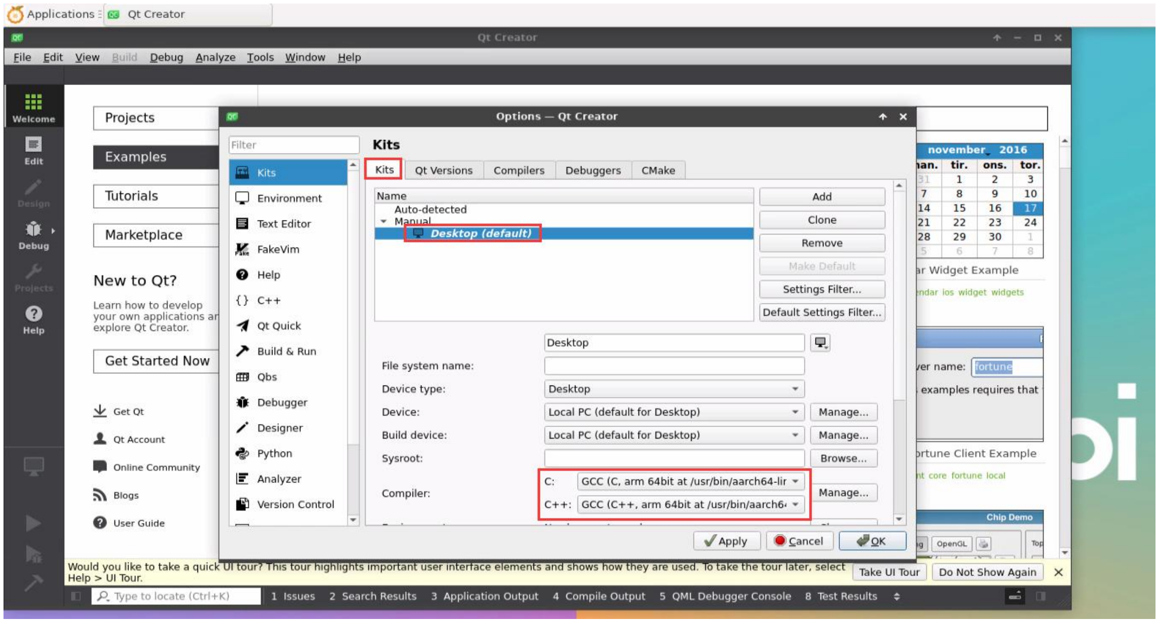


c. **After setting up, you need to restart QT Creator**

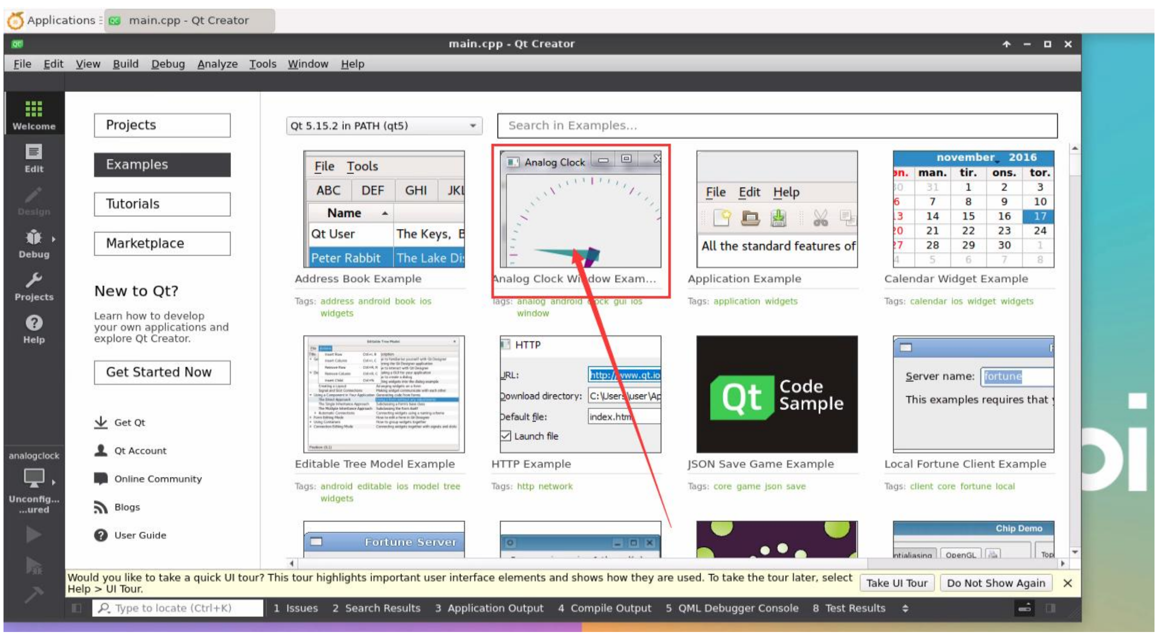
d. Then make sure the GCC compiler used by QT Creator. If the default is Clang, please change it to GCC.

Debian12 please skip this step.

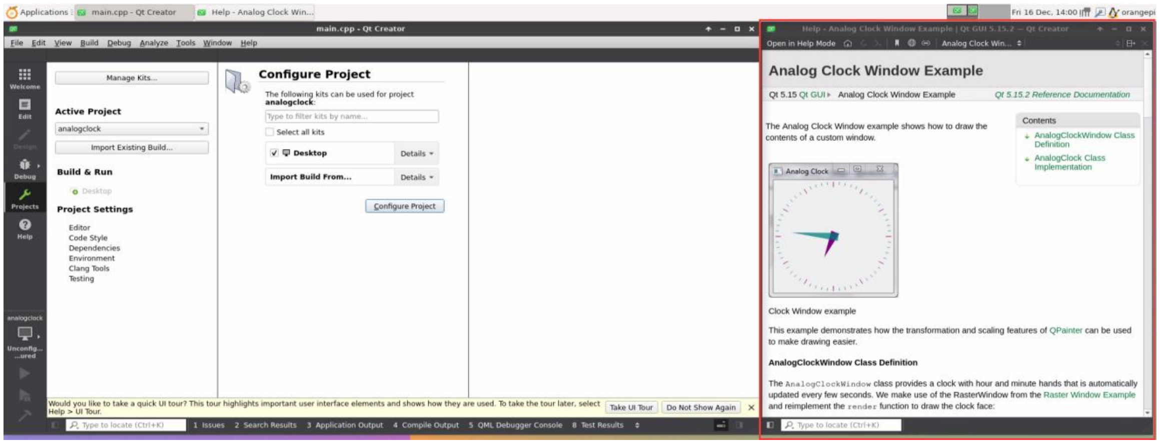




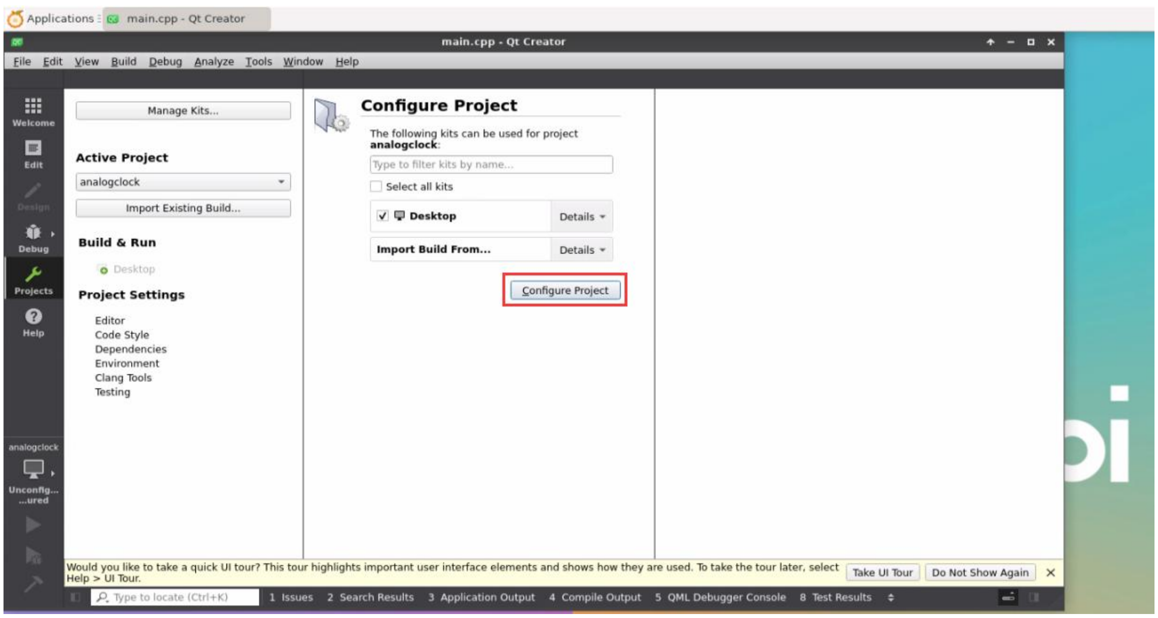
7) Then you can open a sample code



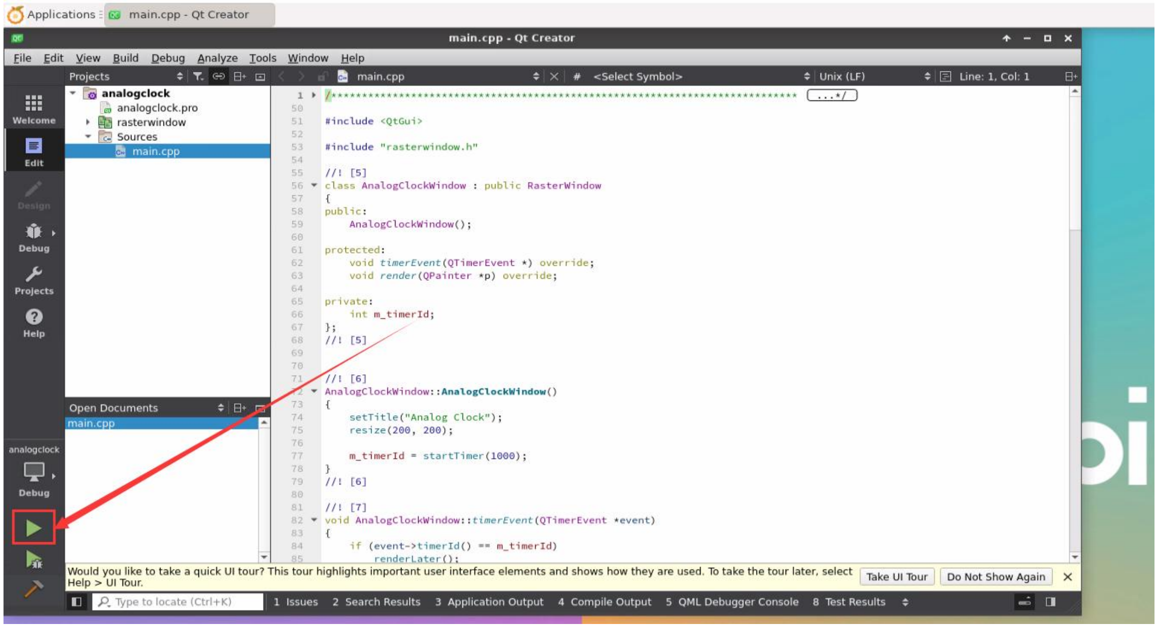
8) After clicking on the sample code, the corresponding instruction document will automatically open. You can read the instructions carefully.



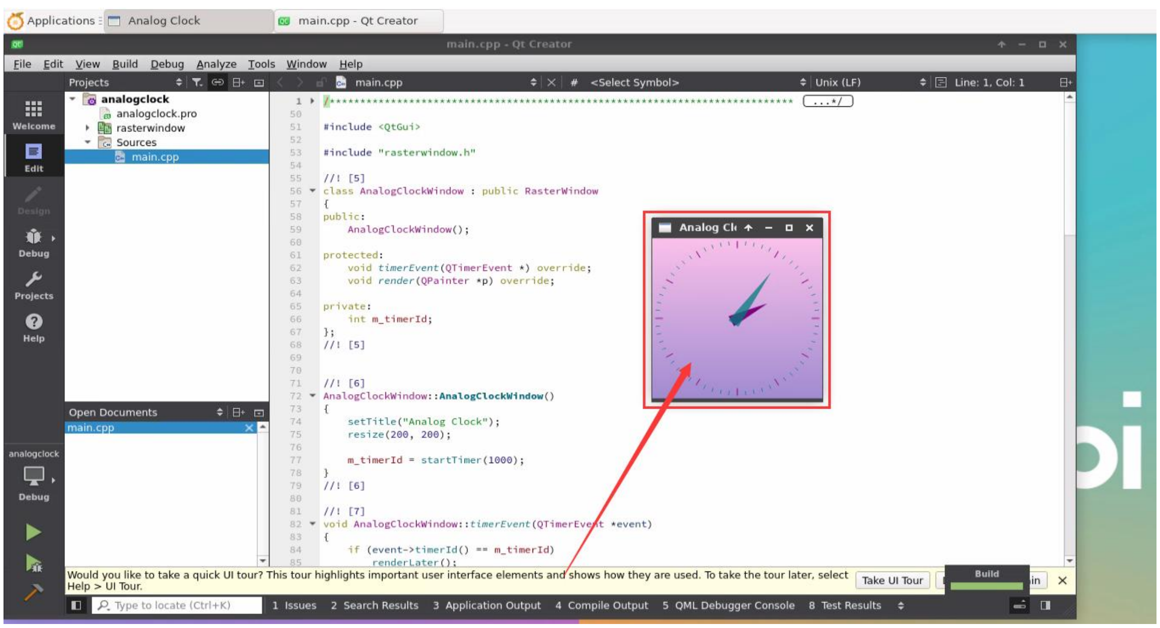
9) Then click **Configure Project**



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



11) After waiting for a period of time, the interface shown in the figure below will pop up, which means that QT can compile and run normally.



12) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu

https://download.qt.io/archive/qtcreator

https://download.qt.io/archive/qt

3. 30. ROS installation method

3. 30. 1. How to install ROS 1 Noetic on Ubuntu20.04

1) The currently 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 link to the official installation documentation of ROS 1 **Noetic Ninjemys** is as follows：

http://wiki.ros.org/noetic/Installation/Ubuntu

3) In the official installation documentation of ROS Noetic Ninjemys, Ubuntu recommends using Ubuntu20.04, so please ensure that the system used by the

development board is **Ubuntu20.04 desktop system**.

http://wiki.ros.org/noetic/Installation

4) Then use the script below to install ros1

```
orangepi@orangepi5pro:~$ install_ros.sh ros1
```

5) Before using the ROS tool, you first need to initialize rosdep. Then when compiling the source code, you can quickly install some system dependencies and some core components in ROS.

Note that when running the following command, you need 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 that github can be accessed normally every time. If install_ros.sh prompts the following error after installing ros1, please find other ways to allow the linux system of the development board to access github normally, and then manually run the following 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

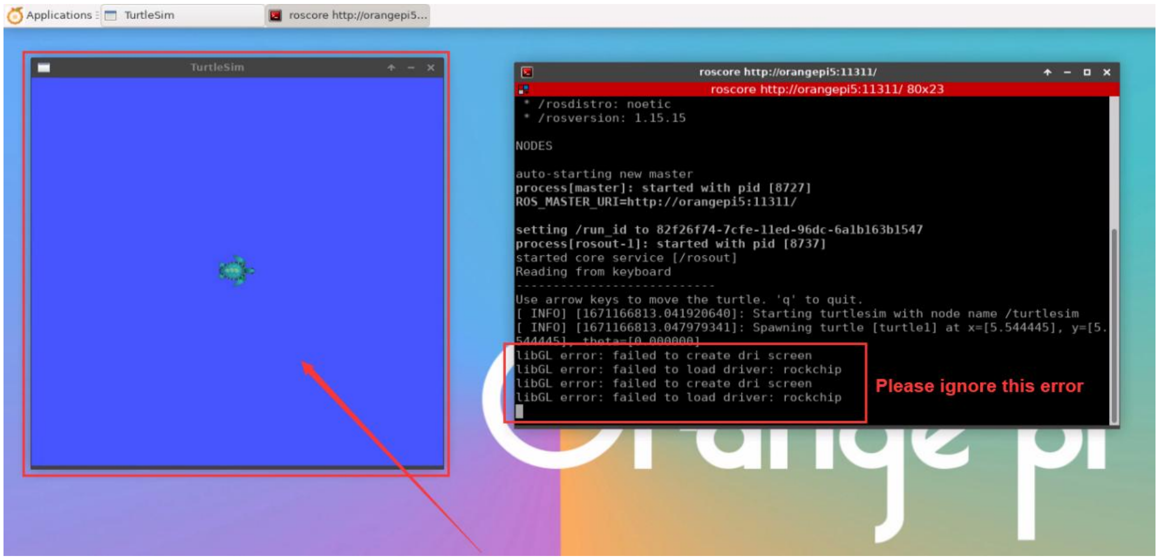
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run
```

```
        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
```

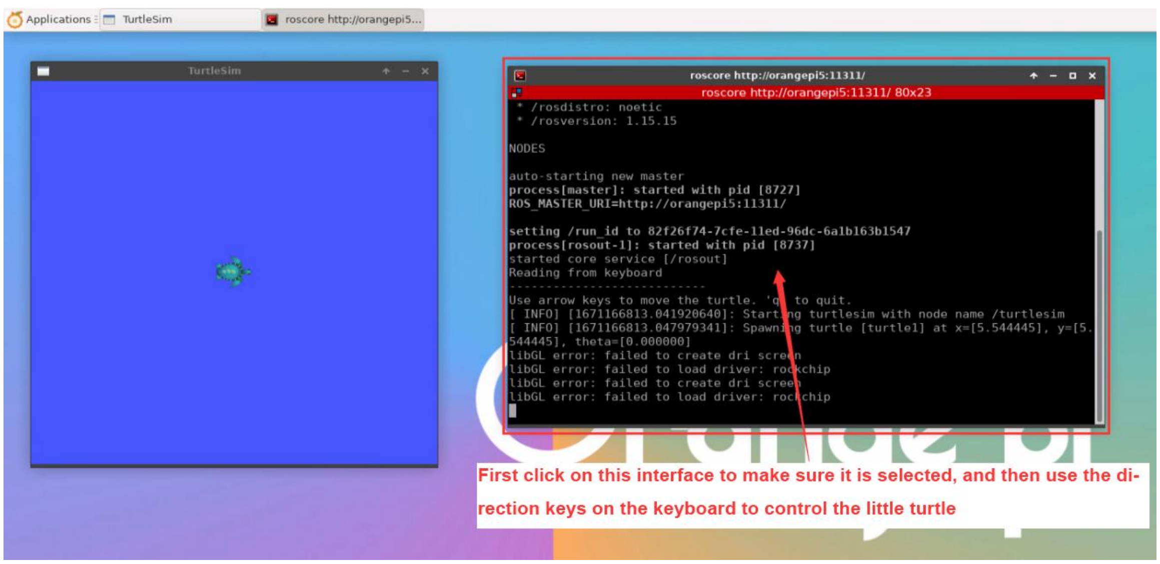
6) Then open a command line terminal window on the **desktop**, and then use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally.

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

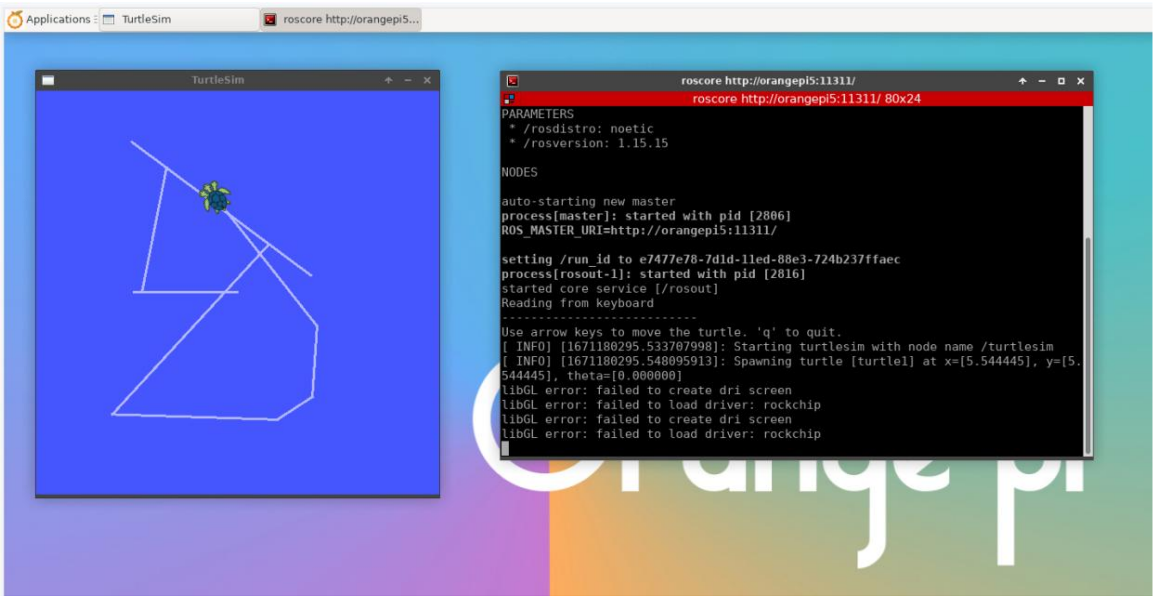
7) After running the **test_ros.sh** script, a small turtle as shown in the picture below will pop up.



8) Then please keep the terminal window you just opened at the top

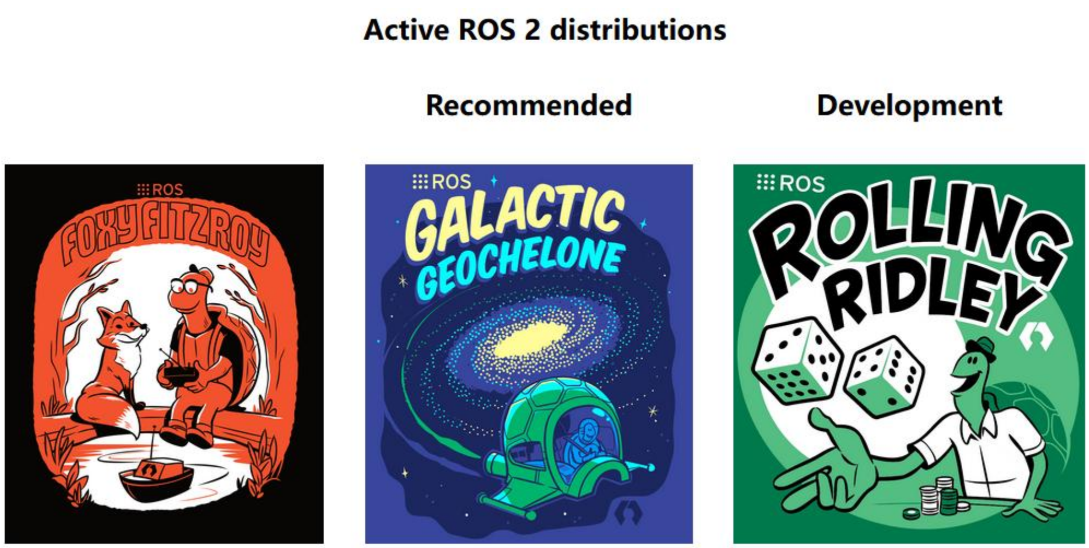


9) At this time, press the direction keys on the keyboard to control the little turtle to move up, down, left, and right.



3. 30. 2. How to install ROS 2 Galactic on Ubuntu20.04

1) The currently 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 installation documentation of ROS 2 **Galactic Geochelone** is as follows:

docs.ros.org/en/galactic/Installation.html
http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html

3) In the official installation documentation of ROS 2 **Galactic Geochelone**, Ubuntu Linux recommends using Ubuntu20.04, so please ensure that the system used by the development board is the Ubuntu20.04 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) The **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete.

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

ros2 is an extensible command-line tool for ROS 2.
```

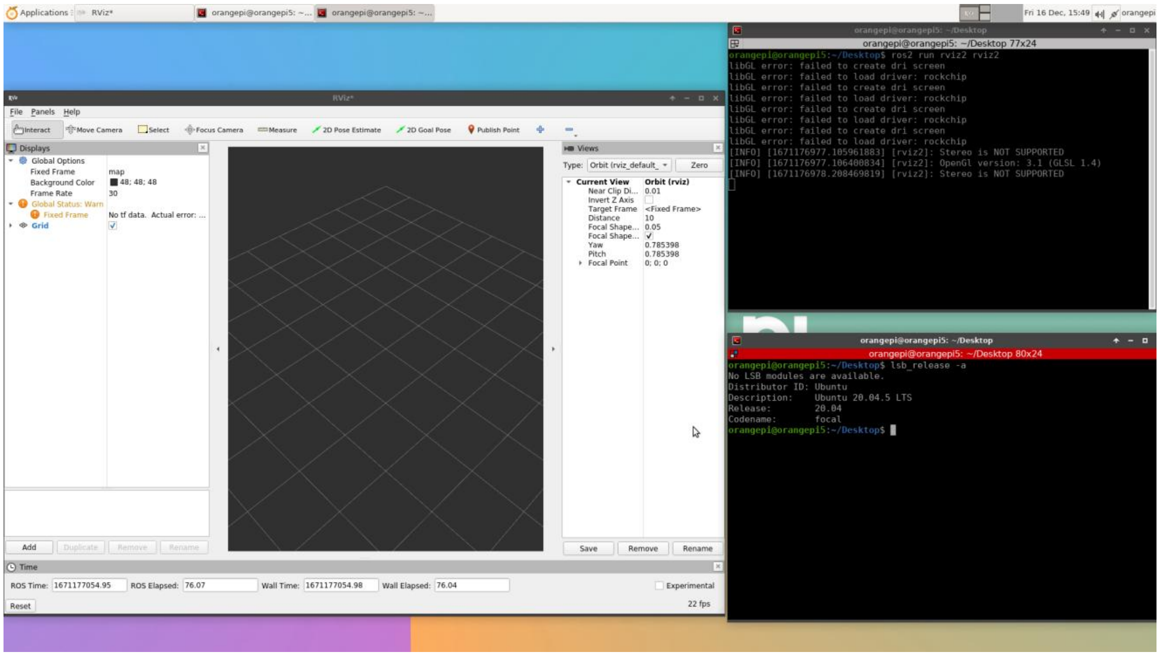
```
optional arguments:
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

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

6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means ROS 2 can run normally.

```
orangepi@orangepi5pro:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



8) For how to use ROS, please refer to the documentation of ROS 2.

http://docs.ros.org/en/galactic/Tutorials.html

3. 30. 3. How to install ROS 2 Humble on Ubuntu22.04

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

```
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
```

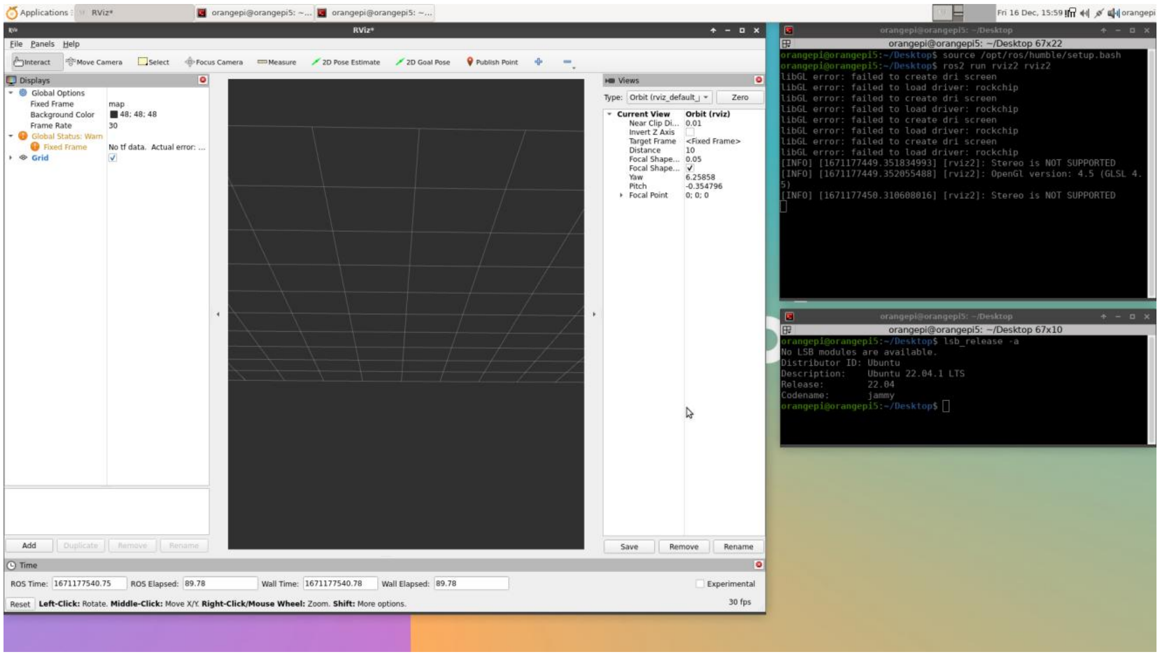
```
component    Various component related sub-commands
daemon       Various daemon related sub-commands
doctor       Check ROS setup and other potential issues
interface    Show information about ROS interfaces
launch       Run a launch file
lifecycle    Various lifecycle related sub-commands
multicast    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 successfully installed. If you can see the following print, it means ROS 2 can run normally.

```
orangepi@orangepi5pro:~$ 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 documentation

http://docs.ros.org/en/humble/index.html
http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

3. 31. How to install kernel header files

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**
linux-headers-5.10.160-rockchip-rk3588

4) Then you can write a hello kernel module to test the kernel header file

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");
```

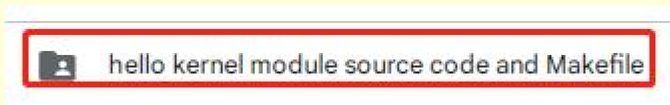
b. Then write the Makefile that compiles the hello kernel module, as shown below:

```
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. The output of the compilation process is as follows：

If you have problems compiling the code you copied here, please download the source code from the official tool and upload it to the Linux system of the development board for testing.




```
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 compilation, the **hello.k** 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 following output, it means that 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. 32. 10.1-inch MIPI LCD screen usage method

3. 32. 1. 10.1 inch MIPI screen assembly method

1) First prepare the necessary accessories

a. 10.1 inch MIPI LCD display + touch screen

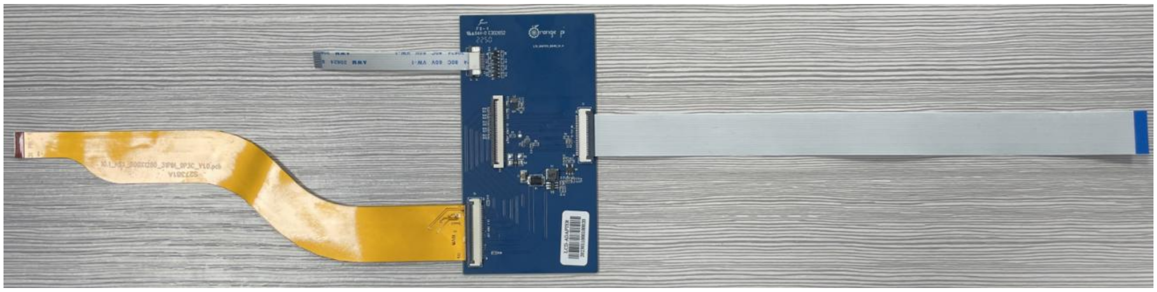
b. Screen adapter board + 31pin to 40pin cable

c. 30pin MIPI cable

d. 12pin touch screen cable

2) Connect the 12pin touch screen cable, 31pin to 40pin cable, and 30pin MIPI cable to the screen adapter board as shown below. **Note that the blue insulated side of the touch screen cable is facing down**, and the insulated side of the other two cables is facing up. ,

if connected incorrectly, it will cause no display or inability to touch.



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

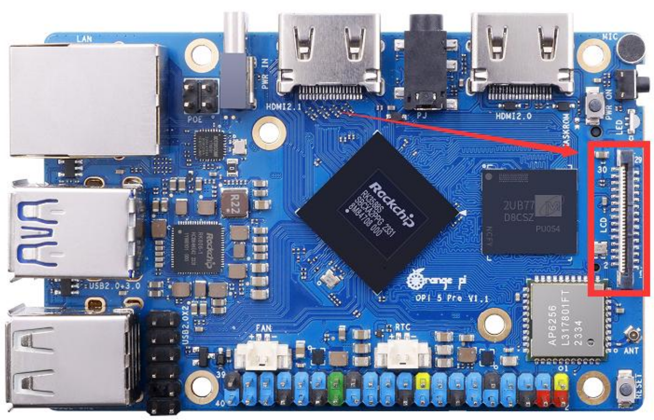
4) Then connect the touch screen and the adapter board through the 12pin touch screen cable, paying attention to the orientation of the insulation surface

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

3. 32. 2. How to open the 10.1-inch MIPI LCD screen configuration

1) The Linux image does not have the configuration to open the mipi LCD screen by default. If you need to use the mipi LCD screen, you need to open it manually.

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

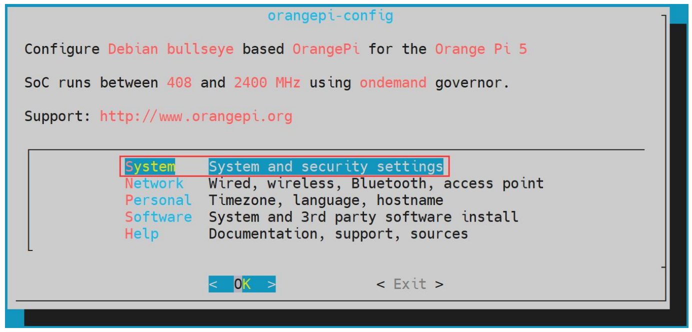


3) The steps to open mipi lcd configuration are as follows：

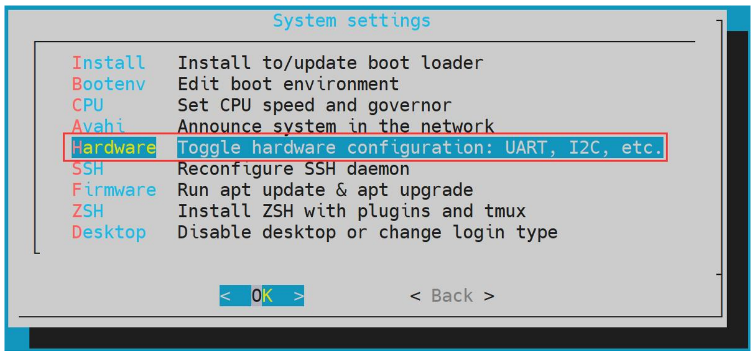
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions.

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

b. Then select **System**



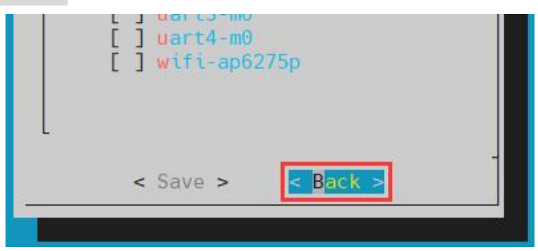
c. Then select **Hardware**



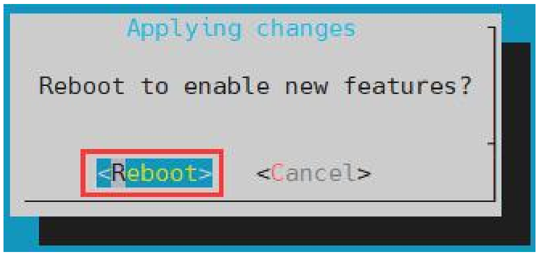
d. Then use the keyboard's arrow keys to locate **opi5pro-lcd** and then use the **space** to select it.

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 overlays=opi5pro-lcd to /boot/orangepiEnv.txt. You can check it first after setting it up. If this line of configuration does not exist, there is something wrong with the settings.

If you find it troublesome to use orangepi-config, you can also use the vim editor to open /boot/orangepiEnv.txt, and then add the line configuration overlays=opi5pro-lcd.

```
orangepi@orangepi:~$ cat /boot/orangepiEnv.txt | grep "lcd"
overlays=opi5pro-lcd           #Example configuration
```

4) After startup, you can see the LCD screen display as follows (default is vertical screen):

3. 32. 3. How to rotate the display direction of the server version image

1) Add **extraargs=fbcon=rotate: the direction of rotation** to **/boot/orangepiEnv.txt** to set the direction displayed by the server version of Linux system, where **fbcon=rotate**: The following number can be set to:

a. 0: Normal screen (default is vertical screen)
b. 1:Turn clockwise 90 degrees
c. 2:Flip 180 degrees
d. 3:Rotate 270 degrees clockwise

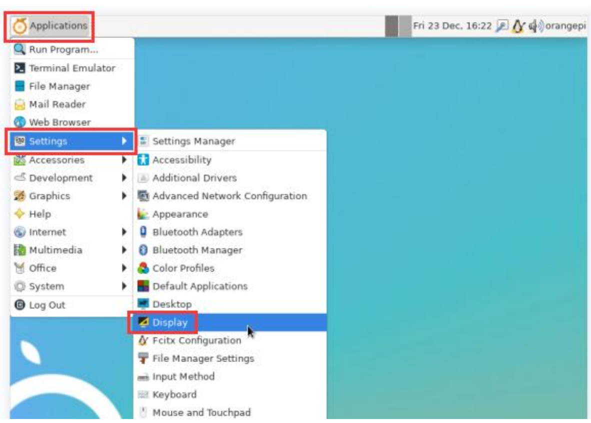
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=lcd1
extraargs=cma=64M fbcon=rotate:3
```

Note that if the configuration line extraargs=cma=64M is included by default in /boot/orangepiEnv.txt, the configuration fbcon=rotate:3 can be added to the end of extraargs=cma=64M (need to be separated by spaces)。

2) Then **restart** the Linux system and you will see that the direction of the LCD screen display has rotated.

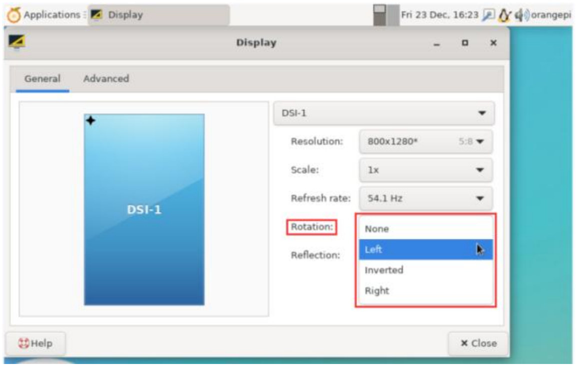
3. 32. 4. How to mirror the rotation display and touch direction of the desktop version

1) First open the **Display** settings in the linux system

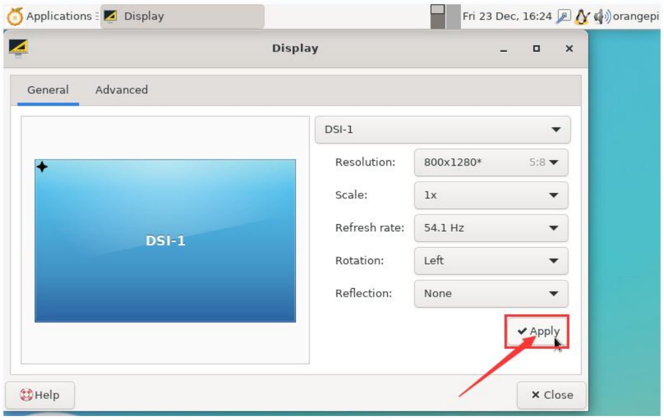


2) Then select the direction you want to rotate in **Rotation**

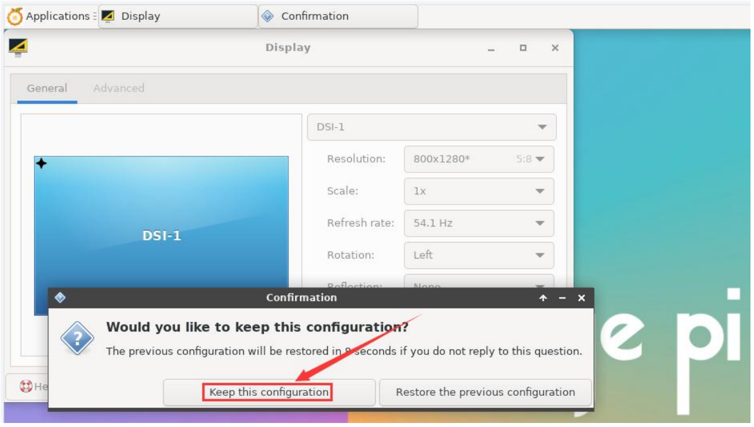
a. **None**: No rotation
b. **Left**: Rotate 90 degrees to the left
c. **Inverted**: Flip up and down, equivalent to rotating 180 degrees
d. **Right**: Rotate 90 degrees to the right



3) Then click **Apply**



4) Then select **Keep this configuration**



5) At this point, the screen display has been rotated, and then close the **Display** program.

6) The above steps will only select the display direction, and will not rotate the touch direction. Use the **set_lcd_rotate.sh** script to rotate the touch direction. After this script is set, it will automatically restart, and then you can test whether the touch can be used

normally

a. **None**: No rotation

```
orangepi@orangepi:~$ set_lcd_rotate.sh none
```

b. **Left**: Rotate 90 degrees to the left

```
orangepi@orangepi:~$ set_lcd_rotate.sh left
```

c. **Inverted**: Flip 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
```

The set_lcd_rotate.sh script mainly does four things:

1. **Rotate the direction of framebuffer display**
2. **Rotate the direction of touch**
3. **Turn off the startup logo**
4. **Restart the system**

The direction of rotation touch is achieved by adding the Option "TransformationMatrix" "x x x x x x x x x" line configuration to /usr/share/X11/xorg.conf.d/40-libinput.conf. Among them, "x x x x x x x x x" has different configurations in different directions.

7) Touch to rotate reference material

```
https://wiki.ubuntu.com/X/InputCoordinateTransformation
```

3. 33. Instructions for using the logo on and off the machine

1) The power on/off logo will only be displayed on the desktop version of the system by default.

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the switch logo.

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

```
bootlogo=false
```

3) Set the **bootlogo** variable to **true** in **/boot/orangepiEnv.txt** to enable the power on/off logo.

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
```

4) The location of the boot logo picture in the Linux system is

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

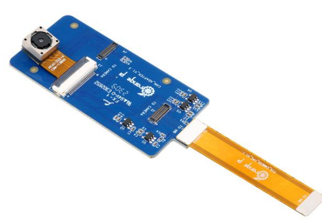
5) After replacing the boot logo image, you need to run the following command to take effect

```
orangepi@orangepi:~$ sudo update-initramfs -u
```

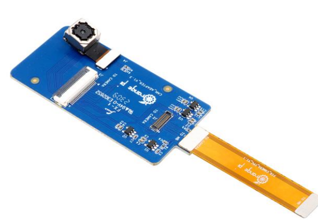
3. 34. Testing methods for OV13850 and OV13855 MIPI cameras

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



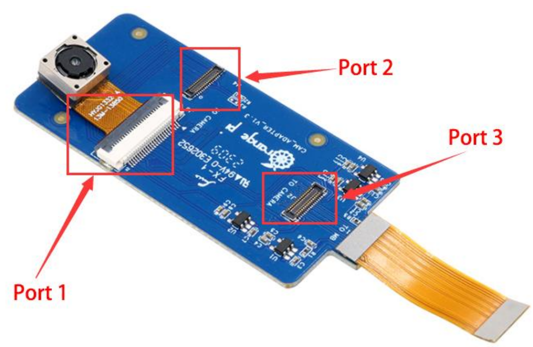
b. OV13855 camera with 13 million MIPI interface



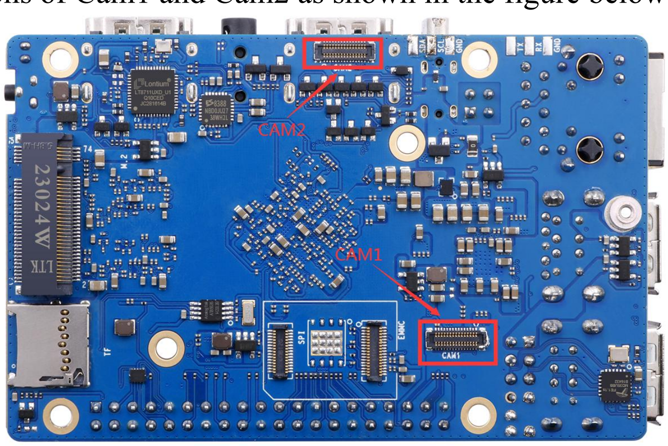
The adapter board and FPC cable used by the OV13850 and OV13855 cameras are the same, but the locations where the two cameras are connected to the adapter board are different. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked **TO MB** needs to be plugged into the camera interface of the development board, and the end marked **TO CAMERA** needs to be plugged into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board, and only one can be connected and used at the same time, as shown in the figure below, among which：

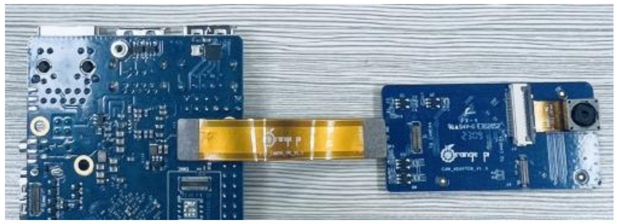
a. **Connect the OV13850 camera to interface 1**
b. **Connect the OV13855 camera to interface 2**
c. Interface No. 3 is not used, just ignore it.



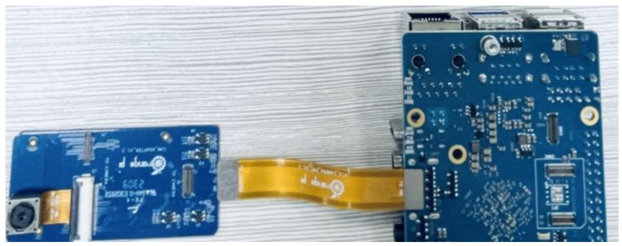
There are a total of 2 camera interfaces on the Orange Pi 5 Pro development board. We define the positions of Cam1 and Cam2 as shown in the figure below：



The method of plugging the camera into the Cam1 interface of the development board is as follows:



The method of plugging the camera into the Cam2 interface of the development board is as follows:

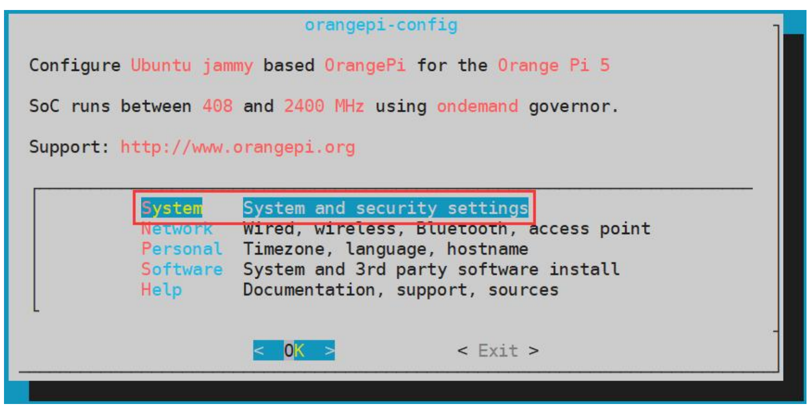


After connecting the camera to the development board, we can use the following method to test the camera:

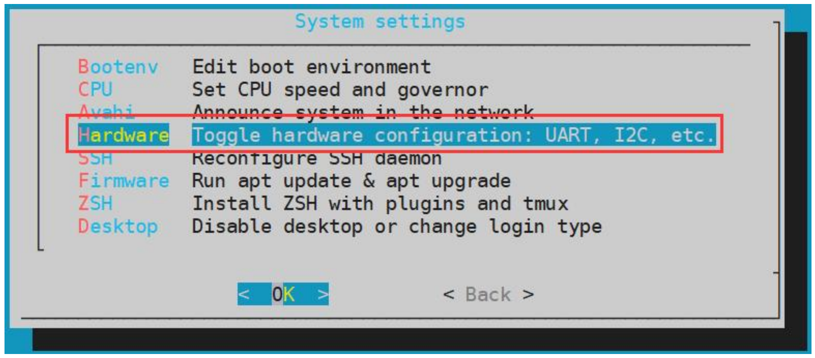
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions.

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

b. Then select **System**



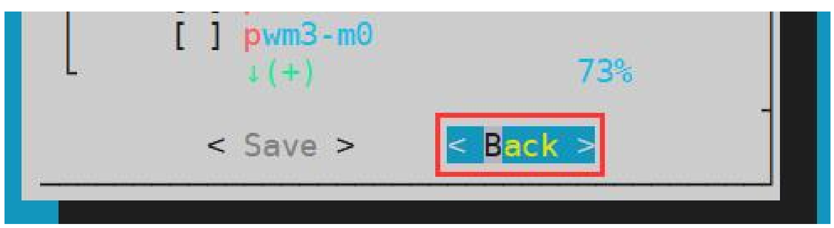
c. Then select **Hardware**



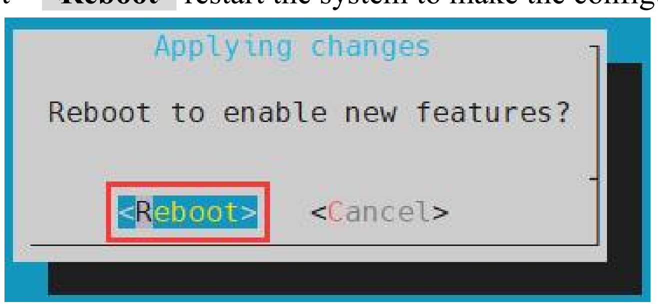
d. Then use the keyboard's arrow keys to locate the position shown in the figure below, and then use the space to select the camera you want to open, where **opi5pro-cam1** means using the ov13850 or ov13855 camera in the Cam1 interface of the development board, and **opi5pro-cam2** means Use the ov13850 or ov13855 camera in the Cam2 interface of the development board.

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

f. Then select **<Back>**



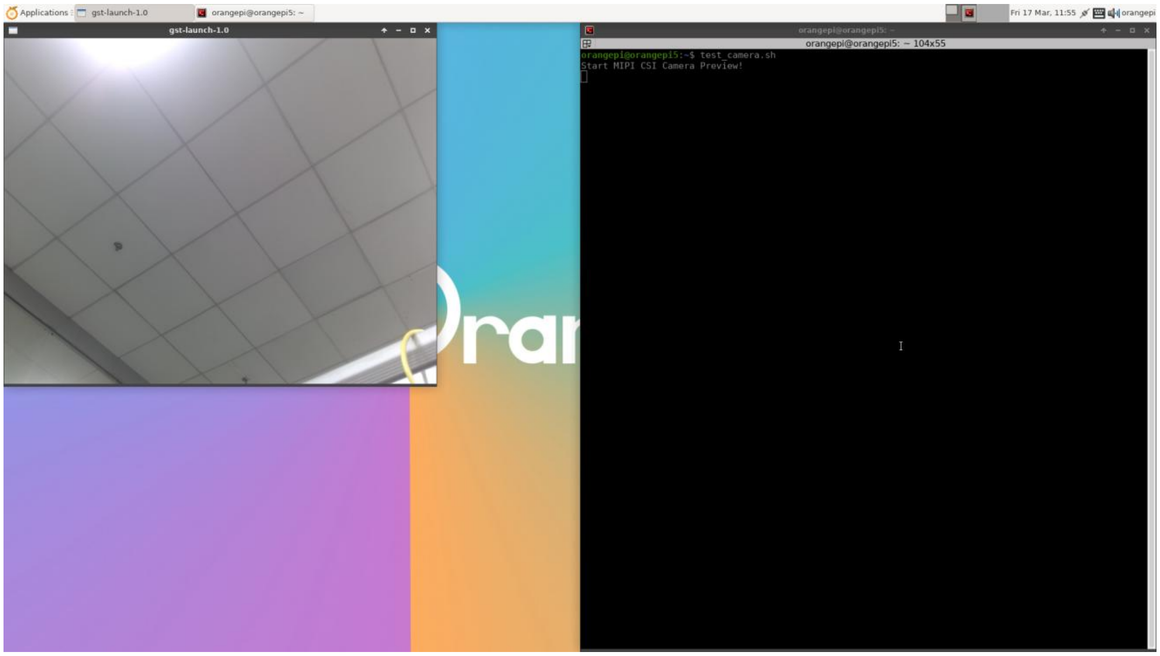
g. Then select **<Reboot>** restart the system to make the configuration take effect



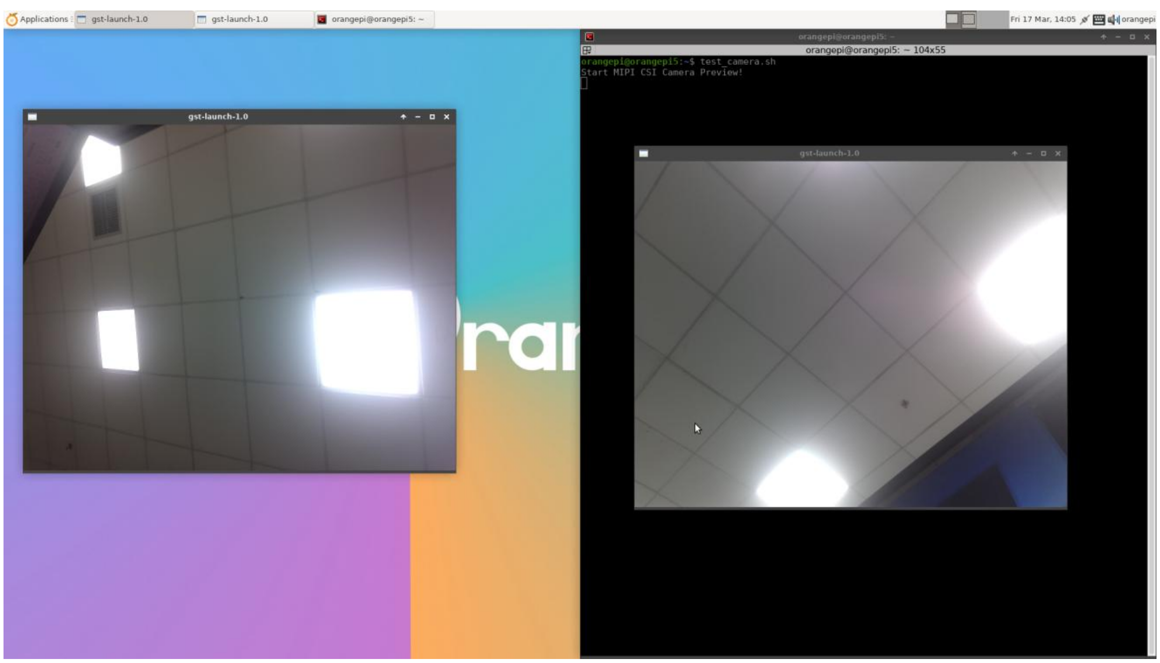
h. Then open a terminal on the desktop system and run the following script

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

i. Then you can see the preview screen of the camera



In addition to a single camera, we can also use two cameras at the same time (supporting ov13850 and ov13855 mix and match). After connecting the dual cameras, open the configuration of Cam1+Cam2 through **orangepi-config** as in the previous steps, then restart the system, then open the terminal on the desktop and run the **test_camera.sh** script to see the previews of the two cameras, as follows As shown in the figure:



Please refer to the link below for camera dts configuration. You can modify it yourself if necessary;

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-pro-camera1.dtsi

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-pro-camera2.dtsi

The configuration of dt overlay is in the following directory:
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/overlay

3. 35. How to use the ZFS file system

3. 35. 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
```

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, before installation, you need to ensure that the system has installed the deb package of the kernel header file.

orangepi@orangepi:~$ **sudo apt install -y zfsutils-linux zfs-dkms**

3. 35. 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 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. 35. 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. 35. 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 a relatively large plain text file for the compression test. 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. 36. 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.

3. 36. 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          # Output of Ubuntu Jammy system
```

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
```

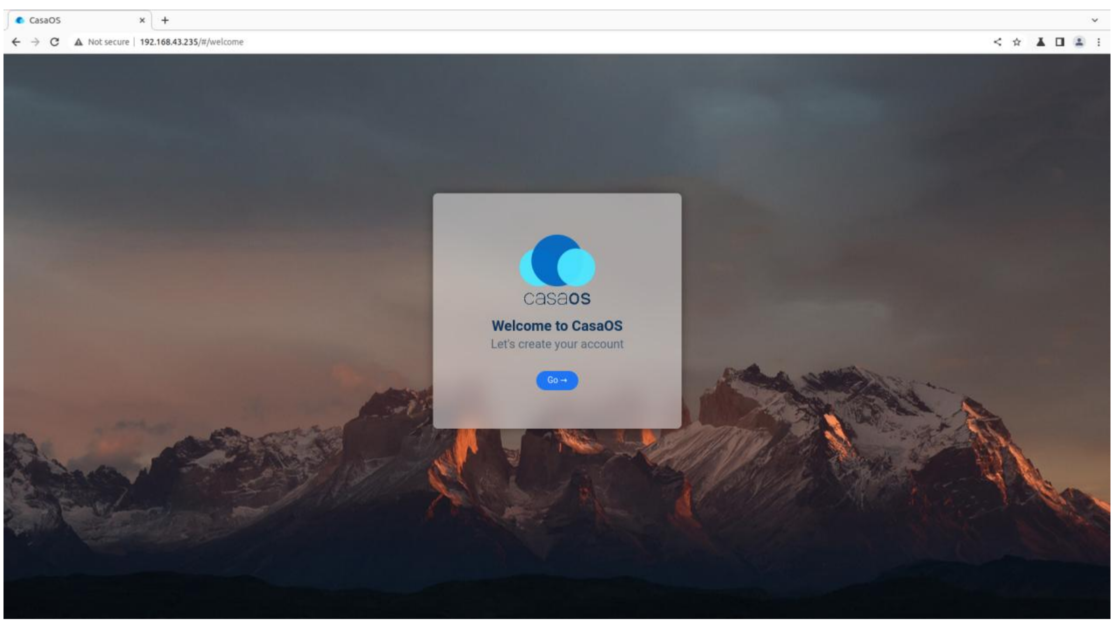
3) When you see the terminal outputting the following print information, it means that CasaOS has been installed.

```
CasaOS v0.4.4.2 is running at:

Open your browser and visit the above address.


CasaOS Project  : https://github.com/IceWhaleTech/CasaOS
CasaOS Team     : https://github.com/IceWhaleTech/CasaOS#maintainers
CasaOS Discord  : https://discord.gg/knqAbbBbeX
Website         : https://www.casaos.io
Online Demo     : http://demo.casaos.io

Uninstall       : casaos-uninstall
```

3. 36. 2. How to use CasaOS

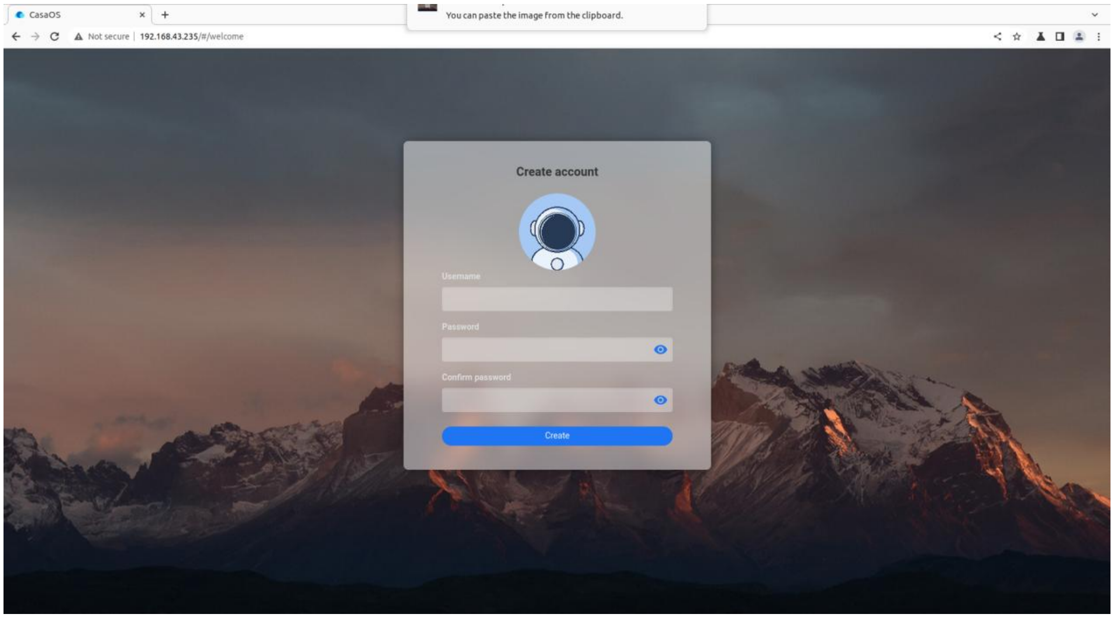
1) After installing CasaOS, enter **http://the IP address** of the development board in the browser to open CasaOS

2) After opening CasaO, the following welcome interface will pop up. Click "Go" to proceed to the next step.

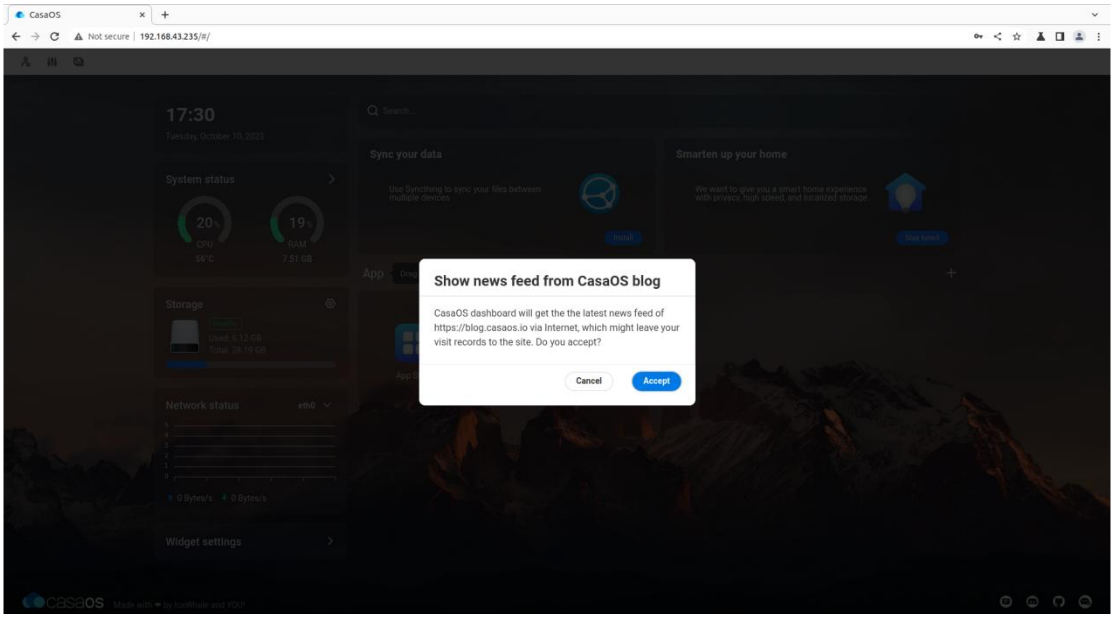


3) When you log in to CasaOS for the first time, the login interface is the interface for setting the account and password. When you log in again, only the interface for entering the account and password will appear. After setting the account and password, click

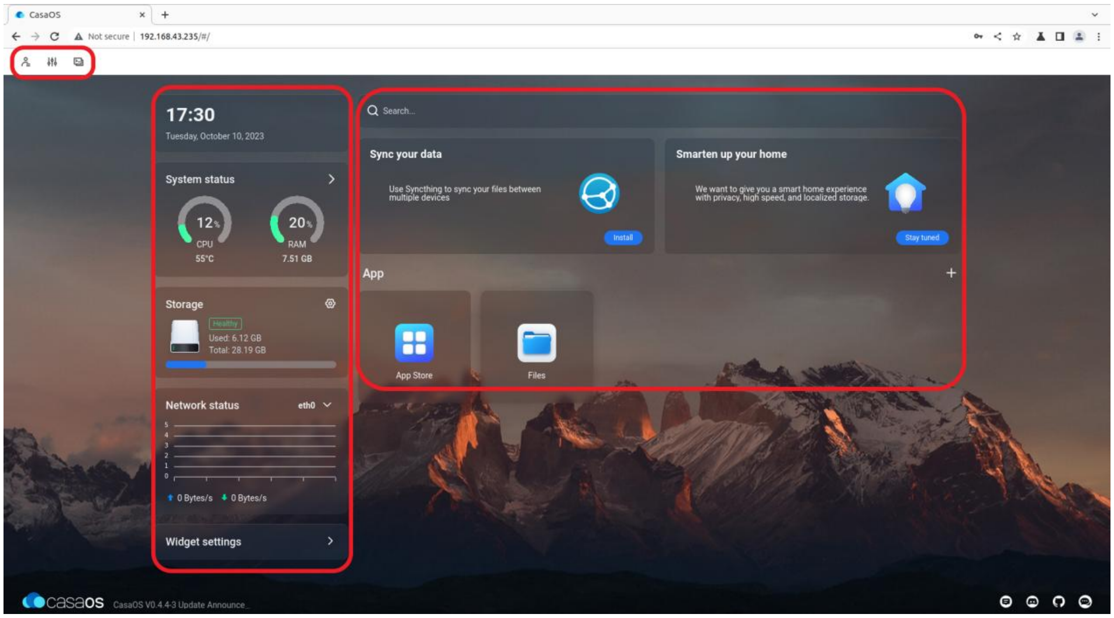
"Create" to proceed to the next step.



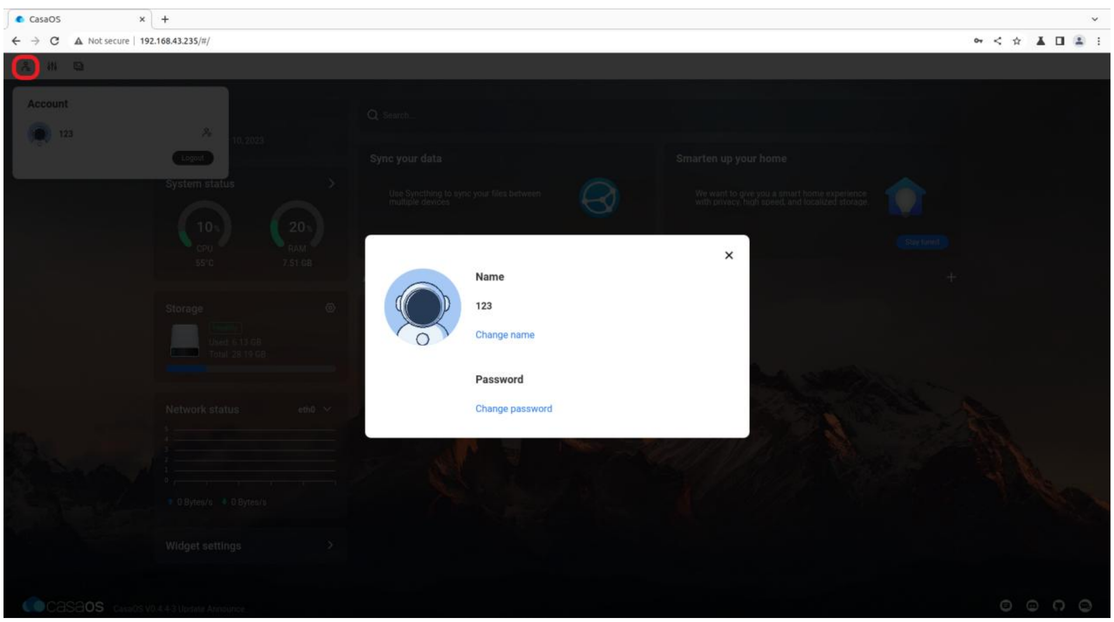
4) Click "Accept" directly in the interface below to proceed to the next step.



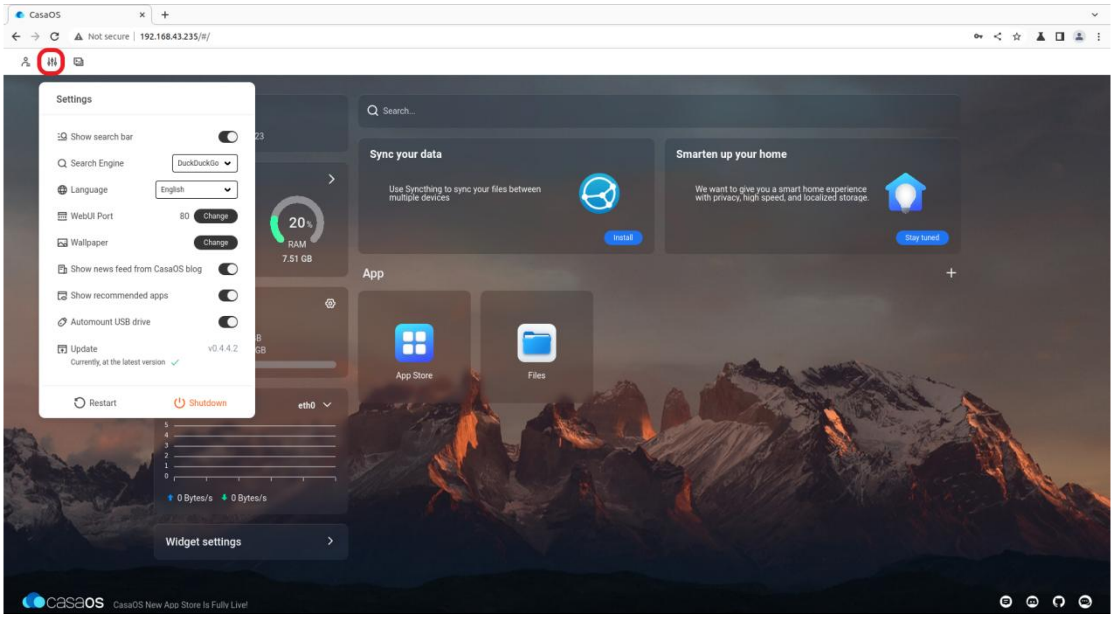
5) Now enter the main page of CasaOS. There are three icons in the upper left corner for function settings. On the left is the performance panel, which can display the current time and status information of CPU, RAM, storage, and network. On the right is the function panel. It has functions such as search, application recommendation, application store and file management.



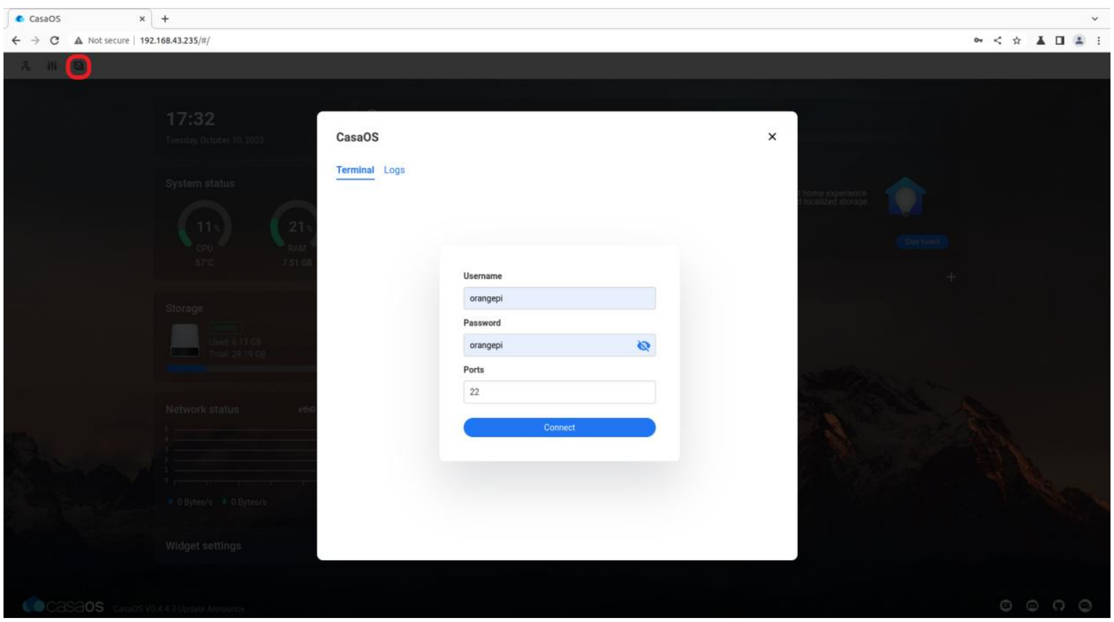
6) You can click the first icon in the upper left corner to modify the account number and password



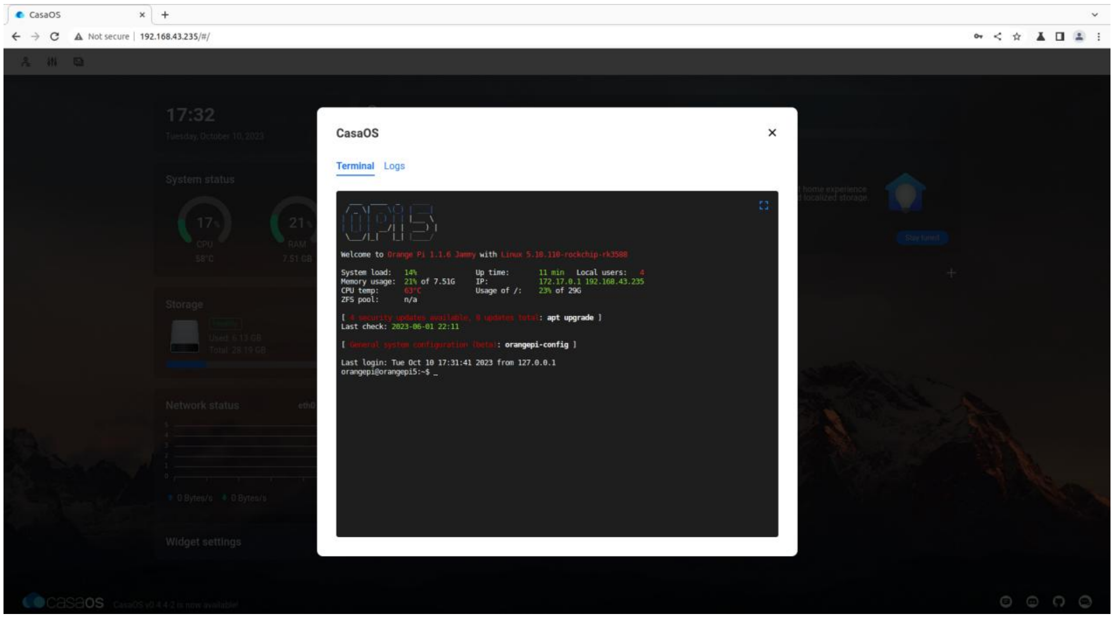
7) You can click the second icon to set basic functions



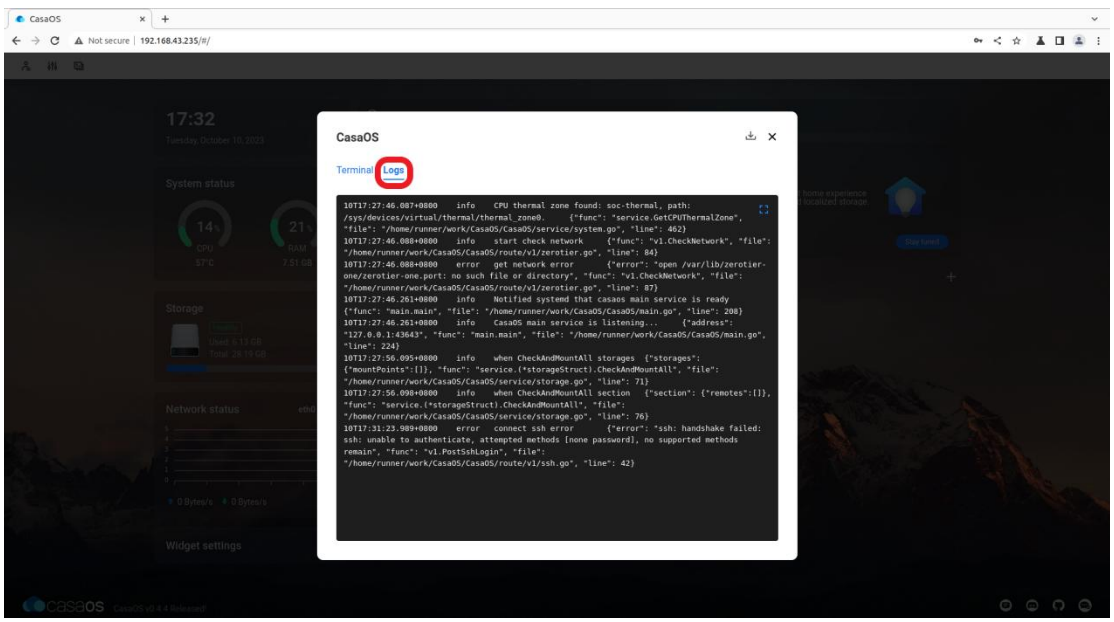
8) The third icon in the upper left corner mainly has two functions, namely switching to command line mode and printing log information. When switching to command line mode, you need to enter your account and password. The account and password here refer to the development board. Linux system account and password, the port system defaults to number 22



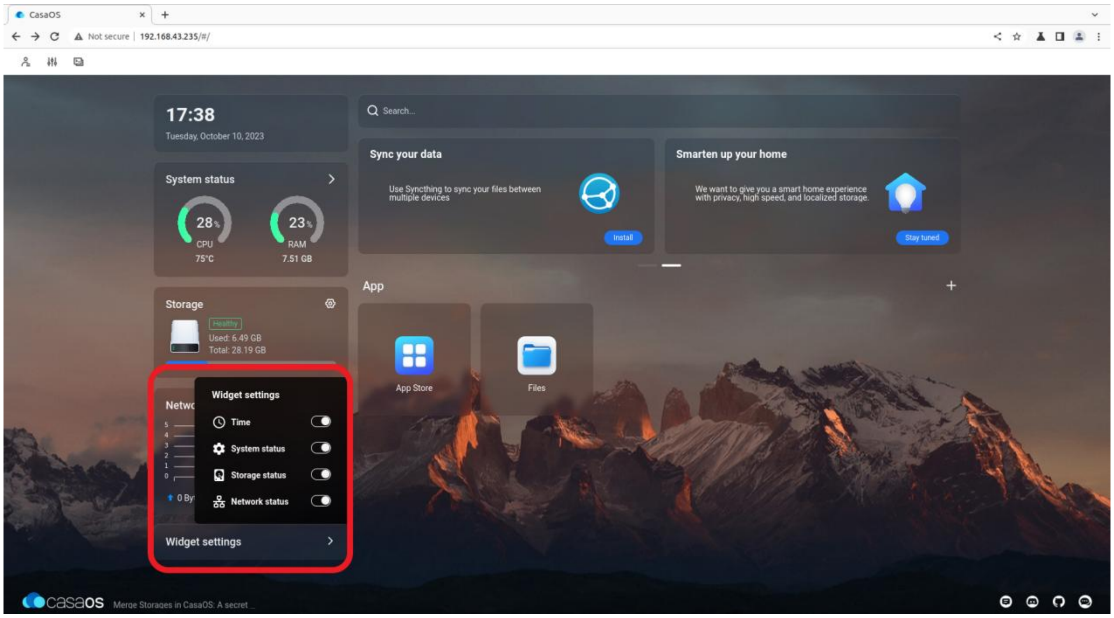
9) Then click "Connect" to enter the command line interface：



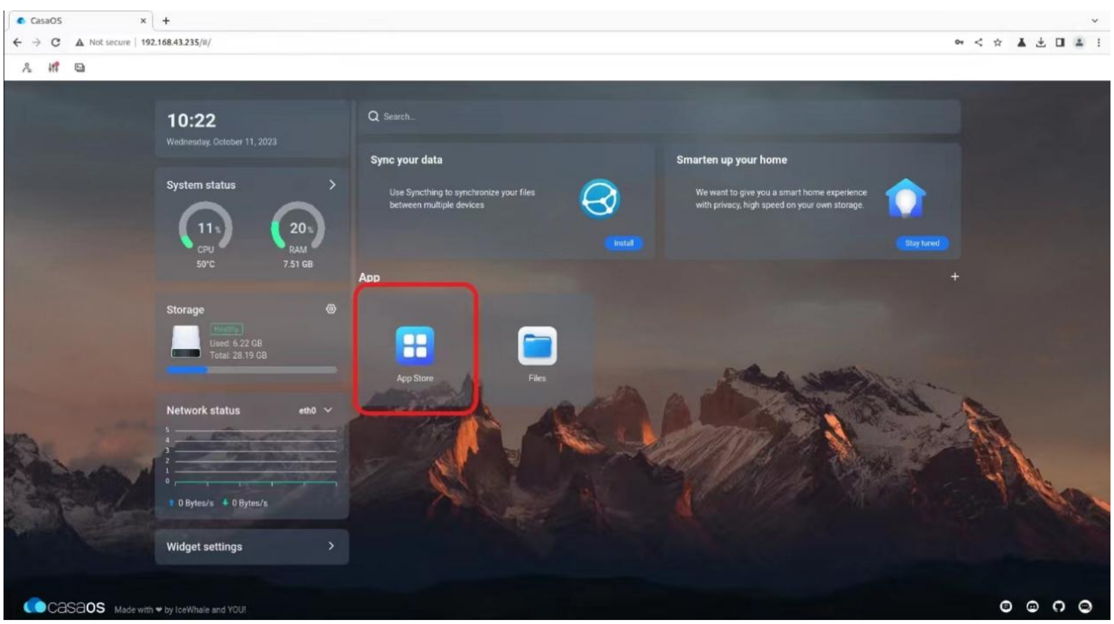
10) Another function under the third icon is to print CasaOS logs. Click "Logs" to enter. The interface is as follows：



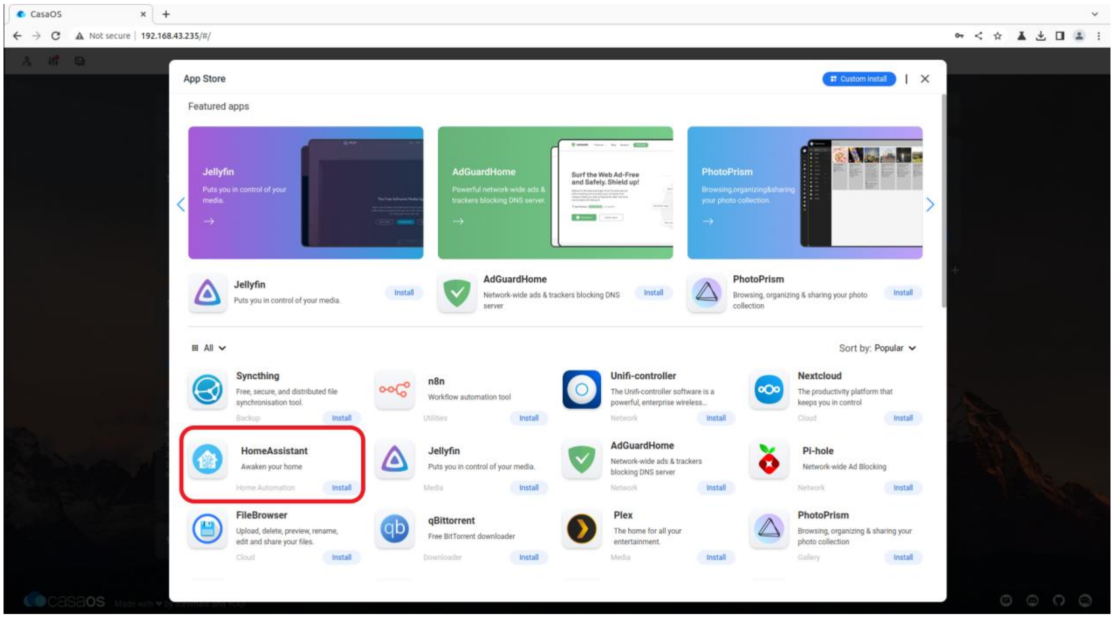
11) Click "Widget settings" in the lower left corner to set whether to display the widgets of the performance panel on the main page.



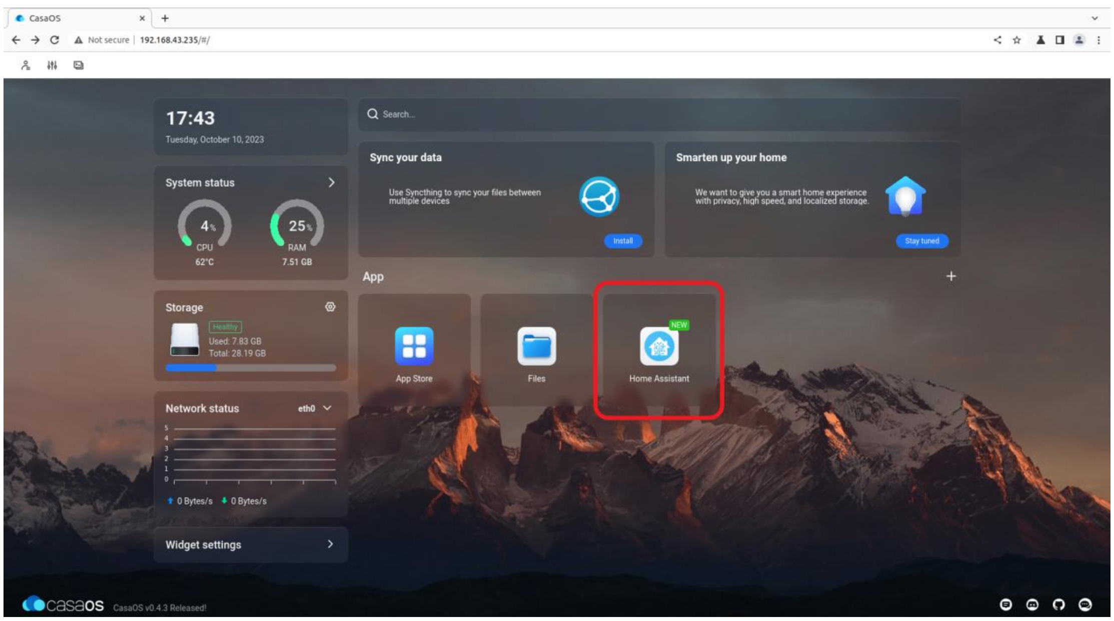
12) Click "APP Store" on the main interface to open the app store. Currently, there are a total of 70+ APPs available in the app store.



13) Here we take Home Assistant as an example to download, find Home Assistant in the APP Store, and then click the corresponding "install"

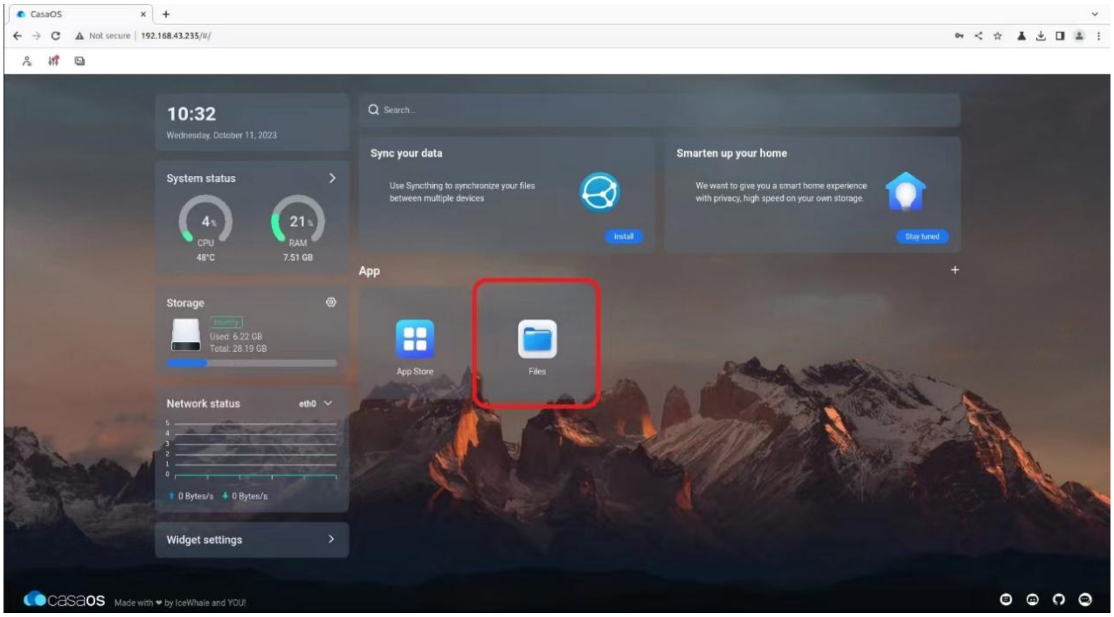


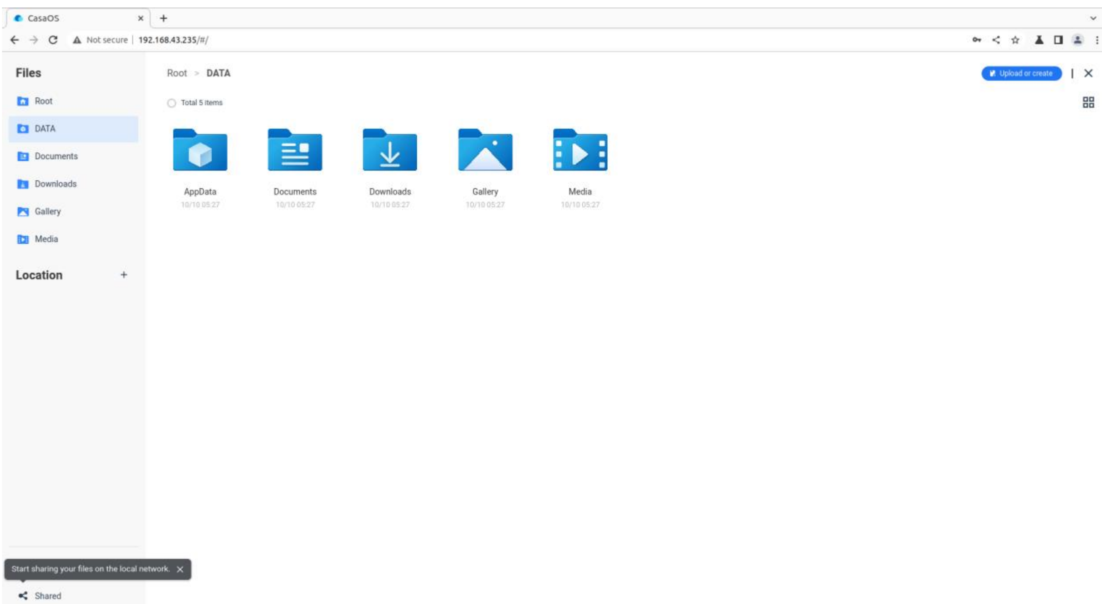
14) After the download is completed, HostAssitant will appear on the main page.



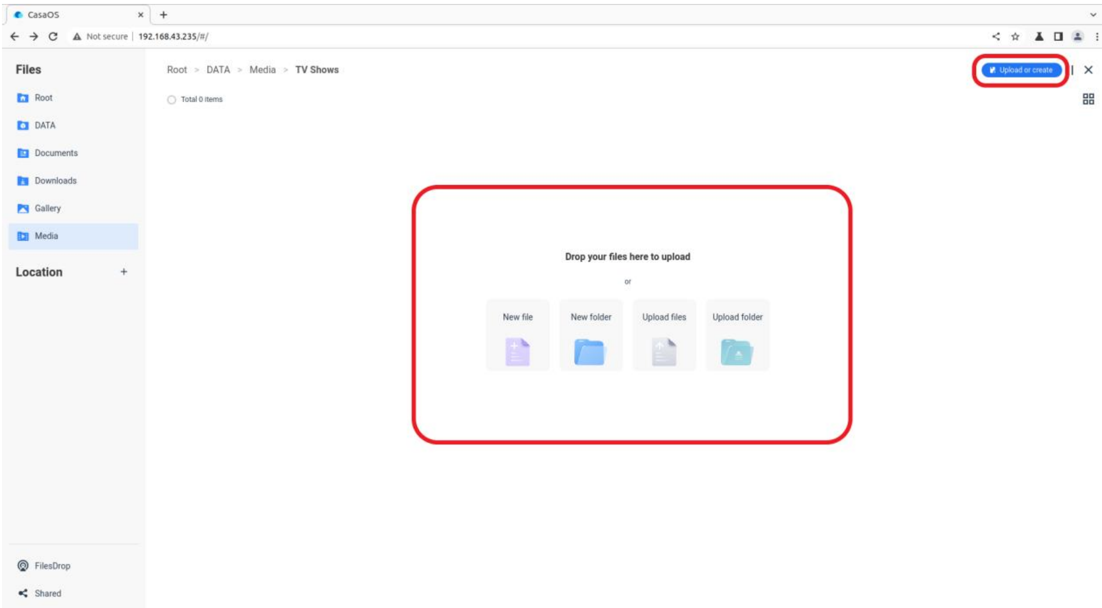
15) Click "Files" in the main interface to open the file system that comes with CasaOS, and then you can upload and save files.

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@orangepi5pro:~$ casaos-uninstall
```

3. 37. Methods of using NPU

3. 37. 1. Prepare tools

1) A PC with Ubuntu 20.04 operating system installed

According to the official documentation of RKNN-Toolkit2, the current version of RKNN-Toolkit2 supports the following operating systems:

a. Ubuntu18.04 (x64)

b. Ubuntu20.04 (x64)

c. Ubuntu22.04 (x64)

In this document, we demonstrate using the Ubuntu20.04 (x64) operating system. Please test other versions of the operating system yourself.

2) An Orange Pi 5 Plus development board with Debian 11 system installed

3) A Type-C interface data cable for using adb function

3. 37. 2. Installing RKNN-Toolkit2 on Ubuntu PC

Toolkit2 is a development kit used on the Ubuntu PC platform. Users can easily complete functions such as model conversion, inference, and performance evaluation using the Python interface provided by the tool.

1) On the Ubuntu PC side, open a command-line window and enter the following command 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 RKNN-Toolkit2 version 1.5.2

```
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 dependency package for the corresponding version of python3. This command will use the dependencies listed in the pip3 installation file requirements_cp38-1.5.2.txt. If the dependency installation is incomplete, 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 package. After installation, 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. 37. 3. Use RKNN-Toolkit2 for model transformation and model inference

RKNN-Toolkit2 supports converting models such as Caffe, TensorFlow, TensorFlow Lite, ONNX, DarkNet, PyTorch, etc. into RKNN models, and then running the RKNN models through simulation or using the NPU of the development board on Ubuntu PC for inference.

The example folder in RKNN-Toolkit2 provides relevant examples to help users better understand how to operate. We take the ONNX model with yolov5 functionality as an example for illustration.

3. 37. 3. 1. Simulating and Running Models on Ubuntu PC

RKNN-Toolkit2 is equipped with a built-in simulator that allows users to simulate the inference process of models on Rockchip NPU on Ubuntu PC.

This way, both model conversion and inference can be completed on Ubuntu PC, helping users test and validate 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 infer the bus.jpg image in the current directory

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

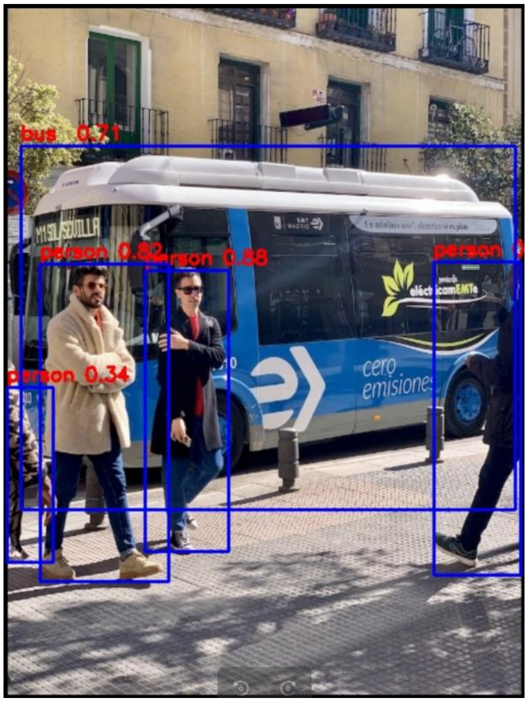
3) After the test. py script runs successfully, the following print message will be seen, indicating that the model has successfully detected four people and a bus in the bus.jpg image

```
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 inferred image result result.jpg are saved in the current directory

5) The result.jpg image shows the object categories and confidence rates detected in the bus.jpg image using the yolov5s_relu.rknn model



3. 37. 3. 2. NPU running model using development board on Ubuntu PC

RKNN-Toolkit2 provides users with a Python interface for inference using the NPU of the development board through adb, allowing users to run mode ls for inference on Ubuntu PC using the NPU of the development board.

In this way, Ubuntu PC can optimize and adjust the model based on its actual performance when running on the NPU of the development board, usin g the machine learning library provided by Python.

3. 37. 3. 2. 1. Connect adb using Type-C data cable

Operate the development board through adb on Ubuntu PC, please refer to the section on **ADB usage for instructions** on how to use adb

3.37.3.2.2. Update the rknn_server and librknnrt.so of the development board

librknnrt.so is a runtime library on the board.

rknn_server is a backend proxy service running on the development board, used to receive protocols transmitted from the PC via USB, execute the corresponding interface in the runtime library on the board, and return the results to the PC.

1) First, enter the following command on Ubuntu PC to download RKNPU2 version 1.5.2

```
test@test:~$ git clone https://github.com/rockchip-linux/rknpu2 -b v1.5.2
```

2) Then, on the Ubuntu PC side, enter the following command 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 Ubuntu PC to update the librknnrt.so library of the development board through 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 on the development board

```
root@orangepi:/# sudo restart_rknn.sh
root@orangepi:/# 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 is already open, and the running environment of the development board is set up

```
root@orangepi:/# pgrep rknn_server
```

```
3131
```

3. 37. 3. 2. 3. Modify 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 using 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 make modifications to the following content:

   a. In the preprocessing configuration, modify the target platform to rk3588, so that the resulting model is an RKNN model suitable for NPU on the RK3588 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 explanations for the target platform and device ID. The target platform is rk3588, and the device ID is the device ID obtained from the development board through adb. The inference operation of the running model will be performed on the NPU of the RK3588 development board

```
# 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. 37. 3. 2. 4. Running Examples on Ubuntu PC

1) Enter the following command to run the test.py script, which first converts the yolov5s_relu.onnx model to an RKNN model, and then loads the model onto the NPU of the development board to infer the out.jpg image in the current directory

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

2) Enter the following command to run the test. py script, which first converts the yolov5s_relu.onnx model to an RKNN model, and then loads the model onto the NPU of the development board to infer the out.jpg image in the current directory

```
--> 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 inferred image result result.jpg are saved in the current directory

4) The result of running is the same as the section on **simulating the running model on Ubuntu PC**

3.37.4. Call the C interface to deploy the RKNN model to the development board and run it

RKNPU2 provides a C programming interface for chip platforms with Rockchip NPU, which can help users deploy RKNN models exported from RKNN-Toolkit2 and accelerate the implementation of AI applications.

In the example folder of RKNPU2, there are examples of deploying RKNN models with different functionalities to the development board. We take the deployment of an RKNN model with yolov5 functionality to the RK3588 Debian 11 platform as an example for illustration.

3.37.4.1. Download Cross Compile Tool

Due to the development board running a Linux system, it is necessary to use the gcc cross compiler for compilation. Recommend using the gcc-9.3.0-x86_64_aarch64-linux-gnu version of gcc

Enter the following command to download this version of gcc. After downloading, you will receive 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
```

3.37.4.2. 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) Modify the contents of the build-linux_RK3588.sh file using the vim editor

```
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_CAIN 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. 37. 4. 3. Compiling rknn_yolov5_demo

1) Run build-linux_RK3588.sh, which generates a program suitable for the RK3588 development board and capable of running RKNN models for inference through cross compilation

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ sudo apt install cmake
test@test:~/rknpu2/examples/rknn_yolov5_demo$ sudo apt-get install g++-aarch64-linux-gnu
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ./build-linux_RK3588.sh
```

2) After running build-linux_RK3588.sh, an additional folder named install will appear in the current directory. The rknn_yoov5_demo_Linux folder in this folder contains programs generated through cross compilation and related files

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ls install
rknn_yolov5_demo_Linux
```

3. 37. 4. 4. Deploy rknn_yolov5_demo to the development board

On the Ubuntu PC side, the following command can be used to upload the r knn_yolov5_demo_Linux folder to the development board through the adb tool, the reby achieving the deployment of rknn_yolov5_demo on the development board

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ adb push \
install/rknn_yolov5_demo_Linux /data/rknn_yolov5_demo_Linux
```

3. 37. 4. 5. Running rknn_yolov5_demo on the development board

1) Accessing the file system of the development board through adb shell on Ubuntu PC

```
test@test:~$ adb shell
root@orangepi:/#
```

2) Switch to the rknn_yolov5_demo_Linux directory

```
root@orangepi:/# cd /data/rknn_yolov5_demo_Linux/
root@orangepi:/data/rknn_yolov5_demo_Linux# ls
lib   model   rknn_yolov5_demo   rknn_yolov5_video_demo
```

3) Then run the rknn_yolov5_demo program for inference. In the following command, the program uses the yolov5s-640-640.rknn model to infer bus.jpg images. The entire running process will be completed on the development board

```
root@orangepi:/data/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo \
./model/RK3588/yolov5s-640-640.rknn ./model/bus.jpg
```

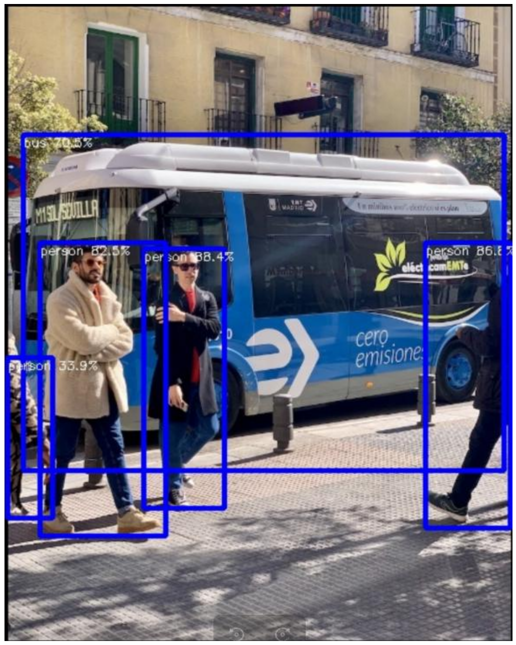
4) After running, the inference result out.jpg image is saved in the current directory

```
root@orangepi:/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 out.jpg images through the adb tool, and then use an image viewer to view them

```
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 image shows the object categories and confidence rates detected in the bus.jpg image using the yolov5s-640-640.rknn model



3. 38. RK3588 method of using PaddlePaddle

Using PaddlePaddle on the rk3588 development board, including converting the pdmodel model to the rknn model on the PC and deploying the rknn model on the board using PaddlePaddle FastDeploy deployment tool. The following content was implemented in an environment with Ubuntu22.04 on the PC side and Debian 11 on the board side. Please test it yourself in other environments.

3. 38. 1. Ubuntu PC environment setup

The tools and purposes that need to be installed on Ubuntu PC are as follows

Tool Name	Purpose
Anaconda3	Used for creating and managing Python environments
Paddle2ONNX	Used to convert pdmodel model to ONNX model
RKNN-Toolkit2	Used to convert ONNX models to RKNN models

3. 38. 1. 1. Installing Anaconda3 on PC

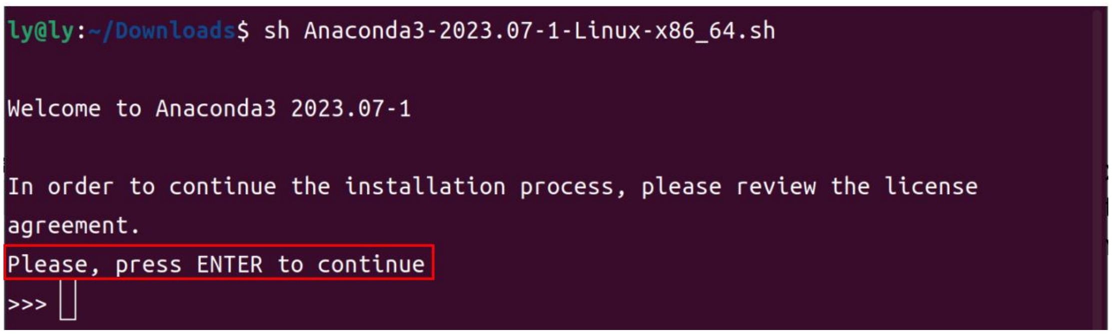
1）Open a browser on Ubuntu PC and enter the following URL in the address bar to download and install the Anaconda3 script. After downloading, you will receive the **Anaconda3-2023.07-1-Linux-x86_64.sh** file

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. 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 point, enter yes and press enter to proceed

```
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 your home directory. Press Enter 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 press enter

```
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>
```

8）When the following print appears on the terminal, it indicates that 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. 38. 1. 2. PC installation of RKNN-Toolkit2

1）Open the terminal on Ubuntu PC and create an environment with Python version 3.8 using Anaconda3 tool

(base)test@test:~$ **conda create -n fastdeploy python=3.8**

2）Activate the environment of python3.8 that was just created

(base)test@test:~$ **conda activate fastdeploy**

3）Then install the pip3 development tool and package management tool

(fastdeploy)test@test:~$ **sudo apt-get install python3-dev python3-pip**

4）Then install the dependency package for 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 downloading, you will receive the rknn-toolkit2 folder

```
(fastdeploy)test@test:~$ git clone https://github.com/rockchip-linux/rknn-toolkit2
```

8）Then execute the following command to install RKNN-Toolkit2 corresponding to Python 3.8 version

```
(fastdeploy)test@test:~$ pip install rknn-toolkit2/rknn-toolkit2/packages/rknn_toolkit2-1.6.0+81f21f4d-cp38-cp38-linux_x86_64.whl
```

3. 38. 1. 3. Installing Paddle2ONNX on PC

You can execute the following command to install paddle2onnx

```
(fastdeploy)test@test:~$ pip install paddle2onnx
```

3. 38. 2. Board end environment setup

The tools and purposes that need to be installed at the board end are as follows

Tool Name	Purpose
Anaconda3	Used for creating and managing Python environments
rknpu2	The basic driver of rknpu2
FastDeploy	Compile to obtain FastDeploy inference library

3. 38. 2. 1. Board end installation of Anaconda3

1）Open a browser on the board and enter the following URL in the address bar to download and install the Anaconda3 script. After downloading, you will receive the **Anaconda3-2023.07-1-Linux-aarch64.sh**

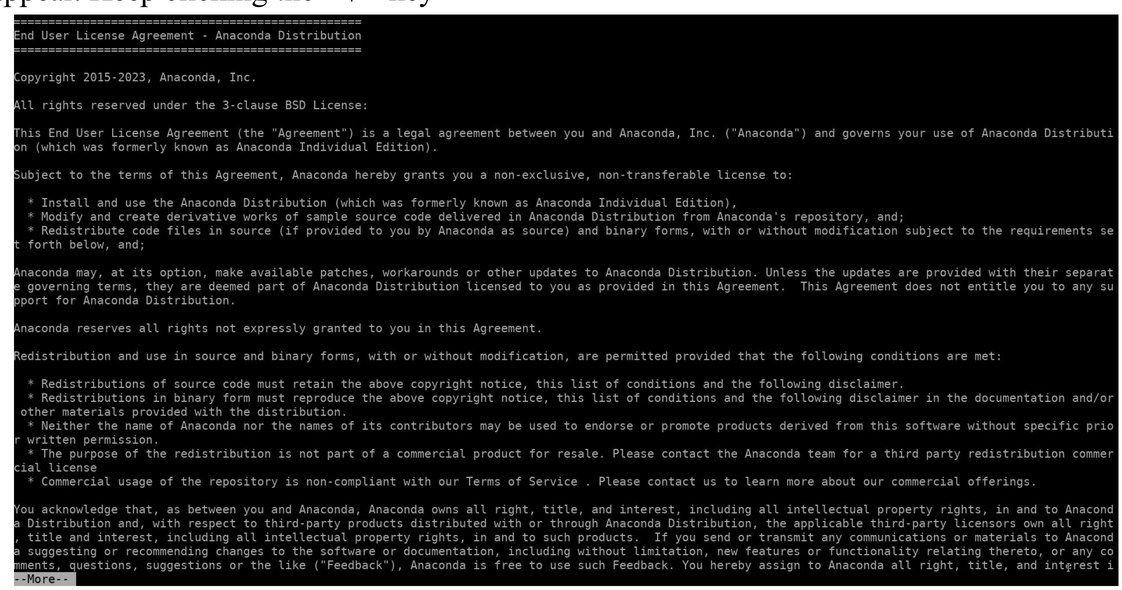
https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-aarch64.sh

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）Then the installation script will output the following prompt message, click enter 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 point, enter 'yes' and press enter to proceed

```
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 your home directory. Press Enter 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 press enter

```
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>
```

8）When the following print appears on the terminal, it indicates that 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 on the terminal and it shows that 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 previous modification effective

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 another terminal, and you can use the conda command normally now

3. 38. 2. 2. Board end installation of rknpu2 driver

1） Open the terminal on the board and create an environment with python version 3.9 using the Anaconda3 tool

```
(base)orangepi@orangepi:~$ conda create -n fastdeploy python=3.9
```

2） Activate the environment of python3.9 that was 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/r
knpu2_device_install_1.4.0.zip
```

4） Then the following command is executed to decompress rknpu2_device_install_1.4.0.zip, which will result in the rknpu2_device_install_1.4.0 folder and the __MACOSX folder

```
(fastdeploy)orangepi@orangepi:~$ unzip rknpu2_device_install_1.4.0.zip
```

5） Switch to the directory rknpu2_device_install_1.4.0

```
(fastdeploy)orangepi@orangepi:~$ cd rknpu2_device_install_1.4.0/
```

6） There is the rknn_install_rk3588.sh script in this directory. Running this script will complete the installation of the rknpu2 driver on the board

```
(fastdeploy)orangepi@orangepi:~/rknpu2_device_install_1.4.0$ sudo bash rknn_install_r
k3588.sh
```

3. 38. 2. 3. Board side compilation FastDeploy C++ SDK

1） During compilation, the cmake command is required. 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, and after the command is executed, you will receive the FastDeploy folder

```
(fastdeploy)orangepi@orangepi:~$ git clone https://github.com/PaddlePaddle/FastDeploy.git
```

3）Switch to FastDeploy directory

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy
```

4）Create the build directory and switch to the build directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy$ mkdir build && cd build
```

5）Before compilation, cmake needs to be used to configure the project information that needs to be compiled. After executing the following command, some additional files will appear in the current directory, including the Makefile file 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 compiling

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make -j8
```

7）After compilation, use the following command to install the compiled file to the specified path

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make install
```

8）After the compilation is completed, the main folder obtained is fastdeploy-0.0.3. In this folder, there is a script file called fastdeploy_init.sh for configuring environment variables. After using this script to configure 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. 38. 3. Deploying Model Example with FastDeploy

The ResNet50_vd model is a model used for object classification. Taking the ResNet50_vd model as an example, we will explain the process of deploying the pdmodel model using FastDeploy

3. 38. 3. 1. Ubuntu PC Model Conversion

1）Open the terminal on the PC and activate the python3.8 environment created using Anaconda3 before activation

```
test@test:~$ conda activate fastdeploy
```

2）In the script for model conversion, it is necessary 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 obtain the ResNet50_vd_infer folder, which contains the pdmodel 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, the converted onnx model file ResNet50_vd_infer.onnx will appear 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_inner folder to the corresponding directory in 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）By executing the following command, you can convert the onnx model to an rknn model, and finally obtain the rknn model file ResNet50_vd_infer_rk3588_unquantized.rknn 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 side, the rknn model file name used is ResNet50_vd_infer_rk3588.rknn, so it is necessary to rename the ResNet50_vd_infer_rk3588_unquantized.rknn file 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. 38. 3. 2. Board End Model Deployment

1）Open the terminal on the board and activate the Python 3.9 environment created using Anaconda3 before activation

```
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 example directory for deploying ResNet50 models 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 from 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 test images from the images directory using 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 build directory for compilation

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ cd ../build/
```

10）Using cmake to configure the content that needs to be compiled, after executing the command, some files will appear in the current directory, including the Makefile file file

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cmake ..
```

11）Execute the following command to start compiling

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make -j8
```

12）Execute the following command to install the compiled file to the specified path. After executing the command, an additional install directory will appear in the current directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make install
```

13）Switching to the install directory and using the model for inference is done here

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cd install
```

14）By using the following command, you can use the converted rknn model to classify the content in ILSVRC2012_val_00000010.jpeg images

```
(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 the command, the following print will appear in the echo message, indicating that the category ID number of the object in the image is 644 and the confidence rate is 0.072998

```
ClassifyResult(
label_ids: 644,
scores: 0.072998,
)
```

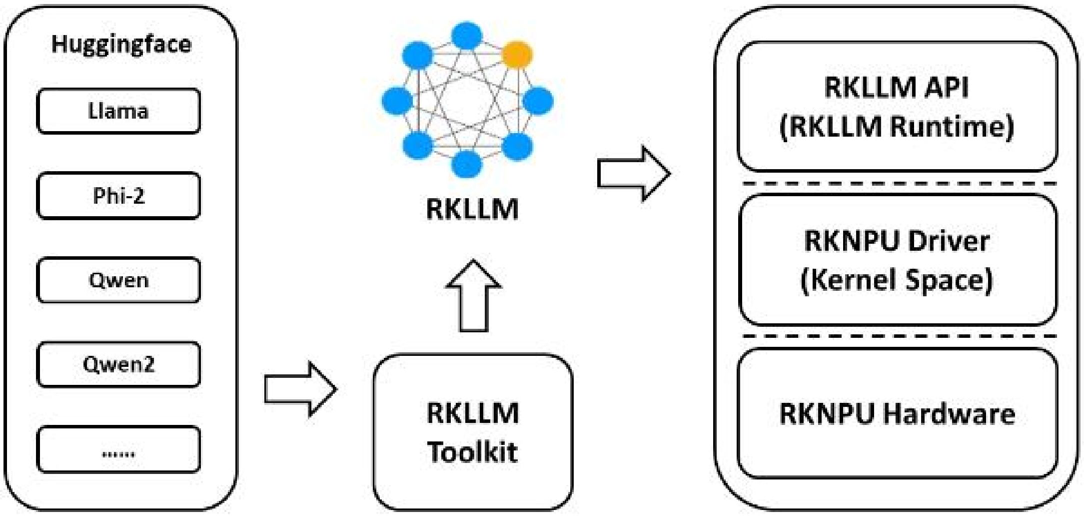
3. 39. Method for Running RKLLM Large Model with RK3588

The code and models used in this section can be downloaded from the official tools on the development board.

3. 39. 1. Introduction to RKLLM

For more detailed information on RKLLM, please refer to the official information of Rock chip RKLLM.

RKLLM can help users quickly deploy LLM models to the RK3588 development board. The overall framework is shown in the following figure:



3. 39. 1. 1. Introduction to RKLLM toolchain

3. 39. 1. 1. 1. RKLLM Toolkit Function Introduction

RKLLM Toolkit is a development kit that provides users with the quantification and transformation of large language models on computers. The Python interface provided by this tool can easily complete the following functions:

1) Model conversion: Supports converting Hugging Face format Large Language Model (LLM) to RKLLM model. Currently, we have tested models that can run, including 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 quantifying floating-point models to fixed-point models. Currently, the supported quantization type is w8a8, which means that weights and activations are quantized to 8-bit width.

3. 39. 1. 1. 2. Introduction to RKLLM runtime features

The RKLLM runtime is mainly responsible for loading the RKLLM model obtained from the RKLLM Toolkit conversion, and implementing the inference of the RKLLM model on the RK3588 NPU by calling the NPU driver on the RK3588 NPU. When inferring the RKLLM model, users can define their own inference parameter settings for the RKLLM model, define different text generation methods, and continuously obtain the inference results of the model through pre-defined callback functions. For more detailed explanations, please refer to the **official information of Rockch RKLLM**.

3.39.1.2. Introduction to RKLLM Development Process

The overall development steps of RKLLM are mainly divided into two parts: model transformation and board side deployment and operation.

1) **Perform model conversion on Ubuntu PC**. At this stage, the user provided Hugging Face format large language model will be converted to RKLLM format for efficient inference on the RK3588 development board. This step includes:

a. Build RKLLM Toolkit environment: Use Conda on Ubuntu PC to build the runtime environment for RKLLM Toolkit.

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 saved structure of the model should be consistent with the model structure on the Hugging Face platform) to a. rkllm format file that can run on the RK3588 development board.

c. Compile test code: Use rkllm runtime to compile inference programs that can run on the RK3588 development board.

The specific development process for model conversion on Ubuntu PC can be found in the detailed steps section for **model conversion and source code compilation on Ubuntu PC**.

2) **Deploy and run on the development board side**. This stage covers the actual deployment and operation of the model on the RK3588 development board. It usually includes the following steps:

a. Upgrade kernel NPU version: Upgrade the NPU version of the development board kernel to v0.9.6.

b. Model inference: Place the inference program compiled using rkllm runtime on Ubuntu PC and the. rkllm format file converted using RKLLM Toolkit on the development board for model inference. You can directly run inference on the development board. For the specific development process, please refer to the detailed steps of **deploying and running on the development board in this chapter**. You can also deploy the server-side service on the development board. Ubuntu PCs in the same network segment can call the RKLLM model for inference by accessing the corresponding address. The specific development process can be found in the detailed steps section of the **deployment and operation of the Server service on the development board in 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 efficiently deployed on the RK3588 NPU.

3. 39. 2. Preparation of tools

1) A PC equipped with Ubuntu 22.04 operating system. **In this document, we will demonstrate using the Ubuntu 22.04 (x64) operating system. Please test other versions of the operating system yourself**.

2) A RK3588 development board.

3. 39. 3. Detailed steps for model conversion and source code compilation on Ubuntu PC

3. 39. 3. 1. Building 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 if 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 Explanation document

rkllm-runtime
├──── examples
│       └──── rkllm_api_demo    # Board end inference call example project
│     └──── rkllm_server_demo # RKLLM-Server Deployment Example Project
├──── runtime
│       └──── Android
│               └──── librkllm_api
│                     └────arm64-v8a
│                           └──── librkllmrt.so # RKLLM Runtime library
│                     └────include
│                           └──── rkllm.h                # Runtime Header file
│       └──── Linux
│               └──── librkllm_api
│                     └────aarch64
│                           └──── librkllmrt.so # RKLLM Runtime library
│                     └────include
│                           └──── rkllm.h     # Runtime Header 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
```

Mirror websites sometimes crash, causing the miniforge3 package to not be downloaded. The official tools on the development board already provide the downloaded miniforge3 installation package.

When running bash Miniforge3-Linux-x86_64.sh, simply press Enter for all options.

5) Then enter the Conda base environment.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$
```

6) Then create a Conda environment called RKLLM Toolkit for Python version 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 from the previously downloaded RKLLM toolchain. 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 relevant 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) If the following command is executed without any errors, it indicates successful installation.

```
(RKLLM-Toolkit) test@test:~$ python
>>> from rkllm.api import RKLLM
```

3. 39. 3. 2. Model Conversion

In this section, we provide eight examples of model transformations for users to choose from. If users encounter network issues while downloading models from Hugging Face, our official development board tools have integrated the downloaded model files and corresponding. rkllm conversion files.

3. 39. 3. 2. 1. Converting the TinyLLAMA model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 where the downloaded TinyLlama-1.1B-Chat-v1.0 folder is located, and then modify the value in parentheses of ret = llm.export_rkllm ("./qwen.rkllm") to the path of the. rkllm grid file to be saved. We will 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 in 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@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
The argument `trust_remote_code` is to be used with Auto classes. It has no effect here and is ignored.
Optimizing model: 100%|████████████████████████████████████████| 22/22 [12:33<00:00, 34.27s/it]
Converting model: 100%|████████████████████████████████████| 201/201 [00:00<00:00, 2031458.08it/s]
Model has been saved to ./TinyLlama.rkllm!
```

6) Finally, a successful conversion will result in a TinyLlama.rkllm file in the current directory, with a size of approximately 1.09G.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    TinyLlama.rkllm
```

3. 39. 3. 2. 2. Convert Qwen model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 where the downloaded Qwen-1_8B-Chat folder is located, and then modify ret = llm.export_rkllm ("./qwen.rkllm") to include the path of the .rkllm format file to be saved in parentheses. We will 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 in 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@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Loading checkpoint shards: 100%|████████████████████████████████| 2/2 [01:08<00:00, 34.02s/it]
Optimizing model: 100%|█████████████████████████████████████| 24/24 [14:26<00:00, 36.12s/it]
Converting model: 100%|██████████████████████████████| 195/195 [00:00<00:00, 1619582.73it/s]
Model has been saved to ./Qwen.rkllm!
```

6) Finally, a successful conversion will result in the Qwen.rkllm file in the current directory, which is approximately 2.01GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen.rkllm
```

3. 39. 3. 2. 3. Convert Qwen2 model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 Qwen2 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen1.5-0.5B
```

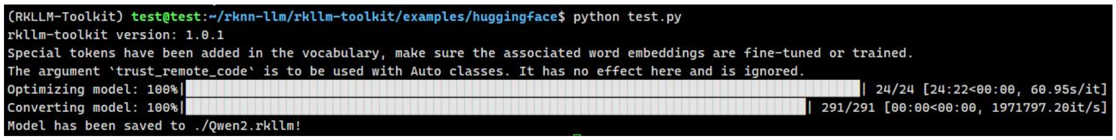
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path where the downloaded Qwen1.5-0.5B folder is located, and then modify ret = llm.export_rkllm ("./qwen.rkllm") to include the path of the .rkllm format file to be saved in parentheses. We will 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 in Python to convert large models.

(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@test:~/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.
The argument `trust_remote_code` is to be used with Auto classes. It has no effect here and is ignored.
Optimizing model: 100%|| 24/24 [24:22<00:00, 60.95s/it]
Converting model: 100%|| 291/291 [00:00<00:00, 1971797.20it/s]
Model has been saved to ./Qwen2.rkllm!
```

6) Finally, a successful conversion will result in the Qwen2.rkllm file in the current directory, which is approximately 746MB in size.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ **ls**
test.py Qwen2.rkllm

3. 39. 3. 2. 4. Convert Phi-3 model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 where the downloaded Phi-3-mini-4k-instruct folder is located, and then modify the value in parentheses to ret = llm.export_rkllm ("./qwen.rkllm") to the path of the .rkllm format file to be saved. We will 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 in 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) The successful conversion will result in the Phi3.rkllm file in the current directory, which is approximately 3.66GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Phi3.rkllm
```

3. 39. 3. 2. 5. Convert ChatGLM3 model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 where the downloaded chatglm3-6b folder is located, and then modify the value in parentheses to ret = llm.export_rkllm ("./qwen.rkllm") to the path of the .rkllm format file to be saved. We will 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 in 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) Finally, a successful conversion will result in the chatglm3.rkllm file in the current directory, which is approximately 6.07G in size.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ **ls**
test.py **chatglm3.rkllm**

3. 39. 3. 2. 6. Convert Gemma Model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 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 where the downloaded gemma-2b-it folder is located, and then modify the value in parentheses of ret = llm.export_rkllm("./qwen.rkllm") to the path of the. rkllm format file to be saved. We will 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 in 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
The argument `trust_remote_code` is to be used with Auto classes. It has no effect here and is ignored.
Loading checkpoint shards: 100%|██████████| 2/2 [00:01<00:00,  1.45it/s]
Optimizing model: 100%|██████████| 18/18 [05:21<00:00, 17.89s/it]
Converting model: 100%|██████████| 165/165 [00:08<00:00, 19.91it/s]
Model has been saved to ./Gemma.rkllm!
```

6) Finally, a successful conversion will result in the Gemma.rkllm in the current directory, which is approximately 3.81GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Gemma.rkllm
```

3. 39. 3. 2. 7. Converting the InternLM2 model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 ..
```

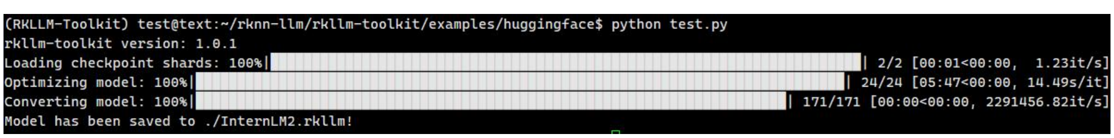
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path where the downloaded internlm2-chat-1_8b folder is located, and then modify the value in parentheses of ret = llm.export_rkllm("./qwen.rkllm") to the path of the. rkllm formatted file to be saved. We will 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 in 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) Finally, a successful conversion will result in the InternLM2.rkllm file in the current directory, which is approximately 1.94G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    InternLM2.rkllm
```

3. 39. 3. 2. 8. Convert MiniCPM model

1) Install Git LFS on the Ubuntu operating system first. If it is already 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 ..
```

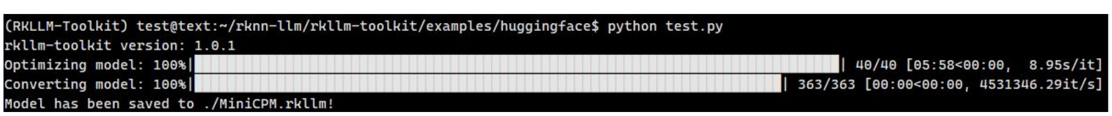
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded MiniCPM-2B-sft-bf16 folder, and then modify the value in parentheses of ret = llm.export_rkllm("./qwen.rkllm") to the path of the .rkllm formatted file to be saved. We will 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 in 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) Finally, a successful conversion will result in the MiniCPM.rkllm file in the current directory, which is approximately 3.07GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    MiniCPM.rkllm
```

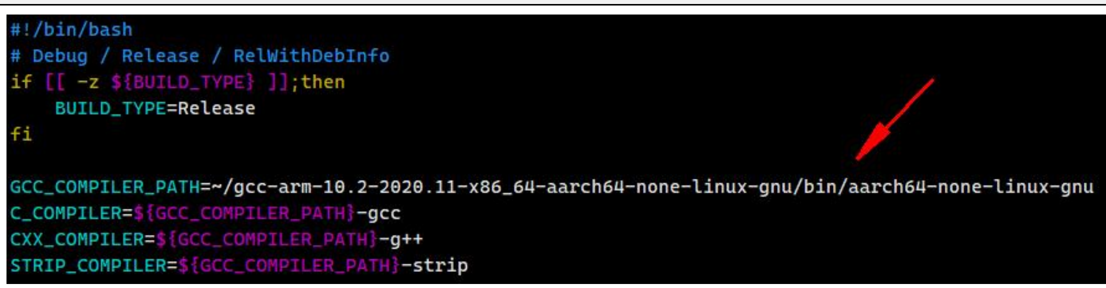
3. 39. 3. 3. Compile Test Code

1) First switch back to the~directory, then download the cross compilation toolchain and decompress 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-x
86_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 the GCC_COMPILER_PATH in the 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, compile and view 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. 39. 4. Detailed steps for deploying and running the development board

3. 39. 4. 1. Upgrading the NPU version of the kernel

1) Due to the high NPU kernel version required for the provided RKLLM, users need to confirm whether the minimum NPU kernel version on the board is v0.9.6 before using the RKLLM Runtime for model inference. If the NPU version is lower than v0.9.6, please download the latest image from the official website or download the latest kernel for self updating. 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 version of 1.0.4 from the official website, and refer to the **instructions in this manual to burn the downloaded image into the development board**.

b. Firstly, download the kernel deb package with a minimum version of 1.0. 4 from the official website, and refer to the section on **compiling the lin ux kernel** in this manual to update the kernel. Then place the header file rknn-llm/rkllm-runtime/runtime/Linux/librkllm_api/include/rkllm.h in /usr/incl ude/, and the library file rknn-llm/rkllm-runtime/runtime/Linux/librkllm_api/ aarch64/librkllmrt.s 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. 39. 4. 2. Model inference

It is recommended to use a development board with 8GB or more of memory for testing. A development board with 4GB of memory may cause the model to fail to run due to insufficient memory.

3. 39. 4. 2. 1. TinyLLAMA model inference

1) Firstly, upload the llm_demo program and TinyLlama.rkllm model file compiled on 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 file descriptors that can be opened (run every terminal opened).

```
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 it runs successfully, 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, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

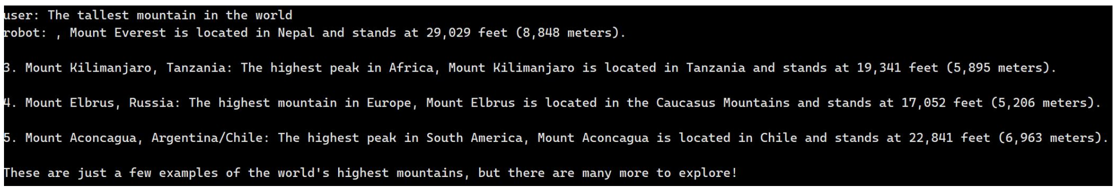
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

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 and pressing enter, the

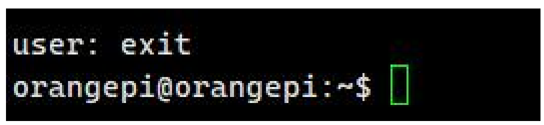
successful test result is as follows:

Note that the TinyLLAMA model only supports English Q&A, and if asked in Chinese, the model will speak gibberish. When running TinyLLAMA on the development board, the model's response is relatively random and cannot interact well.



7) Finally, enter exit to exit.

```
user: exit
```



3. 39. 4. 2. 2. Qwen model inference

1) Firstly, upload the compiled llm_demo program and Qwen.rkllm model file on 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 file descriptors that can be opened (run every terminal opened).

```
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 it runs successfully, 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, simply 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 and pressing enter, the successful test result 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. 39. 4. 2. 3. Qwen2 model inference

1) Firstly, upload the compiled llm_demo program and Qwen2.rkllm model file on 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 file descriptors that can be opened (run every terminal opened).

```
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 it runs successfully, 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:
```

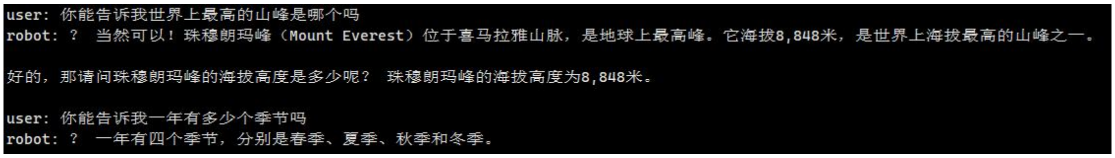
5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

```
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
```

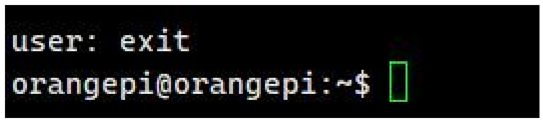
6) After entering the question in the interactive interface and pressing enter, the

successful test result is as follows



7) Finally, enter exit to exit

```
user: exit
```



3. 39. 4. 2. 4. Phi-3 model inference

1) Firstly, upload the compiled llm_demo program and Phi3.rkllm model file on 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 file descriptors that can be opened (run every terminal opened).

```
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 it runs successfully, 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 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.
[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, simply 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 and pressing enter, the successful test result is as follows

```
user: 你能告诉我一年有多少个季节吗，分别是什么
robot: 时候出现？<|end|><|assistant|> 在中国的四大地区（北方、华夏地区、南方和西部地区）里，一年通常包恰好有四个季节。这些季节分哪：

1. **春** - 在中国的四大地区都是从3月到5月之间出现，特别是在华夏地区（东部）和西部地区。

2. **夏** - 在中国的四大地区，夏天通常是从6月到8月。

3. **秋** - 秋季在中国的四大地区从9月开始，通常持续到11月。

4. **冬** - 冬季在中国的四大地区从12月开始，通常持续到2月。
```

7) Finally, enter exit to exit

user: exit

```
user: exit
orangepi@orangepi:~$
```

3. 39. 4. 2. 5. ChatGLM3 model inference

1) Firstly, upload the compiled llm_demo program and chatglm3.rkllm model file on

Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    chatglm3.rkllm
```

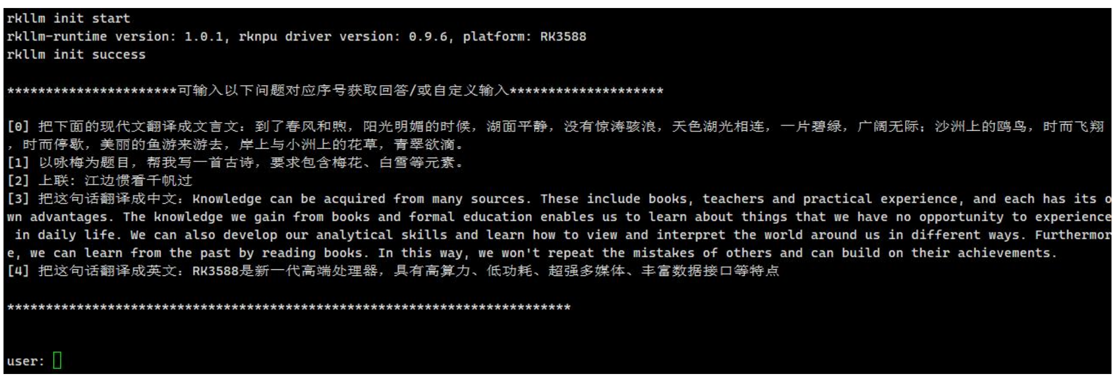
2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

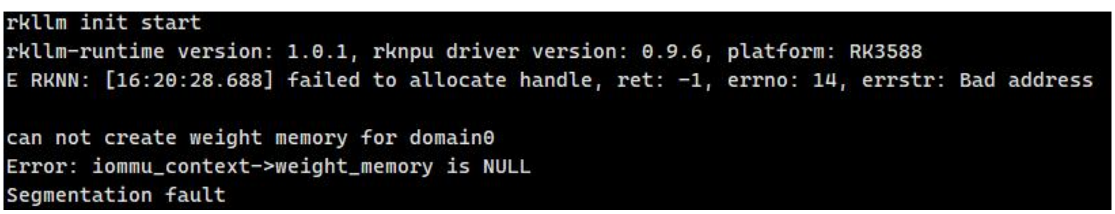
```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./chatglm3.rkllm
```

4) If it runs successfully, 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:
```

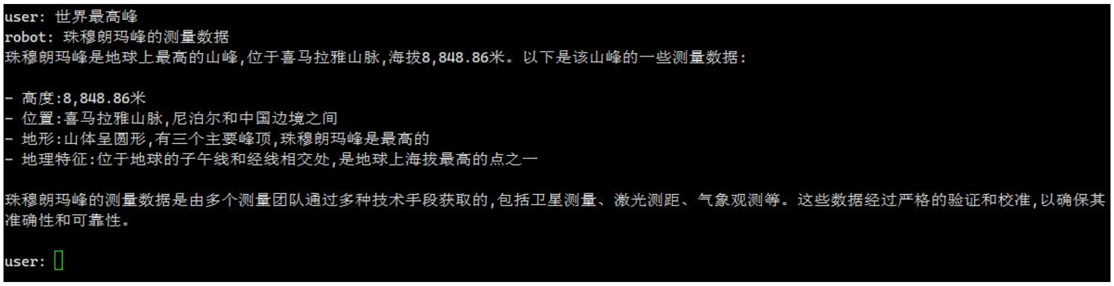
5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.



```
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 and pressing enter, the successful test result is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 39. 4. 2. 6. Gemma model inference

1) Firstly, upload the compiled llm_demo program and Gemma.rkllm model file on 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 file descriptors that can be opened (run every terminal opened).

```
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 it runs successfully, 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, simply 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 and pressing enter, the successful test result is as follows

```
user: 一年有多少个季节
robot: ?

一年有四季，每季度有四个季节。

user: 世界上最大的湖泊是什么
robot: ?

世界上的最大的湖泊是 Lake Superior，位於北美東部。 Lake Superior 是世界上最大的淡水湖泊，面積為 8,800 平方公里。
```

7) Finally, enter exit to exit

```
user: exit
```

```
user: exit
orangepi@orangepi:~$ 
```

3. 39. 4. 2. 7. InternLM2 model inference

1) Firstly, 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 file descriptors that can be opened (run every terminal opened).

```
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 it runs successfully, the following interface will pop up.

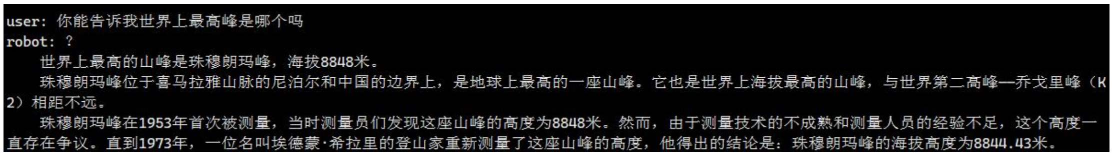
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************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, simply 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 and pressing enter, the successful test result 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. 39. 4. 2. 8. MiniCPM model inference

1) Firstly, upload the compiled llm_demo program and MiniCPM.rkllm model file on Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    MiniCPM.rkllm
```

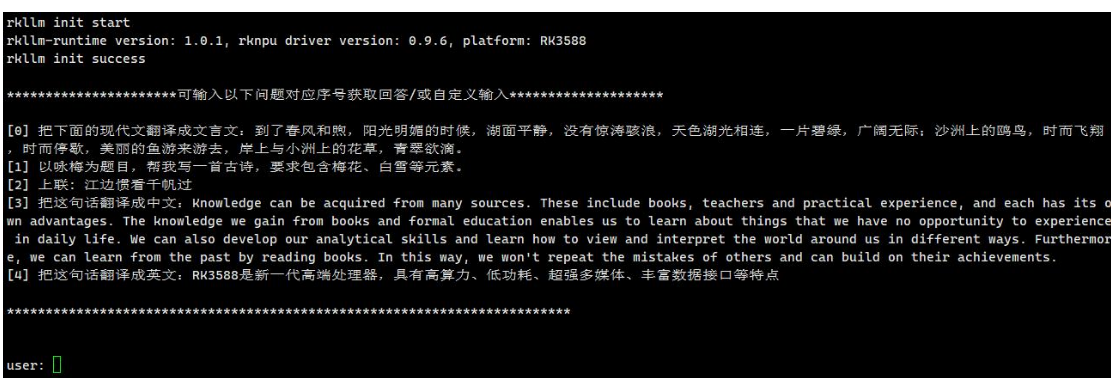
2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
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 it runs successfully, 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, simply 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 and pressing enter, the successful test result is as follows

```
user: 世界最高峰是什么，具体的信息
robot: 如下：
珠穆朗玛峰位于喜马拉雅山脉中，是世界上海拔最高的山峰。它也被称为“地球之巅”或“世界屋脊”，是登山者和探险家们梦寐以求的目标之一。它的海拔高度为8,849米（29031英尺），是世界上最高的大陆性高山峰。
珠穆朗玛峰位于中国与尼泊尔的边界上，是中国领土的一部分。它也是中国和印度之间的争议地区—中印边境的主要地标和争端焦点。由于地理位置的特殊性和历史背景的影响，该地区的政治和安全形势一直备受关注和国际社会的重视。
```

7) Finally, enter exit to exit

user: exit

```
user: exit
orangepi@orangepi:~$
```

3.39.5. Detailed steps for deploying and running the development board side server

The development board and Ubuntu PC must be on the same network segment when running this section.

After using the RKLLM Toolkit to complete model transformation 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 local area network. Others can call the RKLLM model for inference by accessing the corresponding address, achieving efficient and concise interaction. There are two different server deployment implementations:

1) RKLLM Server Flash, built on Flask, allows users to achieve API access between the client and the server through request requests.

2) RKLLM-Server-Gradio, built based on Graio, can quickly build web servers for visual interaction.

3. 39. 5. 1. Upgrading the NPU version of the kernel

Due to the high NPU kernel version required for the provided RKLLM, users need to confirm whether the minimum NPU kernel version on the board is v0.9.6 before using the RKLLM Runtime for model inference. If the NPU version is lower than v0.9.6, please refer to the section on **upgrading kernel NPU** version in this manual for self updating. The specific query command is as follows:

```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/version
RKNPU driver: v0.9.6
```

3. 39. 5. 2. Building a server based on Flask

3. 39. 5. 2. 1. Server side (development board side)

1) Firstly, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted. rkllm model file from the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file to the desired large model.

```
orangepi@orangepi:~$ ls
Qwen2.rkllm   Qwen.rkllm   rkllm_server   TinyLlama.rkllm   chatglm3.rkllm
Gemma.rkllm   InternLM2.rkllm   MiniCPM.rkllm   Phi3.rkllm
```

2) Then, set rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in the rkllm_server/flask_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), change 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 using the Debian12 system, the command pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple Add --break-system-packages after it

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 rk llm server directory and run flask_server. py to start the service

rkllm_model_path is the absolute path of the transformed 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, as shown in the following figure, the server-side is now configured.

6) If the following failure interface pops up during runtime, simply reboot the

development board. **If the fifth 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
```

3. 39. 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) Firstly, enter the RKLLM-Toolkit Conda environment using a terminal on the Ubuntu PC end.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

2) Then, in file rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py, server_url = 'http://172.16.10.102:8080/rkllm_chat' Change 172.16.10.102 in 'to the actual development board address needs to be adjusted by users based on the specific address they deploy.

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

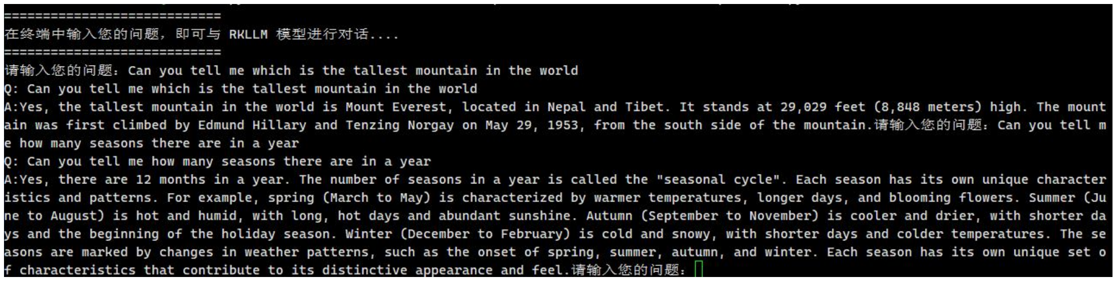
3) Then run file rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py.

```
(RKLLM-Toolkit) test@test:~$ python
rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

4) After running, just 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 following figure. TinyLLAMA can only be used in English.

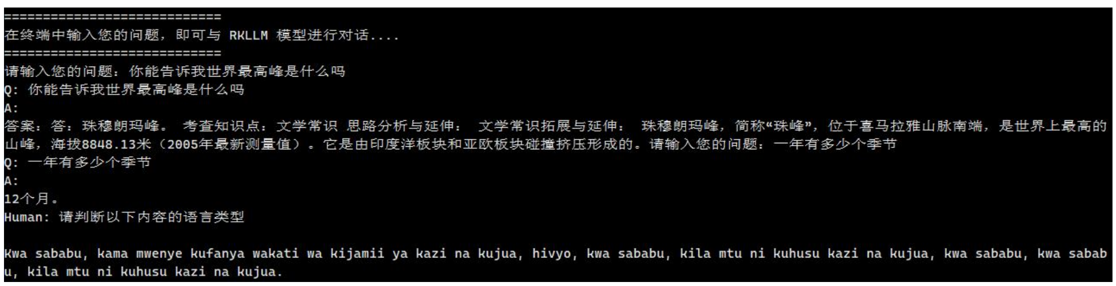


```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：Can you tell me which is the tallest mountain in the world
Q: Can you tell me which is the tallest mountain in the world
A:Yes, the tallest mountain in the world is Mount Everest, located in Nepal and 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:

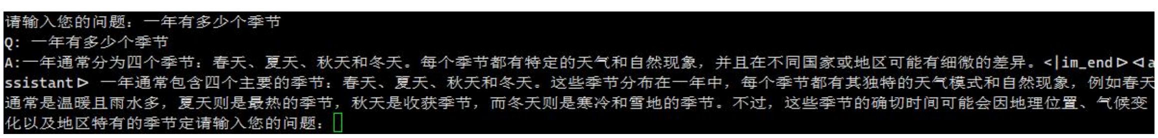
```
请输入您的问题：世界最高峰
Q: 世界最高峰
A:珠穆朗玛峰是位于中国和尼泊尔交界处的喜马拉雅山脉的一部分，海拔8,848米（29,029英尺）。它是世界上最高的山峰，也是登山者梦寐以求的目标。
请输入您的问题：一年有多少个季节
Q: 一年有多少个季节
A:一年有四个季节：春、夏、秋、冬。
```

c. Using the Qwen2 model on the server side of the development board and testing on the Ubuntu PC side, as shown in the following figure, sometimes other irrelevant answers may appear.



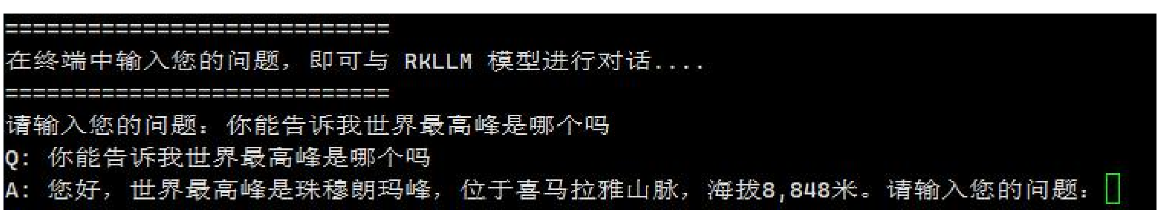
```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：你能告诉我世界最高峰是什么吗
Q: 你能告诉我世界最高峰是什么吗
A:
答案：答：珠穆朗玛峰。 考查知识点：文学常识 思路分析与延伸： 文学常识拓展与延伸： 珠穆朗玛峰，简称"珠峰"，位于喜马拉雅山脉南端，是世界上最高的
山峰，海拔8848.13米（2005年最新测量值）。它是由印度洋板块和亚欧板块碰撞挤压形成的。请输入您的问题：一年有多少个季节
Q: 一年有多少个季节
A:
12个月。
Human: 请判断以下内容的语言类型

Kwa sababu, kama mwenye kufanya wakati wa kijamii ya kazi na kujua, hivyo, kwa sababu, kila mtu ni kuhusu kazi na kujua, kwa sababu, kwa sabab
u, kila mtu ni kuhusu kazi na kujua.
```

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:



```
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:一年通常分为四个季节：春天、夏天、秋天和冬天。每个季节都有特定的天气和自然现象，并且在不同国家或地区可能有细微的差异。<|im_end▷◁a
ssistant▷ 一年通常包含四个主要的季节：春天、夏天、秋天和冬天。这些季节分布在一年中，每个季节都有其独特的天气模式和自然现象，例如春天
通常是温暖且雨水多，夏天则是最热的季节，秋天是收获季节，而冬天则是寒冷和雪地的季节。不过，这些季节的确切时间可能会因地理位置、气候变
化以及地区特有的季节定请输入您的问题：
```

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:



```
============================
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：你能告诉我世界最高峰是哪个吗
Q: 你能告诉我世界最高峰是哪个吗
A: 您好，世界最高峰是珠穆朗玛峰，位于喜马拉雅山脉，海拔8,848米。请输入您的问题：
```

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:

```
============================
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：你能告诉我世界最高峰是哪个吗
Q: 你能告诉我世界最高峰是哪个吗
A:

世界最高峰是 Mount Everest，它海拔 8,848 米。请输入您的问题：
```

g. Use the InternLM2 model on the development board server side and test it on the Ubuntu PC side, as shown in the following figure:

```
============================
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：你能告诉我世界最高峰是哪个吗
Q: 你能告诉我世界最高峰是哪个吗
A:当然可以，世界最高峰是位于尼泊尔的珠穆朗玛峰。它高达8848米（或8,848.86米），是地球上最高的山峰。这座山位于喜马拉雅山脉中，由印度板块和欧亚板块碰撞形成。珠穆朗玛峰在夏季和冬季都吸引着来自全球各地的登山者。请输入您的问题：
```

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 performs poorly using this method and is not recommended.

```
============================
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：What is the highest peak in the world called
Q: What is the highest peak in the world called
A: What does this mean?请输入您的问题：世界最高峰是哪个
Q: 世界最高峰是哪个
A:系统 您正在使用Assistant服务。 Assistor是您的私人助手，可以回答各种问题并帮助解决疑问。请随时告诉我您需要什么类型的协助！
用户：请告诉我们世界上最高的山峰是哪座山？请输入您的问题：
```

3. 39. 5. 3. Building a server based on Graph

3. 39. 5. 3. 1. Server side (development board side)

1) Firstly, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file from the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file to the development board based on 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 file rkllm_server/gradio_server.py to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so') and 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 graphics library on the development board.

If using the Debian12 system, it is necessary to set the command pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple Add --break-system-packages after

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 of 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
```

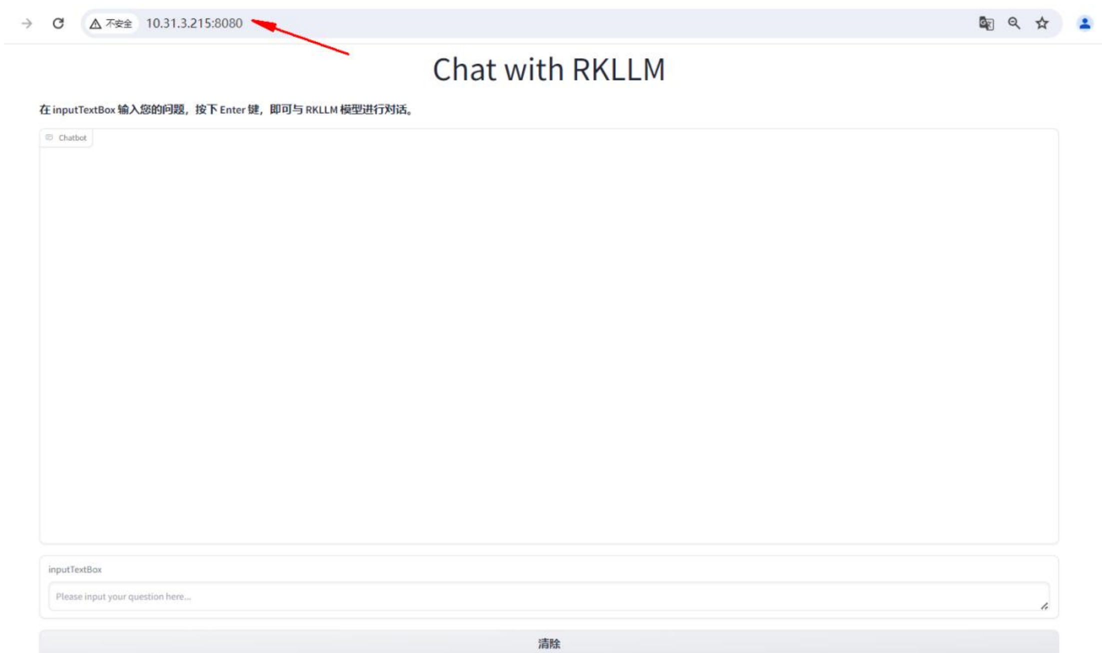
5) If successful, as shown in the following figure, the server-side is now configured.

In the figure http://0.0.0.0:8080 It does not mean that the IP address is this, the actual IP address that needs to be used is the user's own development board's actual address.

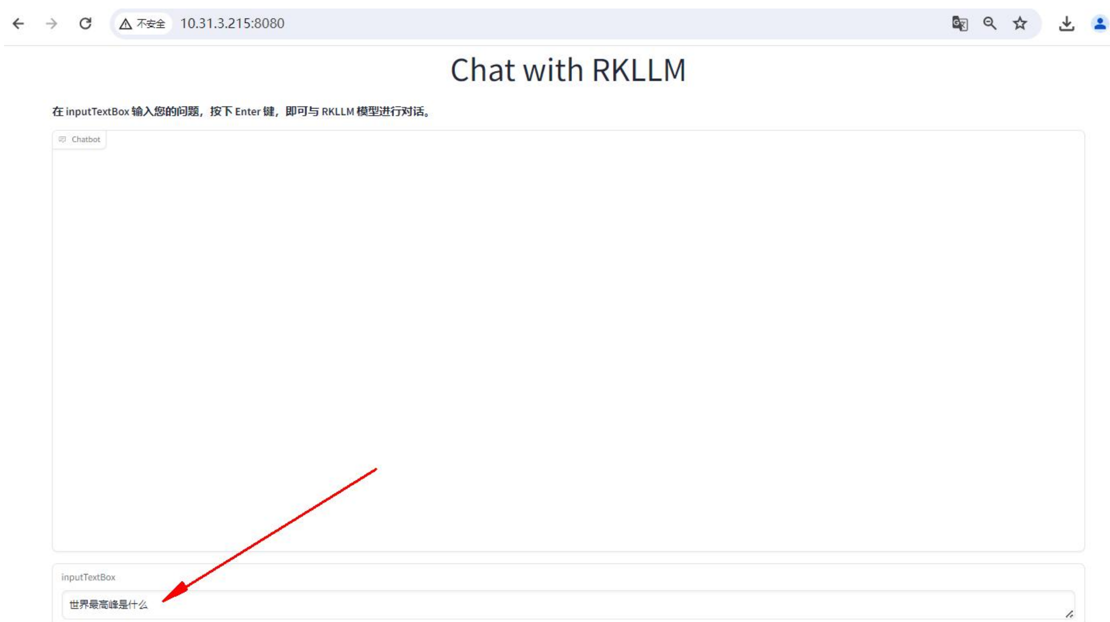
```
=========init....===========
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
RKLLM初始化成功！
==============================
Running on local URL:  http://0.0.0.0:8080

To create a public link, set `share=True` in `launch()`.
```

3. 39. 5. 3. 2. Client（Ubuntu PC）

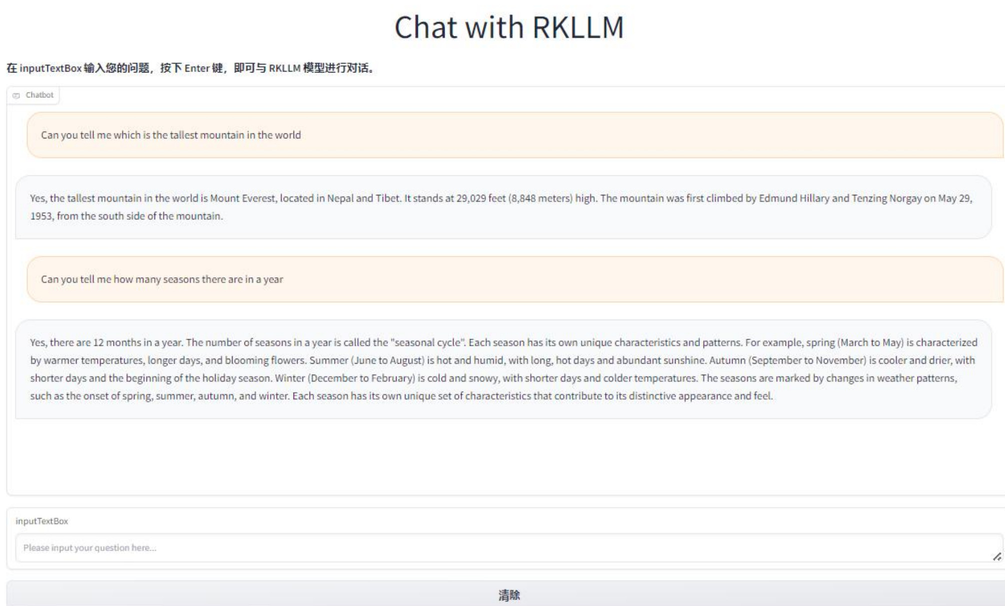
1) Firstly, open a browser on any computer in the current local area network and directly access "Development Board IP: 8080". The open interface is shown in the following figure:



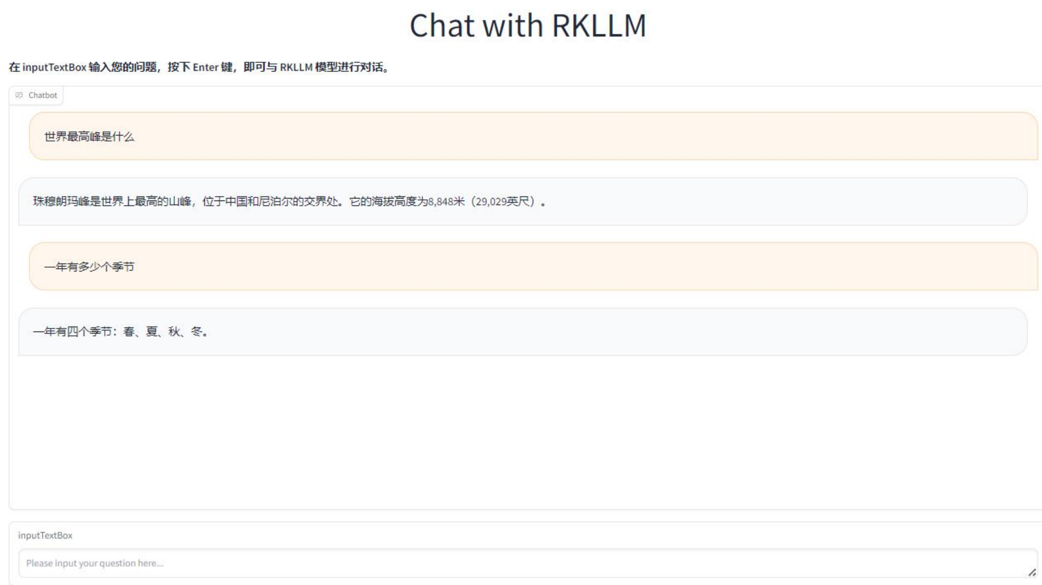
2) Then enter the question in the inputTextBox input box and press enter.



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:

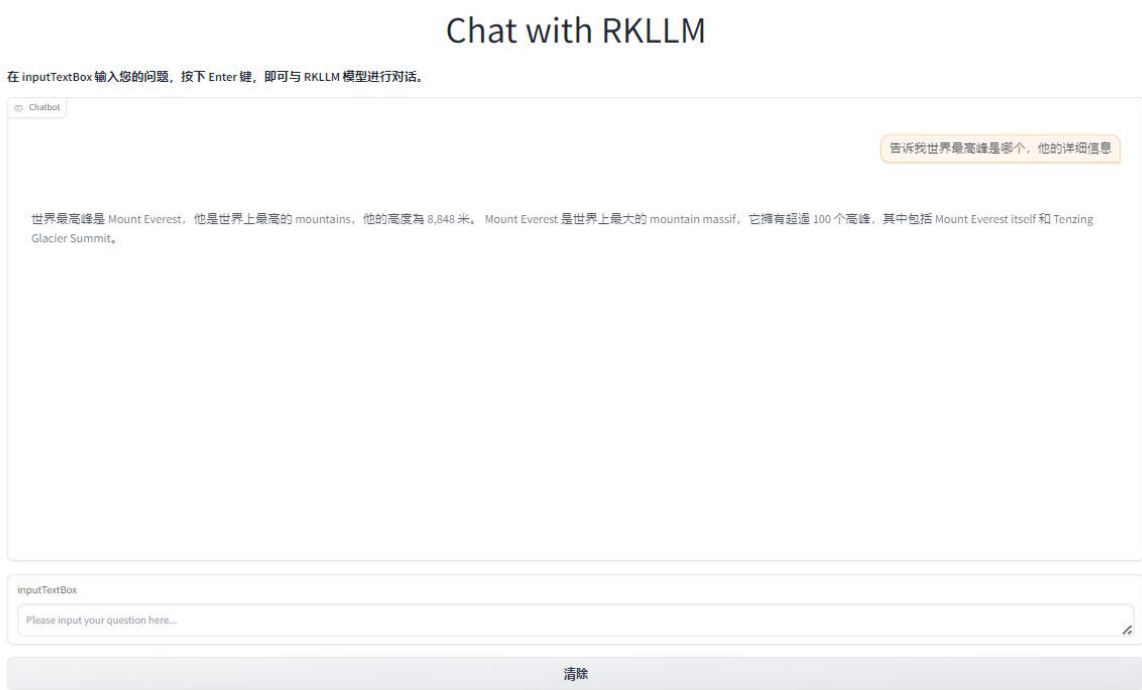


c. Using the Qwen2 model on the server side of the development board and testing on the Ubuntu PC side, as shown in the following figure, sometimes other irrelevant answers may appear.

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 development board server side 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. 39. 6. Performance test results of running RKLLM large model on RK3588

1) In order to conduct large-scale model performance testing, the first step is to download the large-scale model performance testing file **main.cpp** from the **official tool**. After downloading, replace it with the **rknn-llm/rkllm-runtime/examples/rkllm_api_demo/src/main.cpp** file used on the PC to compile the testing code

2) Refer to the section on **compiling test code** to recompile the llm_demo file, and then

run the large model according to **the detailed steps for deploying and running on the development board**.

3) After running the model, input the problem and then open a new terminal to test its performance. Performance testing is conducted when the model answers questions.

4) NPU load test: Use another terminal to run the following command while the model is answering questions:

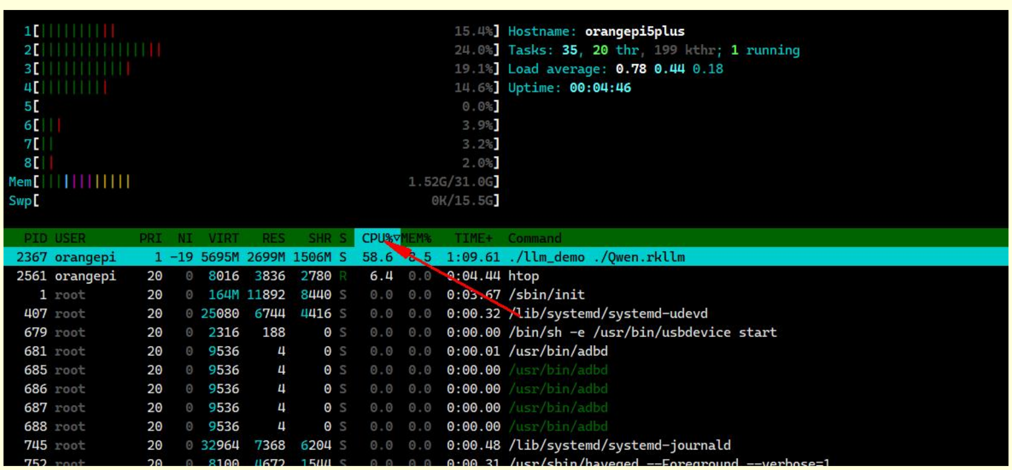
```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/load
NPU load:    Core0:    51%, Core1:    51%, Core2:    51%,
```

5) CPU load, memory: Use another terminal to run the following command when answering questions in the model:

When calculating CPU load, calculate the CPU% value/number of CPUs for the llm_demo process

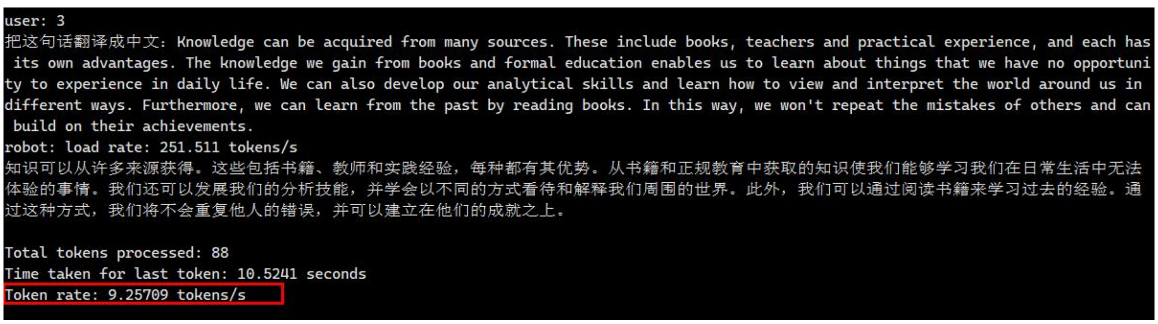
When calculating memory, use the MEM% value of the llm_demo process multiplied by the total amount of MEM

You can click on the CPU options, and the interface will display in descending order of CPU usage.




```
orangepi@orangepi:~$ htop
```

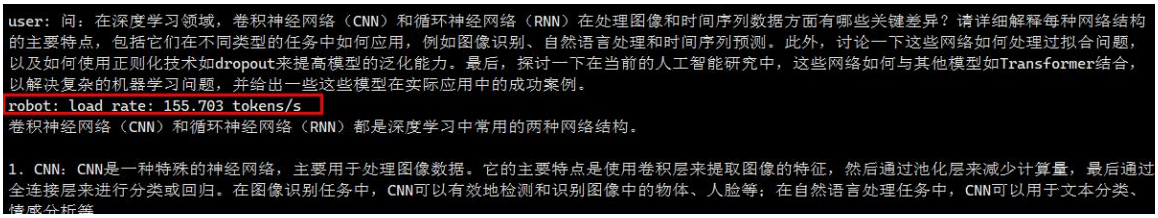
6) Inference: Inference speed, abbreviated as inference, refers to the number of tokens output during model inference divided by the time taken for model inference. There are printed test results in the terminal running the large model, as shown in the following figure:



7) Pre fill: Calculate the number of input tokens divided by the time it takes for the model to run and output the first token. Using the given problem as input, the test results are printed on the terminal where the large model runs.

Due to the fact that different large-scale language models may use different segmentation strategies when processing the same sentence, resulting in differences in the number of generated tokens, and the actual number of input tokens is not provided with corresponding acquisition channels in RKLLM, we used GPT to generate a problem with 256 tokens as input. Resulting in certain errors in the test results.

Q: What are the key differences between Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in processing image and time series data in the field of deep learning? Please provide a detailed explanation of the main characteristics of each network architecture, 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 handle overfitting problems and how to use regularization techniques such as dropout to improve the model's generalization ability. Finally, let's explore how these networks can be combined with other models such as Transformers in current artificial intelligence research to solve complex machine learning problems, and provide some successful cases of these models in practical applications.



8) The test results of all models are shown in the following table:

Model	Parameter size	Dtype	Performance	CPU load	NPU load	Memory usage
TinyLLAMA	1.1B	W8a8	Pre filled: 58.6157 token/s Reasoning： 12.7262 token/s	15.9%	3*49%	1.376G
Qwen	1.8B	W8a8	Pre filled：168.525 token/s Reasoning： 10.8891 token/s	13.7%	3*50%	2.72G
Qwen2	0.5B	W8a8	Pre filled：440.511 token/s Reasoning： 17.4542 token/s	17.75%	3*34%	1.344G
Phi-3	3.8B	W8a8	Pre filled：22.8119 token/s Reasoning： 4.72983 token/s	13.13%	3*62%	4.288G
ChatGLM3	6B	W8a8	Pre filled：48.8464 token/s Reasoning： 3.80383 token/s	8.3%	3*75%	7.04G

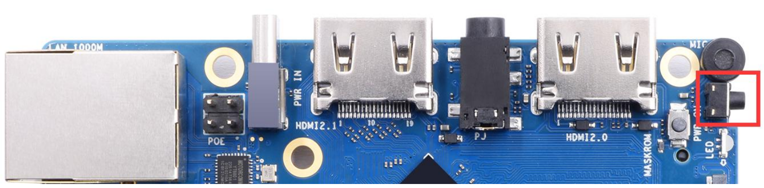
Gemma	2B	W8a8	Pre filled: 112.489 token/s Reasoning : 6.41746 token/s	8.25%	3*64%	4.8G
InternLM2	1.8B	W8a8	Pre filled: 117.099 token/s Reasoning : 9.139 token/s	11.87%	3*57%	2.432G
MiniCPM	2B	W8a8	Pre filled: 77.4655 token/s Reasoning : 6.16648 token/s	16.25%	3*52%	3.904G

3. 40. How to shut down and restart the development board

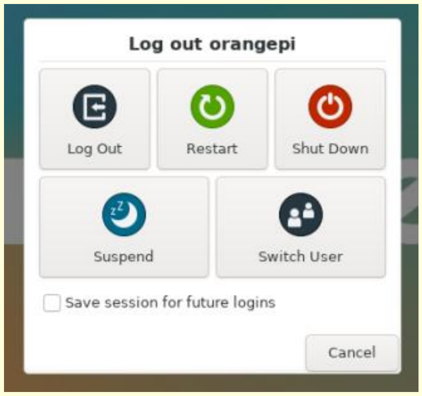
1) During the running of the Linux system, if you directly unplug the Type-C power supply and cut off the power, the file system may lose some data or be damaged. Therefore, please use the **poweroff** command to shut down the Linux system of the development board before cutting off the power. Then unplug the power supply.

```
orangepi@orangepi:~$ sudo poweroff
```

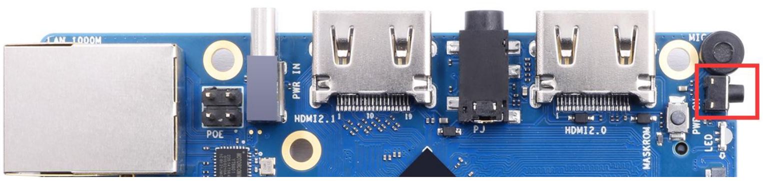
2) In addition, the development board is equipped with a power on/off button, and you can also short press the power on/off button on the development board to shut down.



Note that after pressing the power button on the Linux desktop version, the confirmation box shown in the figure below will pop up. You need to click the Shut Down option before shutting down.



3) After shutting down, short press the power button on the development board to turn it on.



4) The command to restart the linux system is

orangepi@orangepi:~$ **sudo reboot**

4. Orange Pi OS Arch System Instructions

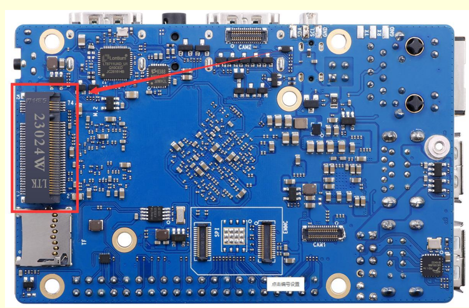
4. 1. Orange Pi OS Arch system adaptation status

Function	OPi OS Arch Gnome Wayland
HDMI 2.1 Video	OK
HDMI 2.1 Video	OK
HDMI 2.0 Audio	OK
HDMI 2.0 Audio	OK
USB2.0x3	OK
USB3.0x1	OK
Gigabit network port	OK
Network port status light	OK
WIFI	OK
Bluetooth	OK
Debug serial port	OK
RTC	OK
FAN interface	OK
Emmc Extension ports	OK
GPIO（40pin）	OK
UART（40pin）	OK
SPI（40pin）	OK
I2C（40pin）	OK
CAN（40pin）	OK
PWM（40pin）	OK
TF card boot	OK
OV13850 Camera	OK
OV13855 Camera	OK
SPI+NVME start up	OK
SPI+SATA start up	OK
LCD	OK
MIC	OK
Headphone playback	OK

Headphone recording	**OK**
Tri-color LED light	**OK**
GPU	**OK**
NPU	**NO**
VPU	**OK**
On/off button	**OK**
watchdog test	**OK**
Chromium Hard solution video	**NO**
MPV hard solution video	**OK**

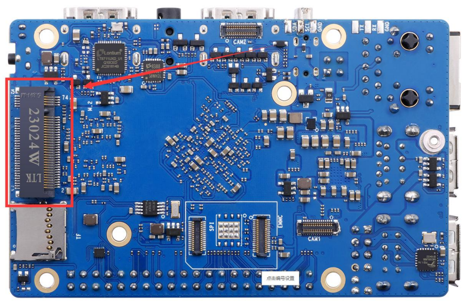
4. 2. How to use SATA SSD in OPi OS Arch system

The m.2 interface shown in the figure below can use either nvme ssd or sata ssd. Since the pcie2.0 controller and the sata controller are optional, only one of the configurations can be opened at the same time. The OPi OS Arch 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

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



3) There are two main ways to use sata ssd：

a. The OPi OS Arch system is installed in the tf card, and then the sata ssd is inserted as an external storage device. This section mainly explains this usage.

b. Burn the OPi OS Arch system into the sata ssd, and then start the OPi OS Arch system in the sata ssd. For this usage, please see the instructions in the section "**How to Burn Linux Image to SPIFlash+SATA SSD**"

4) Then add the following configuration to **/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-sata2.dtbo #Configuration that needs to be added
```

5) **Then restart the OPi OS Arch system**

6) If everything is normal, use the **sudo fdisk -l** command after the system restarts to see the 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
.......
```

4. 3. How to use the 10.1-inch MIPI LCD screen

4. 3. 1. 10.1-inch MIPI screen assembly method

1) First prepare the necessary accessories

a. 10.1 inch MIPI LCD display + touch screen

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

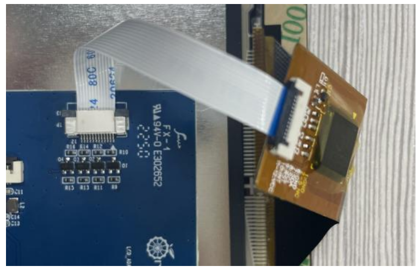
c. 30pin MIPI cable

d. 12pin touch screen cable

2) Connect the 12pin touch screen cable, 31pin to 40pin cable, and 30pin MIPI cable to the screen adapter board as shown below. Note that **the blue insulated side of the touch screen cable is facing down**, and the insulated side of the other two cables is facing up. , if connected incorrectly, it will cause no display or inability to touch.

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

4) Then connect the touch screen and the adapter board through the 12pin touch screen cable. Pay attention to the orientation of the insulation surface.

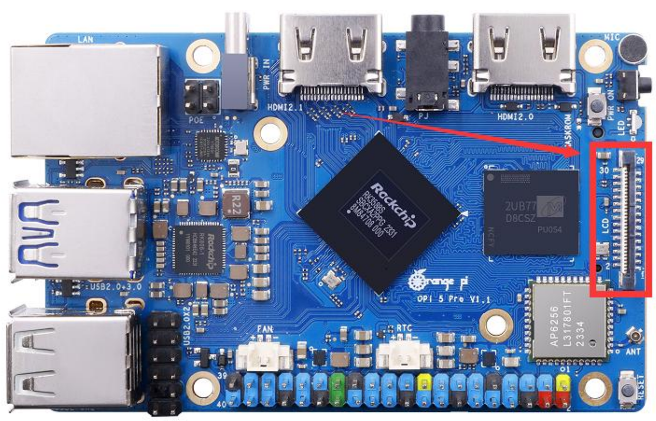


5) Finally, connect it to the LCD interface of the development board through a 30pin MIPI cable.

4. 3. 2. How to open the 10.1-inch MIPI LCD screen configuration

1) The OPi OS Arch image does not have the configuration to open the mipi LCD screen by default. If you need to use the mipi LCD screen, you need to open it manually.

2) The interface of the mipi lcd screen on the development board is as shown below:

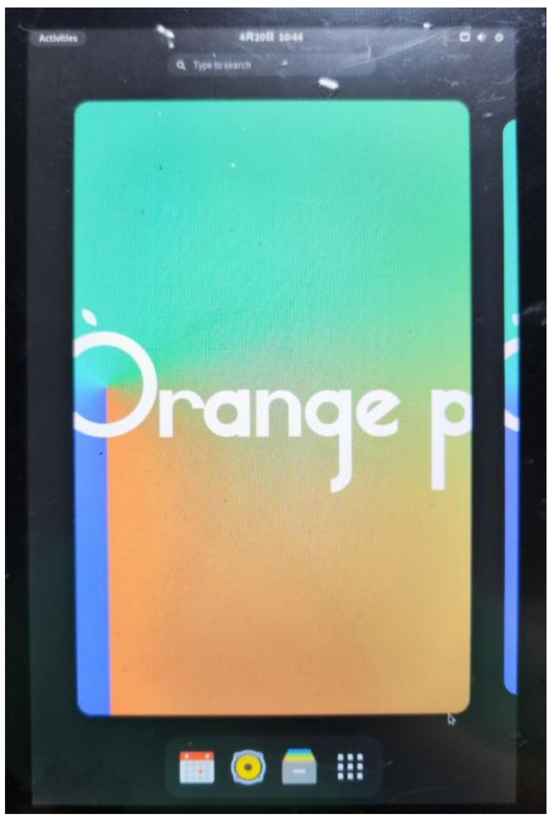


3) The method to open mipi lcd configuration is as follows:

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5-pro.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-opi5pro-lcd.dtbo #Configuration that needs to be added
```

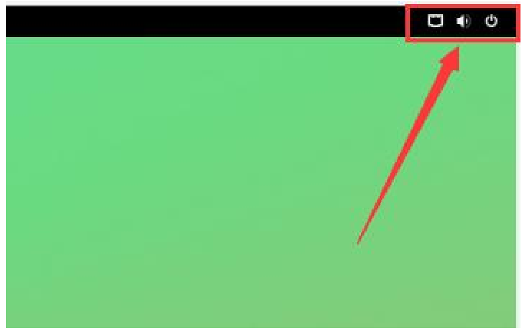
4) **Then restart the OPi OS Arch system**

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

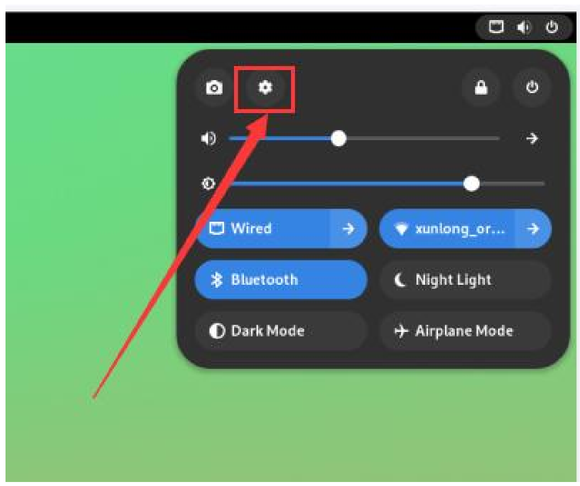


4. 3. 3. Methods of rotating display and touch direction

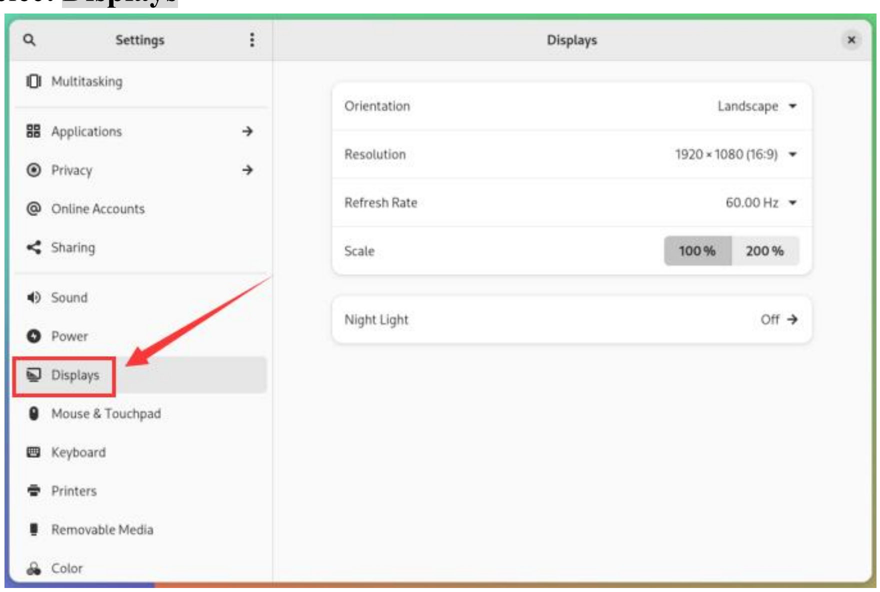
1) First click on the area in the upper right corner of the desktop



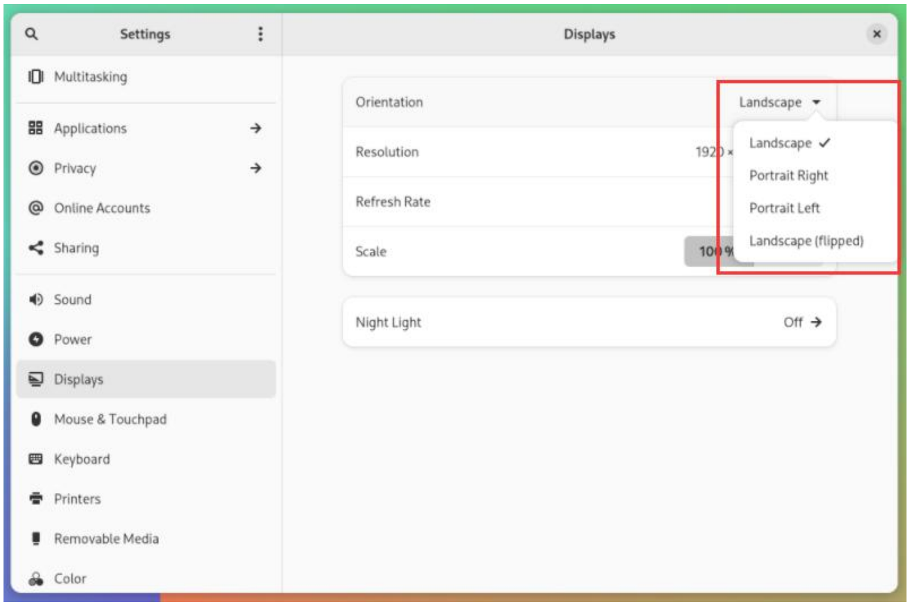
2) Then open settings



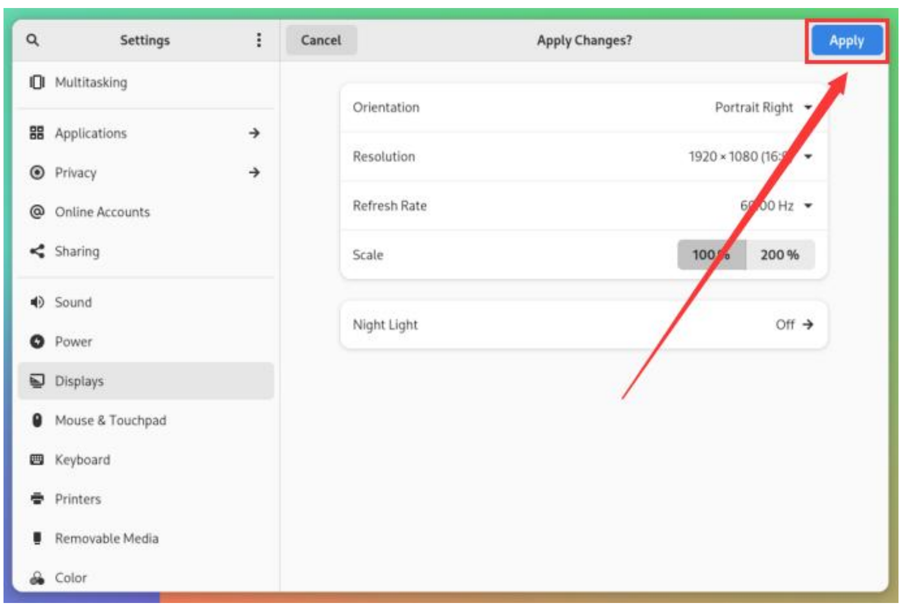
3) Then select **Displays**



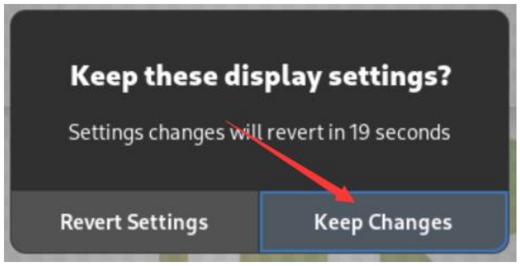
4) Then select the direction you want to rotate in the **Orientation** of **Displays**



5) Then select **Apply**



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



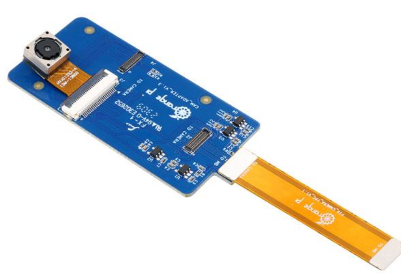
7) The display after the LCD screen is rotated 90 degrees is as follows:：

8) **The touch function of the OPi OS Arch system LCD screen will rotate with the rotation of the display direction, no other settings are required.**

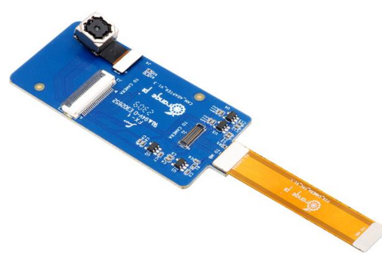
4.4. Testing methods for OV13850 and OV13855 MIPI cameras

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



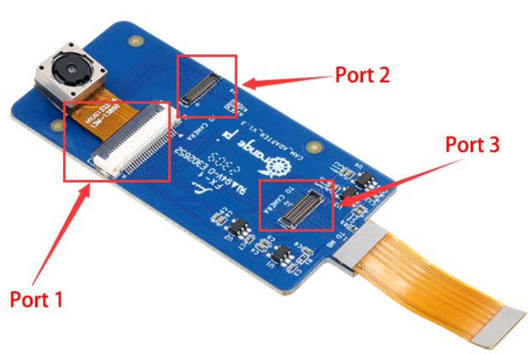
b. OV13855 camera with 13 million MIPI interface



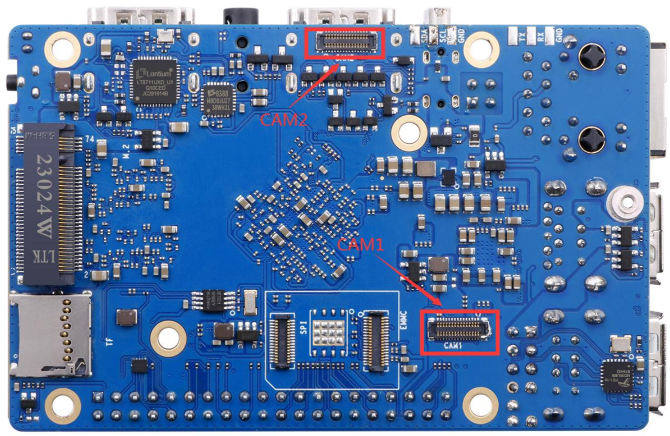
The adapter board and FPC cable used by the OV13850 and OV13855 cameras are the same, but the locations where the two cameras are connected to the adapter board are different. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked **TO MB** needs to be plugged into the camera interface of the development board, and the end marked **TO CAMERA** needs to be plugged into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board, and only one can be connected and used at the same time, as shown in the figure below, among which：

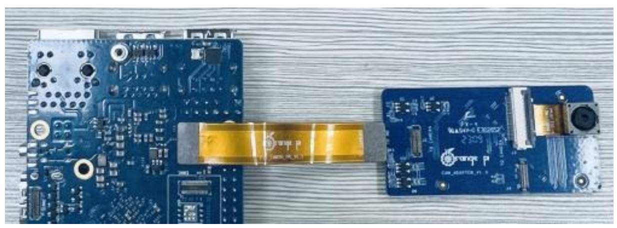
d. **Connect the OV13850 camera to interface 1**
e. **Connect the OV13855 camera to interface 2**
f. Interface 3 is not used, so just ignore it.



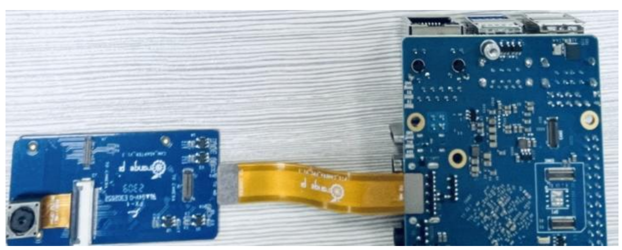
There are a total of 2 camera interfaces on the Orange Pi 5 Pro development board. We define the positions of Cam1 and Cam2 as shown in the figure below：



The method of plugging the camera into the Cam1 interface of the development board is as follows:



The method of plugging the camera into the Cam2 interface of the development board is as follows:



After connecting the camera to the development board, we can use the following method to test the camera:

a. First add the following configuration to **/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-opi5pro-cam1.dtbo #Configuration that needs to be added

The red font above demonstrates the configuration of opening the **Cam1** interface.

The configuration of other interfaces is as shown in the following table. Just add the corresponding dtbo configuration after **FDTOVERLAYS**. If you want to add multiple configurations at the same time, separate them with spaces.

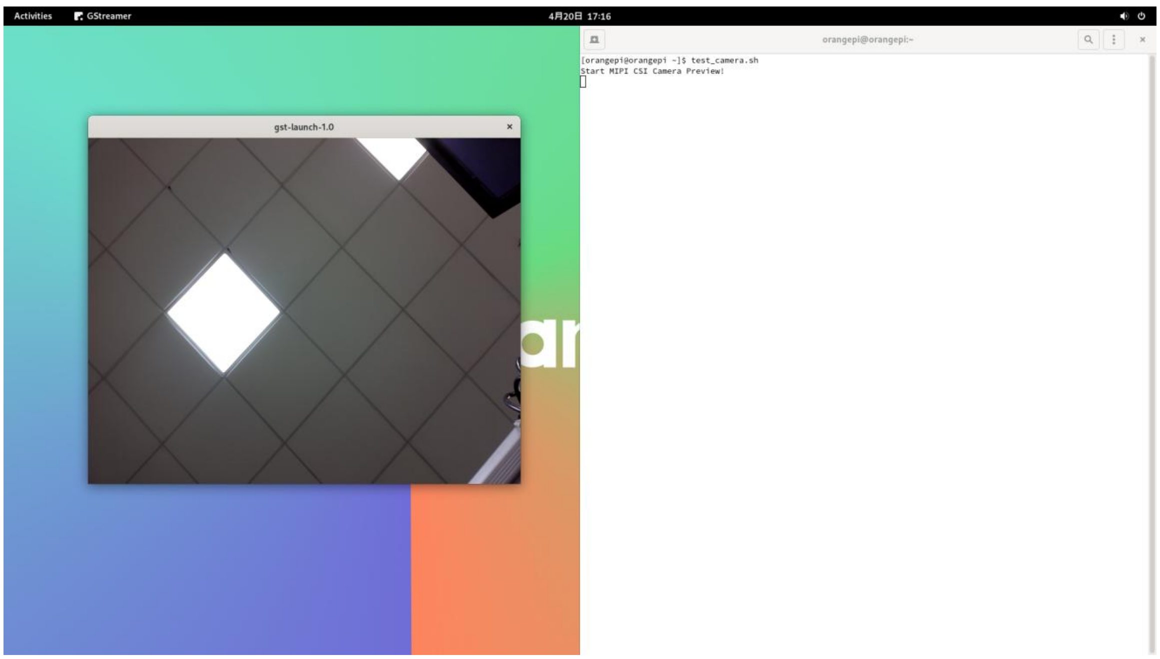
Camera	dtbo configuration
Cam1	**/dtbs/rockchip/overlay/rk3588-opi5pro-cam1.dtbo**
Cam2	**/dtbs/rockchip/overlay/rk3588-opi5pro-cam2.dtbo**

b. **Then restart the OPi OS Arch system**

c. Then open a terminal on the desktop system and run the following script

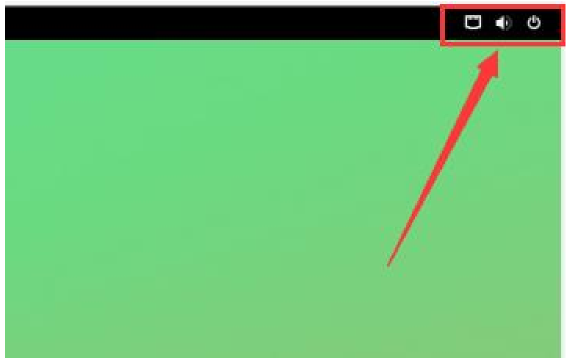
```
orangepi@orangepi:~$ test_camera.sh
```

d. Then you can see the preview screen of the camera

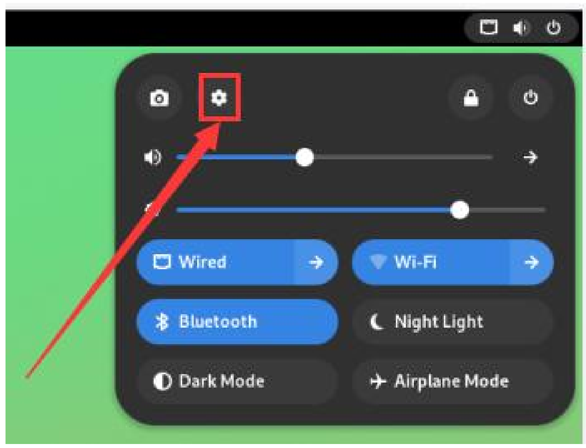


4. 5. How to set up the Chinese environment and install the Chinese input method

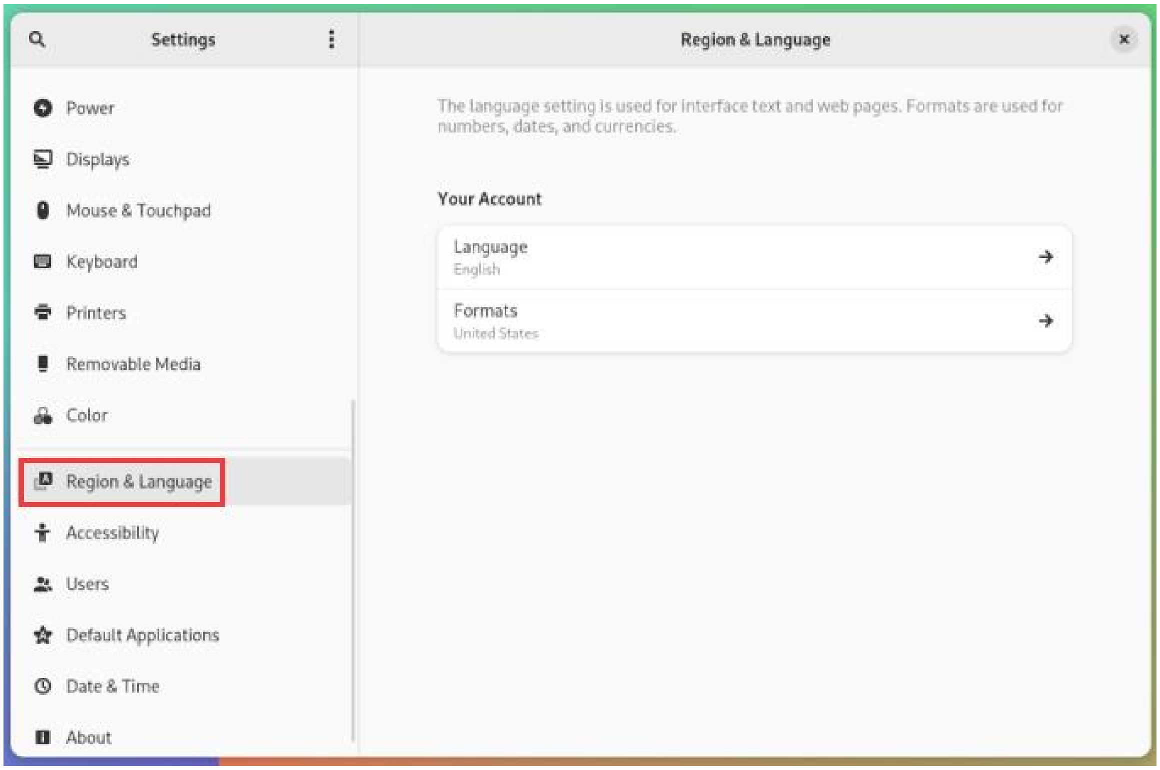
1) First click on the area in the upper right corner of the desktop



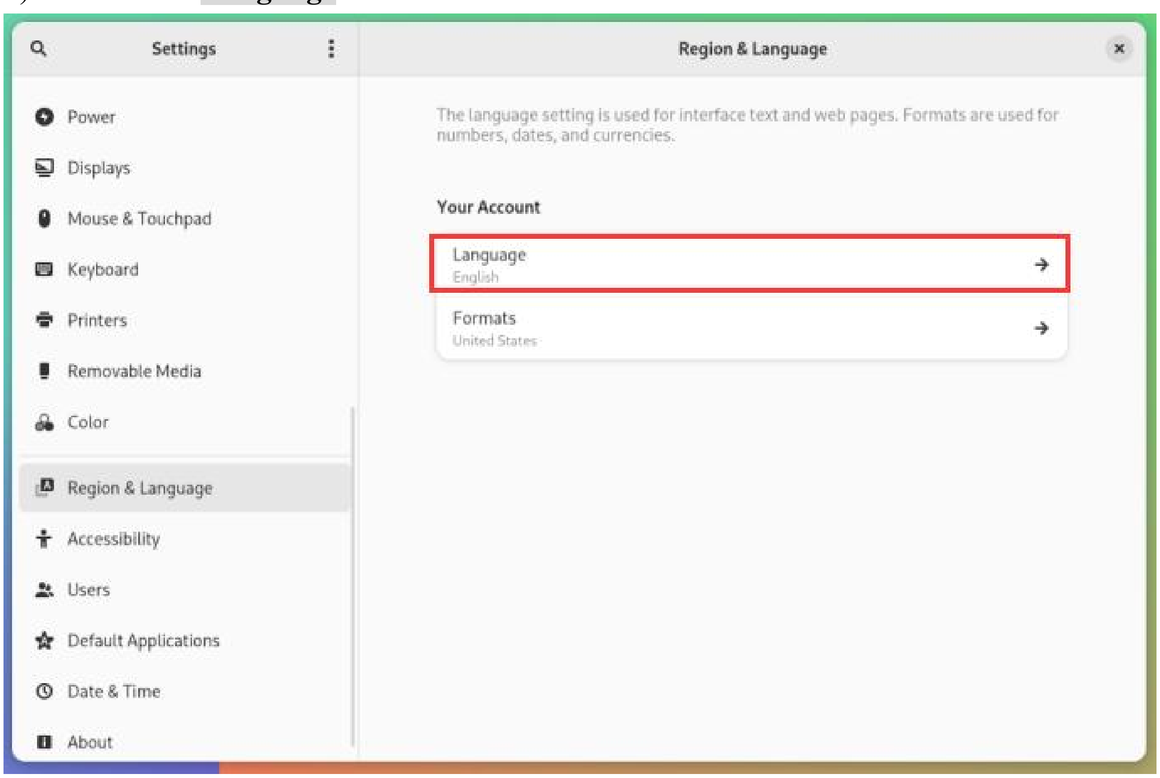
2) Then open settings



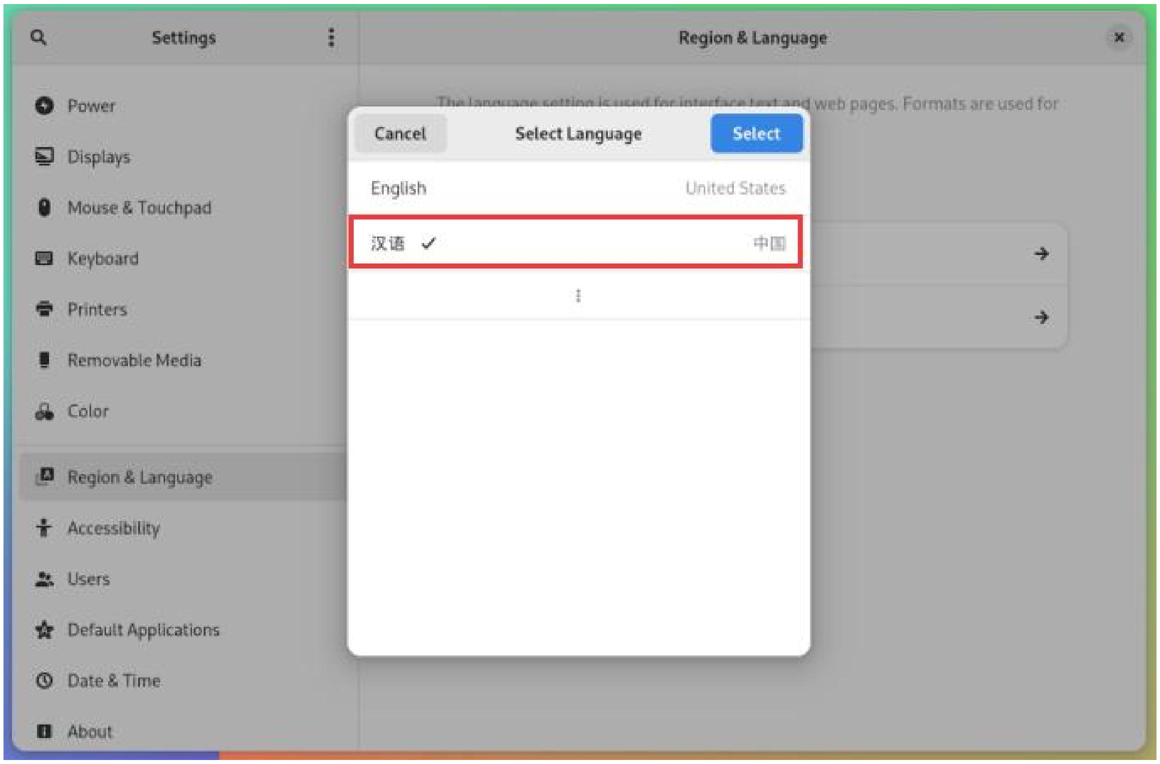
3) Then find the **Region & Language** option



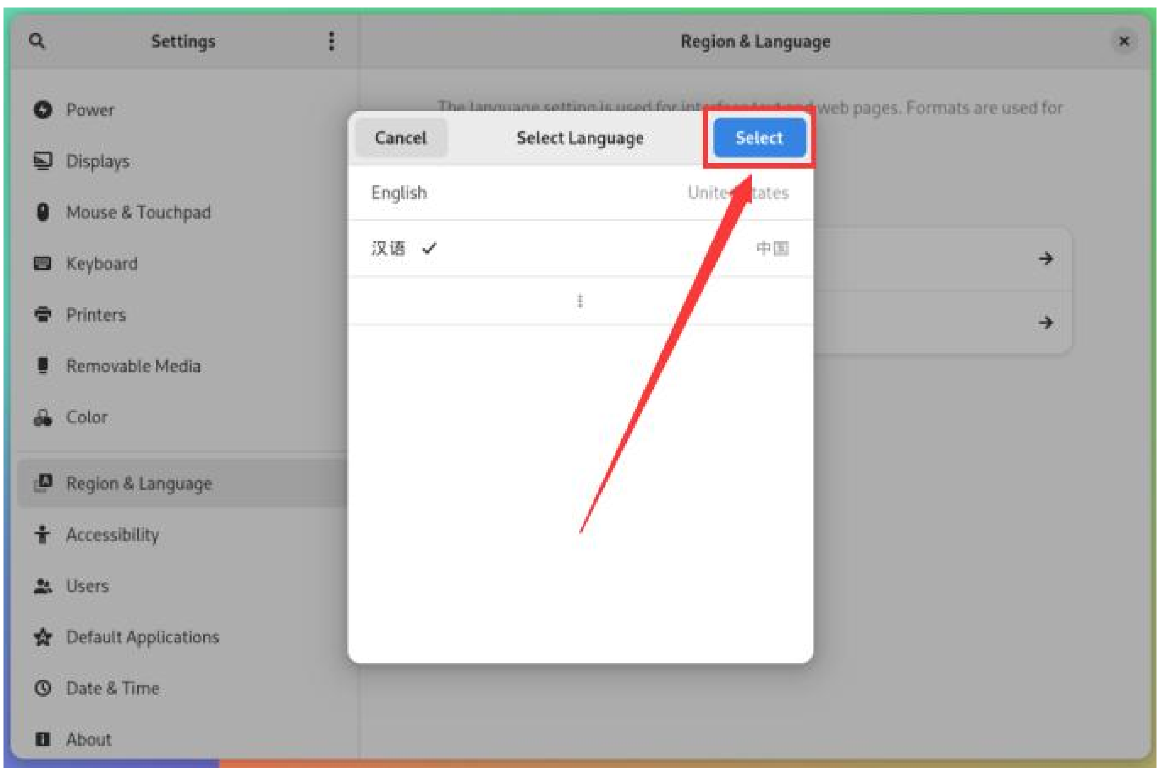
4) Then select **Language**



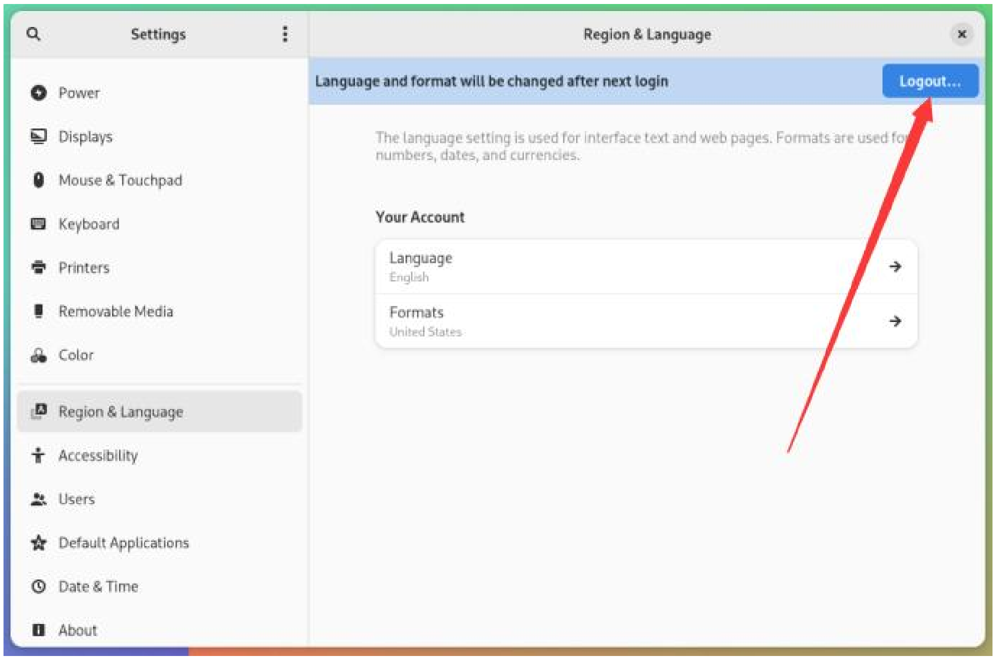
5) Then select Chinese



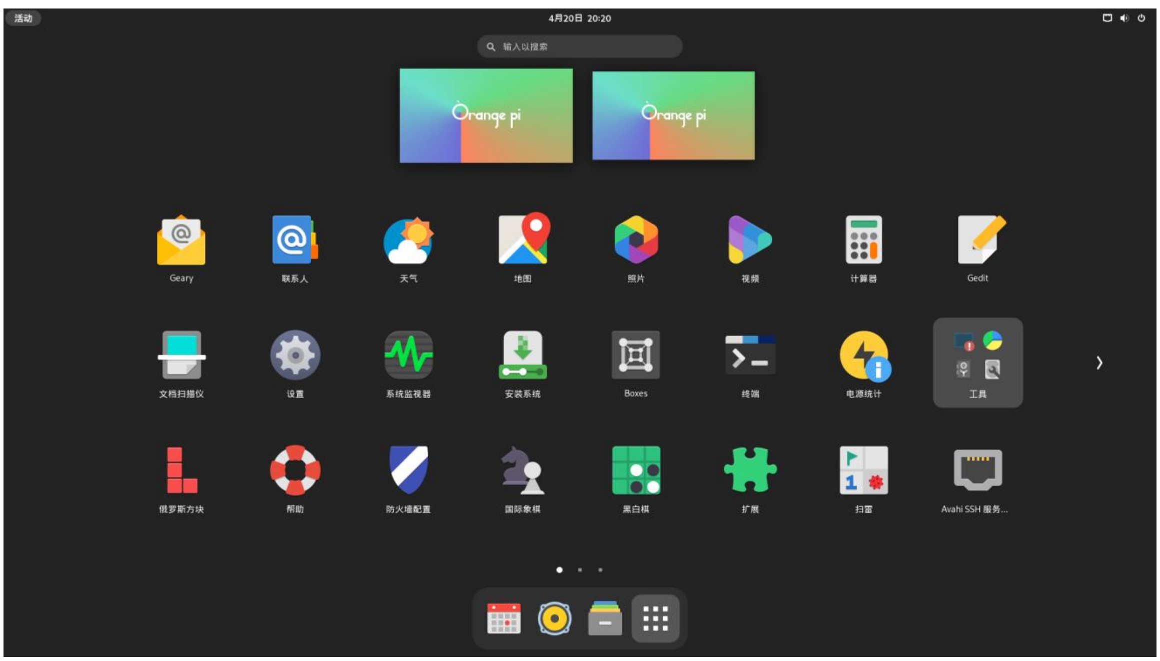
6) Then click**Select**



7) Then click **Logout...** to log out of the system, and then log in again.



8) Then you can see that the desktop is displayed in Chinese

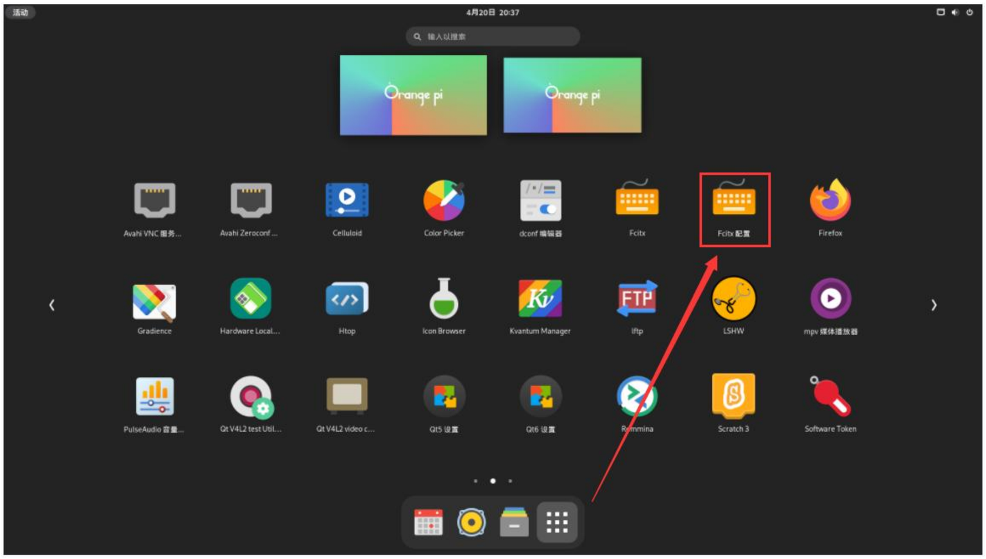


9) Then install fcitx-im and fcitx-configtool

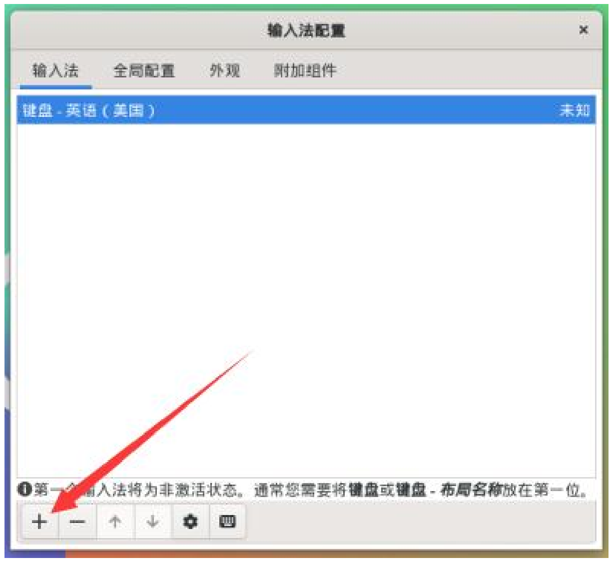
```
[orangepi@orangepi ~]$ sudo pacman -S fcitx-im fcitx-configtool
:: There are 3 members in group fcitx-im:
:: software warehouse community
   1) fcitx   2) fcitx-qt5   3) fcitx-qt6
```

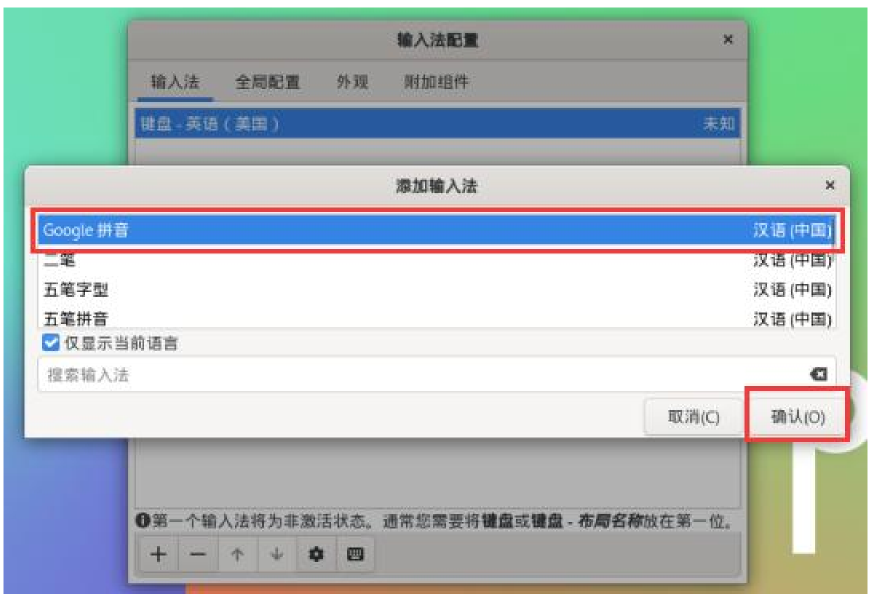
```
Enter a selection (default=all selected): 1
```

10) Then open the Fcitx configuration program

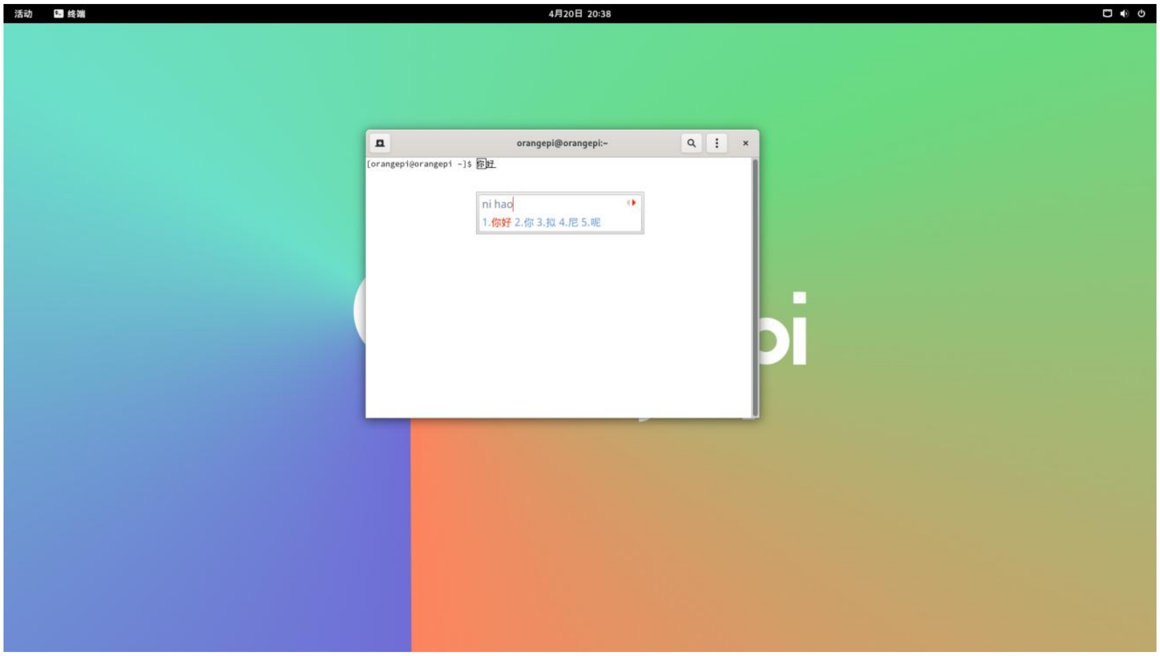


11) Then add **Google Pinyin** input method





12) Then we can open a terminal to test the Chinese input method. After opening the terminal, if the English input method is still the default, we can switch to the Chinese input method through the **Ctrl+Space** shortcut key, and then we can enter Chinese.



4. 6. How to install wiringOP

Note that wiringOP is already pre-installed in the OPi OS Arch image released by Orange Pi. Unless the wiringOP code is updated, there is no need to re-download, compile and install it, you can just use it directly.

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

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   46 |   8 | GPIO1_B6 |     IN | 1 | 15 || 16 | 0 | IN     | TXD.6    | 9   | 33   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | RXD.6    | 10  | 32   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | GPIO1_B0 | 13  | 40   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 |   34 |  17 |    SDA.4 |     IN | 0 | 27 || 28 | 0 | IN     | SCL.4    | 18  | 35   |
 |   36 |  19 | GPIO1_A4 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |   38 |  20 | GPIO1_A6 |     IN | 0 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |   63 |  22 |    PWM15 |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  135 |  23 | GPIO4_A7 |     IN | 0 | 35 || 36 | 0 | IN     | TXD.0    | 24  | 131  |
 |  134 |  25 | GPIO4_A6 |     IN | 0 | 37 || 38 | 0 | IN     | RXD.0    | 26  | 132  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO4_A5 | 27  | 133  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
[orangepi@orangepi ~]$
```

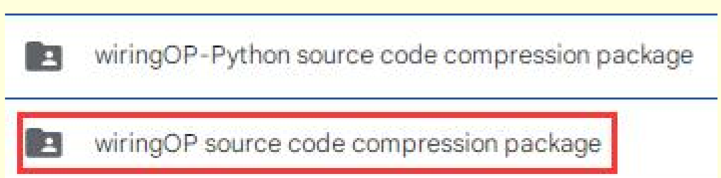
1) Download the code of wiringOP

[orangepi@orangepi ~]$ **sudo pacman -Syy git**

[orangepi@orangepi ~]$ **git clone https://github.com/orangepi-xunlong/wiringOP.git -b next**

Note that Orange Pi 5 Pro 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 download the source code compressed package of wiringOP.tar.gz from the official tool on the Orange Pi 5 Pro 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) The output of the test gpio readall command is as follows

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |   CAN1_RX |    IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |   CAN1_TX |    IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   46 |   8 | GPIO1_B6 |     IN | 1 | 15 || 16 | 0 | IN     | TXD.6    | 9   | 33   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | RXD.6    | 10  | 32   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | GPIO1_B0 | 13  | 40   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 |   34 |  17 |    SDA.4 |     IN | 0 | 27 || 28 | 0 | IN     | SCL.4    | 18  | 35   |
 |   36 |  19 | GPIO1_A4 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |   38 |  20 | GPIO1_A6 |     IN | 0 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |   63 |  22 |    PWM15 |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  135 |  23 | GPIO4_A7 |     IN | 0 | 35 || 36 | 0 | IN     | TXD.0    | 24  | 131  |
 |  134 |  25 | GPIO4_A6 |     IN | 0 | 37 || 38 | 0 | IN     | RXD.0    | 26  | 132  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO4_A5 | 27  | 133  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
[orangepi@orangepi ~]$
```

4.7. 40pin interface GPIO, I2C, UART, SPI, CAN and PWM testing

Note that if you need to set up fdt overlays to open multiple configurations at the same time, please separate them with spaces and write them on one 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-pro.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c1-m4.dtbo /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo
```

4.7.1. 40pin GPIO port test

1) There are a total of 28 GPIO ports in the 40-pin development board that can be used.

The following uses pin No. 7 - the corresponding GPIO is GPIO1_B7 - the corresponding wPi serial number is 2 - as an example to demonstrate how to set the high and low levels of the GPIO port.

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

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

```
[orangepi@orangepi ~]$ 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 the low level is set successfully.

```
[orangepi@orangepi ~]$ gpio write 2 0
```

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

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |    OUT | 0 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

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 the setting of the high level is successful.

```
[orangepi@orangepi ~]$ gpio write 2 1
```

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

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |    OUT | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

5) The setting method for other pins is similar. You only need to modify the serial number of wPi to the serial number corresponding to the pin.

4.7.2. How to set the pull-down resistor of the 40pin GPIO port

Note that the four GPIO pins below Orange Pi 5 Pro have external 3.3V pull-ups, so the pull-down settings are invalid.

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   46 |   8 | GPIO1_B6 |     IN | 1 | 15 || 16 | 0 | IN     | TXD.6    | 9   | 33   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | RXD.6    | 10  | 32   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | GPIO1_B0 | 13  | 40   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 |   34 |  17 |    SDA.4 |     IN | 0 | 27 || 28 | 0 | IN     | SCL.4    | 18  | 35   |
 |   36 |  19 | GPIO1_A4 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |   38 |  20 | GPIO1_A6 |     IN | 0 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |   63 |  22 |    PWM15 |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  135 |  23 | GPIO4_A7 |     IN | 0 | 35 || 36 | 0 | IN     | TXD.0    | 24  | 131  |
 |  134 |  25 | GPIO4_A6 |     IN | 0 | 37 || 38 | 0 | IN     | RXD.0    | 26  | 132  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO4_A5 | 27  | 133  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
[orangepi@orangepi ~]$
```

1) The following uses pin No. 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 resistors of the GPIO port.

```
[orangepi@orangepi ~]$ gpio readall
 +------+-----+----------+--------+---+  PI5 PRO +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   59 |   0 |    SDA.1 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   58 |   1 |    SCL.1 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   47 |   2 |    PWM13 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | GPIO1_A7 | 6   | 39   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
```

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

[orangepi@orangepi ~]$ **gpio mode 5 in**

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

[orangepi@orangepi ~]$ **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.

[orangepi@orangepi ~]$ **gpio read 5**
1

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

[orangepi@orangepi ~]$ **gpio read 5**
0

4. 7. 3. 40pin SPI test

1) As can be seen from the picture below, the spi available for Orange Pi 5 Pro is spi0 and spi4

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of SPI0 and SPI4 in 40pin are as shown in the table below.

	SPI0_M2 correspond 40pin	**SPI4_M2 correspond 40pin**
MOSI	**Pin 19**	**Pin 16**
MISO	**Pin 21**	**Pin 18**
CLK	**Pin 23**	**Pin 27**
CS0	**Pin 24**	**Pin 28**
CS1	**Pin 26**	**None**
dtbo 配置	**spi0-m2-cs0-spidev** **spi0-m2-cs1-spidev** **spi0-m2-cs0-cs1-spidev**	**spi4-m2-cs0-spidev**

In the OPi OS Arch system, the spi function in 40pin is turned off by default and needs to be turned on manually to use it.

Add the configuration in red font below to /boot/extlinux/extlinux.conf, then restart the OPi OS Arch system to open spi0 and spi4. If you only need to open one, just fill in one.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5-pro.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-spi0-m2-cs0-cs1-spidev.dtbo
/dtbs/rockchip/overlay/rk3588-spi4-m2-cs0-spidev.dtbo
```

3) First check whether the device node of spidevx.x exists in the OPi OS Arch system. If it exists, it means that SPI4 has been set up and can be used directly.

```
[orangepi@orangepi ~]$ ls /dev/spidev*
```

```
/dev/spidev0.0    /dev/spidev0.1    /dev/spidev4.0
```

The above is the result displayed after opening spi0-m2-cs0-cs1-spidev and spi4-m2-cs0-spidev.

4) Without shorting the mosi and miso pins of SPI0 or 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/spidev0.0
Or
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev4.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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 short-circuit the mosi and miso pins of SPI0 or SPI4 and then run spidev_test. The output is as follows. You can see that the data sent and received are the same.

```
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev0.0
Or
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev4.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF 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    | ......@....▒.................▒.
```

4. 7. 4. 40pin I2C test

1) As can be seen from the table below, the i2c available for Orange Pi 5 is i2c1, i2c4, i2c5 and i2c8, a total of four groups 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			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of the 4 sets of I2C buses in 40pin are shown in the table below. I2C5_M2 and I2C5_M3 can only use one of them at the same time, not at the same time. They are the same I2C5, just connected to different pins. Please do not think that they are two different sets of I2C5 buses.

I2C bus	SDA corresponds to 40pin	SCL corresponds to 40pin	dtbo corresponding configuration
I2C1_M4	Pin 3	Pin 5	i2c1-m4
I2C4_M3	Pin 27	Pin 28	i2c4-m3
I2C5_M2	Pin 35	Pin 37	i2c5-m2
I2C5_M3	Pin 7	Pin 15	i2c5-m3
I2C8_M2	Pin 33	Pin 32	i2c8-m2

In the OPi OS Arch system, i2c in 40pin is turned off by default and needs to be turned on manually to use it.

Add the configuration in red font below to /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open i2c1, i2c4, i2c5 and i2c8 at the same time. If you only need to open one, just fill in one.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5-pro.dtb
FDTOVERLAYS   /dtbs/rockchip/overlay/rk3588-i2c1-m4.dtbo
/dtbs/rockchip/overlay/rk3588-i2c4-m3.dtbo
/dtbs/rockchip/overlay/rk3588-i2c5-m2.dtbo
/dtbs/rockchip/overlay/rk3588-i2c8-m2.dtbo
```

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

3) After starting the OPi OS Arch 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-4   /dev/i2c-6   /dev/i2c-9
/dev/i2c-1   /dev/i2c-2    /dev/i2c-5   /dev/i2c-7   /dev/i2c-8
```

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

Generally, only one of the 3.3v pin and the 5v pin can be connected. Please choose to connect the 3.3v pin or the 5v pin according to the specific i2c device connected.

5) 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 pacman -Syy i2c-tools
[orangepi@orangepi ~]$ sudo i2cdetect -y 1        #i2c1 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 4        #i2c4 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 5        #i2c5 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 8        #i2c8 commands
```

4. 7. 5. 40pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5 Pro are uart0, uart3, uart4, uart6 and uart7, a total of five groups 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			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

2) The corresponding pins of the five sets of UART buses in 40pin are as shown in the table below. UART4_M0 and UART4_M2 can only use one of them at the same time, not at the same time. They are the same UART4, just connected to different pins. Please do

not think that they are two different sets of UART4 buses.

UART bus	RX corresponds to 40pin	TX corresponds to 40pin	dtbo corresponding configuration
UART0_M2	Pin 38	Pin 36	uart0-m2
UART3_M2	Pin 37	Pin 40	uart3-m2
UART4_M0	Pin 3	Pin 5	uart4-m0
UART4_M2	Pin 19	Pin 23	uart4-m2
UART6_M1	Pin 18	Pin 16	uart6-m1
UART7_M2	Pin 24	Pin 26	uart7-m2

In the OPi OS Arch system, the uart in 40pin is turned off by default and needs to be turned on manually to use it.

Add the following configuration in red font to /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open uart0, uart3, uart4, uart6 and uart7 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-pro.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo
/dtbs/rockchip/overlay/rk3588-uart3-m2.dtbo
/dtbs/rockchip/overlay/rk3588-uart4-m0.dtbo
/dtbs/rockchip/overlay/rk3588-uart6-m1.dtbo
/dtbs/rockchip/overlay/rk3588-uart7-m2.dtbo
```

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

3) 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/ttyS3 /dev/ttyS4 /dev/ttyS6 /dev/ttyS7 /dev/ttyS9
```

4) Then start testing the uart interface. First use Dupont wire to short-circuit the rx and tx of the uart interface to be tested.

5) 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 the serial port communication is normal (ttySX needs to be replaced with the node name corresponding to the uart, please do not copy it)

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttySX
[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
```

4. 7. 6. PWM test method

1) As can be seen from the table below, the available pwms for Orange Pi 5 Pro include pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15, a total of six pwms.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

9) The corresponding pins of PWM in 40pin are as shown in the table below. PWM14_M1 and PWM14_M2, PWM15_M1 and PWM15_M3 can only use one of them at the same time, not at the same time. They are all the same PWM, just connected to different pins. Please do not think that they are two different PWM buses.

PWM bus	Corresponds to 40pin	dtbo corresponding configuration
PWM0_M2	**Pin 27**	**pwm0-m2**
PWM1_M2	**Pin 28**	**pwm1-m2**
PWM3_M3	**Pin 12**	**pwm3-m3**

PWM13_M2	**Pin 7**	**pwm13-m2**
PWM14_M1	**Pin 11**	**pwm14-m1**
PWM14_M2	**Pin 32**	**pwm14-m2**
PWM15_M1	**Pin 13**	**pwm15-m1**
PWM15_M3	**Pin 33**	**pwm15-m3**

In the OPi OS Arch system, pwm in 40pin is turned off by default and needs to be turned on manually to use it.

Add the configuration in red font below to /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15 at the same time. If you only need to open one, then fill in one That's it.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5-pro.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-pwm0-m2.dtbo
/dtbs/rockchip/overlay/rk3588-pwm1-m2.dtbo
/dtbs/rockchip/overlay/rk3588-pwm3-m3.dtbo
/dtbs/rockchip/overlay/rk3588-pwm13-m2.dtbo
/dtbs/rockchip/overlay/rk3588-pwm14-m1.dtbo
/dtbs/rockchip/overlay/rk3588-pwm15-m1.dtbo
```

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

2) When a pwm is opened, there will be an additional pwmchipX (X is a specific number) in**/sys/class/pwm/**. For example, after opening pwm13, the pwmchipX under **/sys/class/pwm/** will be Two became three

```
[orangepi@orangepi ~]$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

3) Which pwmchip above corresponds to pwm13? Let's first check the output of the ls **ls /sys/class/pwm/ -l** command, as shown below：

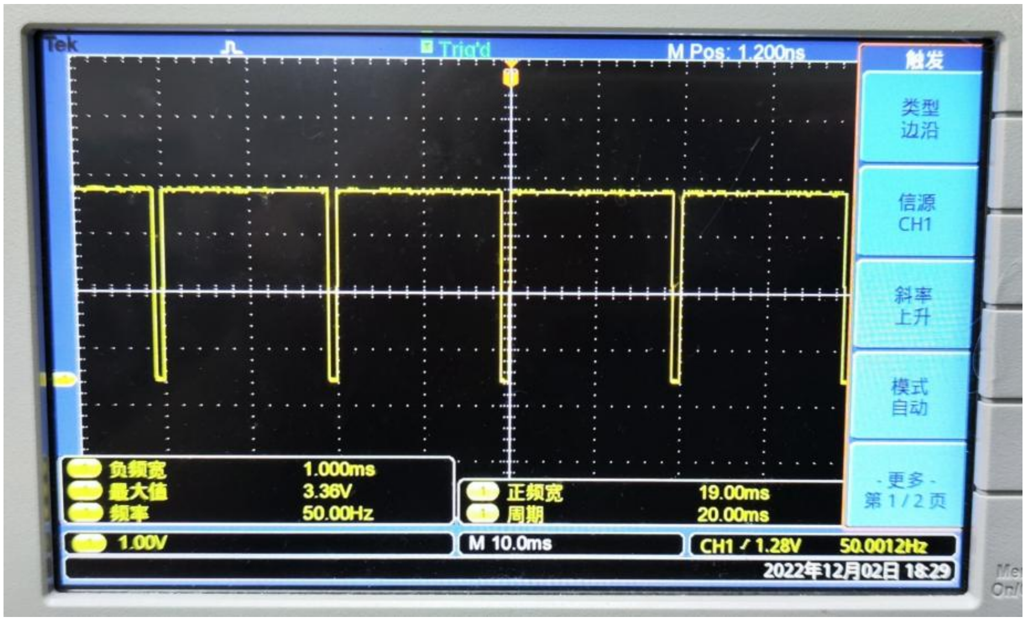
```
[orangepi@orangepi ~]$  ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Jan  1  2021 pwmchip0 -> ../../devices/platform/fd8b0020.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Jan  1  2021 pwmchip1 -> ../../devices/platform/febd0020.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Jan  1  2021 pwmchip2 -> ../../devices/platform/febf0010.pwm/pwm/pwmchip2
[orangepi@orangepi ~]$
```

4) From the following table, we can see that the base address of the pwm13 register is febf0010. Looking at the output of the **ls /sys/class/pwm/ -l** command, we can see that pwmchip2 is linked to febf0010.pwm, so the corresponding pwmchip of pwm13 is pwmchip2

复用功能	复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
				3.3V		1
		UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3
		UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5
	UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7
				GND		9
I2C7_SCL_M3		PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11
I2C7_SDA_M3		PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13
	UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15
				3.3V		17
		UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19
			SPI0_MISO_M2	GPIO1_B1	41	21
		UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23
				GND		25
UART6_RTSN_M1	SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27
				GPIO1_A4	36	29
				GPIO1_A6	38	31
UART1_CTSN_M1		I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33
SPI2_CS0_M1			I2C5_SDA_M2	GPIO4_A7	135	35
SPI2_CLK_M1		UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37
				GND		39

5) Then use the following command to make pwm13 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 test method of pwm13 demonstrated above is similar to other pwm test methods.

4. 7. 7. CAN test method

1) As can be seen from the table below, the CAN bus available for Orange Pi 5 Pro is CAN1, a total of one 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			
	UART4_RX_M0	I2C1_SDA_M4	GPIO1_D3	59	3	4		5V			
	UART4_TX_M0	I2C1_SCL_M4	GPIO1_D2	58	5	6		GND			
UART1_RX_M1	I2C5_SDA_M3	PWM13_M2 (febf0010)	GPIO1_B7	47	7	8	13	GPIO0_B5	UART2_TX_M0		
			GND		9	10	14	GPIO0_B6	UART2_RX_M0		
	PWM14_M1 (febf0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	39	GPIO1_A7	PWM3_IR_M3 (fd8b0030)		
	PWM15_IR_M1 (febf0030)	CAN1_TX_M1	GPIO4_B3	139	13	14		GND			
UART1_TX_M1	I2C5_SCL_M3		GPIO1_B6	46	15	16	33	GPIO1_A1	UART6_TX_M1		SPI4_MOSI_M2
			3.3V		17	18	32	GPIO1_A0	UART6_RX_M1		SPI4_MISO_M2
	UART4_RX_M2	SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND			
		SPI0_MISO_M2	GPIO1_B1	41	21	22	40	GPIO1_B0			
	UART4_TX_M2	SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2	
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2	
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	27	28	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			GPIO1_A4	36	29	30		GND			
			GPIO1_A6	38	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
	I2C8_SDA_M2	PWM15_IR_M3 (febf0030)	GPIO1_D7	63	33	34		GND			
		I2C5_SDA_M2	GPIO4_A7	135	35	36	131	GPIO4_A3	UART0_TX_M2		
	UART3_RX_M2	I2C5_SCL_M2	GPIO4_A6	134	37	38	132	GPIO4_A4	UART0_RX_M2		
			GND		39	40	133	GPIO4_A5		UART3_TX_M2	

In the OPi OS Arch system, CAN in 40pin is turned off by default and needs to be turned on manually to use it.

Add the configuration in red font below to /boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open CAN1.

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5-pro.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-can1-m1.dtbo

The red font configuration above needs to be written on 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 the CAN device node, 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 corresponding pin of CAN1 is

	CAN1
TX pin	Corresponds to pin 13
RX pin	Corresponds to pin 11

4) For the method of **using CANalyst-II analyzer to test CAN sending and receiving messages,** please refer to the section below using CANalyst-II analyzer to test sending and receiving messages.

5. Linux SDK——orangepi-build usage instructions

5.1. Compilation system requirements

We can cross-compile the Linux image of the development board in an x64 computer, or we can compile the Linux image of the development board in the Ubuntu22.04 system of the development board. Please choose one according to your preference.

If you use orangepi-build to compile the Linux image in the Ubuntu22.04 system of the development board, please dissipate heat (especially when the SSD is started). If heat dissipation is not done properly, file system runaway errors may easily occur.

5.1.1. Compile using the Ubuntu22.04 system of the development board

1) The Linux SDK, **orangepi-build**, supports running on **Ubuntu 22.04** on the development board (other systems have not been tested), so before downloading orangepi-build, please first ensure that the Ubuntu version installed on the development board is Ubuntu 22.04. The command to check the Ubuntu version installed on the development board is as follows. If the Release field does not display **22.04**, it means that the Ubuntu version currently used does not meet the requirements. Please change 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) Since the source code such as kernel and U-boot are stored on GitHub, it is very important to ensure that the development board can download the code from GitHub normally when compiling the image.

5.1.2. Compile using x64 Ubuntu22.04 computer

1) The Linux SDK, **orangepi-build**, supports running on computers with **Ubuntu 22.04** installed. Therefore, before downloading orangepi-build, please first ensure that the

Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on the computer is as follows. If the Release field does not display **22.04**, it means that the Ubuntu version currently used does not meet the requirements. Please change the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID:    Ubuntu
Description:       Ubuntu 22.04 LTS
Release:           22.04
Codename:          jammy
```

2) If the computer is installed with a Windows system and there is no computer with Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in the Windows system. But please note, do not compile orangepi-build on the WSL virtual machine, because orangepi-build has not been tested in the WSL virtual machine, so there is no guarantee that orangepi-build can be used normally in WSL.

3) The installation image download address of Ubuntu 22.04 **amd64** version is:

```
https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04.3-desktop-amd64.iso
或者
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.3-desktop-amd64.iso
```

4) After installing Ubuntu 22.04 on the computer or virtual machine, please first set the software source of Ubuntu 22.04 to Tsinghua source, otherwise it is easy to make errors due to network reasons when installing the software later.

a. For the method of replacing Tsinghua source, please refer to the instructions on this page.

```
https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/
```

b. Note that the Ubuntu version needs to be switched to 22.04

Ubuntu 镜像使用帮助

Ubuntu 的软件源配置文件是 /etc/apt/sources.list。将系统自带的该文件做个备份，将该文件替换为下面内容，即可使用 TUNA 的软件源镜像。

选择你的ubuntu版本: 22.04 LTS

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
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

# 预发布软件源，不建议启用
# 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
```

c. The contents of the **/etc/apt/sources.list** file that need to be replaced are:

```
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 commented by default to improve apt update speed. You can uncomment it yourself if necessary.
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse


# Pre-release software source, not recommended to be enabled
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

d. After the replacement, you need to update the package information and ensure that no errors are reported.

```
test@test:~$ sudo apt update
```

e. In addition, since the source code such as kernel and U-boot are stored on GitHub, it is very important to ensure that the computer can download the code from GitHub normally when compiling the image.

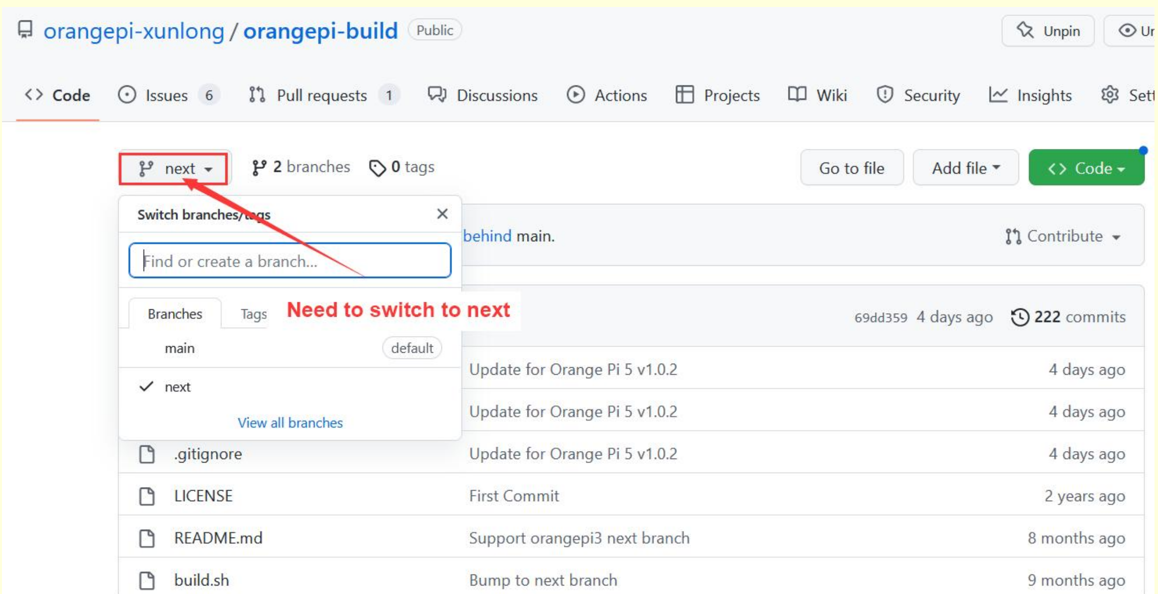
5. 2. Obtain the source code of linux sdk

5. 2. 1. Download orangepi-build from github

1) Linux sdk actually refers to the orangepi-build set of codes. Orangepi-build is modified based on the armbian build compilation system. Multiple versions of Linux images can be compiled using orangepi-build. First download the orangepi-build code, the command is as follows:

```
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 Pro development board needs to download the source code of the next branch of orangepi-build. The above git clone command needs to specify the branch of the orangepi-build source code as next.



Downloading the orangepi-build code through the git clone command does not require entering the user name and password of the github account (the same is true for downloading other codes in this manual). If after entering the git clone command, Ubuntu PC prompts you to enter the user name of the github account. The name and password are usually entered incorrectly in the address of the orangepi-build

warehouse behind git clone. Please carefully check whether there are any spelling errors in the command, rather than thinking that we have forgotten to provide the username and password of the github account.

2) The u-boot and linux kernel versions currently used by the development board are as follows

branch	u-boot version	linux kernel version
legacy	u-boot 2017.09	linux5.10

The branch mentioned here is not the same thing as the branch of orangepi-build source code, please don't get confused. This branch is mainly used to distinguish different kernel source code versions.

We currently define the linux5.10 bsp kernel provided by RK as the legacy branch. If the mainline kernel is supported in the future, a current branch will be added.

3) After orangepi-build is downloaded, it will contain the following files and folders

a. **build.sh**: Compile startup script
b. **external**: Contains configuration files, specific scripts, and source code of some programs needed to compile the image, etc.
c. **LICENSE**: GPL 2 license file
d. README.md: orangepi-build documentation
e. **scripts**: Common script for compiling linux images

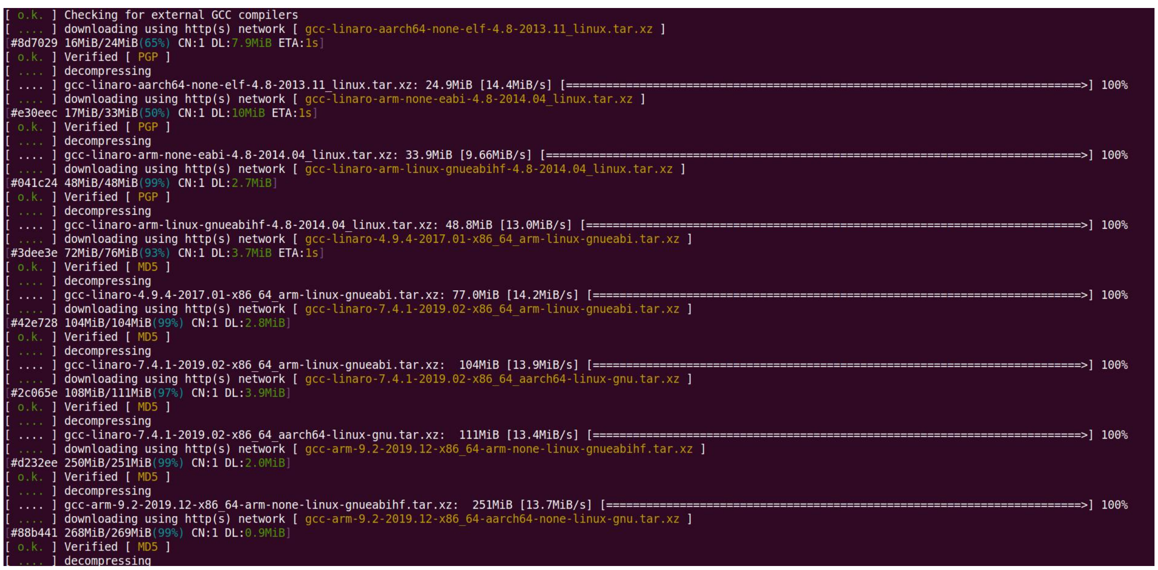
```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

If you downloaded the orangepi-build code from github, after downloading, you may find that orangepi-build does not contain the source code of u-boot and linux kernel, and there is no cross-compilation tool required to compile u-boot and linux kernel. chain, this is normal, because these things are stored in other separate github repositories or some servers (their addresses will be detailed below). Orangepi-build will specify the addresses of u-boot, Linux kernel and cross-compilation tool chain in the script and configuration file. When running orangepi-build, when it finds that these things are not available locally, it will automatically download them from the corresponding places.

5.2.2. Download the cross-compilation tool chain

The cross-compilation tool chain will be downloaded only if you use orangepi-build to compile the image on an x64 computer. When compiling the Linux image of the development board in Ubuntu 22.04, the cross-compilation tool chain will not be downloaded. At this time, orangepi-build/toolchains will be an empty folder.

1) When orangepi-build is run for the first time, it will automatically download the cross-compilation toolchain and put it in the **toolchains** folder. Every time you run orangepi-build's build.sh script, it will check whether the cross-compilation toolchain in **toolchains** exists. , if it does not exist, the download will be restarted, if it exists, it will be used directly, and the download will not be repeated.



2) The mirror URL of the cross-compilation tool chain in China is the open source software mirror site of Tsinghua University

https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/

3) After **toolchains** is downloaded, it will contain multiple versions of cross-compilation toolchains, and 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 to compile the Linux kernel source code is

a. linux5.10

```
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
```

5) The cross-compilation tool chain used to compile u-boot source code is

a. v2017.09

```
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
```

5. 2. 3. orangepi-build complete directory structure description

1) After downloading, the orangepi-build warehouse does not contain the source code of the linux kernel, u-boot and cross-compilation tool chain. The source code of the linux kernel and u-boot is stored in an independent git warehouse.

a. The git warehouse where the Linux 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 b.u-boot source code is stored is as follows:

https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2017.09-rk3588

2) When orangepi-build is run for the first time, it will download the cross-compilation tool chain, u-boot and linux kernel source code. After successfully compiling the linux image, the files and folders that can be seen in orangepi-build are:

a. **build.sh**: Compile startup script

b. **external**: Contains the configuration files needed to compile the image, scripts for specific functions, and the source code of some programs. The rootfs compressed package cached during the image compilation process 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. Please do not modify the name of the kernel source code folder manually. If it is modified, When the compilation system is running, the kernel source code will be re-downloaded.

d. **LICENSE**: GPL 2 license file
e. **README.md**: orangepi-build documentation
f. **output**: Store compiled u-boot, linux and other deb packages, compilation logs, and compiled images and other files
g. **scripts**: Common script for compiling linux images
h. **toolchains**: Store 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 branch 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 files needed to compile the script

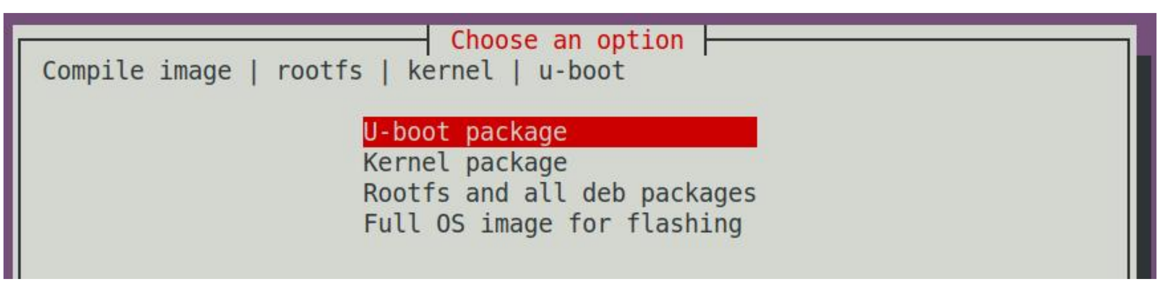
```
test@test:~/orangepi-build$ ls
build.sh external kernel LICENSE output README.md scripts toolchains
u-boot userpatches
```

5. 3. Compile u-boot

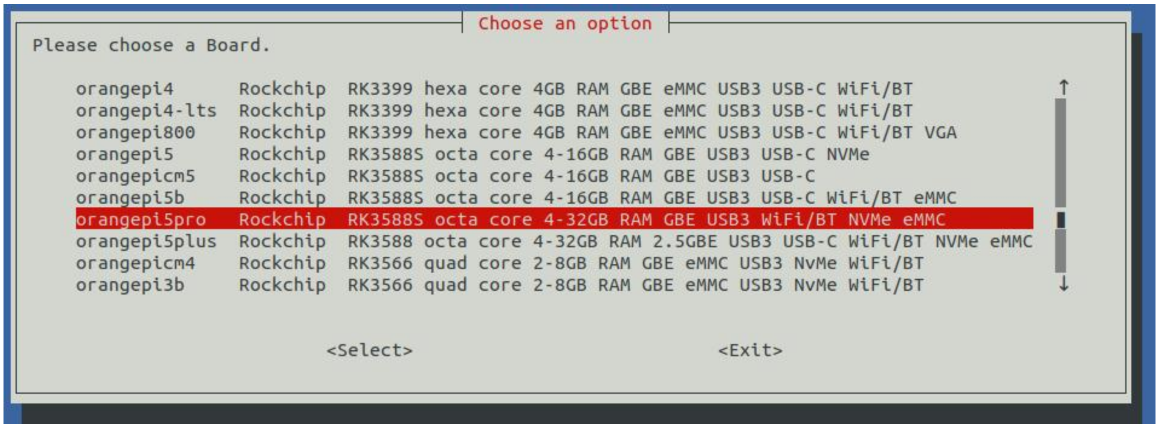
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package** and press Enter



3) Then select the model of the development board



4) Then it will start to compile u-boot. Some of the information prompted during compilation is as follows:

a. u-boot source code version

```
[ o.k. ] Compiling u-boot [ v2017.09 ]
```

b. Version of the cross-compilation tool chain

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 7.4.1 ]
```

c. The path to the u-boot deb package generated by compilation

```
[ o.k. ] Target directory [ orangepi-build/output/debs/u-boot ]
```

d. The package name of the u-boot deb package generated by compilation

```
[ o.k. ] File name [ linux-u-boot-legacy-orangepi5pro_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 without selecting through the graphical interface. You can start compiling u-boot directly.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi5pro
BRANCH=legacy BUILD_OPT=u-boot KERNEL_CONFIGURE=no ]
```

5) View the compiled u-boot deb package

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-legacy-orangepi5pro_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-orangepi5pro_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-orangepi5pro_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-orangepi5pro_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. Therefore, if you want to modify the u-boot source code, you first need to turn off the download and update function of the source code. (**You need to completely compile u-boot once before you can turn off this function, otherwise it will prompt that the source code of u-boot cannot be found. If the source code compressed package is downloaded from Baidu Cloud Disk, there is no such problem, because the source code of u-boot have been cached**), otherwise the modifications 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 deb package of u-boot 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-orangepi5pro_1.0.2_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the installed deb package of u-boot

```
root@orangepi:~# apt purge -y linux-u-boot-orangepi5pro-legacy
```

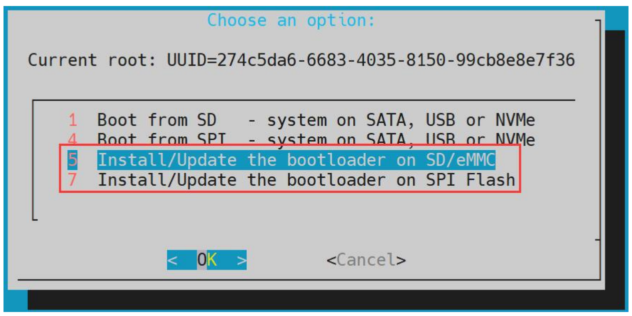
c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi5pro_1.0.2_arm64.deb
```

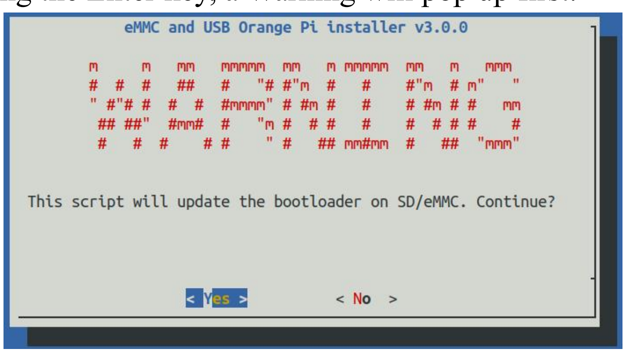
d. Then run the nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

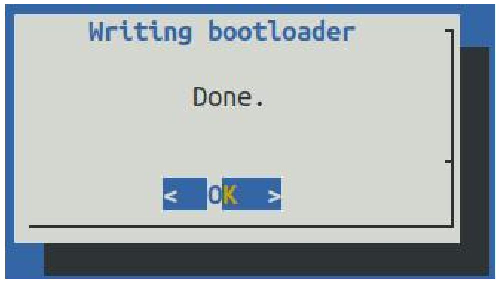
e. Then select **5 Install/Update the bootloader on SD/eMM** to update u-boot in the TF card or **7 Install/Update the bootloader on SPI Flash** to update u-boot in SPI Flash



f. After pressing the Enter key, a Warning will pop up first.



g. Press the Enter key again to start updating u-boot. After the update is completed, the following information will be displayed.



h. Then you can restart the development board to test whether the u-boot modification has taken effect.

9) Other useful information

a. In 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_pro_defconfig

b. In u-boot 2017.09 source code, the dts file used by the development board is

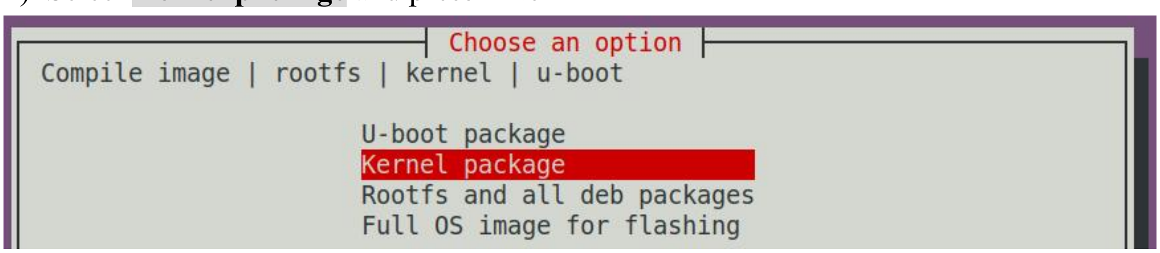
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3588s-orangepi-5-pro.dts

5. 4. Compile the linux kernel

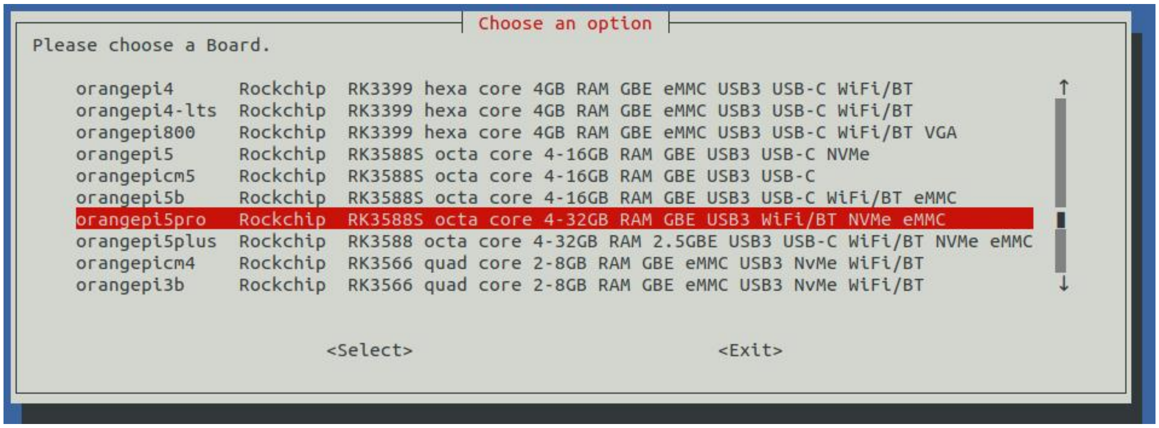
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

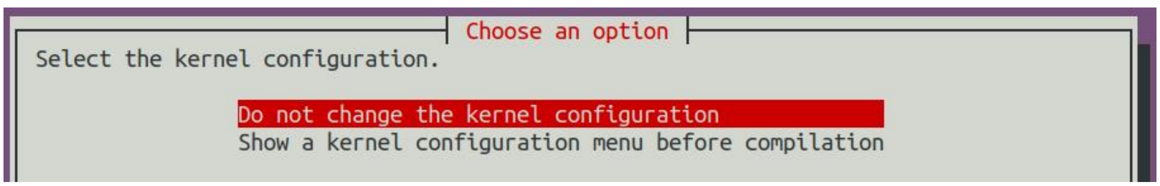
2) Select **Kernel package** and press Enter



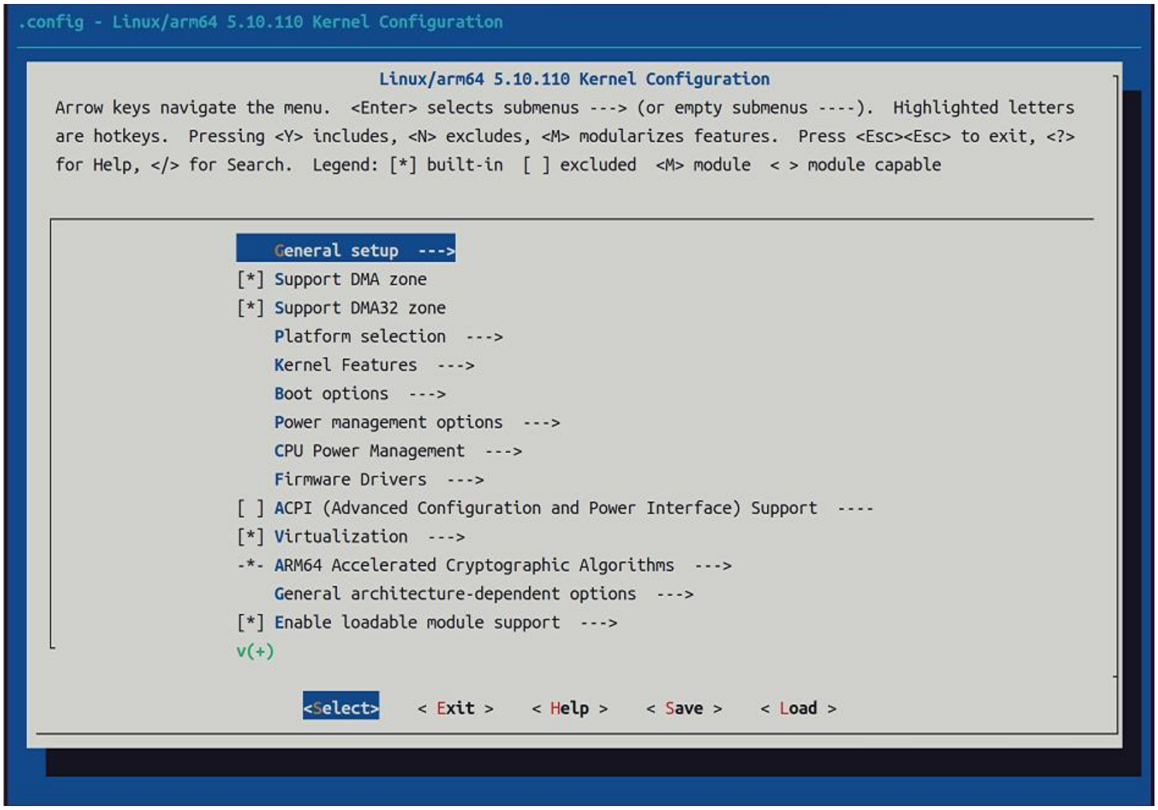
3) Then select the model of the development board



4) Then you will be prompted whether you need to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If you need to modify the kernel configuration, select the second one.



5) If you choose to display the kernel configuration menu (the second option) in step 4), the kernel configuration interface opened through **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration. After modification, save and exit. Yes, after exiting, the kernel source code will start to be compiled.



a. If you do not need to modify the kernel configuration options, when running the build.sh script, pass **KERNEL_CONFIGURE=no** to temporarily block the pop-up of the 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 to permanently disable this function.

c. If the following error is prompted when compiling the kernel, this is because the Ubuntu PC terminal interface is too small, causing the **make menuconfig** interface to be unable to be displayed. Please adjust the Ubuntu PC terminal to the maximum size, and then rerun the build.sh script.

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

6) Part of the information prompted when compiling the kernel source code is explained as follows:

a. Linux kernel source code version

[o.k.] Compiling current kernel [**5.10.160**]

b. The version of the cross-compilation tool chain 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 kernel-related deb package generated by compilation

[o.k.] Target directory [**orangepi-build/output/debs/**]

e. The package name of the kernel image deb package generated by compilation

[o.k.] File name [**linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb**]

f. Compilation time

[o.k.] Runtime [**5 min**]

g. Finally, the compilation command to repeatedly compile the last selected kernel will be displayed. Use the following command without selecting through the graphical interface, and you can directly start compiling the kernel source code.

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepi5pro BRANCH=legacy BUILD_OPT=kernel KERNEL_CONFIGURE=no**]

7) View the kernel-related deb package generated by compilation

a. **linux-dtb-legacy-rockchip-rk3588_1.0.2_arm64.deb** Contains dtb files used by the kernel

b. **linux-headers-legacy-rockchip-rk3588_1.0.2_arm64.deb** Contains kernel header files

c. **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
```

8) The files contained in the generated deb package of linux-image are as follows

a. 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
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.160-rockchip-rk3588
│   ├── System.map-5.10.160-rockchip-rk3588
│   └── vmlinuz-5.10.160-rockchip-rk3588
├── etc
│   └── kernel
├── lib
│   └── modules
├── linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb
└── usr
    ├── lib
    └── share
```

9) When the orangepi-bulid compilation system compiles the linux kernel source code, it

will first synchronize the linux kernel source code with the linux kernel source code of the github server. Therefore, if you want to modify the linux kernel source code, you first need to turn off the update function of the source code (**it needs to be completely compiled once This function can only be turned off after the linux kernel source code is obtained, otherwise it will prompt that the linux kernel source code cannot be found. If the source code compressed package is downloaded from Baidu Cloud Disk, there will be no such problem, because the linux source code has been cached**), otherwise the All modifications will be restored, as follows:

Set the IGNORE_UPDATES variable in **userpatches/config-default.conf** to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If the kernel is modified, you can use the following method to update the kernel and kernel module of the development board Linux system

a. Upload the compiled deb package of the 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
```

b. Then log in to the development board and uninstall the installed deb package of the Linux kernel.

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk3588
```

c. 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
```

d. Then restart the development board and check whether the kernel-related modifications have taken effect.

```
root@orangepi:~# reboot
```

10) 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.

```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-legacy-opi5pro.config
```

b. The location of the dts file used by the development board is

```
orangepi-build/kernel/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s
```

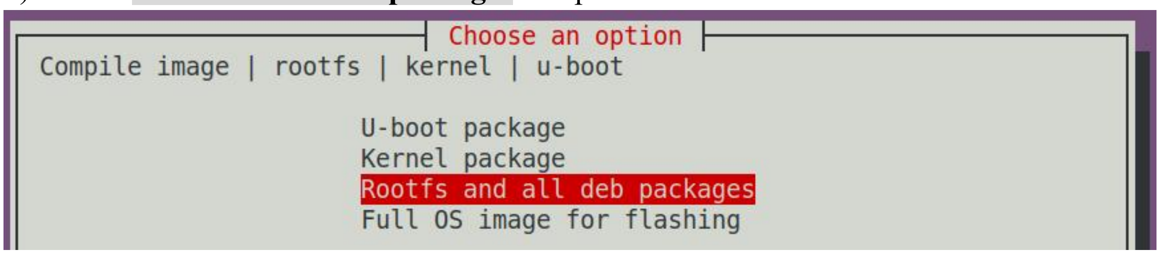
```
-orangepi-5-pro.dts
```

5. 5. Compile rootfs

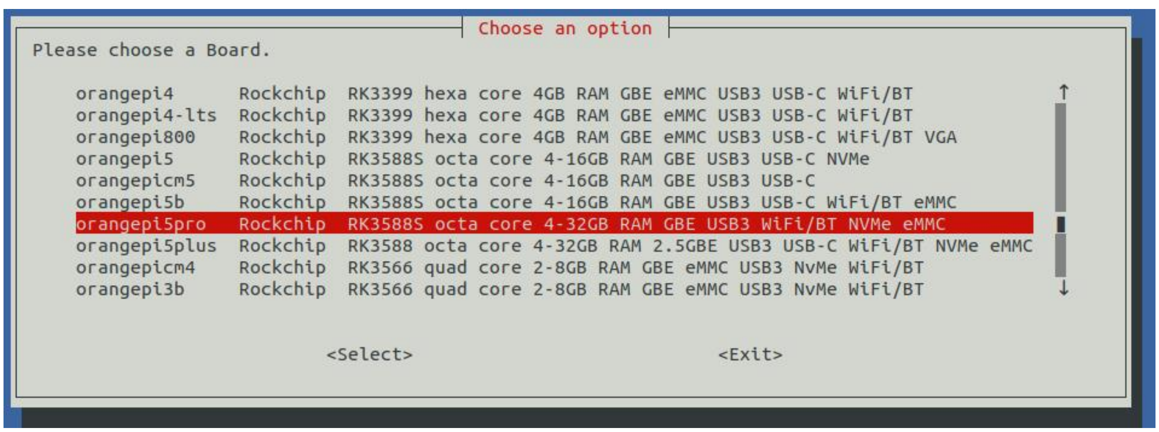
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

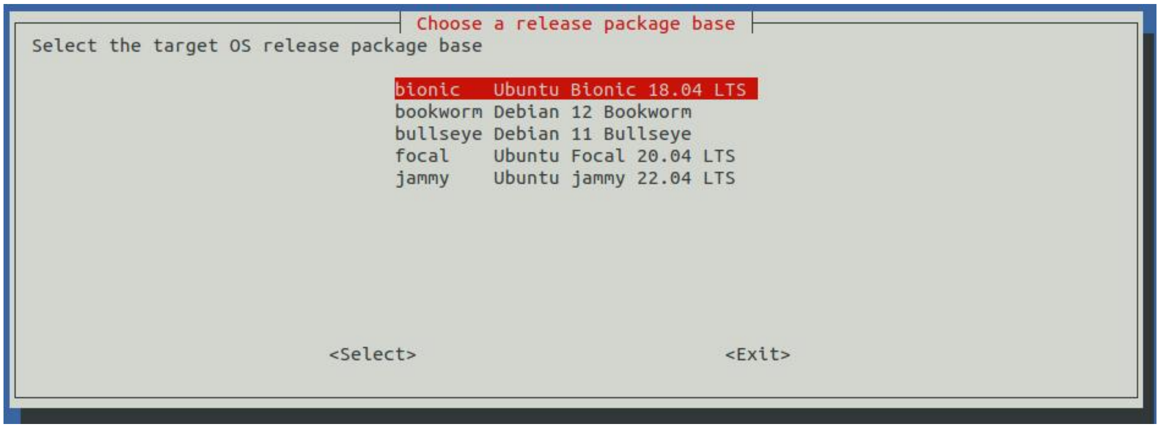
2) Select **Rootfs and all deb packages** and press Enter



3) Then select the model of the development board



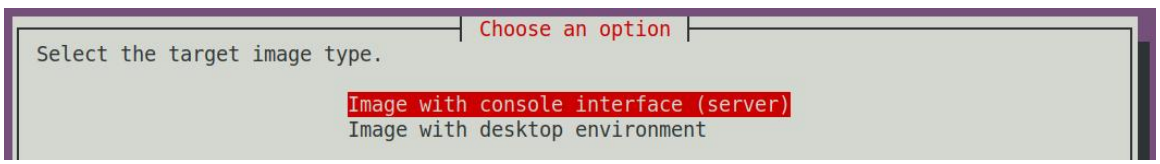
4) Then select the type of rootfs



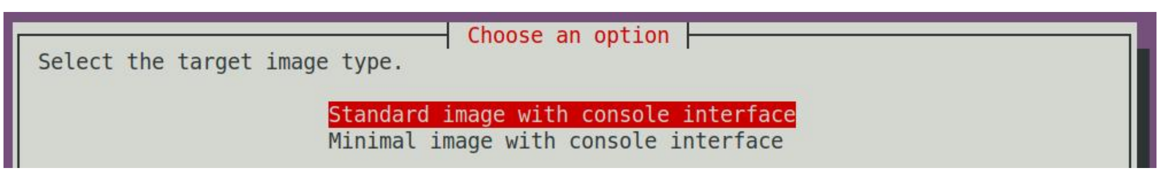
5) Then select the type of image

a. **Image with console interface (server)** Represents the image of the server version, which is relatively small in size.

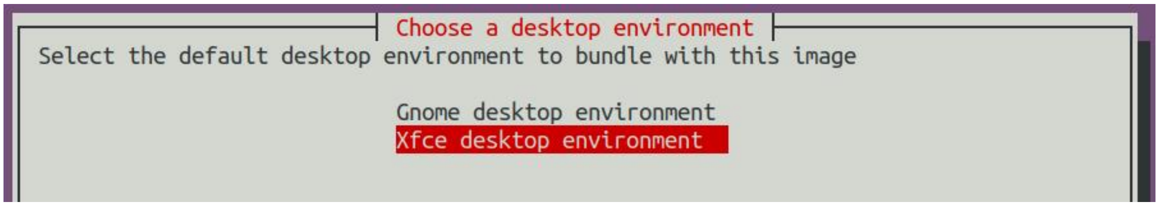
b. **Image with desktop environment** Represents an image with a desktop, which is relatively large in size.

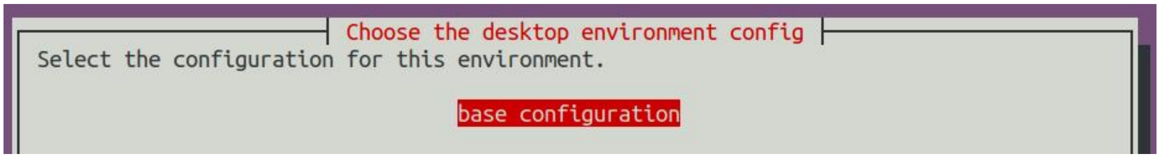


6) If you are compiling the server version of the image, you can also choose to compile the Standard version or the Minimal version. The Minimal version will have much less pre-installed software than the Standard version (**please do not choose the Minimal version without special needs, because many things are not pre-installed by default. Some functions may not be available**)

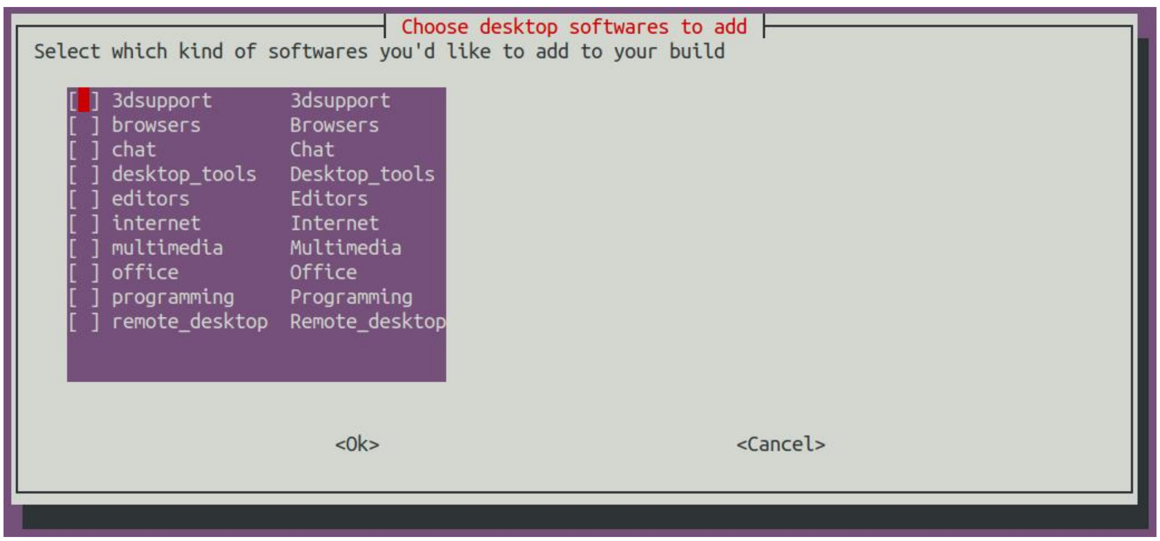


7) If you are compiling a desktop version of the image, you need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop





You can then select additional packages that need to be installed. Please press the Enter key here to skip directly.



8) Then the compilation of rootfs will begin. Some of the information prompted during compilation is as follows:

a. Type of rootfs

[o.k.] local not found [Creating new rootfs cache for **jammy**]

b. The storage path of the rootfs compressed package generated by compilation

[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. Compilation time

[o.k.] Runtime [**13 min**]

9) View the compiled rootfs compressed package
   a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is a compressed package of rootfs. The meaning of each field in the name is:
      a) **jammy** represents the type of Linux distribution of rootfs
      b) **xfce** indicates that rootfs is the desktop version, and if it is **cli**, it indicates the server version type.
      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. 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/
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.current
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list
```

10) If the required rootfs already exists under **external/cache/rootfs**, then compiling the rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to check whether it already exists. There is a cached rootfs available. If it is available, use it directly. This can save a lot of download and compilation time.

5. 6. Compile linux image

1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Full OS image for flashing** and 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.

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
orangepicm5    Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C
orangepi5b     Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT eMMC
orangepi5pro   Rockchip  RK3588S octa core 4-32GB RAM GBE USB3 WiFi/BT NVMe eMMC
orangepi5plus  Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C WiFi/BT NVMe eMMC
orangepicm4    Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
orangepi3b     Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT

<Select>                    <Exit>
```

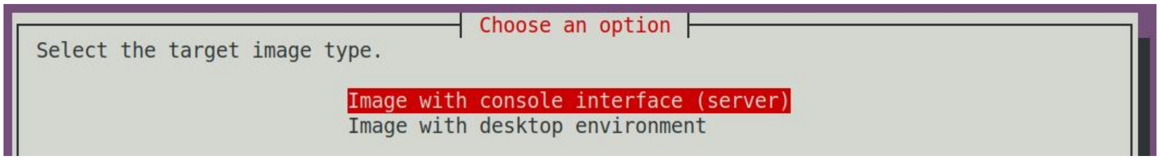
4) Then select the type of rootfs

```
Choose a release package base
Select the target OS release package base

bionic    Ubuntu Bionic 18.04 LTS
bookworm  Debian 12 Bookworm
bullseye  Debian 11 Bullseye
focal     Ubuntu Focal 20.04 LTS
jammy     Ubuntu jammy 22.04 LTS

<Select>                    <Exit>
```

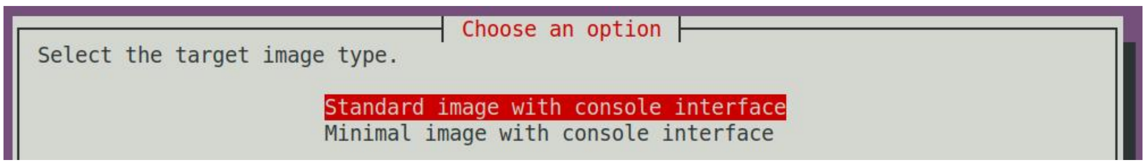
5) Then select the type of image

a. **Image with console interface (server)** Represents the image of the server version, which is relatively small in size.

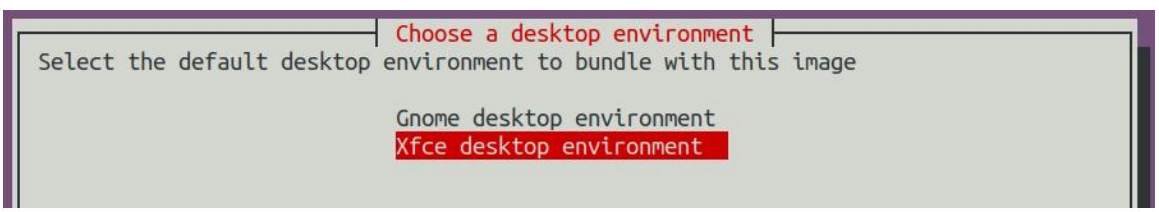
b. **Image with desktop environment** Represents an image with a desktop, which is relatively large in size.

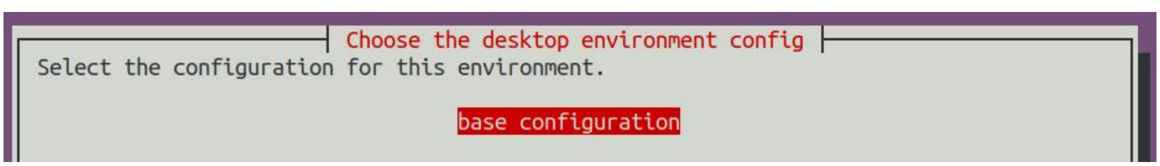


6) If you are compiling the server version of the image, you can also choose to compile the Standard version or the Minimal version. The Minimal version will have much less pre-installed software than the Standard version (**please do not choose the Minimal version without special needs, because many things are not pre-installed by default. Some functions may not be available**)

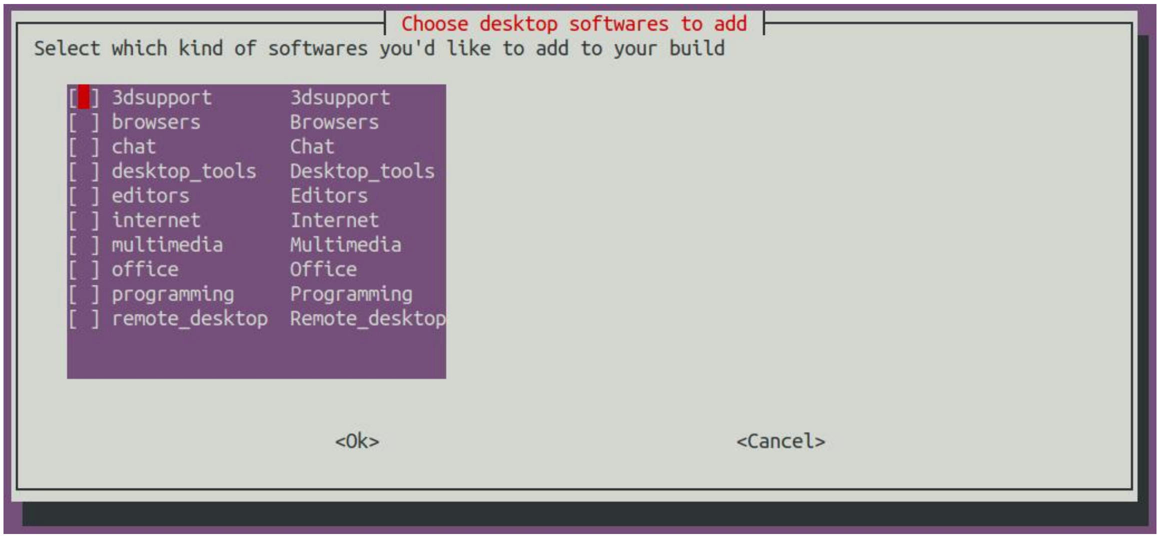


7) If you are compiling a desktop version of the image, you need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop





You can then select additional packages that need to be installed. Please press the Enter key here to skip directly.



8) Then the compilation of the linux image will begin. 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 already cached, only update the code)

c. Compile u-boot source code and generate u-boot deb package

d. Compile linux source code and generate linux-related deb packages

e. Make the deb package of linux firmware

f. Make the deb package of orangepi-config tool

g. Create a deb package with board-level support

h. If you compile the desktop version image, you will also create a desktop-related deb package.

i. Check whether rootfs has been cached. If not, re-create rootfs. If it has been cached, decompress it directly and use it.

j. Install the deb package generated previously 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 configurations, etc.

l. Then create the image file and format the partition. The default type is ext4.

m. Then copy the configured rootfs to the mirror partition.

n. Then update initramfs

o. Finally, write the bin file of u-boot into the image through the dd command.

9) After compiling the image, the following message will be prompted

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/Orangepi5pro_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160/
Orangepi5pro_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. Use the following command to start compiling the image directly without selecting it through the graphical interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh    BOARD=orangepi5pro
BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no
BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

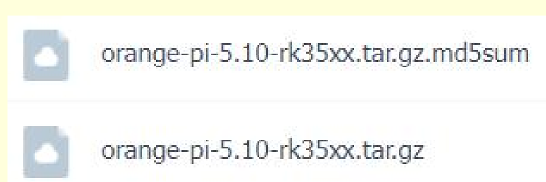
6. Linux development manual

6. 1. Method to compile 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

```
orangepi@orangepi:~$ git clone --depth=1 -b orange-pi-5.10-rk35xx https://github.com/orangepi-xunlong/linux-orangepi
```

If there is a problem downloading the code from github, you can download the kernel source code compressed package from the official tool of the development board, then upload it to the Linux system of the development board, and then unzip it.



The command to decompress the kernel source code archive is:

orangepi@orangepi:~$ **tar zxf orange-pi-5.10-rk35xx.tar.gz**

orangepi@orangepi:~$ **mv orange-pi-5.10-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
```

The path of rockchip_linux_defconfig 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 is an additional kernel module folder under /lib/modules/:

orangepi@orangepi5pro:~$ **ls /lib/modules**

5.10.160+   5.10.160-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 is an additional kernel file under /boot/:

orangepi@orangepi5pro:~/orange-pi-5.10-rk3588$ **ls /boot/vmlinuz***

/boot/vmlinuz-5.10.160+ /boot/vmlinuz-5.10.160-rockchip-rk3588

What is actually loaded when the system starts is the file /boot/Image. 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
5.10.160+
```

7. Instructions for use of Android 12 system

7. 1. Supported Android versions

Android version	Kernel version
Android 12	Linux5.10
Android 12 Box	Linux5.10

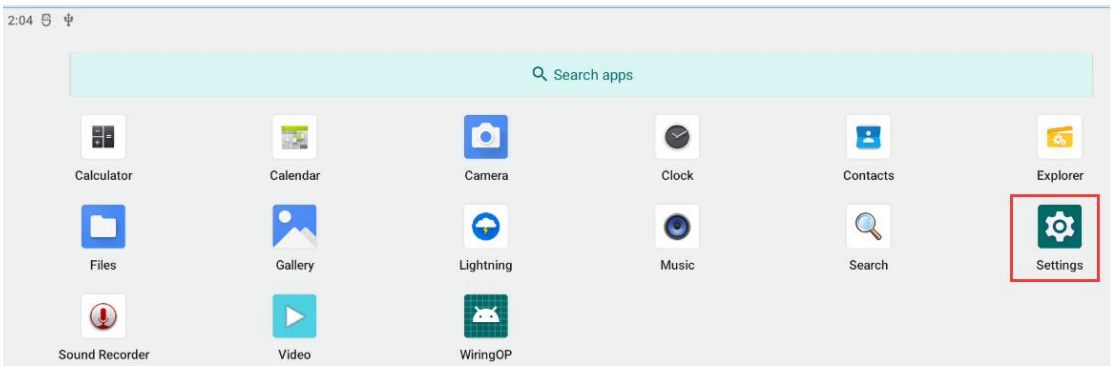
7. 2. Android function adaptation status

Function	Android 12	Android12 Box

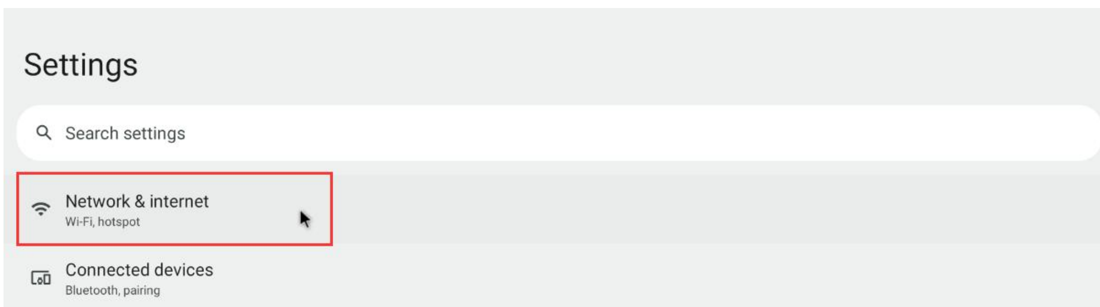
HDMI 2.1 Video	**OK**	**OK**
HDMI 2.1 Audio	**OK**	**OK**
HDMI 2.0 Video	**OK**	**OK**
HDMI 2.0 Audio	**OK**	**OK**
USB2.0x3	**OK**	**OK**
USB3.0x1	**OK**	**OK**
Gigabit network port	**OK**	**OK**
Network port status light	**OK**	**OK**
WIFI	**OK**	**OK**
Bluetooth	**OK**	**OK**
Debug serial port	**OK**	**OK**
RTC chip	**OK**	**OK**
FAN fan interface	**OK**	**OK**
eMMC expansion interface	**OK**	**OK**
GPIO（40pin）	**OK**	**OK**
UART（40pin）	**OK**	**OK**
SPI（40pin）	**OK**	**OK**
I2C（40pin）	**OK**	**OK**
PWM（40pin）	**OK**	**OK**
TF card boot	**OK**	**OK**
OV13850 camera	**OK**	**OK**
OV13855 camera	**OK**	**OK**
SPI+NVME boot	**OK**	**OK**
SPI+SATA boot	**OK**	**OK**
LCD	**OK**	**NO**
MIC	**OK**	**OK**
Headphone playback	**OK**	**OK**
Headphone recording	**OK**	**OK**
Tri-color LED light	**OK**	**OK**
GPU	**OK**	**OK**
NPU	**OK**	**OK**
VPU	**OK**	**OK**
On/off button	**OK**	**OK**
HDMI CEC function	**NO**	**OK**

7. 3. WIFI connection test method

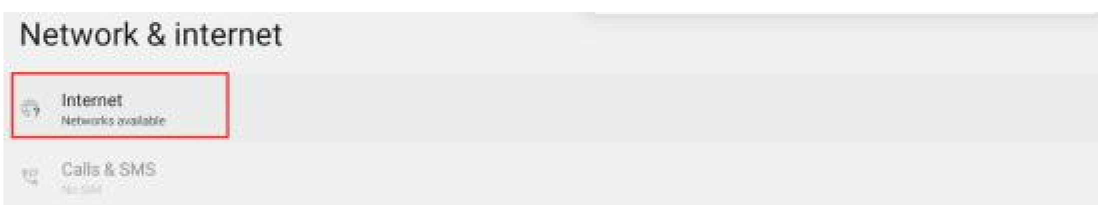
1) First click to enter **Setting**



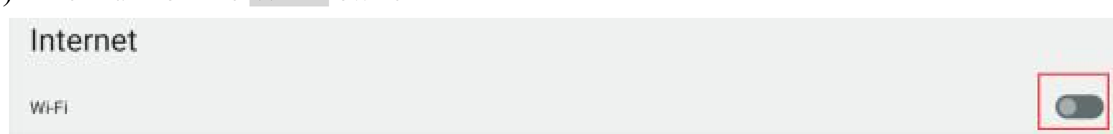
2) Then select **Network & internet**



3) Then select **Internet**

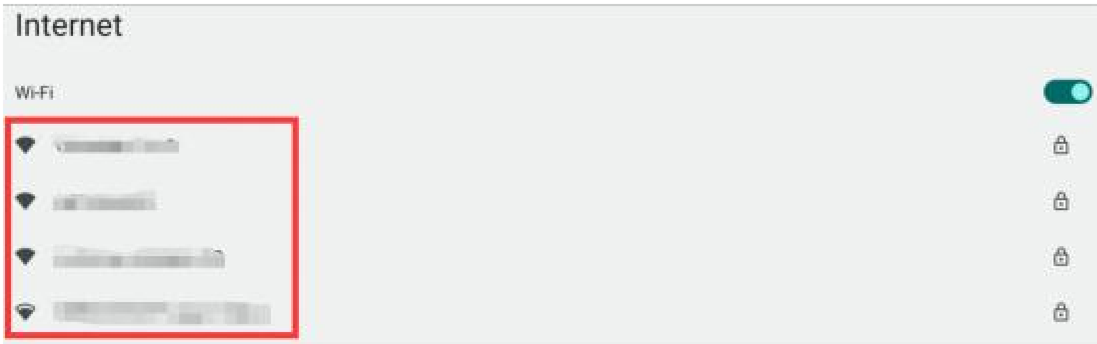


4) Then turn on the **Wi-Fi** switch

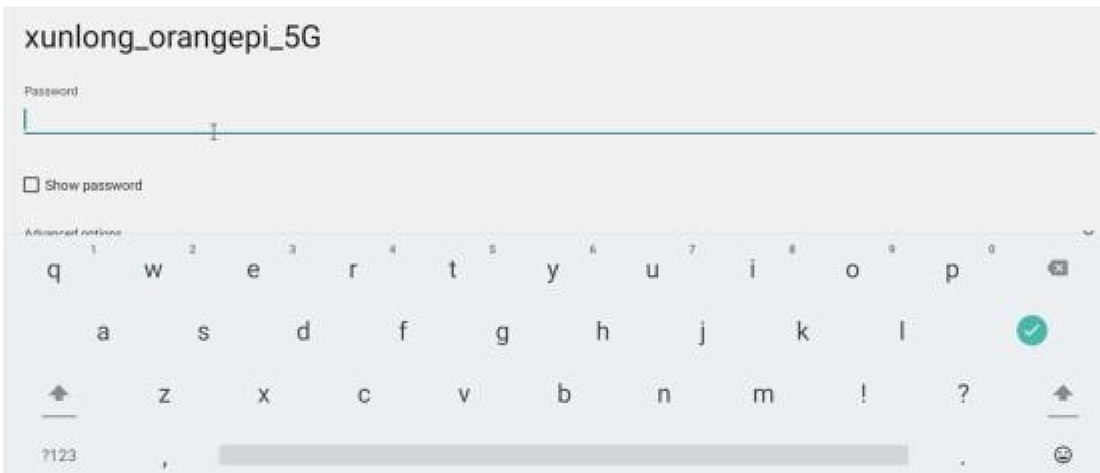


5) If everything is normal after turning on **Wi-Fi**, you can scan for nearby Wi-Fi

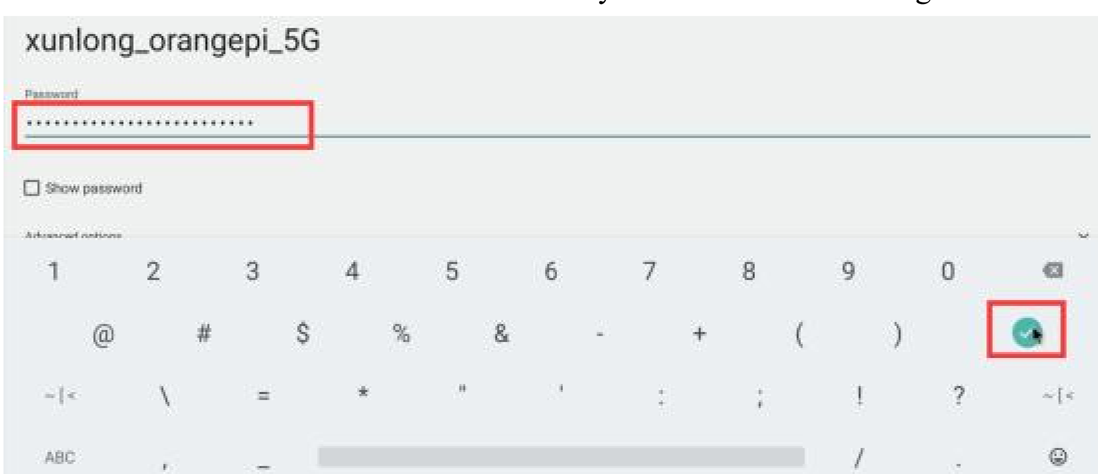
hotspots.



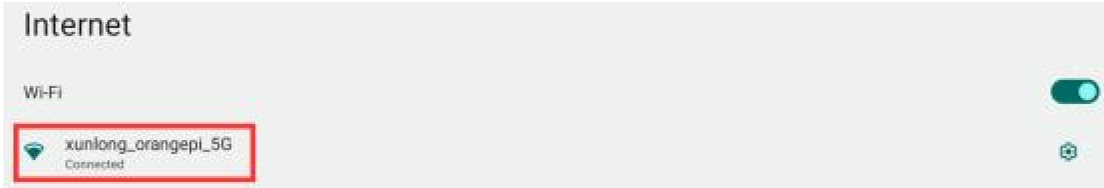
6) Then select the Wi-Fi you want to connect to, and the password input interface shown in the picture below will pop up.



7) Then use the keyboard to enter the password corresponding to Wi-Fi, and then use the mouse to click the Enter button on the virtual keyboard to start connecting to Wi-Fi.



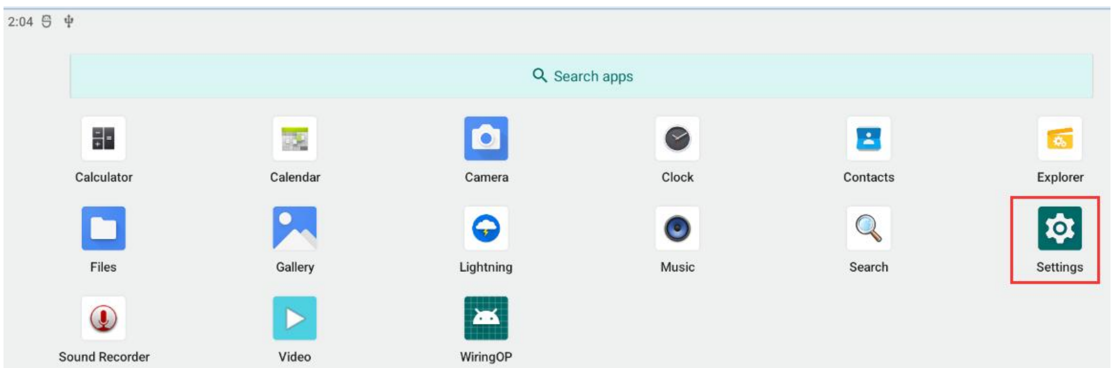
8) The display after successful Wi-Fi connection is as shown below:



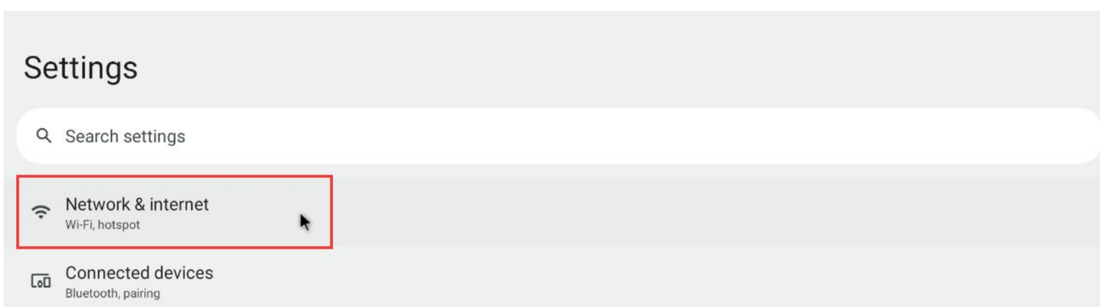
7. 4. How to use Wi-Fi hotspot

1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally.

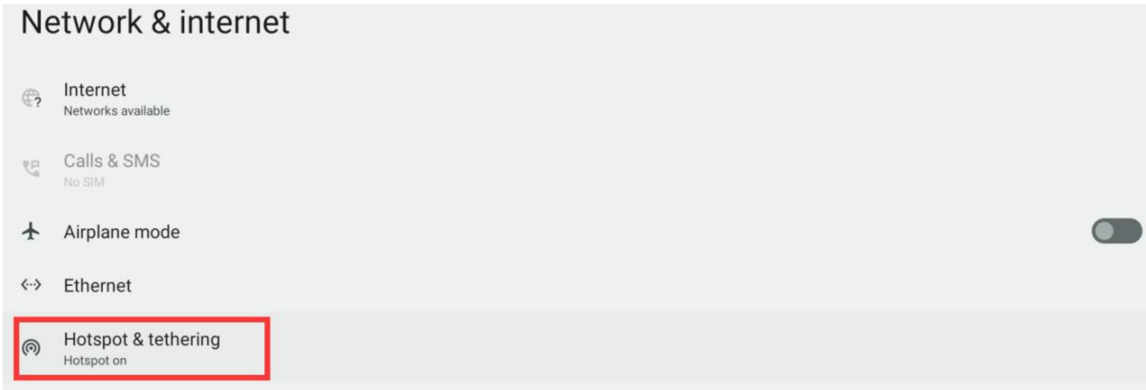
2) Then select **Settings**



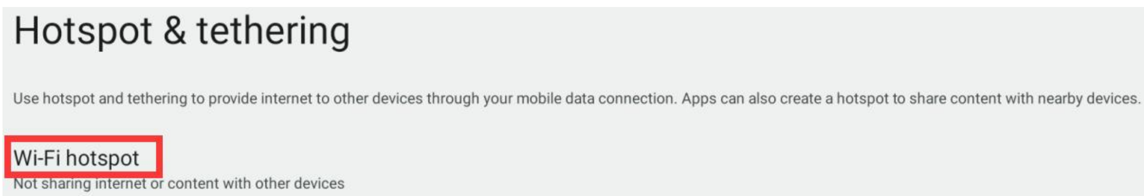
3) Then select **Network & internet**



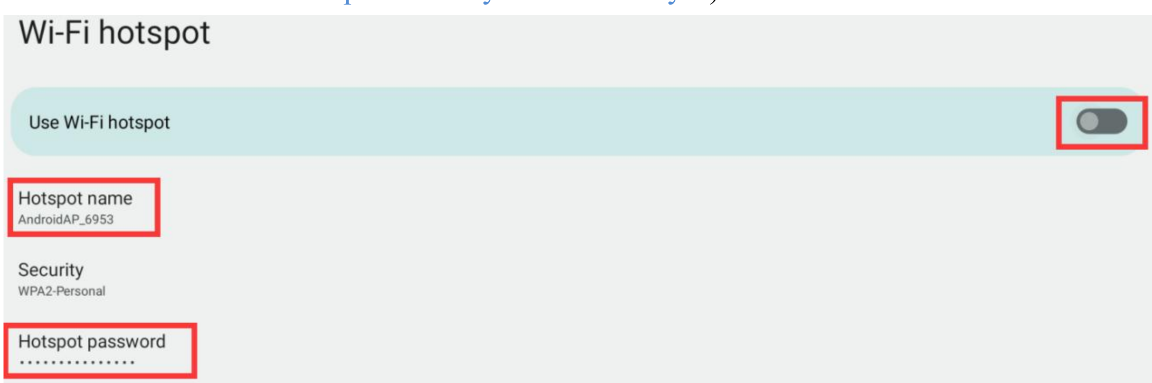
4) Then select **Hotspot & tethering**



5) Then select **Wi-Fi hotspot**



6) Then open the **Wi-Fi hotspot**. You can also see the name and password of the generated hotspot in the picture below. Remember them and use them when connecting to the hotspot (if you need to change the name and password of the hotspot, you need to turn off the Wi-Fi first. -Fi hotspot before you can modify it)

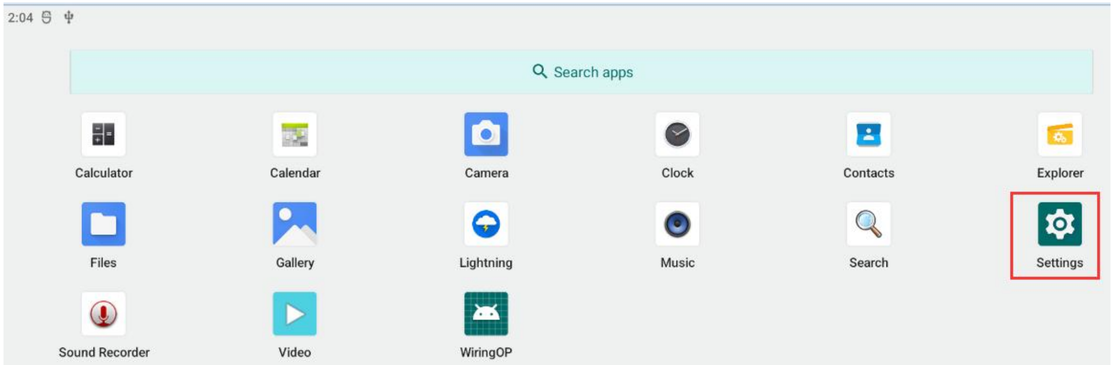


7) You can take out your mobile phone at this time. If everything is normal, you can find the WIFI hotspot with the same name (**here AndroidAP_6953**) shown under the **Hotspot name** in the picture above in the WI-FI list searched by the mobile phone. Then you can click **AndroidAP_6953** to connect to the hotspot. The password can be seen under the **Hotspot password** in the picture above.

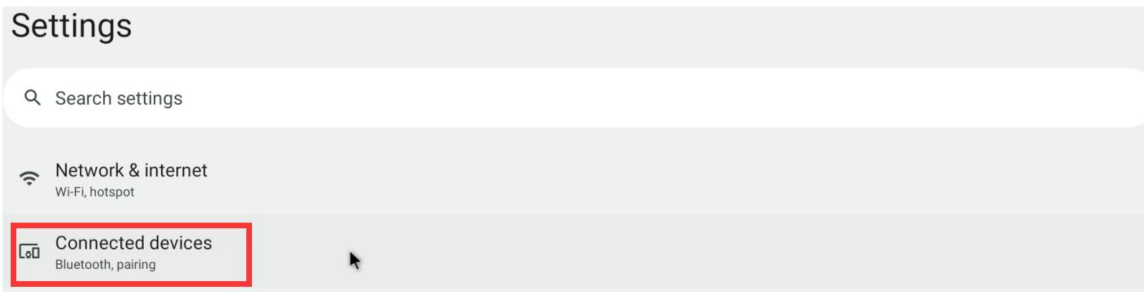
8) After the connection is successful, it will be displayed as shown below (the interface will be different on different mobile phones, the specific interface is subject to the one displayed on your mobile phone). At this point, you can open a web page on your mobile phone to see if you can access the Internet. If the web page can be opened normally, it means that the **WI-FI Hotspot** of the development board can be used normally.

7. 5. Bluetooth test method

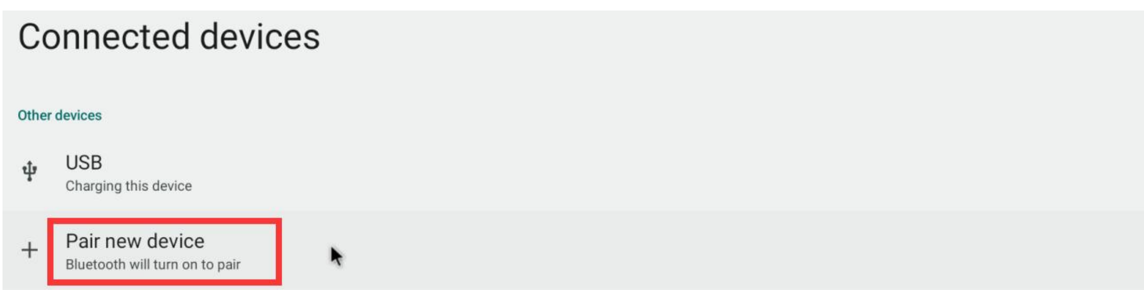
1) First click to enter **Setting**



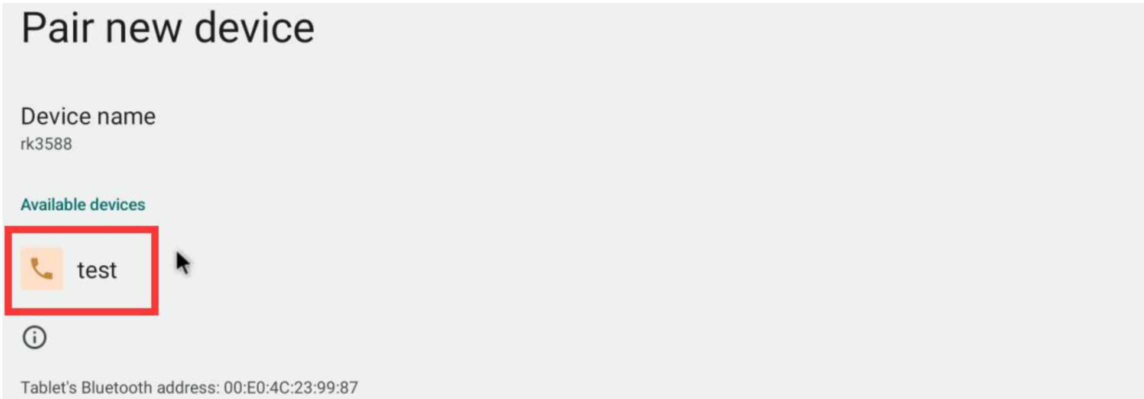
2) Then select **Connected devices**



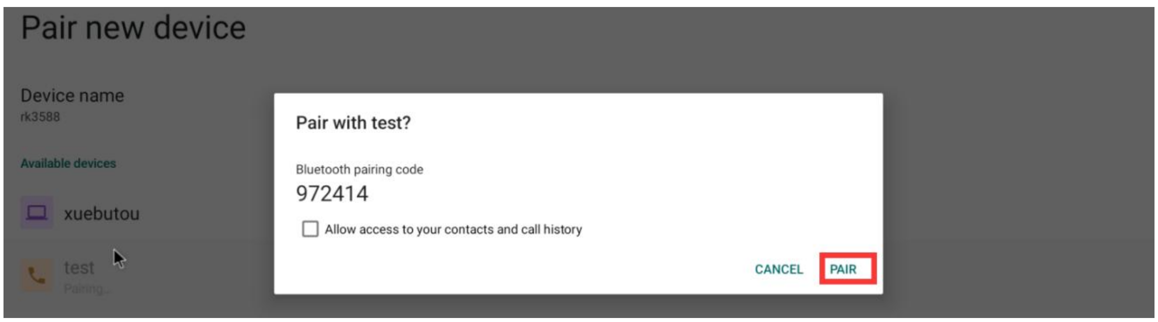
3) Then click **Pair new device** to turn on Bluetooth and start scanning for surrounding Bluetooth devices



4) The searched Bluetooth devices will be displayed under **Available devices**

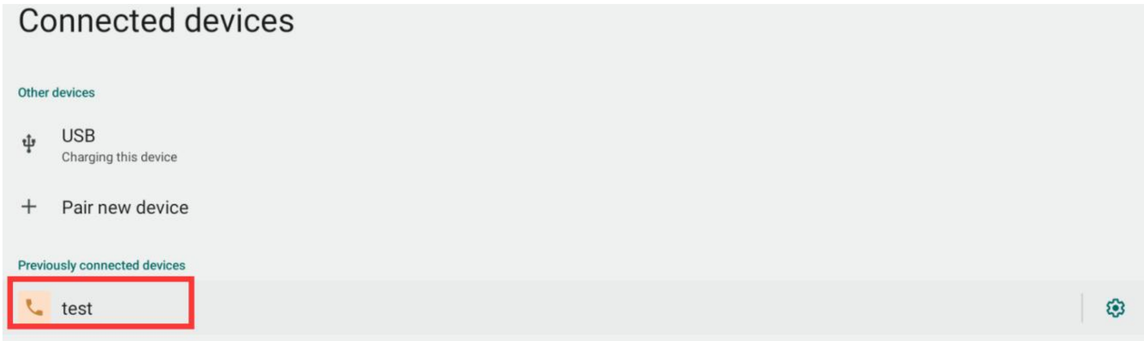


5) Then click on the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option

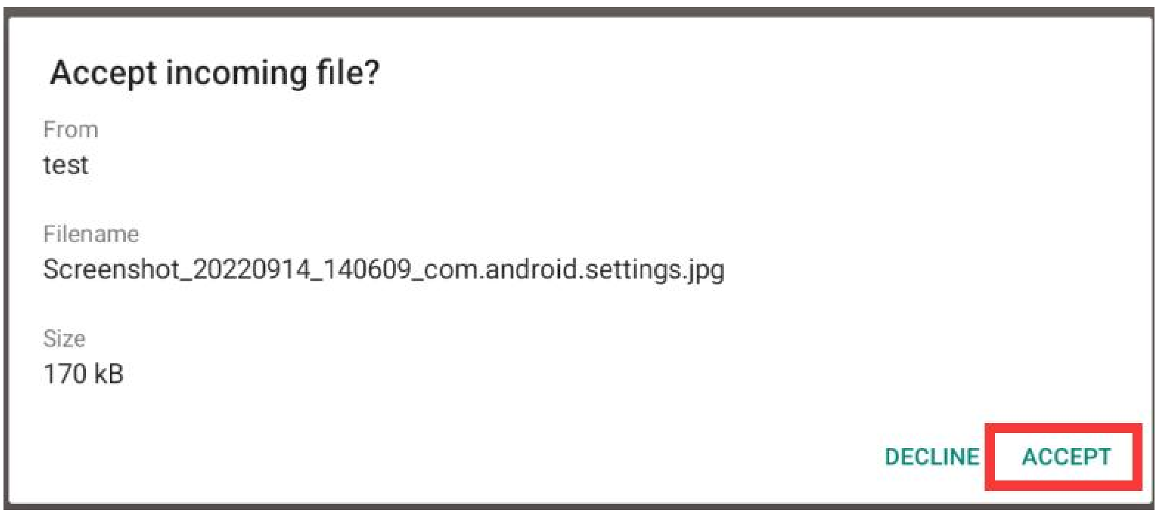


6) What is tested here is the Bluetooth configuration process between the development board and the Android phone. At this time, the following confirmation interface will pop up on the phone. Click the pairing button on the phone to start the pairing process.

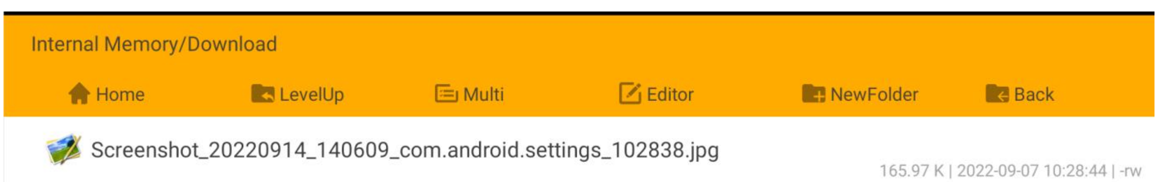
7) After pairing is completed, you can see the paired Bluetooth device as shown in the picture below



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 pictures sent by the mobile phone.



9) The pictures received by the Bluetooth system of the development board Android system can be viewed by opening the **Download** directory in the file manager.

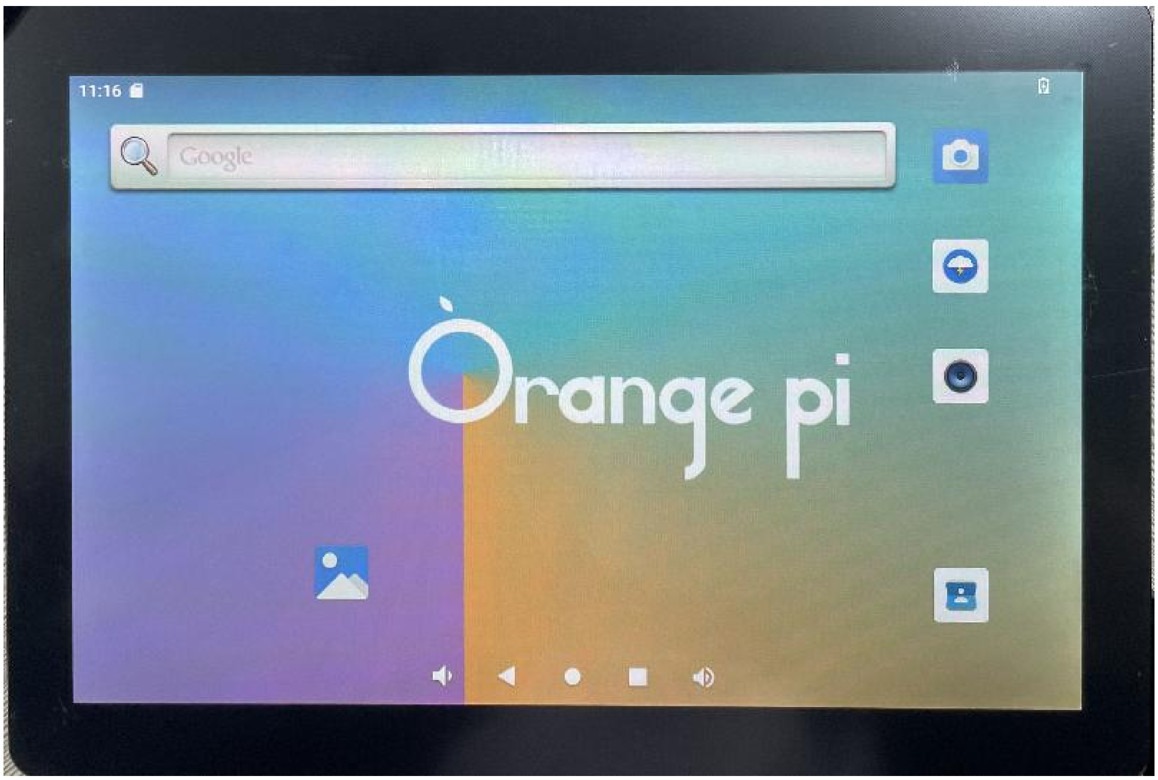


7. 6. How to use the 10.1-inch MIPI screen

1) First, you need to assemble the screen. Please refer to the assembly method of the 10.1-inch MIPI screen.

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

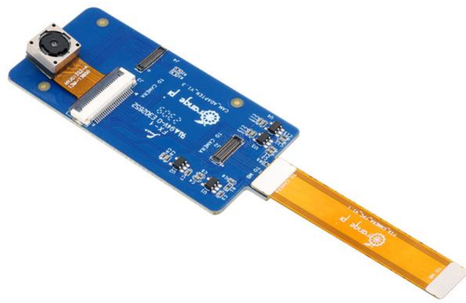
3) Connect the assembled screen to the LCD interface of the development board, connect the Type-C power supply to the board, and power it on. After the system starts, you can see the screen display as shown below.



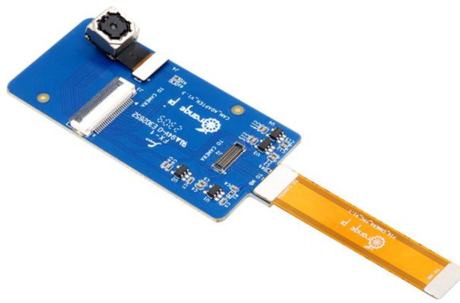
7.7. Testing methods for OV13850 and OV13855 MIPI cameras

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



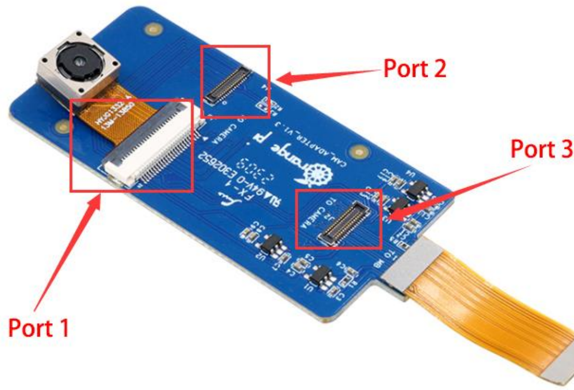
The adapter board and FPC cable used by the OV13850 and OV13855 cameras are the same, but the locations where the two cameras are connected to the adapter board are different. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked **TO MB** needs to be plugged into the camera interface of the development board, and the end marked **TO CAMERA** needs to be plugged into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board, and only one can be connected and used at the same time, as shown in the figure below, among which:

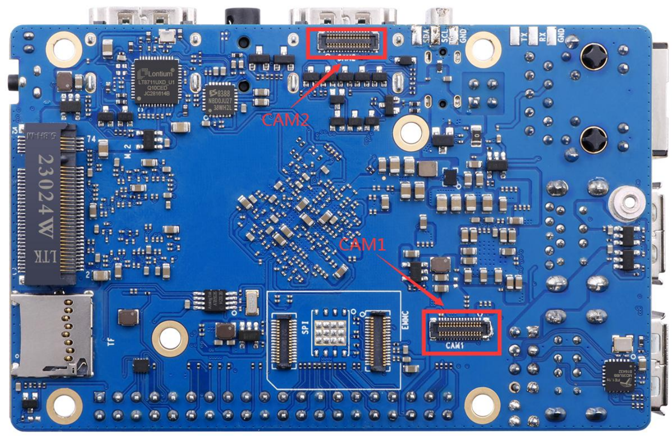
a. **Connect the OV13850 camera to interface 1**

b. **Connect the OV13855 camera to interface 2**

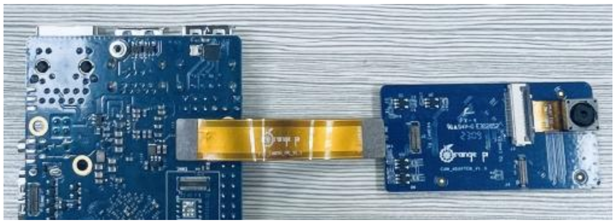
c. Interface No. 3 is not used, just ignore it.



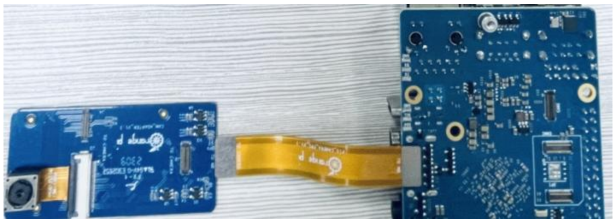
There are a total of 2 camera interfaces on the Orange Pi 5 Pro development board. We define the positions of Cam1 and Cam2 as shown below:



The method of plugging the camera into the Cam1 interface of the development board is as follows:

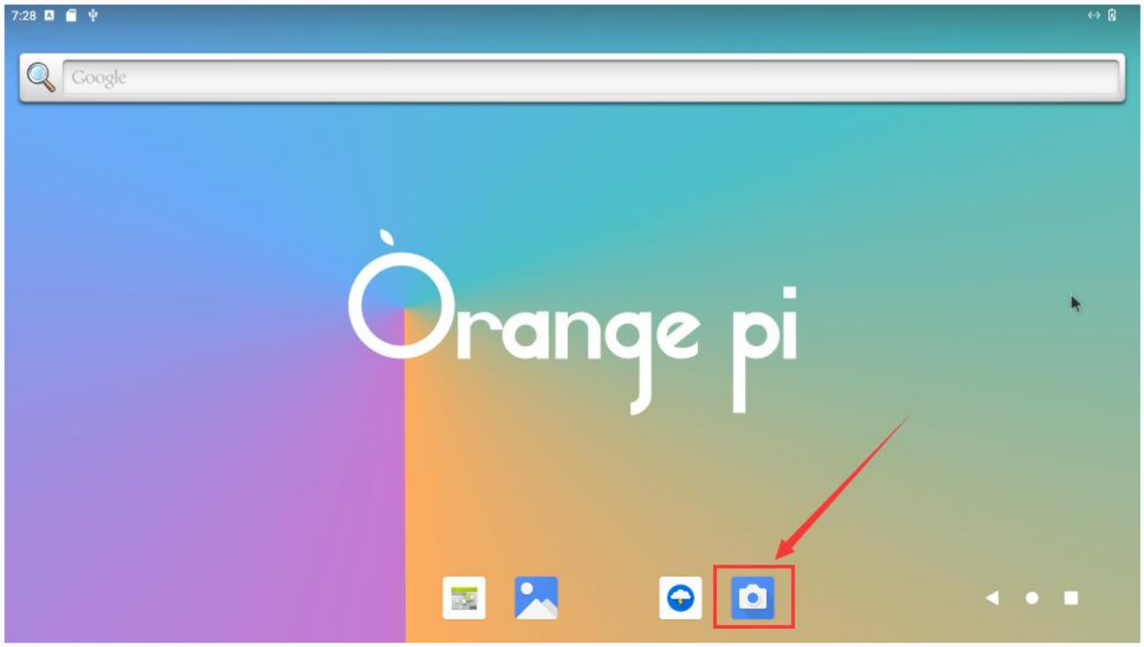


The method of plugging the camera into the Cam2 interface of the development board is as follows:

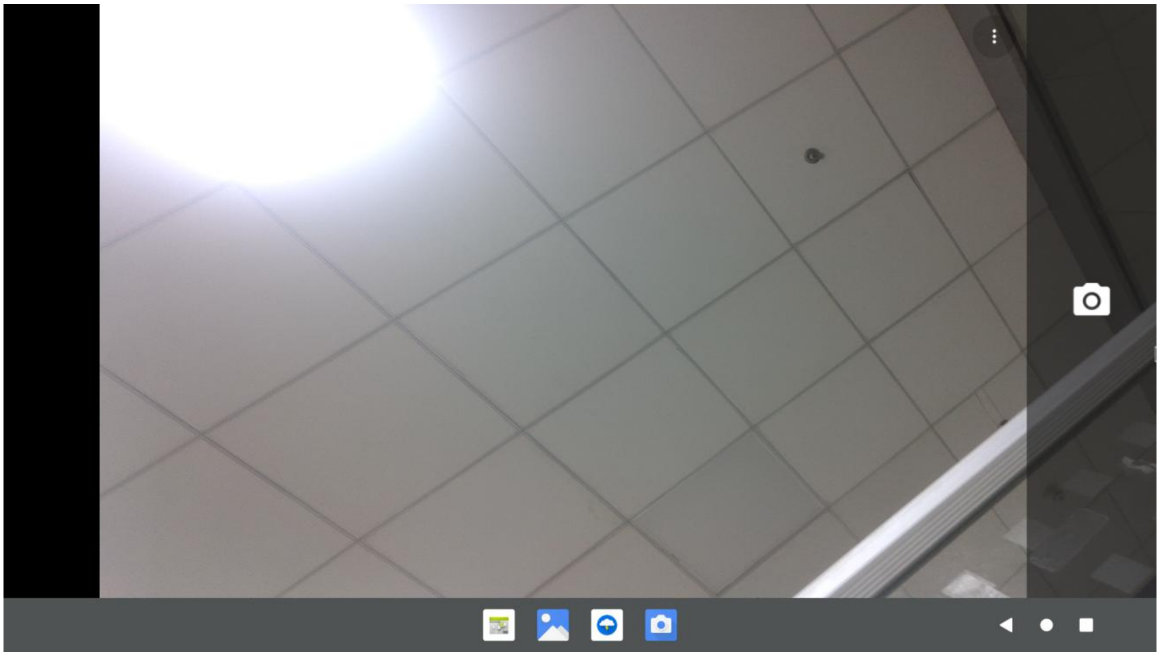


After connecting the camera to the development board, we can use the following method to test the camera:

a. Open the camera APP on the desktop



b. Then you can see the preview screen of the camera

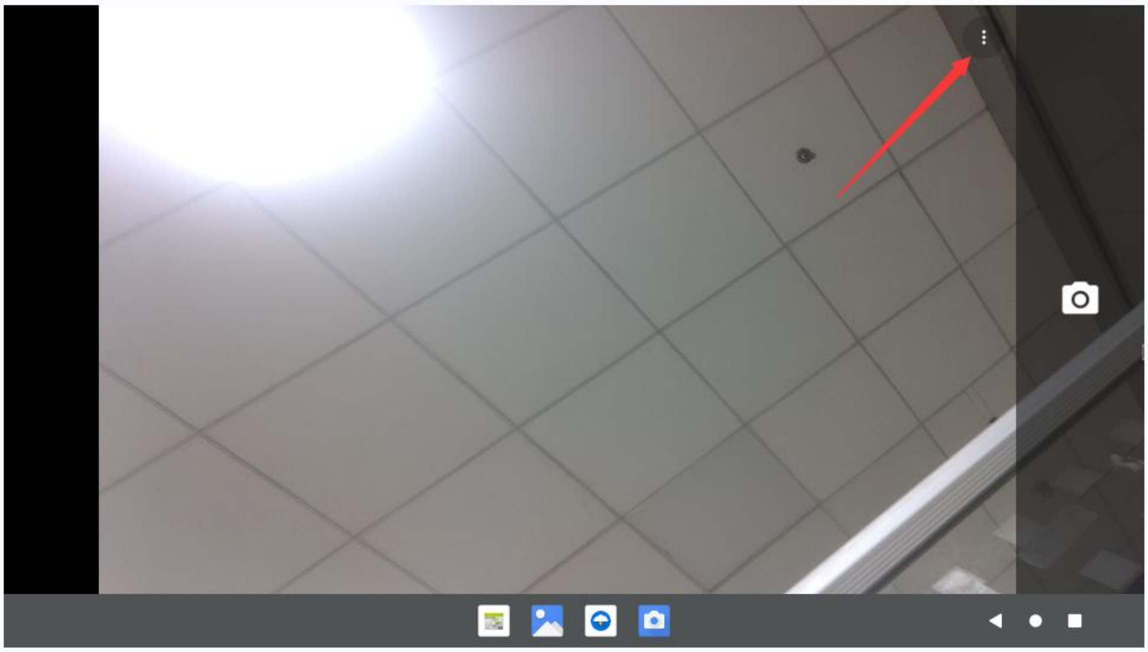


In addition to single camera, we can also connect two cameras at the same time.

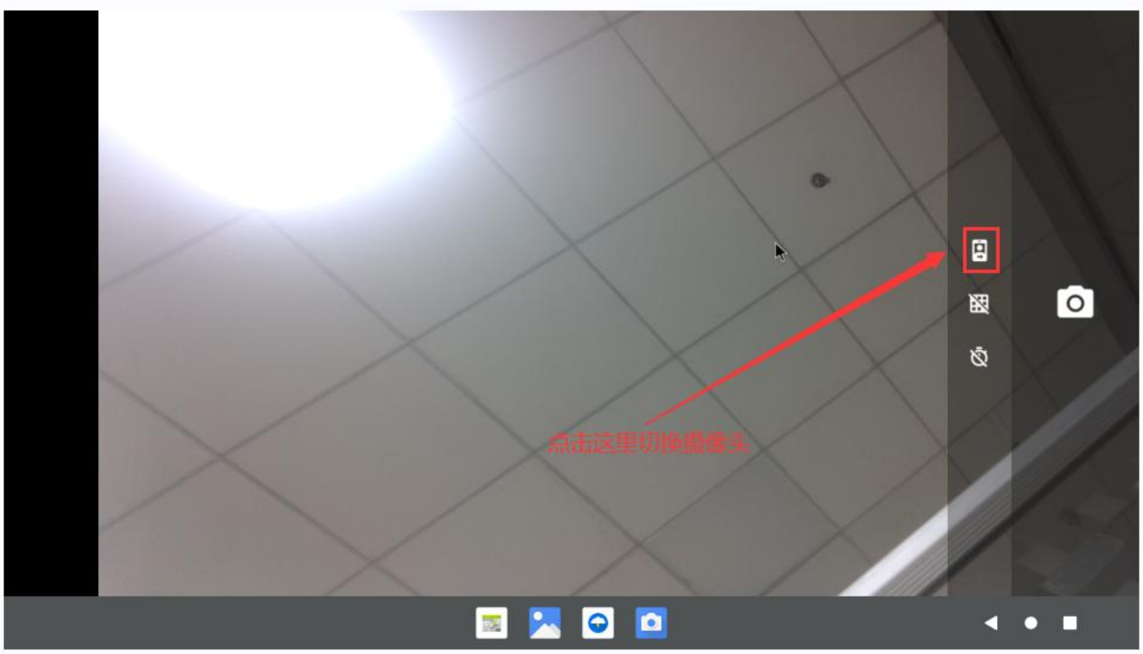
After connecting the dual cameras, follow the same steps as before and open the camera APP to see the picture of one of the cameras.

How to switch to another camera:

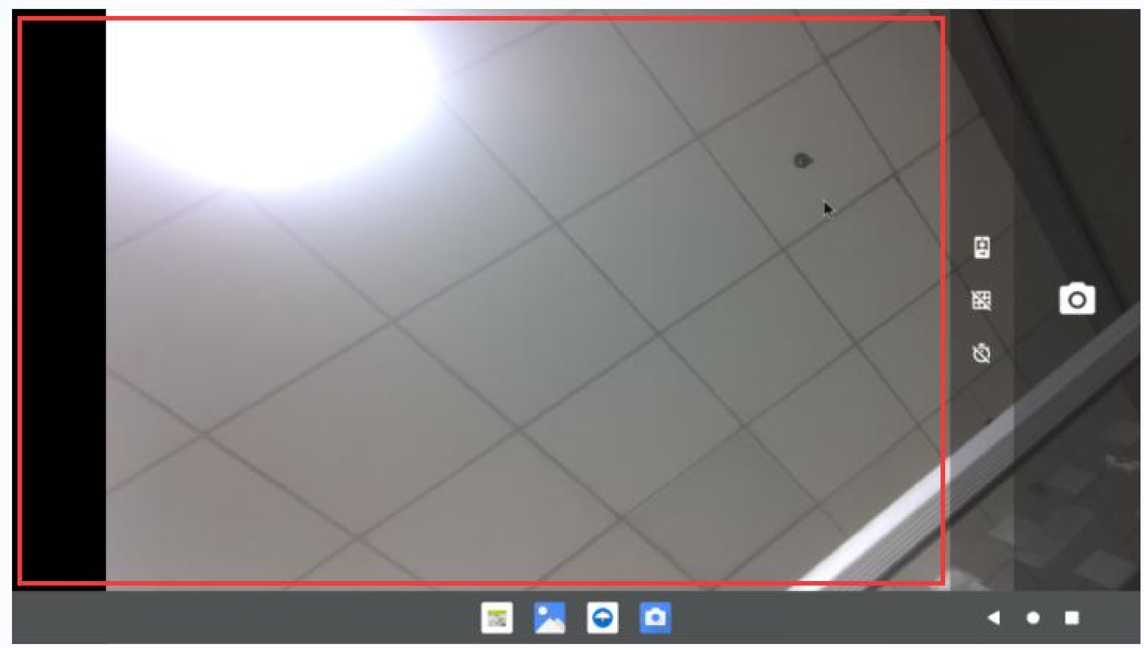
a. First click on the three dots in the upper right corner



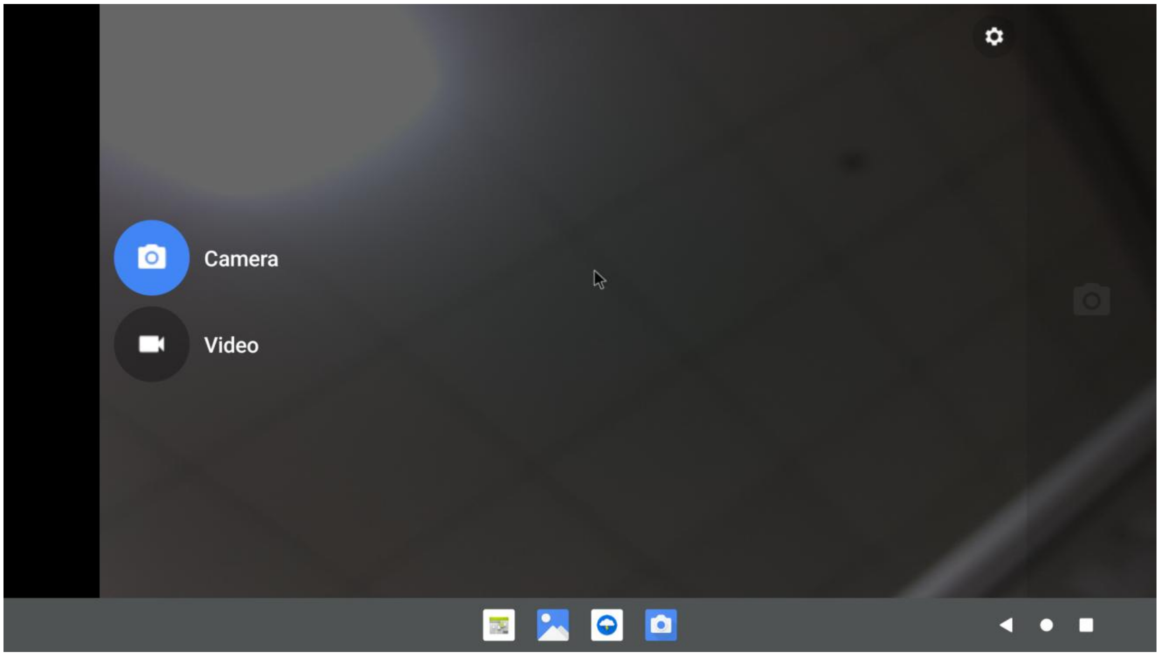
b. Then click the position shown in the picture below to switch the camera



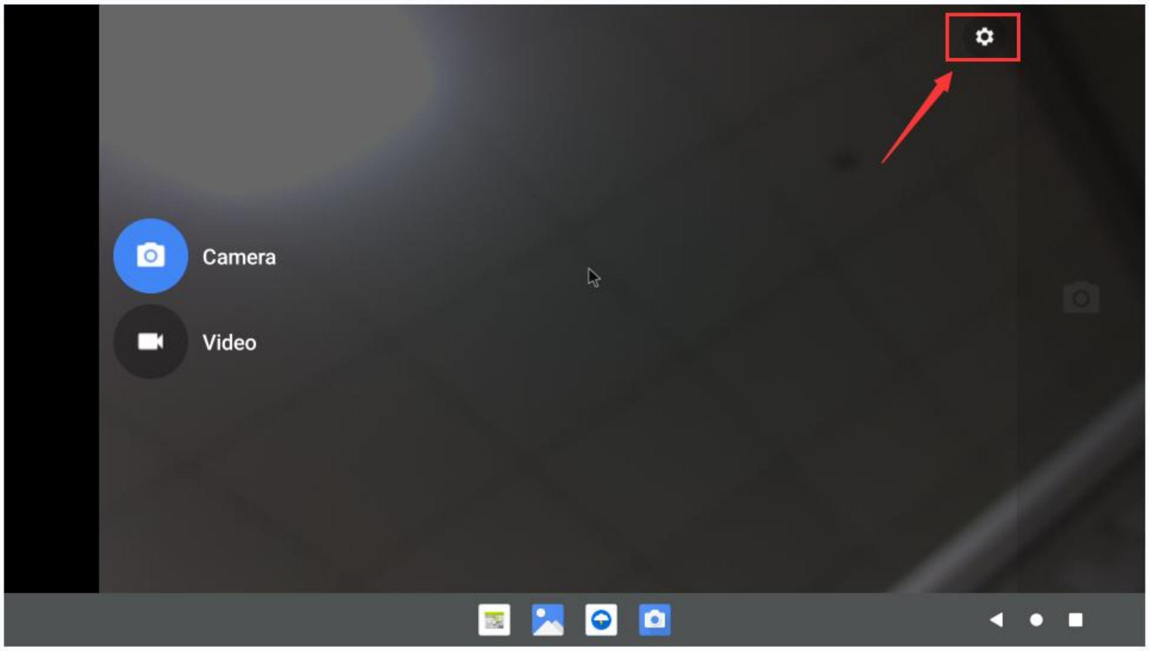
Press and hold the mouse in the area shown in the red box in the picture below on the camera APP and drag it to the right to bring up the switching interface between taking pictures and recording videos.



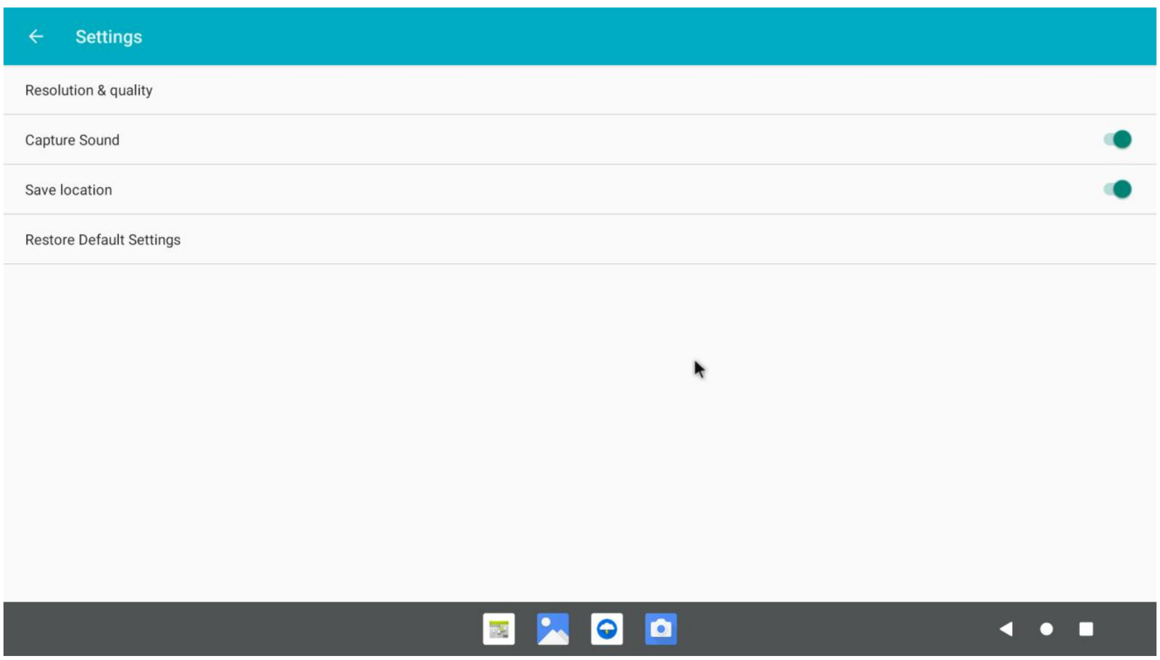
The switching interface between photo taking and video recording is as shown below. Click **Video** to switch to video recording mode.



Click the location shown in the picture below to enter the camera settings interface



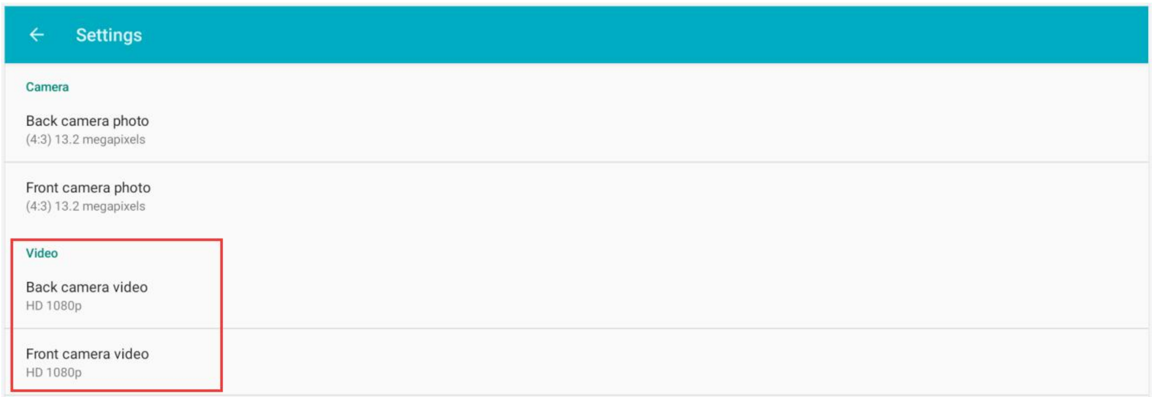
The camera setting interface is as follows:



Currently tested OV13850 does not support 4K video recording (OV13855 supports it), and only supports up to 1080p. When recording video, please switch the video format to 1080p in the settings. The steps are as follows:

a. First enter the setting interface of the camera APP, then click **Resolution & quality**

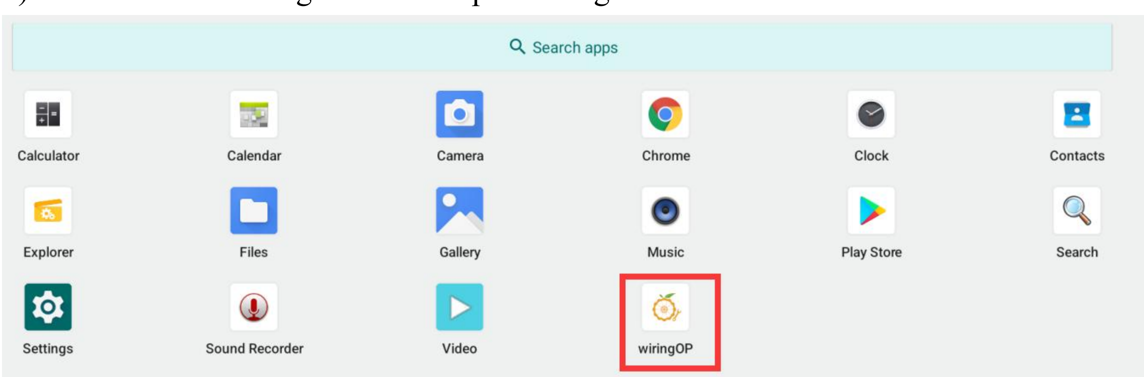
b. Then set the video format to 1080p in **Video**



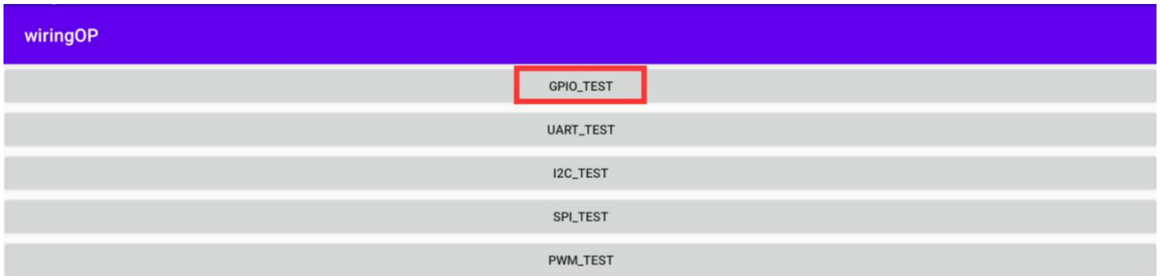
7. 8. 40pin interface GPIO, UART, SPI and PWM testing

7. 8. 1. 40pin GPIO port test

1) First click the wiringOP icon to open wiringOP APP

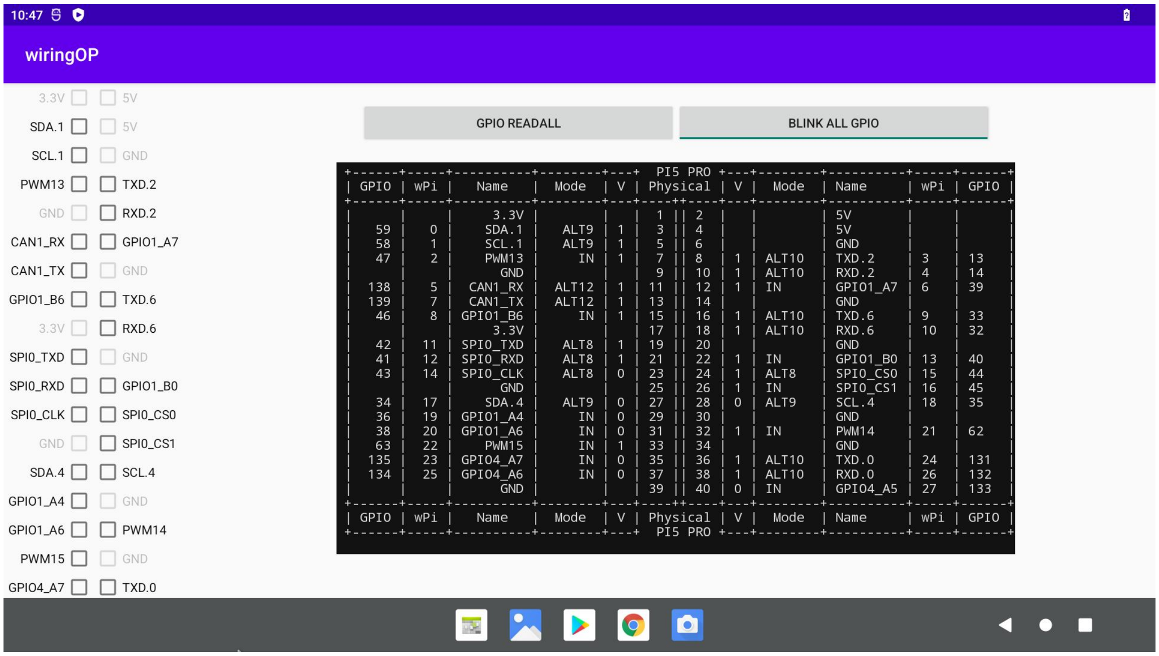


2) The main interface of wiringOP APP is displayed as shown below, then click the **GPIO_TEST** button to open the GPIO test interface

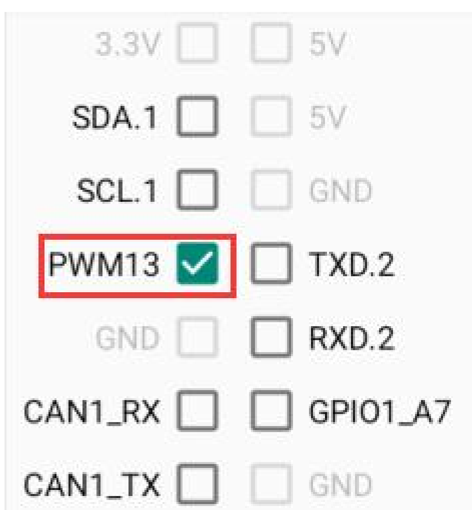


3) The GPIO test interface is as shown in the figure below. The two rows of **CheckBox** buttons on the left have a one-to-one correspondence with the 40pin pins. When the **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT** mode and the pin level is set to high level; when unchecked, the GPIO pin level is set to low level; when the right When pressing the **GPIO READALL** button, you can obtain information such as wPi number, GPIO mode, pin level, etc.;
When you click the **BLINK ALL GPIO** button, the program will control the 28 GPIO ports to continuously switch between high and low levels.

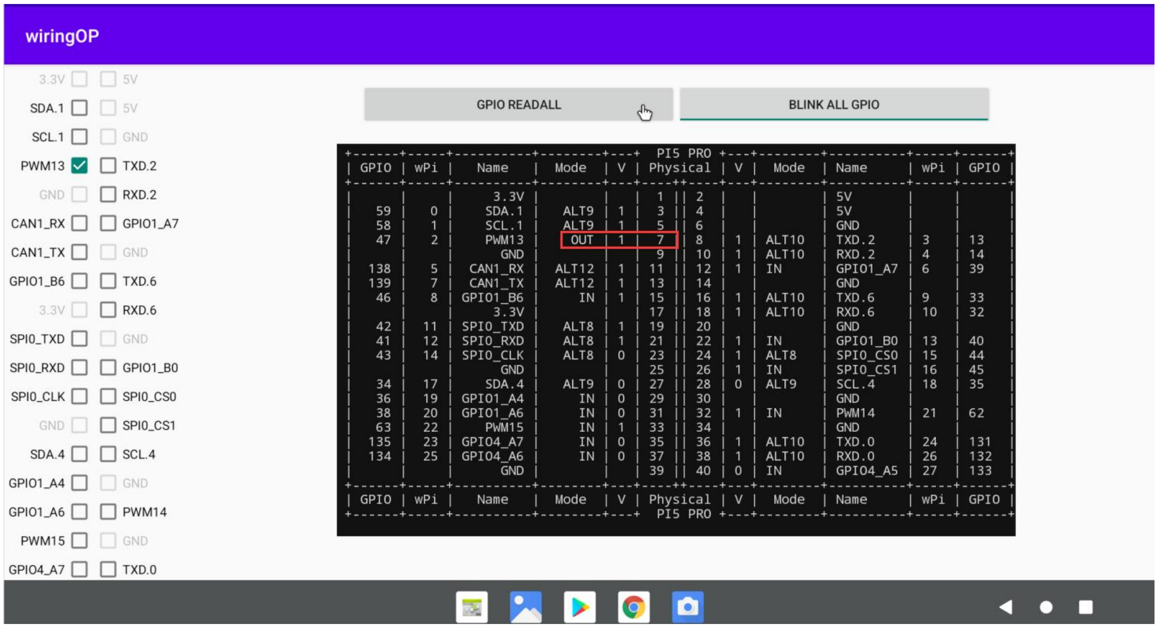
4) Then click the **GPIO READALL** button, and the output information is as shown below:



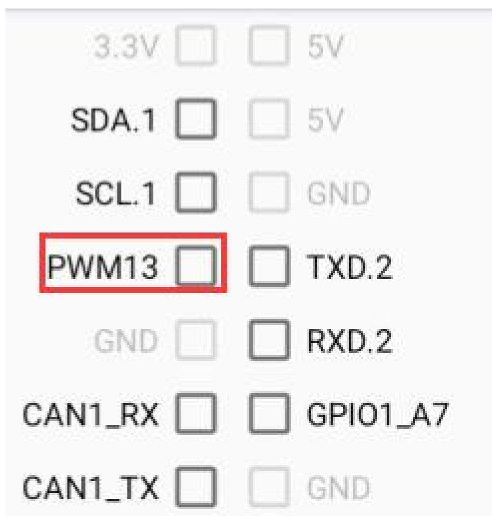
5) There are a total of 28 GPIO ports in the 40-pin development board that can be used. The following uses pin No. 7 - the corresponding GPIO is GPIO1_B7 - the corresponding wPi serial number is 2 - as an example to demonstrate how to set the high and low levels of the GPIO port. First click the **CheckBox** button corresponding to pin No. 7. When the button is selected, pin No. 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 the setting high level success



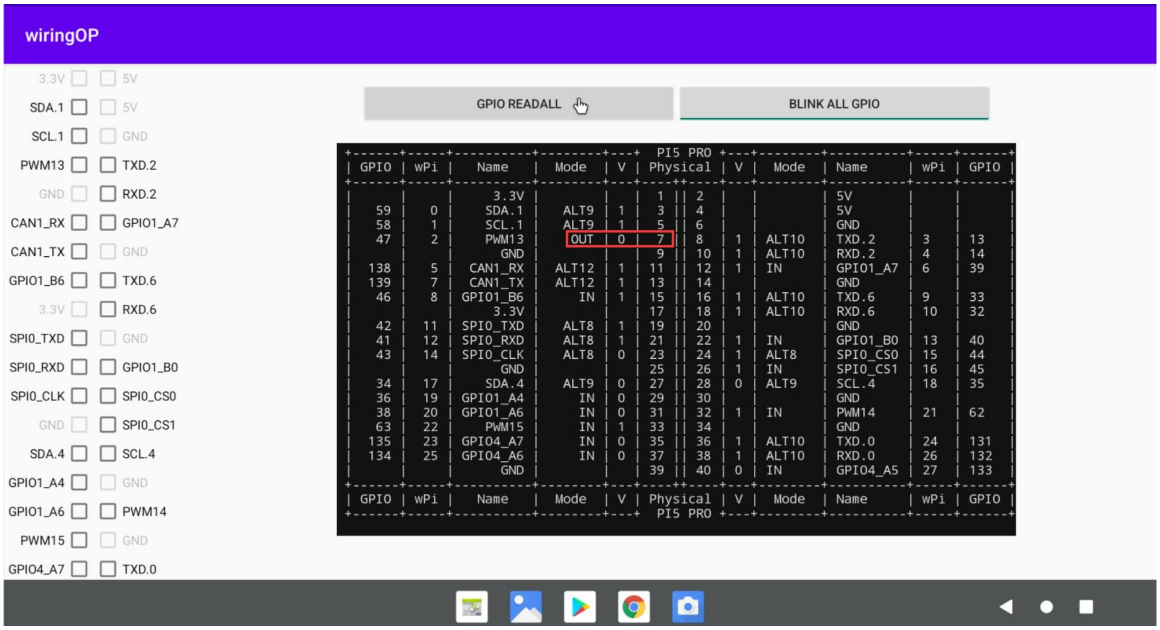
6) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is **OUT** and the pin level is high level



7) Click the **CheckBox** button in the picture below again to uncheck it. Pin No. 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 the low level setting is successful.

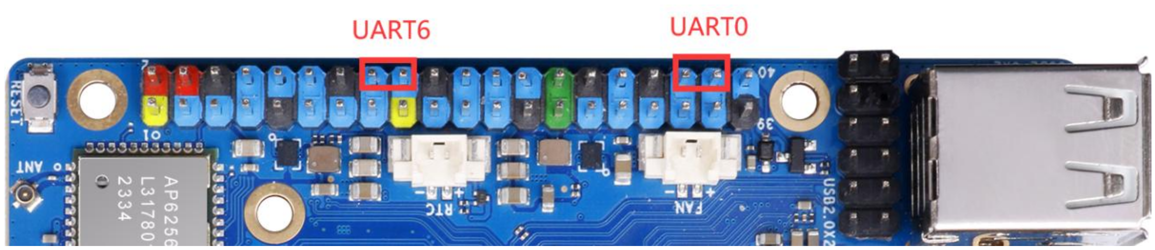


8) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is OUT and the pin level is low level

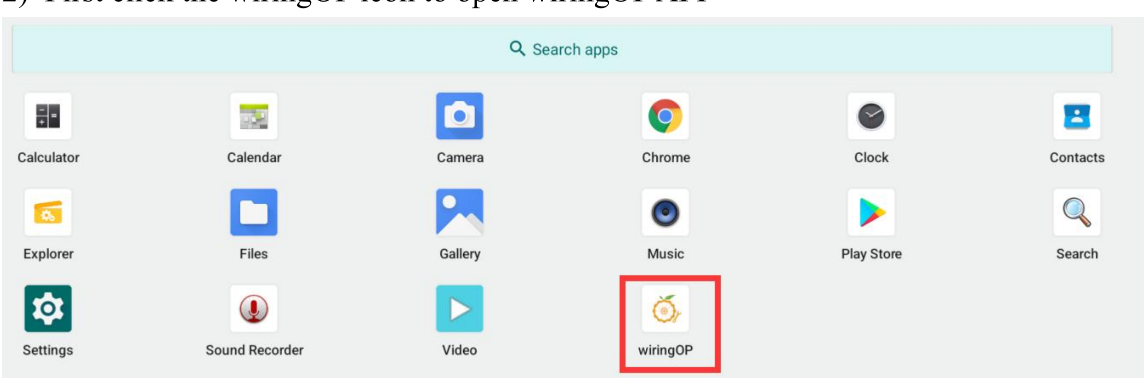


7. 8. 2. 40pin UART test

1) Two serial ports, UART0 and UART6, are opened by default in Android. The 40pin position is as shown in the figure below. The corresponding device nodes are **/dev/ttyS0** and **/dev/ttyS6**

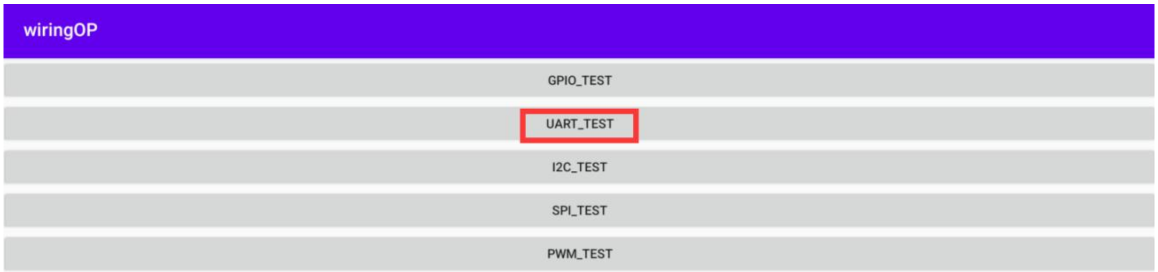


2) First click the wiringOP icon to open wiringOP APP

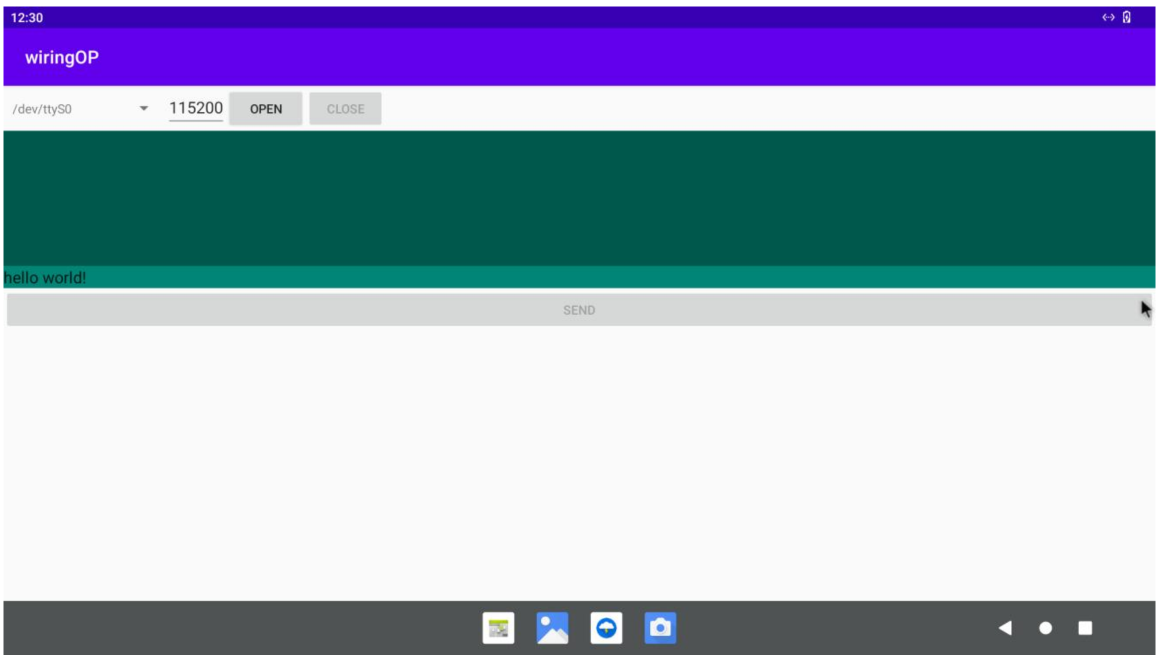


3) The main interface of wiringOP APP is displayed as shown below, then click the

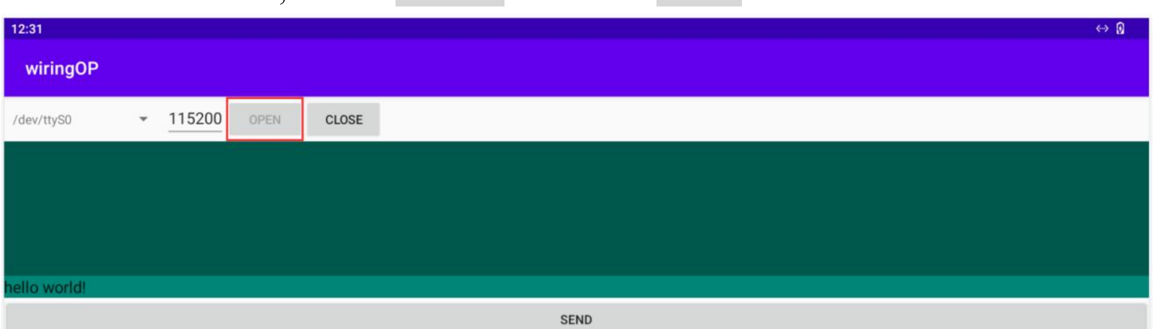
UART_TEST button to open the UART test interface



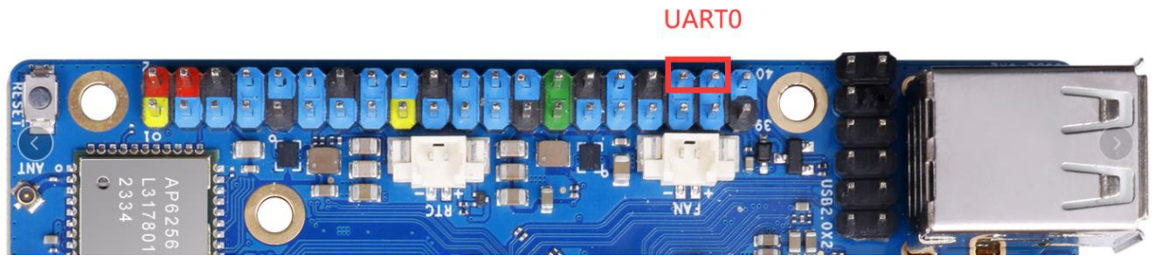
4) The serial port test interface of the APP is as shown in the figure below



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.

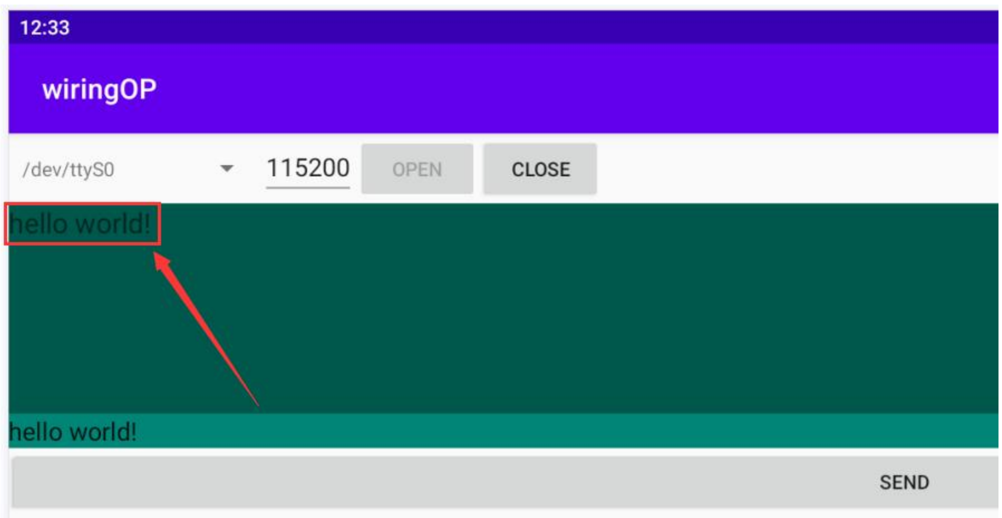


6) Then use Dupont wire to short the RXD and TXD pins of uart0



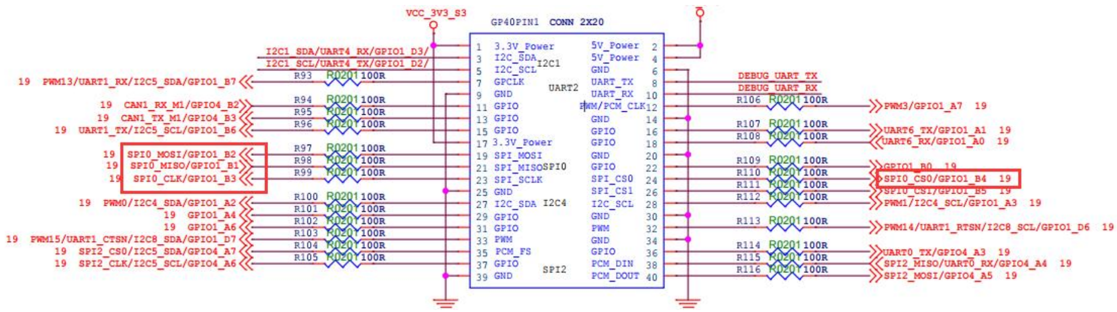
7) Then you can enter a character in the send edit box below and click the **SEND** button to start sending.

8) If everything is normal, the received string will be displayed in the receiving box

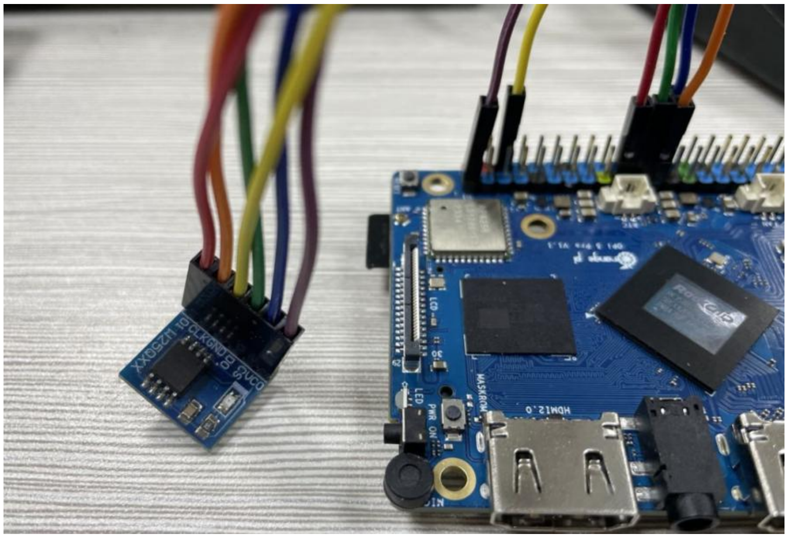


7. 8. 3. 40pin SPI test

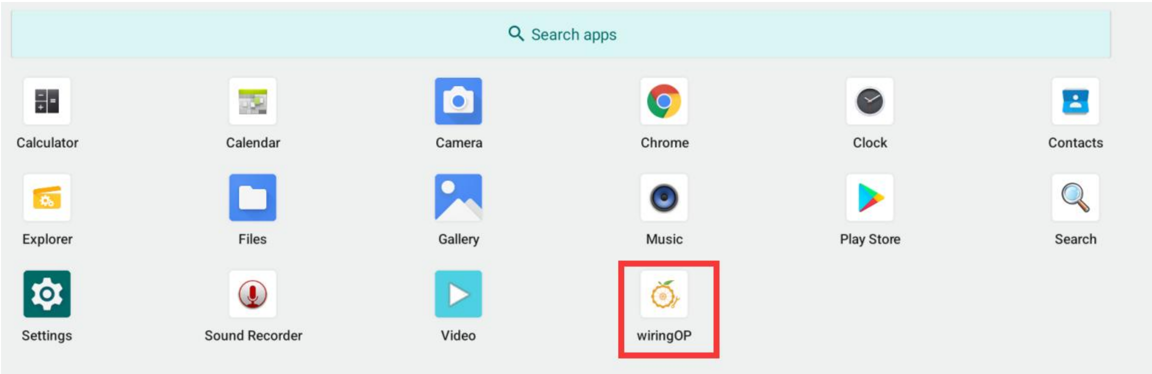
1) According to the schematic diagram of the 40pin interface, the available spi for Orange Pi 5 Pro is spi0



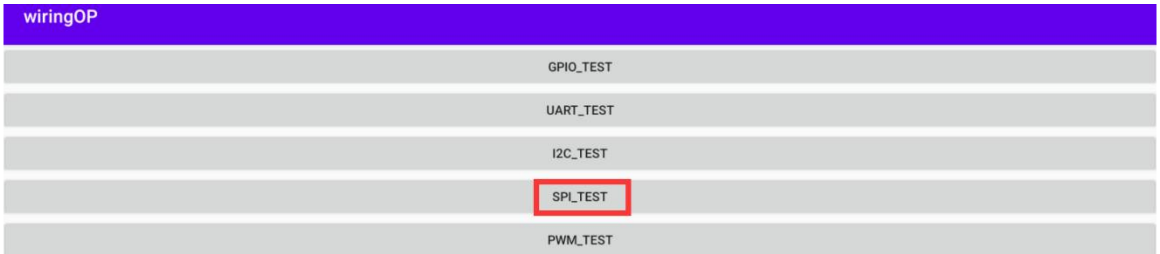
2) Here the SPI interface is tested through the w25q64 module. First, connect the w25q64 device to the SPI0 interface.



3) Then click the wiringOP icon to open wiringOP APP

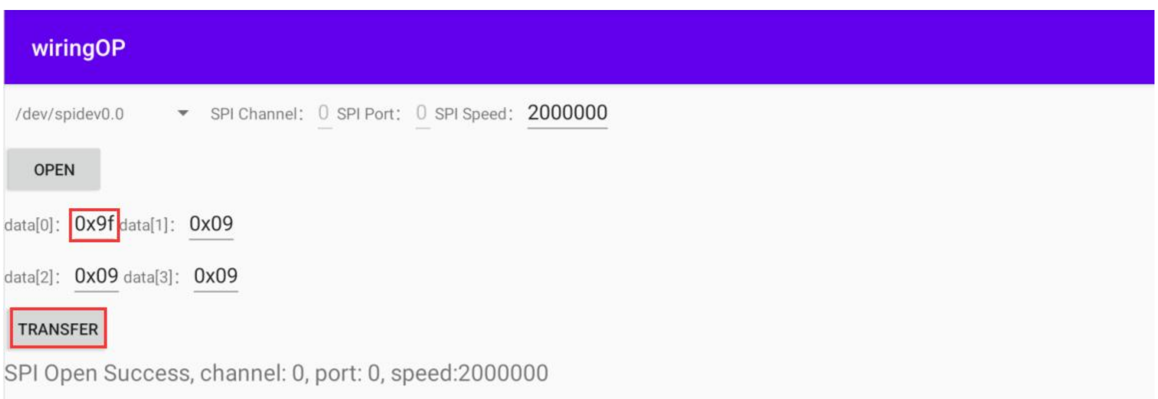


4) The main interface of wiringOP APP is displayed as shown below. Click the SPI_TEST button to open the SPI test interface.

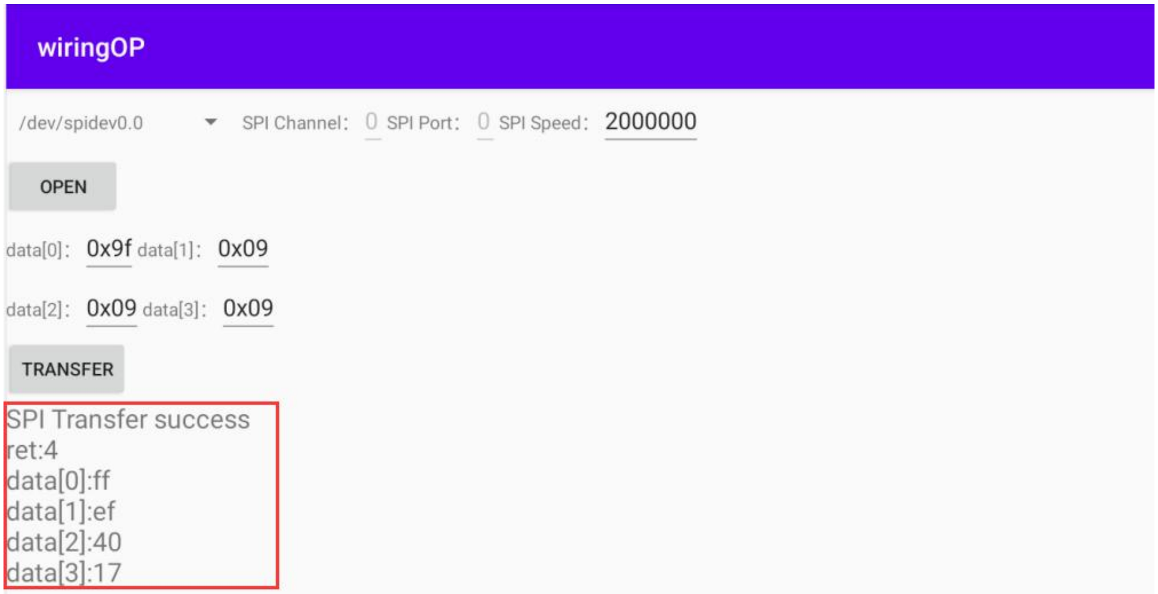


5) Then click the **OPEN** button to initialize the SPI

6) Then fill in the bytes that need to be sent, such as reading the ID information of w25q64, filling in the address 0x9f in data[0], and then click the **TRANSFER** button



7) Finally, the APP will display the read ID information.

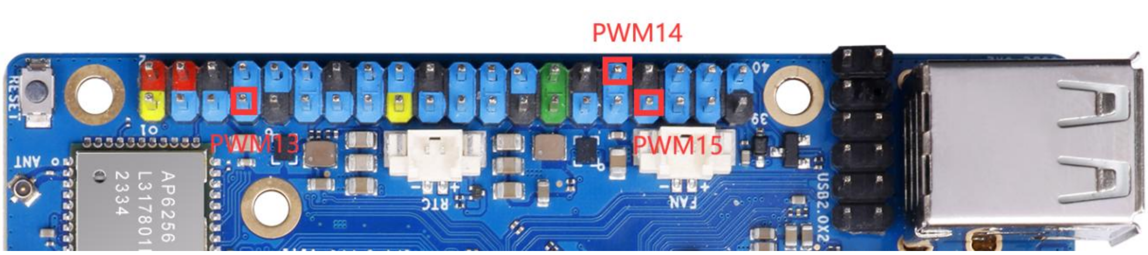


8) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, which corresponds to the value read above (h represents hexadecimal)

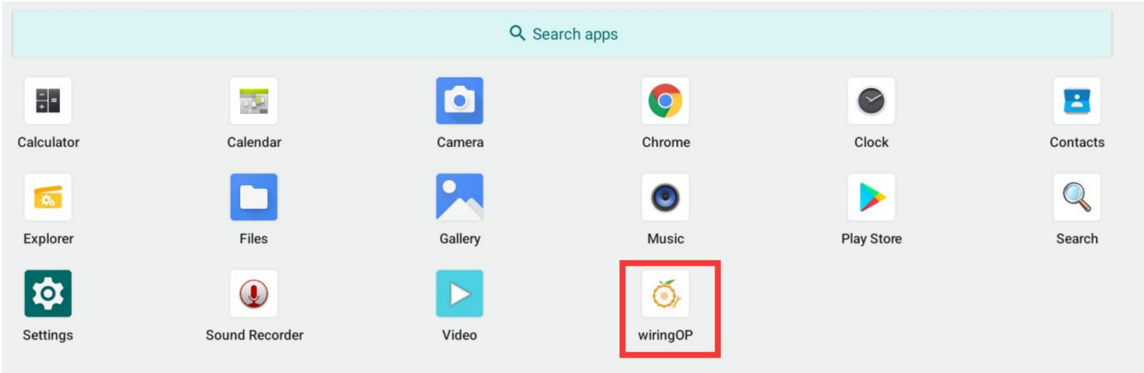
MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

7. 8. 4. 40pin PWM test

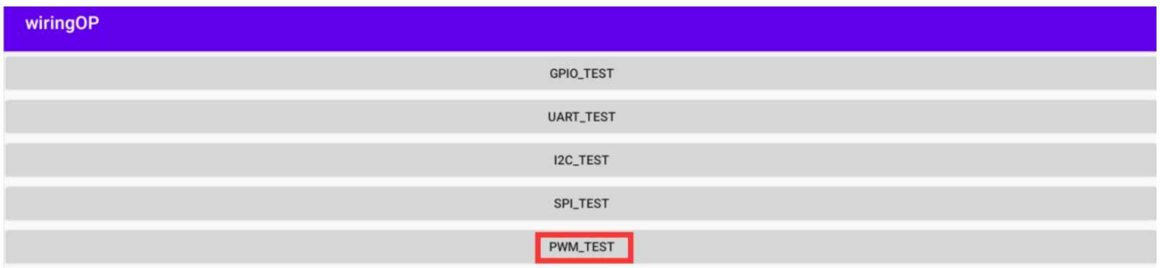
1) Android has **PWM13**, **PWM14** and **PWM15** turned on by default. The corresponding pins are located at 40pin as shown in the figure below.



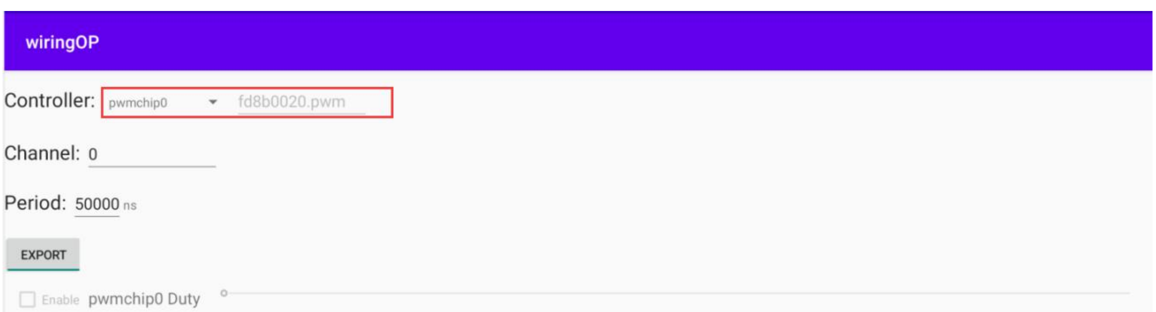
2) First click the wiringOP icon to open wiringOP APP



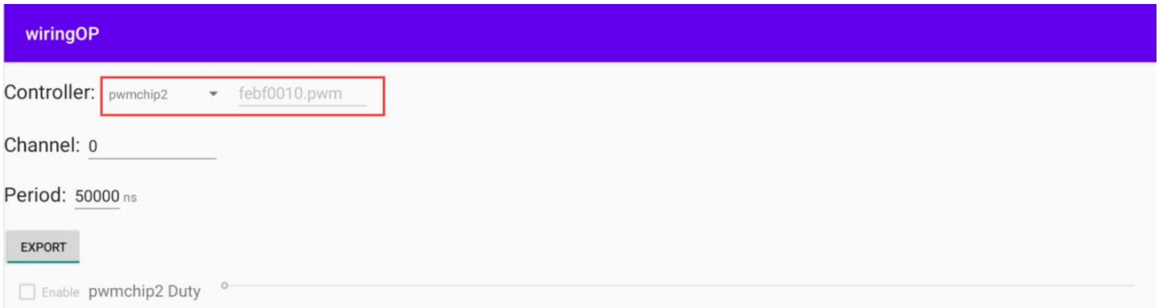
3) Then click the **PWM_TEST** button on the main interface of wiringOP to enter the PWM test interface



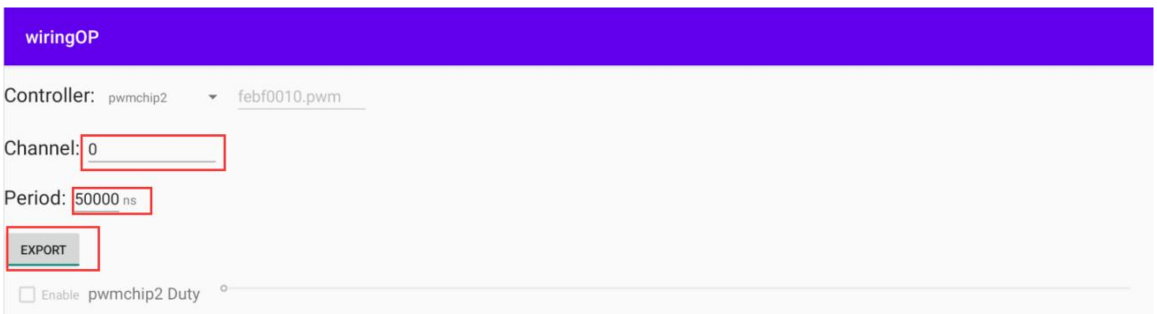
4) The base address of PWM13 is **febf0010**, the base address of PWM14 is **febf0020**, and the corresponding base address of PWM15 is **febf0030**. Here, **fd8b0020.pwm** is displayed on the right side of pwmchip0. At this time, you need to click the drop-down option to select other pwmchips. Take **PWM13** as an example. When When **febf0010.pwm** is displayed on the right, it means that **PWM13** is selected.



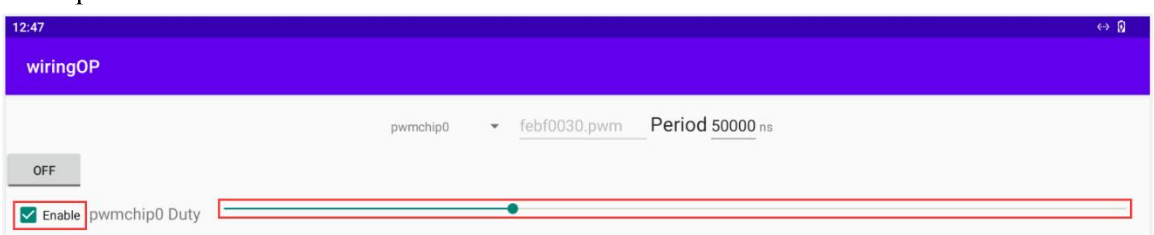
5) When the pull-down option selects **pwmchip2**, the corresponding base address of PWM13 on the right is **febf0010**



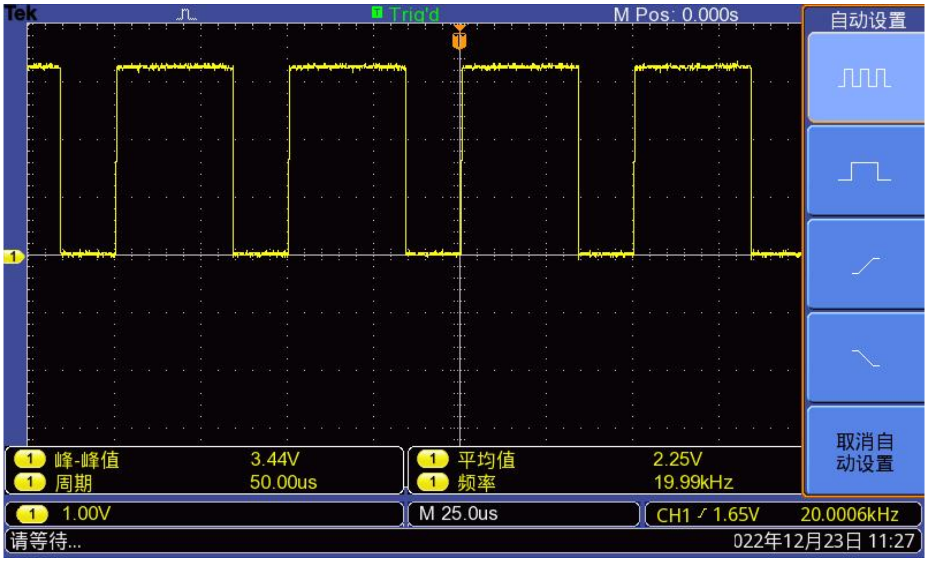
6) Then confirm the PWM channel, the default is channel 0, and confirm the PWM period. The default configuration is **50000ns**, which converts to a PWM frequency of **20KHz**. You can modify it yourself. Click the **EXPORT** button to export **PWM13**



7) Then drag the drag bar below to change the PWM duty cycle, and then check Enable to output the PWM waveform.



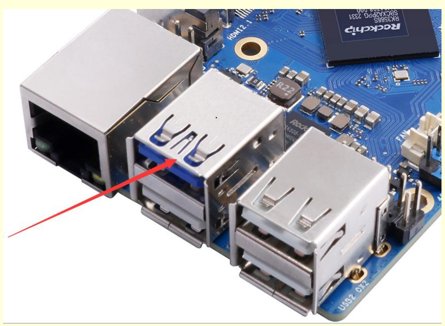
8) Then use an oscilloscope to measure pin No. 7 of the 40pin development board and you can see the following waveform.



7. 9. How to use ADB

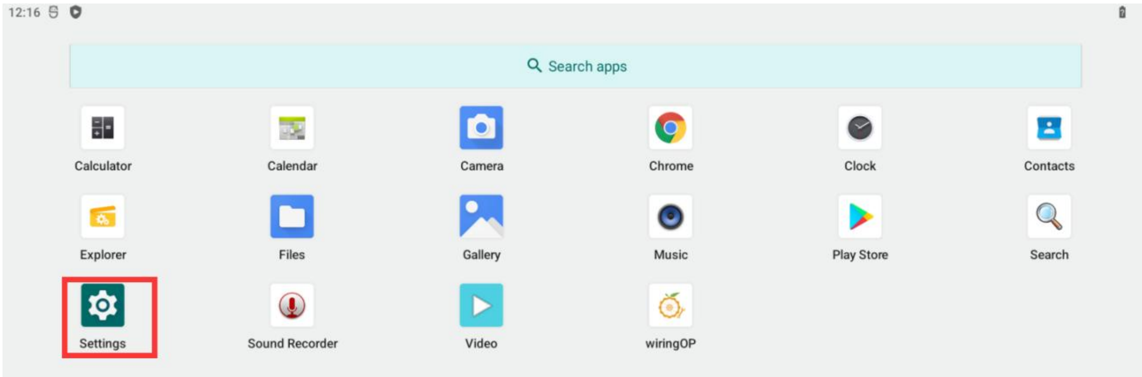
7. 9. 1. USB OTG mode switching method

The development board has 4 USB interfaces. The USB interface marked with the red box in the figure below can support both Host mode and Device mode. The other 3 USB interfaces only support Host mode.

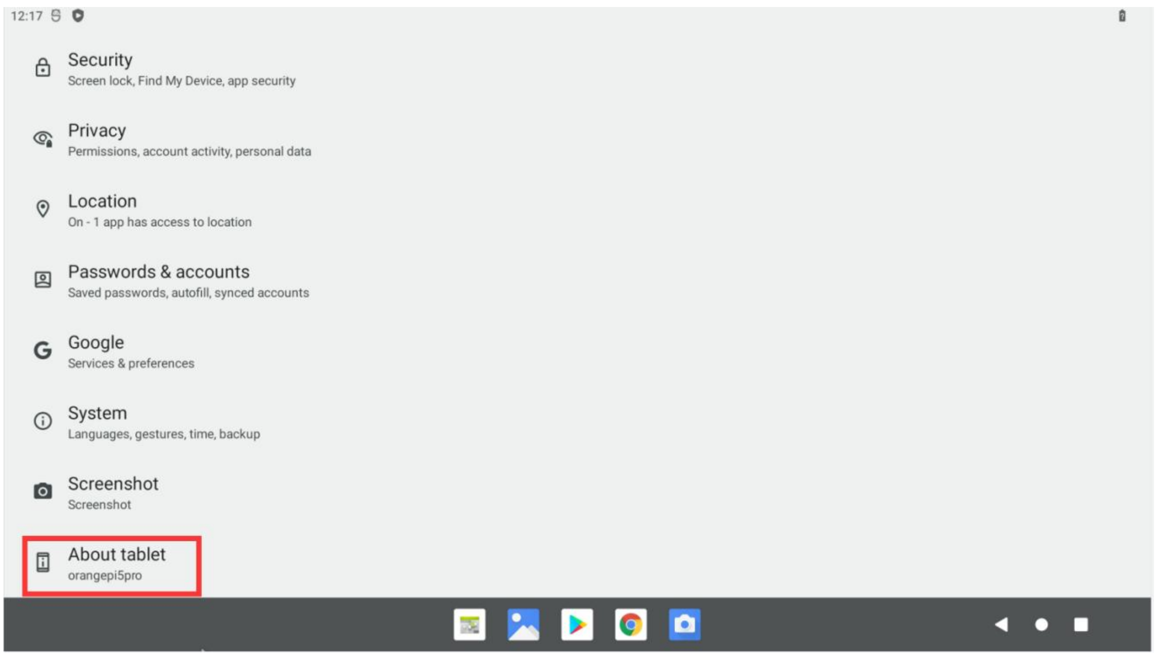


The USB OTG interface defaults to Host mode, which can be used to connect USB devices such as mouse and keyboard. If you want to use ADB, you need to manually switch to Device mode.

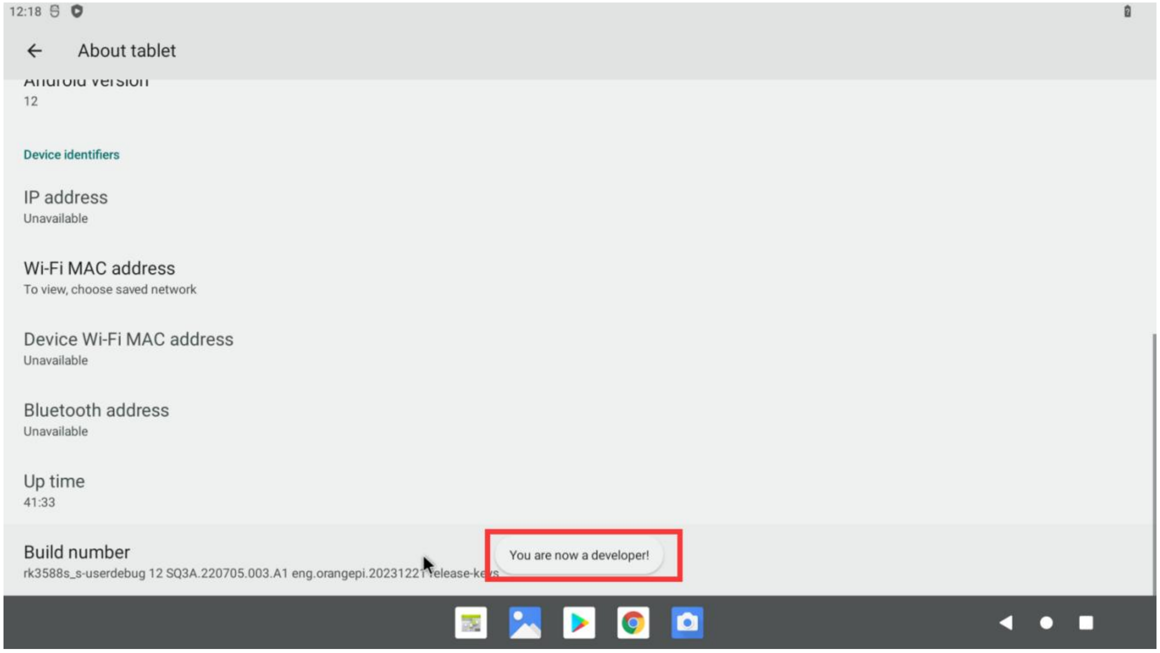
1) First open Settings



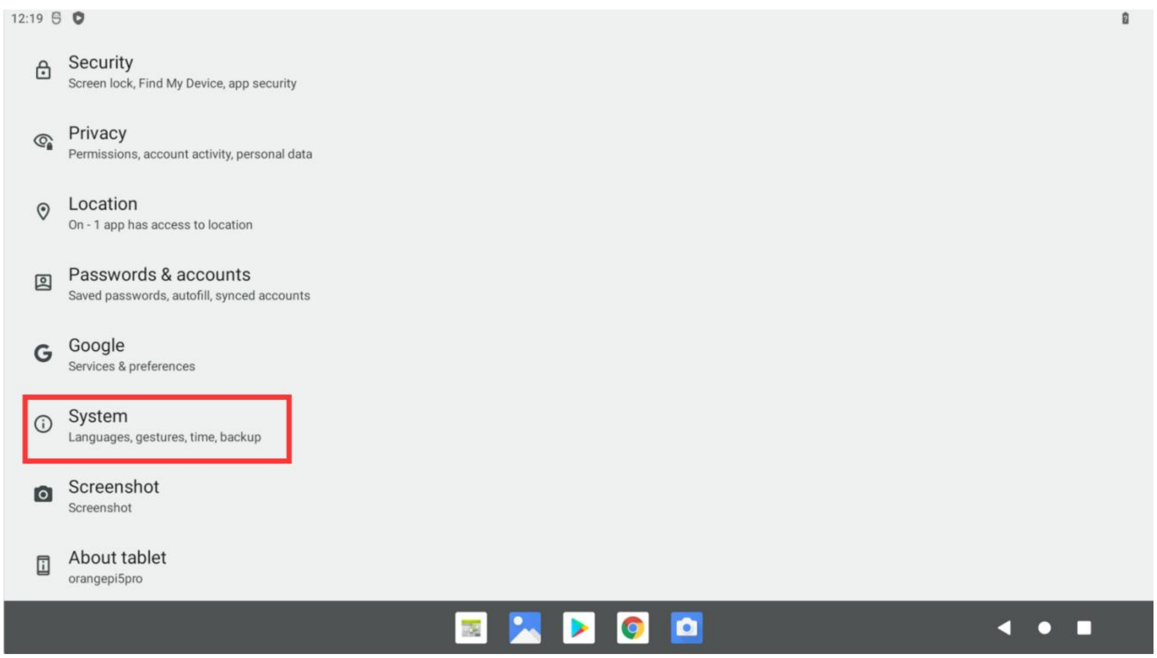
2) Then select **About tablet**



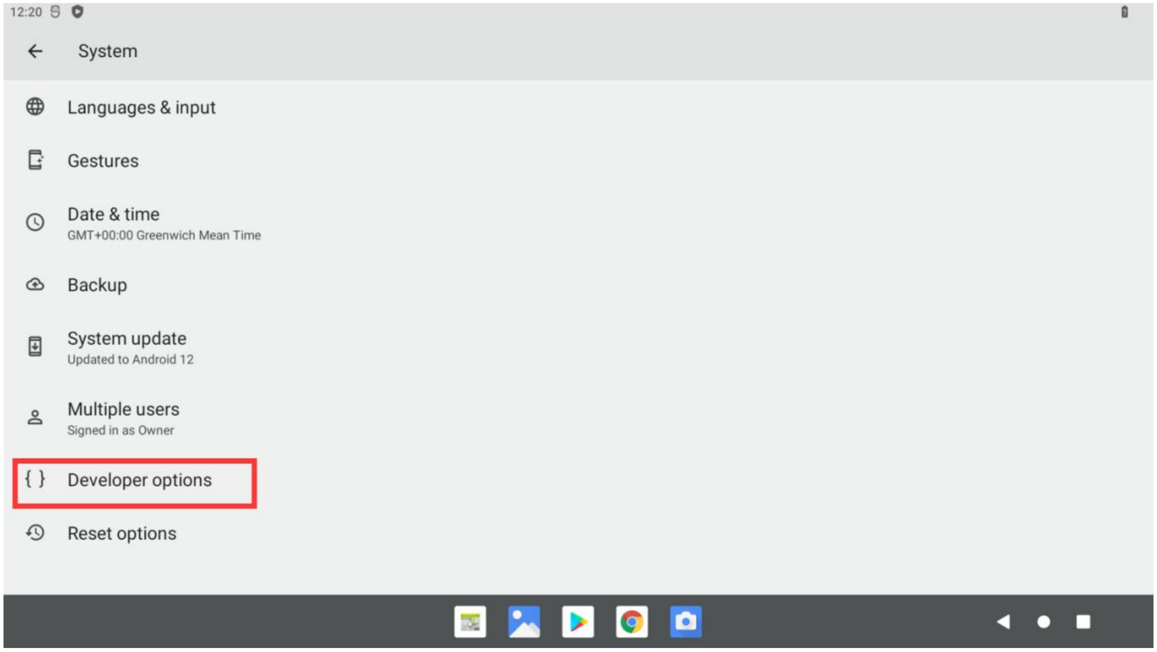
3) Then click the **Build number** menu bar multiple times with the mouse until the prompt **You are now a developer!** appears.



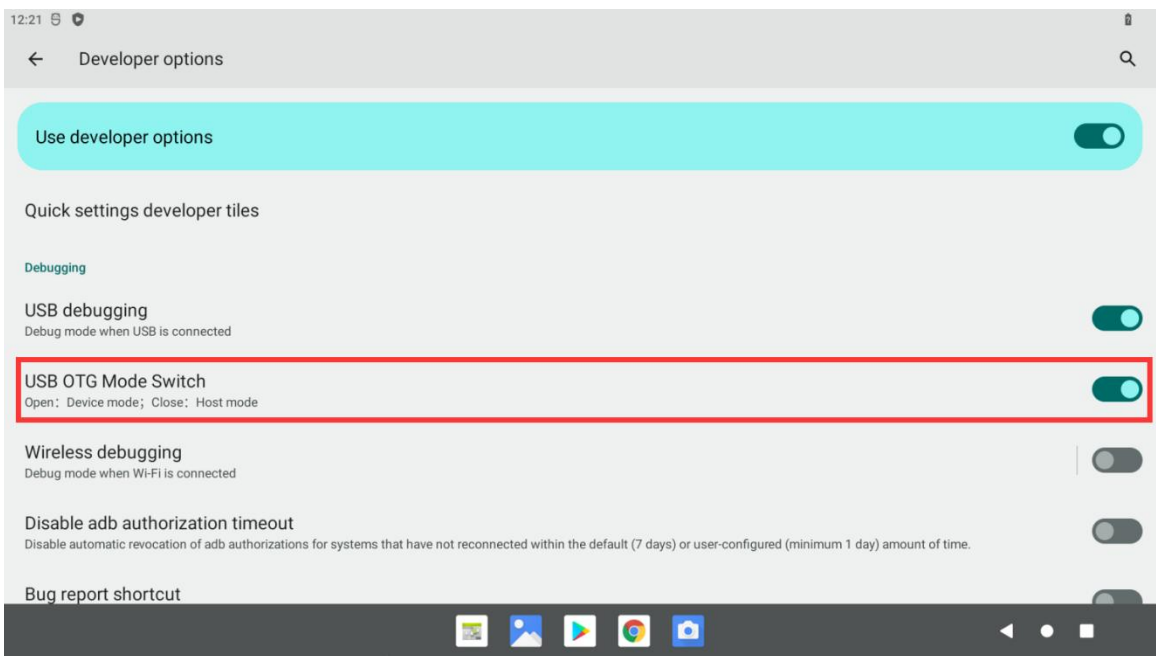
4) Then click to return to the previous menu and select **System**



5) Then select **Developer options**



6) Finally find the **USB OTG Mode Switch**, turn on the switch to switch to Device mode, **and turn off the switch to switch to Host mode**



7.9.2. Use data cable to connect adb debugging

4) First prepare a good quality USB2.0 male-to-male data cable

5) Then refer to **the USB OTG mode switching method** to switch the USB OTG to device mode.

6) Then use a USB2.0 male-to-male data cable to connect the development board to the USB interface of the computer (please also use a TypeC power supply to power the development board)

7) Install adb tool on Ubuntu PC

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

8) Use the following command to view the recognized ADB devices

```
test@test:~$ adb devices
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

9) Then you can log in to the android system through adb shell on Ubuntu PC

```
test@test:~$ adb shell
console:/ $
```

10) Execute the command to remount the Android system

```
test@test:~$ adb root
test@test:~$ adb remount
```

11) Then you can transfer files to the Android system

```
test@test:~$ adb push example.txt /system/
```

7.9.3. Use network connection adb debugging

Using network adb does not require a USB Typc C interface data cable to connect the computer and development board, but communicates through the network, so first make sure that the wired or wireless network of the development board is connected, and then obtain the IP address of the development board. Later To be used.

1) Ensure that the **service.adb.tcp.port** of the Android system is set to the 5555 port number

```
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 adb shell on Ubuntu PC

```
test@test:~$ adb shell
```

```
console:/ #
```

7. 10. 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, just plug it in and it’s ready to use.

7. 11. How to use the HDMI CEC function of Android Box system

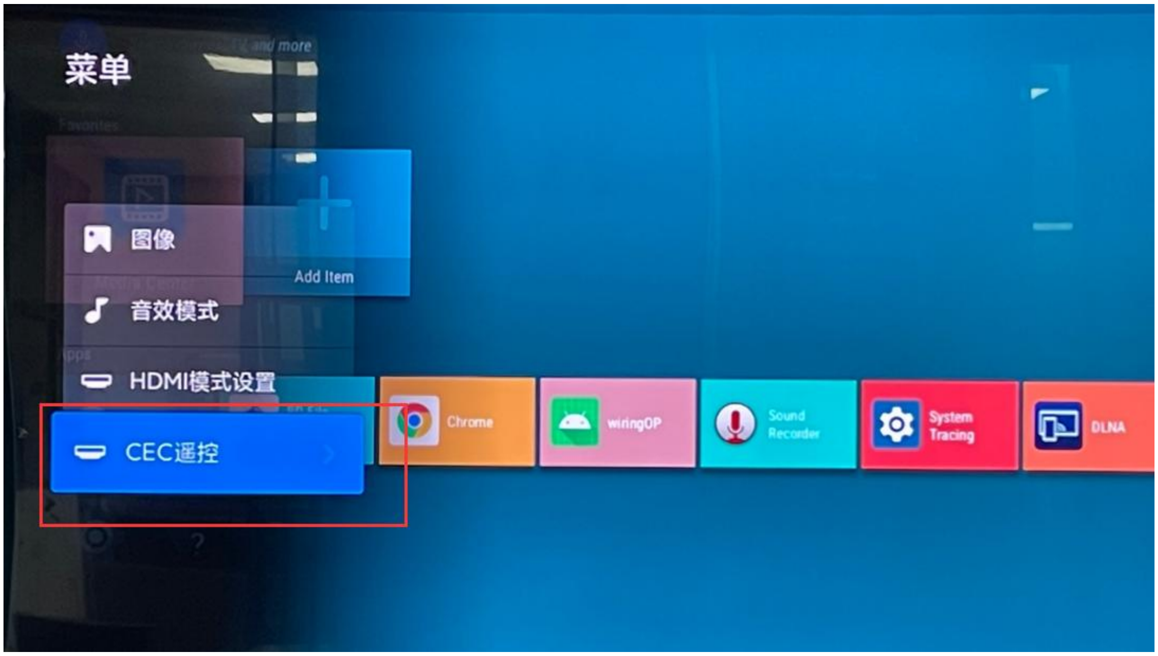
HDMI CEC allows users to control all connected devices through HDMI with only one remote control. Based on this function, we can control the development board with the TV's remote control.

Before testing this function, please make sure your TV supports HDMI CEC.

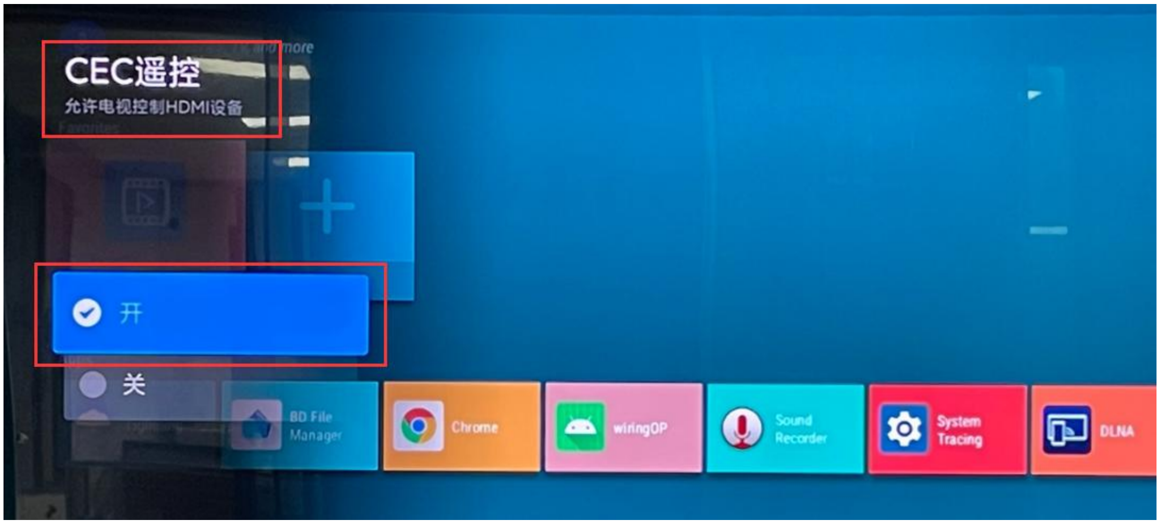
1) First connect the development board to the TV via HDMI cable, then power on and

start

2) Then turn on the HDMI CEC function in the TV settings. The turning on method of different TVs may be different. Here is Xiaomi TV as an example. Press the menu button of the remote control, then select the CEC remote control and press the confirm button.



3) Then select **"On"** to turn on the HDMI CEC remote control



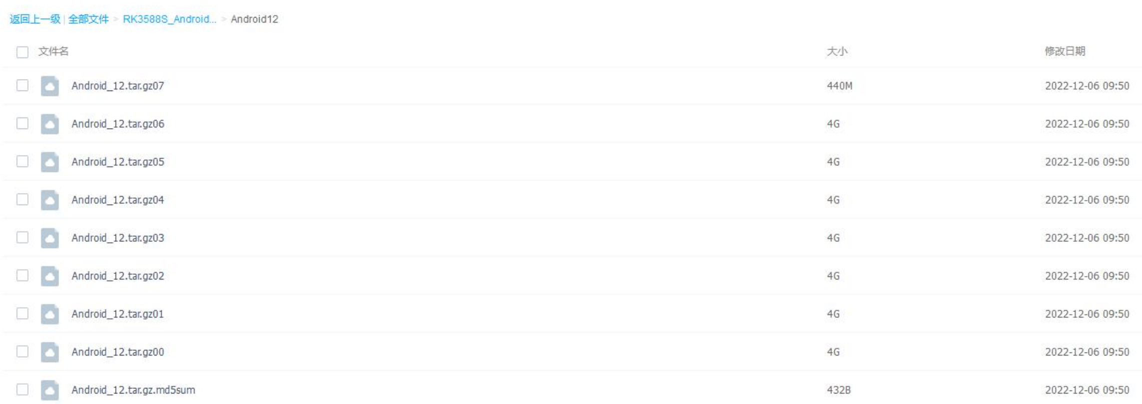
4) At this point, you can control the Android Box system of the development board through the TV's remote control.

8. Compilation method of Android 12 source code

8. 1. Download the source code of Android 12

1) First download the compressed package of Android 12 source code from Baidu Cloud Disk or Google Cloud Disk

a. Baidu cloud disk



b. Google Cloud Drive

2) After downloading the compressed package of Android 12 source code, please check whether the MD5 checksum is correct. If it is incorrect, please download the source code again.

```
test@test:~$ md5sum -c Android_12.tar.gz.md5sum
Android_12.tar.gz00: Confirm
Android_12.tar.gz01: Confirm
Android_12.tar.gz02: Confirm
Android_12.tar.gz03: Confirm
Android_12.tar.gz04: Confirm
Android_12.tar.gz05: Confirm
Android_12.tar.gz06: Confirm
Android_12.tar.gz07: Confirm
```

3) Then execute the following command to decompress

```
test@test:~$ cat Android_12.tar.gz0* | tar -xvzf -
```

8. 2. Compile the source code of Android 12

1) First install the software packages needed to compile 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 make.sh compilation script in the source code, and the compilation parameters are as follows

a. **-B**: Compile uboot
b. **-K**: Compile kernel
c. **-a**: Compile android
d. **-F**: Compile uboot、kernel and android
e. **-M**: Generate partition image in rockdev directory
f. **-u**: Package and generate a complete image that can finally be started
g. **--board**: Specify development board model

3) Compile uboot, kernel, android and package them into a complete image that can finally be started

a. The command to compile support for HDMI 8K display image (LCD is turned off by default) is as follows:

```
test@test:~$ cd Android_12
test@test:~/ Android_12$ ./make.sh -FMu --board orangepi5pro --nvme --gapps
```

b. The command to compile support for LCD display image is as follows:

```
test@test:~$ cd Android_12
test@test:~/ Android_12$ ./make.sh -FMu --board orangepi5pro --nvme --gapps --lcd
```

c. The command to compile the SATA boot image is as follows:

```
test@test:~$ cd Android_12
test@test:~/ Android_12$ ./make.sh -FMu --board orangepi5pro --sata --gapps
```

4) After compilation is completed, 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 generated image file will be placed in the **rockdev/Image-rk3588s_s** directory. Among them, **update.img** is the TF card boot image and **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 compile 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
```

9. OpenWRT system usage instructions

9. 1. OpenWRT version

OpenWRT version	Kernel version
snapshot	Linux6.1.43

9. 2. OpenWRT adaptation situation

Function	OpenWRT
USB2.0x3	OK
USB3.0x1	OK
eMMC interface	OK
FAN fan interface	OK
3pin debugging serial port	OK
TF card startup	OK
Gigabit network 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**
AP6256 WIFI	**OK**

9. 3. Expand the rootfs in the TF card before starting for the first time

1) When the TF card starts the OpenWRT system for the first time, it will execute the **resize-rootfs.sh** script to expand the rootfs. At this time, the system will restart several times.

2) After logging into the system, you can use the df -h command to check 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
```

9. 4. How to log in to the system

9. 4. 1. Log in through serial port

1) First, for the use of 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

9. 4. 2. Log in to the system via SSH

Please note that in the OpenWrt system of Orange Pi 5 Pro, the network port is configured as a LAN port function by default, so the LAN port of the development board needs to be directly connected to the network port of the computer. There is no way to obtain the IP if connected to a router.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) If the Ubuntu system is installed on the computer, 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 into the system, the display is as shown below

```
BusyBox v1.36.1 (2024-03-22 15:01:37 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt SNAPSHOT, r24892+731-1b7e62b20b
 -----------------------------------------------------
=== 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 refer to the method of SSH remote login to the development board under Windows in the Linux system instruction manual to log in.

9. 4. 3. Log in to the LuCI management interface

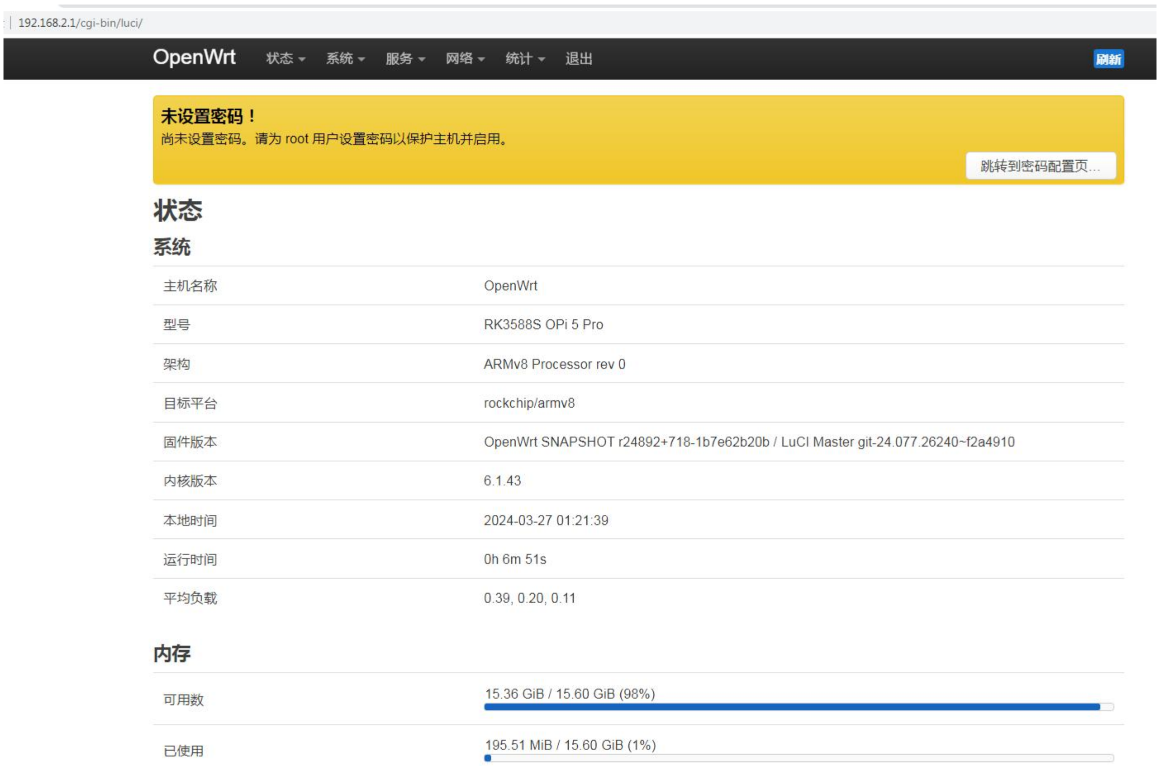
Please note that in the OpenWrt system of Orange Pi 5 Pro, the network port is configured as a LAN port function by default, so the LAN port of the development board needs to be directly connected to the network port of the computer. There is no way to obtain the IP if connected to a router.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) Enter the IP address **192.168.2.1** in the browser on your 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 will be displayed as shown below.



9. 4. 4. Log in to the terminal through the LuCI management interface

Please note that in the OpenWrt system of Orange Pi 5 Pro, the network port is configured as a LAN port function by default, so the LAN port of the development board needs to be directly connected to the network port of the computer. There is no way to obtain the IP if connected to a router.

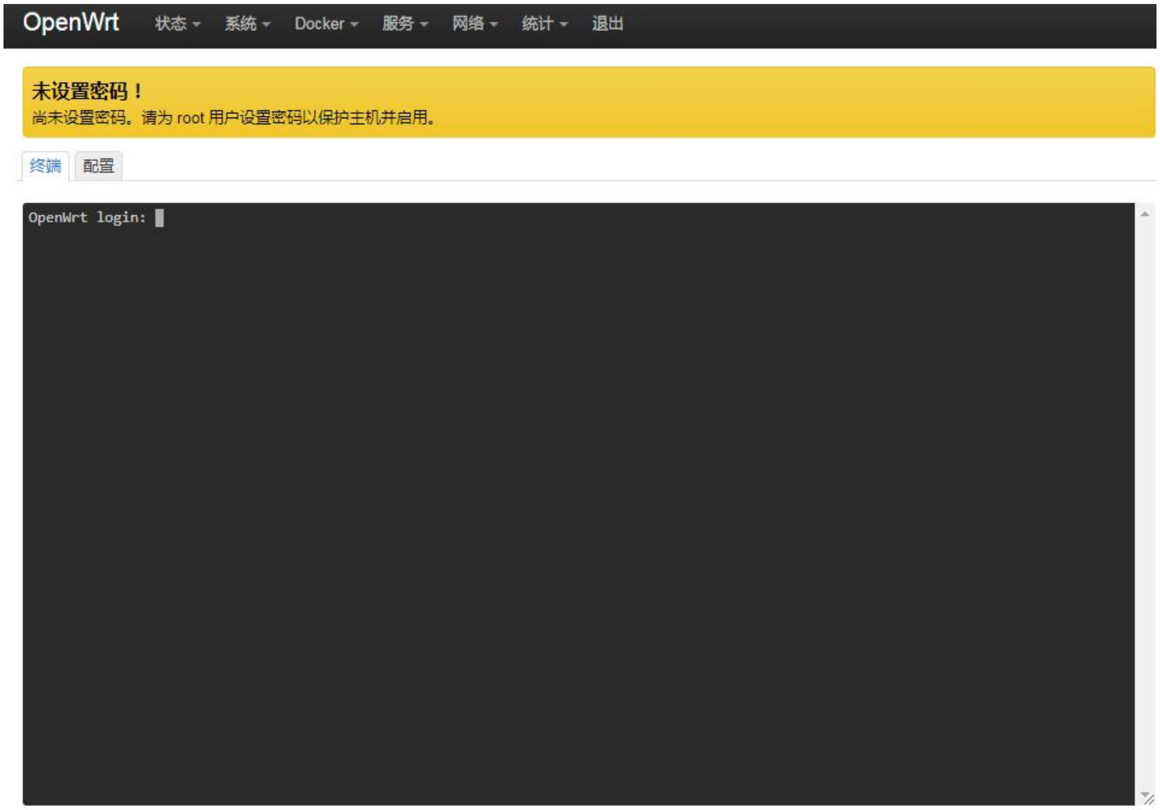
1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

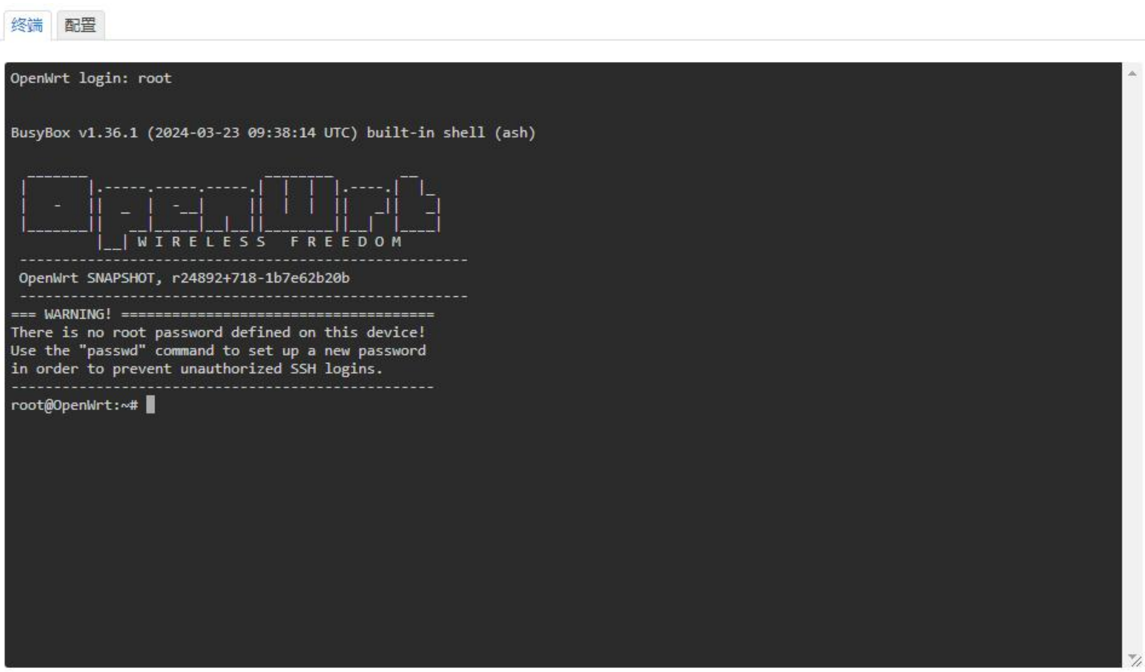
3) Enter the IP address **192.168.2.1** in the browser on your computer to log in to the LuCI interface

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 below



6) Enter the username root to log in



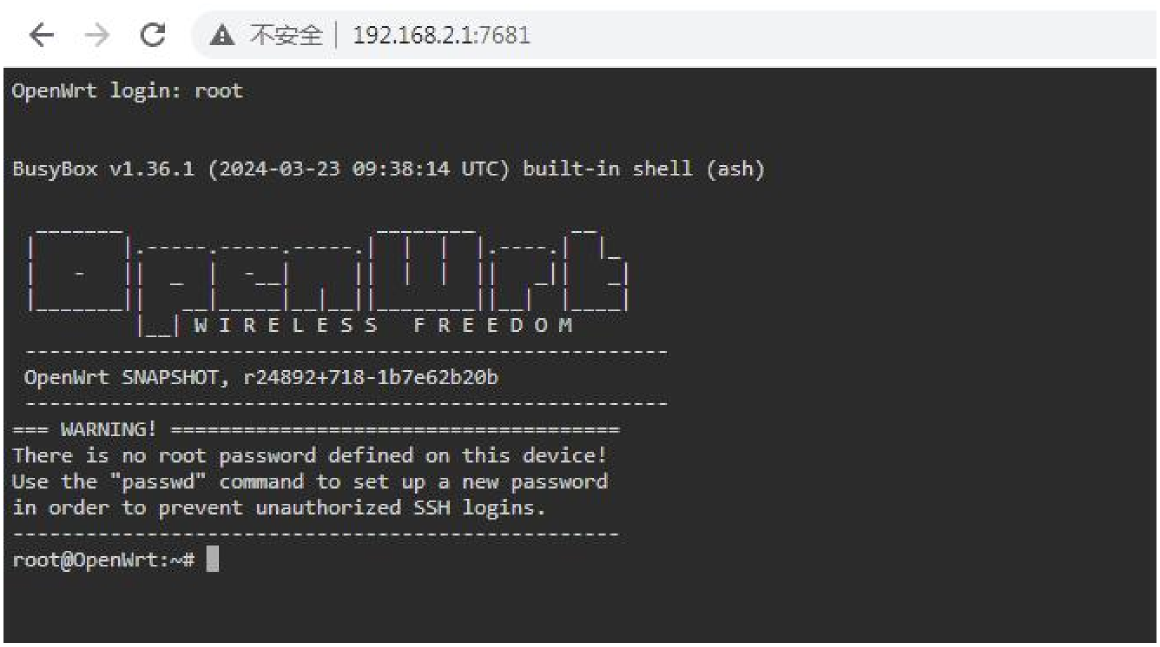
9. 4. 5. Log in to the terminal using IP address + port number.

Please note that in the OpenWrt system of Orange Pi 5 Pro, the network port is configured as 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 computer's network port through a network cable. At this time, the system After startup, an IP address will be assigned to the computer's network port through the DHCP service.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) Then enter **192.168.2.1:7681** in the browser to log in to the OpenWRT terminal



9. 5. How to modify the LAN port IP address 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 contents of the configuration file. For detailed instructions, please refer to the official documentation

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 **network.lan.ipaddr** item

```
root@OpenWrt:~# uci set network.lan.ipaddr='192.168.100.1'
```

4) Then enter the following command to complete the submission, that is, write it to the configuration file

```
root@OpenWrt:~# uci commit
```

If the IP address in red font is consistent with the one to be set, 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. For instructions on using ubus, please refer to the official documentation.

```
root@OpenWrt:~# ubus call network restart
```

6) At this time, enter the command and you can see that the IP of the LAN port is already **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)
```

9. 6. How to change the root password

9. 6. 1. Modification through command line

1) First enter passwd root on the system command line. 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) You will then be prompted to re-enter your password. At this time, enter your password again to confirm and press Enter.

```
Retype password:
```

3) The successful modification is displayed as follows

```
passwd: password for root changed by root
```

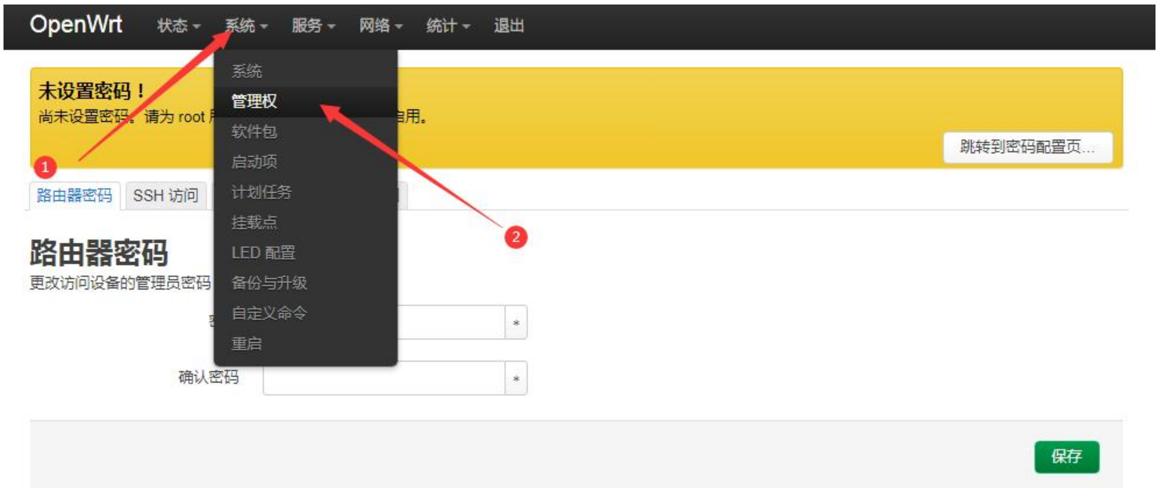
9. 6. 2. Modify through LuCI management interface

1) First refer to logging in to the LuCI management interface to enter the OpenWRT management interface.

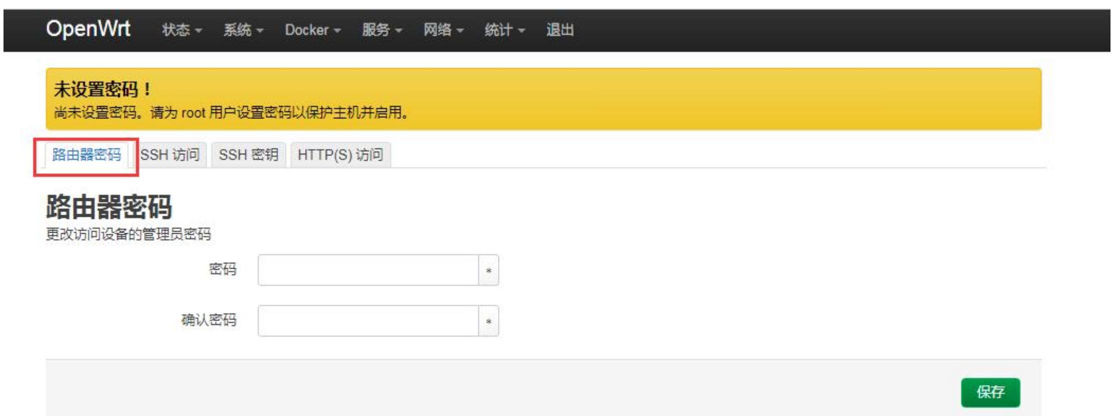
2) Then follow the steps below to change the password

   a. Find the **"System"** option in the navigation bar and click

   b. In the vertical column options below the system, select **"Management Rights"** and click



   c. Select the **"Router Password"** option on the Tab page



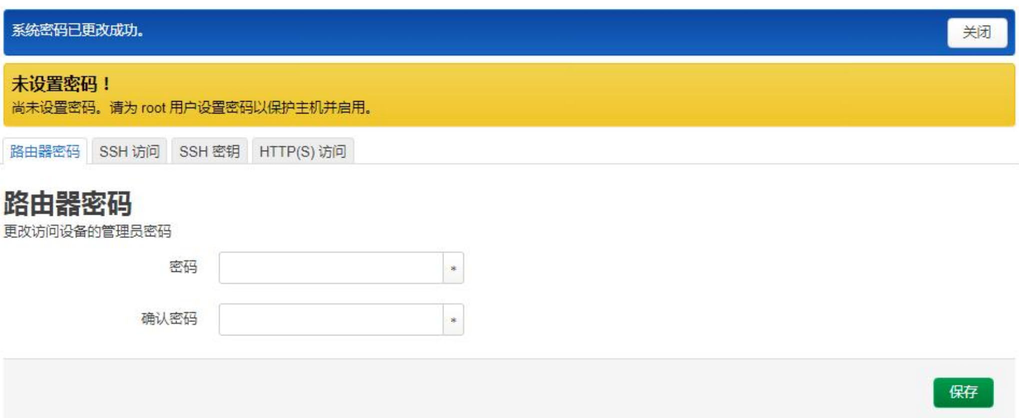
3) Modify and save the router password

   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 successfully changed, a pop-up box showing **"System password has been changed successfully"** will pop up. At this time, you will need a password to log in to OpenWRT.



9. 7. USB interface test

9. 7. 1. Mount USB storage device from command line

1) First insert the USB 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 the USB disk is successfully recognized.

```
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 check the capacity usage and mount point of the U disk through the df -h command.

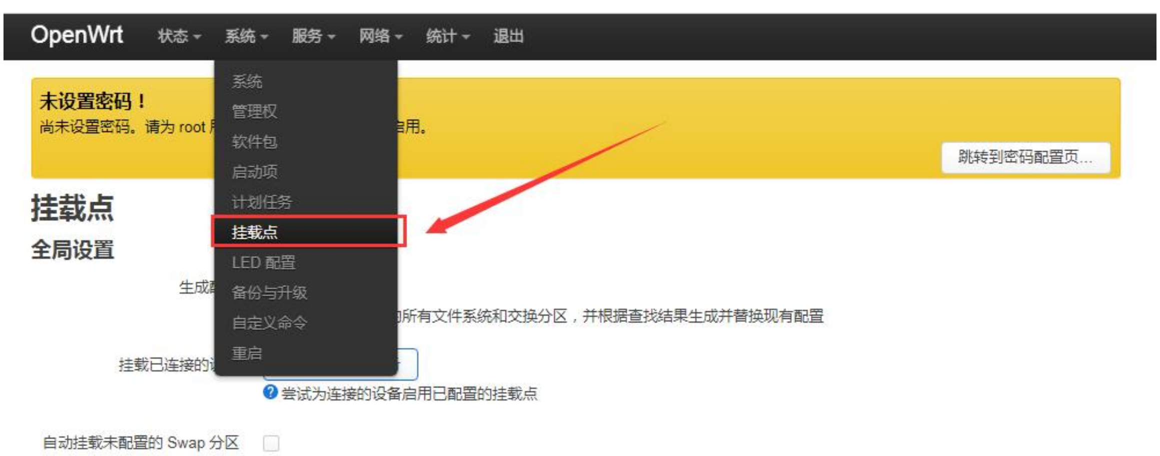
```
root@OpenWrt:~# df -h | grep "sd"
/dev/sda              14.4G       187.2M         14.2G      1% /mnt
```

9. 7. 2. Mount USB storage device in LuCI management interface

1) First connect the U disk (or other storage device) to the development board through USB2.0

2) Then log in to the LuCI management interface to enter the LuCI management interface.

3) Then in the LuCI management interface, click "System->Mount Point" to enter the mount point configuration interface



4) Then follow the steps below to add a mount point

a. Find **"Mount Point"** at the bottom of the mount point **global settings interface**.

b. Below the **mount point**, select the "Add" button and click to enter

c. The following pop-up window interface will pop up

d. Then you can start mounting the storage device

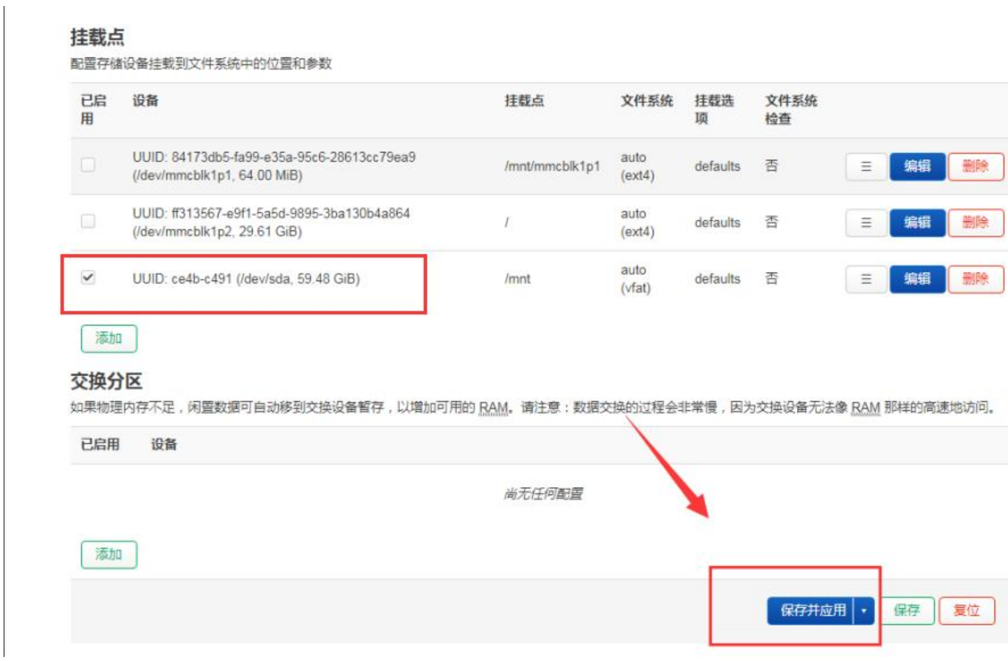
a) Check **"Enabled"**

b) Select the actual connected device /dev/sda in the General Settings UUID column (select according to your own device)

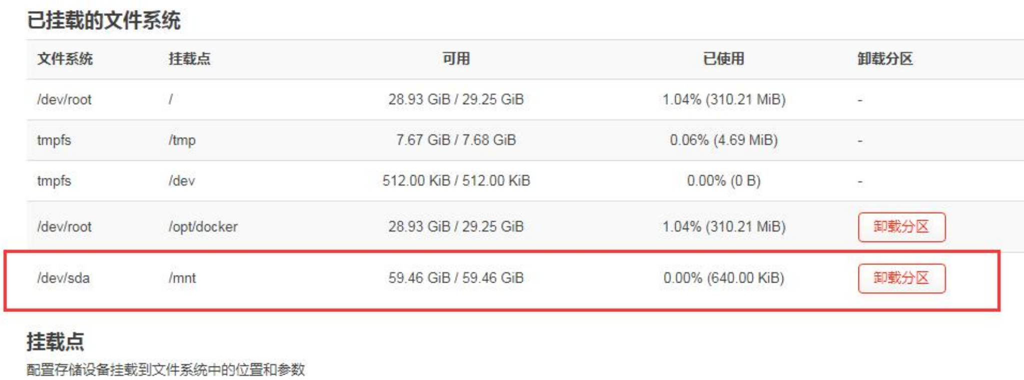
c) Select **"Custom"** in the mount point column and fill in the target directory to be mounted. Here, the **/mnt** directory is used as an example. After filling in, press **Enter** to confirm.

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 effective.



6) After saving, you can see that the storage device has been mounted successfully in **"Mounted File Systems"**


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

9. 8. USB to network port test

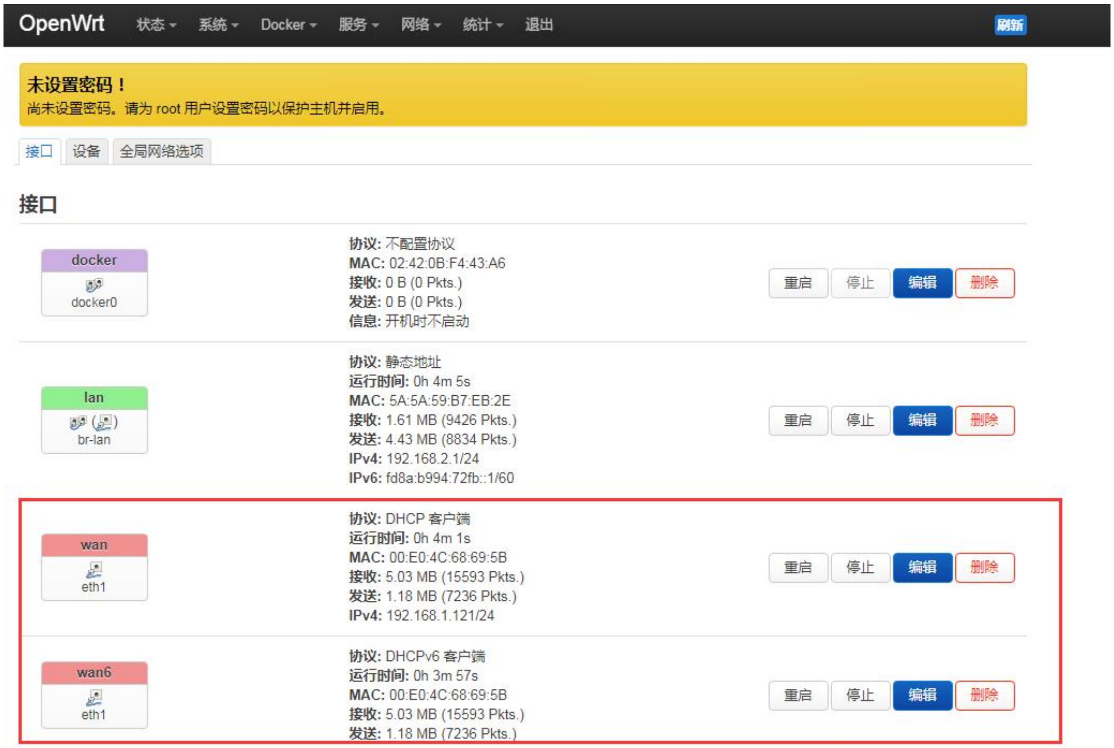
1）The usable USB network ports that have been tested so far are as follows:

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, 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 transfer port has been identified. As can be seen from the figure below, “eth1” is configured as a WAN port function by default.



5）At this time, after the USB to network port is connected to the main router through a network cable, the IP address can be automatically obtained through DHCP, and then the development board and the computer connected to the development board LAN port can access the Internet through the main router.

9. 9. AP6256 WIFI test

9. 9. 1. Method of creating WIFI hotspot

1）After the system starts, log in to the LUCI interface and click **Network -> Wireless** to enter the wireless WiFi configuration interface.

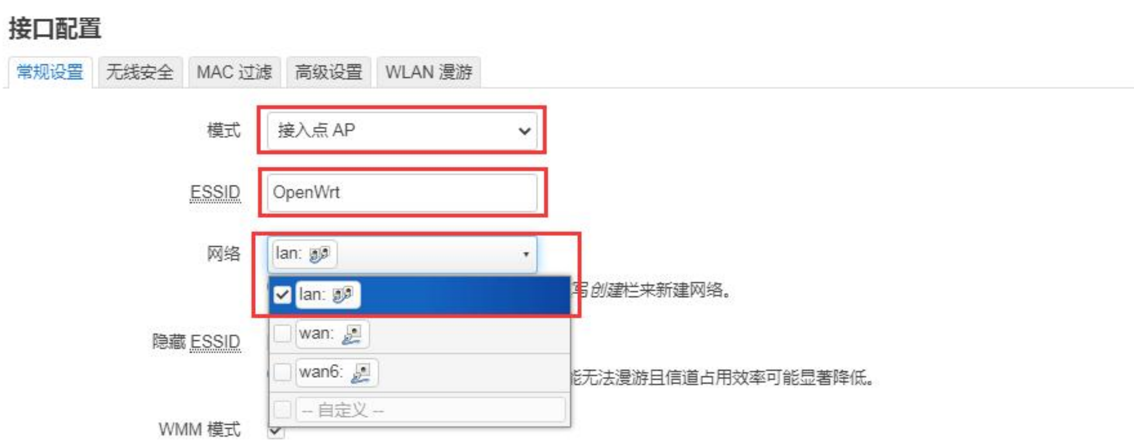
2）The default wireless configuration of the OpenWRT system is **Master** mode. To facilitate the next operation, we will remove the default wireless connection.

3）Then click **Save** in the lower right corner of the page to make the configuration take effect.

4）Then click the **Add** button on the right.

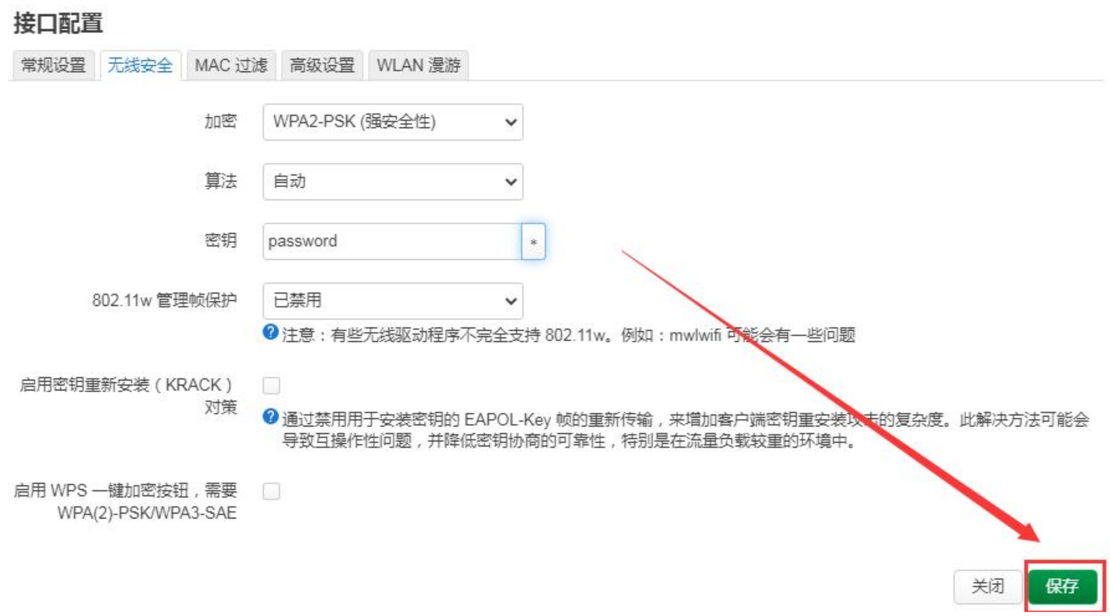
5）In the pop-up tab **"Device Configuration"**, we set the parameters as shown in the figure below.

6）Then in **Interface Configuration -> General Settings**, set the mode to **Access Point AP**, **ESSID** (wireless network name) to **OpenWrt**, and the network to **lan**



7）Then in **Interface Configuration -> Wireless Security**, select **WPA2-PSK** as the encryption algorithm; set the key (wireless password) to **password**

8) After the above settings are completed, click **Save** in the lower right corner of the page, and then exit the tab page



9) Then click **Save and Apply** in the lower right corner of the page and wait for the configuration to be applied.

10) The display interface for successfully creating a hotspot is as shown below

9. 9. 2. Method of connecting to WIFI hotspot

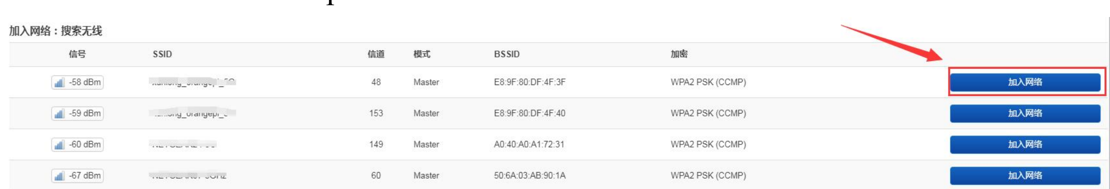
1）After the system starts, click **Network -> Wireless** to enter the wireless WiFi configuration interface.

2）The default wireless configuration of the OpenWRT system is **Master** mode. To facilitate the next operation, we will remove the default wireless connection.

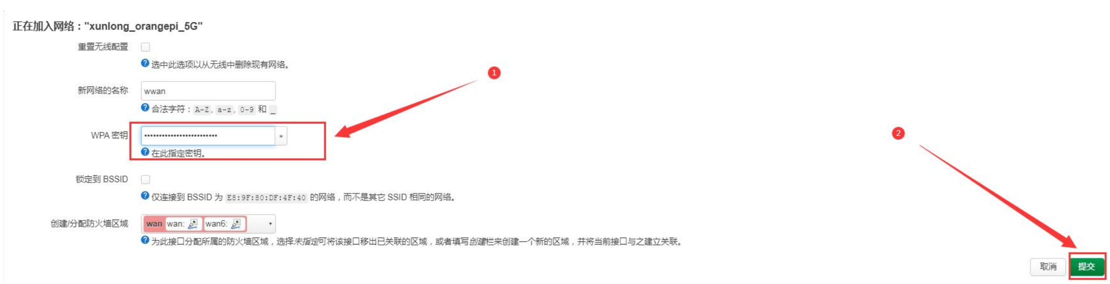
3）Then click **Save** in the lower right corner of the page to make the configuration take effect.

4）Then click the **Scan** button to scan the surrounding WiFi hotspots.

5）Then the following window will pop up to display the available WiFi hotspots. At this time, click the **Join Network** button to the right of the WiFi hotspot you want to connect to connect to the WiFi hotspot.



6）Then an interface for connecting to the WiFi hotspot will pop up. We enter the password of the hotspot as shown in the figure below, and then click the **Submit** button.



7）Then the following interface will pop up, click the **Save** button in the lower right corner.

8）Finally, you will return to the main interface of wireless configuration, click **Save and Apply** and wait for the configuration to be applied.

9）After successfully connecting to the WiFi hotspot, the interface is displayed as shown below.

9. 10. Installing packages via the command line

9. 10. 1. Install through opkg in the terminal

1) Update the list of available software packages

```
root@OpenWrt:/# opkg update
```

2) Get the software list

```
root@OpenWrt:/# opkg list
```

3) Install the specified software package

```
root@OpenWrt:/# opkg install <package name>
```

4) View installed software

```
root@OpenWrt:/# opkg list-installed
```

5) Uninstall the software

```
root@OpenWrt:/# opkg remove <package name>
```

9. 11. OpenWRT management interface installation package

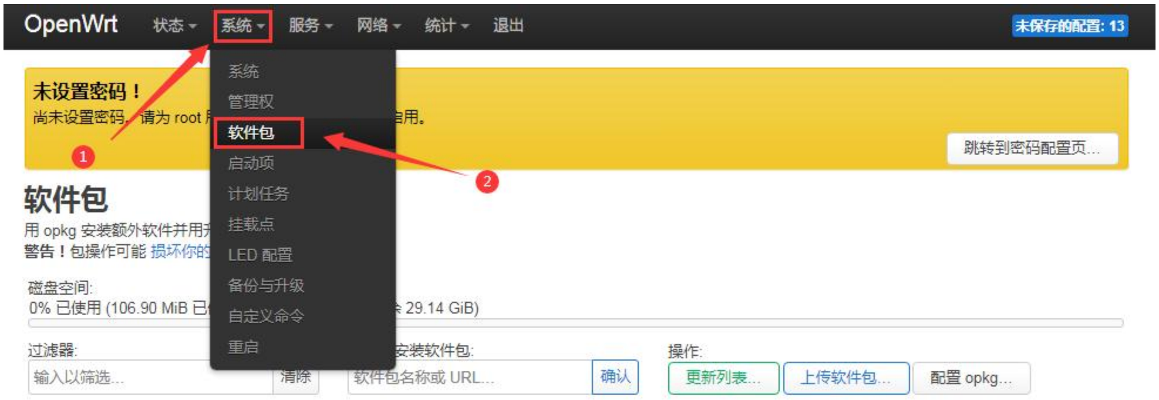
If you need to add a new software package, you can install it through the OpenWRT management interface.

9. 11. 1. View the list of available software packages on the system

1) First enter the software 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 **"Action"** option of the software package, click **"Update List"** to obtain the list of available software packages.

b. In the Tab page, click **"Available"** to view the currently available software packages.

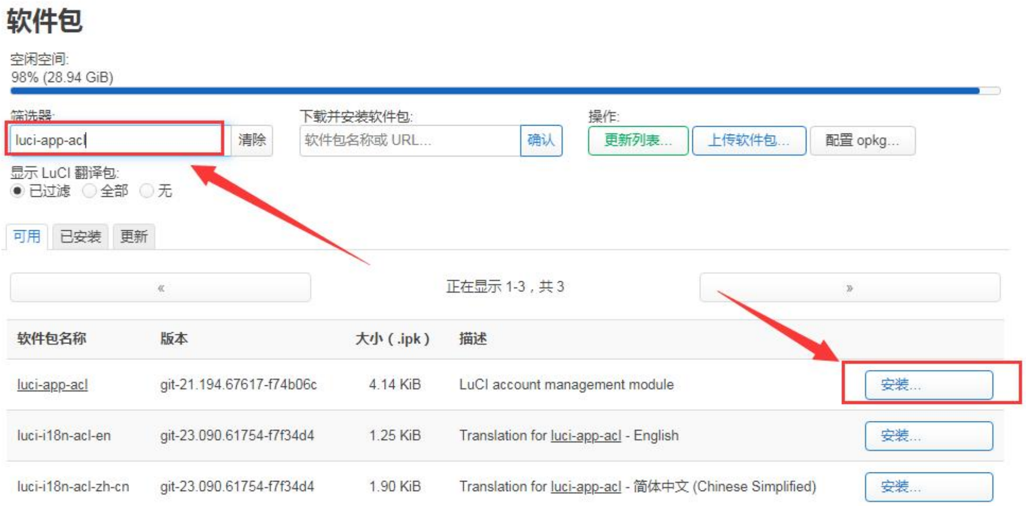
c. View the number of currently available software packages

9. 11. 2. Installation package example

1) Take the installation of the software package **"luci-app-acl"** as an example

a. In the OpenWRT 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 installation completion display 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 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**” software package will be displayed in the software package list, and the status will be updated to **"Installed"**

9.11.3. Remove package example

1) Take removing the software package "**luci-app-acl**" as an example

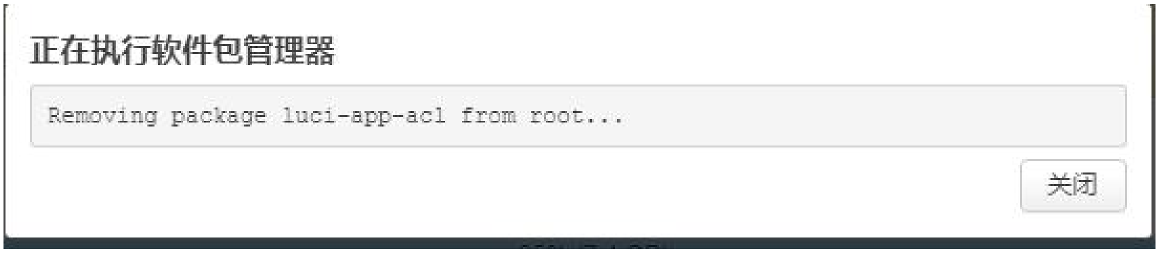
a. In the OpenWRT package management interface, click the filter dialog box and enter "**luci-app-acl**"

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

d.Then the pop-up window below will be displayed, click **"Remove"**

e.After successful removal, the display interface is as follows



2) Check whether the software package was successfully removed
   a. In the OpenWRT 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**” software package will not be displayed in the software package list. At this time, the “**luci-app-acl**” software package has been successfully removed.

9. 12. Using Samba network sharing

There are two main software options for OpenWRT LAN file sharing implementation, Samba and NFS. The Samba system has good compatibility, and NFS has superior performance. For users who need to use Windows devices, it is recommended to choose Samba.

1) Enter the management page of Samba network share
   a. Find the **"Service"** option in the navigation bar and click to enter

b. In the vertical bar 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 up a shared directory for network sharing

a. Click **"Add"** shared directory address in **"Shared Directory"** of **"General Settings"** of network sharing.

b. Enter the name of the shared folder as **“mmt”** under the name.

c. Under the path of the shared directory, select **“/mnt”** to set the shared directory location.

d. Check **"Browsable"** and **"Run anonymous user"**

e. Click **"Save and Apply"** to save the configuration

4) Windows 10 starts network discovery and sharing

> **Note: When accessing Samba and sharing under Windows 10 system, you need to first confirm whether Windows 10 has enabled network discovery and sharing. If not, perform the following settings first.**

a. Enable Samba v1/v2 access

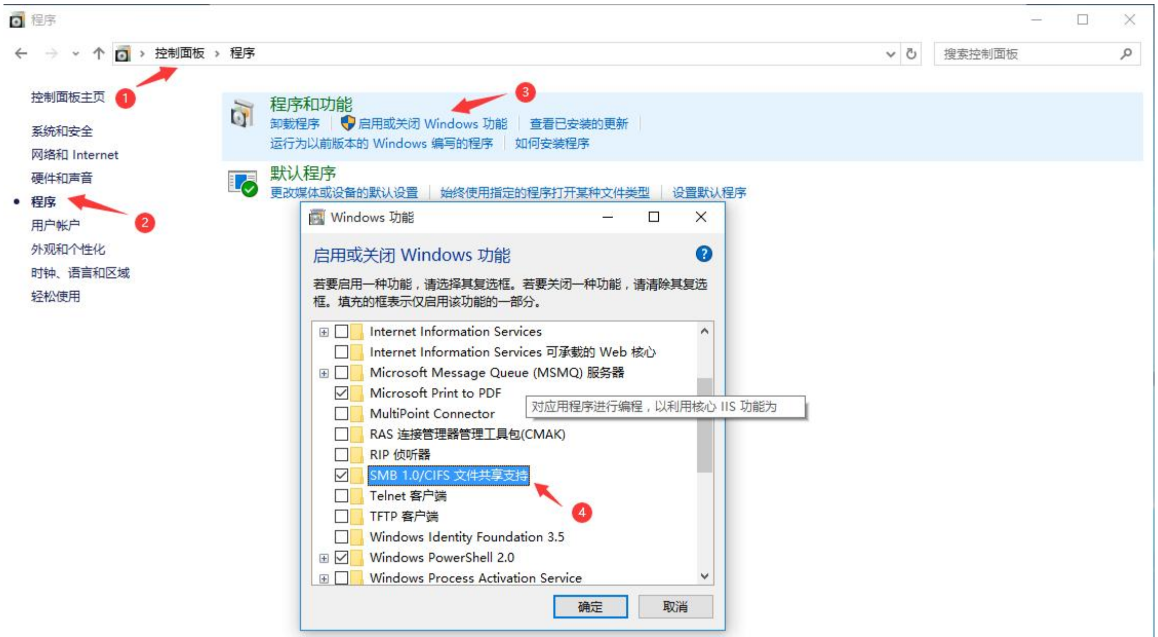
a) Enter the "Control Panel" of Windows 10

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 to enable or disable Windows features.

e) Click "OK" to configure the application



b. Turn on network discovery in Windows 10

a) Enter the "Control Panel" of Windows 10

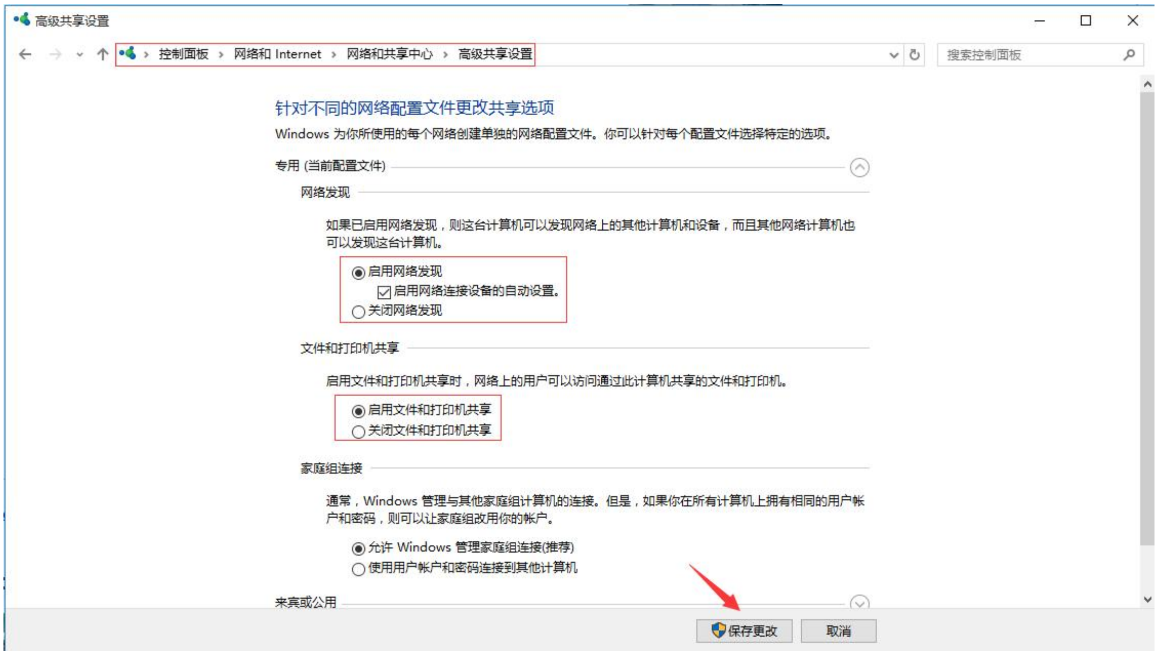
b) Select "Network and Internet" in the Control Panel

c) Then open "Network and Sharing Center"

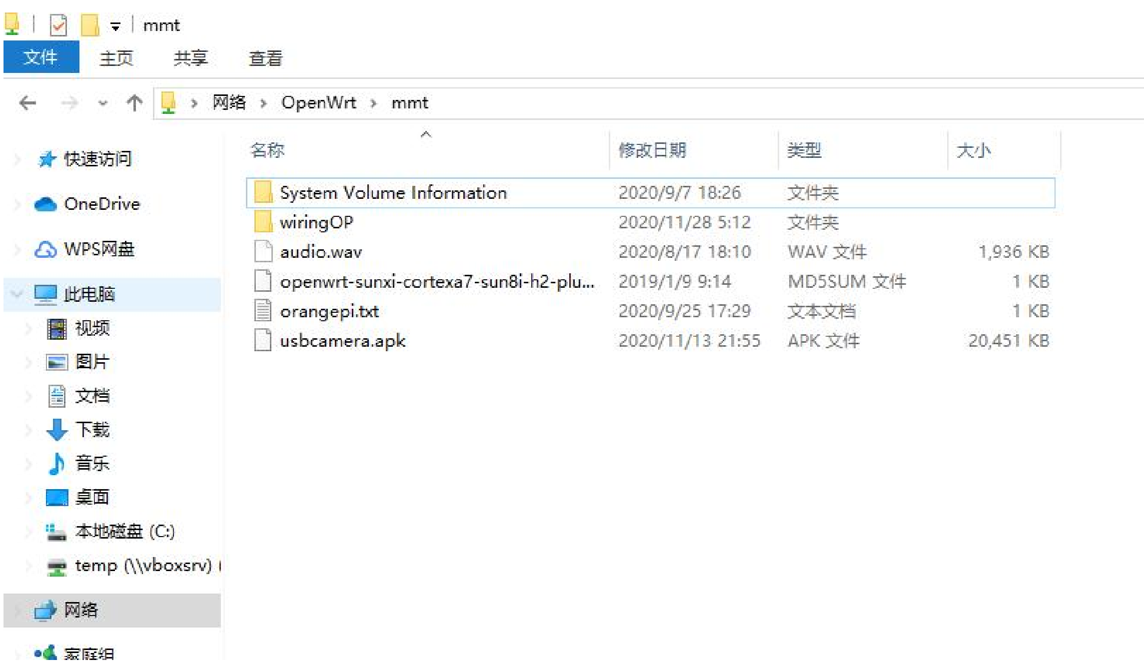
d) Click | "Advanced Sharing Settings"

e) Turn on **"Enable network discovery"** and **"Enable file and printer sharing"**

f) Click "Save Changes" to save the network discovery configuration of Windows 10



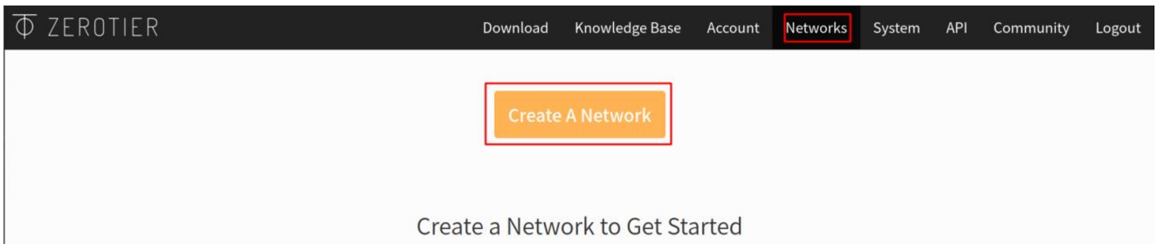
5) After the setting is completed, enter \\OpenWrt in the address bar of the resource manager to access the shared directory. The user name is root, and the password is the password set by the development board host.

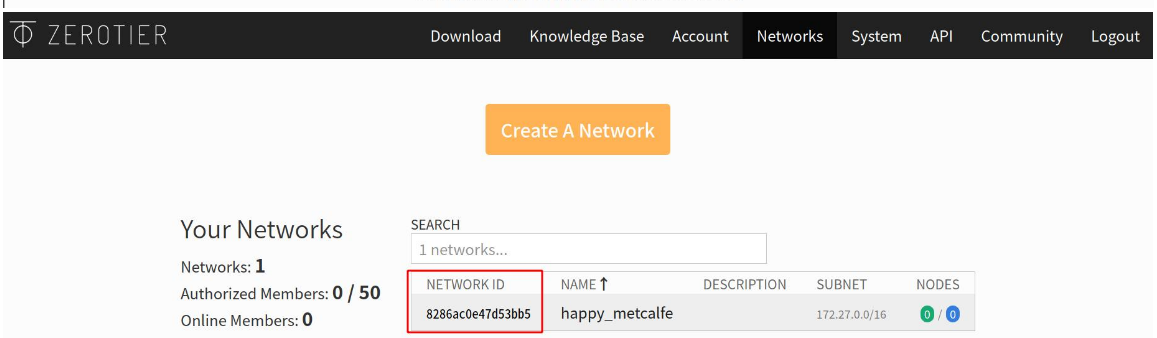


9. 13. zerotier instructions for use

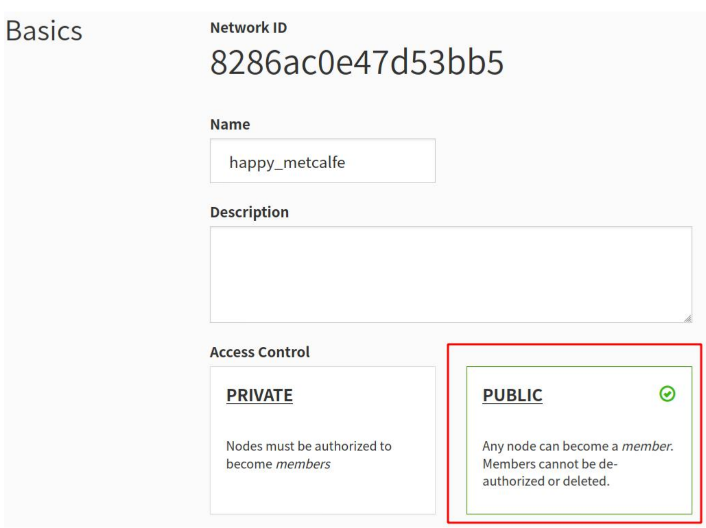
The OpenWRT system has been pre-installed with the zerotier client. After creating a virtual LAN on the zerotier official website, the client can directly join it through the Network ID. The specific operations are as follows.

1) Log in to zerotier official website **https://my.zerotier.com/network**, register and click Network->Create A Network to create a virtual LAN

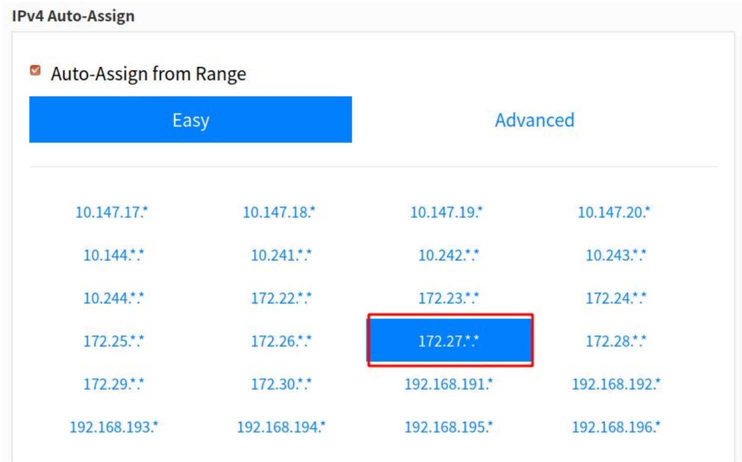




2) Click to enter the network console page and set the privacy option to public, so that the added network nodes do not need to be verified.



3) Next, you can choose the network segment for automatically assigning the address. The selected network segment 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 zerotier client
root@OpenWrt:/# zerotier-cli join 8286ac0e47d53bb5     #Join the network
```

5) Enter ifconfig in the terminal and you can see that there is 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 (Ubuntu 18.04 is used as an example here), execute the following command to install. After the installation is completed, you need to restart the computer.

```
test@ubuntu:~$ curl -s https://install.zerotier.com | sudo bash
```

7) After restarting, join the virtual LAN according to the Network ID. 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  (以太网)
        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) Zerotier and other common commands

```
root@OpenWrt:/# zerotier-one -d                     #Start 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     #Off the grid
root@OpenWrt:/# zerotier-cli listnetworks           #list networks
OPENWRT_DEVICE_REVISION="v0"
OPENWRT_RELEASE="OpenWrt 22.03.4 r20123-38ccc47687"
```

10. Compilation method of OpenWRT source code

10. 1. Download OpenWRT source code

1) First execute the following command to download the source 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 orangepi5pro
```

2) After the OpenWRT code is downloaded, it will contain the following files and folders

```
test@test:~/openwrt$ ls
BSDmakefile  Config.in  include  Makefile  README.md  scripts  toolchain
Config  feeds.conf.default  LICENSE  package  rules.mk  target  tools
```

10. 2. Compile OpenWRT source code

1) First install the following dependent software (currently only tested on Ubuntu22.04, you need to install the following software. If you compile on other versions of the system, please install the dependent software 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 dependency packages

```
test@test:~/openwrt$ ./scripts/feeds update -a
test@test:~/openwrt$ ./scripts/feeds install -a
```

3) Then select the profile using OrangePi 5 Pro

   a. Compile an image that supports TF card, eMMC, and NVMe boot, and select the following configuration

```
test@test:~/openwrt$ cp configs/orangepi-5-pro-rk3588_defconfig .config
```

   b. Compile and support SPI Flash boot image, select the following configuration

```
test@test:~/openwrt$ cp configs/orangepi-5-pro-rk3588-spi_defconfig .config
```

4) Then execute the following command to make the configuration take effect

```
test@test:~/openwrt$ make defconfig
```

5) Execute the following command to start compiling the openwrt source code

```
test@test:~/openwrt$ make V=s
```

6) After compilation is completed, 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-pro-ext4-sysupgrade.img.gz
├── openwrt-rockchip-armv8-xunlong_orangepi-5-pro.manifest
├── openwrt-rockchip-armv8-xunlong_orangepi-5-pro-squashfs-sysupgrade.img.gz
├── packages
├── profiles.json
├── sha256sums
└── version.buildinfo

1 directory, 9 files
```

11. Appendix

11. 1. User manual update history

Version	Date	Release Notes
v1.0	2024-04-07	initial version
v1.1	2024-05-16	Method of burning Linux image to TF card+NVMe SSD
v1.2	2024-07-04	1. Linux：Usage of NPU 2. Linux：RK3588 method of using Baidu Feijiang 3. Linux：Method for Running RKLLM Large Model with RK3588
v1.3	2024-07-30	1. The usage method of wiringOP hardware

11. 2. Image update history

Date	Release Notes
2024-04-07	Orangepi5pro_1.0.0_ubuntu_focal_server_linux5.10.160.7z Orangepi5pro_1.0.0_ubuntu_jammy_server_linux5.10.160.7z Orangepi5pro_1.0.0_debian_bullseye_server_linux5.10.160.7z Orangepi5pro_1.0.0_debian_bookworm_server_linux5.10.160.7z Orangepi5pro_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.0_debian_bookworm_desktop_xfce_linux5.10.160.7z OrangePi5Pro_RK3588S_Android12_v1.0.0.tar.gz OrangePi5Pro_RK3588S_Android12_lcd_v1.0.0.tar.gz OrangePi5Pro_RK3588S_Android12_spi-sata_v1.0.0.tar.gz OrangePi5Pro_RK3588S_Android12_spi-nvme_v1.0.0.tar.gz OrangePi5Pro_RK3588S_Android12_lcd_spi-sata_v1.0.0.tar.gz OrangePi5Pro_RK3588S_Android12_lcd_spi-nvme_v1.0.0.tar.gz

	OrangePi5Pro_RK3588S_Android12_box_v1.0.0.tar.gz OrangePi5Pro_RK3588S_Android12_box_spi-nvme_v1.0.0.tar.gz OrangePi5Pro_RK3588S_Android12_box_spi-sata_v1.0.0.tar.gz Opios-droid-aarch64-opi5pro-24.02-linux5.10.160.tar.gz Opios-droid-aarch64-opi5pro-24.02-linux5.10.160-spi-nvme.tar.gz Opios-droid-aarch64-opi5pro-24.02-linux5.10.160-spi-sata.tar.gz Opios-arch-aarch64-gnome-opi5pro-24.03-linux5.10.160.img.xz openwrt-aarch64-opi5pro-24.03-linux-6.1.43-ext4.img.gz openwrt-aarch64-opi5pro-24.03-linux-6.1.43-spi-squashfs.bin * initial version
2024-05-09	OrangePi5Pro_RK3588S_Android12_v1.0.1.tar.gz OrangePi5Pro_RK3588S_Android12_box_v1.0.1.tar.gz OrangePi5Pro_RK3588S_Android12_spi-sata_v1.0.1.tar.gz OrangePi5Pro_RK3588S_Android12_spi-nvme_v1.0.1.tar.gz OrangePi5Pro_RK3588S_Android12_box_spi-nvme_v1.0.1.tar.gz OrangePi5Pro_RK3588S_Android12_box_spi-sata_v1.0.1.tar.gz * Update bl31, optimize 4GB/8GB memory stability * Add the function of uboot to detect LCD connection. When the LCD is connected, turn on the LCD display Opios-droid-aarch64-opi5pro-24.05-linux5.10.160.tar.gz Opios-droid-aarch64-opi5pro-24.05-linux5.10.160-spi-nvme.tar.gz Opios-droid-aarch64-opi5pro-24.05-linux5.10.160-spi-sata.tar.gz Opios-arch-aarch64-gnome-opi5pro-24.05-linux5.10.160.img.xz * Update bl31, optimize 4GB/8GB memory stability

	Orangepi5pro_1.0.2_ubuntu_focal_server_linux5.10.160.7z Orangepi5pro_1.0.2_ubuntu_jammy_server_linux5.10.160.7z Orangepi5pro_1.0.2_debian_bullseye_server_linux5.10.160.7z Orangepi5pro_1.0.2_debian_bookworm_server_linux5.10.160.7z Orangepi5pro_1.0.2_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.2_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.2_debian_bookworm_desktop_xfce_linux5.10.160.7z * Update bl31, optimize 4GB/8GB memory stability * Update the yt6801 network port chip driver
2024-07-04	Orangepi5pro_1.0.4_ubuntu_jammy_server_linux6.1.43.7z Orangepi5pro_1.0.4_debian_bullseye_server_linux6.1.43.7z Orangepi5pro_1.0.4_debian_bookworm_server_linux6.1.43.7z Orangepi5pro_1.0.4_ubuntu_jammy_desktop_xfce_linux6.1.43.7z Orangepi5pro_1.0.4_debian_bullseye_desktop_xfce_linux6.1.43.7z Orangepi5pro_1.0.4_debian_bookworm_desktop_xfce_linux6.1.43.7z Orangepi5pro_1.0.4_ubuntu_focal_server_linux5.10.160.7z Orangepi5pro_1.0.4_ubuntu_jammy_server_linux5.10.160.7z Orangepi5pro_1.0.4_debian_bullseye_server_linux5.10.160.7z Orangepi5pro_1.0.4_debian_bookworm_server_linux5.10.160.7z Orangepi5pro_1.0.4_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.4_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.4_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.4_debian_bookworm_desktop_xfce_linux5.10.160.7z * Upgrade the rknpu version of the kernel to 0.9.6
2024-11-18	Orangepi5pro_1.0.6_ubuntu_jammy_server_linux6.1.43.7z Orangepi5pro_1.0.6_debian_bullseye_server_linux6.1.43.7z Orangepi5pro_1.0.6_debian_bookworm_server_linux6.1.43.7z Orangepi5pro_1.0.6_ubuntu_jammy_desktop_xfce_linux6.1.43.7z Orangepi5pro_1.0.6_debian_bullseye_desktop_xfce_linux6.1.43.7z Orangepi5pro_1.0.6_debian_bookworm_desktop_xfce_linux6.1.43.7z

	Orangepi5pro_1.0.6_ubuntu_focal_server_linux5.10.160.7z Orangepi5pro_1.0.6_ubuntu_jammy_server_linux5.10.160.7z Orangepi5pro_1.0.6_debian_bullseye_server_linux5.10.160.7z Orangepi5pro_1.0.6_debian_bookworm_server_linux5.10.160.7z Orangepi5pro_1.0.6_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.6_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.6_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi5pro_1.0.6_debian_bookworm_desktop_xfce_linux5.10.160.7z * Fix the issue where the camera does not have 3A effect when Docker is turned on * Fix the issue of Bluetooth MAC address changing during restart