Orange Pi 5 Max
User Manual

Catalogue

1. Basic features of Orange Pi 5 Max

1. 1. What is Orange Pi 5 Max

The Orange Pi 5 Max adopts the new generation of eight core 64 bit ARM processors from Ruixin Micro RK3588, specifically the quad core A76 and quad core A55, using Samsung's 8nm LP process technology. The maximum clock speed of the large core can reach 2.4GHz, integrated with ARM Mali-G610 MP4 GPU, embedded with high-performance 3D and 2D image acceleration modules, and built-in AI accelerator NPU with up to 6 Tops computing power. It has 4GB/8GB/16GB (LPDDR5) memory and up to 8K display processing capability.

The Orange Pi 5 Max has introduced a wide range of interfaces, including HDMI output, Wi Fi 6, M.2 M-key PCIe 3.0x4, 2.5G Ethernet ports, USB 2.0, USB 3.1 interfaces, and 40 pin extension 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 Max supports Orange Pi OS, the official operating system developed by Orange Pi, as well as Android 13, Debian11, Debian12, Ubuntu 20.04, and Ubuntu 22.04 operating systems.

1. 2. The purpose of Orange Pi 5 Max

We can use it to achieve:

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

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

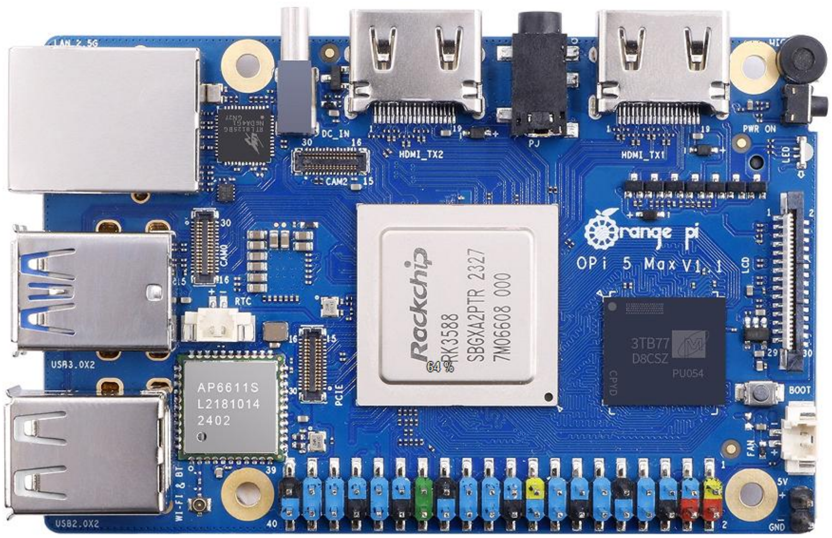
1. 3. Hardware Features of Orange Pi 5 Max

Introduction to Hardware Features	
CPU	• Rockchip RK3588 (8nm LP process） • 8-core 64 bit processor • 4-core Cortex-A76 and 4-core Cortex-A55 typical small and large core architectures • The maximum main frequency of the large core is 2.4GHz, and the maximum main frequency of the small core is 1.8GHz
GPU	• Integrated ARM Mali-G610 • OpenGL ES1.1/2.0/3.2、OpenCL 2.2 和 Vulkan 1.2
NPU	• Built in AI accelerator NPU with up to 6 Tops computing power • Support INT4/INT8/INT16 mixed operations
Video output	• 2 * HDMI 2.1，Maximum support for 8K @60Hz • 1 * MIPI D-PHY TX 4Lane
Memory	4GB/8GB/16GB（LPDDR5）
Camera	• 2 * MIPI CSI 4Lane • 1 * MIPI D-PHY RX 4Lane
PMU	RK806-1
Onboard storage	• eMMC Socket，Can be connected to an external eMMC module • 16MB QSPI Nor FLASH • MicroSD (TF) Card slot • PCIe3.0x4 M.2 M-KEY (SSD) slot
Ethernet	1 * PCIe 2.5G Ethernet port（RTL8125BG)
Audio frequency	• 3.5mm headphone jack audio input/output • Onboard MIC input • 2 * HDMI output

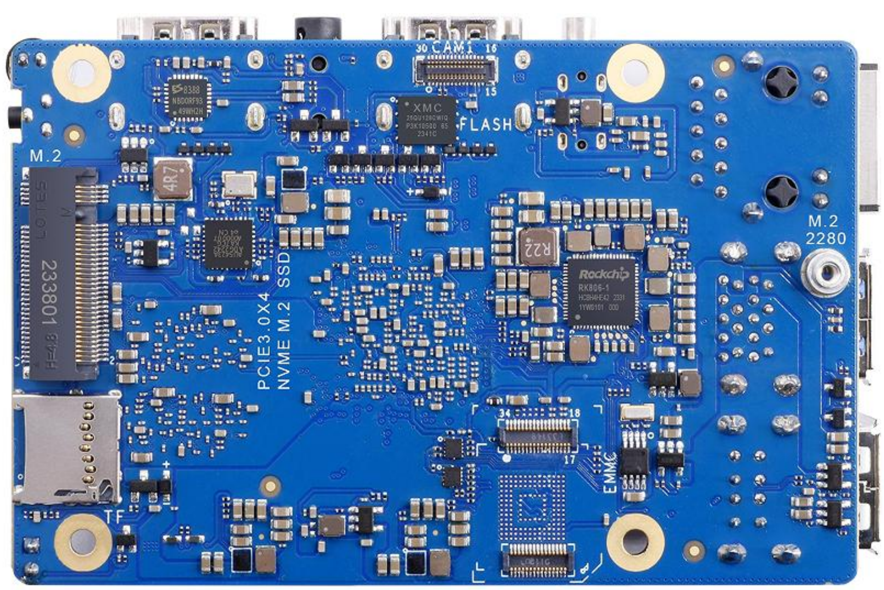
PCIe M.2 M-KEY	• PCIe 3.0 x 4 lanes，Used for connecting NVMe SSD solid state drives
USB Interface	• 1 * USB 3.0 supports Device or HOST mode • 1 * USB3.0 HOST • 2 * USB2.0 HOST
40pin Expand needle arrangement	Used for expanding UART, PWM, I2C, SPI, CAN, and GPIO interfaces
Debug UART	Included in the 40PIN expansion port
LED lights	RGB LED Tri color indicator light
Key	1 * MaskROM key, 1 * power on/off key
Power supply	Type-C interface power supply 5V/5A
Supported operating systems	Orange Pi OS（Droid）、Orange Pi OS（Arch）、Android13、Debian11、Debian12、Ubuntu20.04 and Ubuntu22.04 etc.
Introduction to appearance specifications	
Product size	89mm*56mm
Weight	58g
rangePi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd	

1.4. Top and Bottom Views of Orange Pi 5 Max

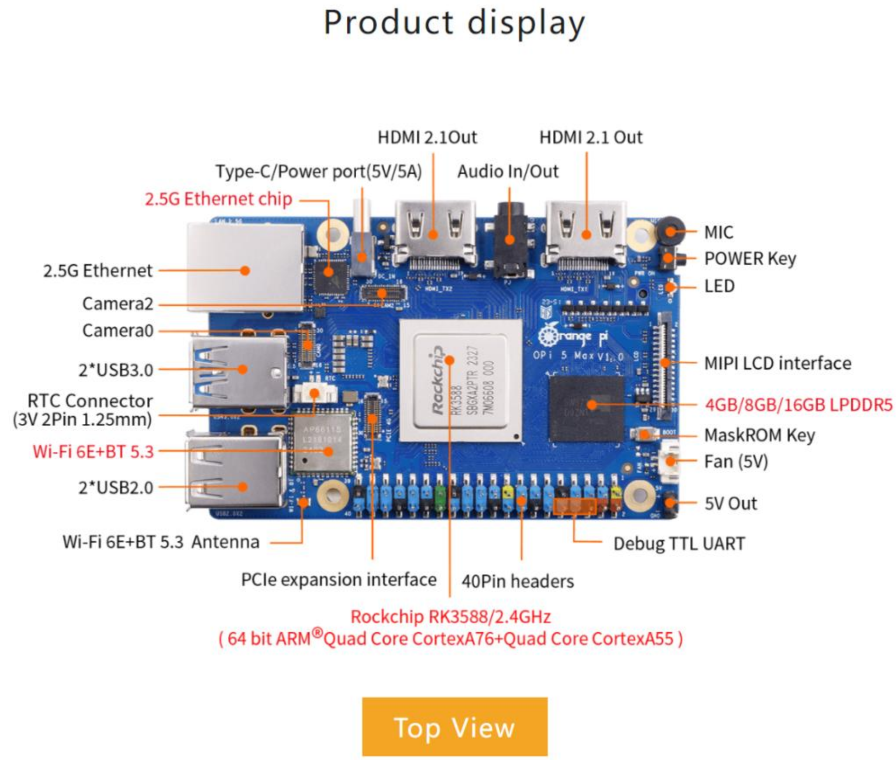
Top view:

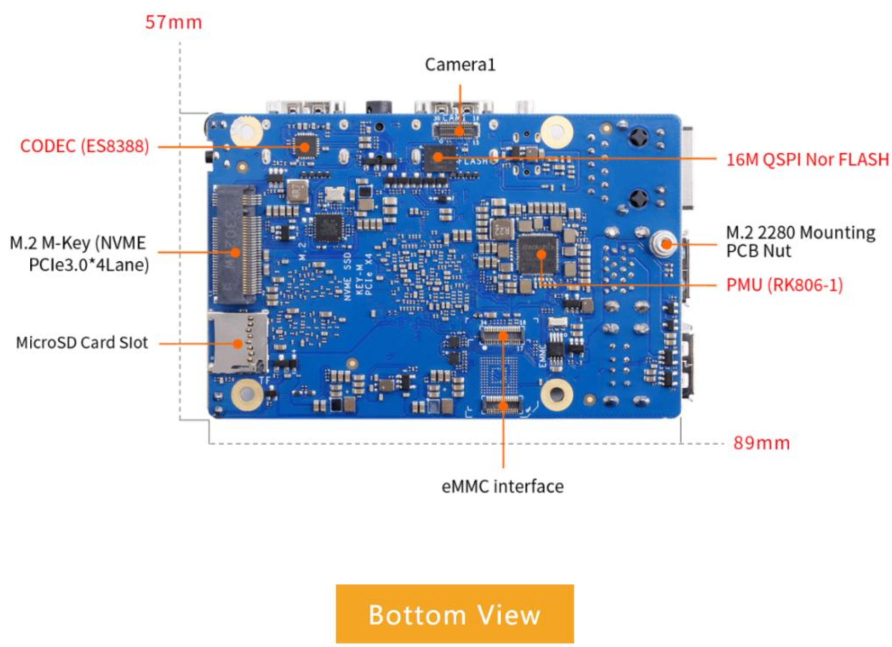


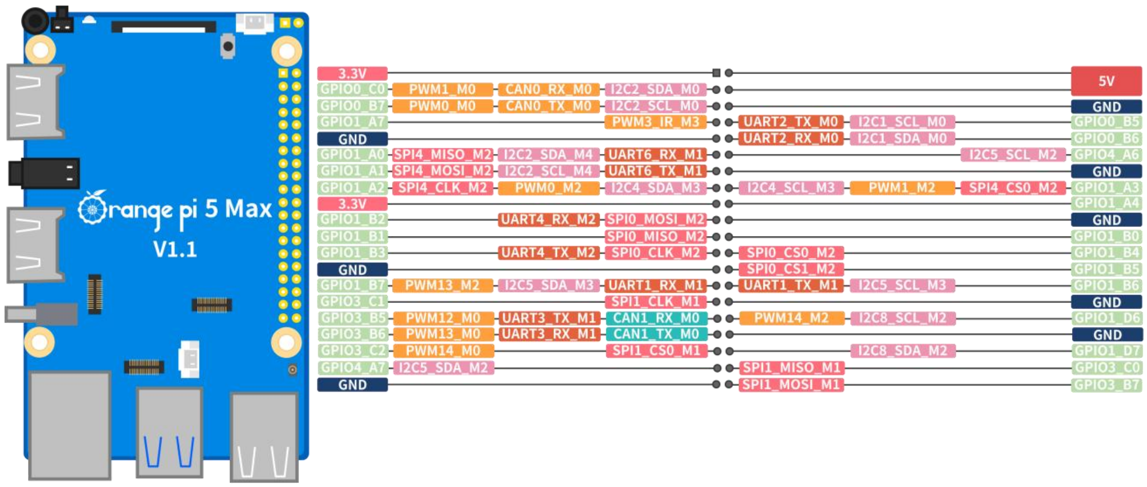
Bottom view:



1.5. Interface Details of Orange Pi 5 Max







The diameter of the four positioning holes is 2.7mm.

2. Introduction to using the development board

2. 1. Prepare the necessary accessories

1) TF card, a high-speed flash card with a minimum capacity of 16GB (recommended 32GB or above) and a class 10 or above.

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

3) Display with HDMI interface.

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 the adapter board and OPi5Plus/OPi5B/OPi5/OPi5Pro/OPi5Max universal)

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

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

7) USB interface mouse and keyboard, any standard USB interface mouse and keyboard

can be used to control the Orange Pi development board

8) USB camera

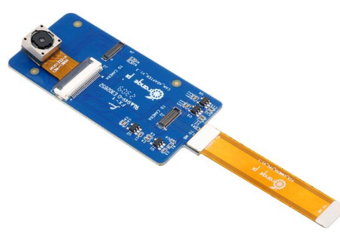
9) 5V cooling fan. As shown in the figure below, there is an interface on the development board for connecting the cooling fan, with a specification of **2pin 1.25mm spacing**

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

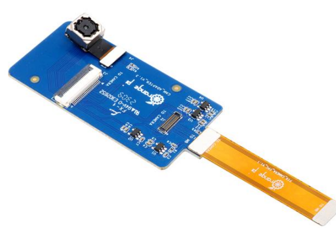
10) 100Mbps or 1G Ethernet cable, used to connect the development board to the Internet

11) USB 2.0 male to male 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

15) Personal computers with Ubuntu and Windows operating systems installed

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

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

1) The download link for the English version of the material is:

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-5-Max.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. Method of burning Linux image to TF card based on Windows PC

Note that the Linux image referred to here specifically refers to Linux distribution images such as Debian, Ubuntu, OpenWRT, or OPi OS Arch

downloaded from the Orange Pi download page.

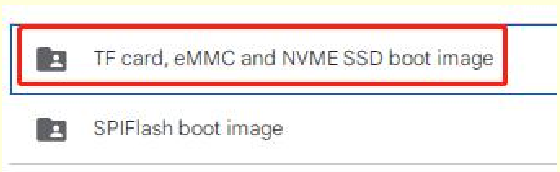
2. 3. 1. How to use balenaEtcher to burn Linux

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

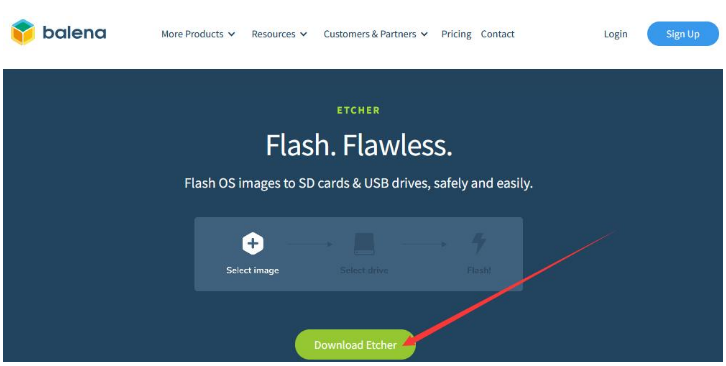
Note that if you are downloading an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the image file in the "TF Card, eMMC, and NVME SSD Boot Image" folder.



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

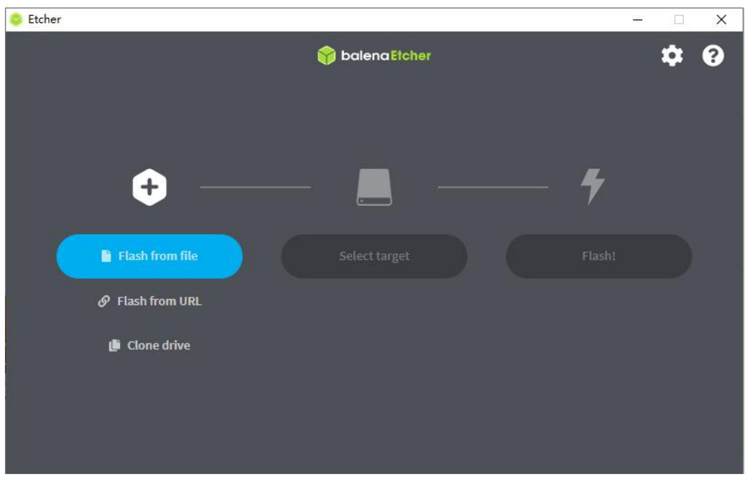
https://www.balena.io/etcher/

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

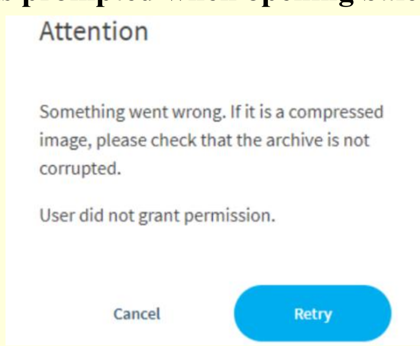


6) Then you can choose to download the Portable version of BalenaEtcher software. The Portable version does not need to be installed, and can be opened by double clicking to use it

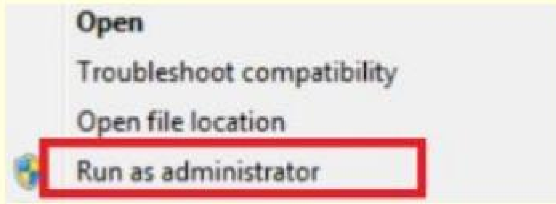
7) If you are downloading a version of BalenaEtcher that requires installation, please install it before using it. If you download the Portable version of balenaEtcher, simply double-click to open it. The interface of balenaEtcher after opening is shown in the following figure:



If the following error is prompted when opening balenaEtcher:



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

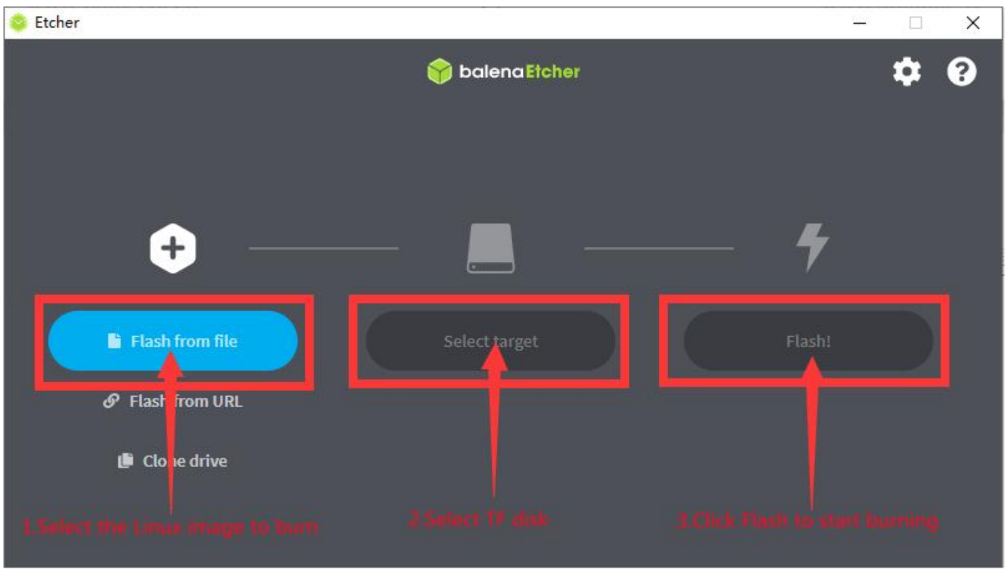


8) The specific steps to use balenaEtcher to burn the Linux image are aa follow

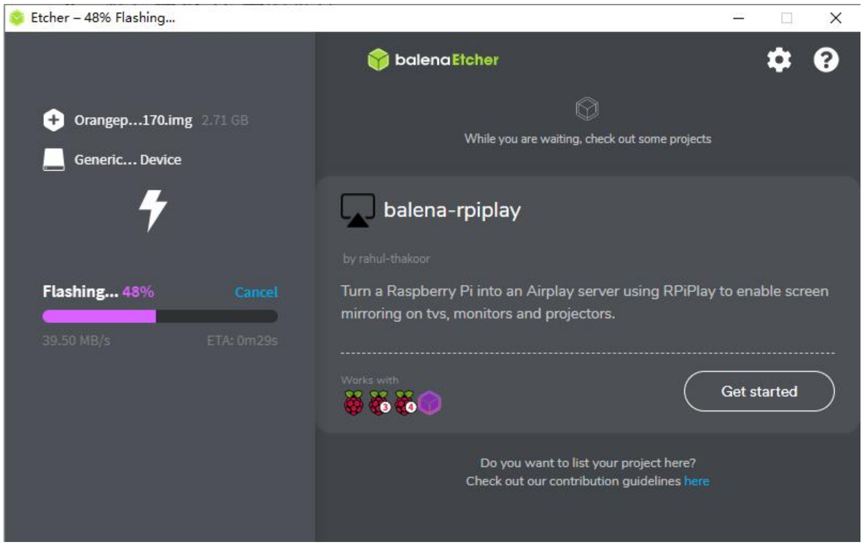
a. First select the path of the Linux image file to burned

b. Then select the drive letter of the TF card

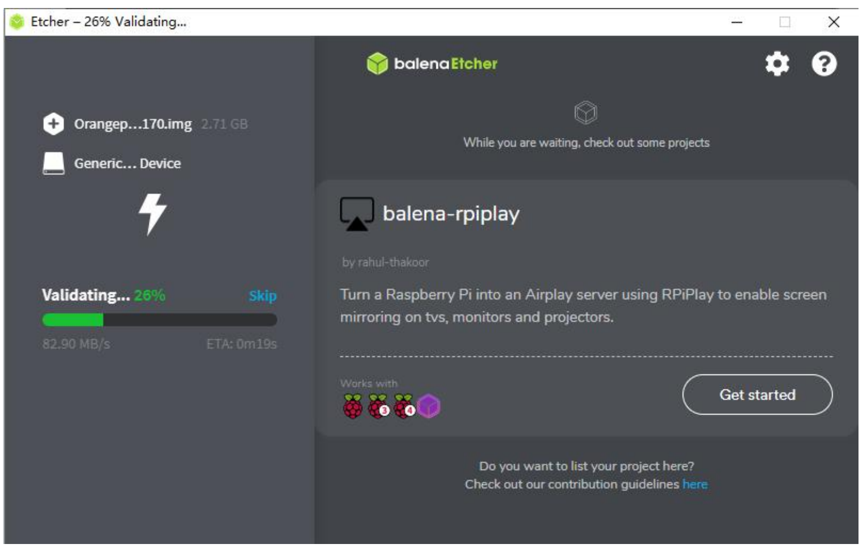
c. Finally, click Flash to start burning the Linux image to the TF card



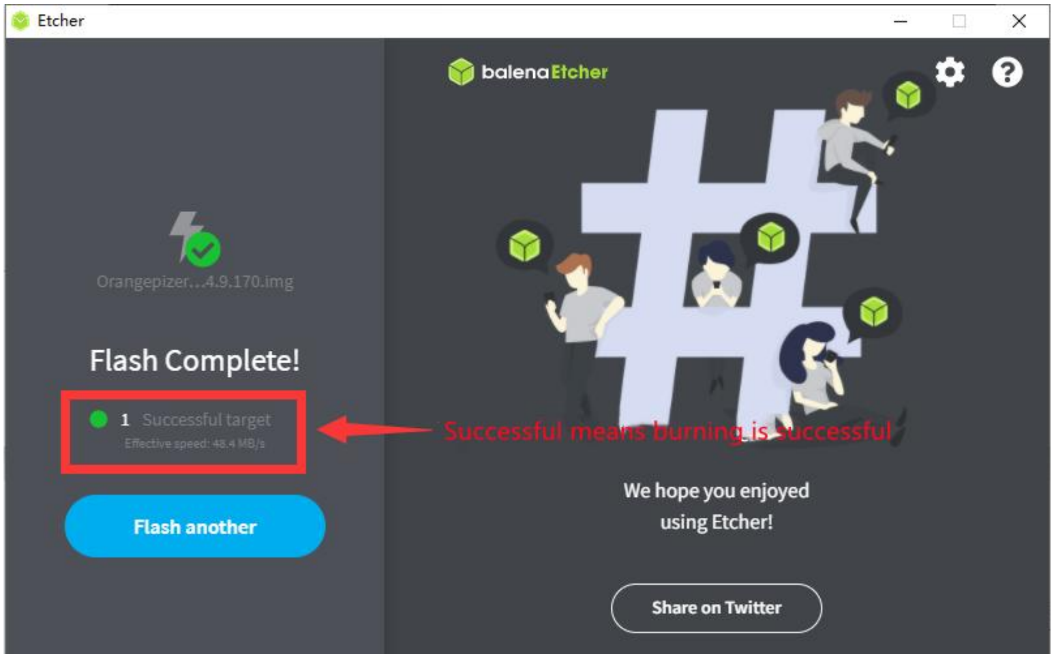
9) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card



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

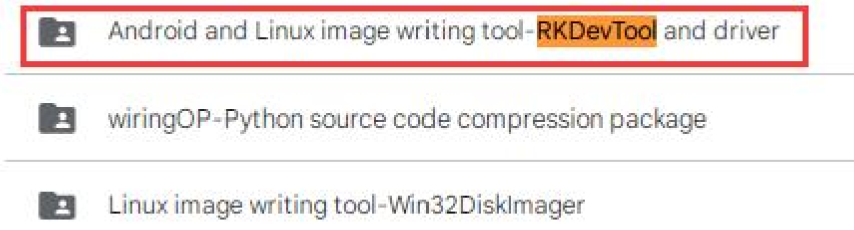
1) Please select balenaEtcher, right-click, and then select Run as administrator.

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

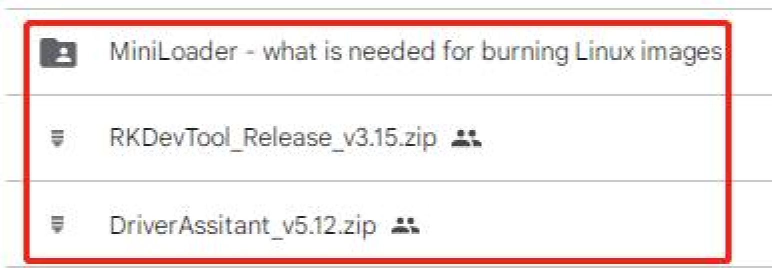
3) Then insert the TF card into the card slot of the development board

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



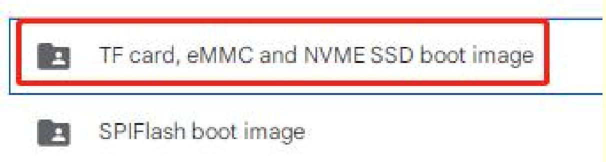
b. Then download all the files below



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

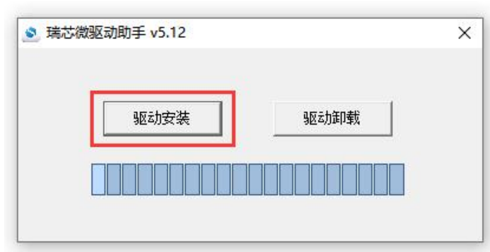
Note that if you are downloading an OpenWRT image, you will see the following three types of images in the download link of the OpenWRT image. Please select the image file in the "TF Card, eMMC, and NVME SSD Boot Image" folder.



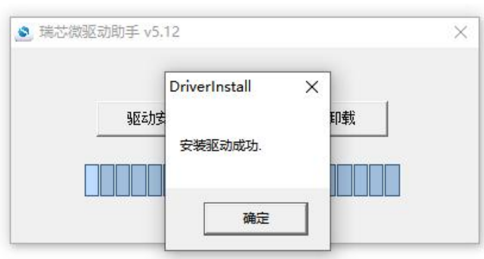
6) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**. zip, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

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.



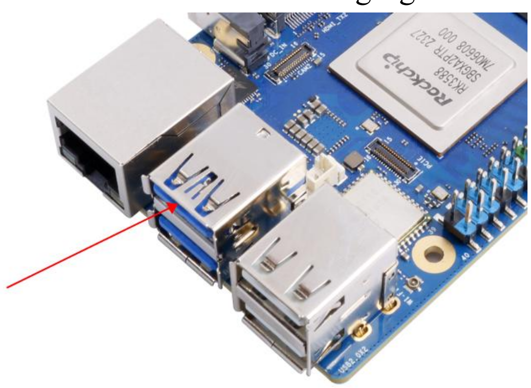
8) Then decompress **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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**"

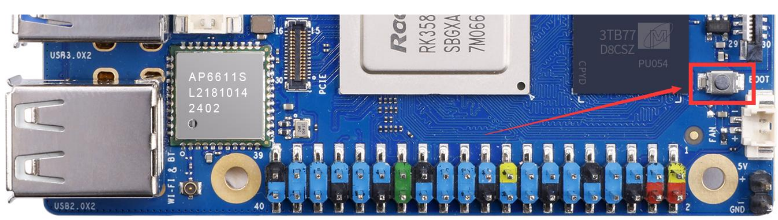
10) Then start burning the Linux image to the TF card

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

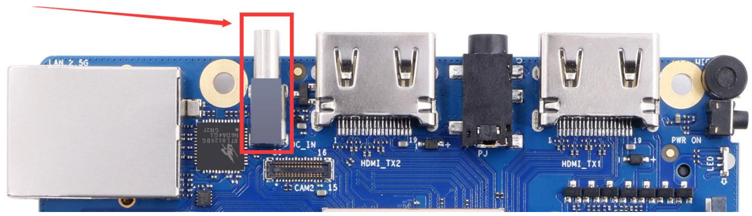


b. Then insert the TF card into the development board and ensure that the board is not connected to a power source

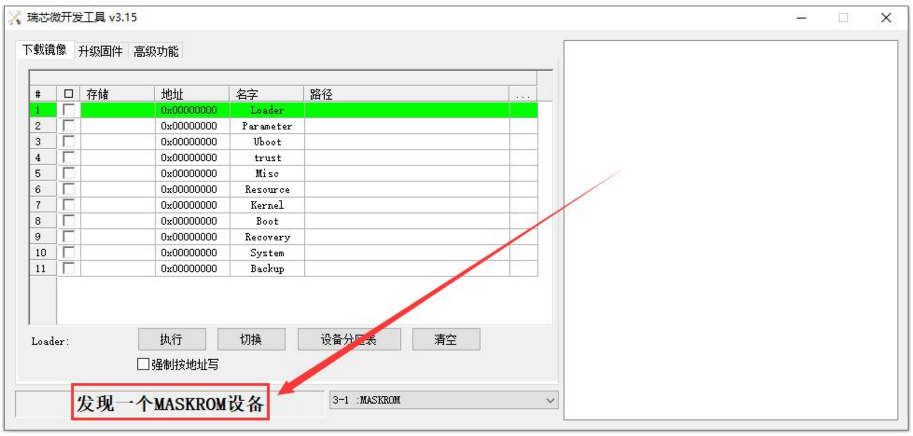
c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:



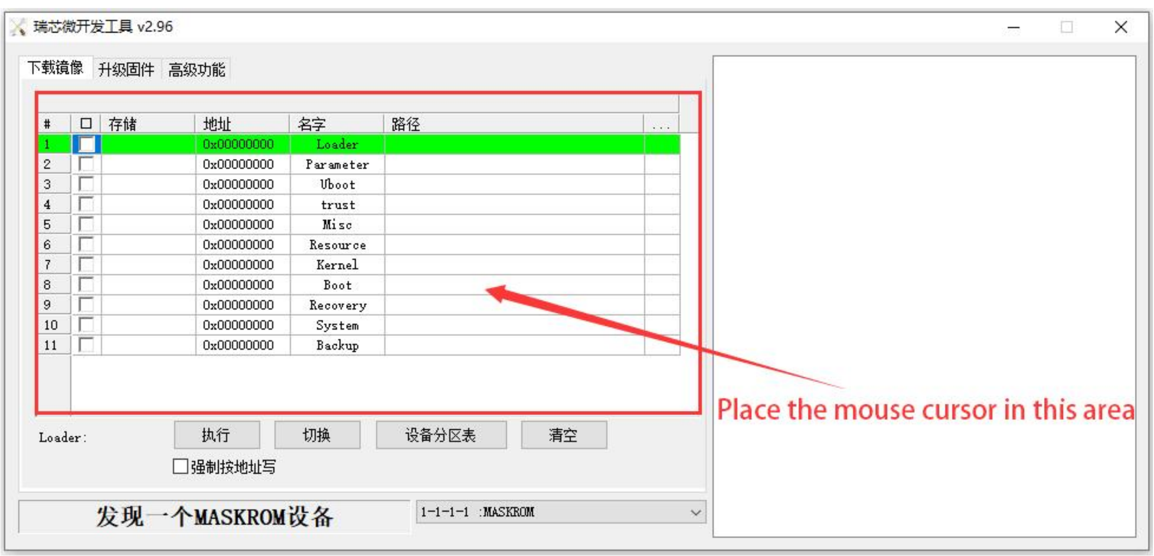
d. Then connect the Type-C interface power supply to the development board and power it on, then you can release the MaskROM button



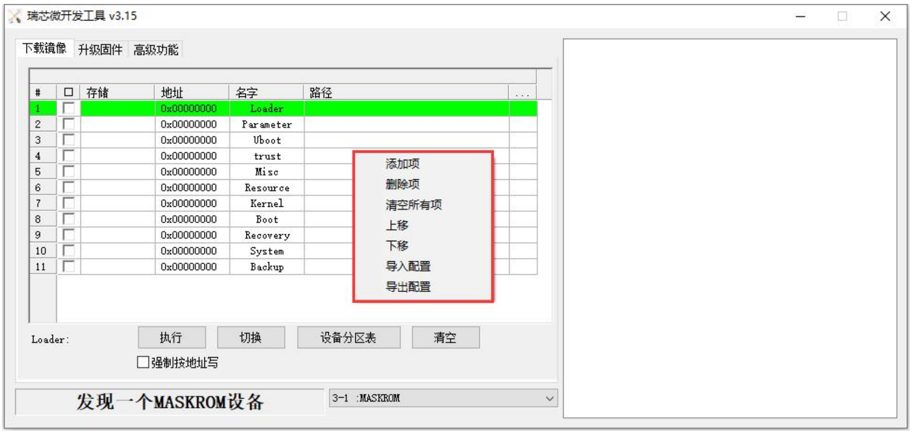
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



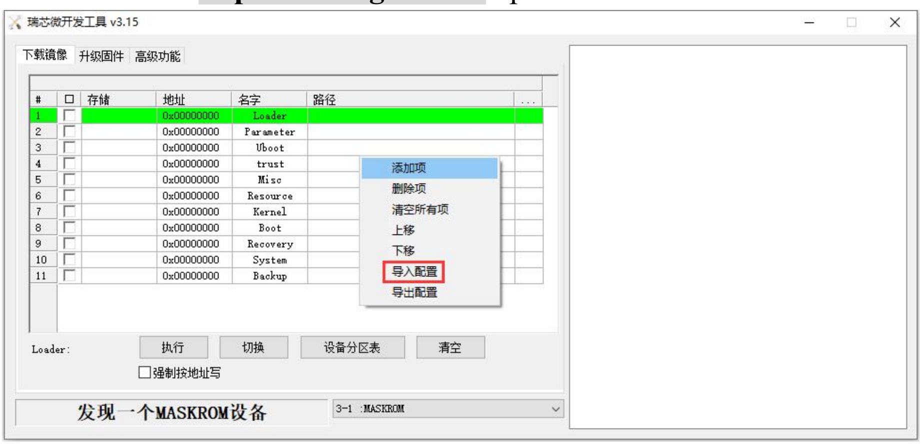
f. Then place the mouse cursor in the area below



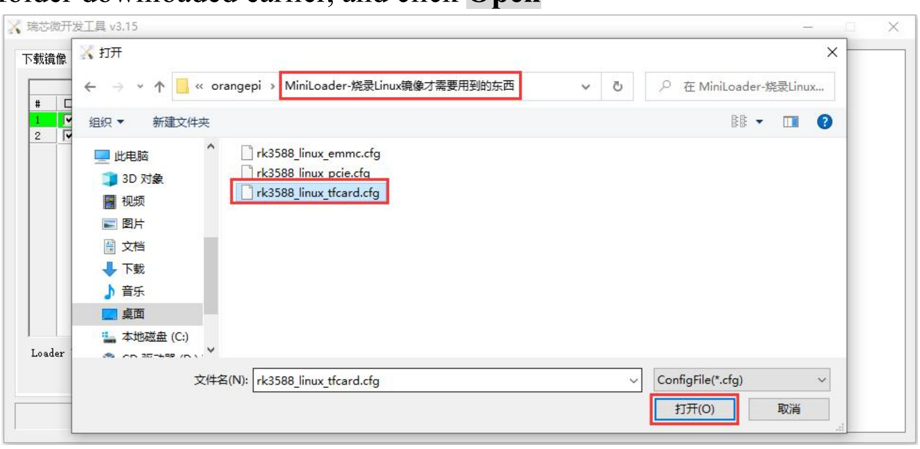
g. Then click the right mouse button and the selection interface shown in the figure below will pop up



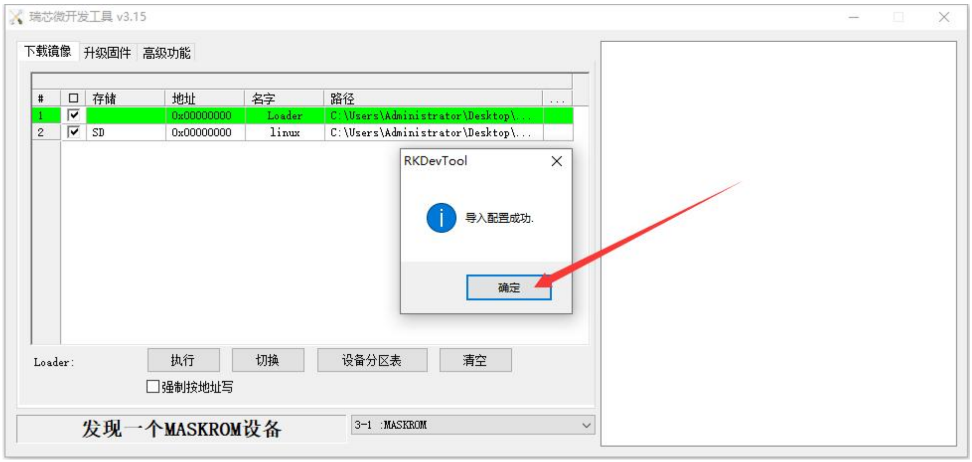
h. Then select the **import configuration** option



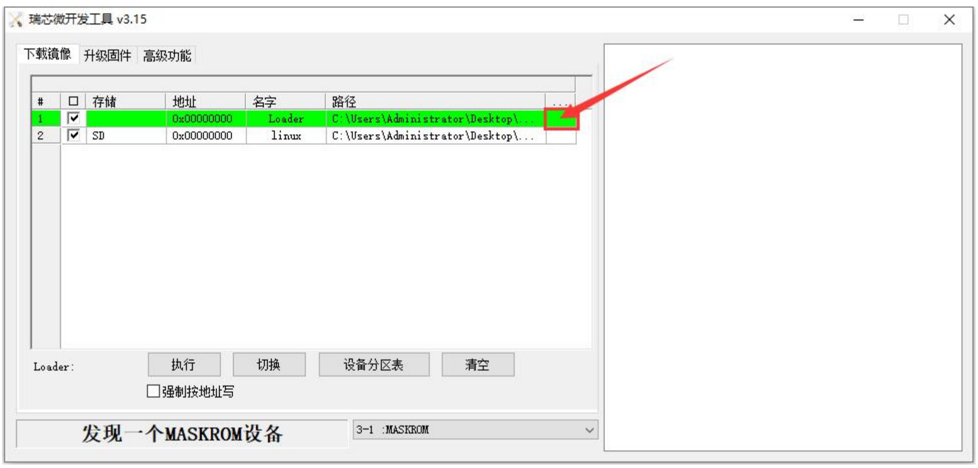
i. Then select the r**k3588_linux_tfcard.cfg** configuration file in the MiniLoader folder downloaded earlier, and click **Open**



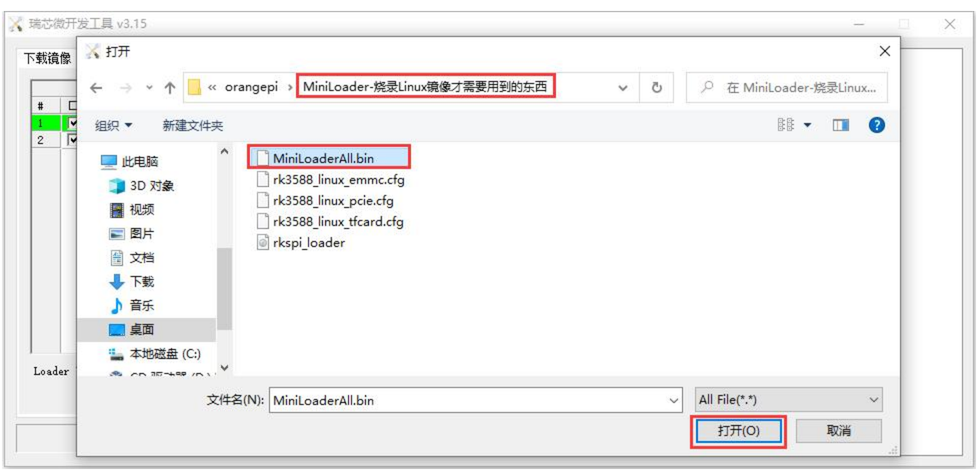
j. Then click **OK**



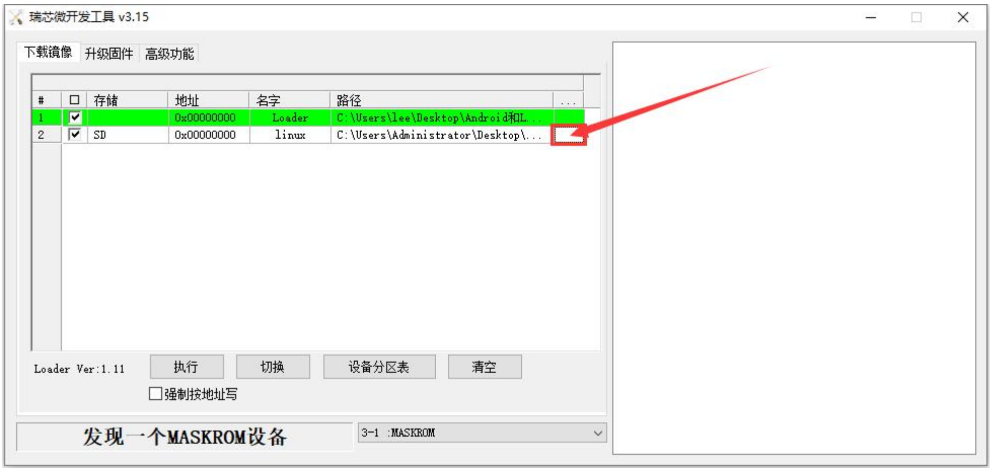
k. Then click the position shown in the figure below



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

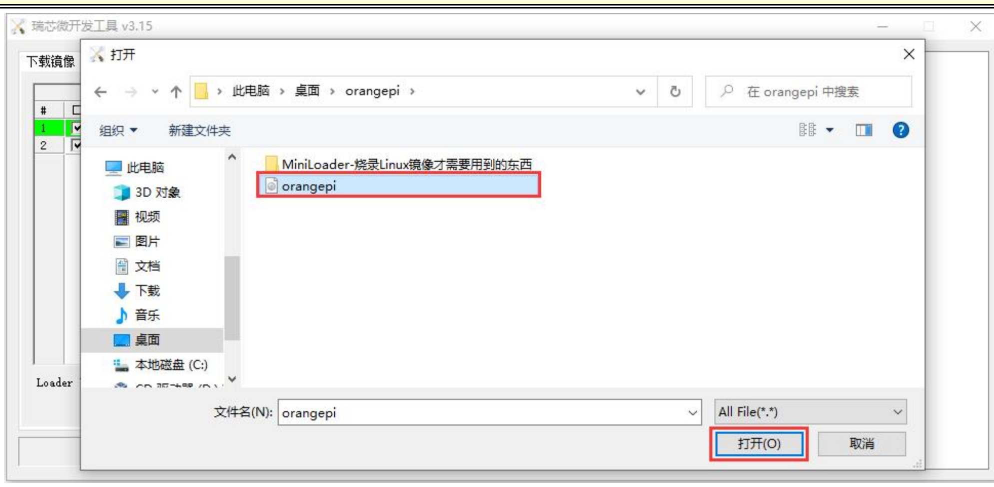


m. Then click on the location shown in the following image

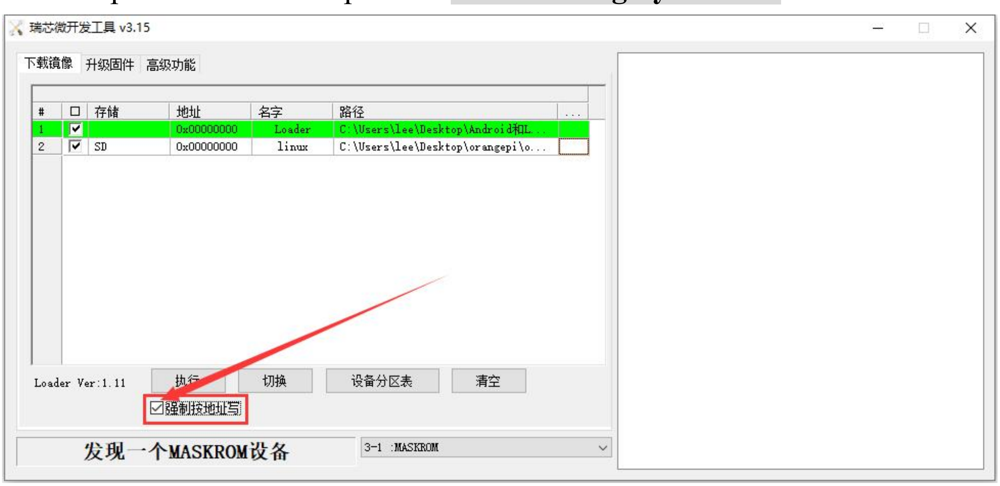


n. Then select the path to the Linux image you want to burn, and **click open**

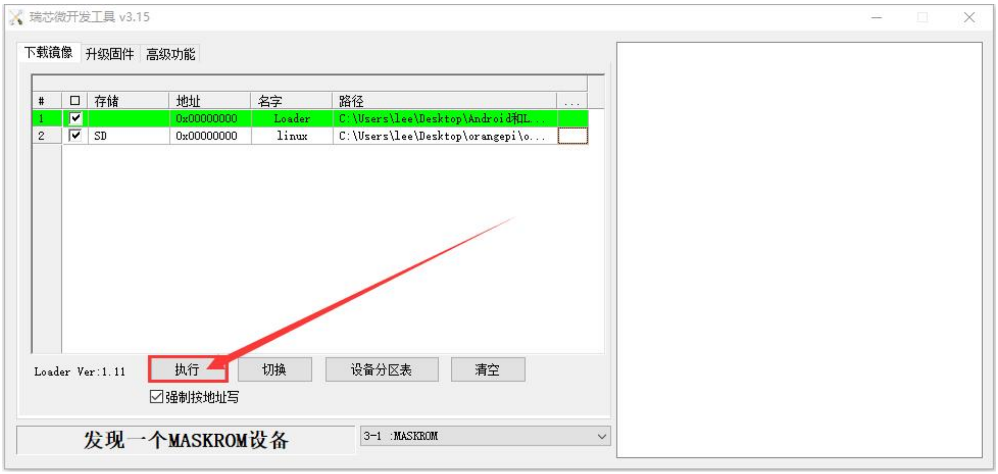
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



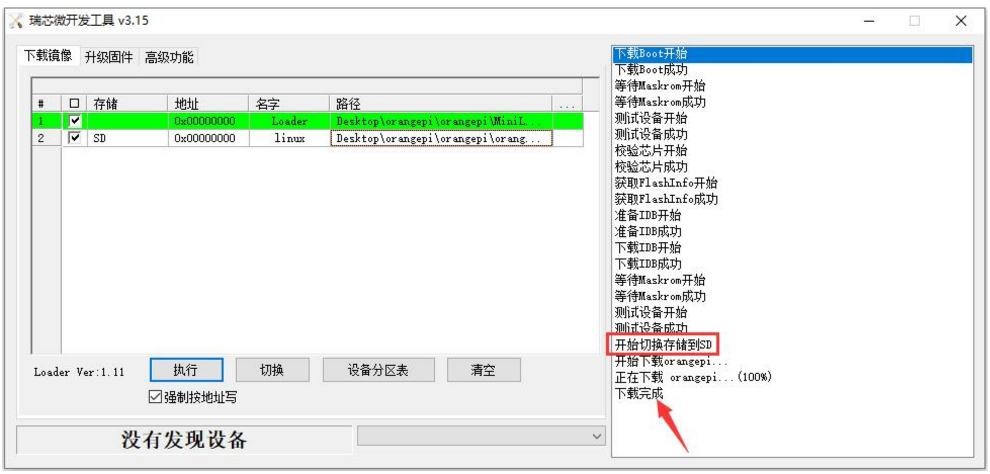
o. Then please check the option to **force writing by address**



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

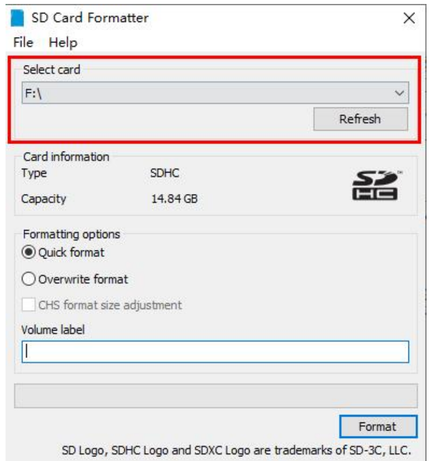
3) Then format the TF

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

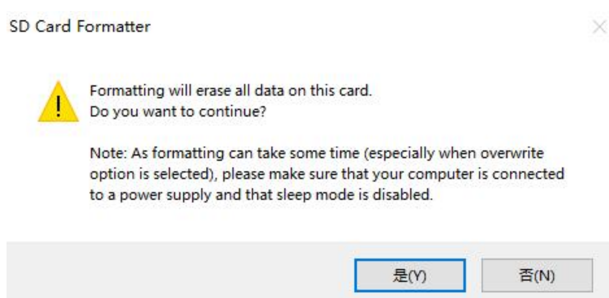
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

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

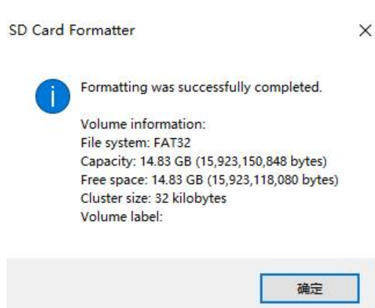
c. If only a TF card is inserted into the computer, the drive letter of the TF card will be displayed in the "**Select card**" column. If multiple USB storage devices are inserted into the computer, you can select the corresponding drive letter of the TF card through the drop-down box.



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

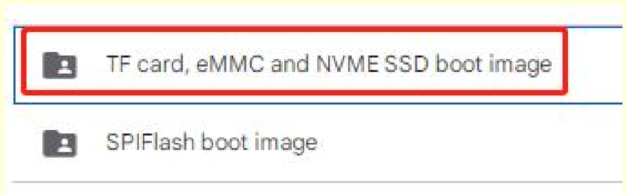


e. After formatting the TF card, the information shown in the figure below will pop up, click OK



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

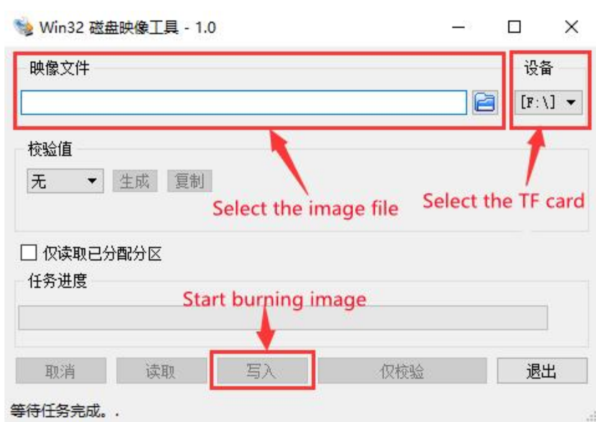
http://sourceforge.net/projects/win32diskimager/files/Archive/

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.

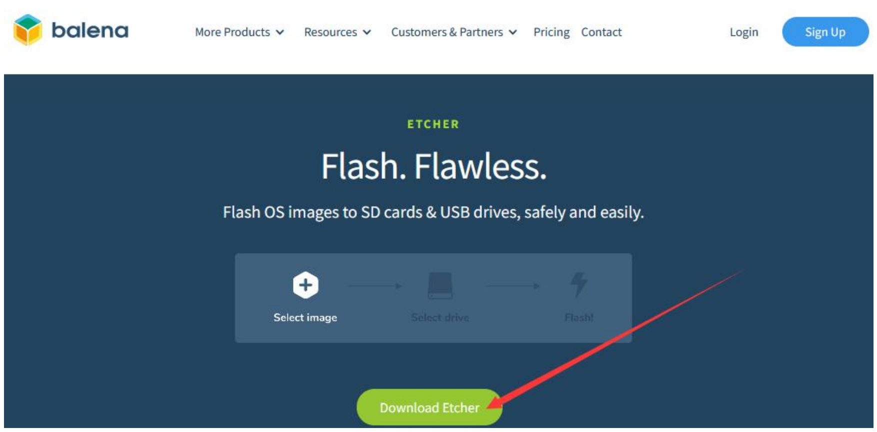
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brand

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

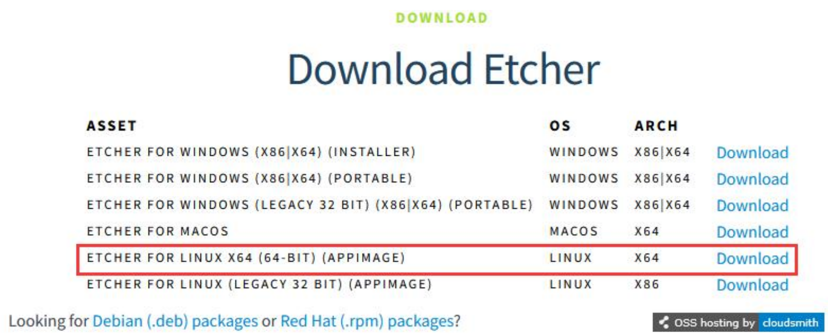
3) Download the balenaEtcher software, the download address is:

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

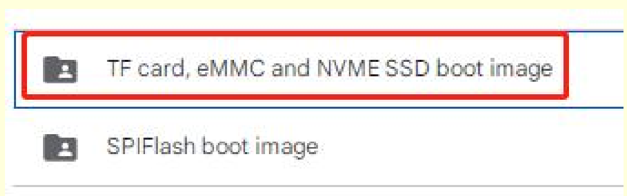


5) Then choose to download the Linux version of the software



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.



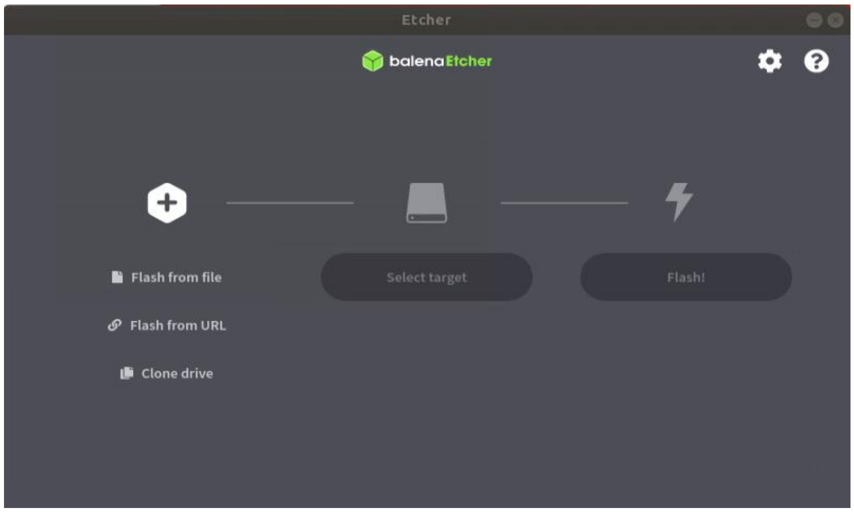
The decompression command for the compressed file ending in 7z is as follows

```
test@test:~$ 7z x Orangepi5max_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
test@test:~$ ls Orangepi5max_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.*
Orangepi5max_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
Orangepi5max_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.sha      #Verification
and file
Orangepi5max_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img      #Mirror file
```

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

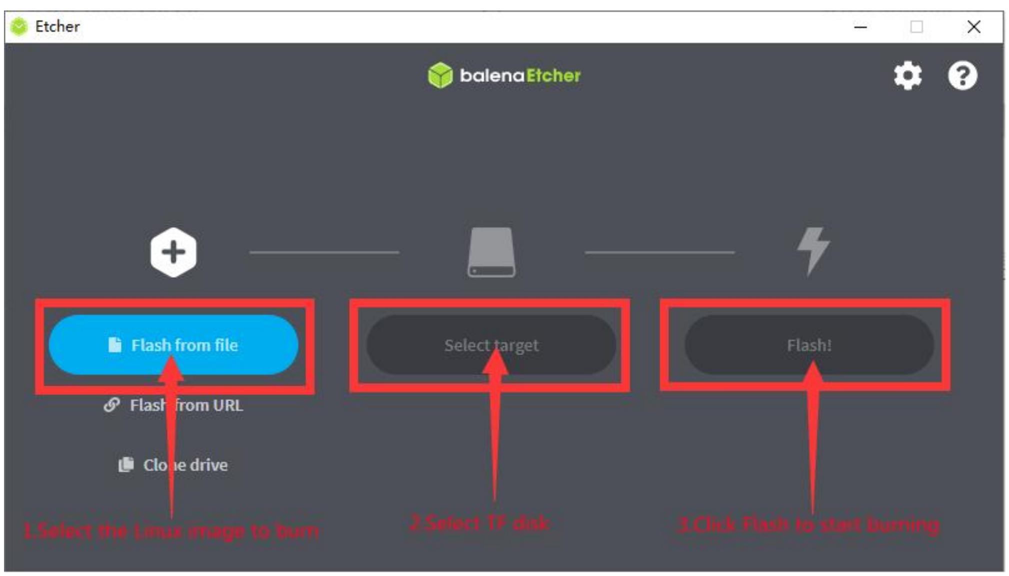
```
test@test:~$ sha256sum -c *.sha
Orangepi5max_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img: OK
```

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



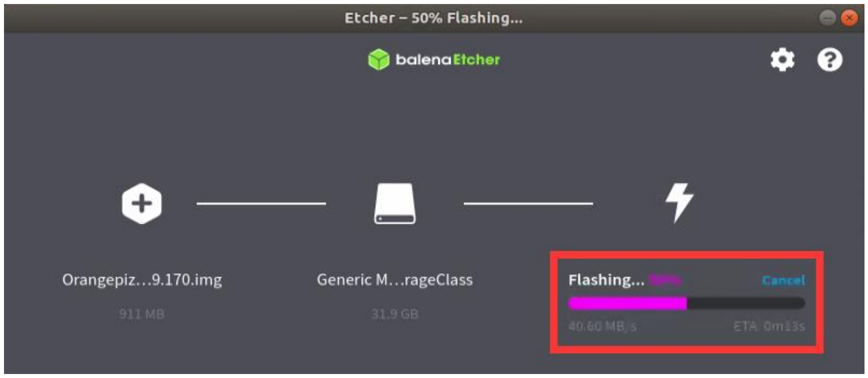
9) The specific steps to use balenaEtcher to burn the Linux image are as follows

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

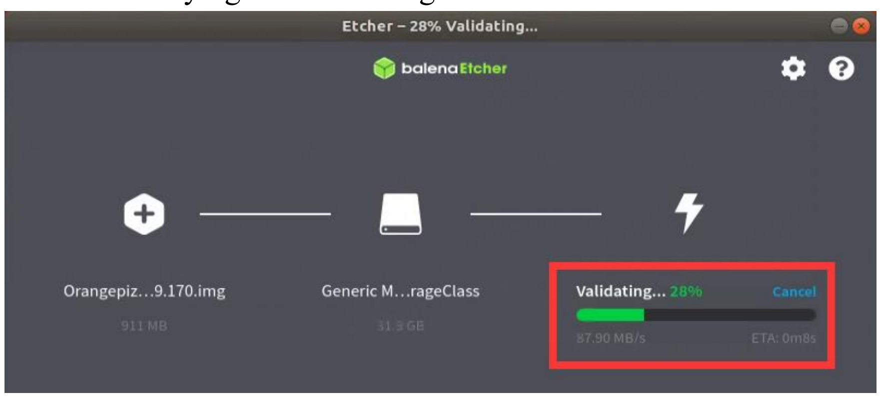


10) The interface displayed during the process of burning a Linux image with

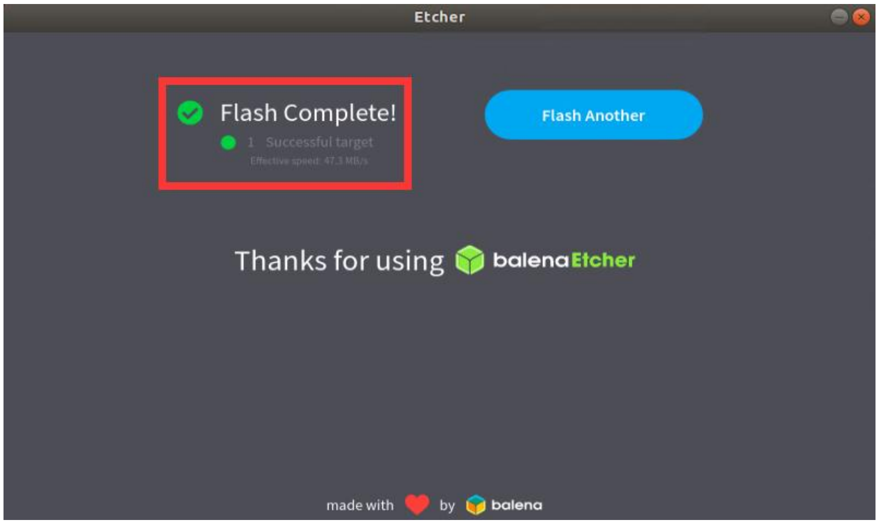
BalenaEtcher is shown in the following figure. In addition, the progress bar displays purple, indicating that the Linux image is being burned to the TF card



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.



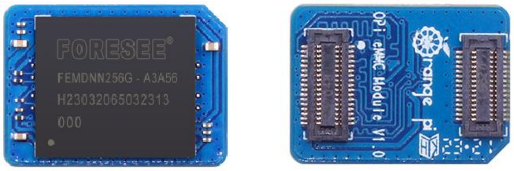
2. 5. The method of burning Linux images into eMMC

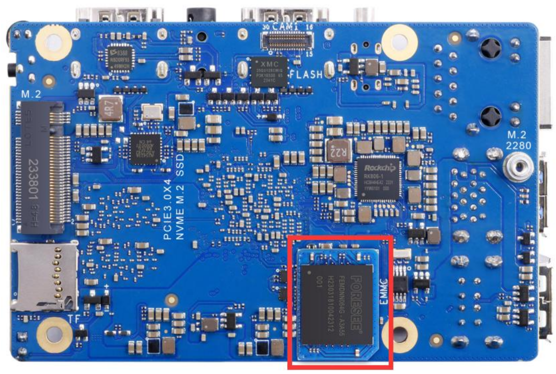
2. 5. 1. Method of burning Linux images into eMMC using RKDevTool

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:



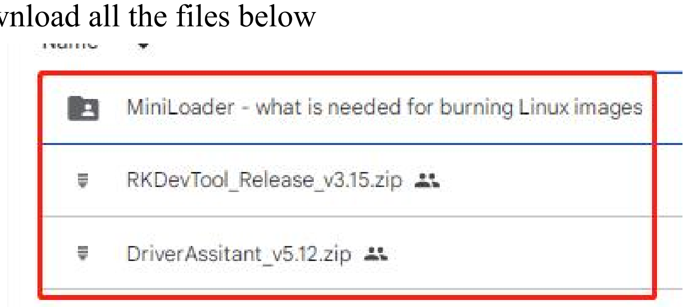


2) Also need to prepare a good quality USB2.0 public to public data cable

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

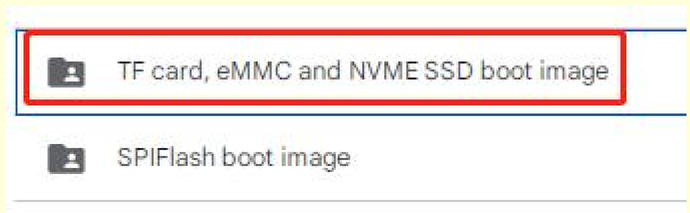
b. Then download all the files below



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

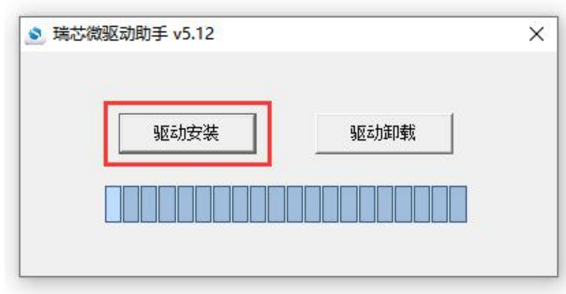
Note that if you are downloading an OpenWRT image, you will see the following two types of images in the download link of the OpenWRT image. Please download the image files from the "TF Card, eMMC, and NVME SSD Boot Images" folder.



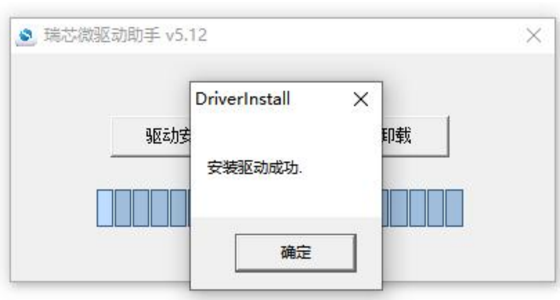
5) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

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.

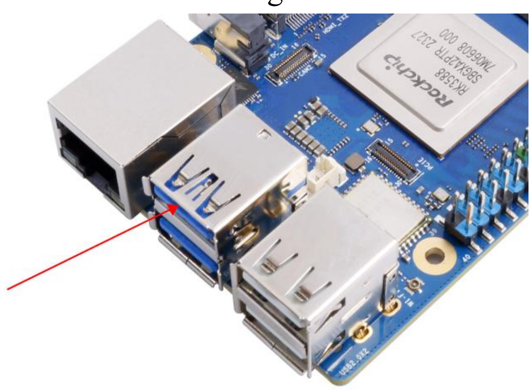


7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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**"

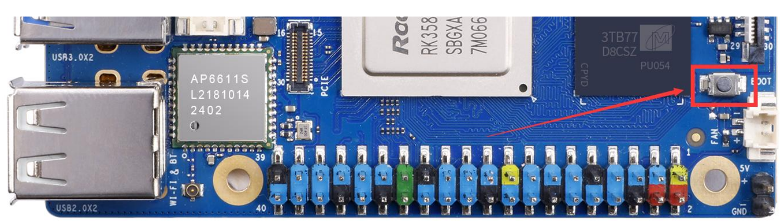
9) Then start burning Linux images into eMMC

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

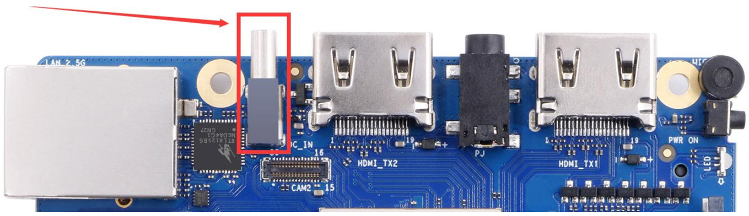


b. Ensure that the development board is not inserted with a TF card or connected to a power source

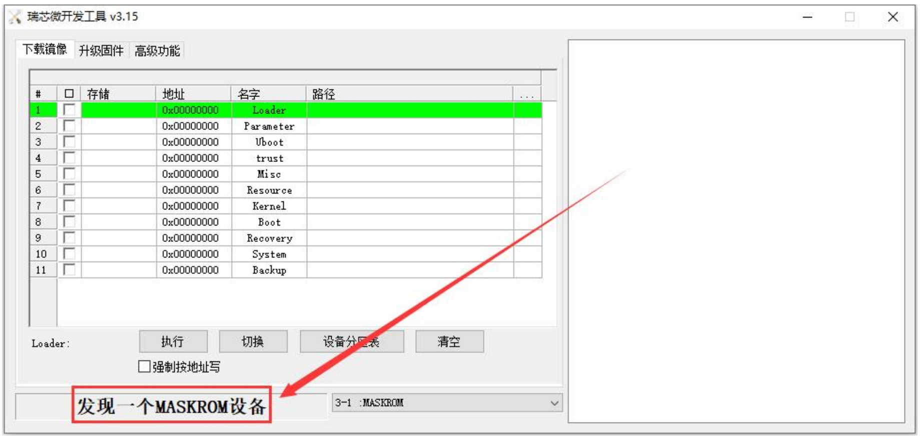
c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:



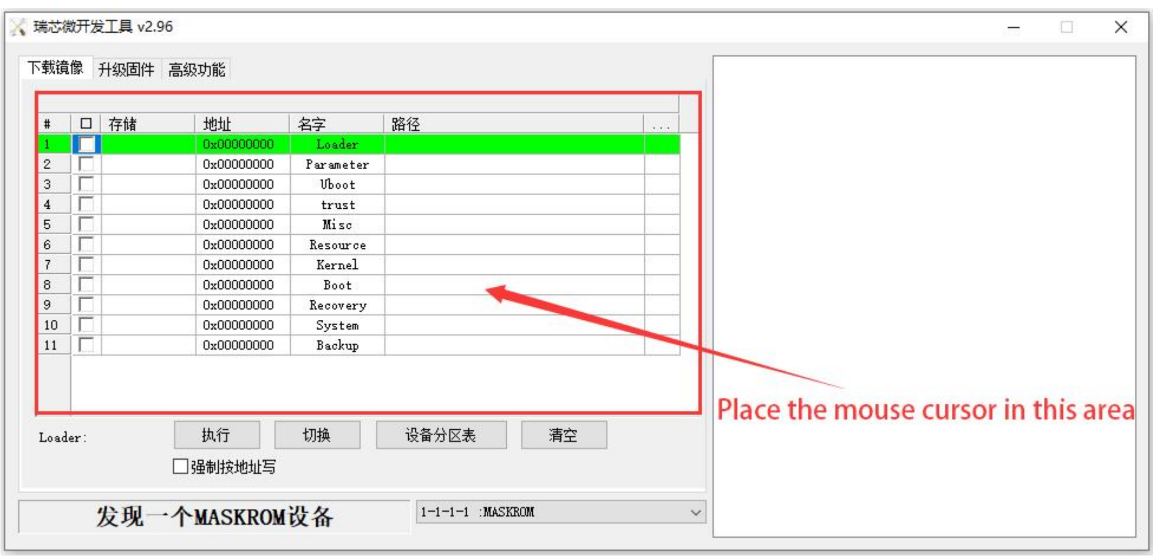
d. Then connect the Type-C interface power supply to the development board and power it on, then you can release the MaskROM button



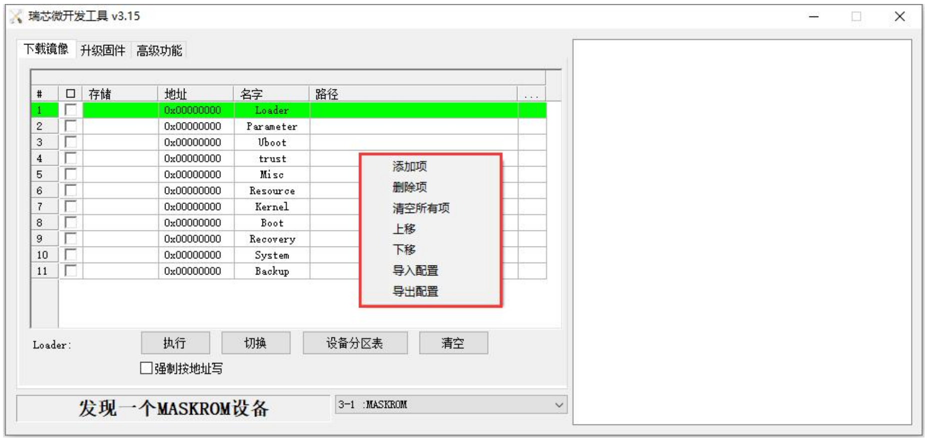
e. If the previous steps are successful, the development board will enter **MASKROM** mode and the interface of the burning tool will prompt "**Found a MASKROM device**"



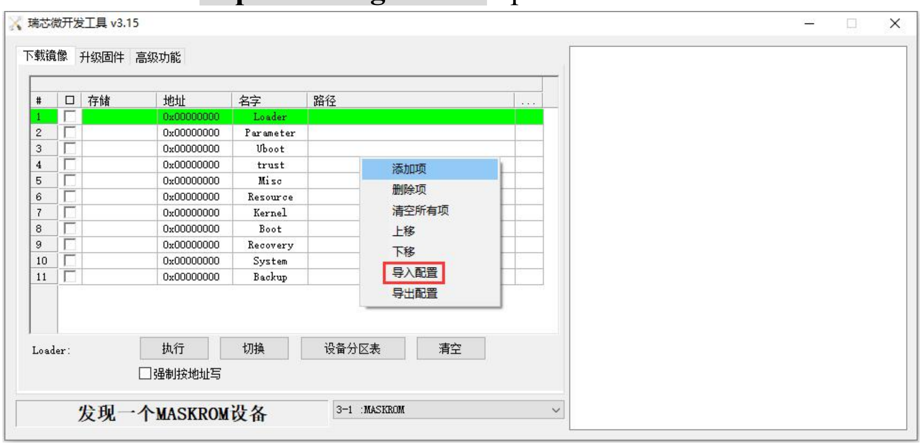
f. Then place the mouse cursor in the area below



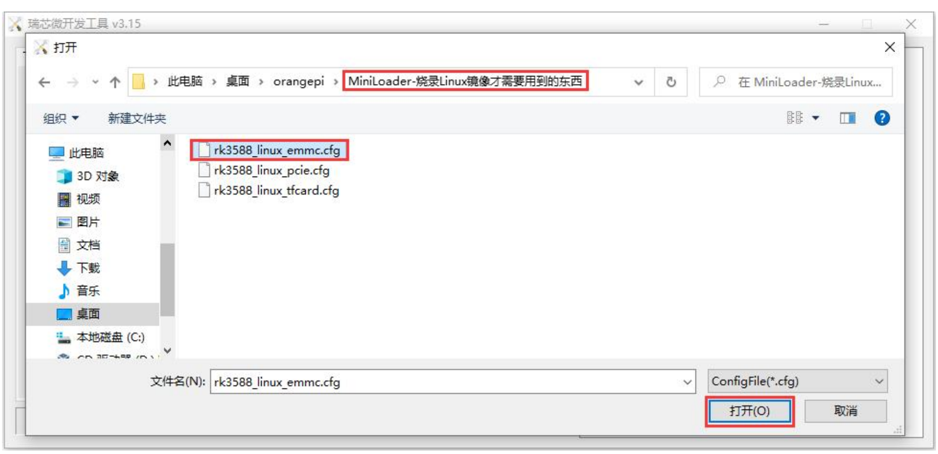
g. Then, clicking the right mouse button will pop up the selection interface shown in the following figure



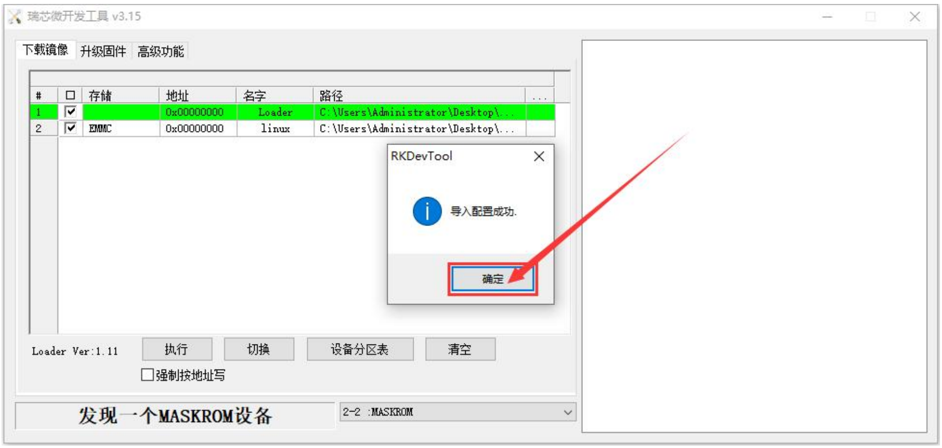
h. Then select the **import configuration** option



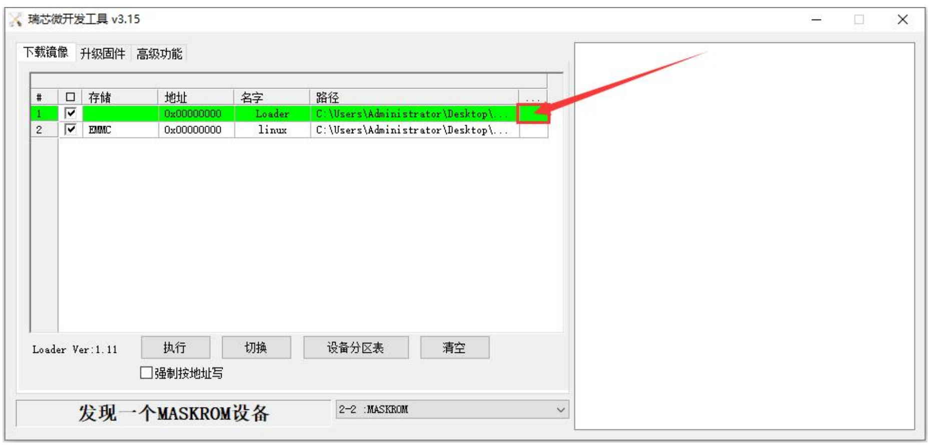
i. Then select the **rk3588_linux_emmc.cfg** configuration file from the **MiniLoader** folder downloaded earlier, and click to **open**



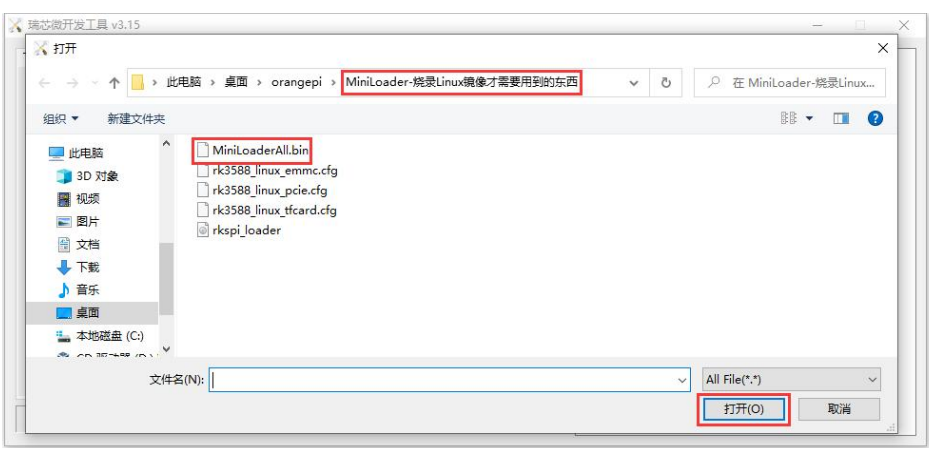
j. Then click **OK**



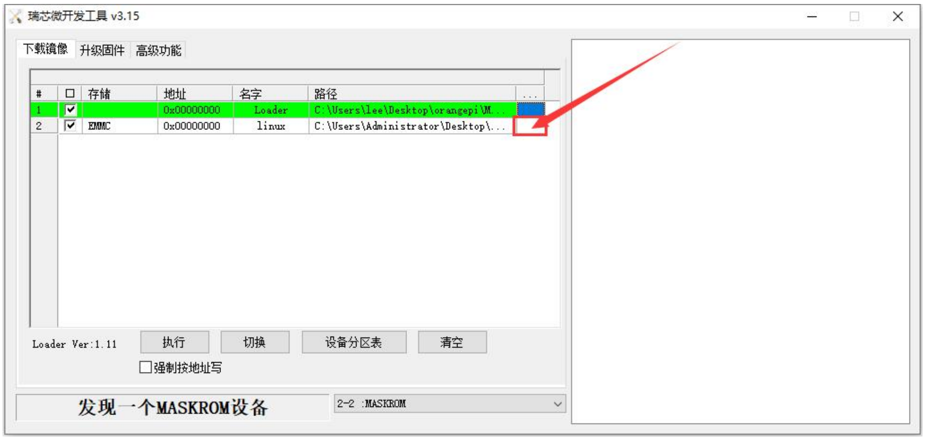
k. Then click on the location shown in the following image



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

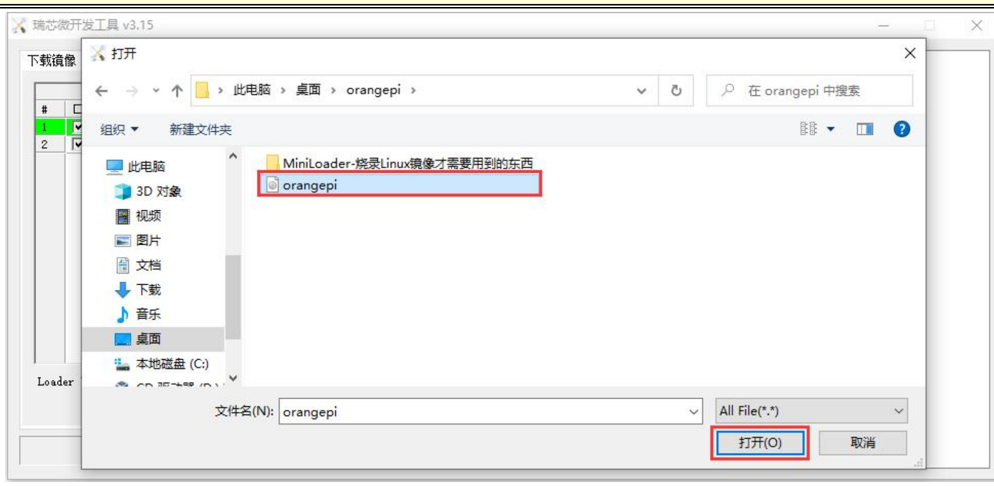


m. Then click on the location shown in the following image

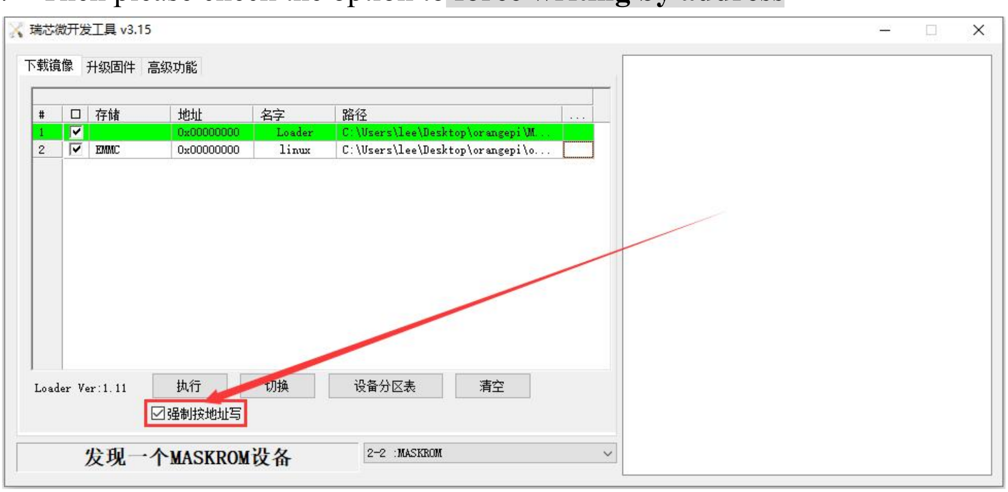


n. Then select the path to the Linux image you want to burn, and click **open**

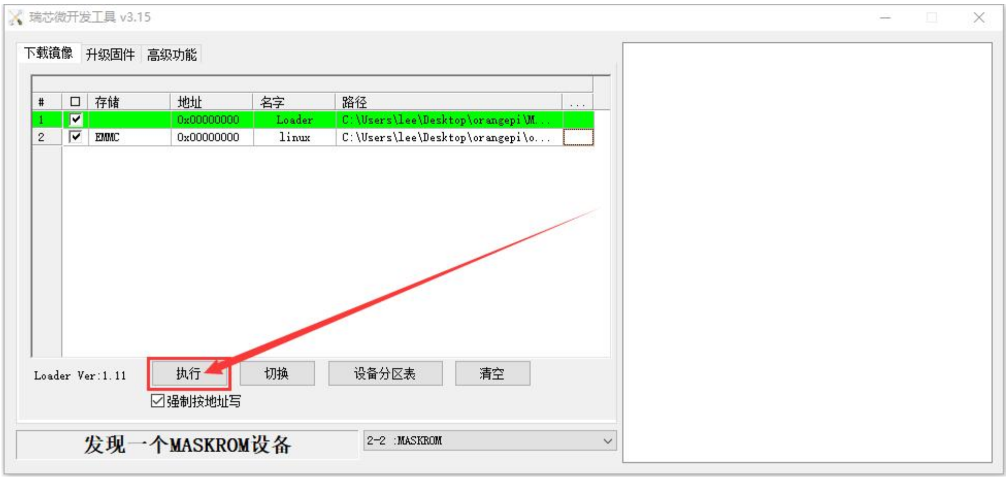
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



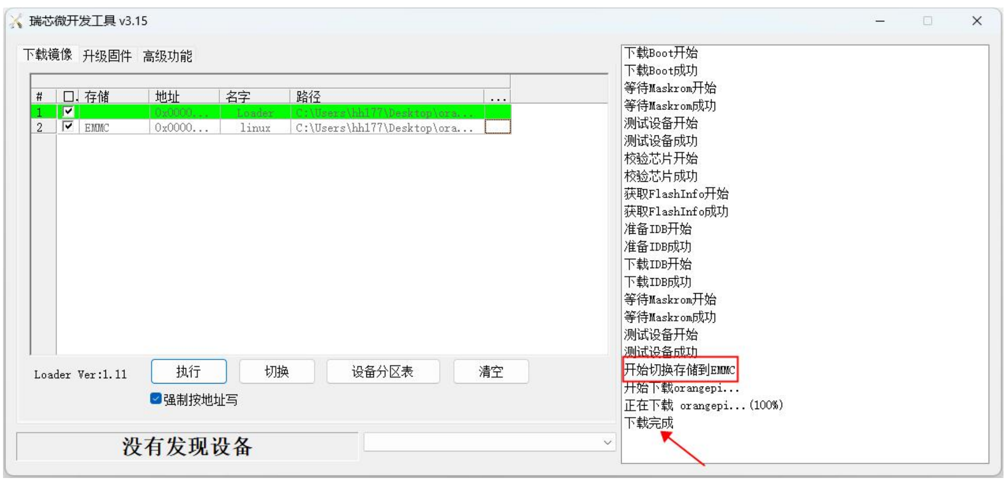
o. Then please check the option to **force writing by address**



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



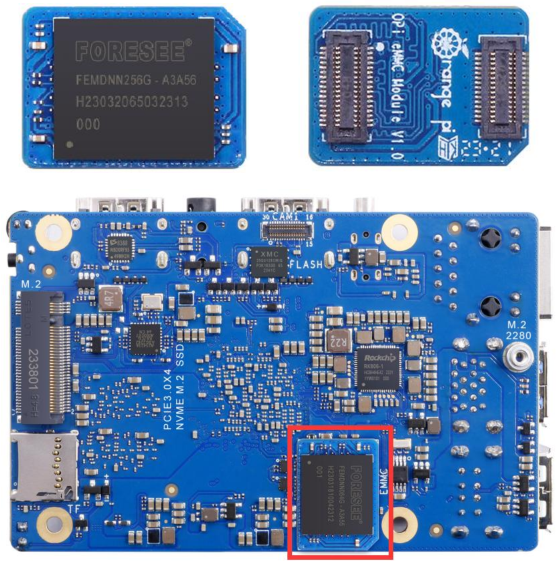
r. After burning the Linux image into eMMC, the Linux system will automatically start.

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
```

```
Orangepi5max_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=Orangepi5max_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 for burning Linux images to SPIFlash+NVMe SSD

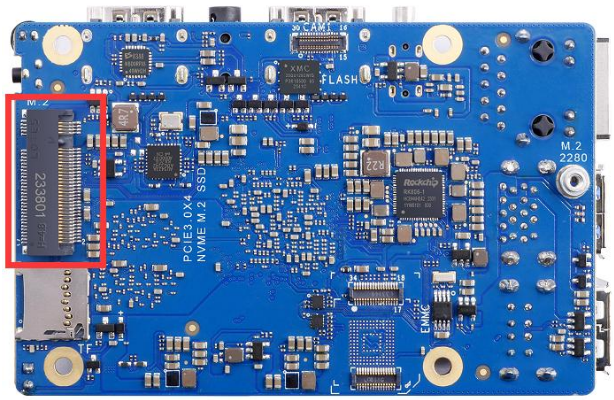
Note that the Linux image referred to here specifically refers to Linux distribution images such as Debian, Ubuntu, OpenWRT, or OPi OS Arch downloaded from the Orange Pi data download page.

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

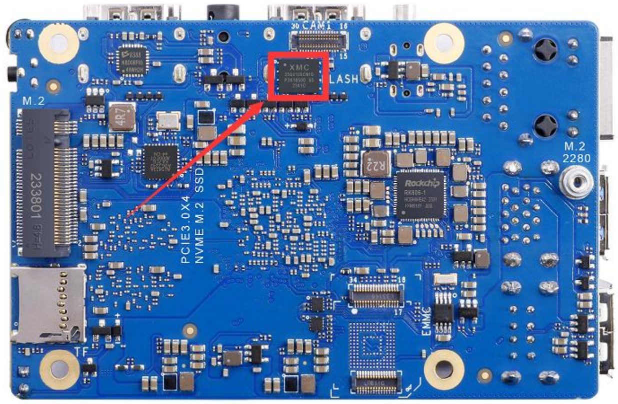
2.6.1. Method of burning using RKDevTool

1) Firstly, it is necessary to prepare an NVMe SSD solid state drive with a PCIe interface specification of PCIe3.0x4 for the M.2 slot of the development board.

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



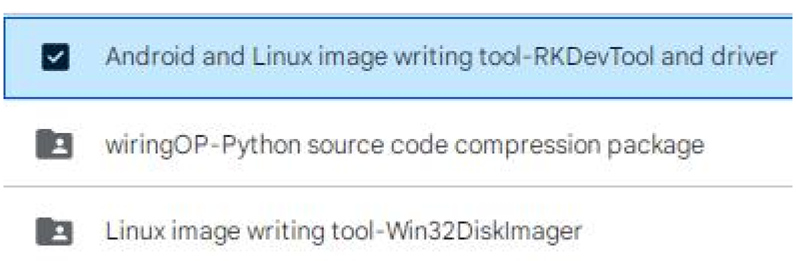
3) The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting the burning process



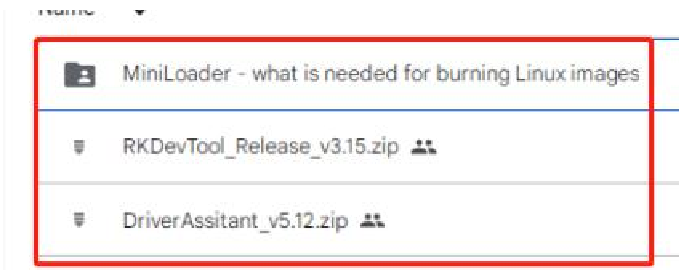
4) Then we need to prepare a high-quality USB 2.0 male to male data cable

5) Then download the Rockchip micro driver **DriverAssitant_v5.12.zip**, **MiniLoader**, and burning tool **KDevTool_Release_v3.15.zip** from the **Orange Pi's download page**

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



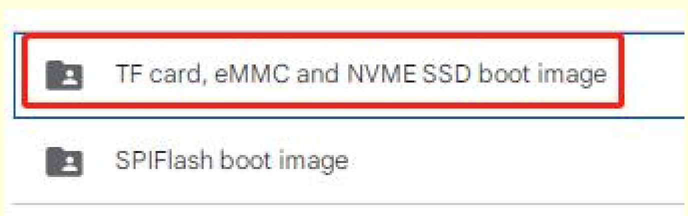
b. Then download all the files below



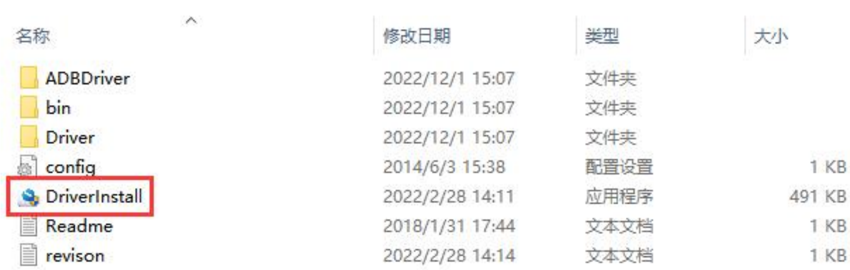
Note that the "MiniLoader - something needed to burn Linux images" folder is hereinafter referred to as the MiniLoader folder.

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.

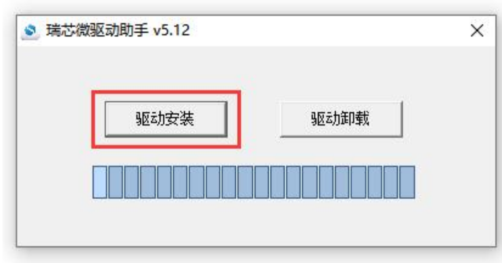


7) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

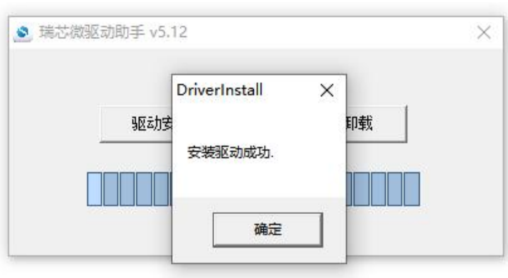


8) The steps to install the Ruixin micro driver after opening **DriverInstall.exe** are as follows

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

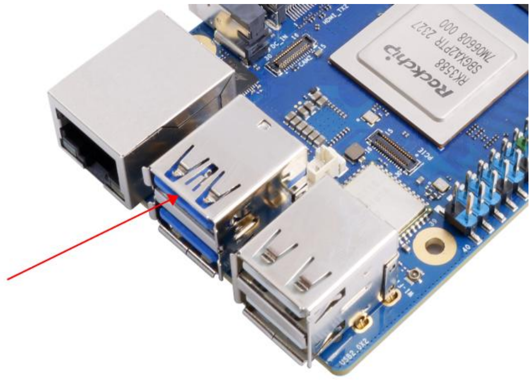


9) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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**"

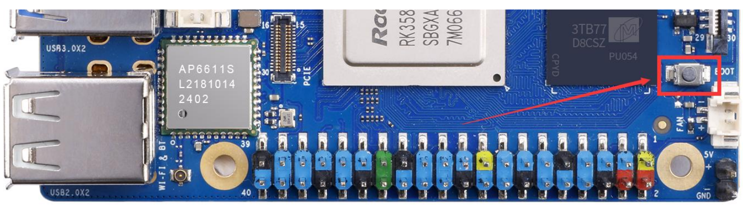
11) Then start burning the Linux image to the SSD

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

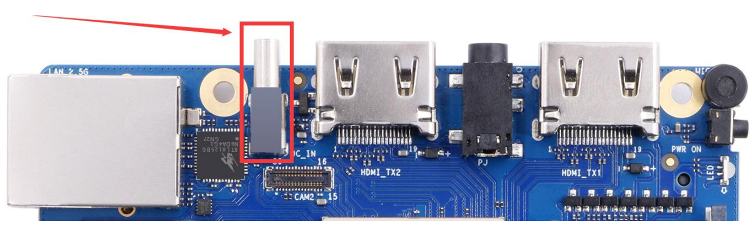


b. Ensure that the development board is not connected to a power source or inserted with a TF card

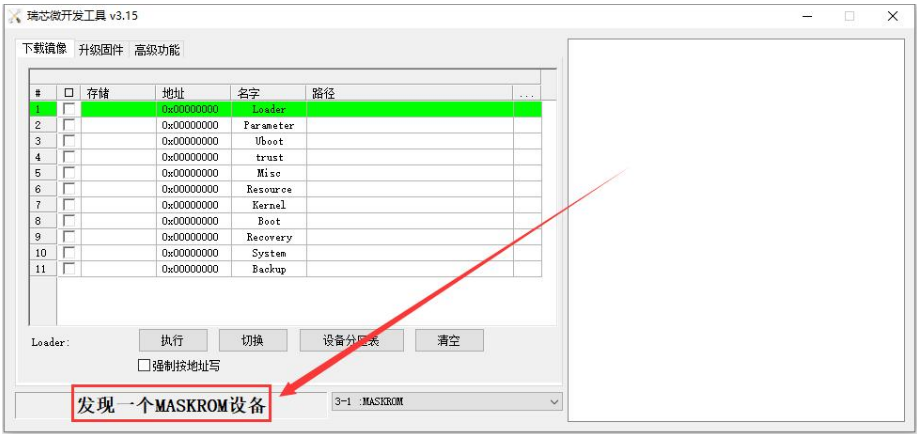
c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:



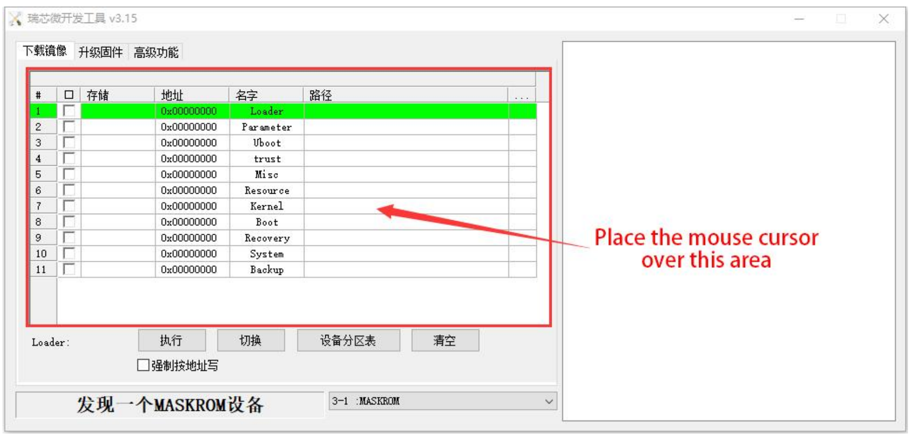
d. Then connect the Type-C interface power supply to the development board and power it on, then you can release the MaskROM button



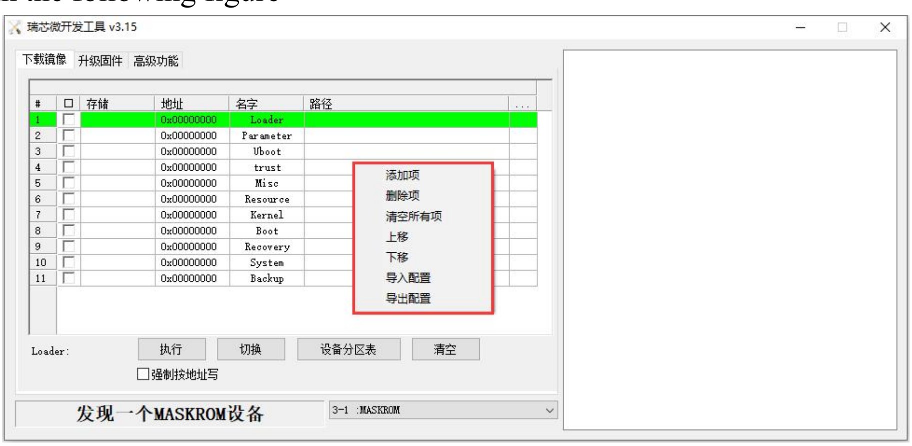
e. If the previous steps are successful, the development board will enter **MASKROM** mode and the interface of the burning tool will prompt "**Found a MASKROM device**"



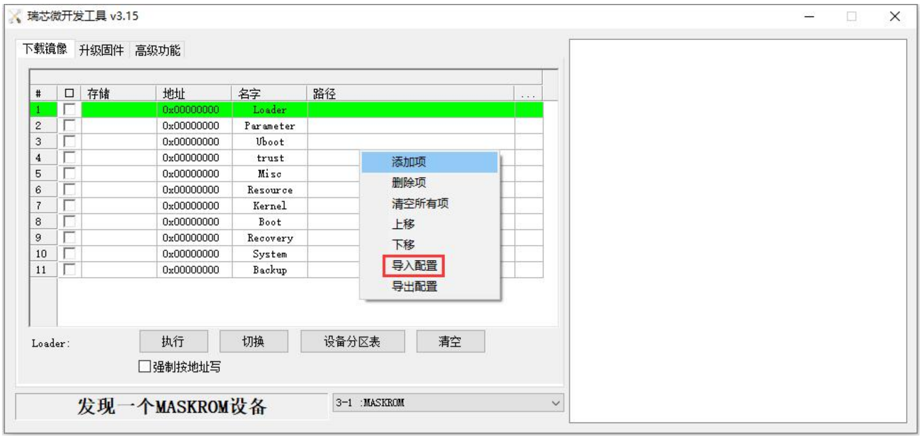
f. Then place the mouse cursor in the area below



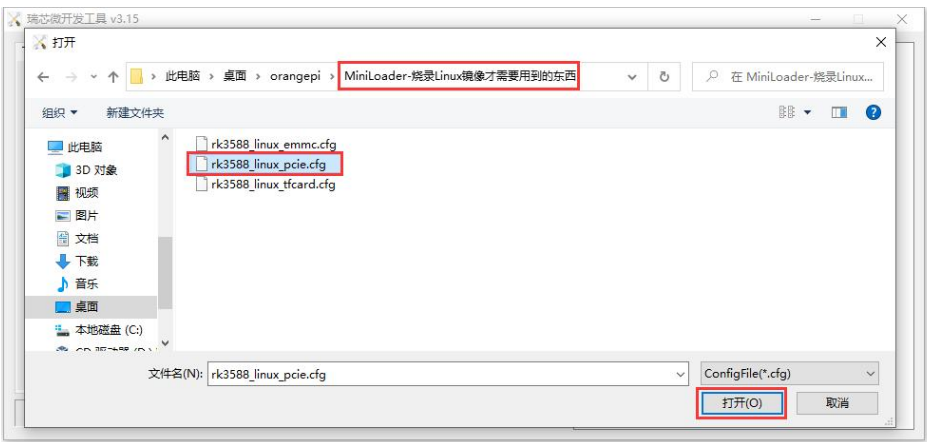
g. Then, clicking the right mouse button will pop up the selection interface shown in the following figure



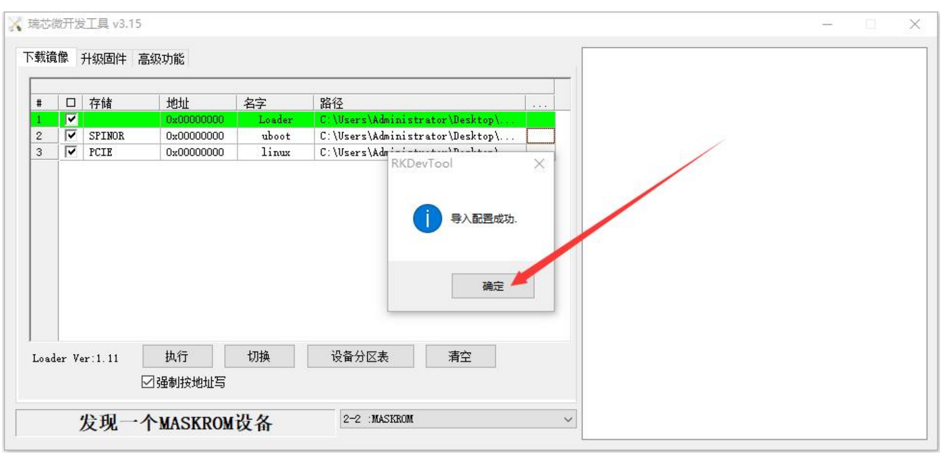
h. Then select the **import configuration** option



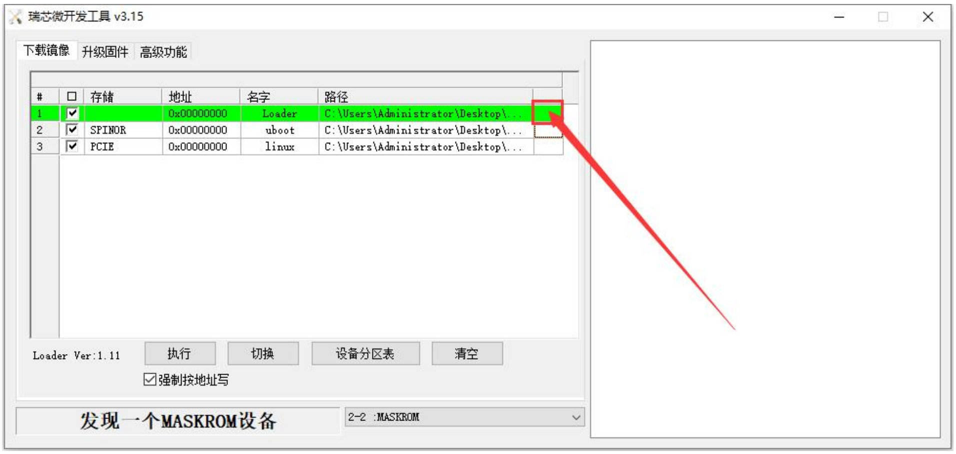
i. Then enter the MiniLoader folder downloaded earlier, select the **rk3588_linux_pcie.cfg** configuration file, and click **open**



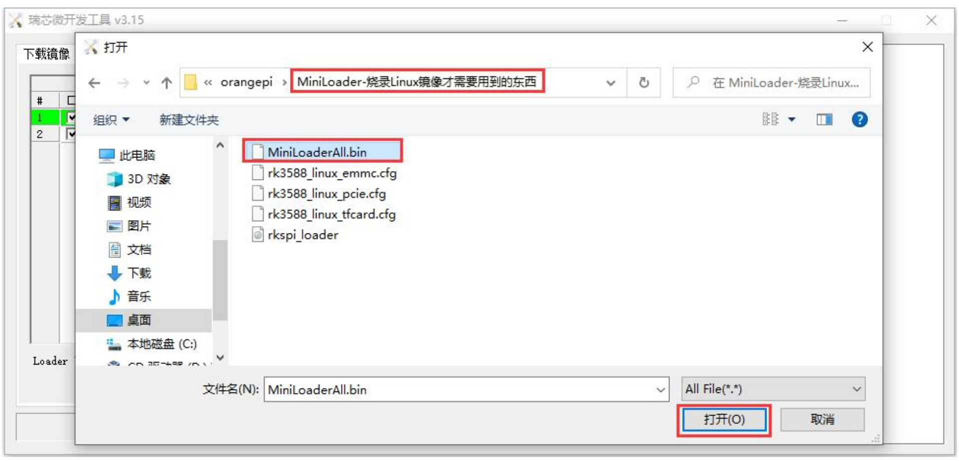
j. Then click **OK**



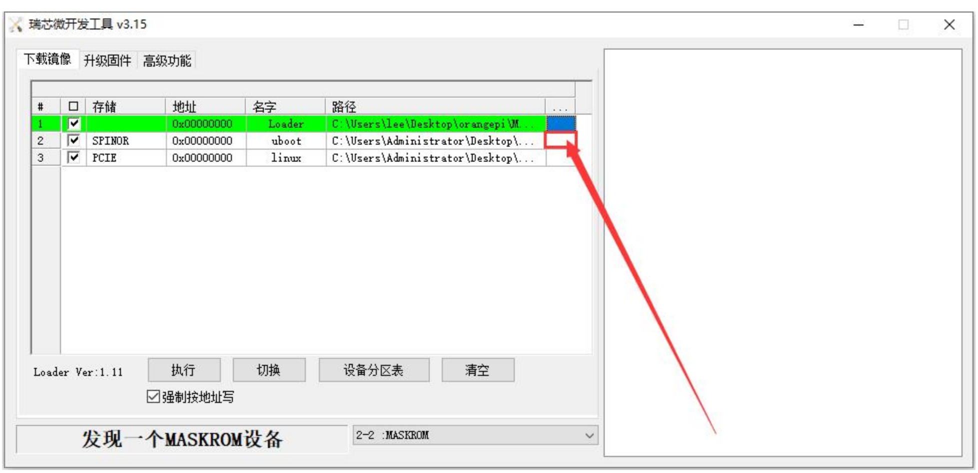
k. Then click on the location shown in the following image



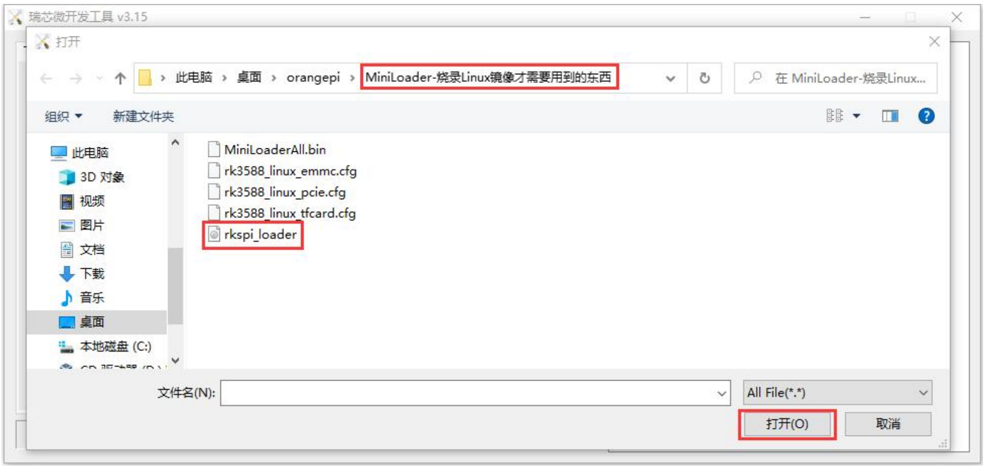
l. Select **MiniLoaderAll.bin** from the **MiniLoader** folder downloaded earlier, and then click **open**



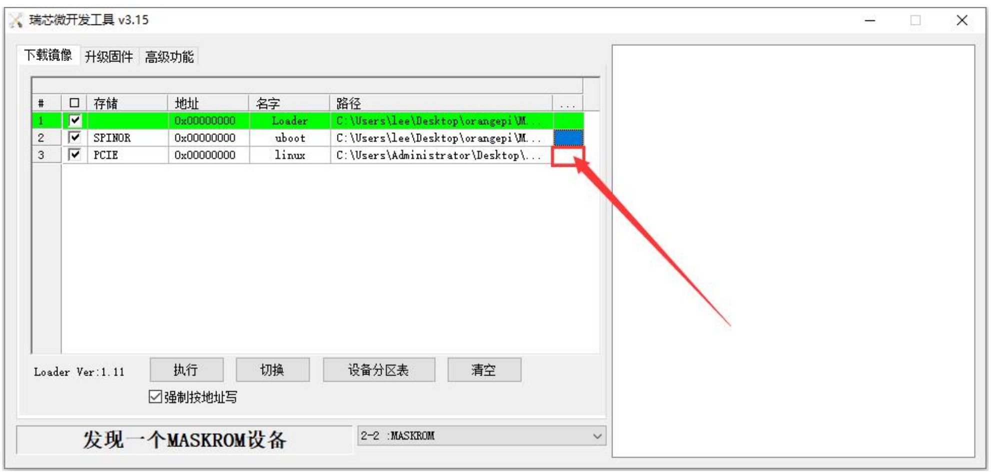
m. Then click on the location shown in the following image



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

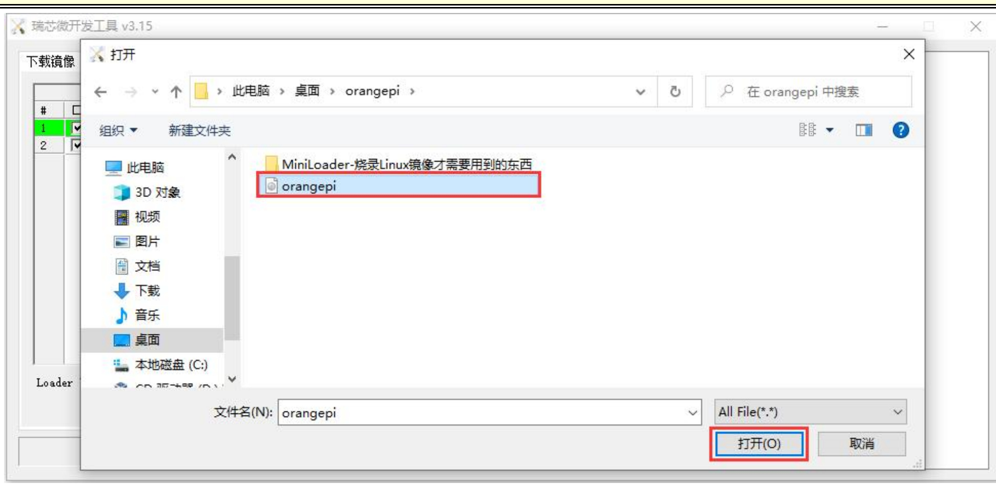


o. Then click on the location shown in the following image

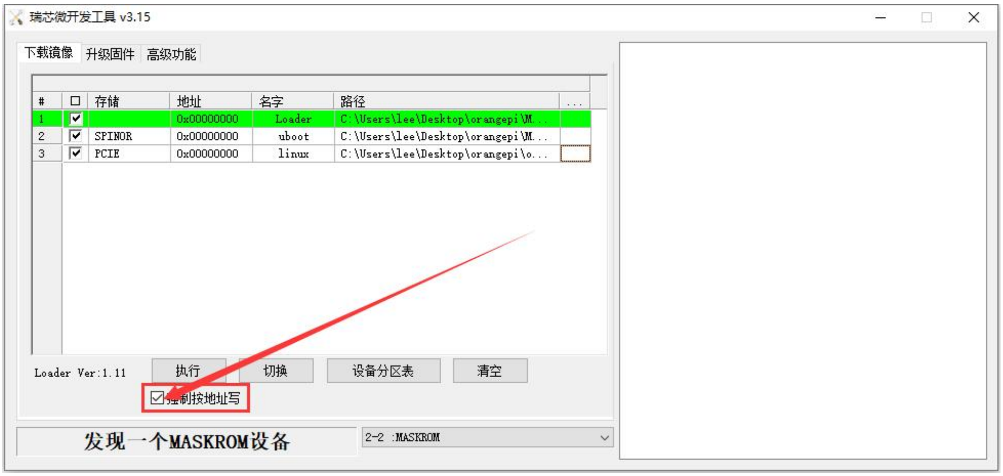


p. Then select the path to the Linux image you want to burn, and click **open**

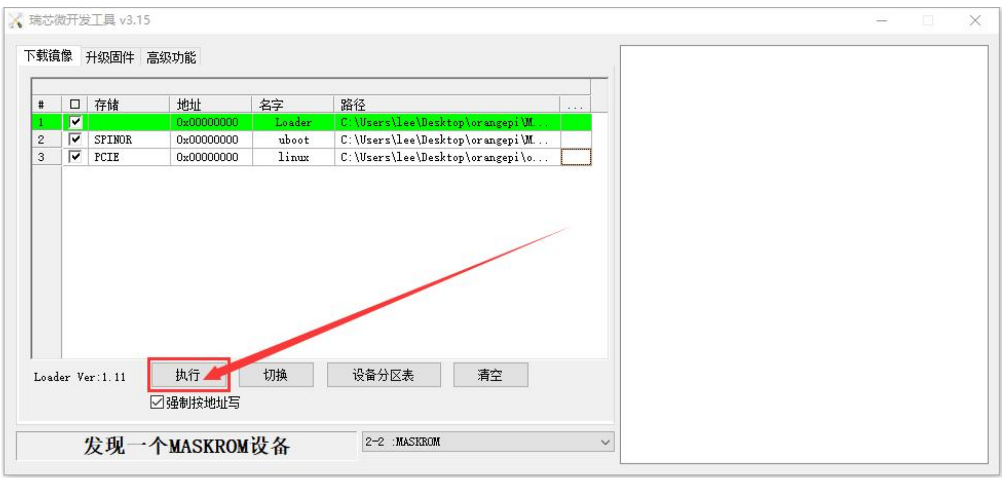
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



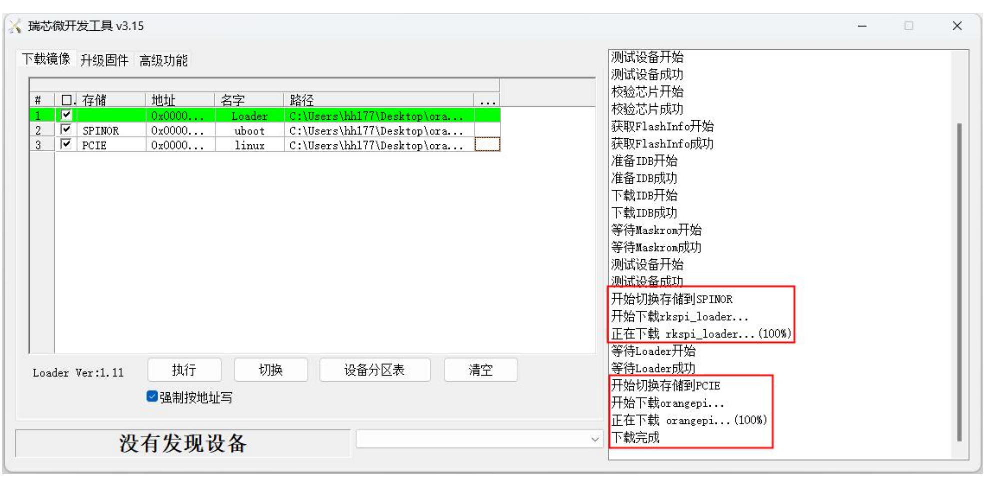
q. Then please check the option to **force writing by address**



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.

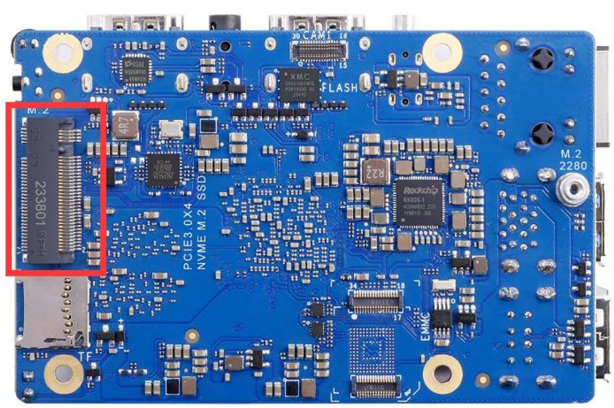
2. 6. 2. How to use the dd command to burn

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.

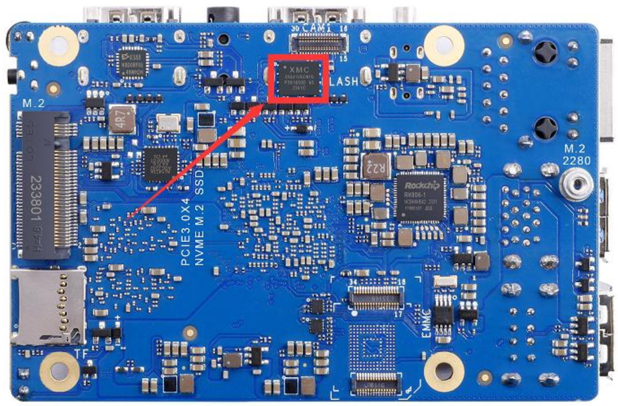
a. The M.2 2230 SSD is as follow

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



9) 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



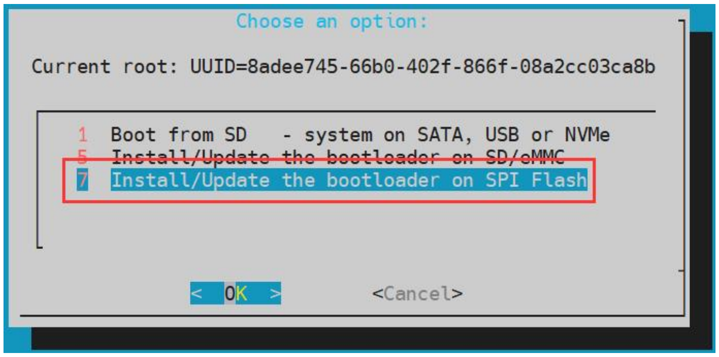
10) Burning a Linux image to a SPIFlash+NVMe SSD requires the use of a TF card, so the first step is to burn the Linux image onto the TF card, and then use the TF card to boot the development board into the Linux system. The method of burning a Linux image to a TF card can be found in the two sections: **the method of burning a Linux image to a TF card based on Windows PC** and **the method of burning a Linux image to a TF card based on Ubuntu PC**.

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

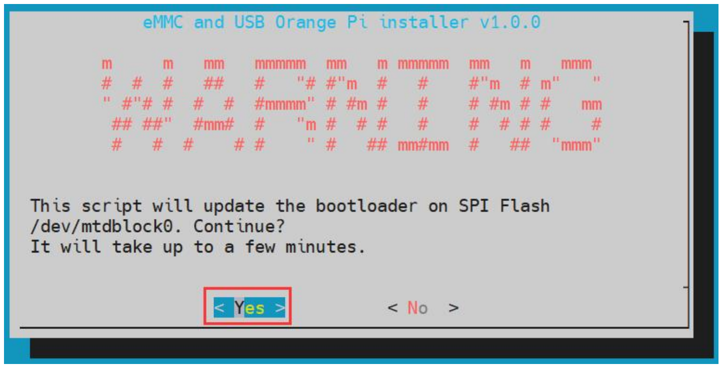
a. First, run **nand-sata-install**. **Regular users should remember to grant sudo privileges**

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

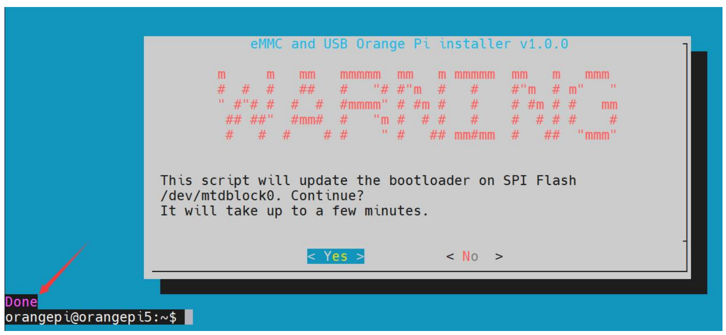
b. Then choose **7 Install/Update ther bootloader on SPI Flash**



c. Then choose **<Yes>**



d. Then please be patient and wait for the burning to complete. After the burning is completed, the following will be displayed (a **Done** will appear in the bottom left corner):

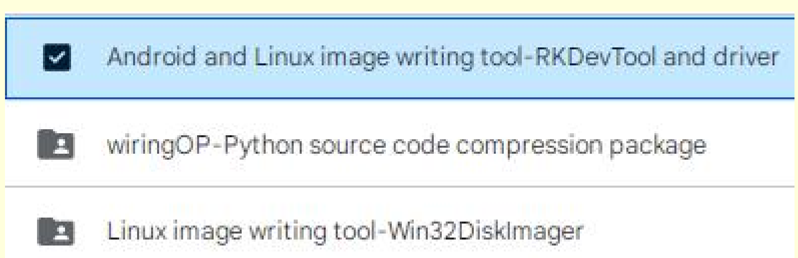


There is no nand-sata-install script in the OPi OS Arch system. Please use the following command to mirror u-boot to SPI Flash:

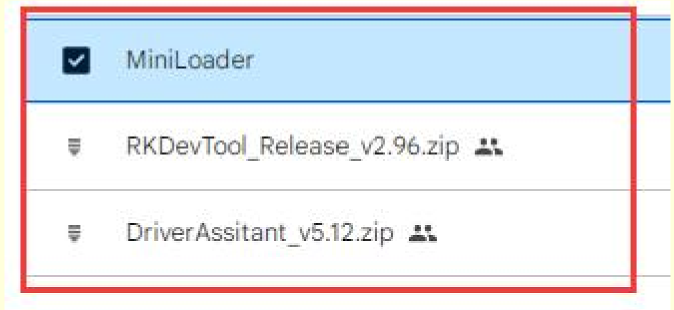
[orangepi@orangepi ~]$ **sudo dd if=/boot/rkspi_loader.img of=/dev/mtdblock0**

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 into SPI Flash. The download steps are as follows:

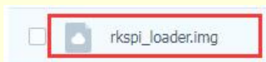
a. First, enter the download page of the development board, then select the official tool on the download page, and then go to the folder below



b. Then select to enter the directory below



c. Then download rkspi_loader.img



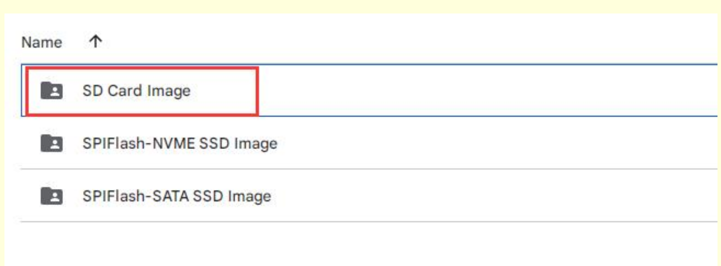
d. Then upload rkspi_loader.img to the Ubuntu, Debian, or OPi OS Arch system on the development board. Please refer to the instructions in the section on uploading files to the Linux system on the development board for the upload method.

Finally, execute the following command to burn the u-boot image to 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**

12) Then upload the Linux image file (downloaded from the official website as a Debian, Ubuntu, or OpenWRT image) to the TF card. Please refer to the instructions in the section on **uploading Linux image files to the development board Linux system** for the method of uploading files to the development board.

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



13) 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
Orangepi5max_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.

14) 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)
```

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

```
orangepi@Orangepi5max:~/Desktop$ sudo dd bs=1M if=/dev/zero of=/dev/nvme0n1 count=2000 status=progress
orangepi@Orangepi5max:~/Desktop$ sudo sync
```

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

a. The **if=** parameter in the following command should be followed by the full path where the Linux image is stored and 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, 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=Orangepi5max_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 compressed Linux image file ending in .7z, .xz, or .gz, please remember to decompress it before burning it with the dd command.

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

17) 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 unplug the TF card and press the power button briefly to start the Linux system in SPIFlash+NVMe SSD.

18) After starting the system in NVMe SSD, using the **df -h** command can show 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
```

19) When the TF card and NVMe SSD burn the same system, **if both the TF card and NVMe SSD are inserted in the development board, power on the development board and u-boot will start the system in the TF card first**. However, since the systems in the TF card and the NVMe SSD are identical, the UUID of the /**boot** partition and the **rootfs** partition in the two storage devices are also the same. This may cause the partition in the NVMe SSD to be loaded when the TF card starts. Running the following script can solve this problem.

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

An identical system means that the image name is exactly the same. Even with the Debian11 system, different versions are different.

The fix_mmc_ssd.sh script is not available in the OPi OS Arch system.

2. 6. 3. Method of burning using balenaEtcher software

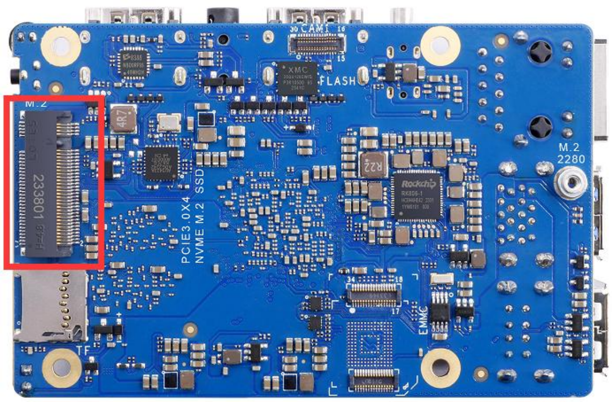
Please do not use this method for OPi OS Arch and OpenWRT systems.

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.

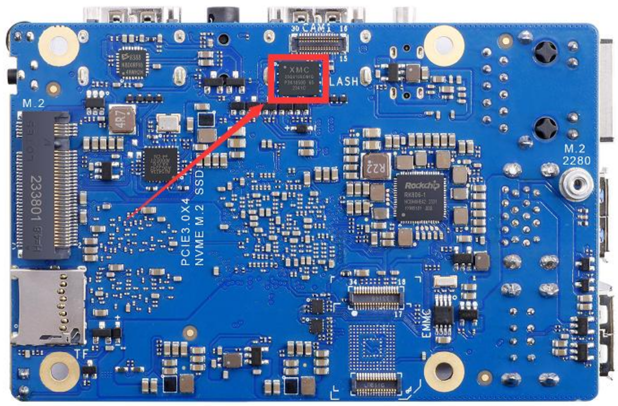
a. The M.2 2230 SSD is as follows

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**.

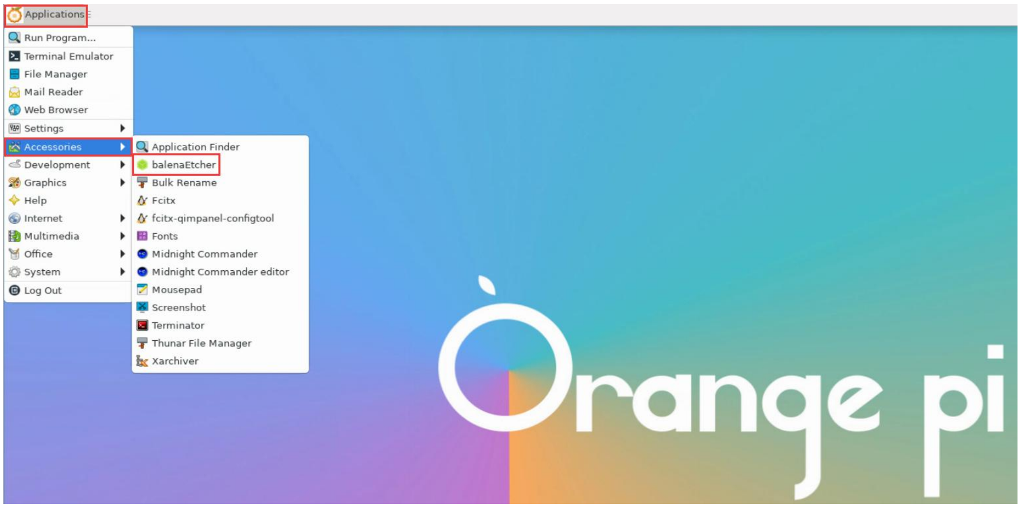
5) After booting into the linux system in the TF card, please confirm that the NVMe SSD has been properly recognized by the linux of the development board. If the NVMe SSD is recognized normally, use the **sudo fdisk -l** command to see **nvme**-related information.

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

Using the **lspci** command, you can see an NVMe related PCI device

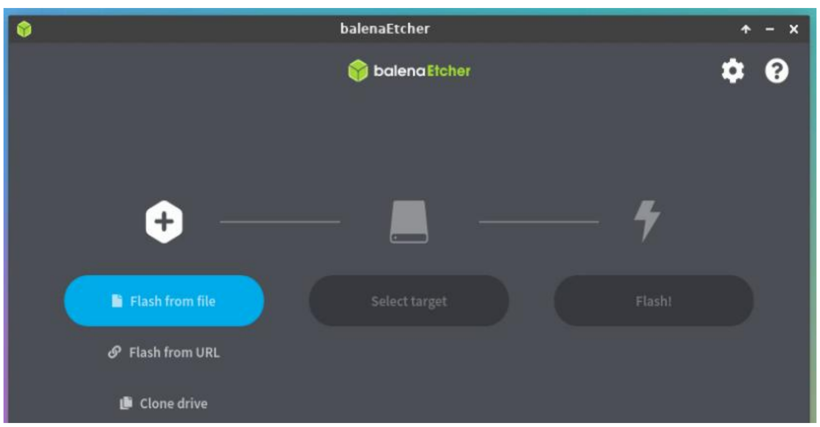
```
orangepi@orangepi:~/Desktop$ lspci
0004:40:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3588 (rev 01)
0004:41:00.0 Non-Volatile memory controller: MAXIO Technology (Hangzhou) Ltd. NVMe SSD Controller MAP1202 (rev 01)
```

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



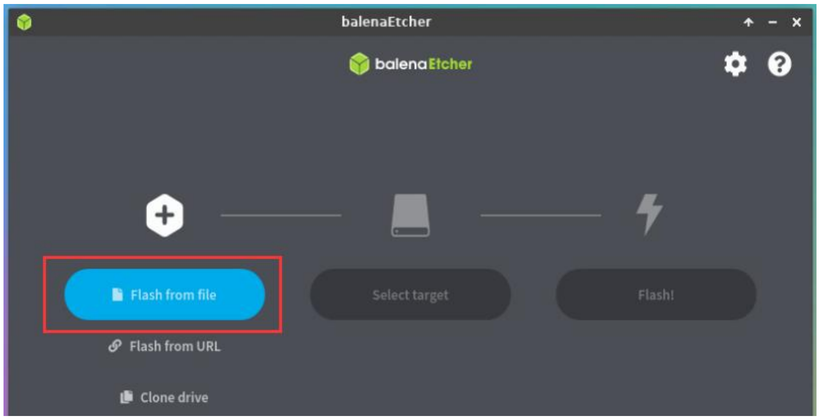
If it is not pre installed, please refer to the instructions in the section on downloading and installing the arm64 version of balenaEtcher.

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

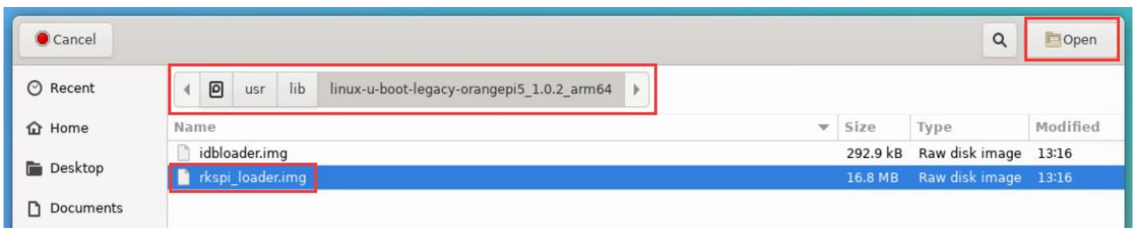


8) The method of burning u-boot to SPI Flash on the development board using BalenaEtcher is as follows:

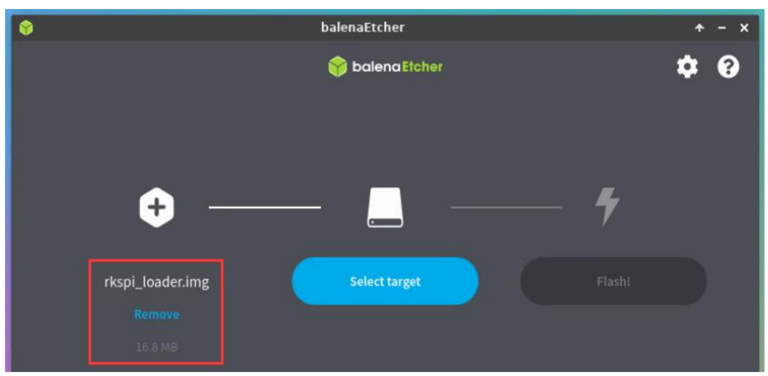
a. Firstly, open the balenaEtcher software and click on **Flash from file**



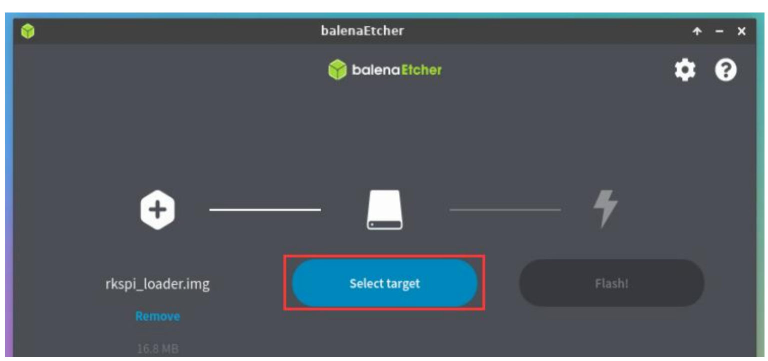
b. Then go to the **/usr/lib/linux-u-boot-legacy-orangepi5max_1.0.0_arm64/** directory, select **rkspi_loader.img**, and click **Open** to open it



c. The interface after opening **rkspi_loader.img** is shown below:

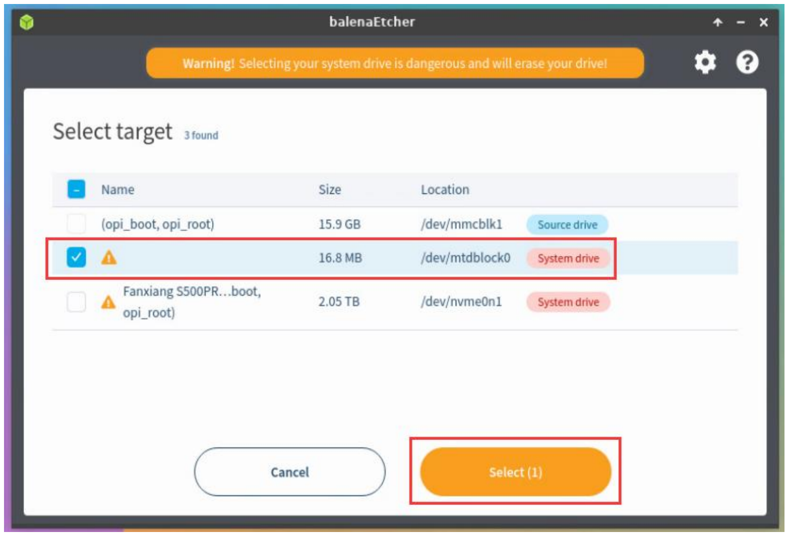


d. Then click **Select target**

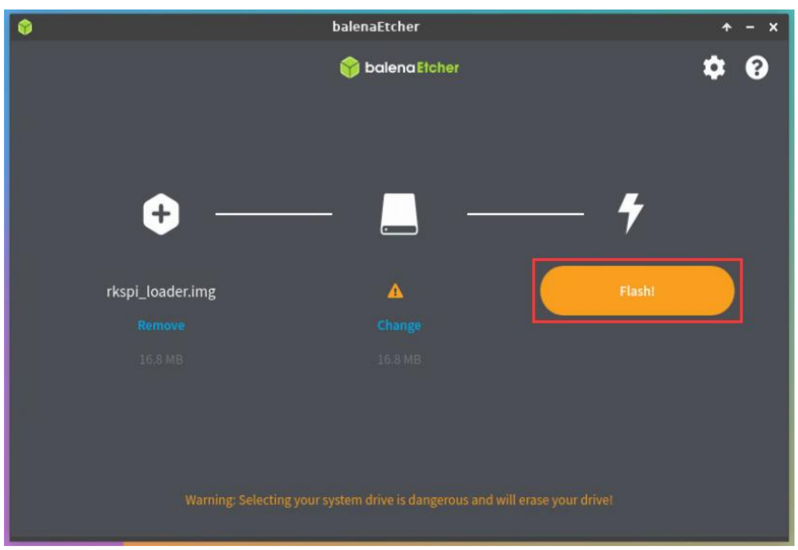


e. Then click on **Show 2 hidden** to open more storage device options

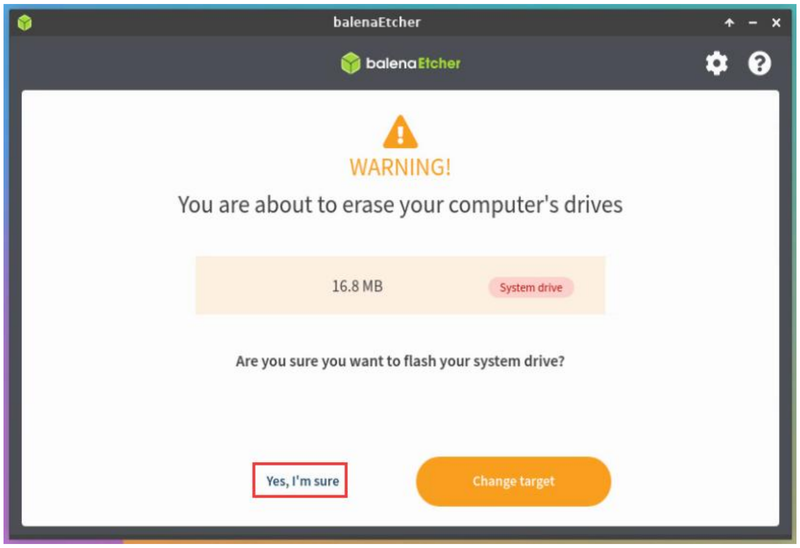
f. Then select the device name **/dev/mtdblock0** for SPI Flash, and click **Select**



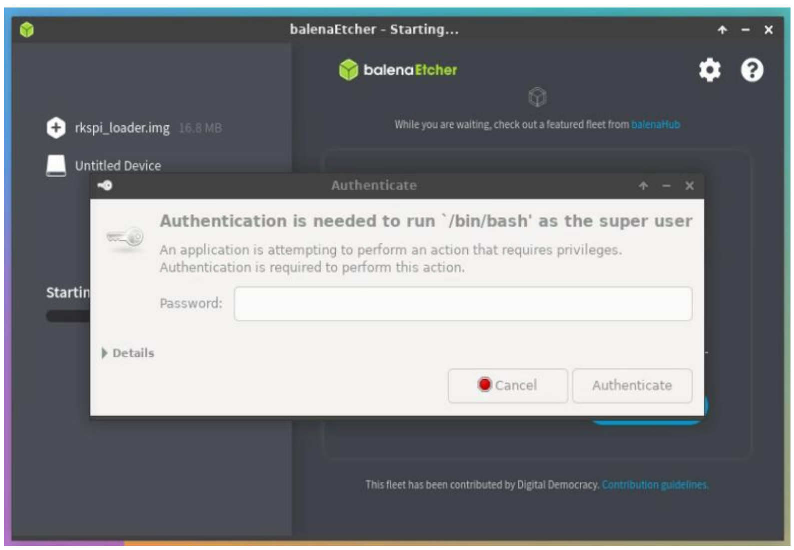
g. Then click on **Flash**



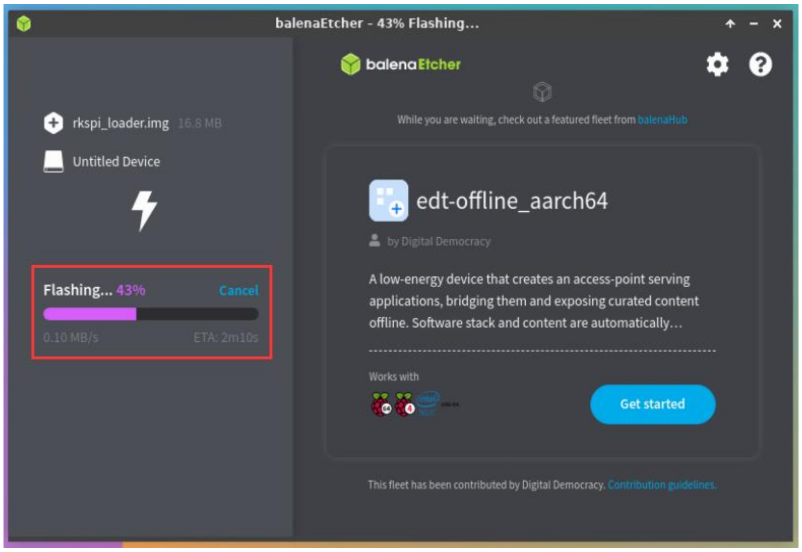
h. Then click **Yes, I'm sure**



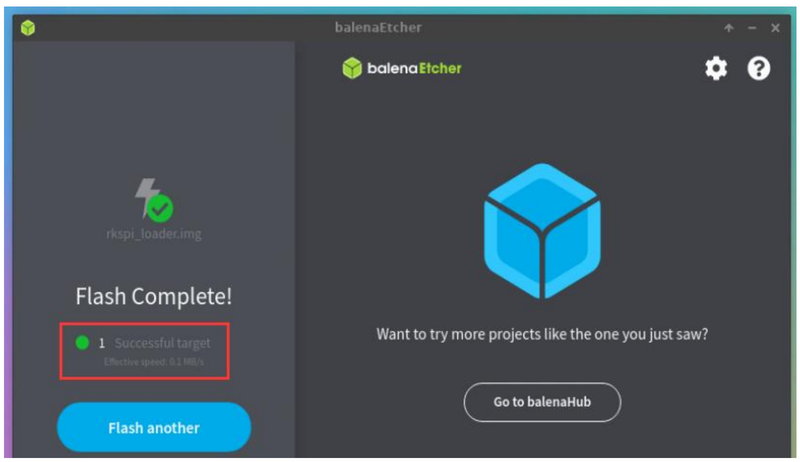
i. Then enter the password **orangepi** for the Linux system on the development board, and the u-boot image will be burned into SPI Flash



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

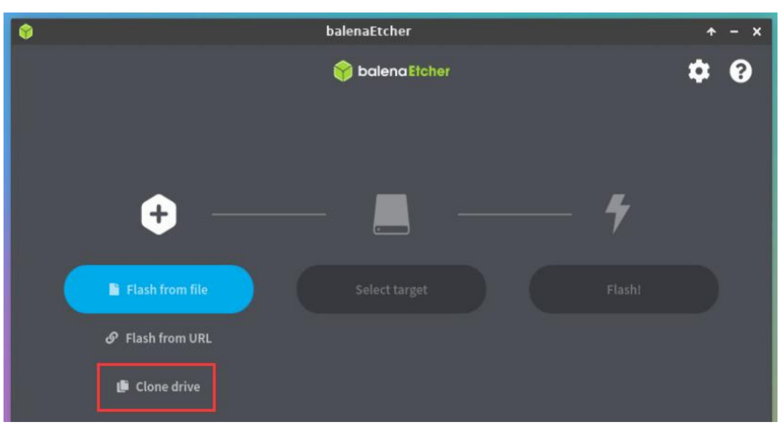


k. The display after burning is as follows:

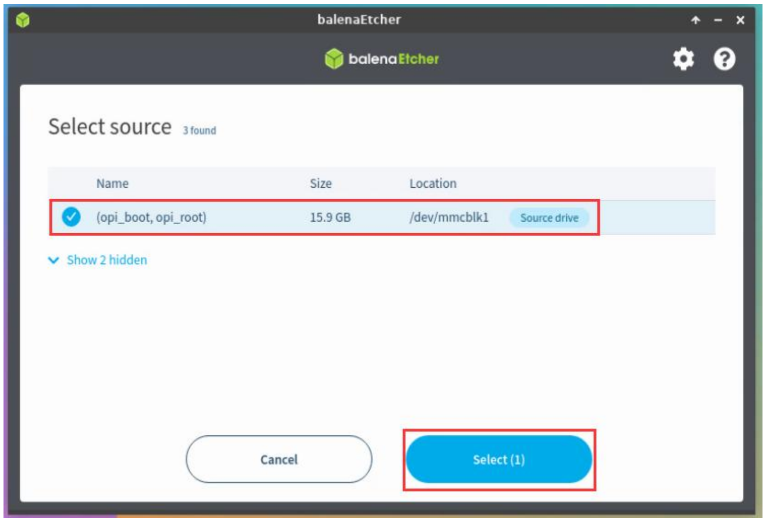


9) The method of burning Linux system from TF card to NVMe SSD (this method is equivalent to cloning the system from TF card to NVMe SSD)

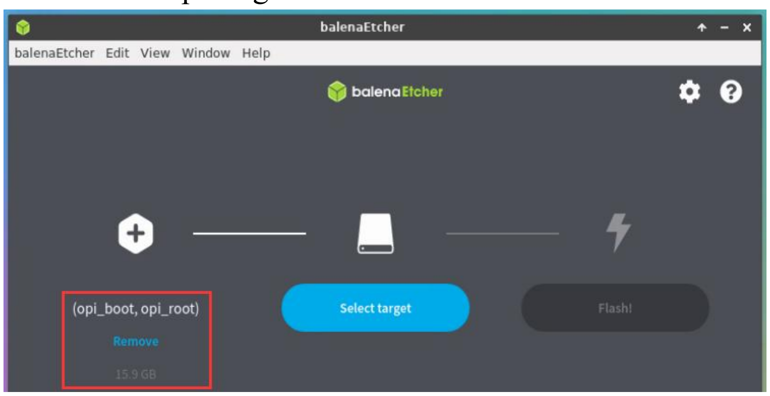
a. First click **Clone drive**



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

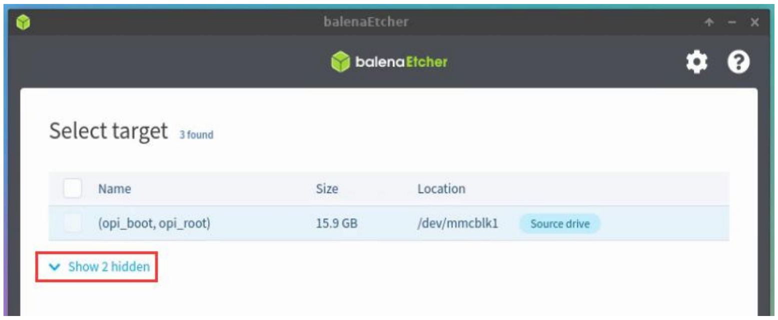


c. The interface after opening the TF card is shown below:

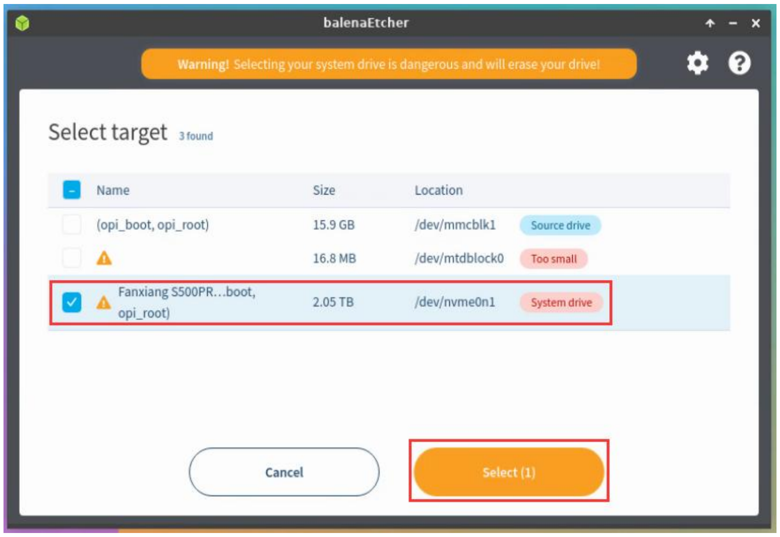


d. Then click **Select target**

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

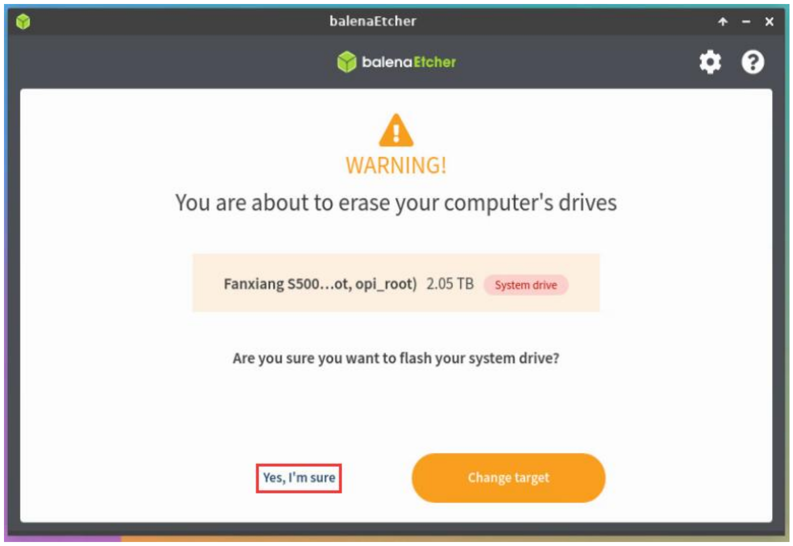


f. Then select the device name **/dev/nvme0n1** for NVMe SSD, and click **Select**

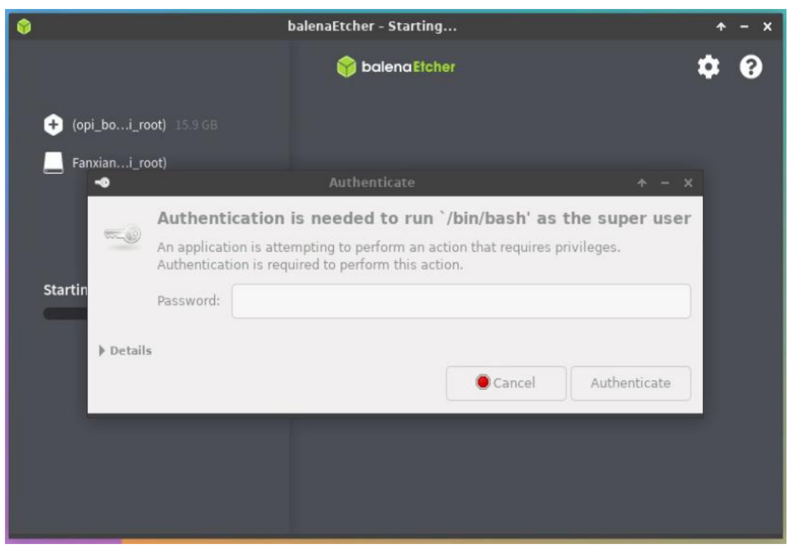


g. Then click **Flash**

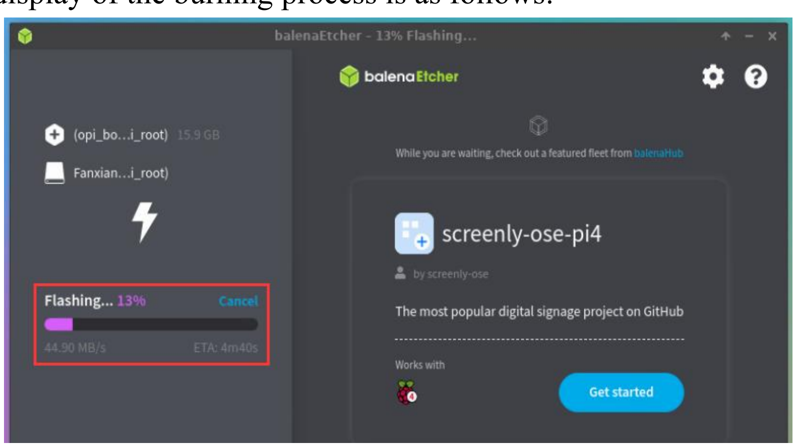
h. Then click **Yes, I'm sure**

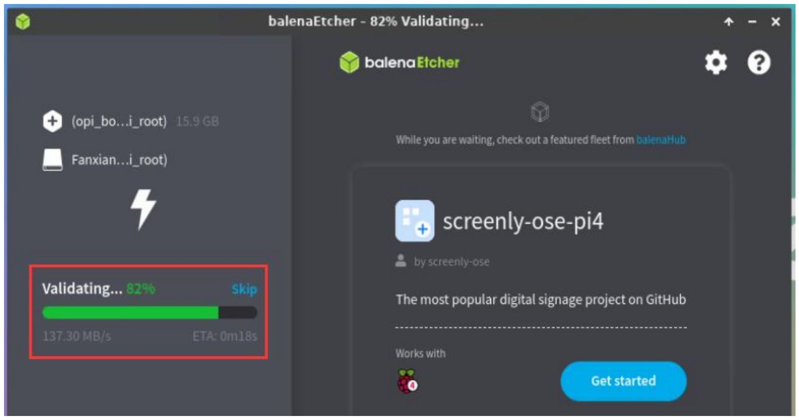


i. Then enter the password orangepi for the Linux system on the development board, and the Linux image will be burned to the SSD

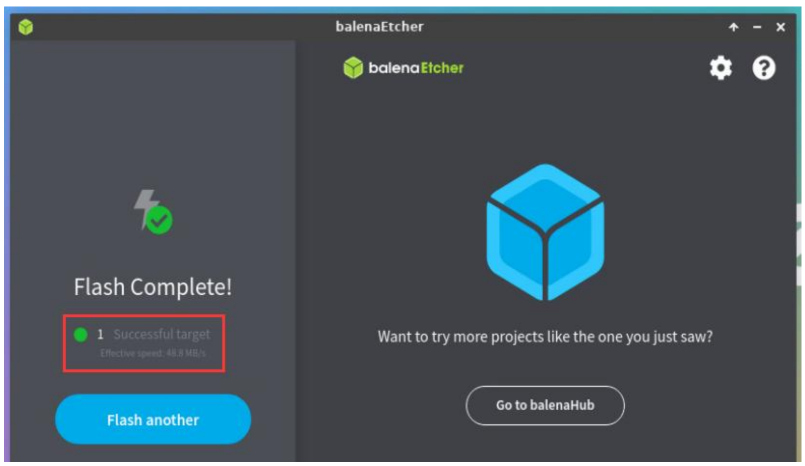


j. The display of the burning process is as follows:





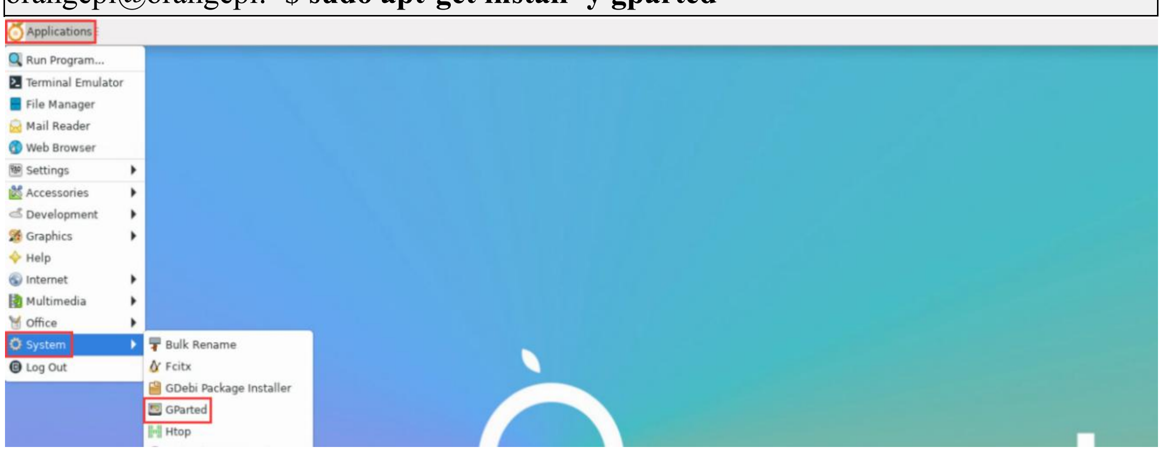
k. The display after burning is as follows:



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

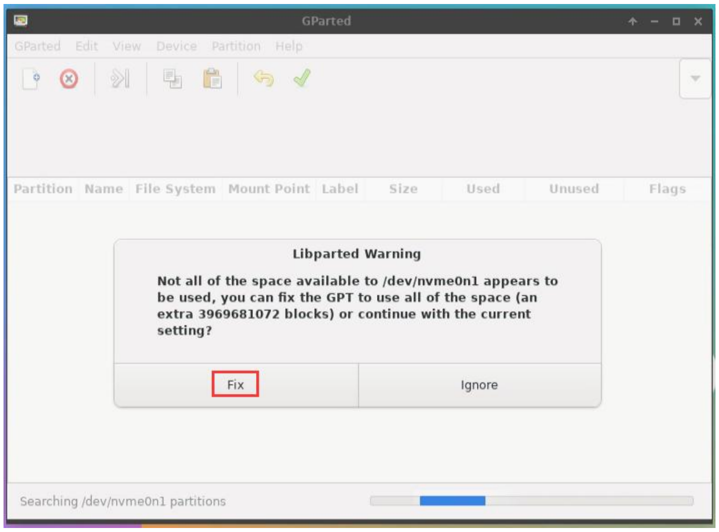
a) First, open **GParted**. If GParted is not pre installed on the system, please use the apt command to install it

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

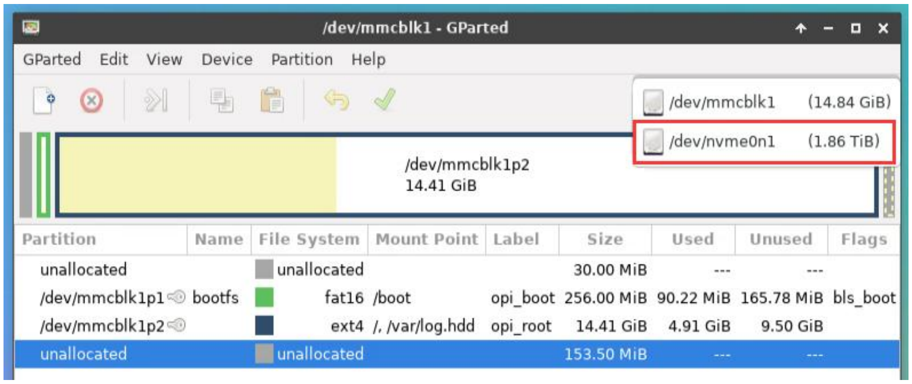


b) Then enter the linux system password orangepi, and click **Authenticate**

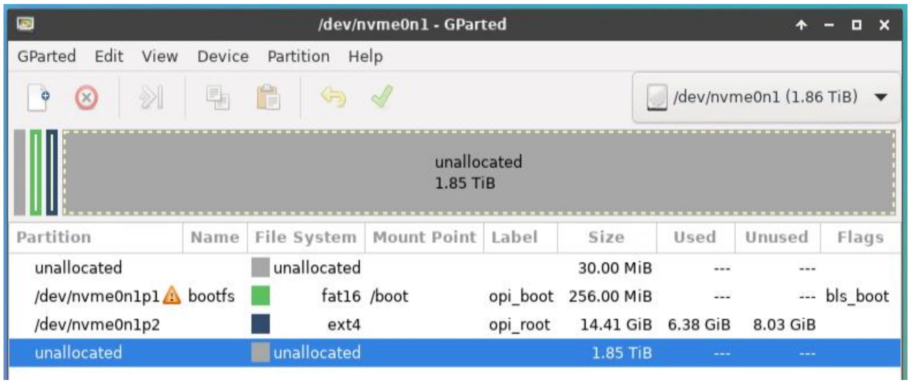
c) Then click **Fix**



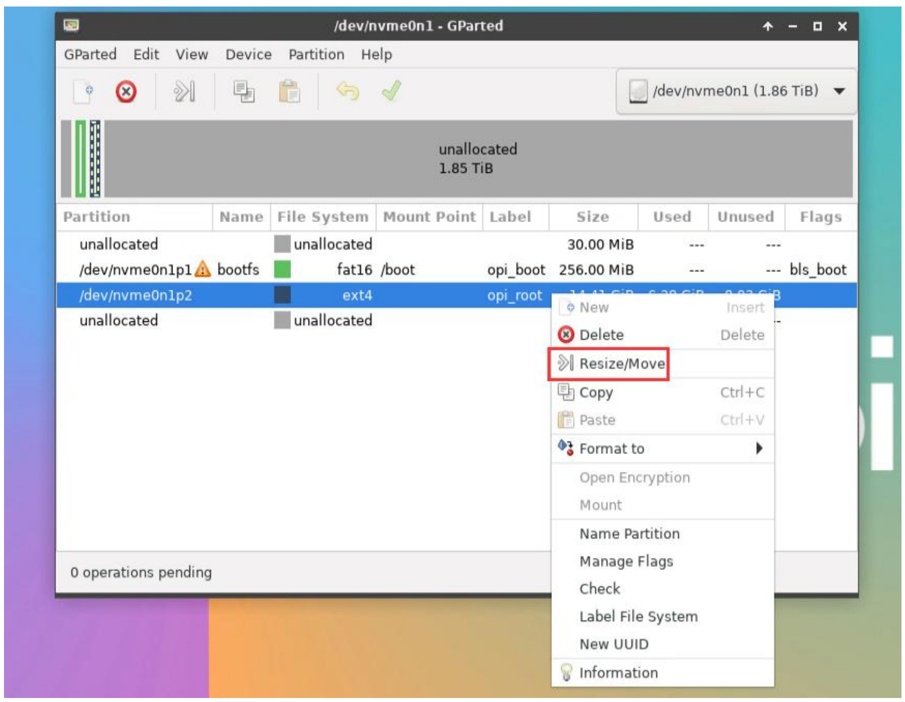
d) Then click NVMe SSD



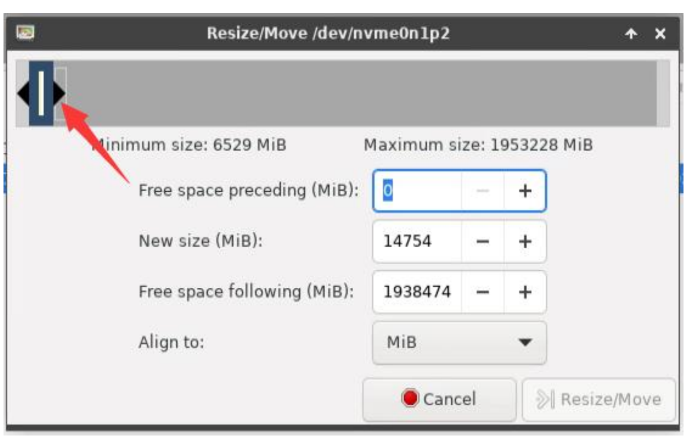
e) The display interface after selecting NVMe SSD is shown below:



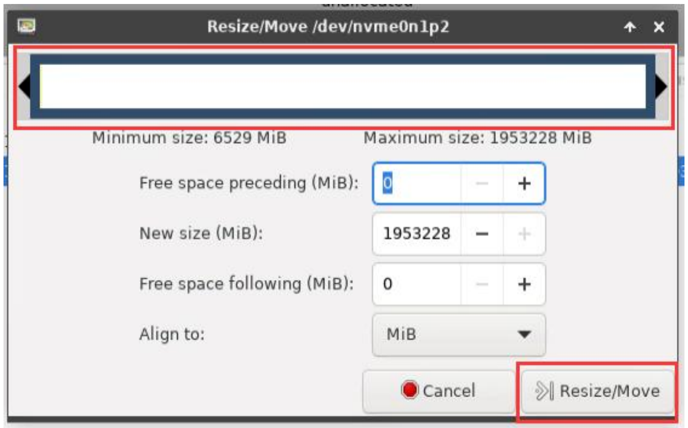
f) Then select the **/dev/nvme0n1p2** partition, right-click and choose **Resize/Move**



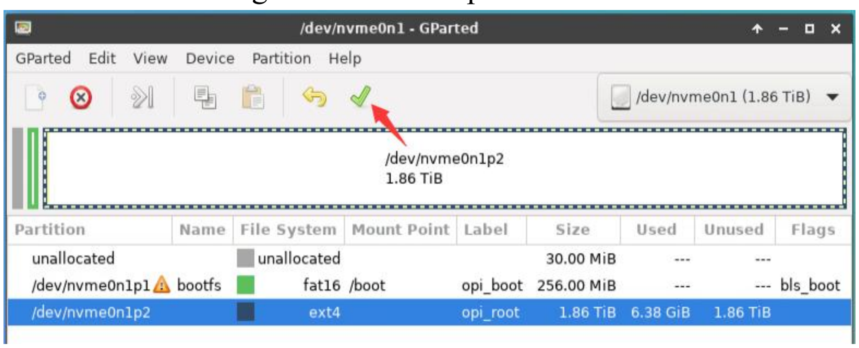
g) Then drag the capacity to its maximum at the position shown in the figure below



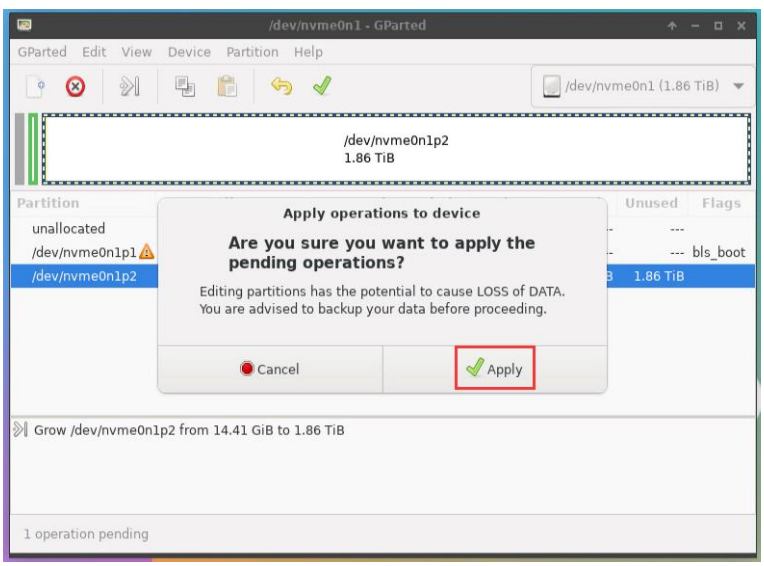
h) Then click **Resize/Move**



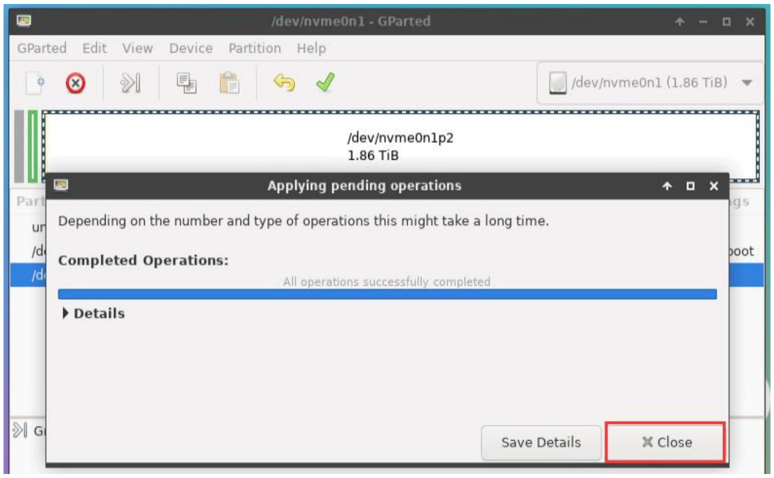
i) Then click on the green √ in the position shown below



j) Click **Apply** again



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 press the power button briefly to start the linux system in SPIFlash+NVMe SSD.

10) Step **9)** is to clone the system from the TF card to the NMVe SSD. We can also directly burn the Linux image file to the NVMe SSD. Here are the general steps:

a. Upload the linux image file to the linux system on 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. The first startup will automatically expand it.**

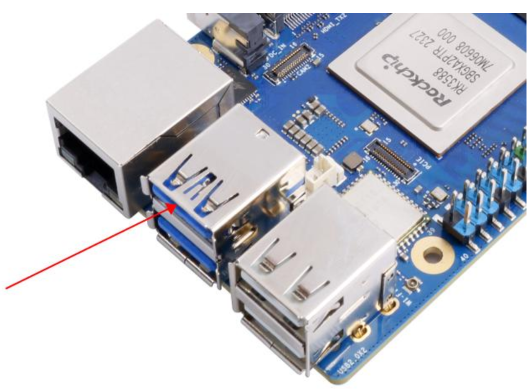
2.7. Method for burning Linux images to SPIFlash+USB storage devices

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 data download page.

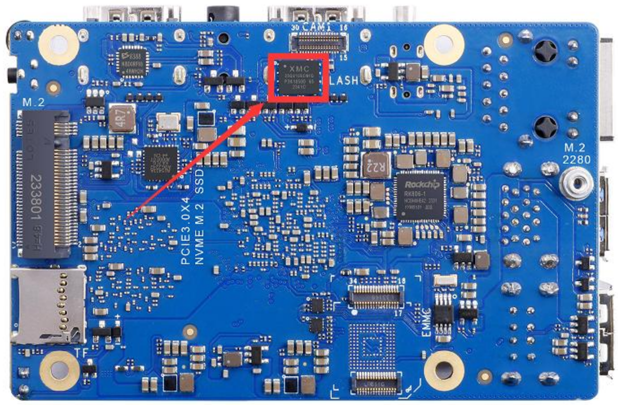
1) Firstly, it is necessary to prepare a USB storage device, such as a USB drive

2) Then please refer to the instructions in 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** to burn Linux images to USB storage devices. There is no difference between burning a Linux image to a USB storage device and burning a Linux image to a TF card (when the TF card is inserted into the card reader, the reader is actually equivalent to a USB flash drive)

3) Then insert the USB storage device that has burned the Linux system into the USB interface of the development board. **Note that only the three USB 2.0 interfaces shown in the following figure support booting the Linux system, and the blue USB 3.0 interface does not support it**



4) 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



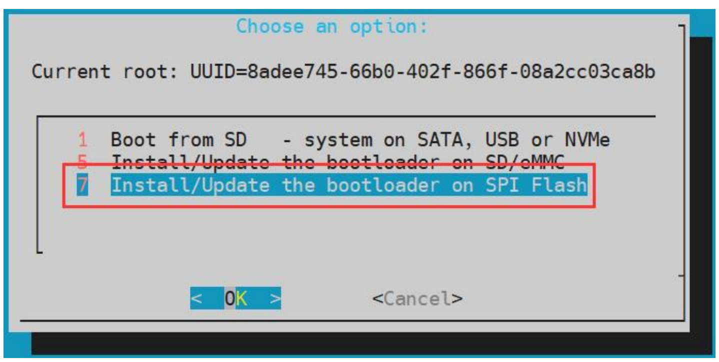
5) Burning the u-boot image to SPIFlash requires the use of a TF card, so the first step is to burn the Linux image onto the TF card, and then use the TF card to boot the development board into the Linux system. The method of burning a Linux image to a TF card can be found in the two sections: **the method of burning a Linux image to a TF card based on Windows PC** and **the method of burning a Linux image to a TF card based on Ubuntu PC**.

6) After starting the Linux system with a TF card, you can burn the u-boot image to SPI Flash

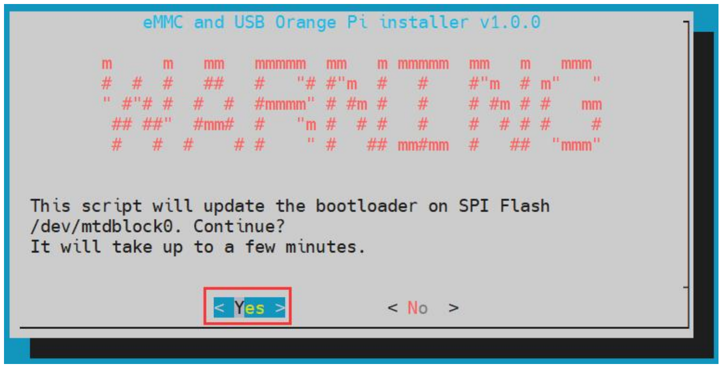
a. First, run **nand-sata-install**. **Regular users should remember to grant sudo privileges**

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

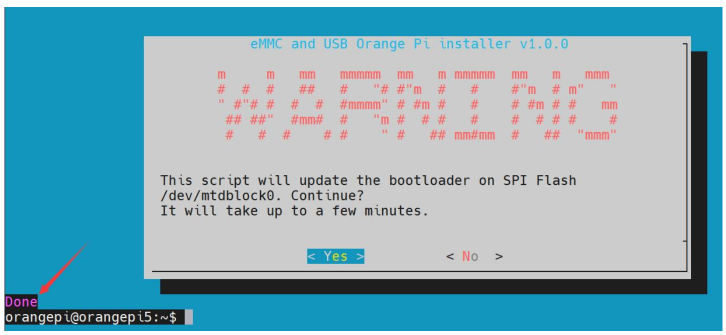
b. Then choose **7 Install/Update ther bootloader on SPI Flash**



c. Then choose **<Yes>**



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

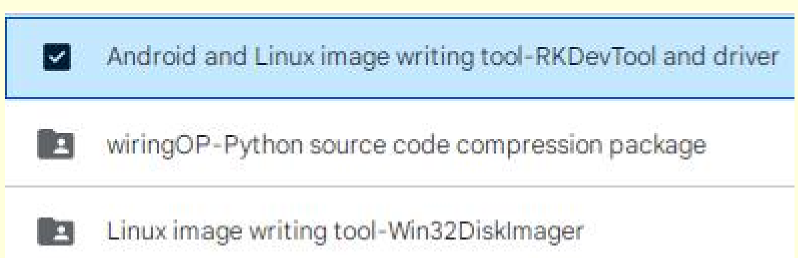


There is no nand-sata-install script in the OPi OS Arch system. Please use the following command to mirror u-boot to SPI Flash:

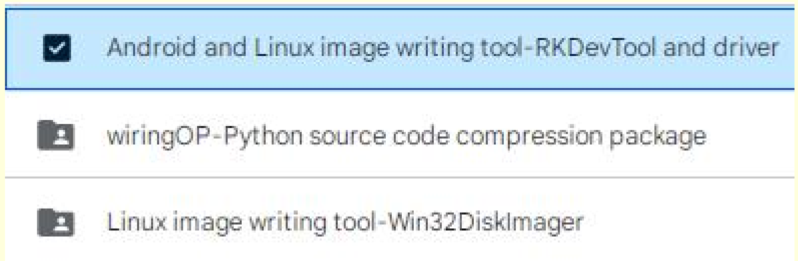
[orangepi@orangepi ~]$ **sudo dd if=/boot/rkspi_loader.img of=/dev/mtdblock0**

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 into SPI Flash. The download steps are as follows:

a. First, enter the download page of the development board, then select the official tool on the download page, and then go to 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 on the development board. Please refer to the instructions in the section on uploading files to the Linux system on the development board for the upload method.

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

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

7) At this point, you can use the **poweroff** command to shut down. Then please unplug the TF card and press the power button briefly to turn on the linux system in the SPIFlash+USB storage device

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

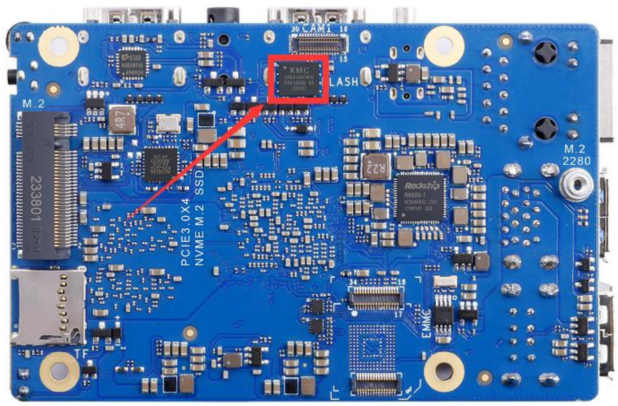
```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G  8.0K  3.8G   1% /dev
tmpfs           769M  588K  769M   1% /run
/dev/sda2        15G  1.6G   13G  11% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock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

2. 8. Method for burning OpenWRT images to SPI FLASH

The method introduced in this section is to burn the entire OpenWRT image into SPI Flash, without the need for an SSD or USB drive. That is to say, u-boot, kernel, and rootfs are all stored in SPI Flash.

2. 8. 1. Method of burning using RKDevTool

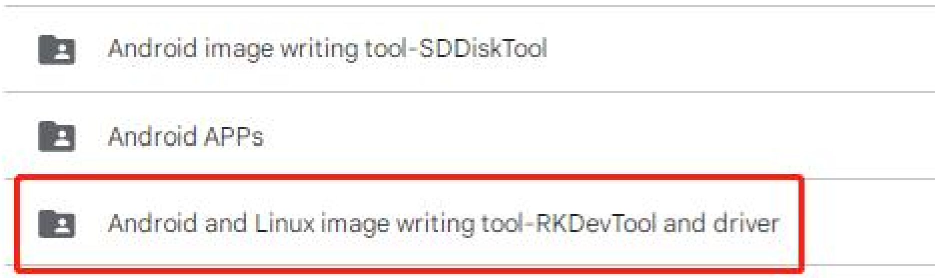
1) The position of SPI Flash on the development board is shown in the following figure, and no other settings are required before starting the burning process



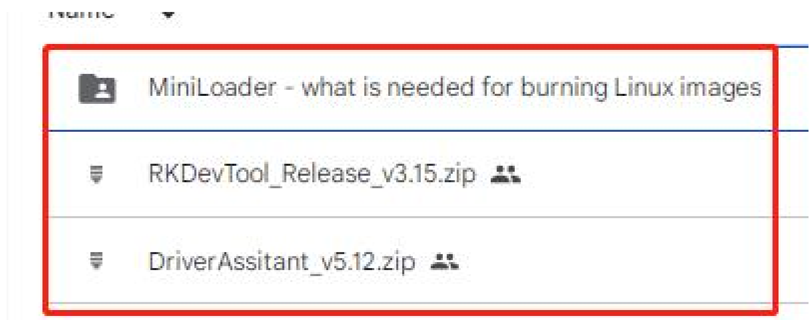
2) Then we need to prepare a high-quality USB male to male data cable

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

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

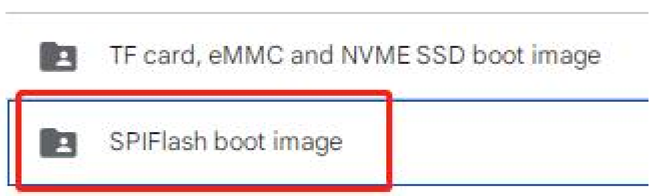


b. Then download all the files below

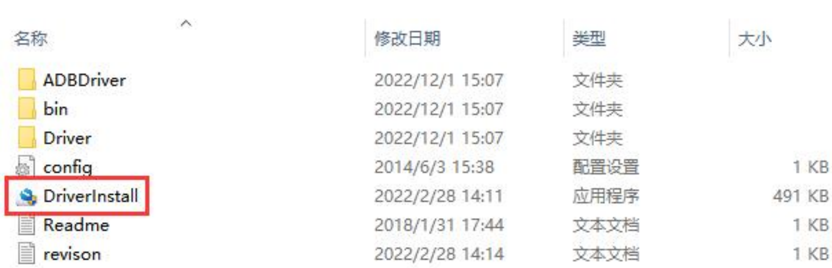


Note that the "MiniLoader-things needed to burn Linux images" folder will be referred to as the MiniLoader folder below.

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.

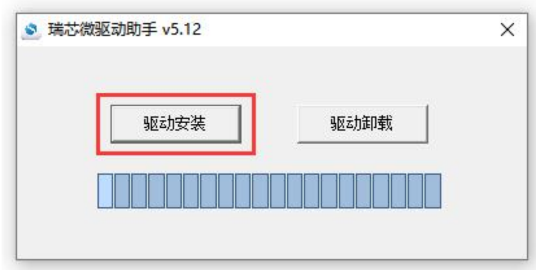


5) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

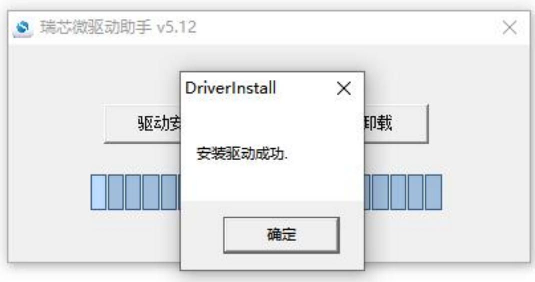


6) Open **DriverInstall.exe** and install the Rockchip microdriver as follows:

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

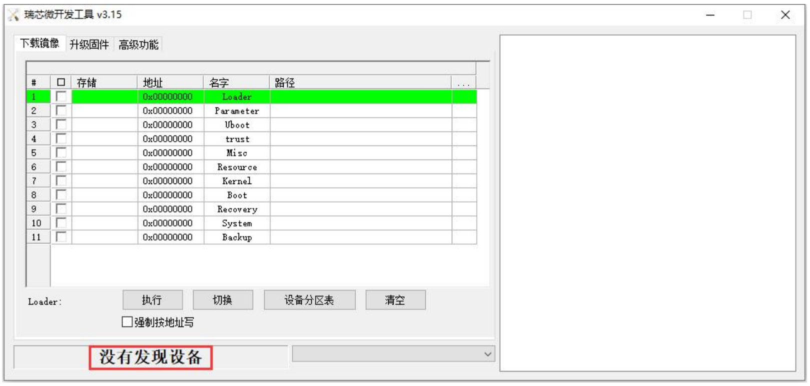


b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



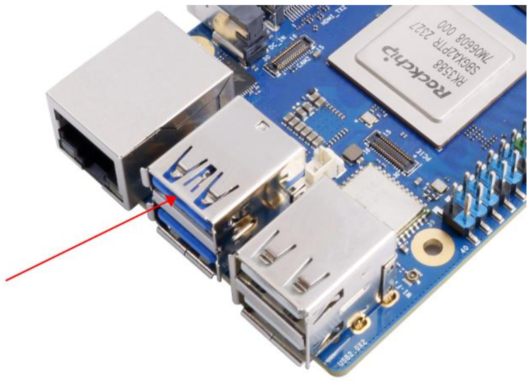
7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

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.



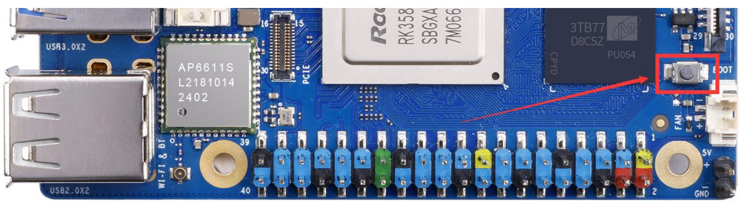
9) Then start burning the OpenWRT image into SPI FLASH

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.

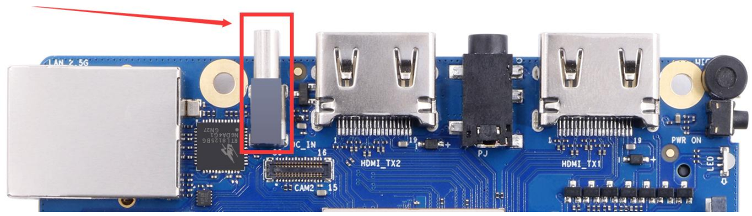


b. Make sure the development board is not connected to the power supply and the TF card is not inserted.

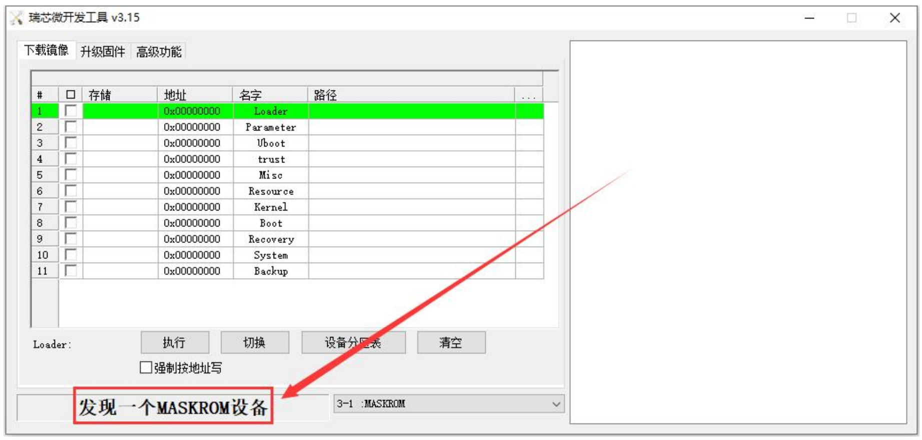
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



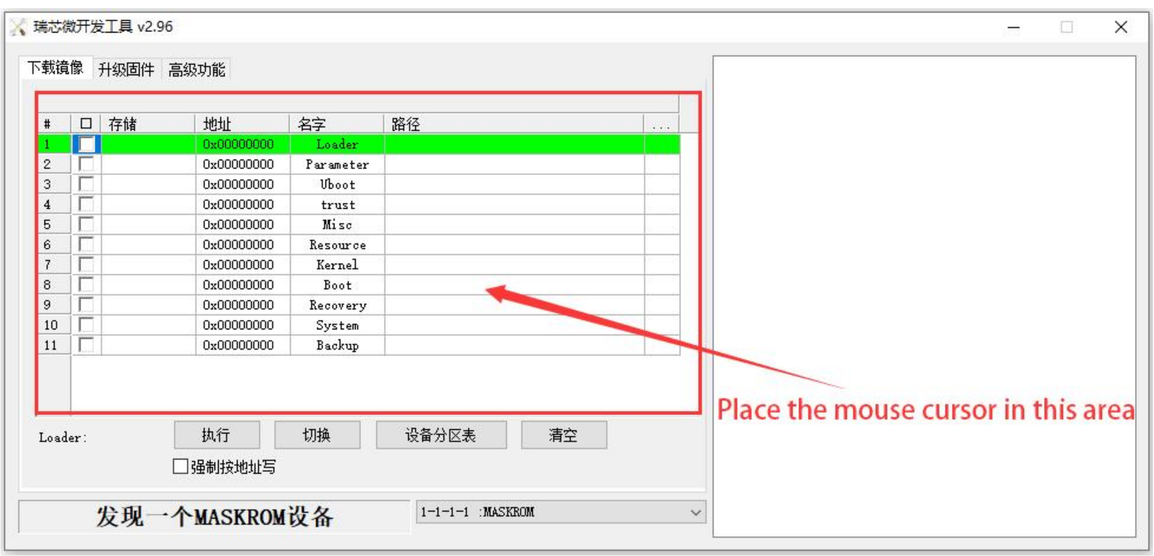
d. Then connect the development board to the power supply of the Type-C interface, power it on, and then release the MaskROM button.



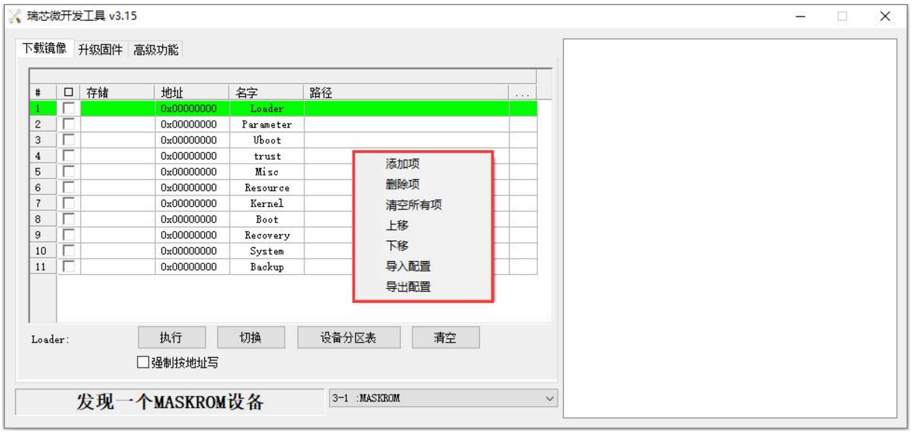
e. If the previous steps go well, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device was found**"



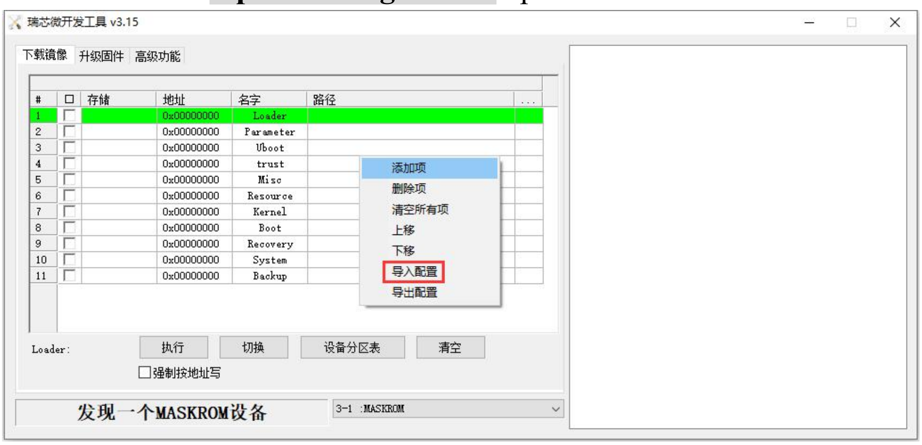
f. Then place the mouse cursor in the area below



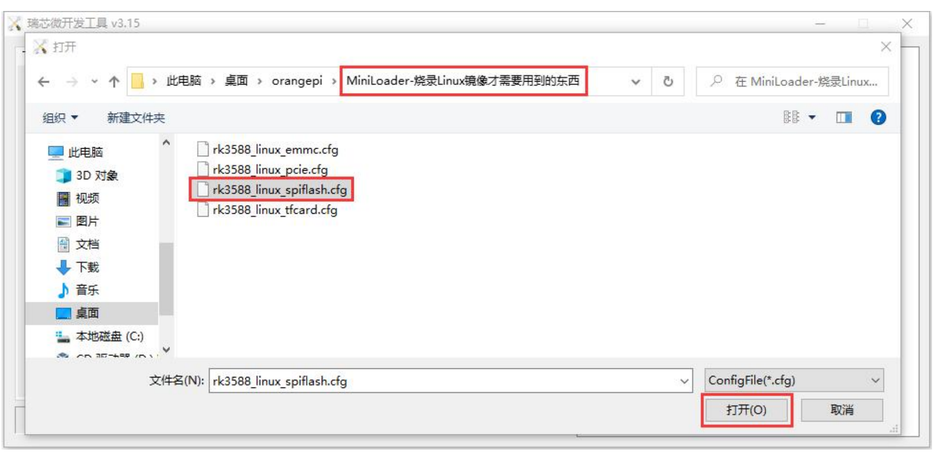
g. Then click the right button of the mouse and the selection interface shown below will pop up.



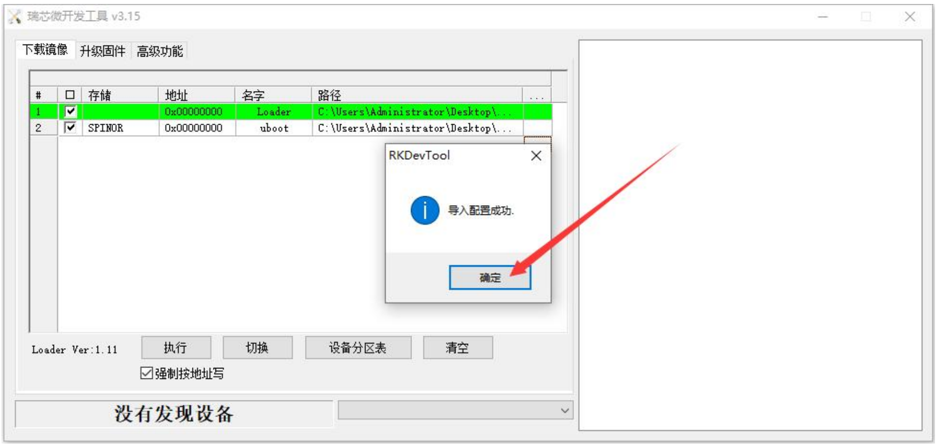
h. Then select the **Import Configuration** option



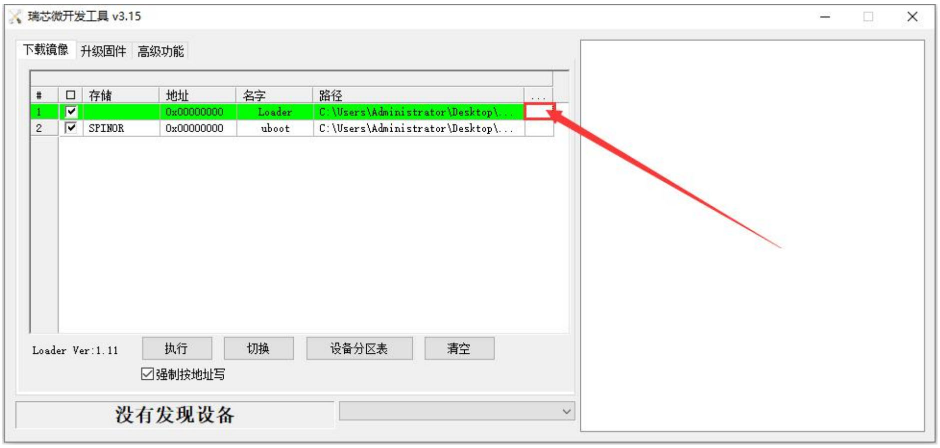
i. Then select the **rk3588_linux_spiflash.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and then click **Open**



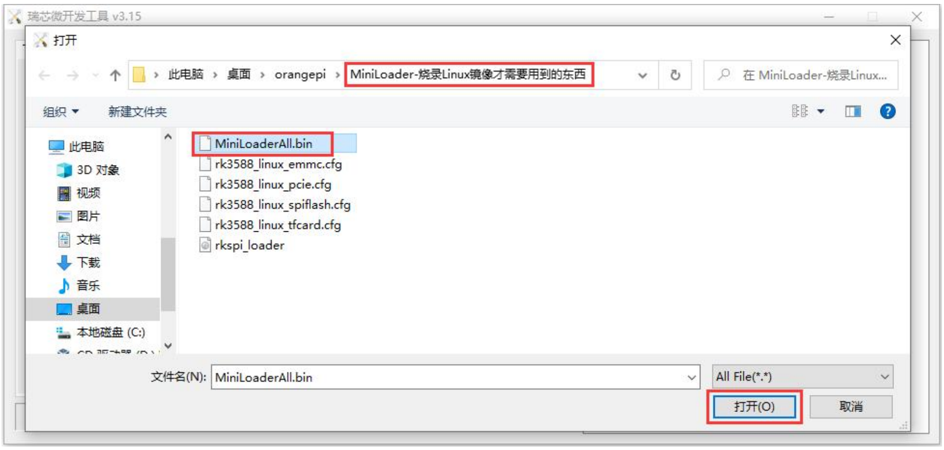
j. Then click **OK**



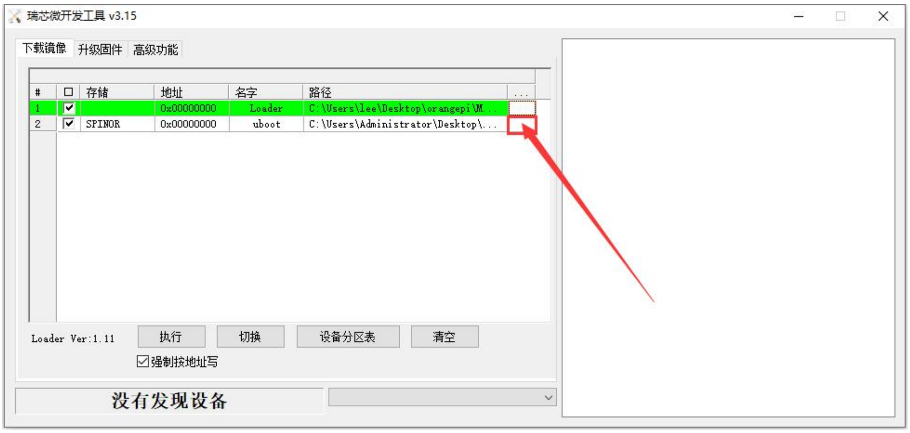
k. Then click the location shown in the picture below



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

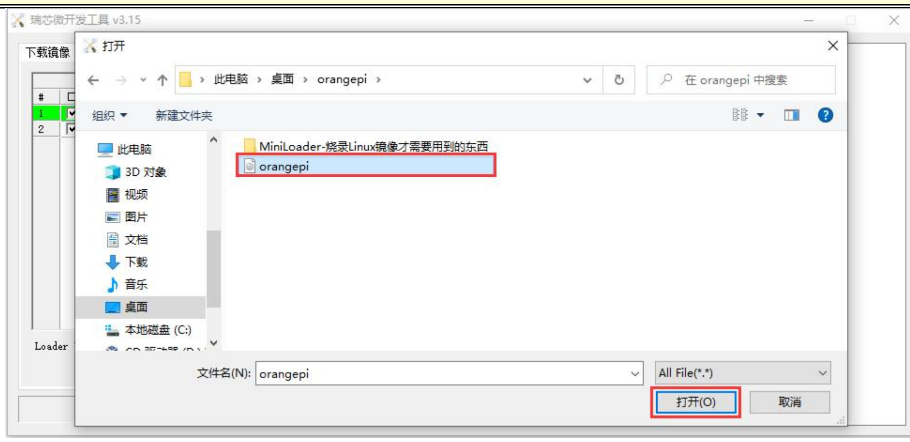


m. Then click the location shown in the picture below

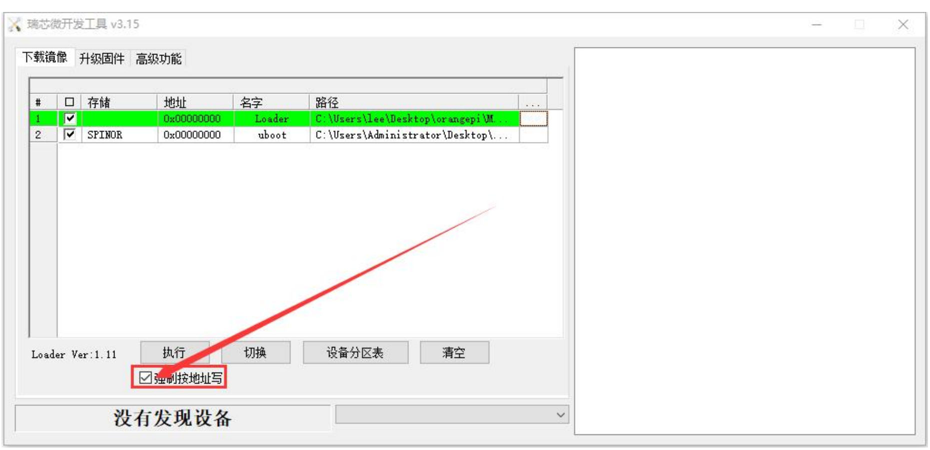


n. Then select the path of the OpenWRT image you want to burn, and then click **Open**

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.

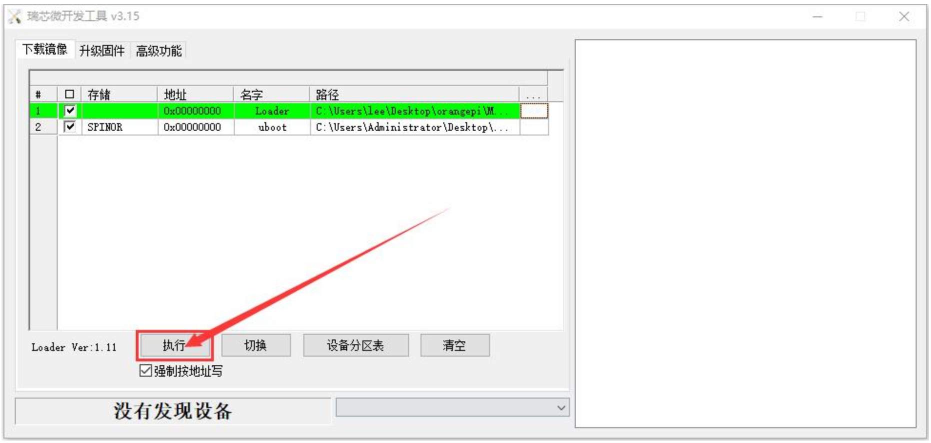


o. Then make sure the **Force writing by address** option is checked.

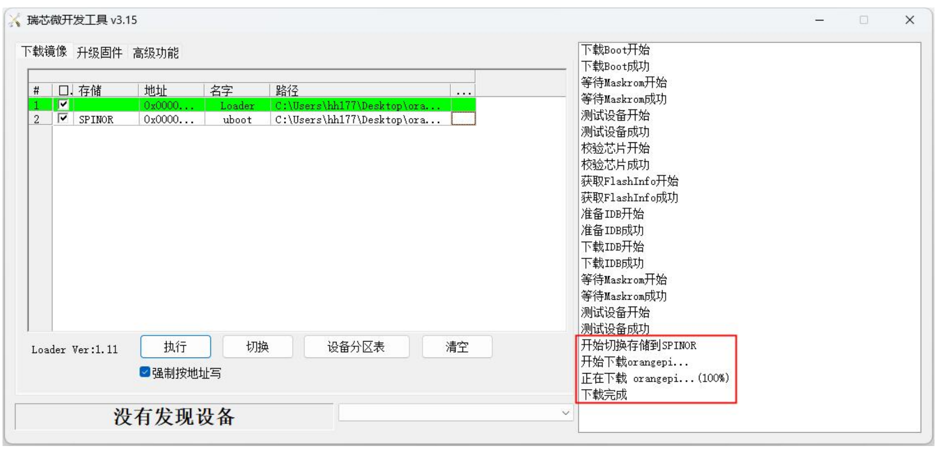


p. Click the **Execute** button again to start burning the OpenWRT image into

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.

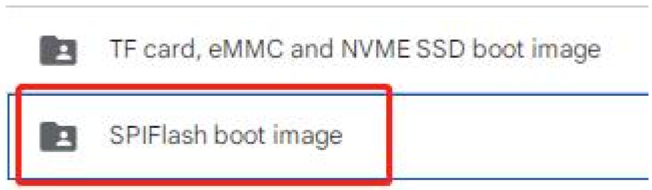
2. 8. 2. Burning method using dd command

1) Burning OpenWRT images to SPIFlash requires the use of a TF card, so the first step is to burn OpenWRT images that support TF card startup onto the TF card, and then use the TF card to boot the development board into the OpenWRT system. The methods for burning OpenWRT images to TF cards are described in 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**.

2) Then download the **OpenWRT** image that can be launched from **SPIFlash** from the

Orange Pi's download page.

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



3) Then upload the image downloaded from the official website to the TF card.

4) Then execute the following command to burn the OpenWRT image to SPIFlash. **Note that if= needs to be followed by specifying the actual path where the image is stored**

```
root@OpenWrt:~# dd if=openwrt-rockchip-armv8-xunlong_orangepi-5-max-spi-squashfs-sysupgrade.bin of=/dev/mtdblock0
```

5) Then you can use the **poweroff** command to shut down. Then please unplug the TF card and press the power button briefly to turn on the OpenWRT system in SPIFlash

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

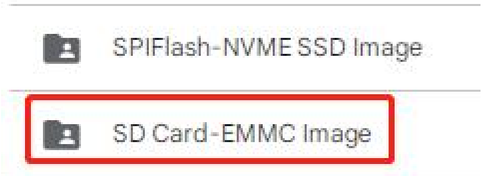
2. 9. 1. Method of burning using RKDevTool

1) Firstly, it is necessary to prepare a high-quality USB 2.0 male to male data cable

2) Then download the Rockchip micro driver **DriverAssitant_v5.12.zip** and the burning tool **RKDevTool_Release_v3.15.zip** from the **Orange Pi's download page**

3) Then download the Android image from the **Orange Pi's data download page**. After opening the download link for the Android image, you can see the following two types of Android images. Please select the image in the **SD card and eMMC startup image**

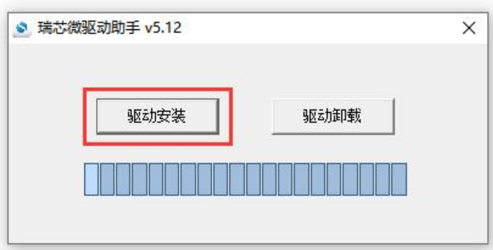
folder to download



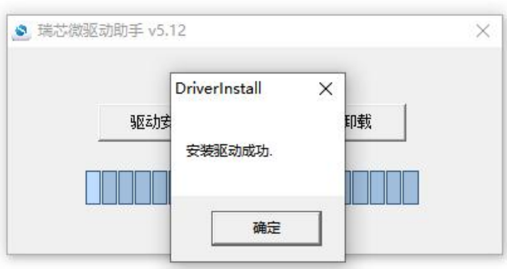
4) Then use decompression software to extract the **DriverAssitant_v5.12.zip** file, and then find the **DriverInstall.exe** executable file in the extracted folder and open it

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.

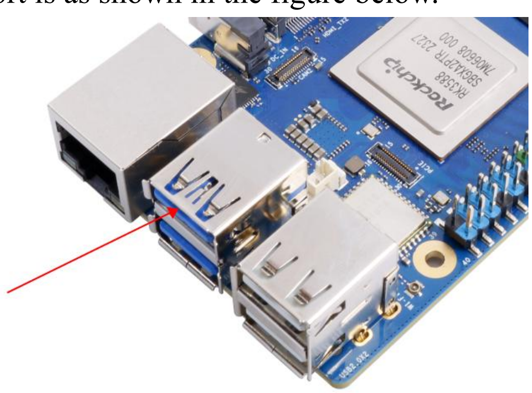


6) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

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

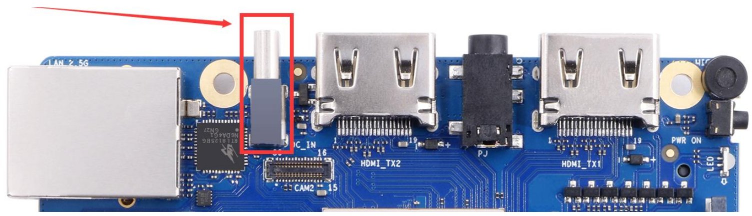
a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the development board's USB programming port is as shown in the figure below.



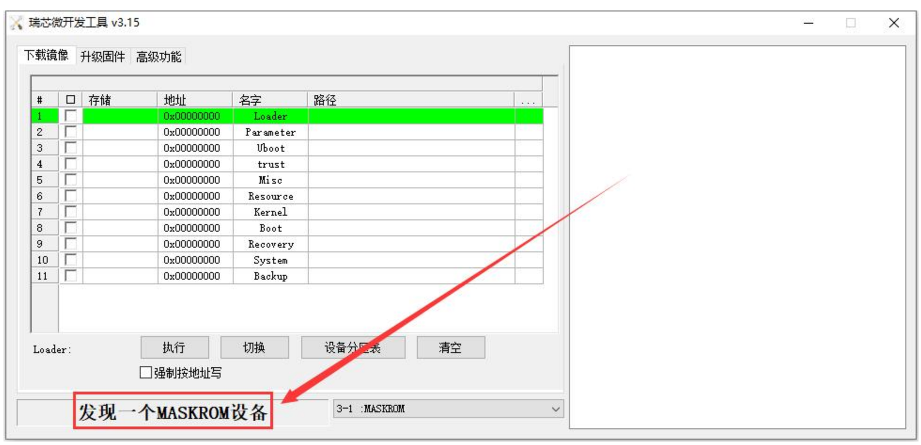
b. Then insert the TF card into the development board and make sure that the development board is not connected to the power supply.

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:

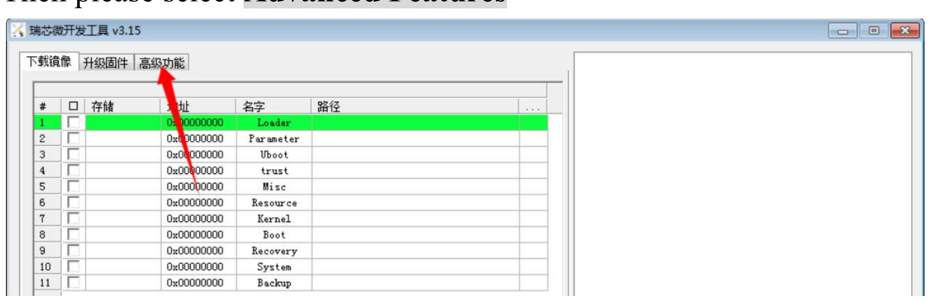
d. Then connect the development board to the power supply of the Type-C interface, power it on, and then release the MaskROM button.



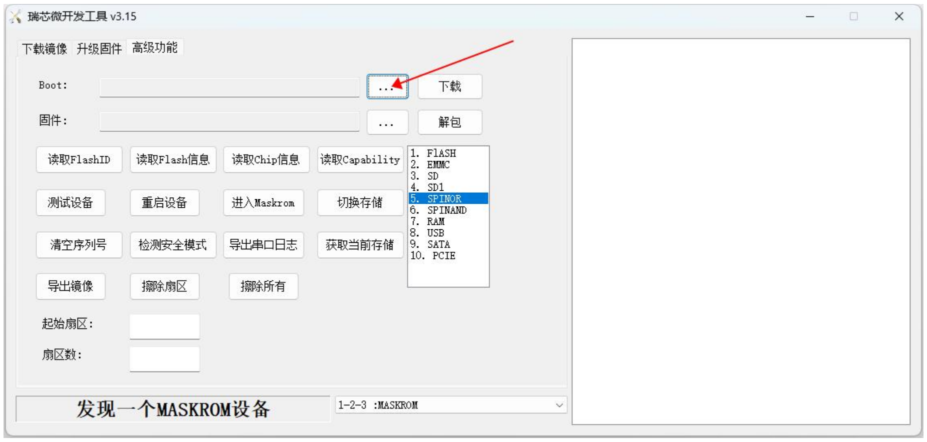
e. If the previous steps go well, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device was found**"



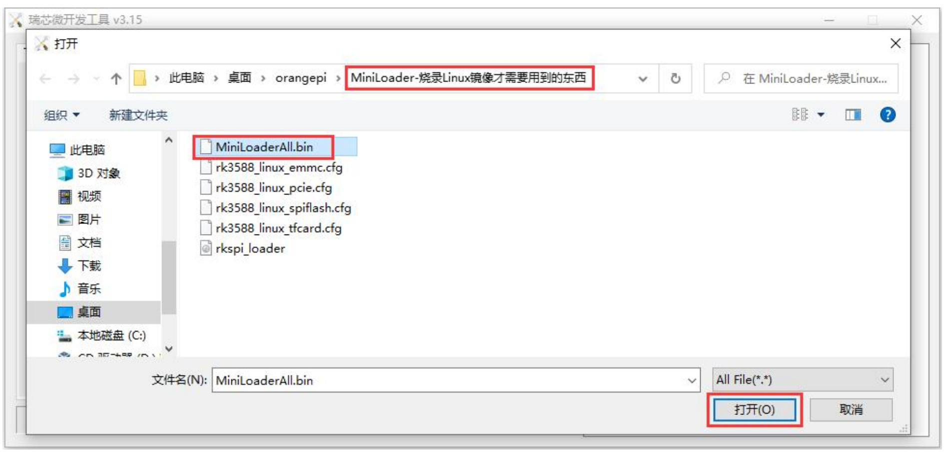
f. Then please select **Advanced Features**



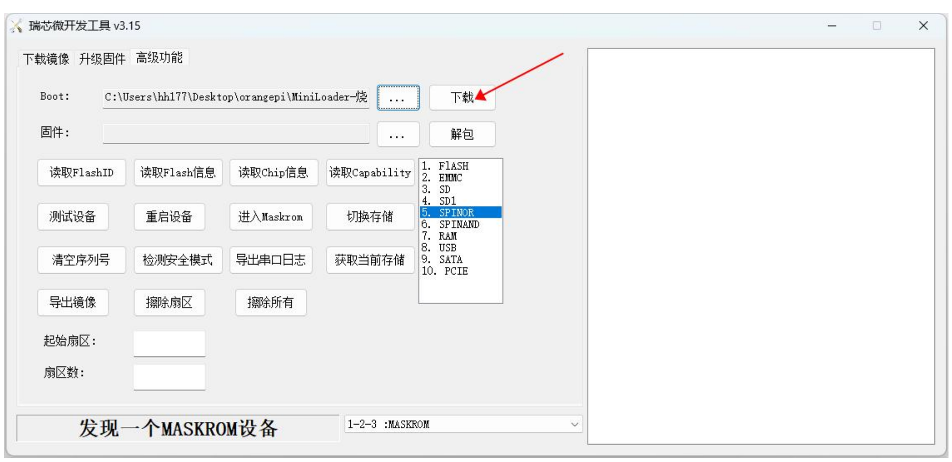
g. Then click the location shown in the picture below



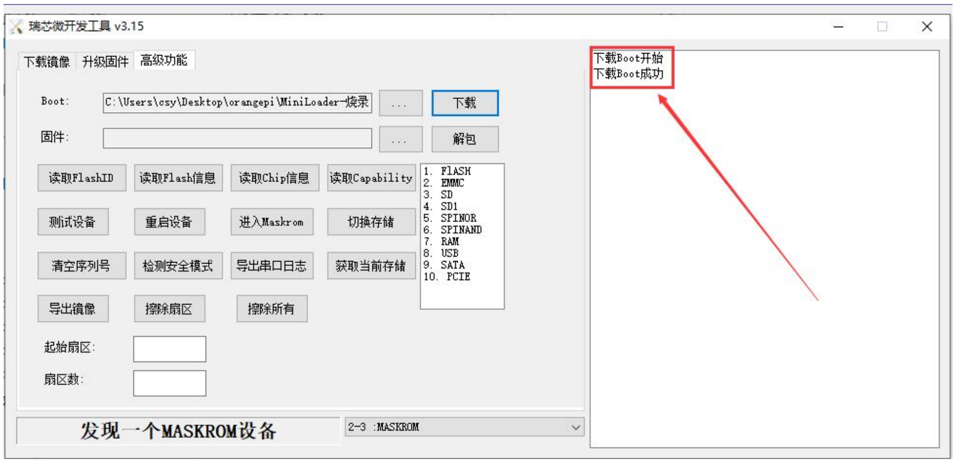
h. Then select **MiniLoaderAll.bin** in the **MiniLoader**r folder downloaded earlier, and then click Open



i. Then click **Download**

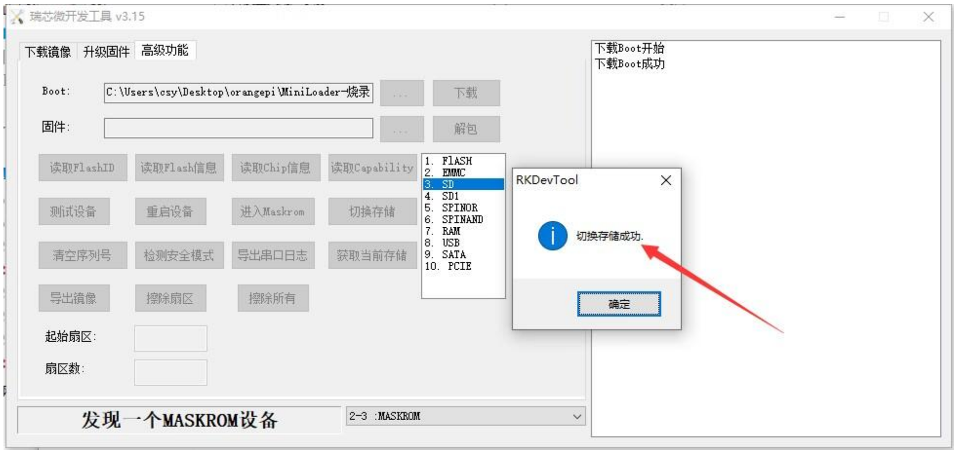


j. After downloading **MiniLoaderAll.bin**, the display is as shown below



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. 9. 2. Method of burning using SDDiskTool tool

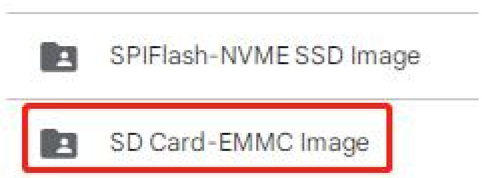
This method is not recommended as some TF cards are prone to getting stuck in the startup logo and unable to start

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

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

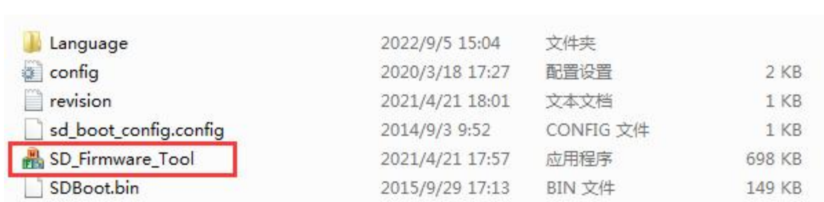
3) Then download the SDDiskTool burning tool from the **Orange Pi's download page**, **please ensure that the SDDiskTool tool version is the latest v1.72**

4) Then download the Android image from the **Orange Pi's data download page**. After opening the download link for the Android image, you can see the following two types of Android images. Please select the image in the **TF card and eMMC startup image** folder to download



5) Then use decompression software to decompress the downloaded Android image compressed file. In the decompressed file, the file ending with ".img" is the Android image file, with a size of 1GB or more

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



7) After opening **SDDiskTool**, if the TF card recognition is normal, the inserted disk device will be displayed in the **"Select Removable Disk Device"** column. **Please make sure that the displayed disk device matches the drive letter of the TF card you want to burn**. If it does not display, you can try unplugging or unplugging the TF card 开 **SDDiskTool**

8) After confirming the drive letter, you can first format the TF card by clicking the **restore disk** button in `SDDiskTool`, or you can 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 Start"** in the **"Select Function Mode"**

b. Then select the path of the Android image in the **"Choose Upgrade Firmware"** column

c. Finally, clicking the **'Start Creating'** button will start burning the Android image onto the TF card

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

2. 10. Method for burning Android images to eMMC

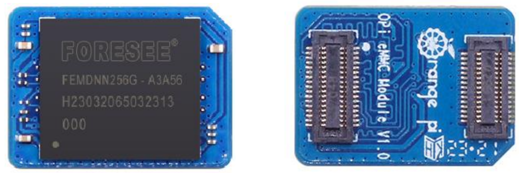
2. 10. 1. Method of burning using RKDevTool

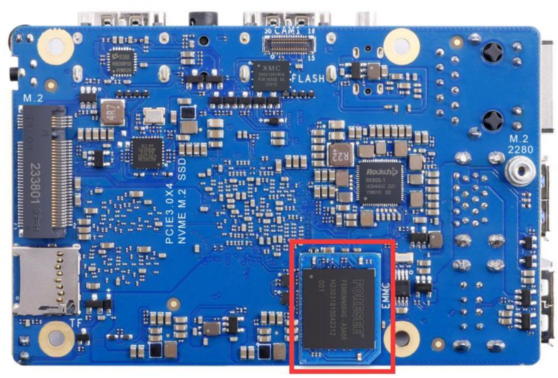
Note that all the following operations were performed on a Windows computer.

1) The development board has reserved an extension interface for eMMC. Before burning the system to eMMC, it is necessary to purchase an eMMC module that matches

the eMMC interface of the development board. Then install the eMMC module onto the development board.

The method of inserting the eMMC module into the development board is as follows:

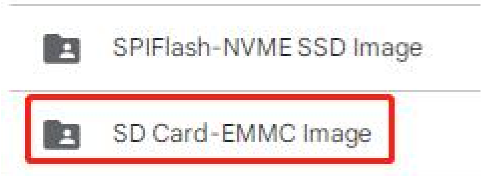




2) We also need to prepare a high-quality USB2.0 male to male data cable

3) Then download the Rockchip micro driver **DriverAssitant_v5.12.zip** and the burning tool **RKDevTool_Release_v3.15.zip** from the **Orange Pi's download page**

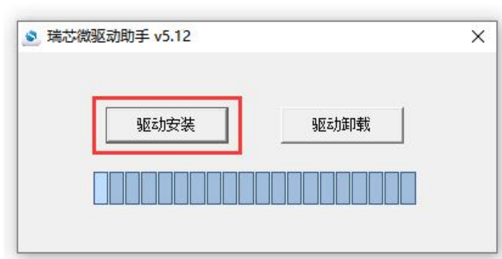
4) Then download the Android image from the **Orange Pi's data download page**. After opening the download link for the Android image, you can see the following two types of Android images. Please select the image in the **TF card and eMMC startup image** folder to download



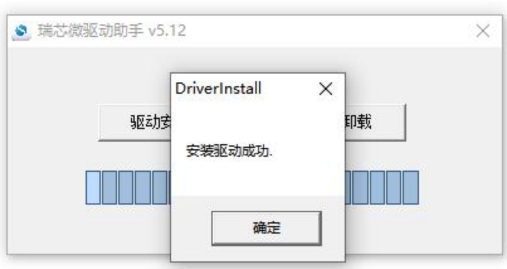
5) Then use decompression software to extract the **DriverAssitant_v5.12.zip** file, and then find the **DriverInstall.exe** executable file in the extracted folder and open it 可

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

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

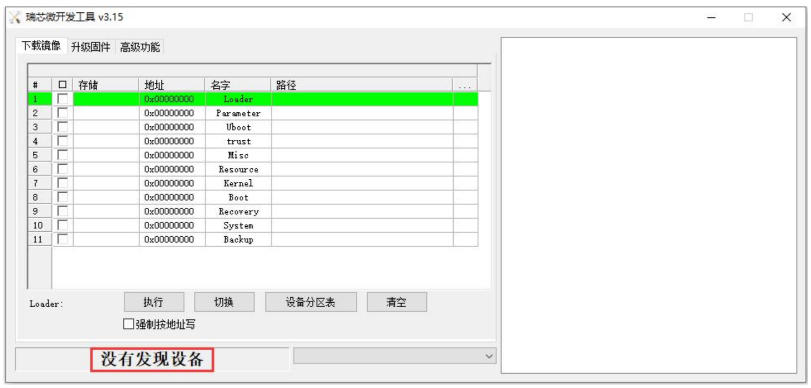


b. After waiting for a period of time, a window will pop up prompting **"Driver installation successful"**, then click the **"OK"** button.



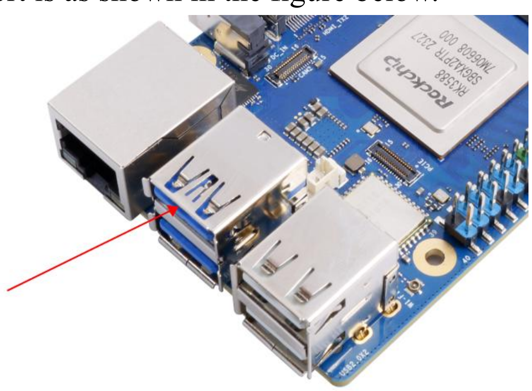
7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

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.



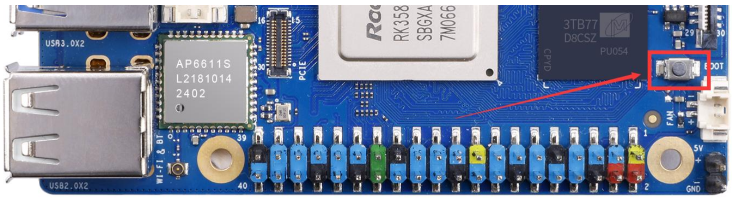
9) Then start burning the Android image into eMMC

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.

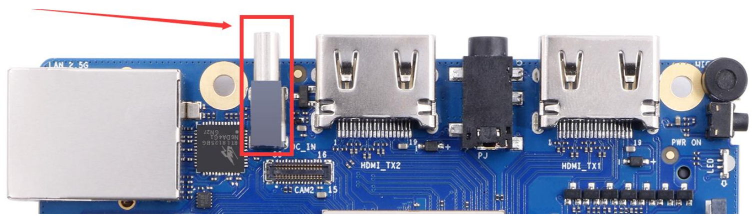


b. Make sure the development board is not connected to the power supply and the TF card is not inserted.

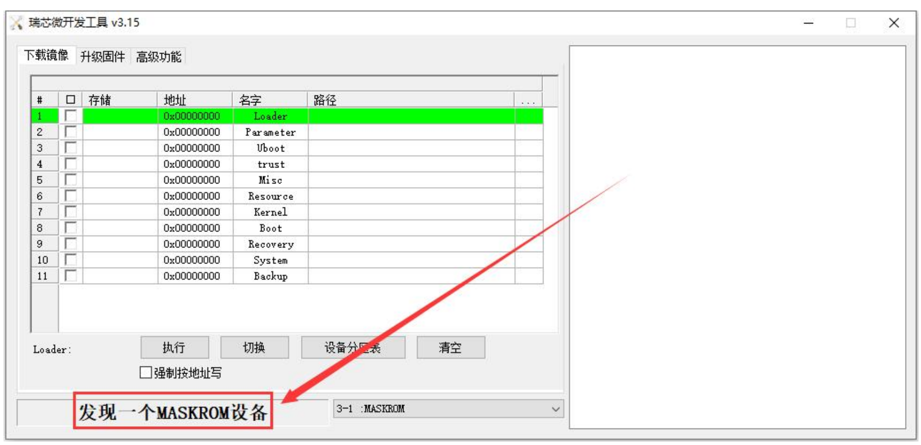
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



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

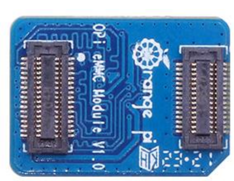
g. Then click the "**Firmware**" button to select the path of the Android image that needs to be burned.

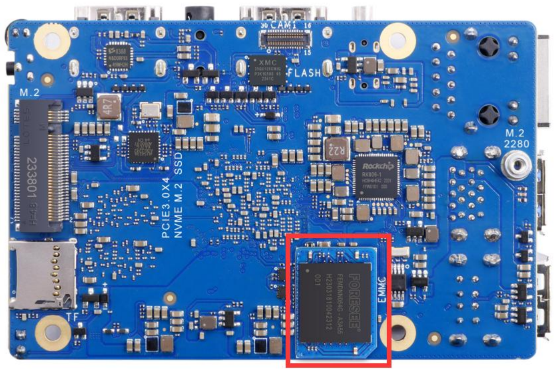
h. Finally, click the "**Upgrade**" button to start burning. The log during the burning process is as shown below. After the burning is completed, the Android system will automatically start.

2. 10. 2. How to burn using SDDiskTool

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

1) The development board has reserved an eMMC expansion interface. 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 on the development board. The eMMC module and the method of inserting it into the development board are as follows:



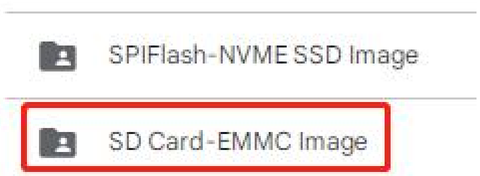


2) You also need to prepare a TF card with a capacity of 8GB or larger. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card from a brand such as SanDisk.

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

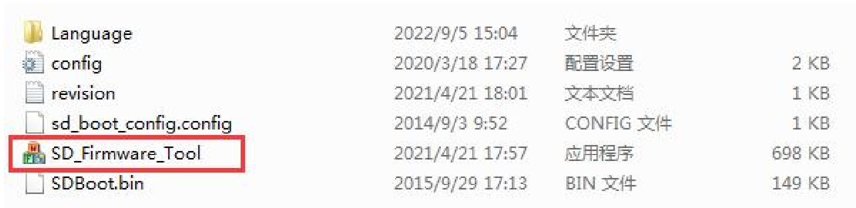
4) Then download the SDDiskTool flashing tool from **Orange Pi's download page**. **Please make sure that the version of SDDiskTool is the latest v1.72**

5) Then download the Android image from **Orange Pi's download page**. After opening the Android image download link, 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.



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

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



8) After opening **SDDiskTool**, if the TF card is recognized normally, the inserted disk device will be displayed in the "**Select Removable Disk Device**" column. **Please make sure that the displayed disk device is consistent with the drive letter of the TF card you want to burn.** If it is not displayed, you can try to unplug the TF card.

9) After confirming the drive letter, you can format the TF card first by clicking the **Restore Disk** button in **SDDiskTool**. You can also use the **SD Card Formatter** mentioned above to format the TF card.

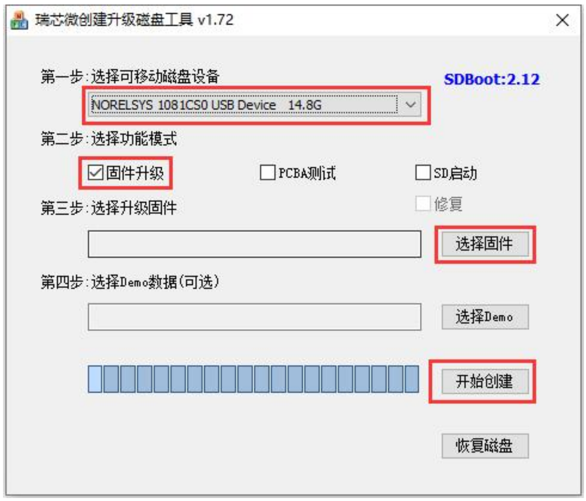
10) Then start writing the Android image to the TF card

a. First, confirm that the drive letter displayed under "**Select removable disk device**" is the drive letter corresponding to the TF card

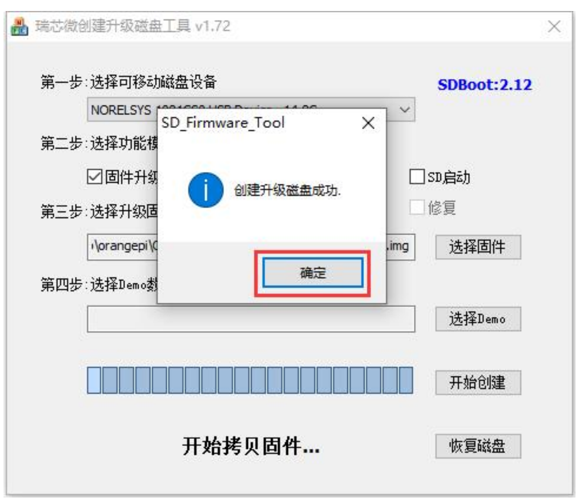
b. Then select "**Firmware upgrade**" in "Select function mode"

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

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

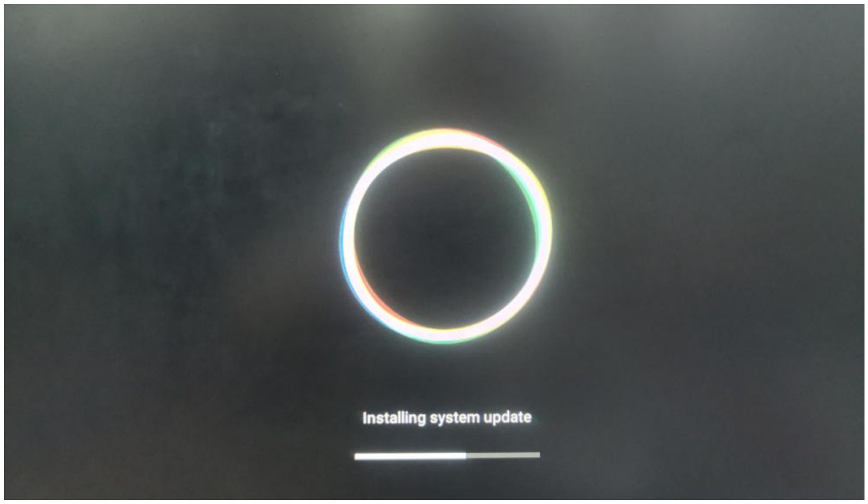


11) After the burning is completed, the display will be as shown below, and then you can exit SDDiskTool

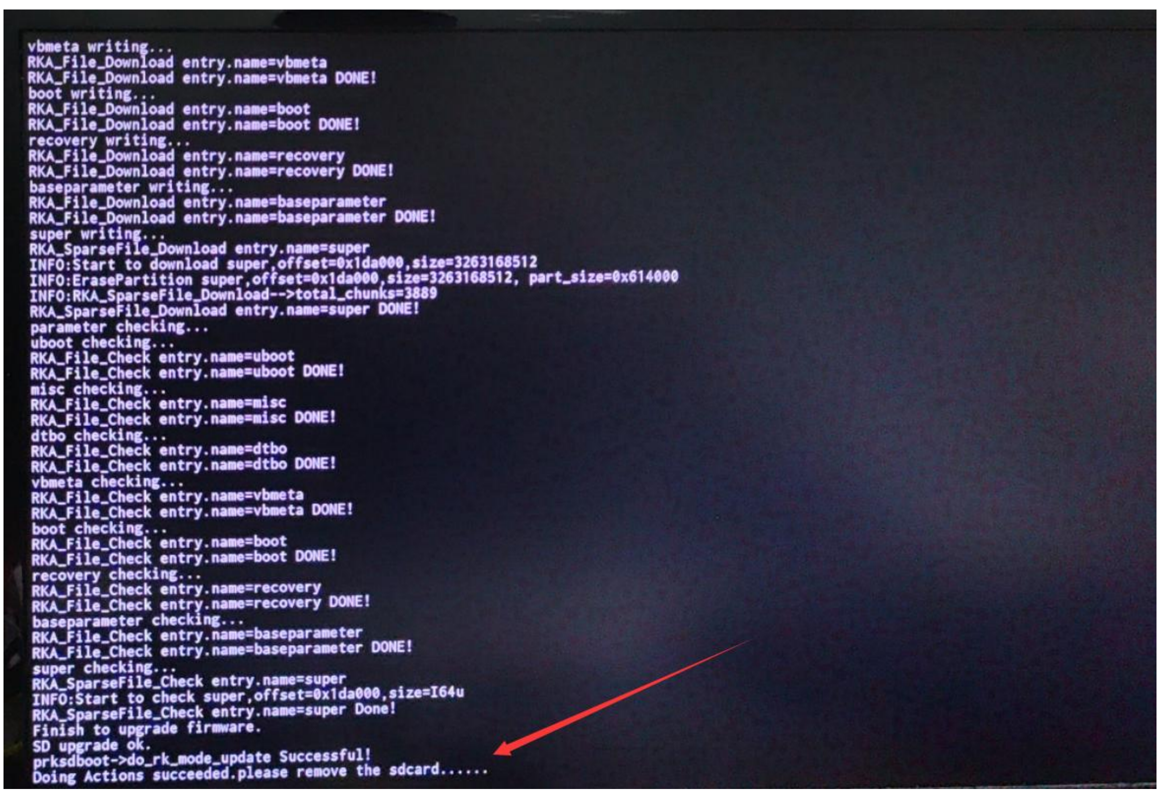


12) Then remove the TF card from the computer and insert it into the development board. After the development board is powered on, it will automatically start burning the Android image in the TF card to the eMMC of the development board.

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



14) When the HDMI monitor displays the following information, it means that the burning of the Android image to the eMMC is complete. At this time, you can pull out the TF card, and then the Android system in the eMMC will start.

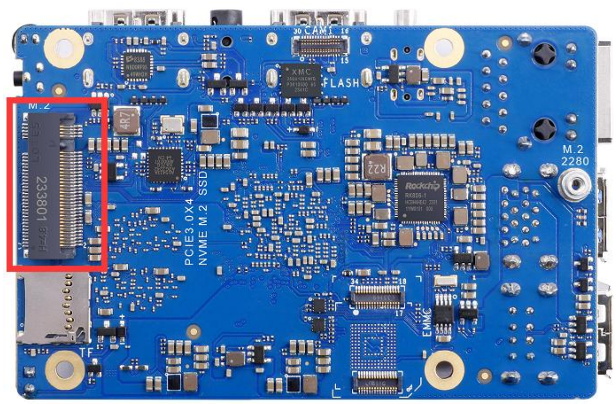


2.11. How to burn Android image to SPIFlash+NVMe SSD

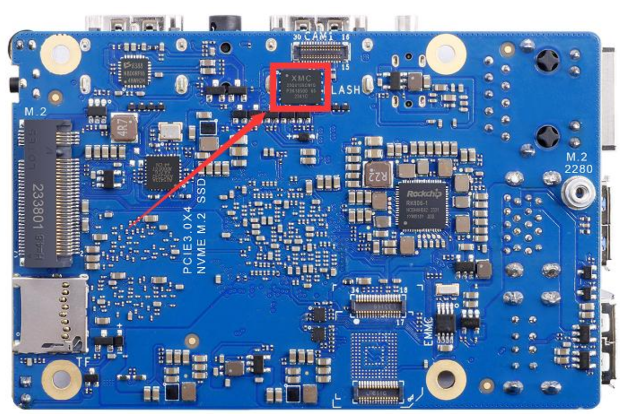
Note that all the following operations are performed on a Windows computer.

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

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 SPI Flash attached. The location of SPI Flash on the development board is shown in the figure below. No other settings are required before starting to burn.

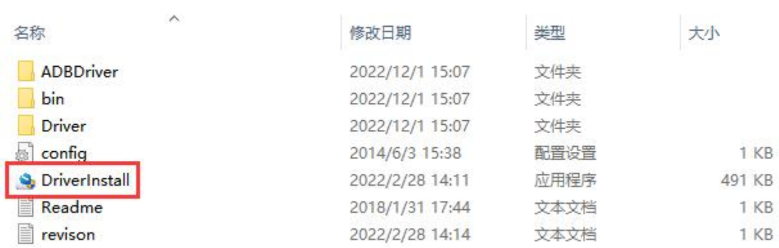


4) You also need to prepare a good quality USB2.0 male to male data cable

5) Then download Rockchip **DriverAssitant_v5.12.zip** and **RKDevTool_Release_v3.15.zip** from **Orange Pi's download page**

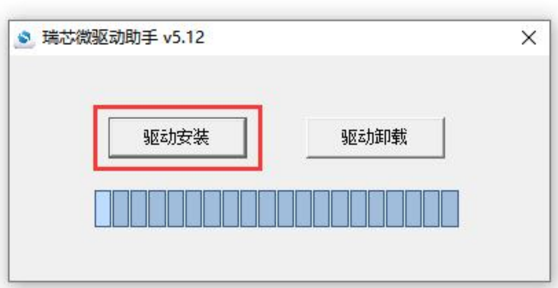
6) Then download the Android image. After opening the Android image download link, you can see the following two types of Android images. Please select the image in the **SPIFlash-NVME SSD** folder to download.

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

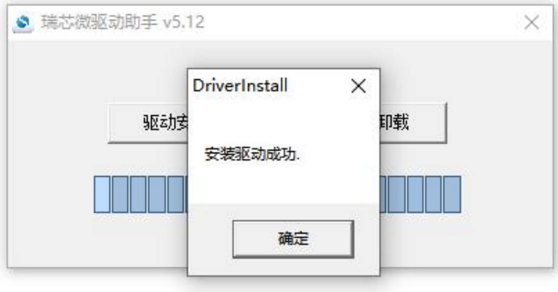


8) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

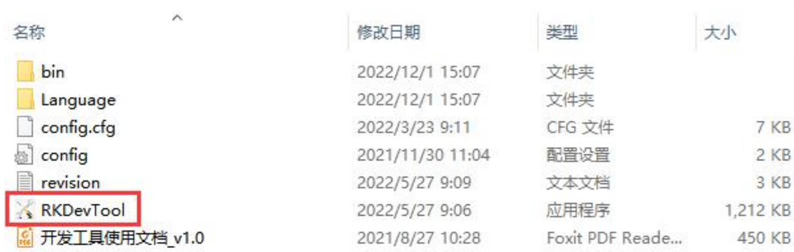
a. Click the "**Driver Installation**" button



b. After waiting for a while, a window will pop up saying "**Driver installation successful**", then click the "**OK**" button.



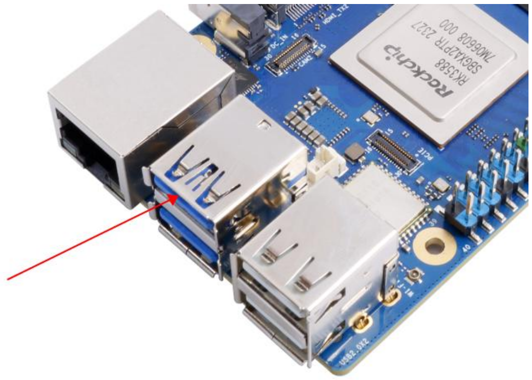
9) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Find **RKDevTool**in the unzipped folder and open it.



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

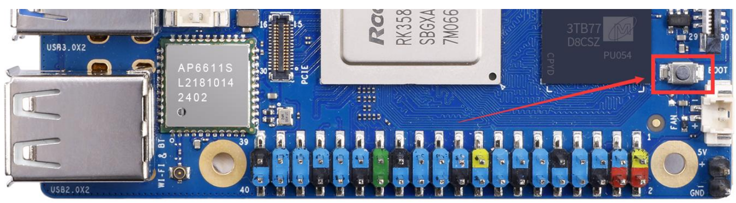
11) Then start burning the Android image to SPIFlash+NVMe SSD

a. First of all, through the USB2.0 male-to-male cable to connect the development board and the Windows computer, the development board USB burning port location is shown in the following figure

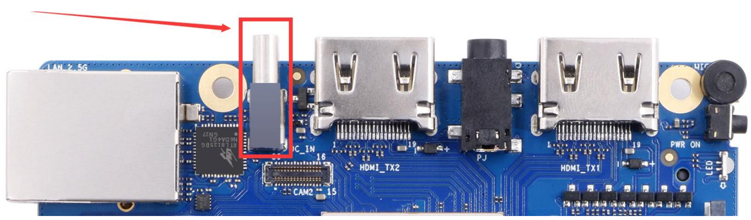


b. Make sure the development board is not plugged into a TF card and is not connected to a power source.

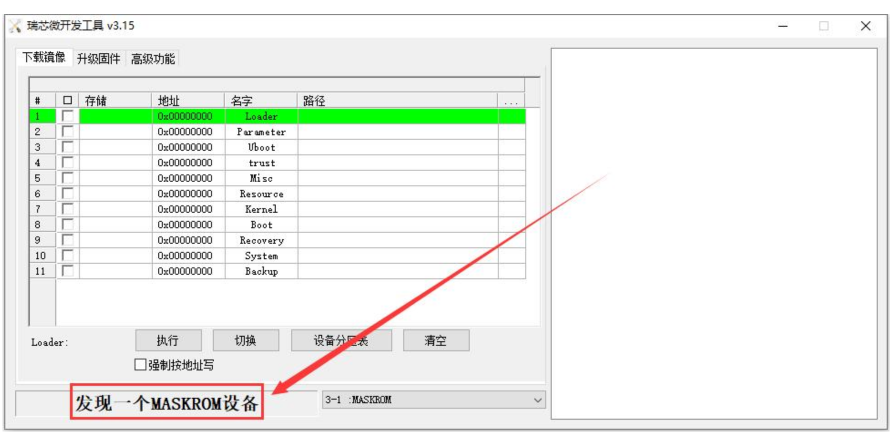
c. Make sure the development board is not plugged into a TF card and is not connected to a power source.



d. Make sure the development board is not plugged into a TF card and is not connected to a power source.



e. If the previous steps are successful, the development board will enter the **MASKROM** mode, and the burning tool interface will prompt "**A MASKROM device is found**"



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

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

h. Finally, click the "**Upgrade**" button will start burning, the burning process is shown in the figure below, you can see that the first will burn the firmware into the SPIFlash, and then burn the firmware into the PCIE. The Android system will start automatically after the burning is completed.

2. 12. Burning an Orange Pi OS (Droid) image to a TF card

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

1) First of all, prepare a TF card with a capacity of 8GB or more, the transfer speed of the TF card must be class10 or above, it is recommended to use SanDisk and other brands of TF cards.

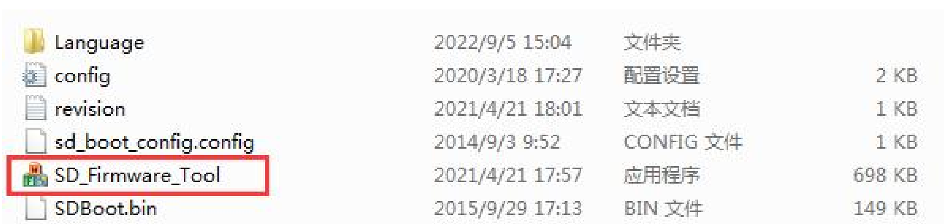
2) Then use a card reader to insert the TF card into the computer.

3) Then download the SDDiskTool from **the Orange Pi's download page,** **please make sure the version of the SDDiskTool is the latest v1.72.**

4) Then download the Orange Pi OS (Droid) image from **the Orange Pi's data download page.**

5) Then use decompression software to decompress the downloaded Orange Pi OS (Droid) image zip file, after decompression, the file ending with ".img" is the Orange Pi OS (Droid) image file, the size of 1GB or more.

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



7) Open **SDDiskTool**, if the TF card recognition is normal, will be in the "**Select Removable Disk Device**" column to display the inserted disk device, please be sure to confirm that the display of the disk device and you want to burn the TF card is the same as the disk drive letter, if not shown you can try to unplug the TF card!

8) After confirming the disk drive letter, you can format the TF card, click on the **Recover Disk** button in the SDDiskTool, or you can use the previously mentioned **SD Card Formatter** to format the TF card.

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

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

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

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

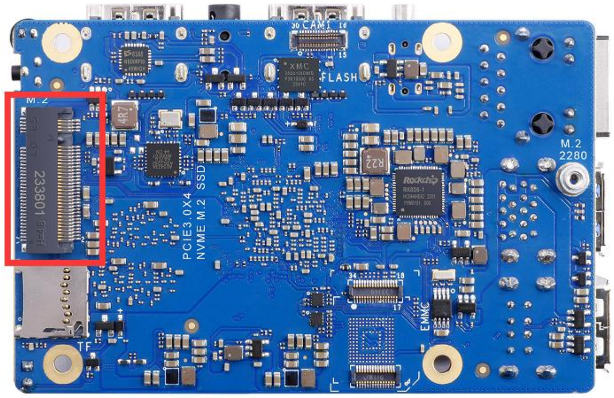
10) After burning, you can exit the SDDiskTool software, and then you can pull the TF card out of the computer and insert it into the development board to start.

2.13. Burn Orange Pi OS (Droid) images to SPIFlash+NVMe SSDs

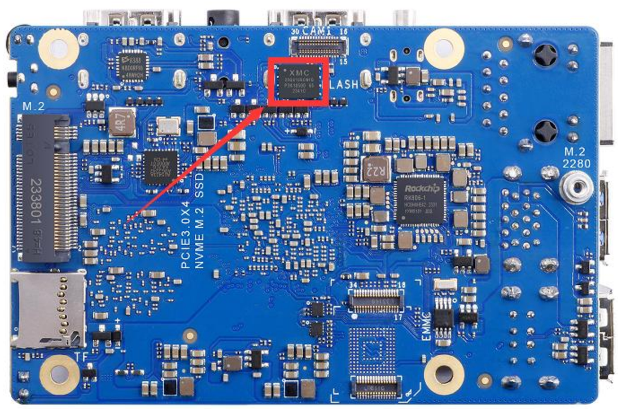
Note that all of the following operations are performed on a Windows computer.

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

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



3) Please ensure that the development board has been affixed to the SPI Flash, SPI Flash in the development of the board's position as shown in the figure below, before starting to burn without other settings

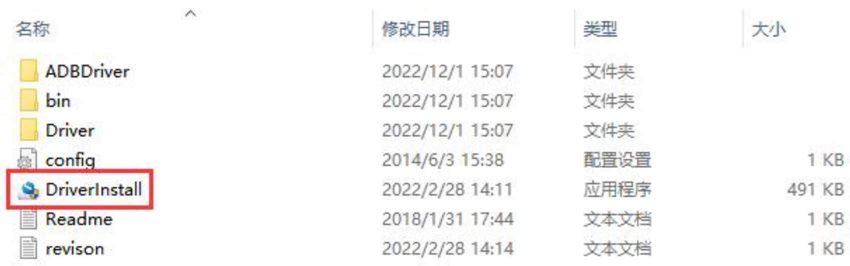


4) A good quality USB 2.0 male-to-male cable is also required.

5) Then download the Rexchip driver **DriverAssitant_v5.12.zip** and the burn tool **RKDevTool_Release_v3.15.zip** from the **Orange Pi's data download page**.

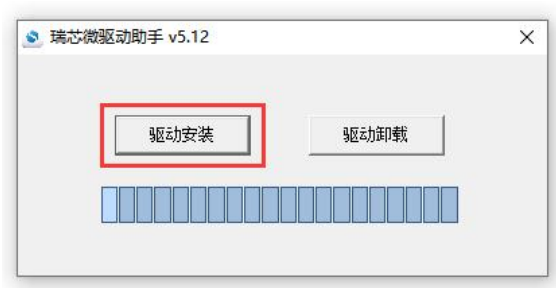
6) Then download the Orange Pi OS (Droid) image.

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

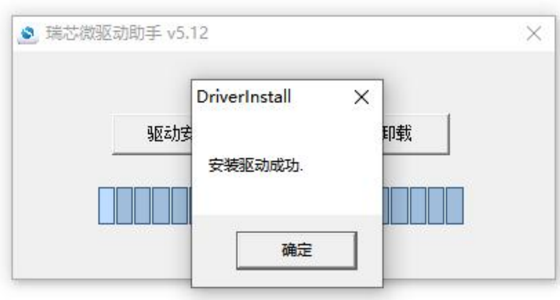


8) Open **DriverInstall.exe** and then install the Rexchip driver as follows

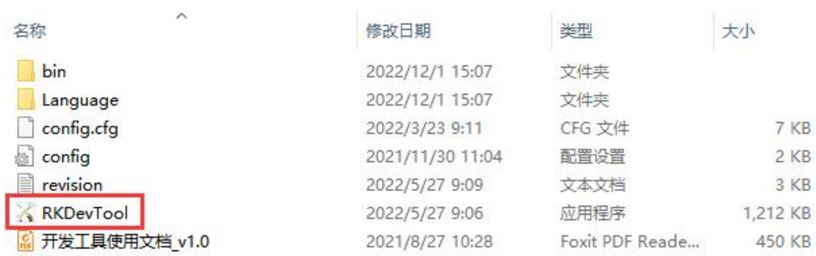
a. Click the "Driver Installation" button.



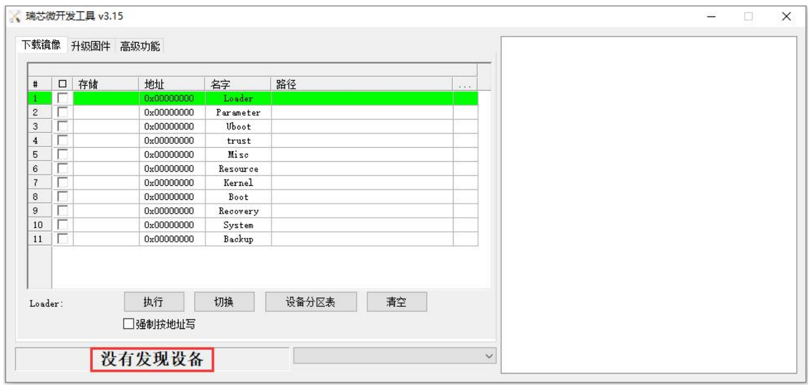
b. Wait for a period of time, a pop-up window will prompt "**Installation of the driver is successful**", and then click on the "**OK**" button can be



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

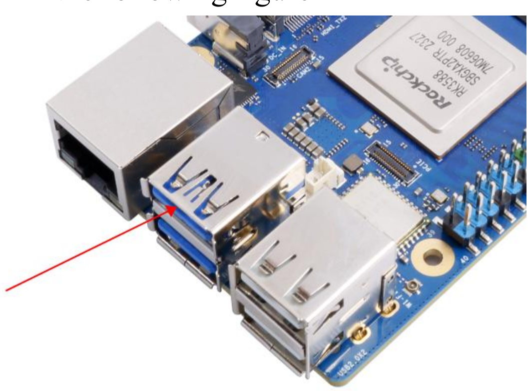


10) Open the**RKDevTool** burn tool, because the computer at this time has not been connected to the development board through the USB2.0 male-to-male cable, so the lower left corner will prompt "**no device found**".



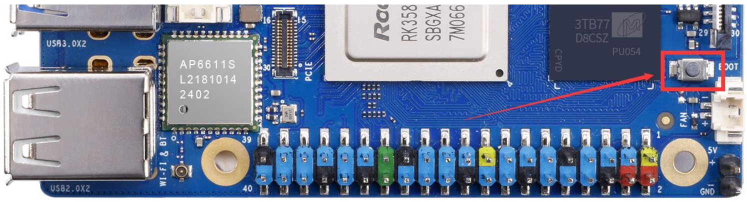
11) Then start burning Orange Pi OS (Droid) image to SPIFlash+NVMe SSD

a. First of all, through the USB2.0 male-to-male cable to connect the development board and the Windows computer, the development board USB2.0 burning port location is shown in the following figure

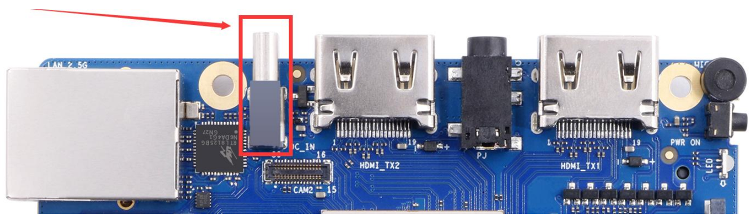


b. Ensure that the board does not have a TF card inserted and is not connected to a power source.

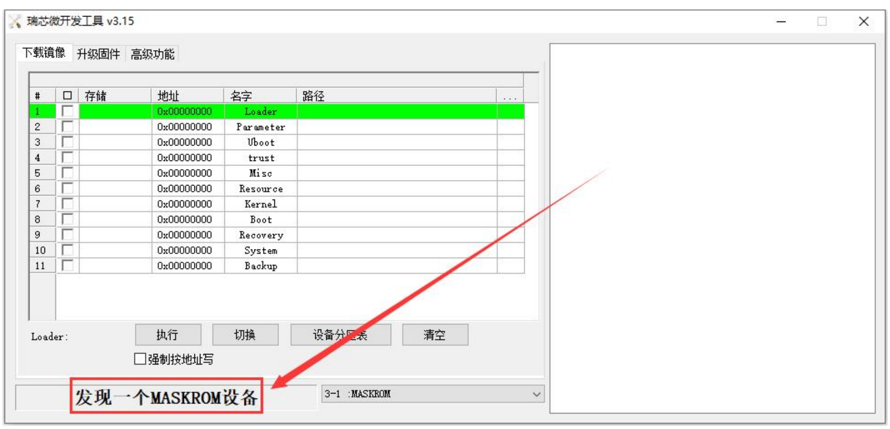
c. Then press and hold the MaskROM button on the development board. The location of the MaskROM button on the development board is shown in the following figure:



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



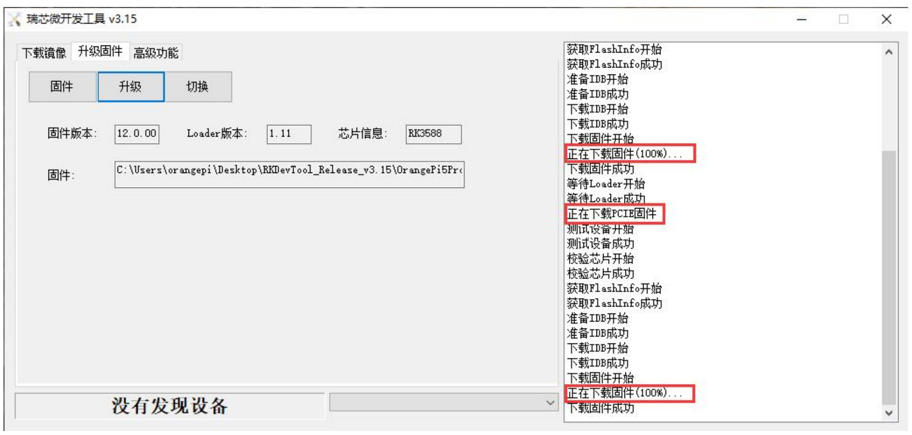
e. If the previous steps go well, the board will enter the **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device has been found**".



f. Then click on the "Upgrade Firmware" section of the burning tool.

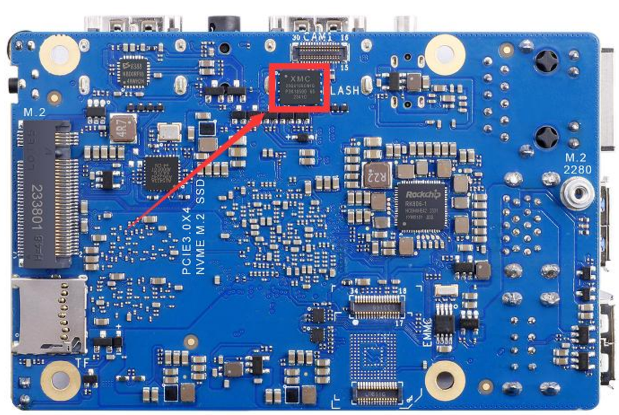
g. Then click the "Firmware" button to select the Orange Pi OS (Droid) image to be burned.

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 first be burned into SPIFlash, and then burned into the PCIE. The Orange Pi OS (Droid) system will start automatically after the burning process is completed.



2. 14. Using the RKDevTool to clear SPIFlash

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

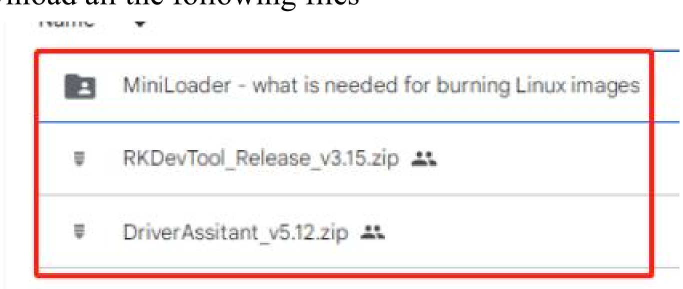


2) The location of SPI Flash on the development board is shown in the following figure

3) Then download the Rexchip **DriverAssitant_v5.12.zip** and **MiniLoader**as well as the burn-in tool **RKDevTool_Release_v3.15.zip** from the **Orange Pi's download page**.

a. On the Orange Pi data download page first select Official Tools and then go to the following folder

b. Then download all the following files



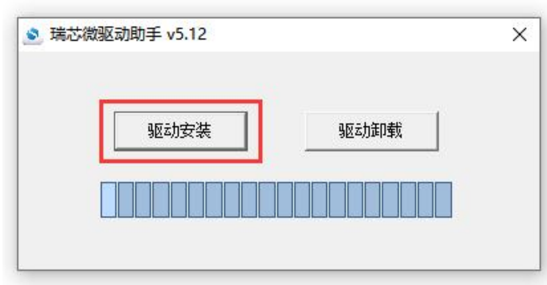
Note that the folder "MiniLoader - things you need to burn Linux images" is hereafter referred to as the MiniLoader folder.

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

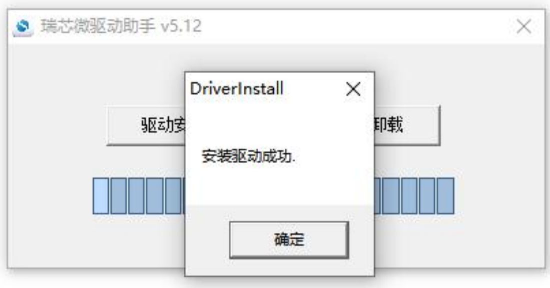
名称	修改日期	类型	大小
ADBDriver	2022/12/1 15:07	文件夹	
bin	2022/12/1 15:07	文件夹	
Driver	2022/12/1 15:07	文件夹	
config	2014/6/3 15:38	配置设置	1 KB
DriverInstall	2022/2/28 14:11	应用程序	491 KB
Readme	2018/1/31 17:44	文本文档	1 KB
revison	2022/2/28 14:14	文本文档	1 KB

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

a. Click the “Driver Installation” button.

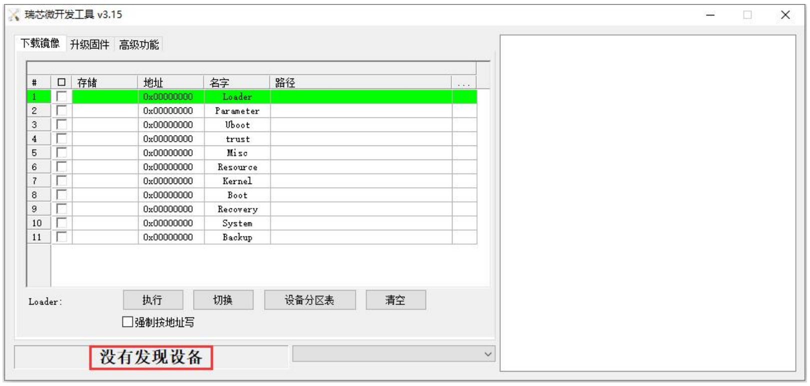


b. Wait for a period of time, a pop-up window will prompt “**Installation of the driver is successful**”, and then click on the “**OK**” button can be



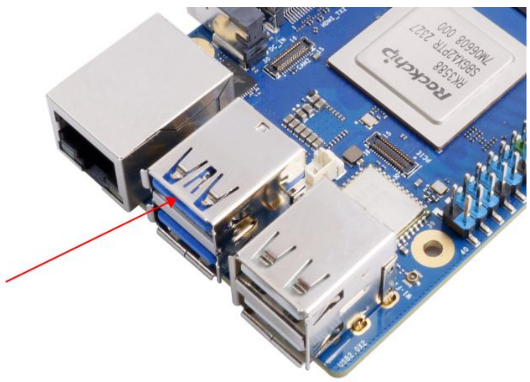
6) Then unzip **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, in the unzipped folder to find the **RKDevTool** to open it

7) Open the**RKDevTool**burning tool, because the computer at this time has not been connected to the development board through the USB2.0 male-to-male cable, so the lower left corner will prompt “**no device found**”.



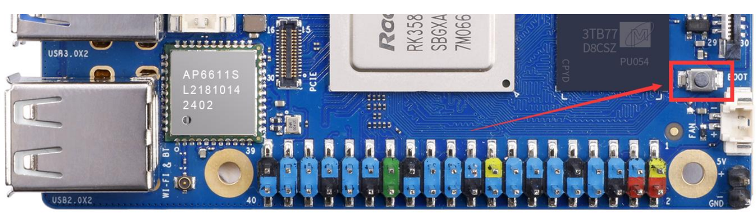
8) Then you can start to clear the contents of the SPI FLASH.

a. First of all, connect the development board and Windows computer through the Type-C cable, the location of the Type-C interface of the development board is shown in the following figure

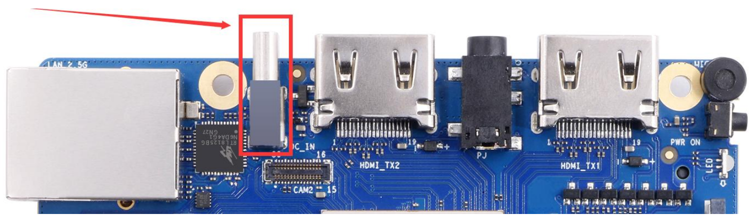


b. Ensure that the board does not have a TF card inserted and is not connected to a power source.

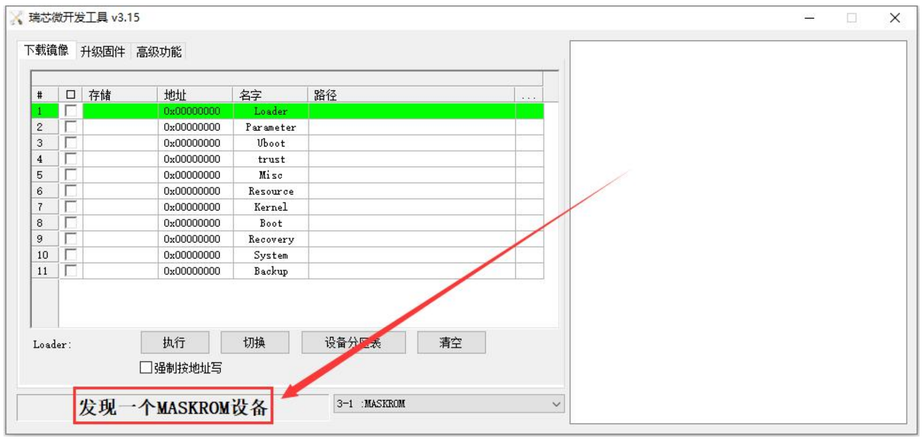
c. Then press and hold the MaskROM button on the development board. The location of the MaskROM button on the development board is shown in the following figure:



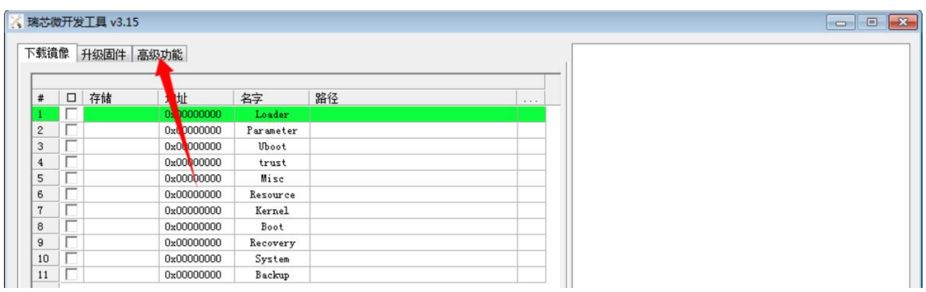
d. Then connect the Type-C port to the development board and power it up, then you can release the MaskROM button



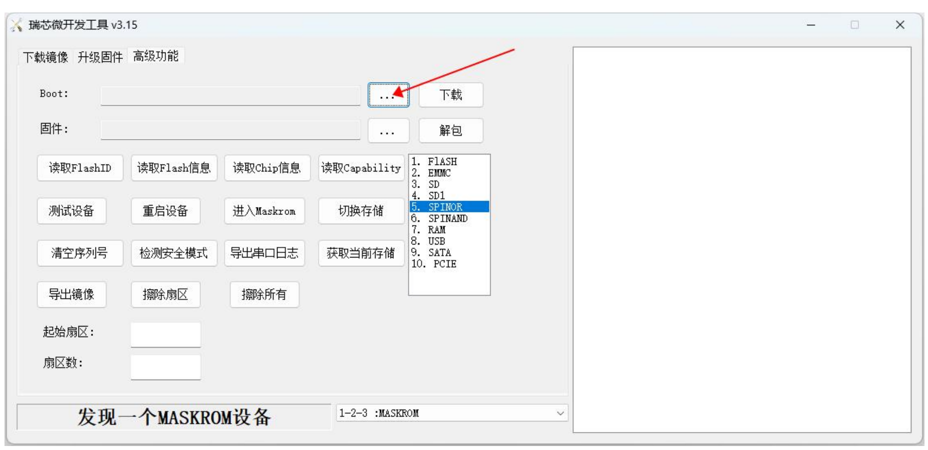
e. If the previous steps go well, the board will enter the **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device has been found**".



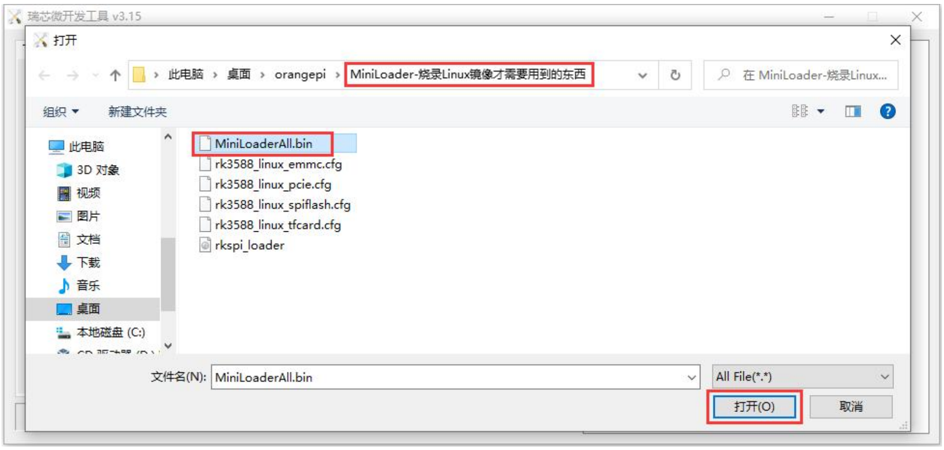
f. Then please select Advanced Features



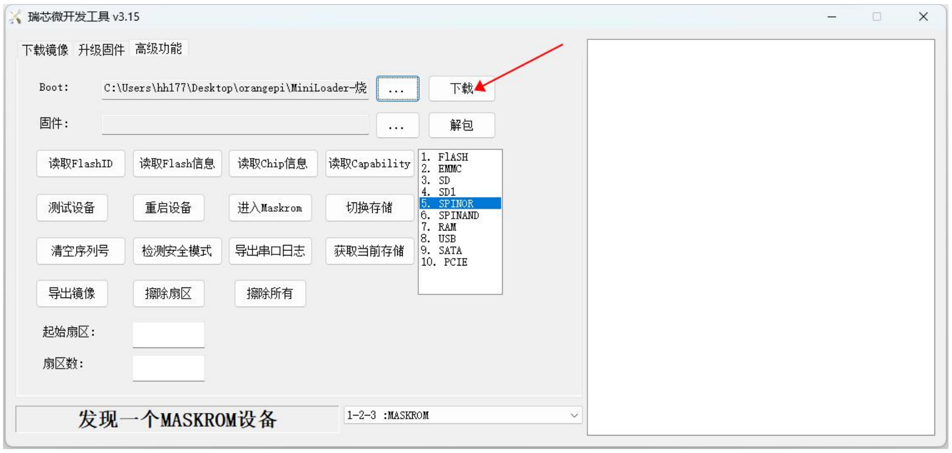
g. Then click on the location shown below



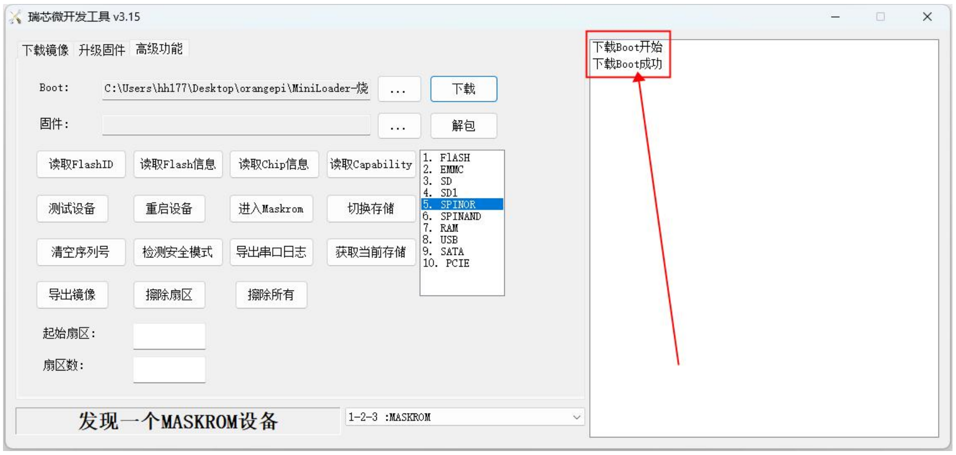
h. Select **MiniLoaderAll.bin** in the **MiniLoader** folder you downloaded earlier, and click Open.



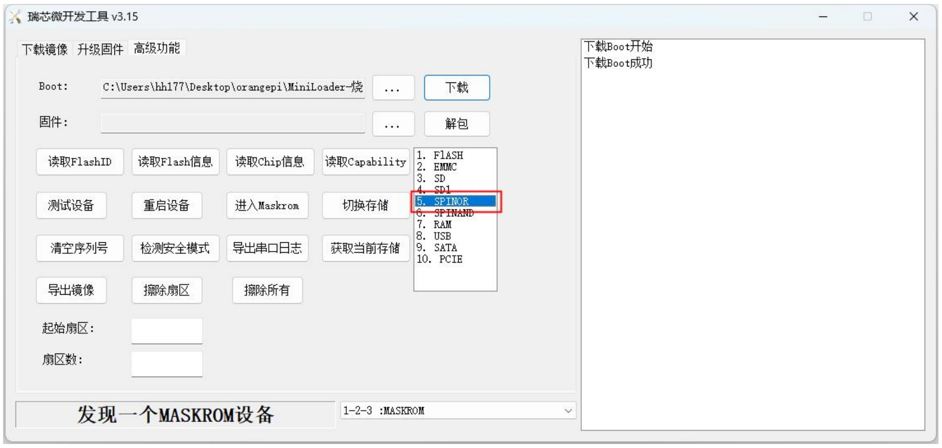
i. Then click **Download**



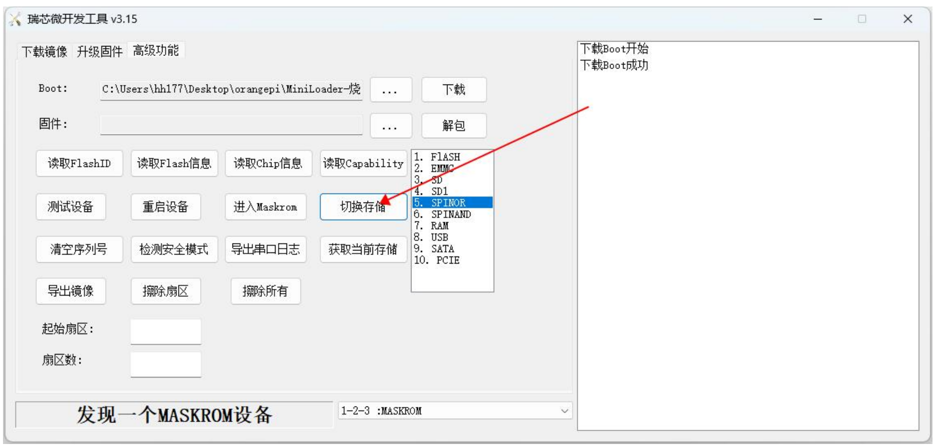
j. After downloading **MiniLoaderAll.bin**, the display is as shown below



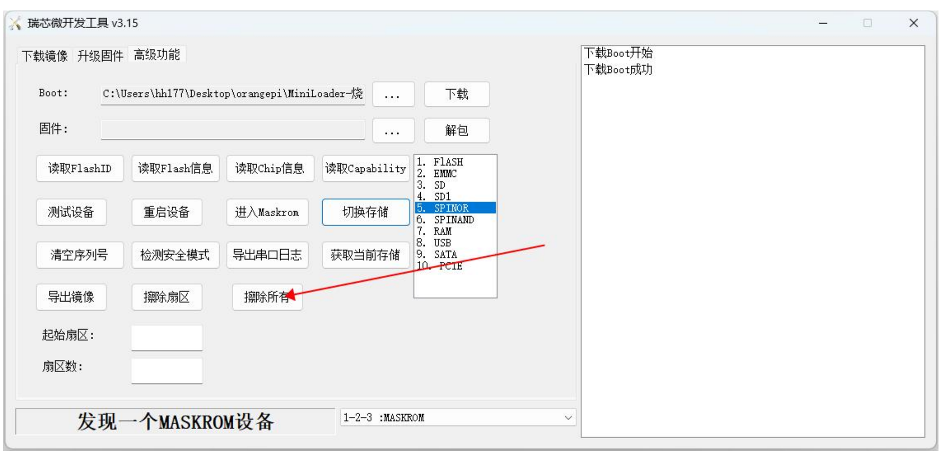
k. Then select the storage device as **SPINOR**



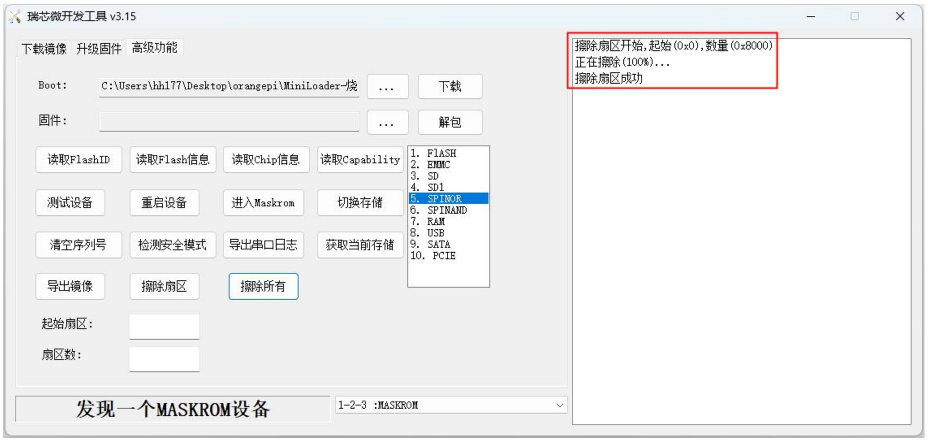
l. Then click **Switch Storage**



m. Then click on Erase All to start erasing SPIFlash



n. The display log after erasing SPIFlash is shown below



2. 15. Starting the Orange Pie Board

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

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

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

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

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

Remember not to insert a power adapter with a voltage output greater than 5V, which will burn the development board.

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

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

Please do not connect to the USB port of your computer to power the board.

6) Then turn on the power adapter switch, if everything is normal, at this time the HDMI monitor or LCD screen will be able to see the system boot 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 board to the computer, please refer to the section of debugging the use of the serial port for the serial port connection method.

2. 16. Debugging the use of serial ports

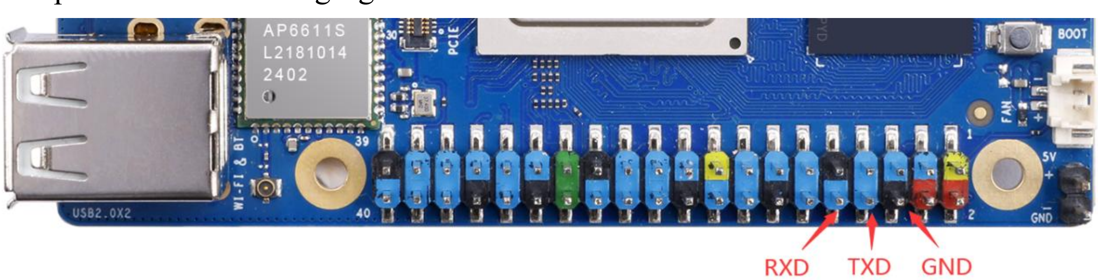
2. 16. 1. Connection instructions for debugging the serial port

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

For better compatibility, CH340 USB to TTL module is recommended, please don't use CP2102, PL2303 type USB to TTL module.

Before purchasing USB to TTL module, please make sure the module supports baud rate of 1500000 rate.

2) The debugging serial port GND, RXD and TXD pins of the development board correspond to the following figure



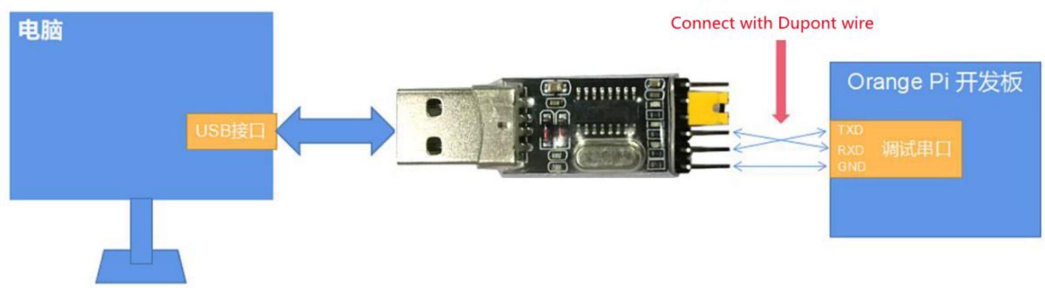
3) USB to TTL module GND, TXD and RXD pins need to be connected to the debug serial port of the development board through the DuPont cable.

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

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

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

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



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

The TX and RX of the serial port need to be cross-connected, if you don't want to carefully distinguish the order of TX and RX, you can connect the TX and RX of the serial port randomly first, if the test does not have an output and then exchange the order of TX and RX, so that there is always a kind of order is right!

2. 16. 2. Ubuntu platform debugging the use of serial ports

Linux can be used under the serial port debugging software there are many, such as putty, minicom, etc., the following demonstrates the use of putty.

1) First of all, the USB to TTL module will be inserted into the USB port of the Ubuntu computer, if the USB to TTL module connection is recognized as normal, you can see the corresponding device node name under the **/dev** of the Ubuntu PC, remember the node name, and then set up the serial port software will be used!

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

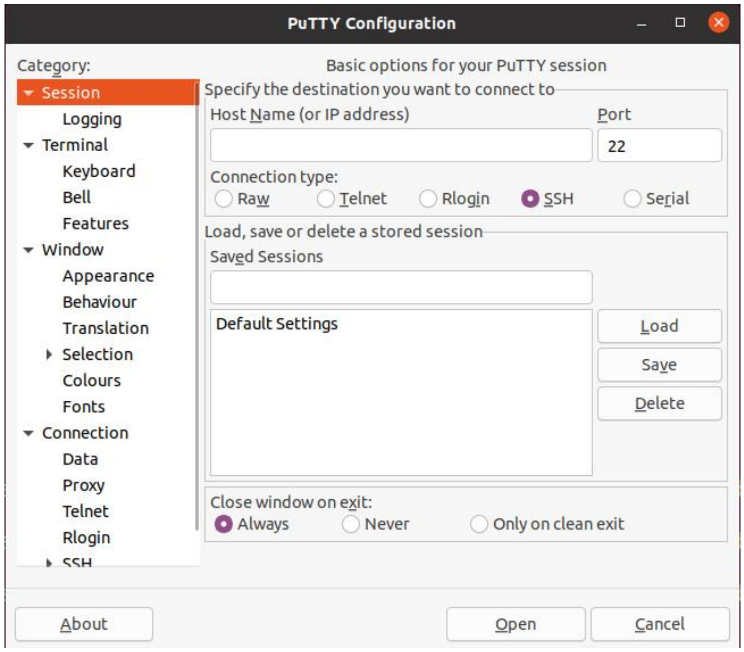
2) Then install putty on your Ubuntu PC using the following command

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

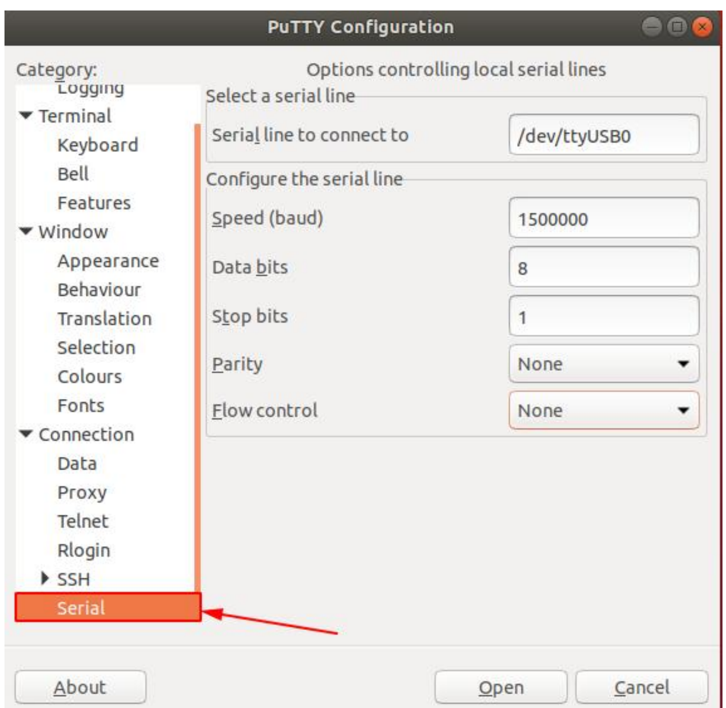
3) Then run putty, **remembering to add sudo privileges.**

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

4) The following screen will pop up after executing the putty command



5) First, select the setting interface of the serial port

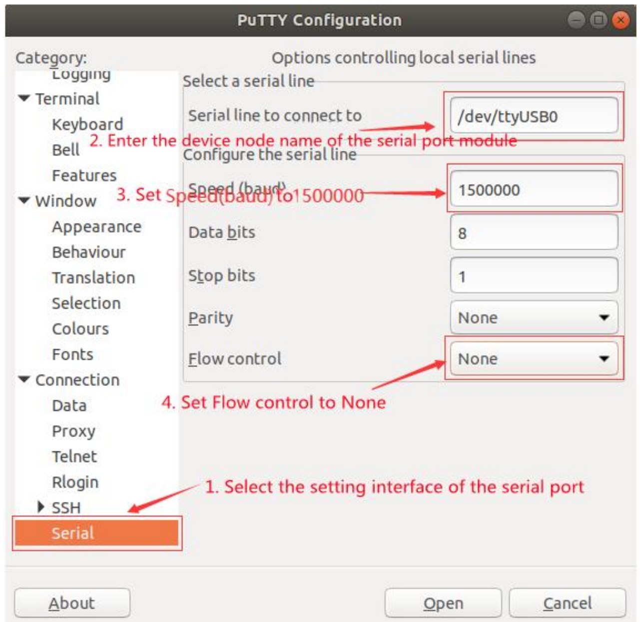


6) Then set the parameters of the serial port

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

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

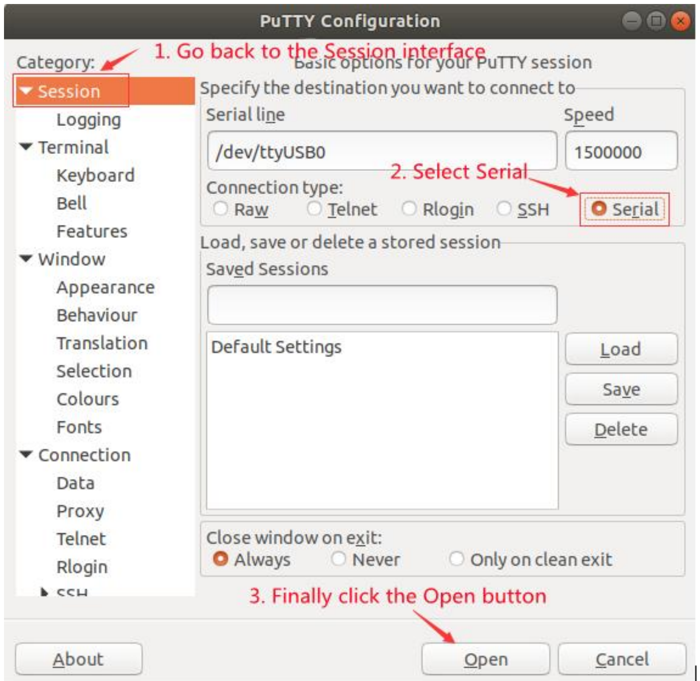
c. Set Flow control to None



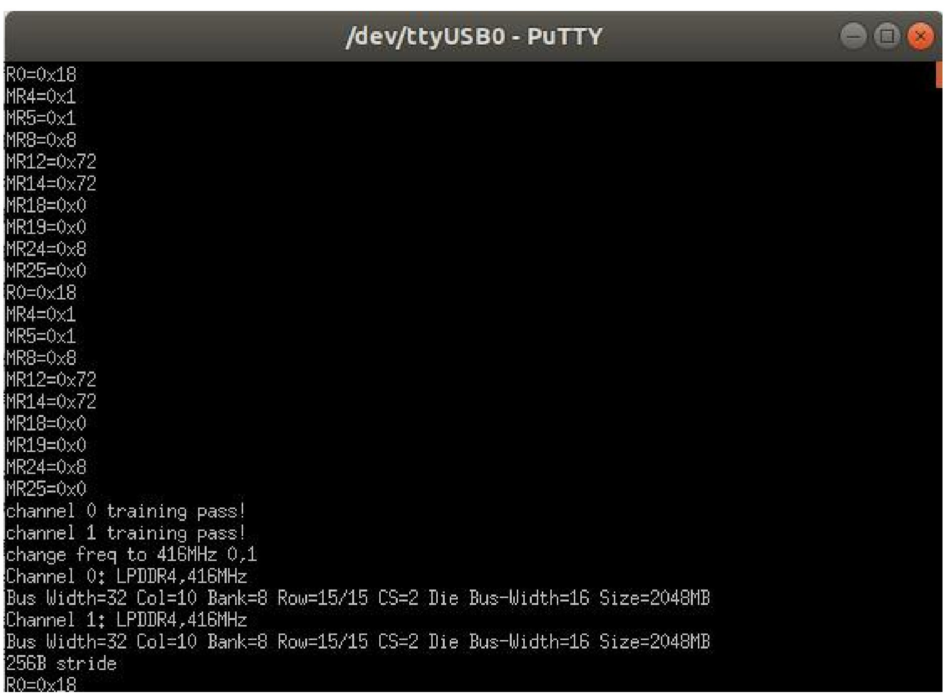
7) After setting up the serial interface, go back to the Session interface.

a. Serial First select theConnection type as Serial.

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



8) After starting the board, you will be able to see the Log messages output by the system from the open serial terminal.



2. 16. 3. Windows platform debugging serial port usage

Windows can use a lot of serial debugging software, such as SecureCRT,

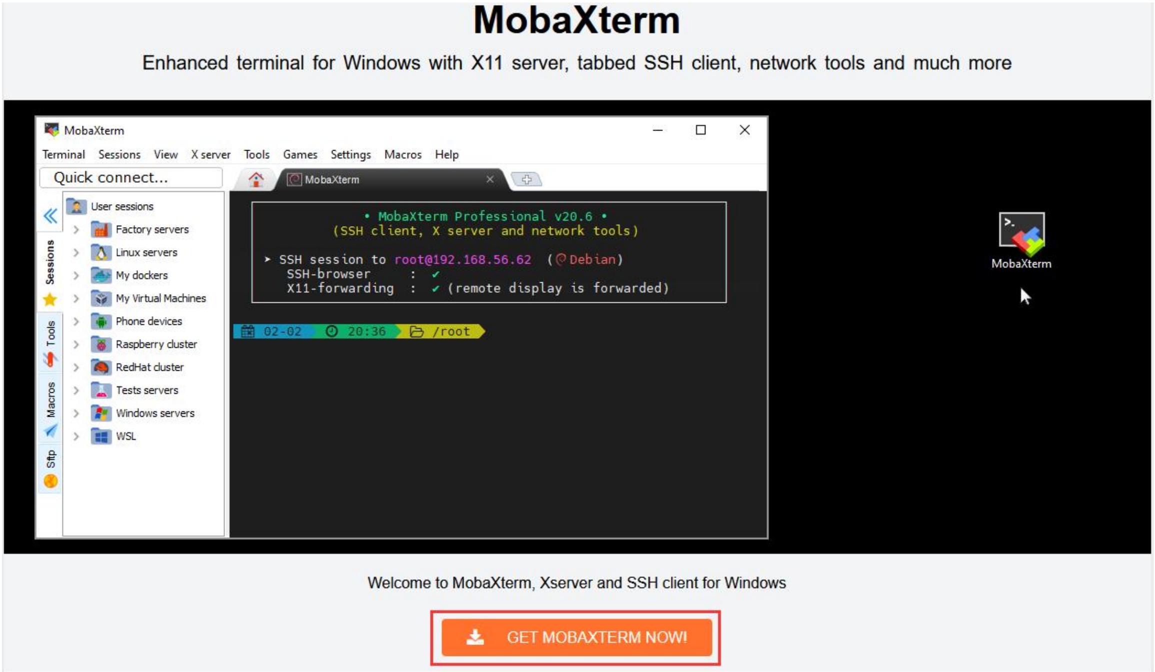
MobaXterm, etc., the following demonstrates the use of MobaXterm, this software has a free version, no need to purchase a serial number can be used.

1) Download MobaXterm

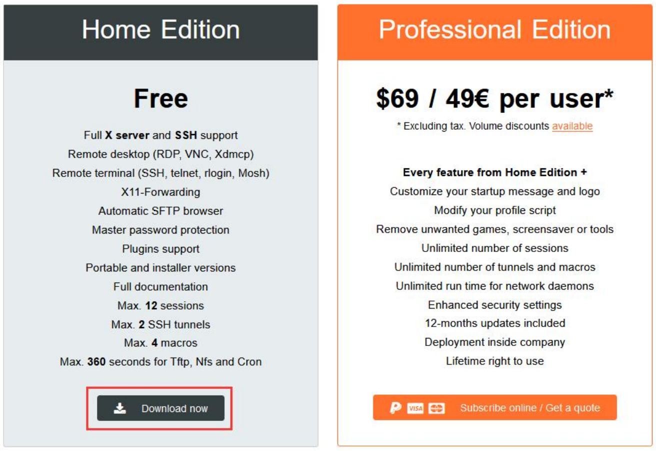
a. Download MobaXterm at the following address

https://mobaxterm.mobatek.net

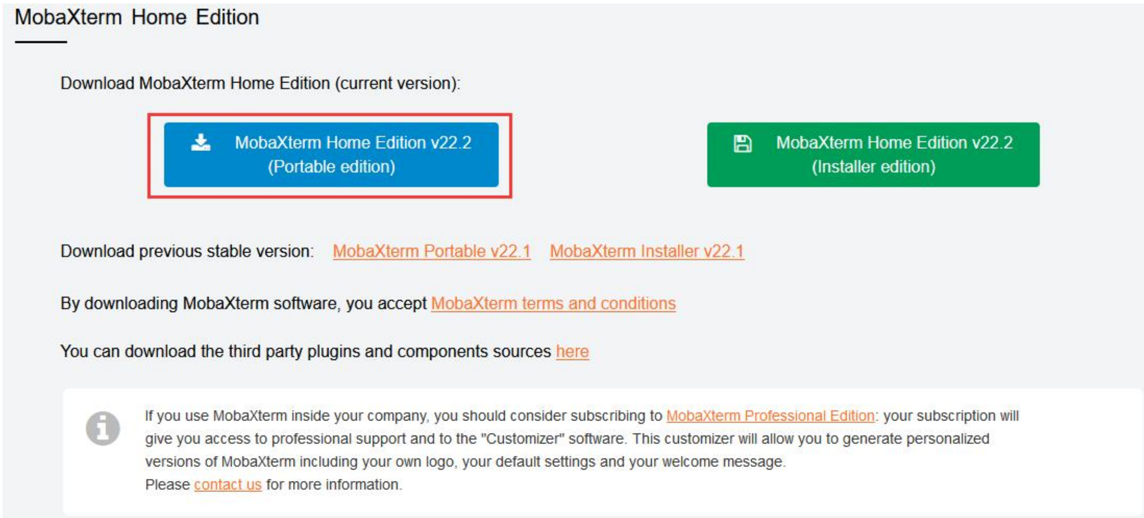
b. Go to the MobaXterm download page and click on the **GET XOBATERM NOW!**



c. Then choose to download the Home version



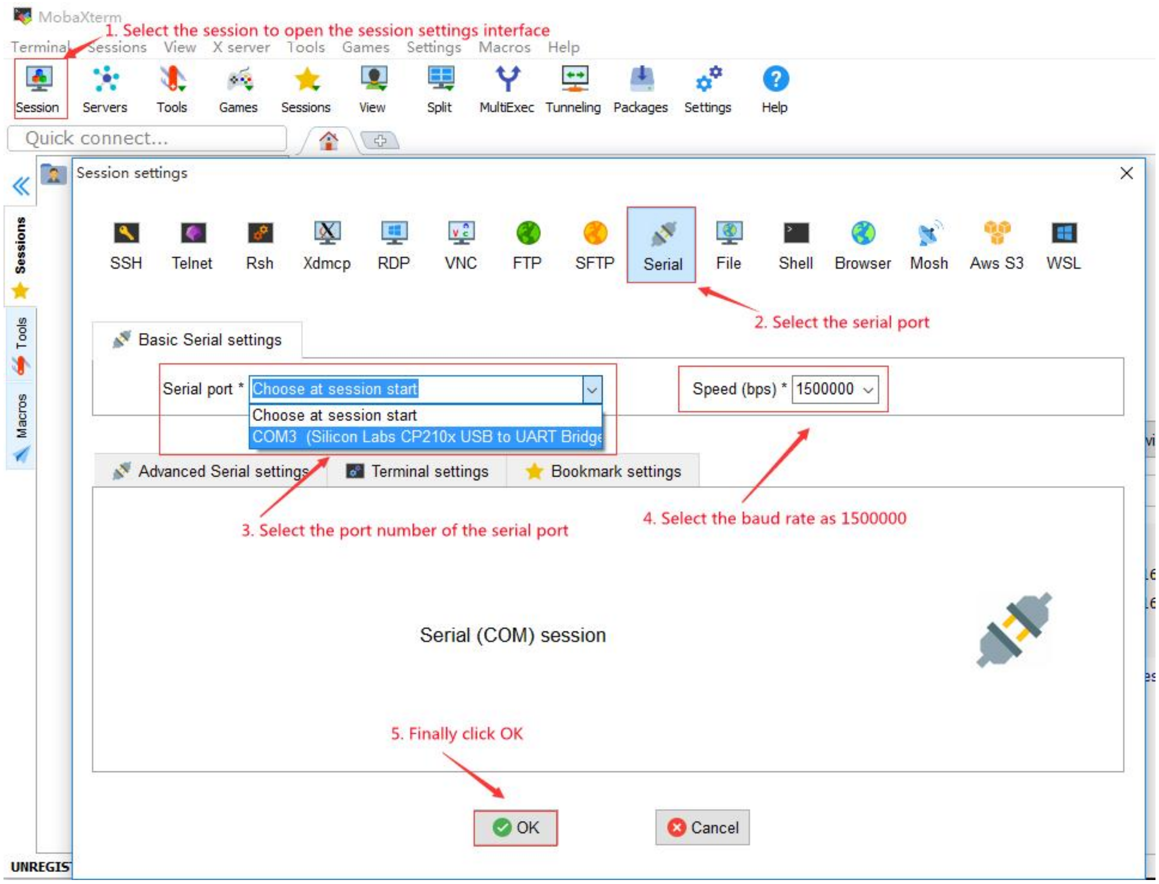
d. Then select the Portable version, after downloading, no need to install, directly open the can be used!



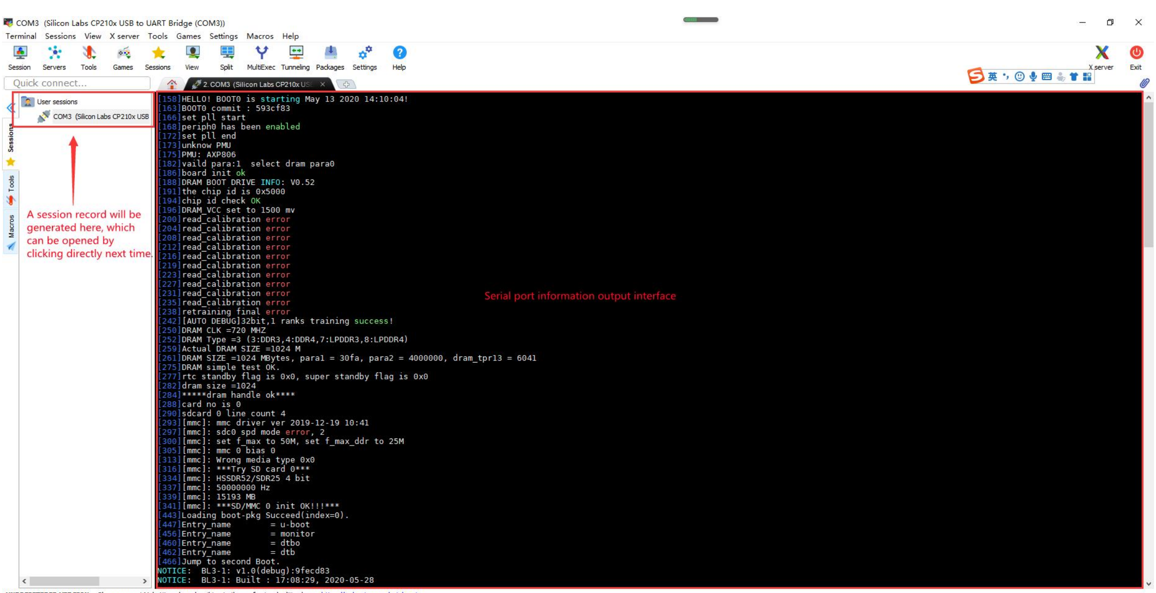
2) After downloading, use the decompression software to decompress the downloaded package, you can get the executable software of MobaXterm, and then double-click to open it.

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

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



4) Click the "**OK**" button to enter the following interface, at this time to start the development board will be able to see the output information of the serial interface



2. 17. Instructions for supplying power using the 5v pin in the 40pin connector of the development board

Our recommended way to power the board is to use a 5V/4A Type C power cord plugged into the board's Type-C power connector. If you need to use the 5V pin in the 40pin connector to power the board, please make sure that the power cord and power adapter can meet the power needs of the board. If the power supply is unstable, please switch back to the Type-C power supply.

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

The power cord shown above is available on website, so please search for it and buy it yourself.

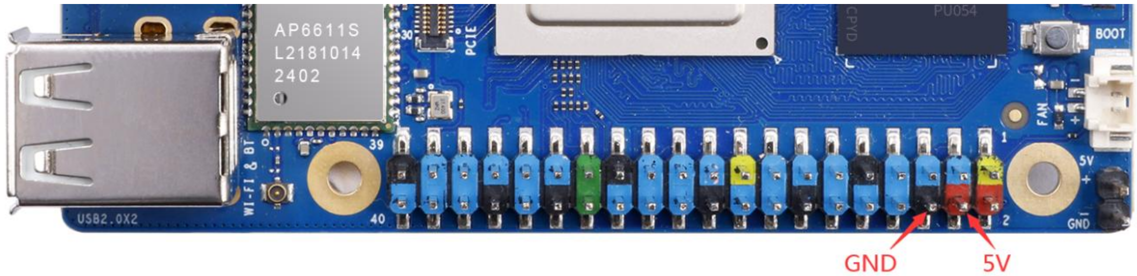
2) Use the 5V pin in the 40pin connector to supply power to the development board, and connect the power cable as follows

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

b. The red DuPont cable needs to be plugged into the 40pin 5V pin on the development board.

c. The black DuPont cable needs to be plugged into the GND pin of the 40pin connector.

d. 40pin interface 5V pin and GND pin in the development of the board in the position shown in the following figure, r**emember not to connect the reverse**



3. Instructions for using Ubuntu/Debian Server and Xfce desktop systems

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 see the chapter on Orange Pi OS Arch system instructions.

3. 1. Supported Linux image types and kernel versions

Linux image types	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
Debian 12 - Bookworm	Linux6.1	Support	Support
Ubuntu 22.04 - Jammy	Linux6.1	Support	Support

3. 2. Linux 5.10 System Adaptations

Function	Debian11	Debian12	Ubuntu20.04	Ubuntu22.04
HDMI TX1 Video	OK	OK	OK	OK
HDMI TX1 Audio	OK	OK	OK	OK

HDMI TX2 Video	OK	OK	OK	OK
HDMI TX2 Audio	OK	OK	OK	OK
USB2.0x2	OK	OK	OK	OK
USB3.0x2	OK	OK	OK	OK
2.5G network port	OK	OK	OK	OK
Ethernet port status light	OK	OK	OK	OK
WIFI	OK	OK	OK	OK
Bluetooth	OK	OK	OK	OK
Debugging Serial Ports	OK	OK	OK	OK
RTC Chip	OK	OK	OK	OK
FAN Interface	OK	OK	OK	OK
eMMC Extended Interface	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
PWM（40pin）	OK	OK	OK	OK
OV13850 Camera	OK	OK	OK	OK
OV13855 Camera	OK	OK	OK	OK
SPI+NVME 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
Power On/Off Button	OK	OK	OK	OK
Watchdog test	OK	OK	OK	OK
Chromium	OK	OK	OK	OK

hardening video				

3. 3. Linux 6.1 System Adaptations

Function	Debian12	Ubuntu22.04
HDMI TX1 Video	OK	OK
HDMI TX1 Audio	OK	OK
HDMI TX2 Video	OK	OK
HDMI TX2 Audio	OK	OK
USB2.0x2	OK	OK
USB3.0x2	OK	OK
Gigabit Ethernet port	OK	OK
Ethernet port status light	OK	OK
WIFI	OK	OK
Bluetooth	OK	OK
Debugging Serial Ports	OK	OK
RTC Chip	OK	OK
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
CAN（40pin）	NO	NO
PWM（40pin）	OK	OK
OV13850 Camera	OK	OK
OV13855 Camera	OK	OK
SPI+NVME Start	OK	OK
LCD	OK	OK
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**
Watchdog Test	**OK**	**OK**
Chromium hardening video	**OK**	**OK**

3. 4. Description of the format of the linux commands in this manual

1) All commands in this manual that need to be entered on a Linux system are boxed below

```
```

As shown below, the contents of the yellow box indicate the contents that require special attention, except for the commands here.

```
```

2) Description of the type of prompt that precedes the command

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

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

a. **root@orangepi:~$** The prompt indicates that the command was entered on the development board's Linux system, and the **$** at the end of the prompt indicates that the current user of the system is an ordinary user, and that you need to add sudo when executing privileged commands. **sudo**

b. **root@orangepi:~#** The prompt indicates that the command was entered on the development board's Linux system, and the **#** at the end of the prompt indicates that the current user of the system is root, and can execute any commands desired.

c. **test@test:~$** The prompt indicates that the command was entered on an Ubuntu PC or Ubuntu virtual machine, not on the board's linux system. The **$** at the end of the prompt indicates that the current user of the system is a normal user, and when executing privileged commands, you need to add **sudo**

d. **root@test:~#** The prompt indicates that the command was entered on an Ubuntu PC or Ubuntu virtual machine, not on the development board's linux system. The # at the end of the prompt indicates that the current user of the system is root and can execute any commands he or she wishes.

3) What are the commands to be entered?

a. As shown below, the **bolded part in black** is the command that needs to be entered, and below the command is the output (some commands have output, some may not), which does not need to be inputted

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

b. As shown below, some commands can not be written in one line will be placed on the next line, as long as the black bolded part of the command are required to enter. When these commands are entered on one line, the "\" at the end of each line needs to be removed, this is not part of the command. In addition, there are spaces in different parts of the commands, so 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. 5. linux system login instructions

3. 5. 1. default login and password for linux systems

Account	Password
root	orangepi

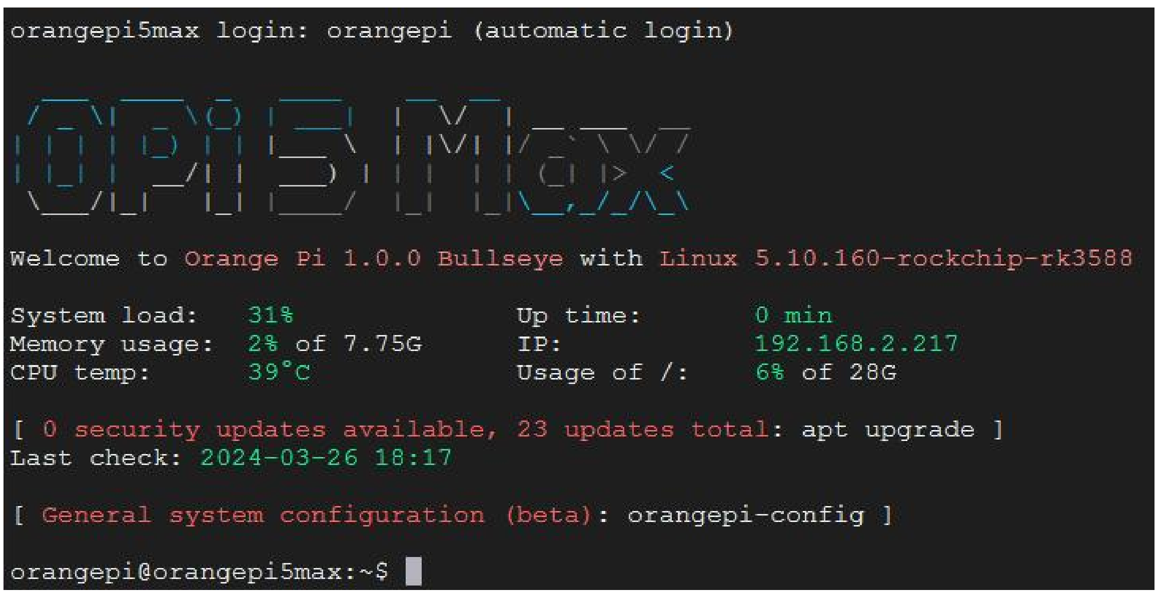
orangepi	**orangepi**

Note that when you enter the password, the screen will not display the specific content of the password entered, please do not think that there is some kind of malfunction, after entering directly back to the car can be.

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

3. 5. 2. Setting up automatic login for linux system terminals

1) By default, the linux system automatically logs into the terminal, and the default login username 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

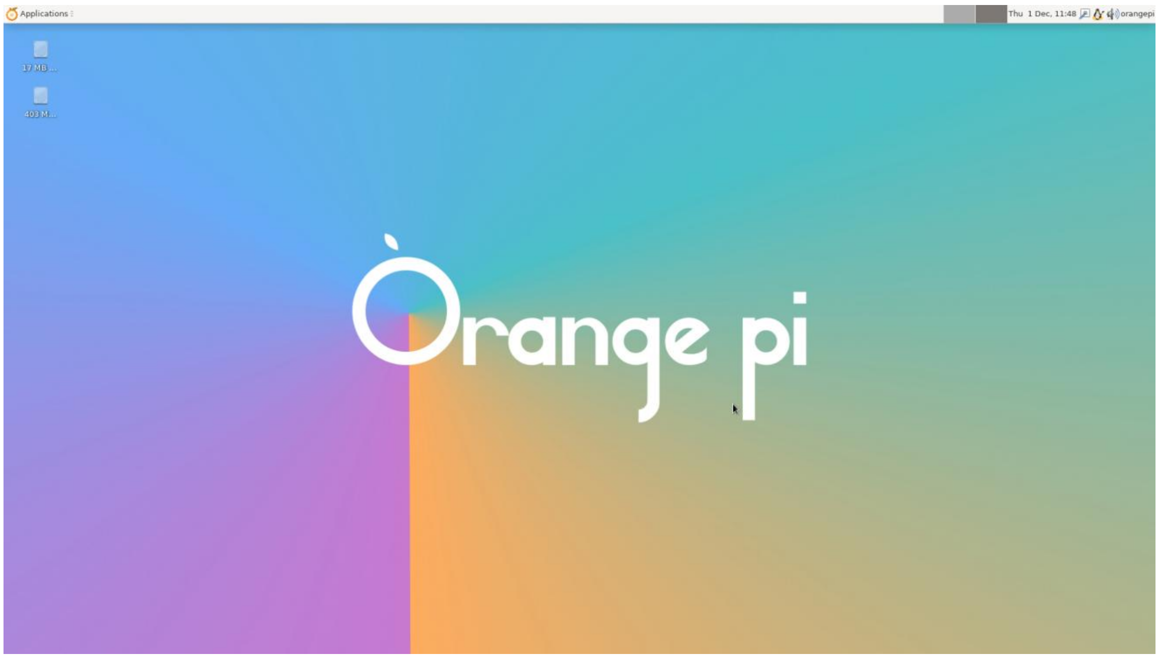
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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. 5. 3. linux desktop system auto-login instructions

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



2) Run the following command to disable the desktop version of the system from automatically logging into the desktop

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

3) Then reboot the system and the login dialog box will appear, at this time you need to enter the password to enter the system

3. 5. 4. Linux desktop system root user automatic login setting method

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

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

2) Then reboot your system and you will be automatically logged in to the desktop as the root user

Note that if you are logged into the desktop system with the root user, you will not be able to use pulseaudio in the upper right corner to manage audio devices.

Also note that this is not a bug, as pulseaudio is not allowed to run under root in the first place.

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

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

3.5.5. Disabling the Desktop on Linux Desktop Edition Systems

1) First, enter the following command at the command line, **please remember to add sudo privileges**

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

2) Then reboot your Linux system and you'll see that the desktop won't be displayed.

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

3) To reopen the desktop, proceed as follows:

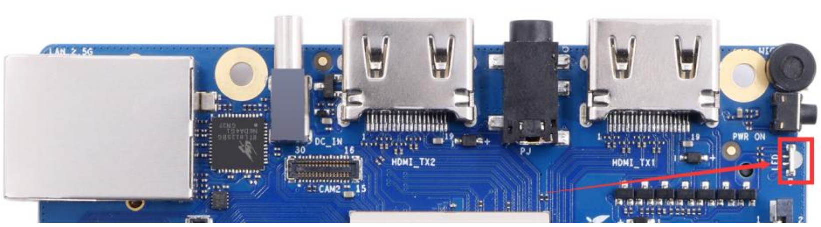
a. First, enter the following command at the command line, **please remember to add sudo privileges**

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

b. The monitor will display the desktop after the selection is made

3.6. On-Board LED Test Description

1) There is a red, green and blue light on the development board at the location shown below:



2) There is a red, green and blue light on the development board at the location shown below:

3) The green and blue LEDs will keep blinking after the kernel is booted, which is controlled by the software.

4) Set the green light on and off and blinking as follows

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

a. First enter the green light setup directory

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

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

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

5) Use the commands to set the blue light on and off and blinking as shown below:

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

a. First enter the blue light's setup directory

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

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

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

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

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

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

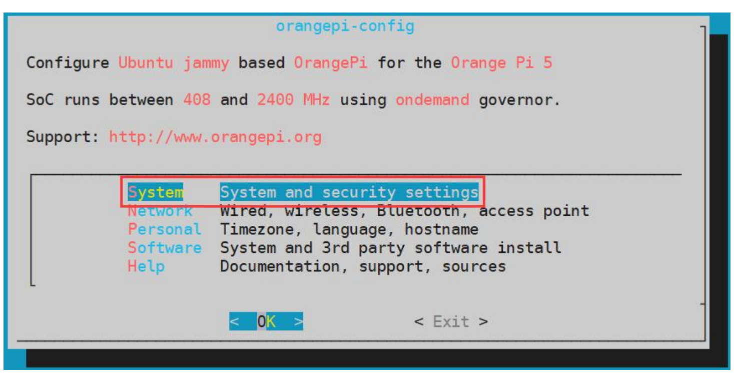
```
root@orangepi:/sys/class/leds/blue_led# echo heartbeat > trigger
```

6) If you don't need the LEDs to blink after powering up, you can use the following method to turn off the green and blue lights

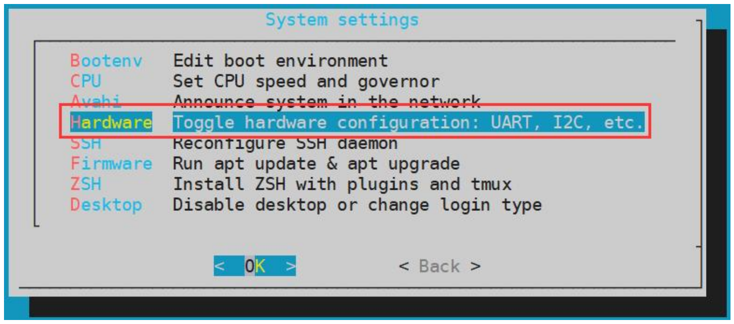
a. First run **orangepi-config**, ordinary users remember to add **sudo** privileges

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

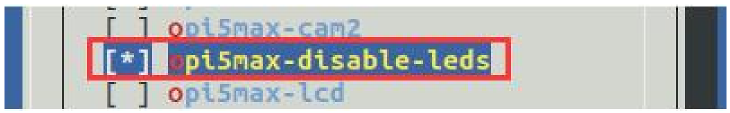
b. Then select **System**



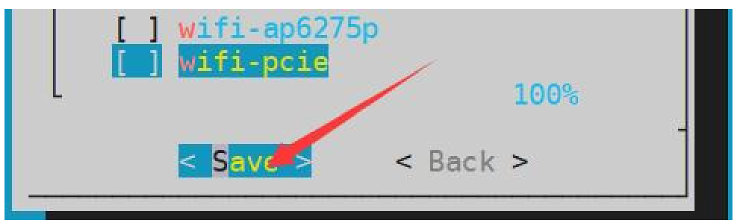
c. Select **Hardware**



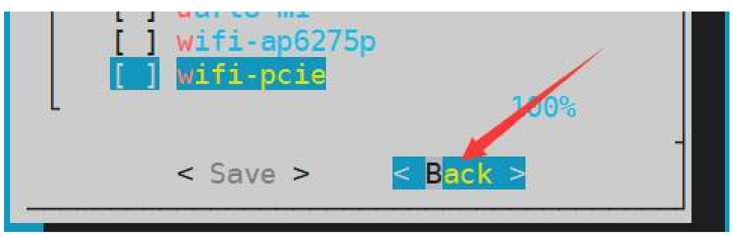
d. Then use your keyboard's arrow keys to locate the position shown in the following figure, and then use space to check the **opi5max-disable-leds** configuration



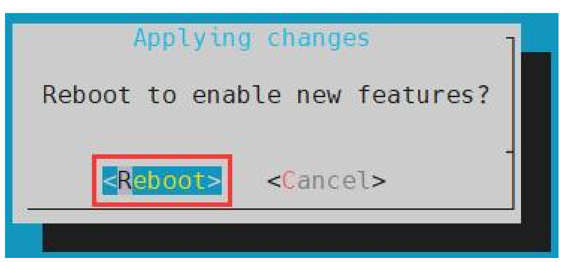
e. Then select **<Save>** to save



f. The select **<Back>**



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



h. After reboot, you can see that only the red light on the board is always on, and the green and blue lights do not blink.

3. 7. Network connectivity testing

3. 7. 1. Ethernet port test

1) First, plug one end of the cable into the Ethernet port on the board, connect the other end of the cable to the router, and make sure the network is open.

2) The system automatically assigns an IP address to the Ethernet card via DHCP after startup, **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: enP3p49s0: <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 enP3p49s0
           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 view the IP address, if you are prompted with the following message, 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
```

There are three ways to view the IP address after the development board has booted:

1. Connect the HDMI monitor, and then log on to the system to use theip addr show command to view the IP address

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

3. If there is no debugging serial port and no HDMI monitor, you can also use the router's management interface to view the IP address of the development board's network port. However, this method often people will not be able to see the IP address of the development board normally. If you can not see, the debugging method is shown below:

A）First of all, check whether the Linux system has been started normally, if the green light of the development board is blinking, it is generally a normal startup, if only the red light, it means that the system is not even started normally;

B）Check if the cable is plugged in tightly, or try a different cable;

C）Try a different router (there are many router problems encountered, such as the router can't assign an IP address properly, or the IP address has been assigned properly but can't be seen in the router);

D）If you don't have a router to replace it you can only connect an HDMI monitor or use the debug serial port to view the IP address.

Also note that the development board DHCP automatically assigns IP addresses without any settings.

4) The following commands are used to test network connectivity. The **ping** command can be interrupted by the **Ctrl+C** shortcut.

```
orangepi@orangepi:~$ ping www.baidu.com -I enP3p49s0
```

```
PING www.a.shifen.com (183.2.172.185) from 10.31.2.249 enP3p49s0: 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
```

3.7.2. WIFI connection test

Do not connect to WIFI by modifying the /etc/network/interfaces configuration file, there will be problems connecting to a WIFI network for use in this way.

3.7.2.1. Server version image connects to WIFI via command

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

1) Login to the linux system first, there are three ways as below
   a. If the board is connected to a network cable, **you can remotely log in to the linux system via ssh.**
   b. If the board is connected to a debugging serial port, you can use the serial terminal to log into the linux system.
   c. If the development board is connected to an HDMI display, you can log in to the linux system through the terminal on the HDMI display.

2) First use the **nmcli dev wifi** command to scan for WIFI hotspots around you.

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

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

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

a. **wifi_name** You need to change the name of the WIFI hotspot you want to connect to.

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

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

4) You can view the IP address of the wifi by using 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
```

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

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

3.7.2.2. Server version image graphically connected to WIFI

1) Login to the linux system first, there are three ways as below

a. af the board is connected to a network cable, **you can log in to the linux system remotely via ssh.**

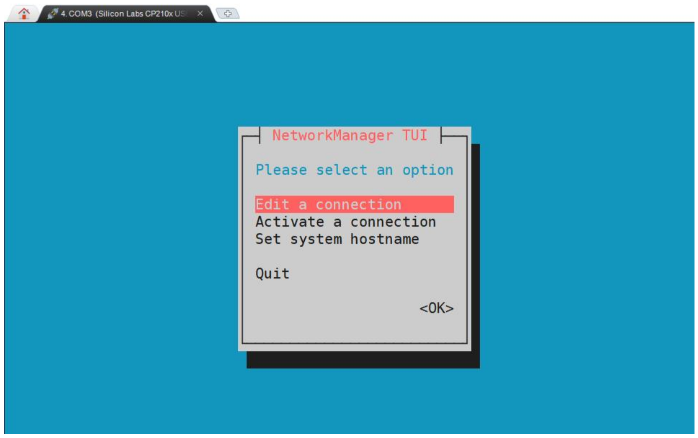
b. If the board is connected to a debugging serial port, you can use the serial terminal to log in to the linux system (please use MobaXterm for the serial software, minicom cannot display the graphical interface).

c. If the board is connected to a HDMI monitor, you can log in to the linux system through the HDMI monitor terminal.

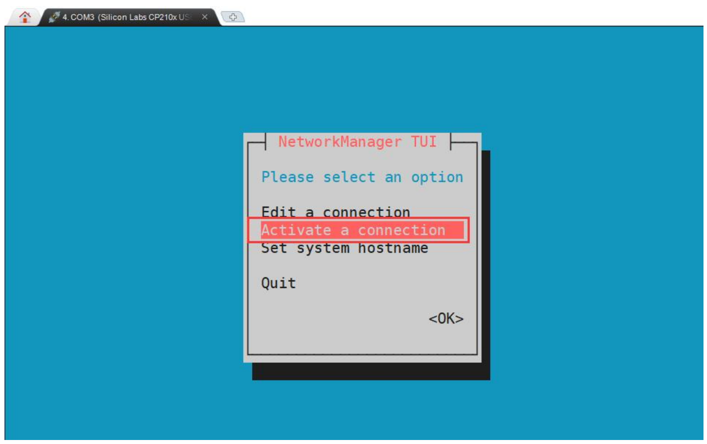
2) Then enter the nmtui command on the command line to open the wifi connection interface.

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

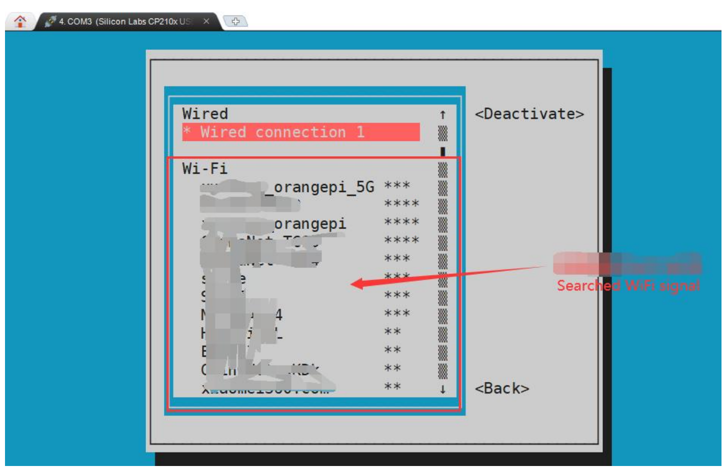
3) Entering the nmtui command opens the interface as follows



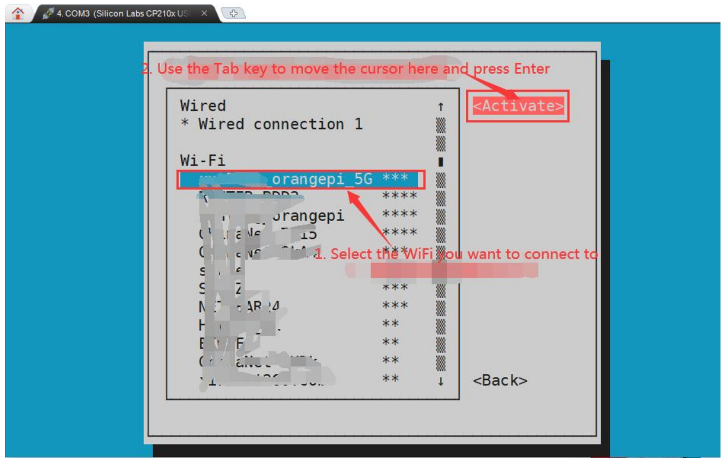
4) Select **Activate a connect** and press Enter



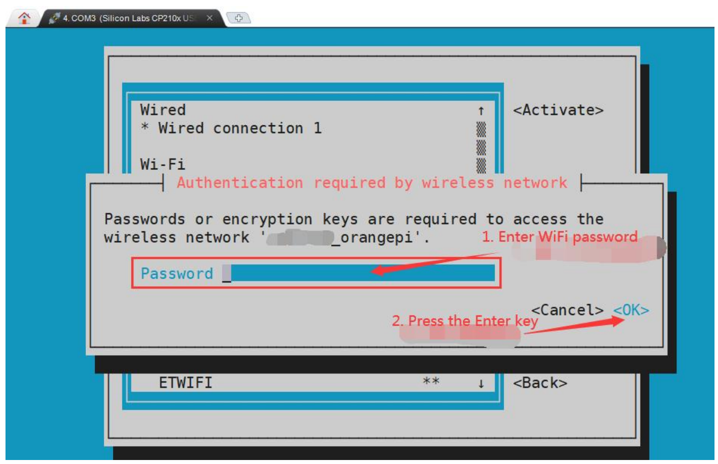
5) Then you can see all the WIFI hotspots you have searched.



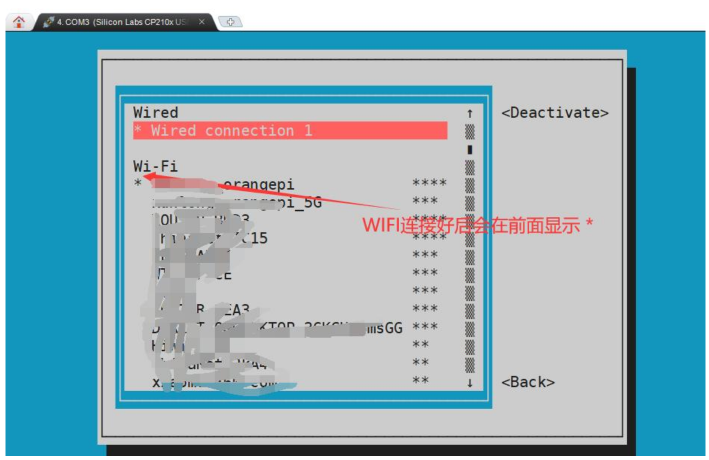
6) Select the WIFI hotspot you want to connect to and then use the Tab key to position the cursor to **Activate** and enter.



7) Then a dialog box will pop up to enter the **Password**, enter the corresponding password in Password and then enter to start connecting to WIFI.



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



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

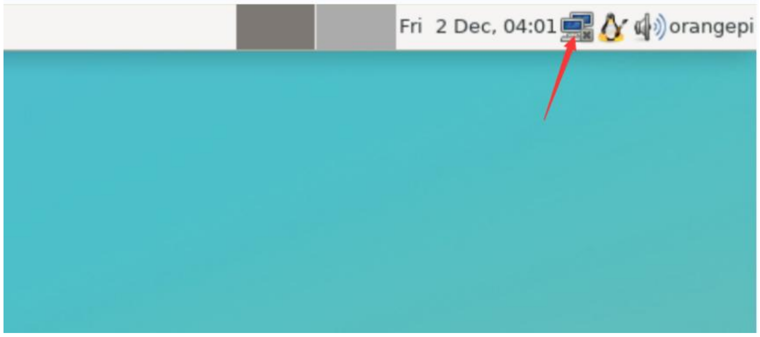
```
orangepi@orangepi:~$ ip addr show wlan0
11: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 24:8c:d3:aa:76:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
        valid_lft 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 a wifi network. The **ping** command can be interrupted by the **Ctrl+C** shortcut.

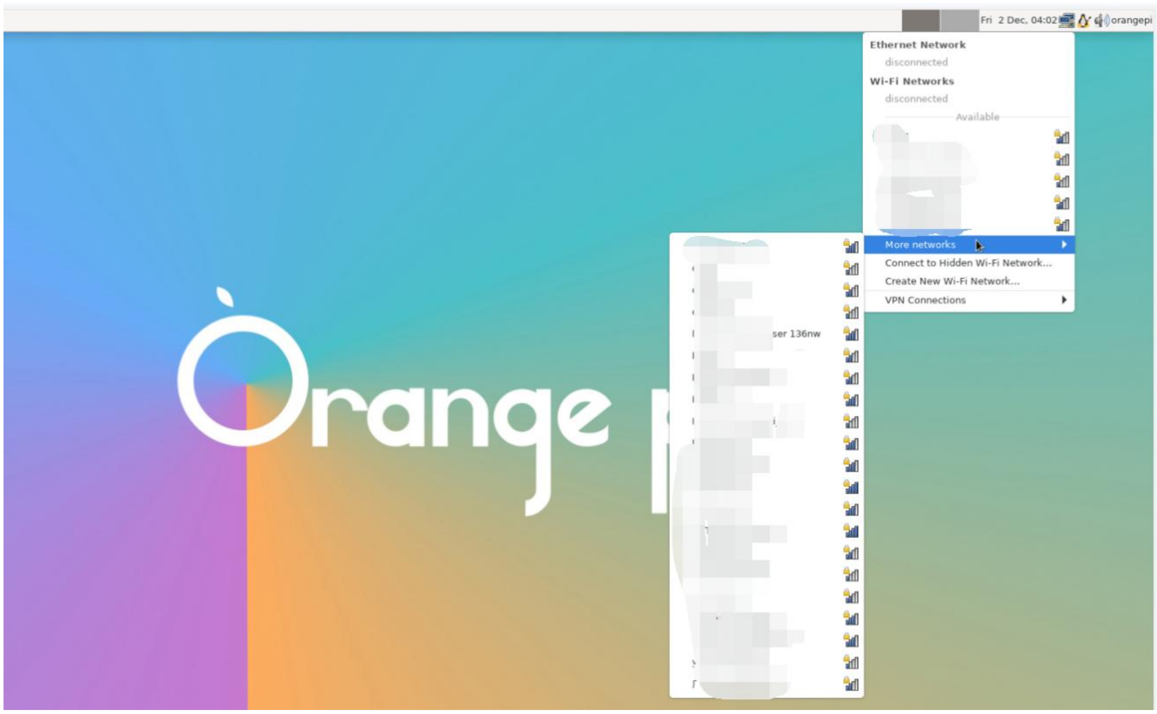
```
orangepi@orangepi:~$ ping www.orangepi.org -I wlan0
PING www.orangepi.org (182.92.236.130) from 192.168.1.49 wlan0: 56(84) bytes of
data.
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=1 ttl=52 time=43.5 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=2 ttl=52 time=41.3 ms
64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

3. 7. 2. 3. Test methods for desktop images

1) Click the Network Configuration icon on the top right corner of your desktop (please do not connect the network cable when testing WIFI)

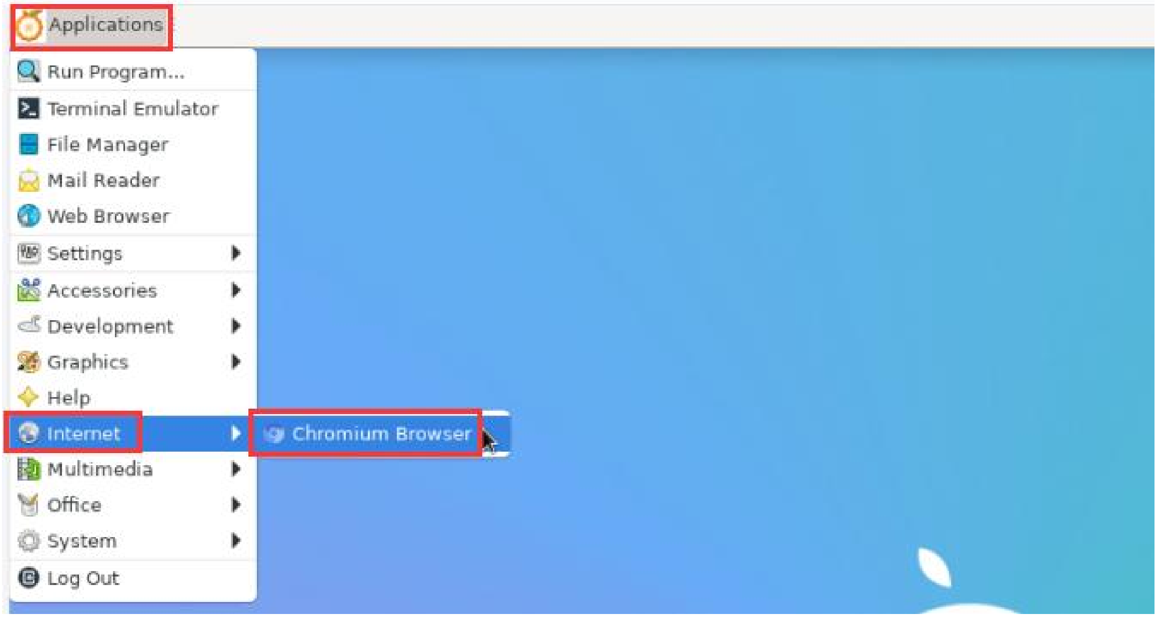


2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to.

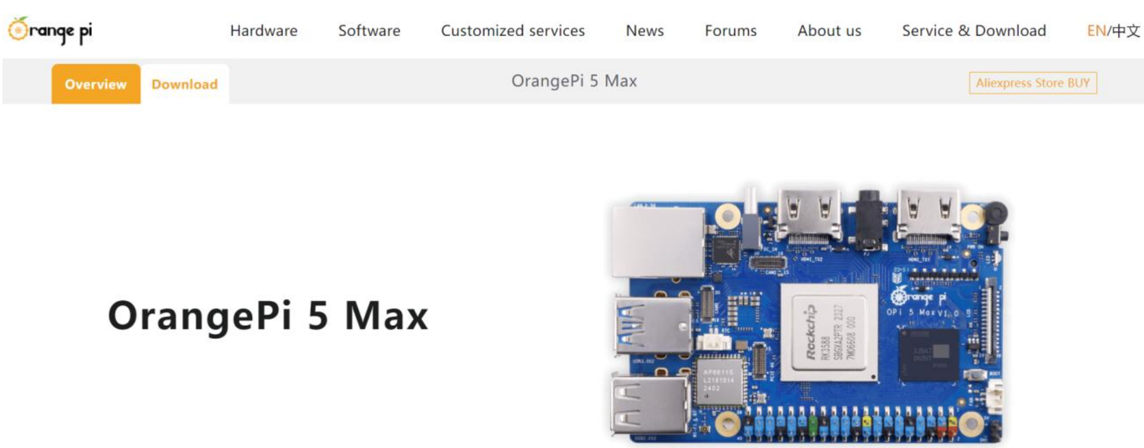


3) Then enter the password of the WIFI hotspot and click **Connect** to start connecting to WIFI.

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



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



3.7.3. Methods for setting static IP addresses

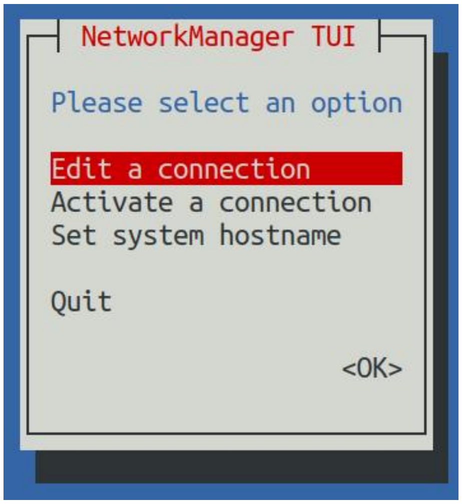
Do not set a static IP address by modifying the /etc/network/interfaces configuration file.

3.7.3.1. Using the nmtui command to set a static IP address

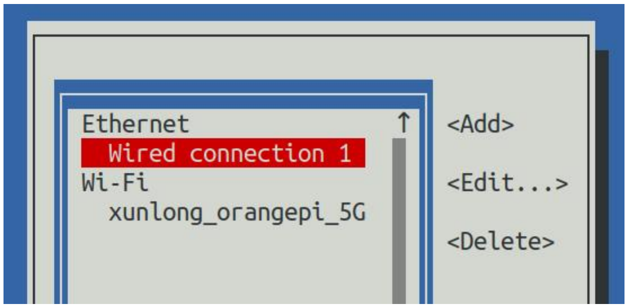
1) First run the **nmtui** command

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

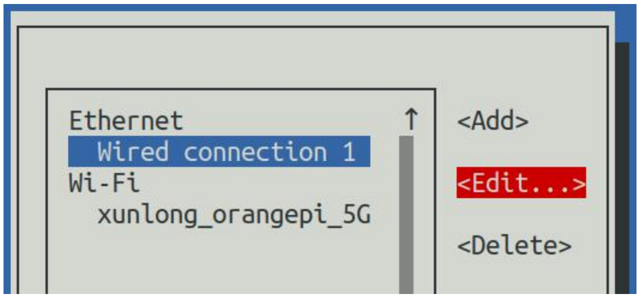
2) Then select **Edit a connection** and press enter.



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

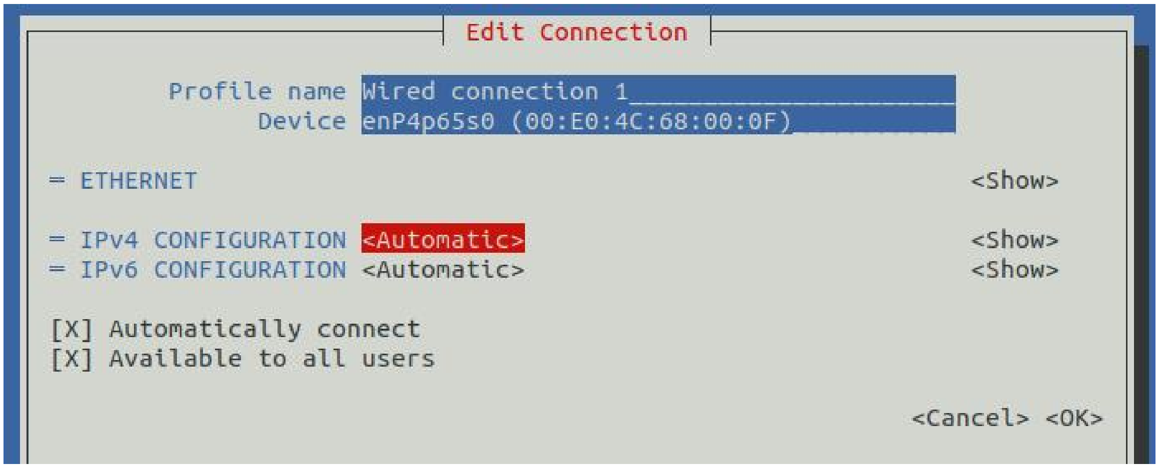


4) Then select **Edit** with the **Tab** key and press Enter.

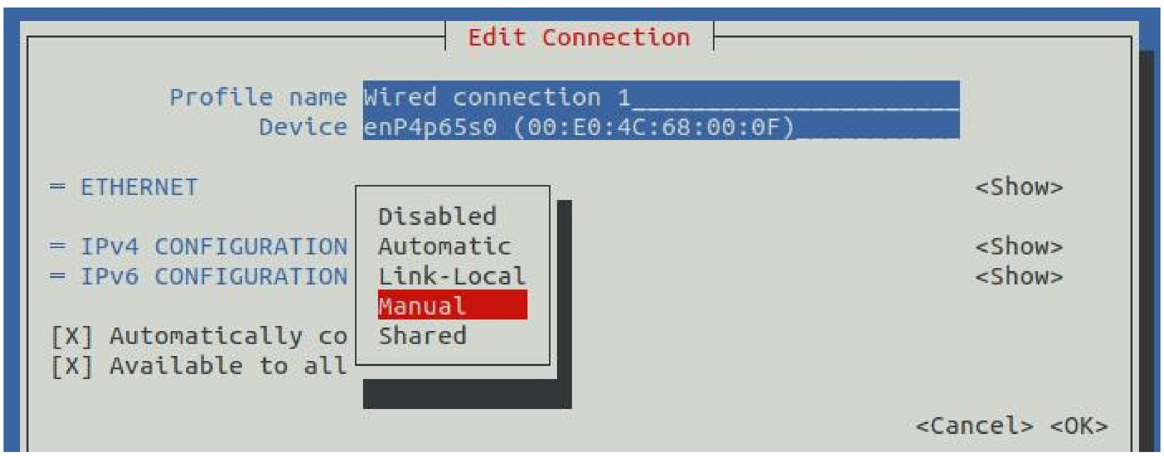


5) Then use the Tab key to move the cursor to the **<Automatic>** position as shown in the

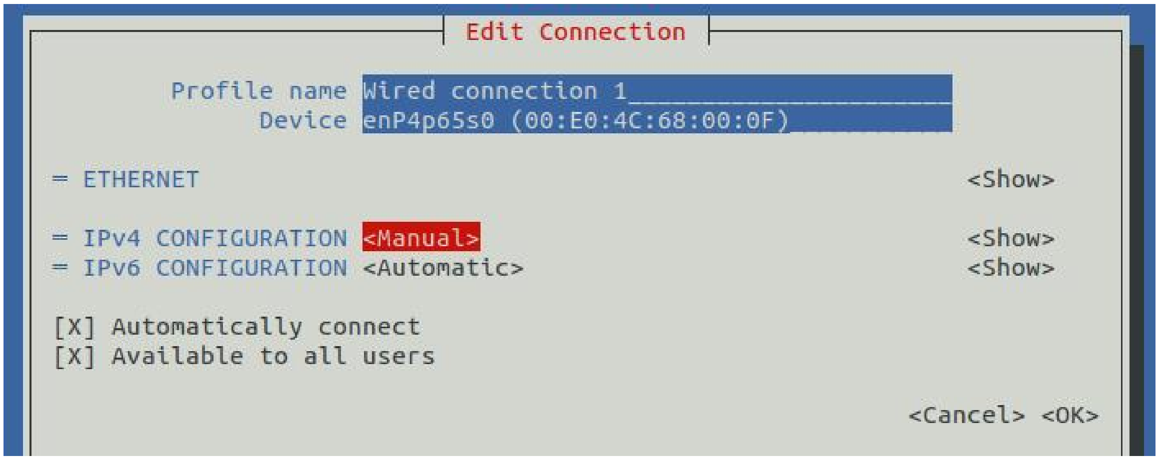
figure below to configure IPv4.



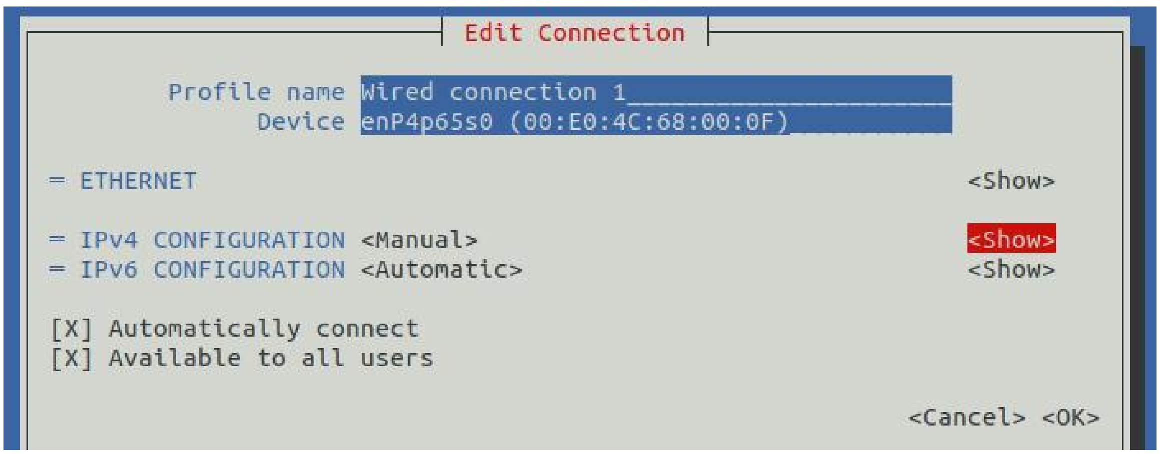
6) Then enter, select **Manual** with the up and down arrow keys, and then enter to confirm.



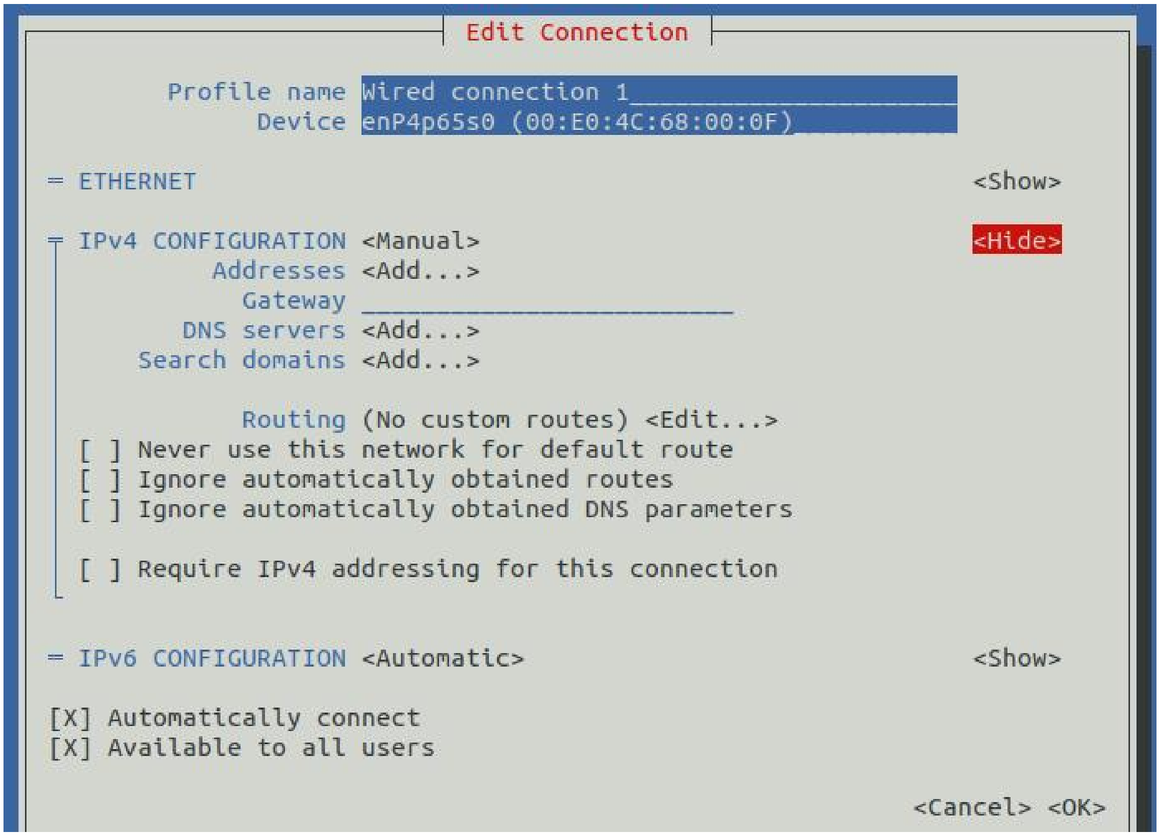
7) The display after selection is shown below



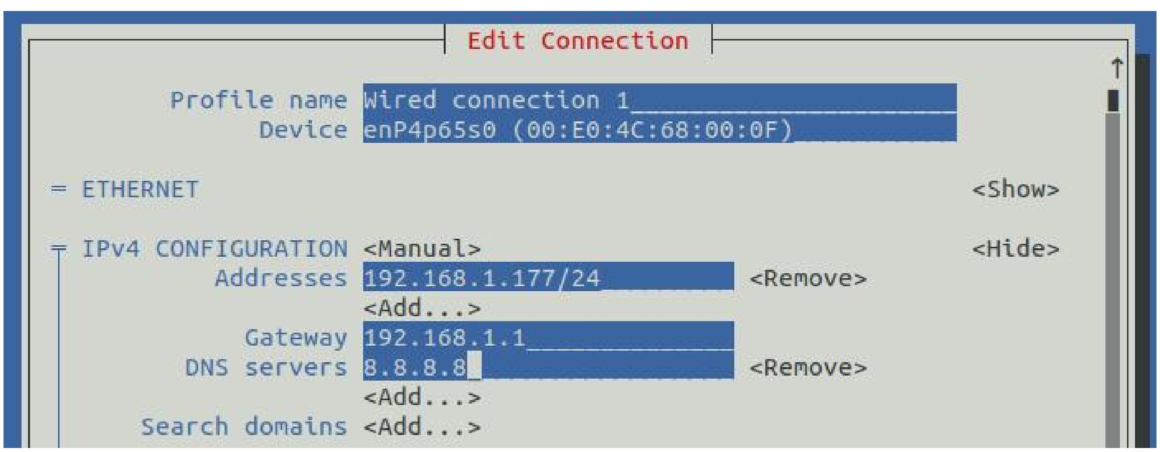
8) Then move the cursor to **<Show>** by using the Tab key.



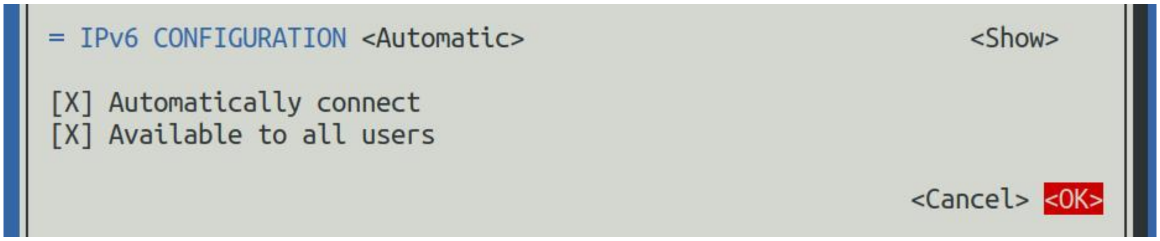
9) and then enter, enter will pop up the following settings interface



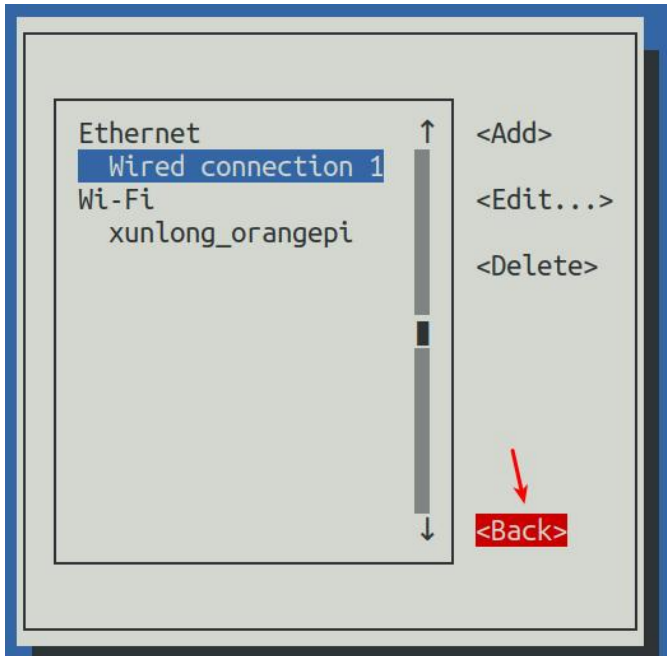
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the location shown in the following figure (there are many other settings inside the option, please explore), **according to their specific needs to set up the value of the following figure is just an example of the settings)**



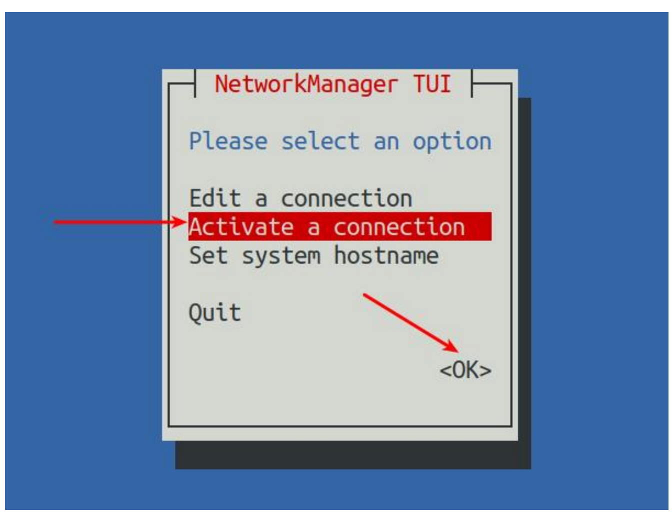
11) Move the cursor to **<OK>** in the lower right corner after setting, then enter to confirm.



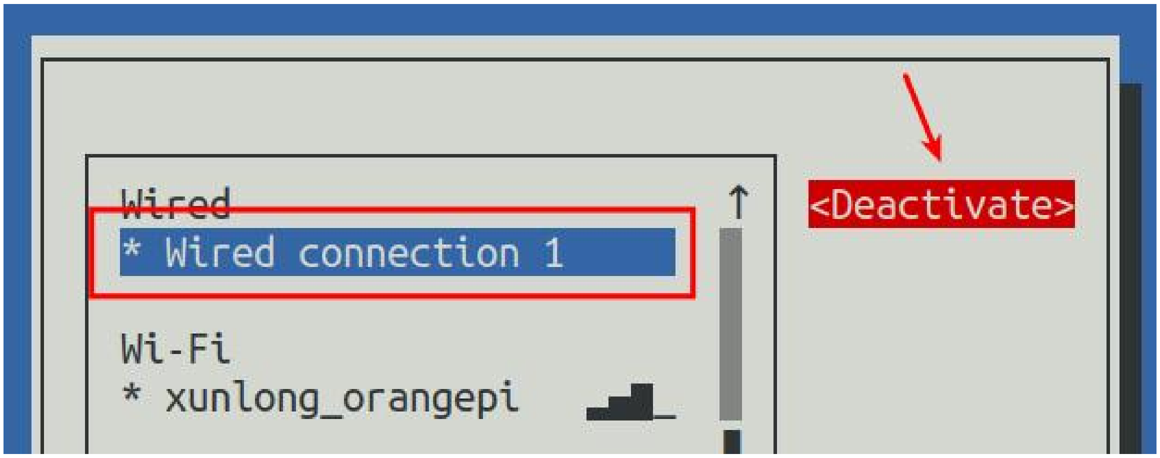
12) Then click **<Back>**to go back to the previous selection screen.



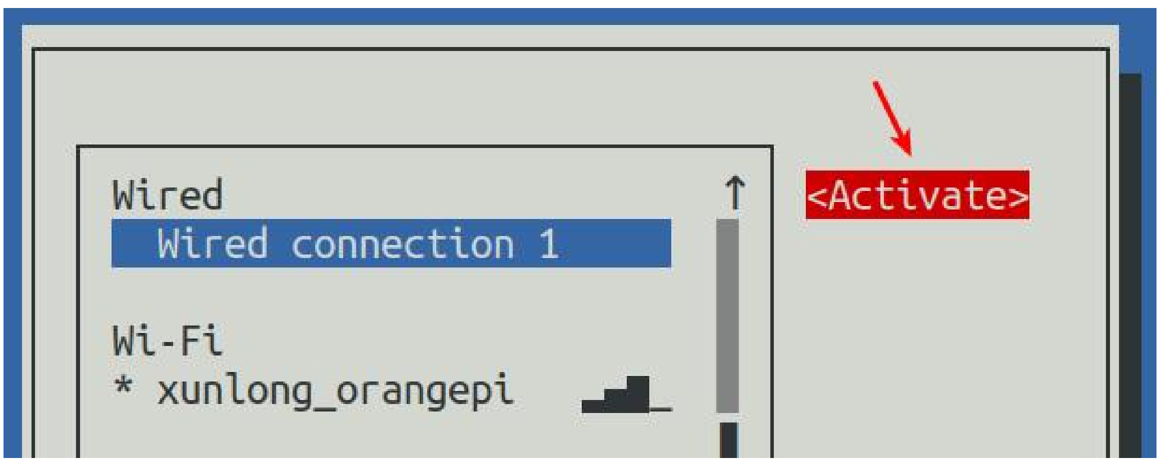
13) Then select **Activate a connection**, then move the cursor to**<OK>**, and finally click Enter.



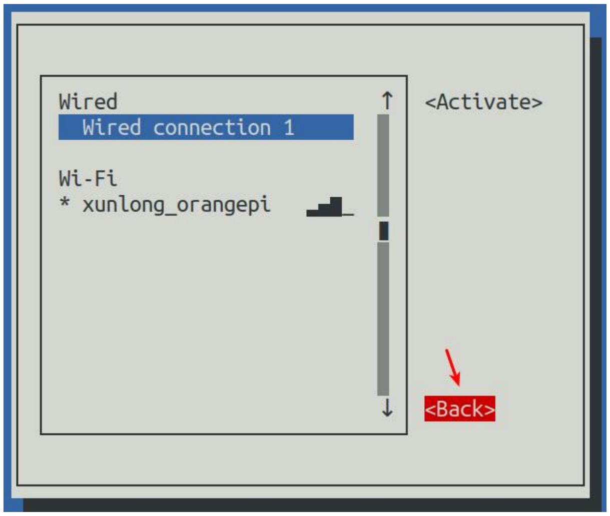
14) Then select the network interface that needs to be set, such as **Wired connection 1**, and then move the cursor to **<Deactivate>**, and then press the Enter key to disable **Wired connection 1**

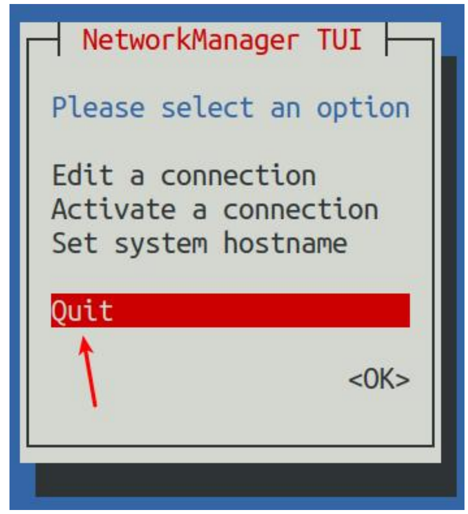


15) Then please do not move the cursor, and then press the Enter key to re-enable the **Wired connection 1**, so that the static IP address set earlier will take effect



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





17) Then through the **ip addr show enP3p49s0** you can see that the IP address of the network interface has been changed to the static IP address set earlier.

```
orangepi@orangepi:~$ ip addr show enP3p49s0
2: enP3p49s0: <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 enP3p49s0
       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 connectivity of the network to check whether the IP address is configured OK, **ping** command can be interrupted by the **Ctrl+C** shortcut to run the command

```
orangepi@orangepi:~$ ping 192.168.1.47 -I enP3p49s0
PING 192.168.1.47 (192.168.1.47) from 192.168.1.188 eth0: 56(84) bytes of data.
64 bytes from 192.168.1.47: icmp_seq=1 ttl=64 time=0.233 ms
64 bytes from 192.168.1.47: icmp_seq=2 ttl=64 time=0.263 ms
64 bytes from 192.168.1.47: icmp_seq=3 ttl=64 time=0.273 ms
64 bytes from 192.168.1.47: icmp_seq=4 ttl=64 time=0.269 ms
64 bytes from 192.168.1.47: icmp_seq=5 ttl=64 time=0.275 ms
^C
--- 192.168.1.47 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4042ms
```

```
rtt min/avg/max/mdev = 0.233/0.262/0.275/0.015 ms
```

3.7.3.2. Using the nmcli command to set a static IP address

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

2) Then you can view the name of the network device by using the **nmcli con show** command as follows

a. **orangepi** is the name of the WIFI network interface (names do not have to 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, where

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

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

c. **ipv4.gateway** indicates 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 reboot 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 enP3p49s0
3: enP3p49s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 5e:ae:14:a5:91:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.110/32 brd 192.168.1.110 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 240e:3b7:3240:c3a0:97de:1d01:b290:fe3a/64 scope global dynamic noprefixroute
        valid_lft 259183sec preferred_lft 172783sec
    inet6 fe80::3312:861a:a589:d3c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

3.7.4. Creating a WIFI hotspot via create_ap

create_ap is a script to help quickly create a WIFI hotspot on Linux, and support bridge and NAT mode, can automatically combine hostapd, dnsmasq and iptables to complete the WIFI hotspot setup, avoiding the user to carry out complex configurations, github address is as follows:

https://github.com/oblique/create_ap

If you are using the latest image, the create_ap script is already pre-installed, and you can create a WIFI hotspot by using the create_ap command, the basic command format of create_ap is shown below:

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

*** options: You can use this parameter to specify the encryption method, the band of the WIFI hotspot, the bandwidth mode, the network sharing method, and so on, and you can use create_ap -h to get what options are available.**

*** wifi-interface: Name of the wireless card**

*** interface-with-internet: Name of the network card that can be networked, usually eth0**

*** access-point-name： Hot Spot Name**

*** passphrase：The code of the hotspot**

3. 7. 4. 1. create_ap to create a WIFI hotspot in NAT mode

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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP3p49s0 orangepi orangepi
```

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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP3p49s0 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 cell phone, in the list of searched WIFI can be found in the development board created by the name **orangepi** WIFI hotspot, and then you can click orangepi to connect to the hotspot, the password is set above **orangepi**!

4）The display after successful connection is shown below

5）In NAT mode, the wireless device connected to the development board hotspot is requesting an IP address from the development board's DHCP service, so there will be two different network segments, such as the IP of the development board here is 192.168.1.X

```
orangepi@orangepi:~$ ifconfig enP3p49s0
enP3p49s0: 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 DHCP service of the development board will assign an IP address of **192.168.12.0/24** to the device that accesses the hotspot by default. At this time, click on the WIFI hotspot **orangepi** that has been connected, and then you can see that the IP address of the phone is **192.168.12.X**

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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP3p49s0 orangepi orangepi -g 192.168.2.1
```

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

7）In the case of not specifying the **--freq-band** parameter, the default hotspot created is the 2.4G band, if you want to create a hotspot for the 5G band you can specify the **--freq-band 5** parameter, the specific commands are as follows

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 enP3p49s0 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 enP3p49s0 orangepi orangepi --hidden
```

At this time the phone can not search the WIFI hotspot, you need to manually specify the WIFI hotspot name and enter the password to connect to the WIFI hotspot

3. 7. 4. 2. create_ap to create a WIFI hotspot in bridge mode

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP3p49s0 orangepi
orangepi
```

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP3p49s0 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 cell phone, in the list of searched WIFI can be found in the development board created by the name of **orangepi**WIFI hotspot, and then you can click on the **orangepi**to connect to the hotspot, the password is set above the **orangepi**

4）The display after successful connection is shown below

5）In bridge mode, the wireless device connected to the board's hotspot also requests an IP address from the DHCP service of the main router (the router to which the board is connected), e.g., the board's IP is **192.168.1.X**

```
orangepi@orangepi:~$ ifconfig enP3p49s0
enP3p49s0: 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 accessing the WIFI hotspot is also assigned by the main route, so the cell phone connecting to the WIFI hotspot and the development board are in the same network segment. At this time, click on the WIFI hotspot **orangepi** that has been connected, and then you can see that the cell phone's IP address is **192.168.1.X** as well.

6） In the case of not specifying the **--freq-band** parameter, the default hotspot created is the 2.4G band, if you want to create a hotspot for the 5G band you can specify the **--freq-band 5** parameter, the specific commands are as follows

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP3p49s0 orangepi orangepi --freq-band 5
```

7） If you need to hide the SSID, you can specify the **--hidden** parameter with the following command

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 enP3p49s0 orangepi orangepi
--hidden
```

At this time the phone can not search the WIFI hotspot, you need to manually specify the WIFI hotspot name and enter the password to connect to the WIFI hotspot

3. 8. SSH remote login to the development board

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

3. 8. 1. SSH Remote Login to Development Board under Ubuntu

1) Get the IP address of the development board

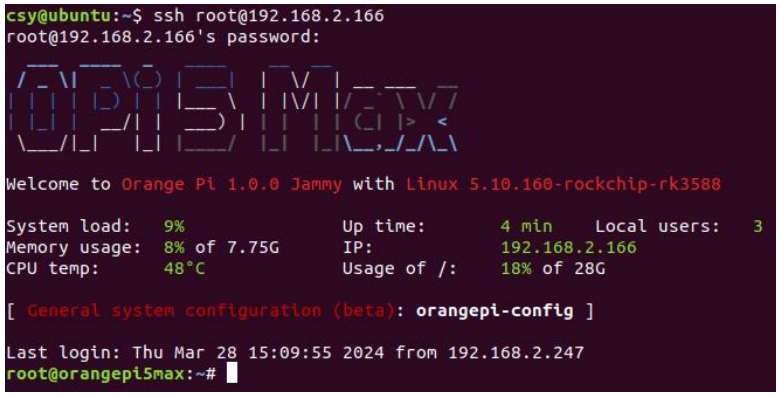
2) Then you can remotely log in to the linux system via the ssh command

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

Note that when entering the password, the screen will not show the exact content of the password entered, please do not think that there is some kind of malfunction, just enter directly after typing.

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

3) The display after successfully logging in the system is shown below



If ssh can't log into linux system normally, first of all, please check whether the IP address of the development board can be pinged, if the pinging is OK, you can log into the linux system through the serial port or the HDMI monitor and then try to connect again after inputting the following commands on the development board:

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

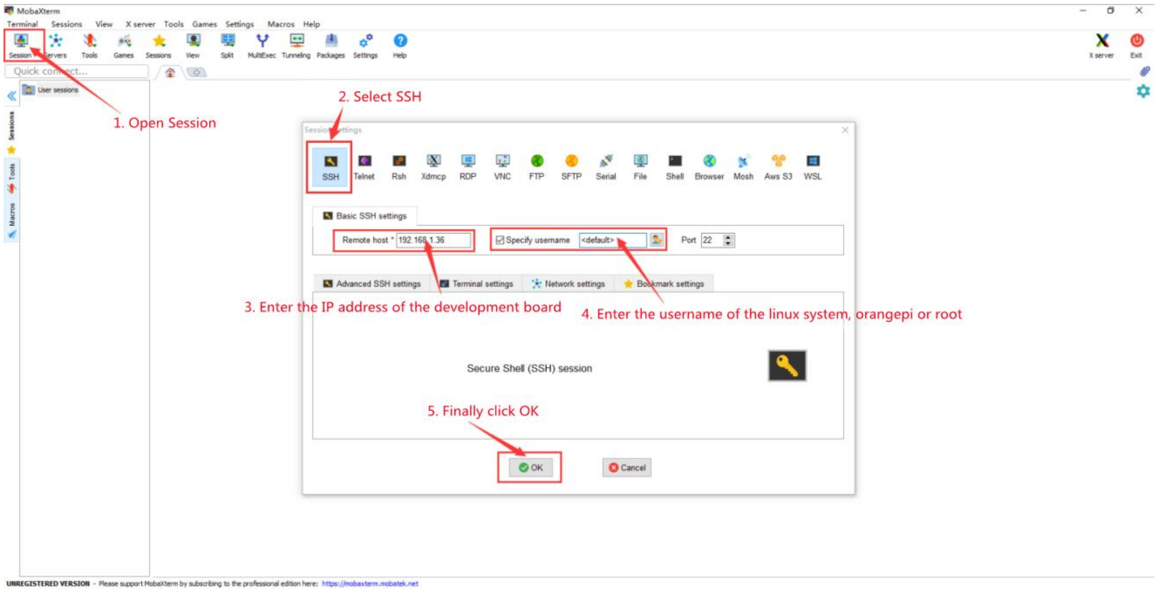
If that doesn't work, try rebooting the system.

3. 8. 2. SSH Remote Login to Development Board under Windows

1) First get the IP address of the development board

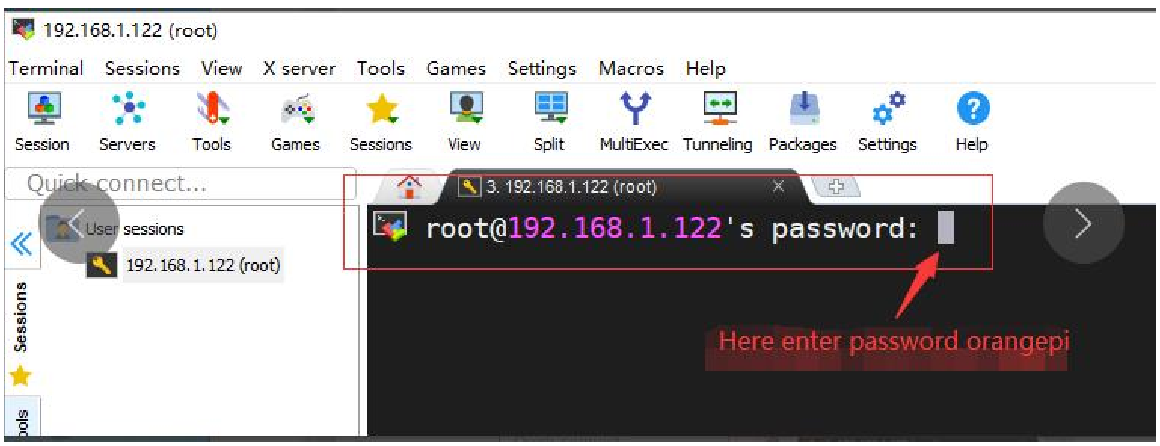
2) You can use MobaXterm to remotely login to the board under windows, first create a new ssh session.

a. Open **Session**
b. Select **SSH** in **Session Setting**.
c. In **Remote hos**, enter the IP address of the board.
d. In **Specify username**, enter the linux username **root** or **orangepi**.
e. Finally, click **OK**.

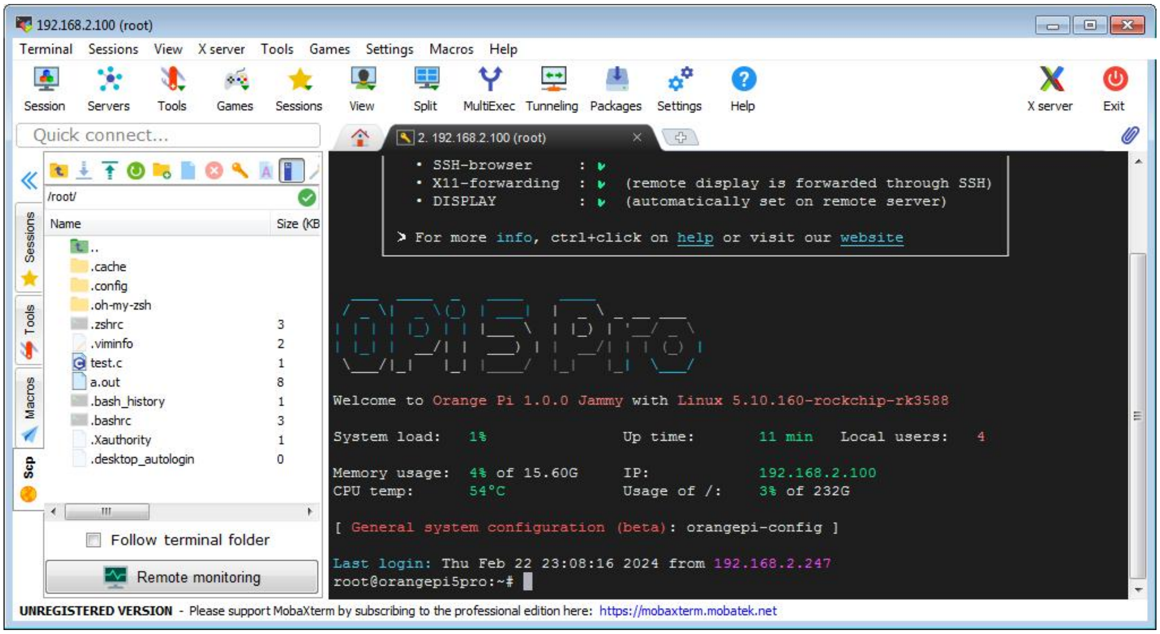


3) Then you will be prompted for a password, the default password for both root and orangepi users is orangepi

Note that when you enter the password, the screen will not display the specific content of the password entered, please do not think that there is some kind of malfunction, after entering directly back to the car can be.



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



3. 9. How ADB is used

3. 9. 1. network adb usage

1) Please make sure that adbd has started after the system boots up.

```
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 board and write it down.

3) Then install the adb utility on the Ubuntu PC

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

4) Then use the following command to connect to the network adb

```
test@test:~$ adb connect 192.168.1.xx:5555     #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 development board's linux system

```
test@test:~$ adb shell
    root@orangepi5max:/#     <--- When you see this prompt, you have successfully logged on to the board.
```

6) The command to upload a file to the development board using adb is as follows

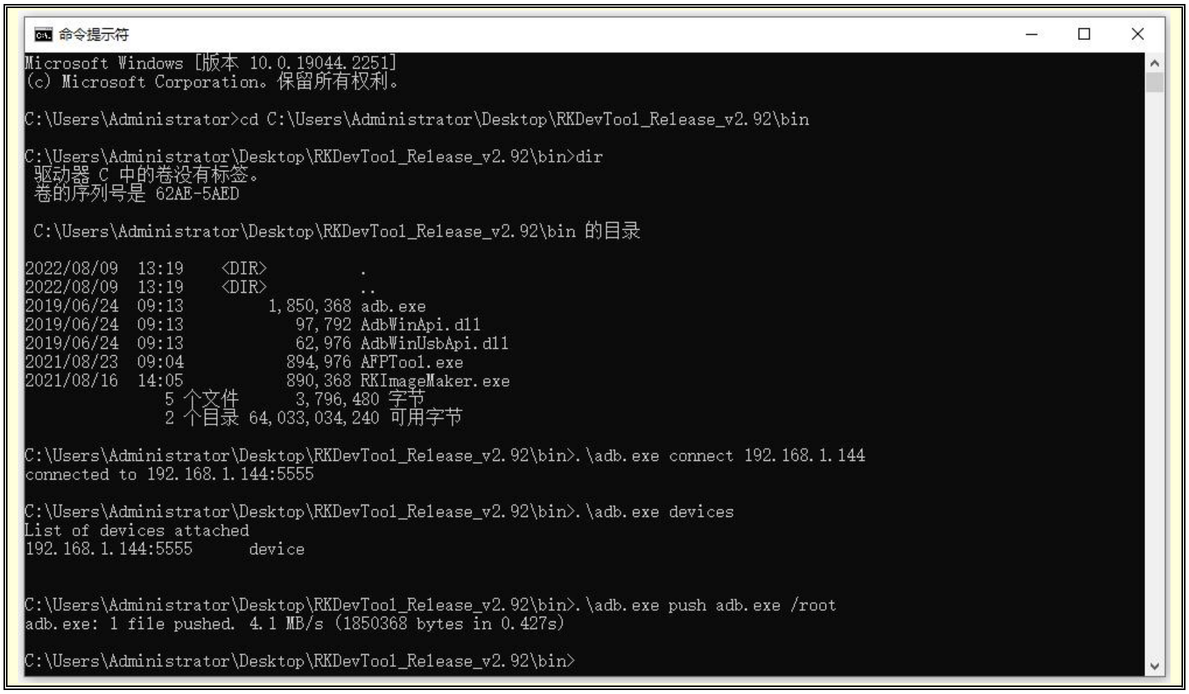
```
test@test:~$ adb push filename /root
filename: 1 file pushed. 3.7 MB/s (1075091 bytes in 0.277s)
```

7) The command to reboot the board using adb is as follows

```
test@test:~$ adb reboot
```

If you don't have an adb tool on your Windows system, you can use the adb program in the RKDevTool software (the Methods for Burning Android Images to SPIFlash+NVMe SSDs subsection is useful for this software).

An example of using adb in Windows is shown below:

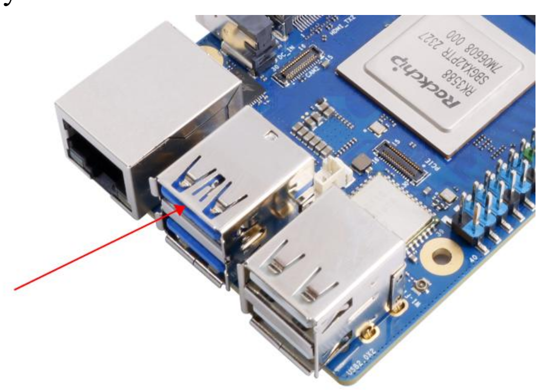


```
命令提示符
Microsoft Windows [版本 10.0.19044.2251]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Administrator>cd C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>dir
 驱动器 C 中的卷没有标签。
 卷的序列号是 62AE-5AED

 C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin 的目录

2022/08/09  13:19    <DIR>          .
2022/08/09  13:19    <DIR>          ..
2019/06/24  09:13         1,850,368 adb.exe
2019/06/24  09:13            97,792 AdbWinApi.dll
2019/06/24  09:13            62,976 AdbWinUsbApi.dll
2021/08/23  09:04           894,976 AFPTool.exe
2021/08/16  14:05           890,368 RKImageMaker.exe
               5 个文件      3,796,480 字节
               2 个目录 64,033,034,240 可用字节

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>.\adb.exe connect 192.168.1.144
connected to 192.168.1.144:5555

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>.\adb.exe devices
List of devices attached
192.168.1.144:5555      device


C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>.\adb.exe push adb.exe /root
adb.exe: 1 file pushed. 4.1 MB/s (1850368 bytes in 0.427s)

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>
```

3.9.2. Connecting adb using a USB 2.0 male-to-male cable

1) Prepare a good quality USB2.0 male-to-male cable.

2) Prepare a good quality USB2.0 male-to-male cable.



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 on your linux system, 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 make sure adbd is up and running.**

```
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 your Ubuntu PC.

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

6) Then use the following command to see if the adb device is recognized

```
test@test:~$ adb devices
List of devices attached
    e0f9f71bc343c305    device
```

8) Then use the following command to log into the development board's linux system

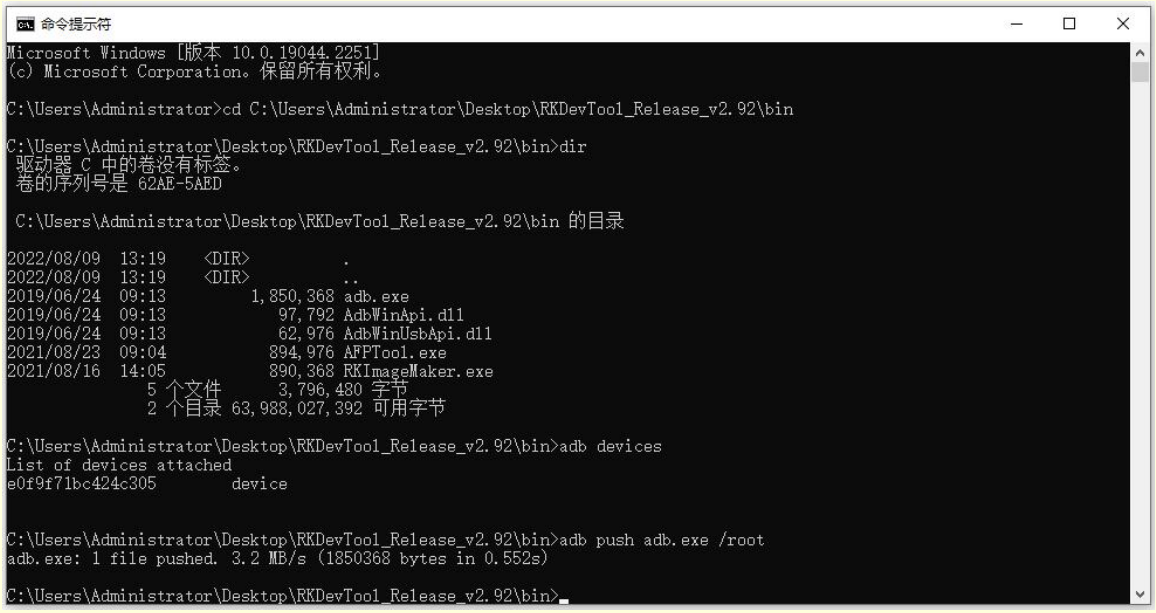
```
test@test:~$ adb shell
    root@orangepi5max:/#    <--- When you see this prompt, you have successfully logged on to the board.
```

9) The command to upload a file to the development board using adb is as follows

```
test@test:~$ adb push filename /root
filename: 1 file pushed. 3.7 MB/s (1075091 bytes in 0.277s)
```

If you don't have an adb tool on your Windows system, you can use the adb program in the RKDevTool software (the Methods for Burning Android Images to SPIFlash+NVMe SSDs subsection is useful for this software).

An example of using adb in Windows is shown below:



3. 10. Uploading Files to the Development Board Linux System

3. 10. 1. Uploading files from Ubuntu PC to the development board Linux system

3. 10. 1. 1. uploading files using the scp command

1) Use the scp command to upload files from Ubuntu PC to the Linux system of the development board, the specific command is as follows

a. **file_path:** Needs to be replaced with the path of the file to be uploaded
b. **orangepi:** is the username of the linux system of the development board, which can be replaced by other ones, such as root.
c. **192.168.xx.xx:** is the IP address of the development board, please change it according to the actual situation.
d. **/home/orangepi:** The path in the linux system of the development board can be

changed to other paths.

```
test@test:~$ scp   file_path   orangepi@192.168.xx.xx:/home/orangepi/
```

2) If you want to upload a folder, you need to add the -r parameter

```
test@test:~$ scp   -r   dir_path   orangepi@192.168.xx.xx:/home/orangepi/
```

3) There are more uses for scp, check the man page with the following commands

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

3. 10. 1. 2. Uploading files using filezilla

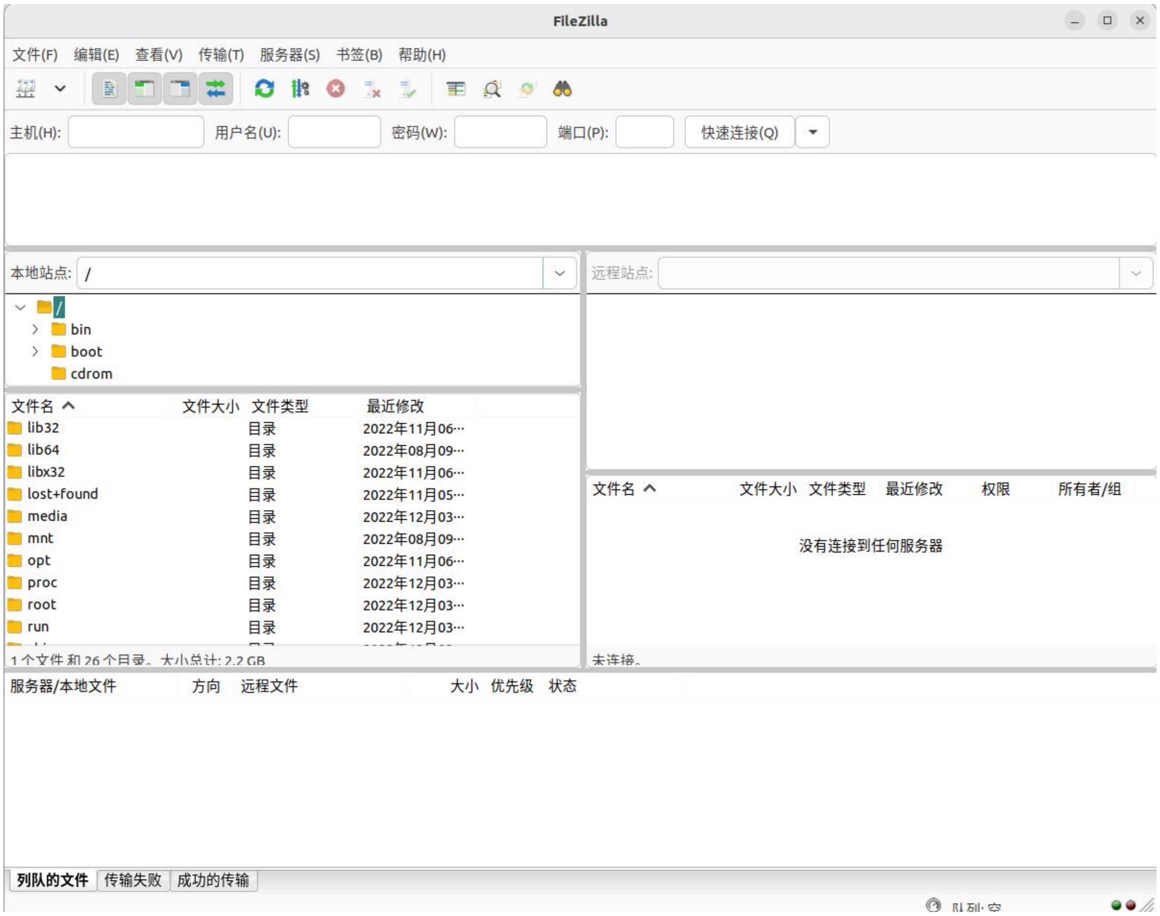
1) First install filezilla on your Ubuntu PC.

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

2) Then open filezilla with the following command

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

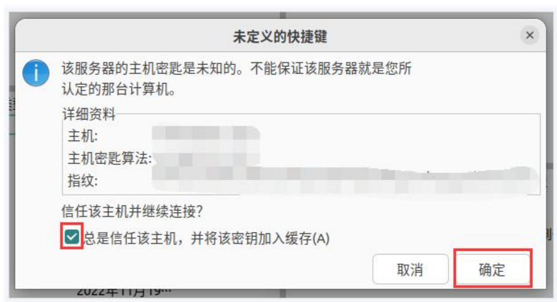
3) filezilla opens with the following interface, at this time the right side of the remote site below the display is empty



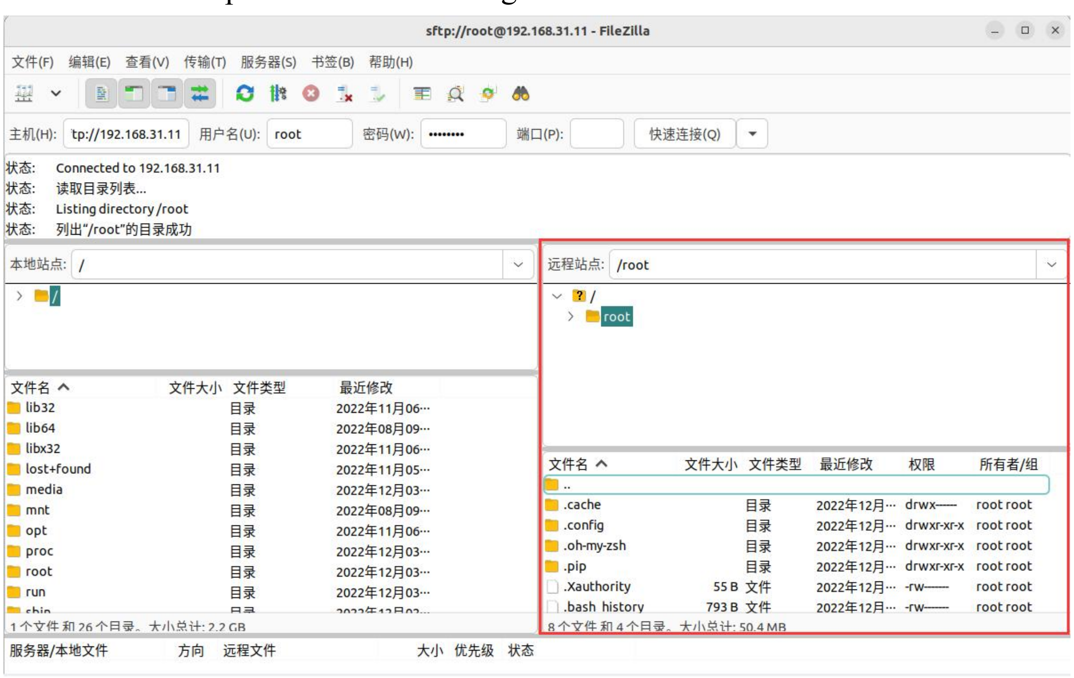
4) The method of connecting to the development board is shown in the following figure

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

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

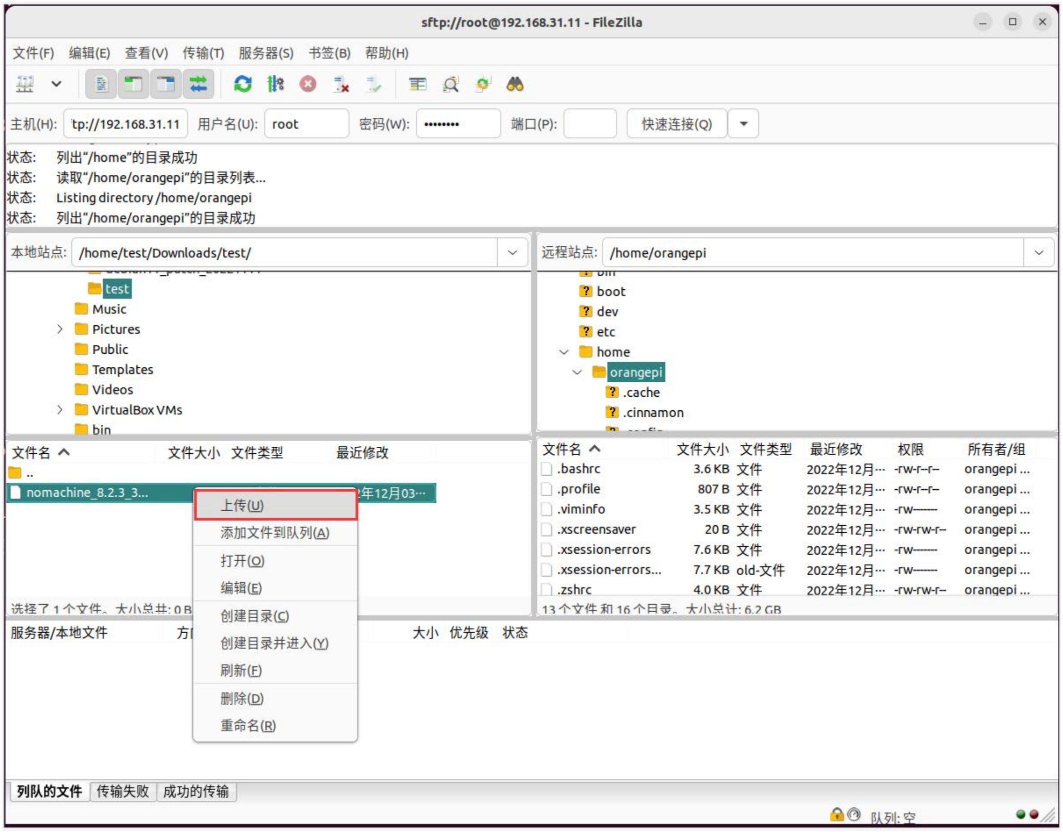


7) After successful connection, you can see the directory structure of the linux file system of the development board on the right side of the filezilla software.



8) Then select the path you want to upload to the board on the right side of the filezilla

software, then select the file you want to upload on the Ubuntu PC on the left side of the filezilla software, then click the right button of the mouse, and then click the upload option to start uploading the file to the development board.



9) After the upload is complete, you can go to the corresponding path in the linux system of the development board to check the uploaded files.

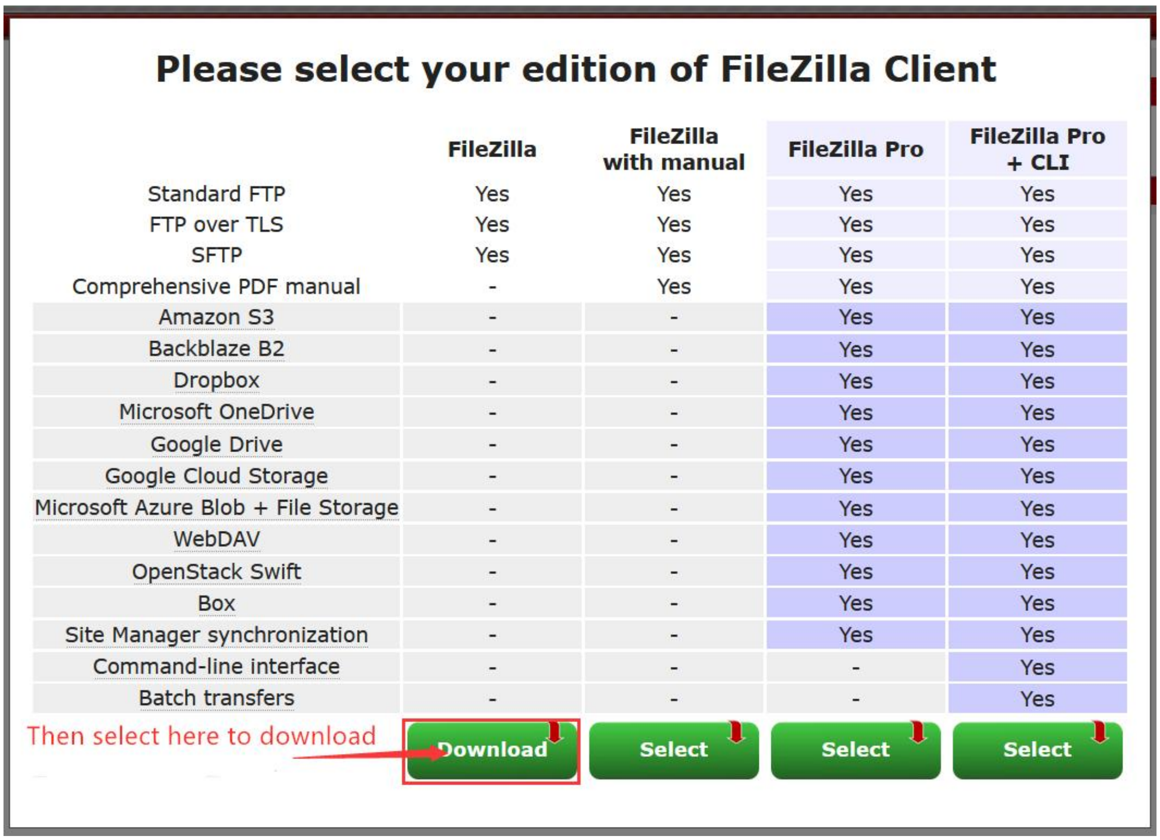
10) The method of uploading a folder is the same as uploading a file, so I won't repeat it here.

3. 10. 2. Uploading files from a Windows PC to the development board's Linux system

3. 10. 2. 1. Uploading files using filezilla

1) First of all, download the Windows version of the filezilla software installation file, the download link is as follows

https://filezilla-project.org/download.php?type=client



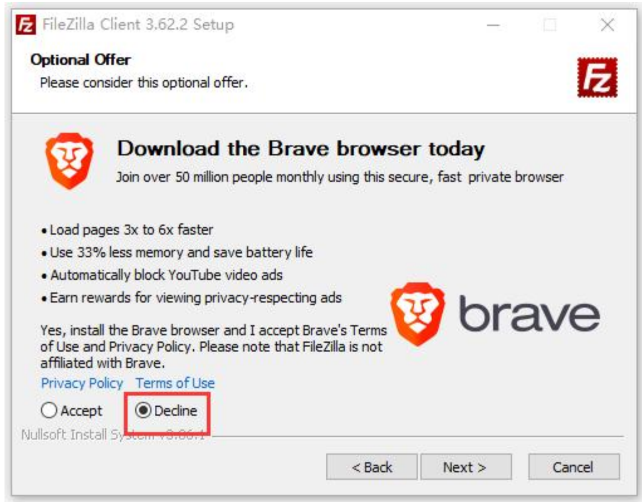
Please select your edition of FileZilla Client

	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
Then select here to download	Download	Select	Select	Select

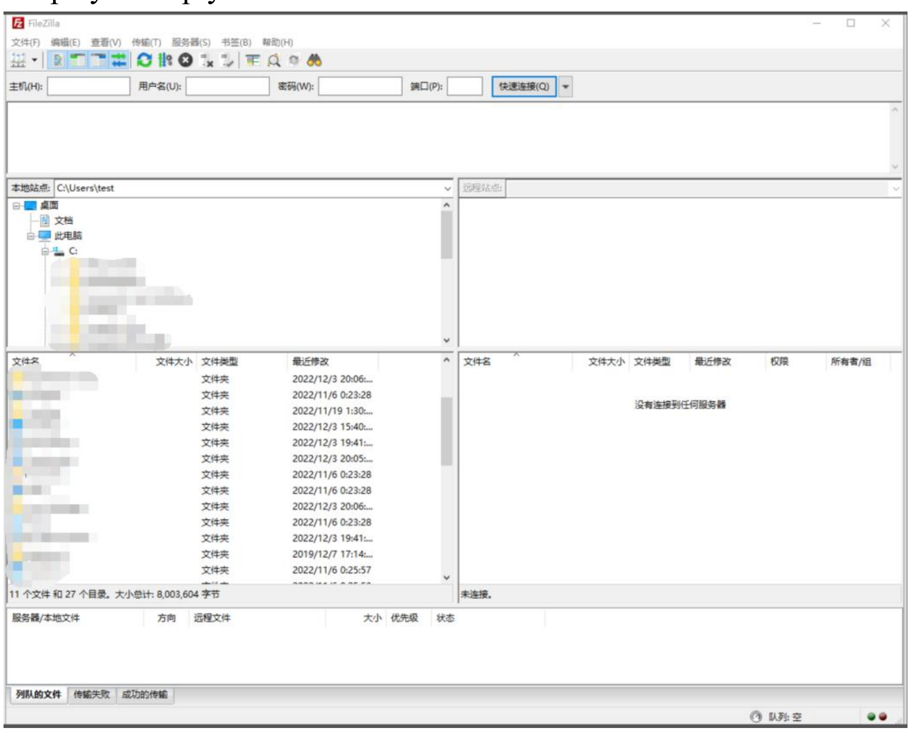
2) Download the installation package as shown below, and then double-click to install it directly

FileZilla_Server_1.5.1_win64-setup.exe

During the installation process, please select **Decline** in the following installation screen, and then select **Next>**.

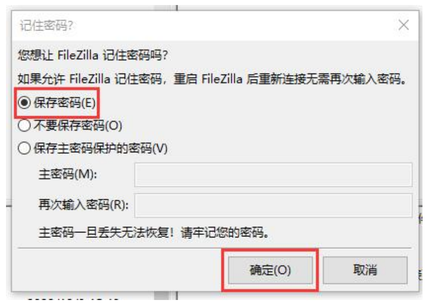


3) filezilla opens with the following interface, at this time the right side of the remote site below the display is empty

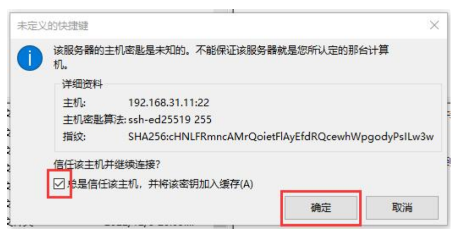


4) Connect the development board as shown below:

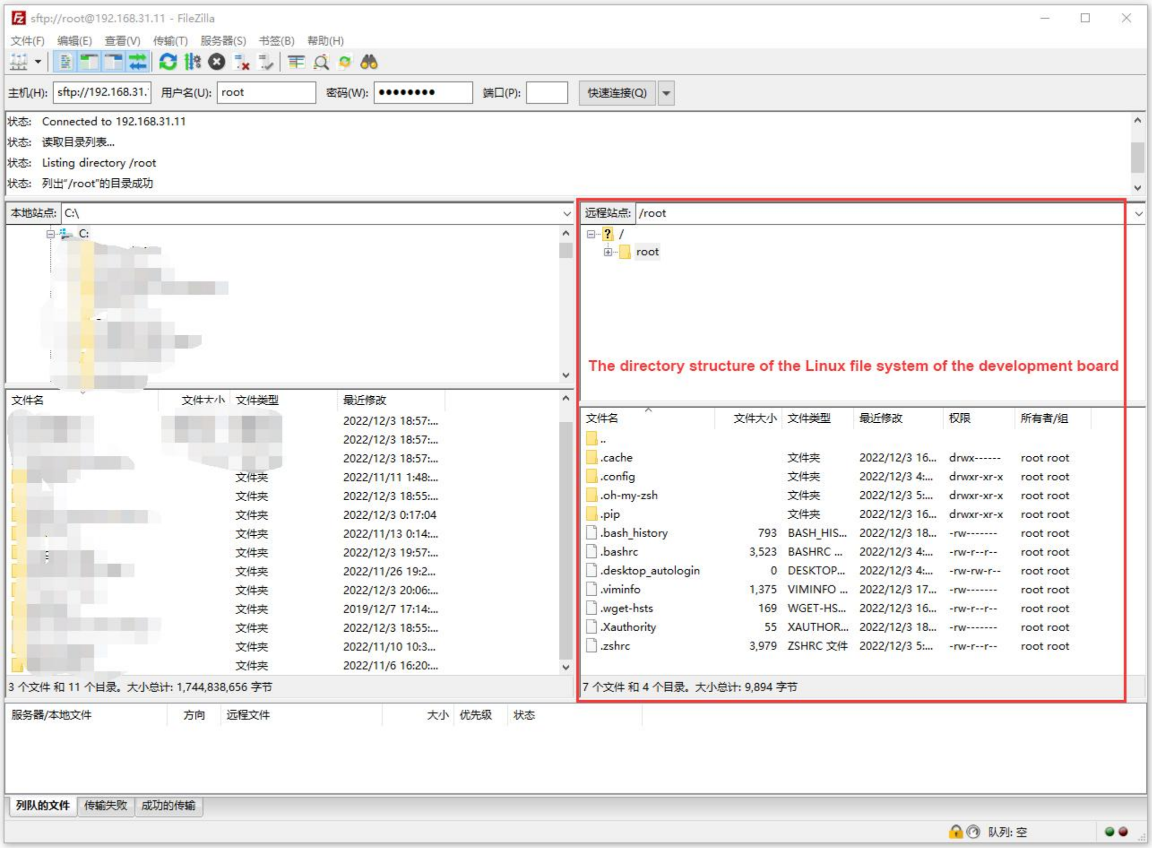
5) Then select **Save Password** and click **OK**.



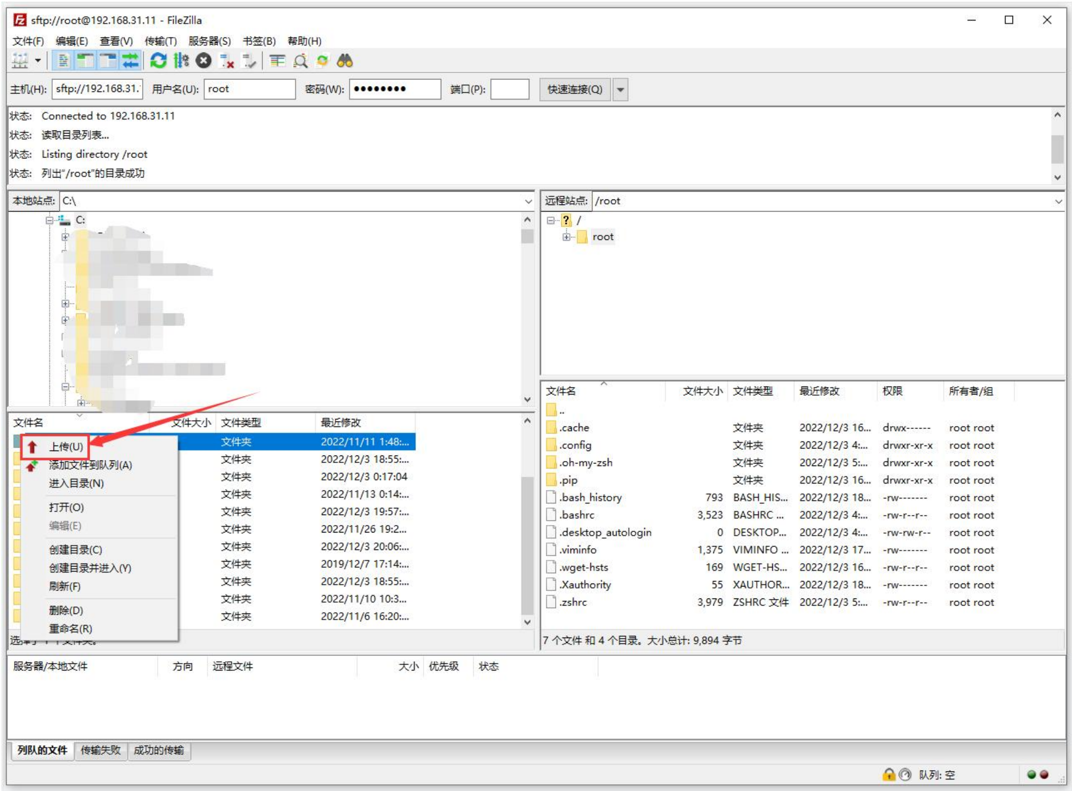
6) Then s**elect Always trust this host** and click **OK 确定**



7) After successful connection, you can see the directory structure of the linux file system of the development board on the right side of the filezilla software.



8) Then select the path to be uploaded to the development board on the right side of the filezilla software, then select the file to be uploaded in the Windows PC on the left side of the filezilla software, then click the right mouse button, and then click the upload option to start uploading files to the development board.



9) After the upload is complete, you can go to the corresponding path in the linux system of the development board to check the uploaded files.

10) The method of uploading a folder is the same as uploading a file, so I won't repeat it here.

3. 11. HDMI Testing

3. 11. 1. HDMI display test

1) Use HDMI to HDMI cable to connect Orange Pi development board and HDMI monitor.

2) If there is image output from the HDMI monitor after booting the linux system, it means the HDMI interface is working properly.

Note that many laptops have HDMI ports, but the laptop's HDMI port is generally only output function, there is no HDMI in function, that is to say, it can not be other devices HDMI output display to the laptop's screen.

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

When the HDMI is not displayed, please check the HDMI cable is not plugged tightly, after confirming that the wiring is not a problem, you can change a different screen to try to have a display.

3. 11. 2. HDMI to VGA display test

1) First you need to prepare the following accessories

a. HDMI to VGA converter

b. A VGA cable

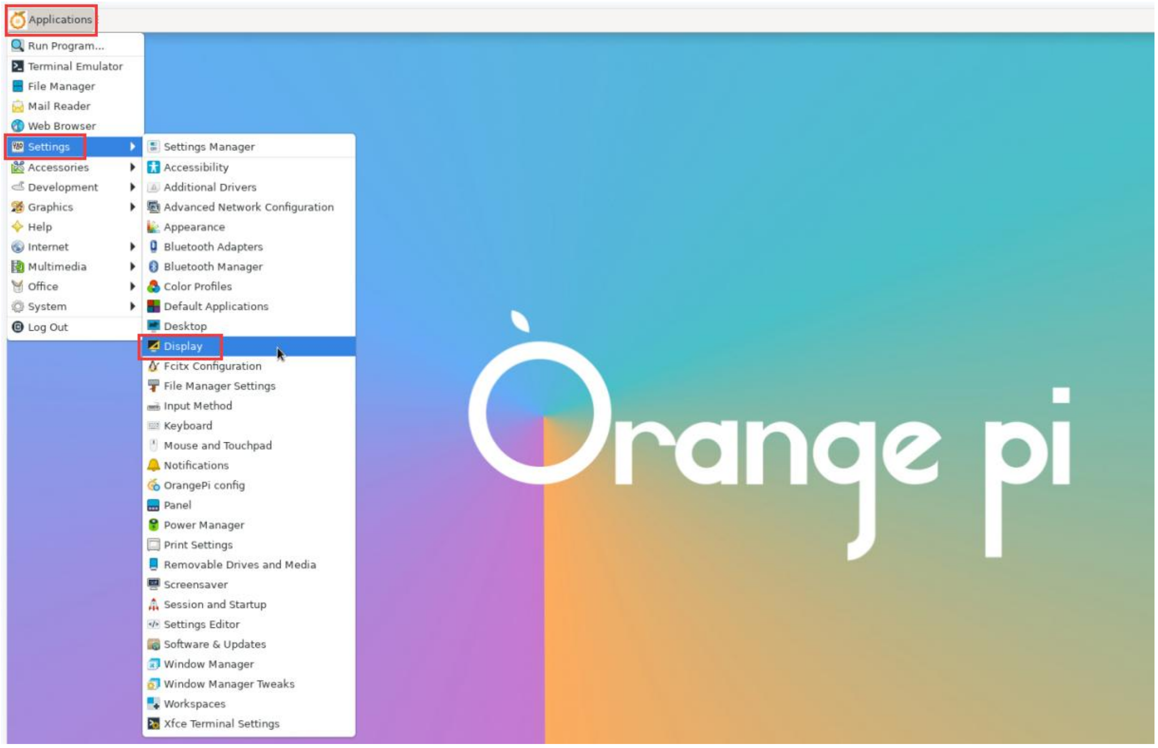
c. A monitor or television that supports a VGA interface.

2) HDMI to VGA display test as follows

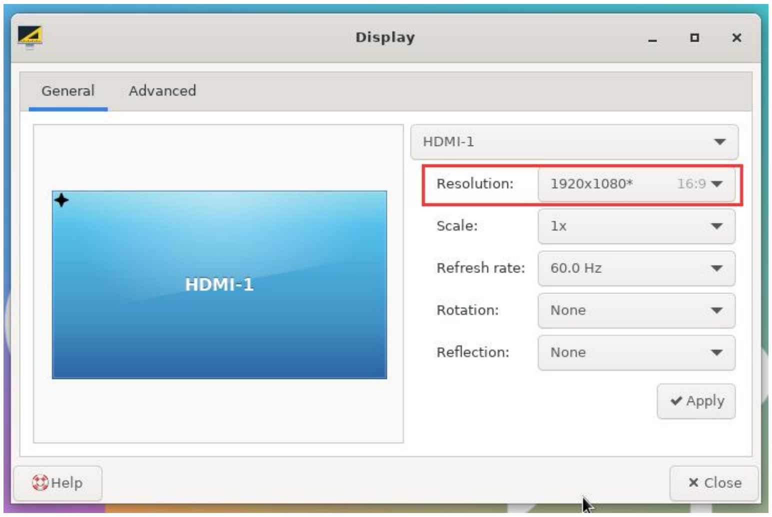
When using HDMI to VGA display, the development board as well as the Linux system of the development board do not need to do any setup, only the HDMI interface of the development board can display normally. So if there is a problem with the test, please check if there is a problem with the HDMI to VGA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

3. 11. 3. HDMI resolution setting method

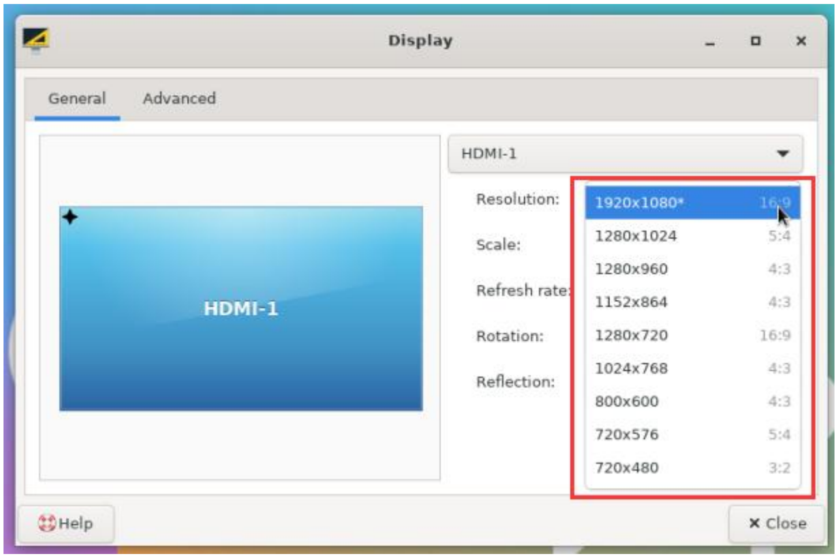
1) First, open **Display** in **Settings**.



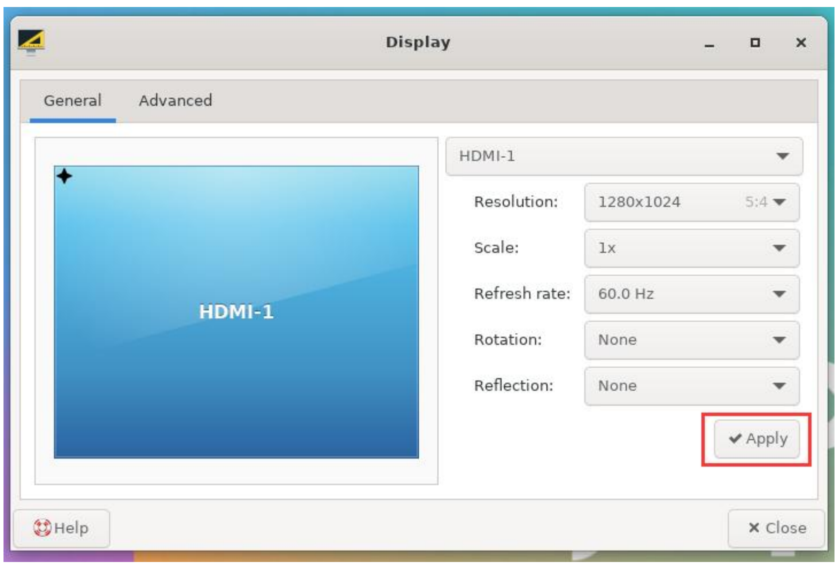
2) Then you can see the current resolution of the system



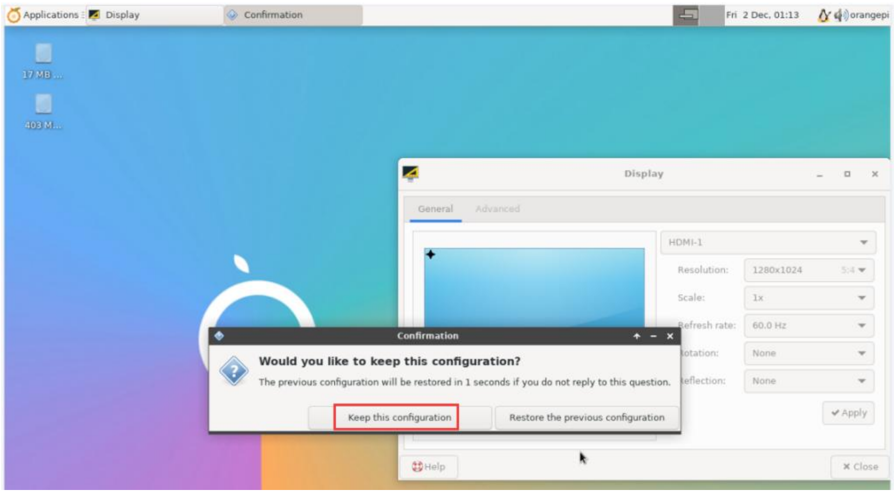
3) Click on the Resolution drop-down box to see all the resolutions currently supported by your monitor.



4) Then select the resolution you want to set and click Apply.



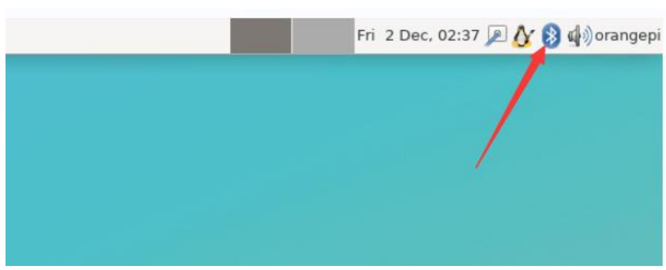
5) Wait for the new resolution to be set and then select **Keep the configuration**



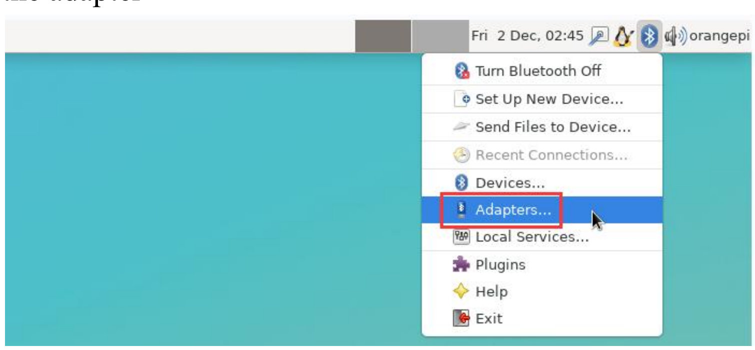
3. 12. Bluetooth usage

3. 12. 1. Test methods for desktop images

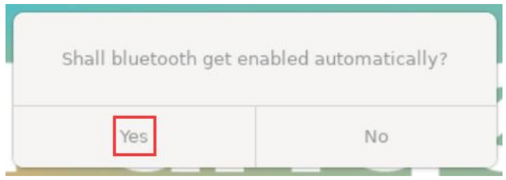
1) Click the Bluetooth icon on the top right corner of your desktop.



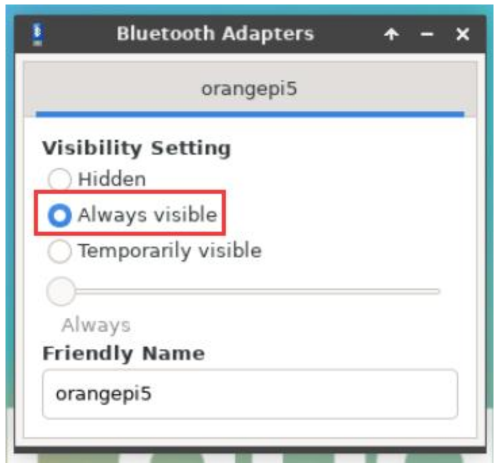
2) Then select the adapter



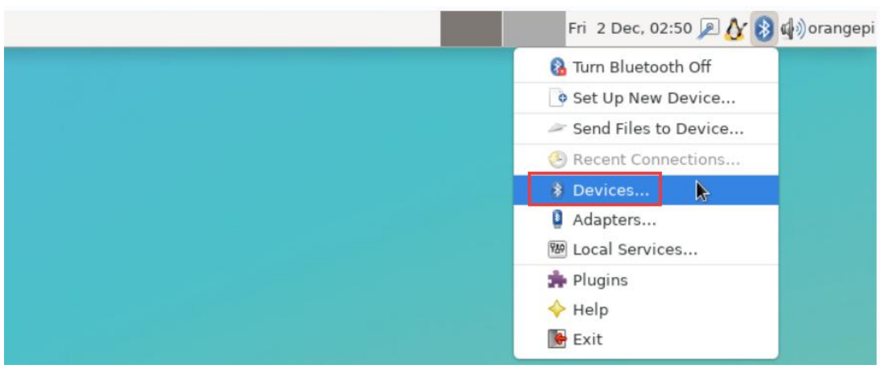
3) Select **Yes** if prompted by the following screen



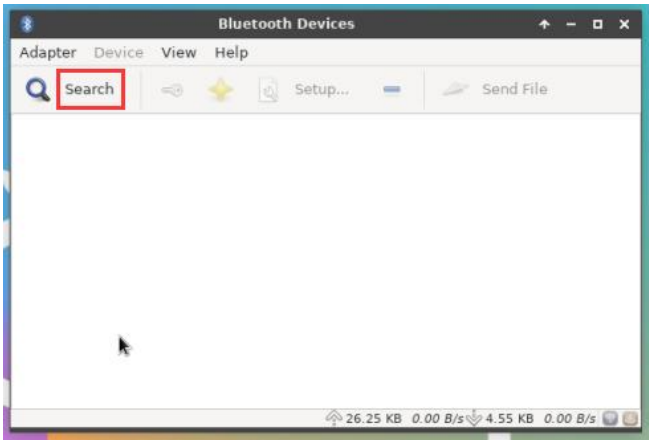
4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter settings interface, and then turn it off.



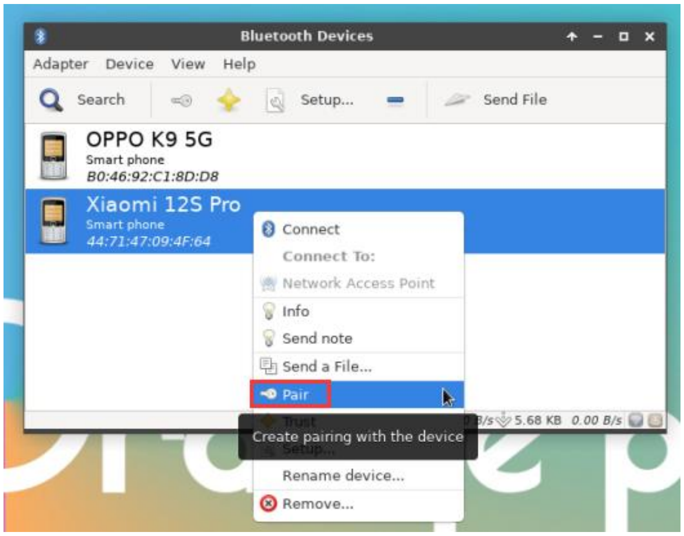
5) Then open the configuration screen of the Bluetooth device



6) Click **Search** to start scanning for Bluetooth devices around you.

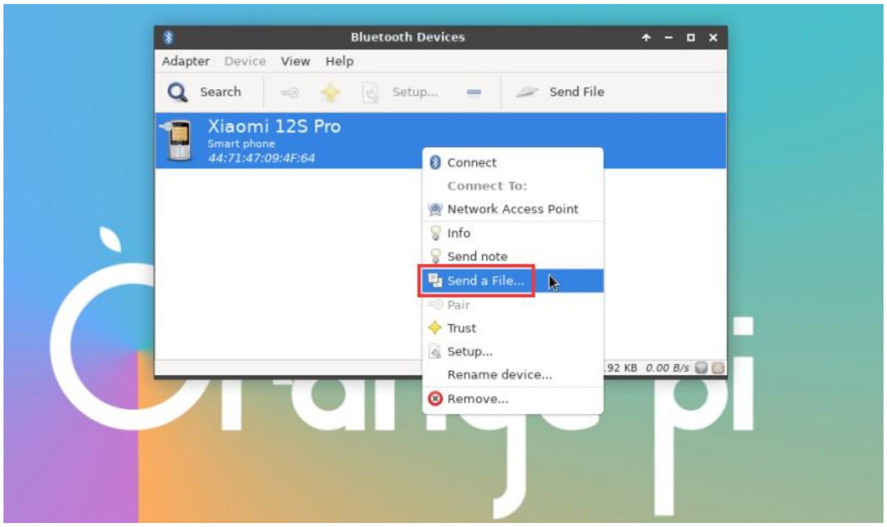


6) Then select the Bluetooth device you want to connect, and then click the right mouse button will pop up the Bluetooth device interface, select **Pair** to start pairing, here is the demonstration of the pairing of android cell phones

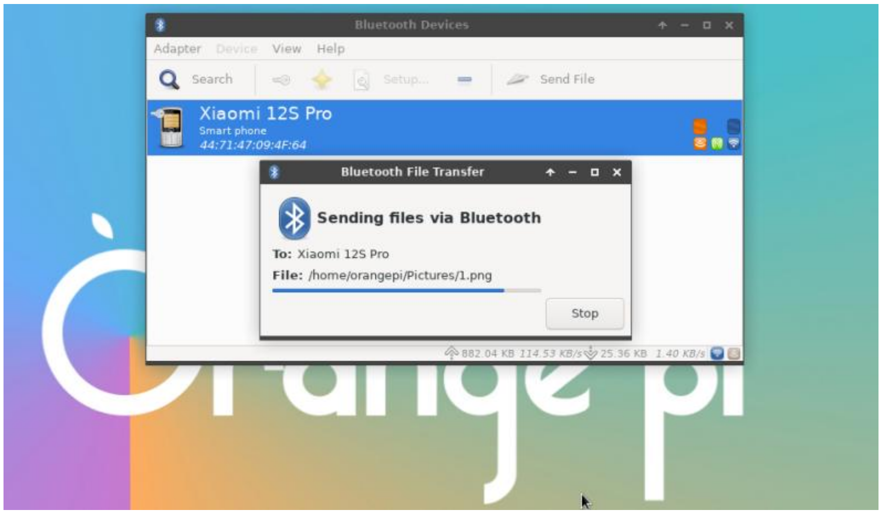


7) When pairing, the top right corner of the desktop will pop up the pairing confirmation box, select **Confirm** to confirm, at this time the same need to be confirmed on the phone

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



9) The interface for sending images is shown below



3. 13. USB interface testing

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

3. 13. 1. Connecting a USB mouse or keyboard test

1) Plug the USB keyboard into the USB port on the Orange Pi development board.

2) Connect the Orange Pi development board to an HDMI monitor.

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

3. 13. 2. Connecting a USB storage device test

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

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

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

3) Use the mount command to mount the USB flash drive to **/mnt**, and then you can view the files on the USB flash drive.

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

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

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

3. 13. 3. USB wireless card test

Currently tested USB wireless card can be used as follows, other models of USB wireless card, please test yourself, if you can not use it will need to port the corresponding USB wireless card driver

No.	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. 13. 3. 1. RTL8723BU test

1) First, insert the RTL8723BU wireless card module into the USB port 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 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 RTL8723BU module by 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
```

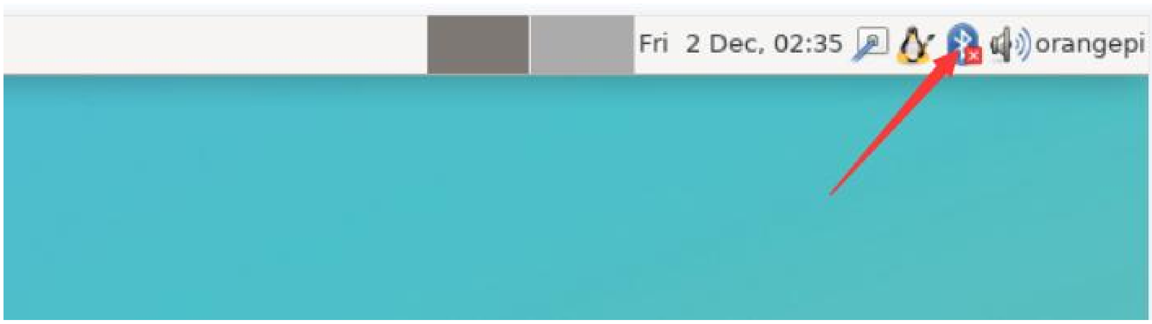
4) Then through the **sudo ifconfig** command you can see the RTL8723BU WIFI device node, WIFI connection and test method please refer to the WIFI connection test section, will not repeat here.

```
orangepi@orangepi:~$ sudo ifconfig wlx0013eff458ae
wlx0013eff458ae: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:13:ef:f4:58:ae  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

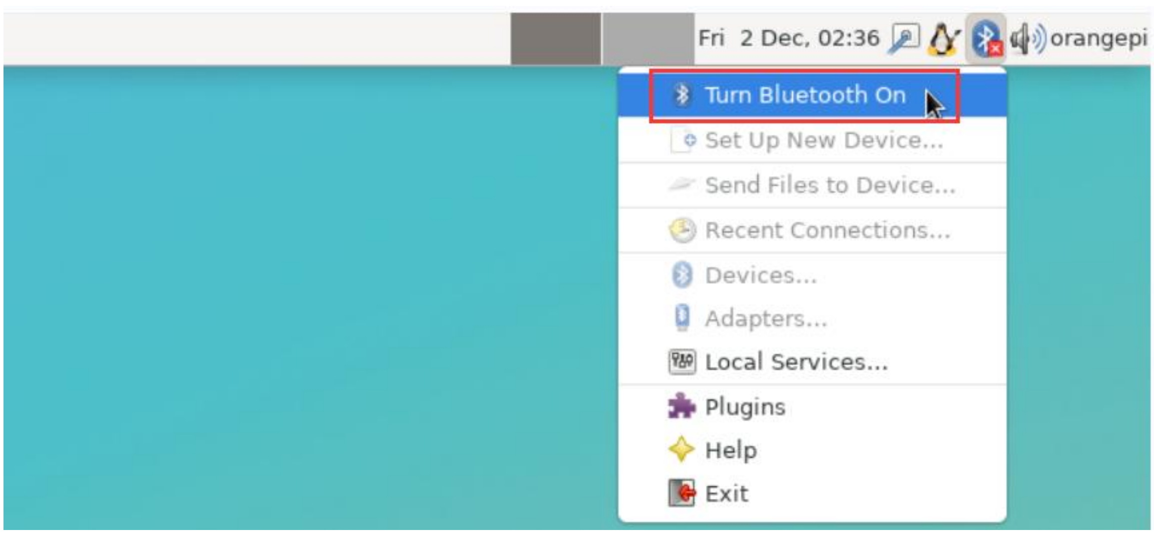
5) Then you can see the USB Bluetooth device through the **hciconfig** command.

```
orangepi@orangepi:~$ sudo apt update && sudo apt install bluez
orangepi@orangepi:~$ hciconfig
hci0:    Type: Primary   Bus: USB
         BD Address: 00:13:EF:F4:58:AE   ACL MTU: 820:8   SCO MTU: 255:16
         DOWN
         RX bytes:1252 acl:0 sco:0 events:125 errors:0
         TX bytes:23307 acl:0 sco:0 commands:125 errors:0
```

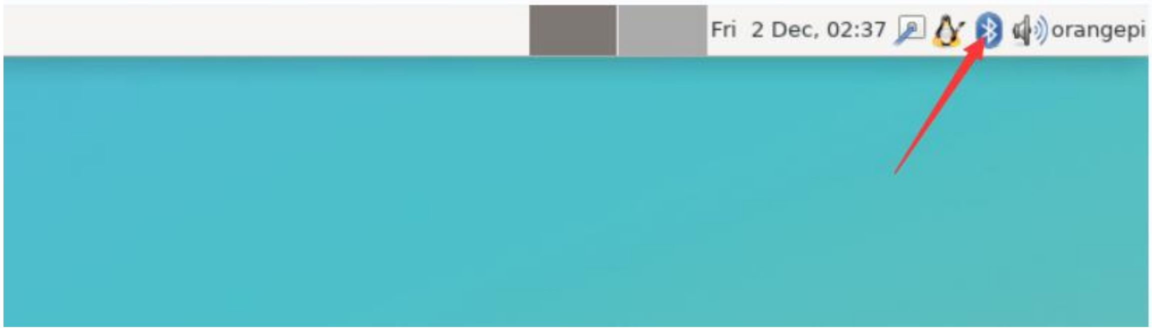
6) You can also see the Bluetooth icon on the desktop, at this time Bluetooth has not been turned on, so it will show a red **x**



7) Click **Turn Bluetooth On** to turn on Bluetooth.



8) The display after turning on Bluetooth is as follows



9) Please refer to **the section on how to use Bluetooth** for the Bluetooth test method, which will not be repeated here.

3. 13. 3. 2. RTL8811 test

1) First, insert the RTL8811 wireless card module into the USB port of the development board.

2) Then the linux system will automatically load the RTL8811 WIFI-related kernel modules, through the lsmod command you can see 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 by using 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 through the **sudo ifconfig** command you can see the WIFI device node, WIFI connection and test method please refer to **the WIFI connection test section**, will not repeat 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. 13. 3. 3. RTL8821CU Tests

1) First, insert the rtl8821cu wireless card module into the usb interface of the development board.

2) Then use the **lsusb** command to see the device information of the rtl8821cu usb wifi module, please make sure 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 lsusb command is in Driver CDROM Mode, please unplug the USB WIFI module again. If it doesn't work, please manually execute the following command to switch the mode:

orangepi@orangepi:~$ **sudo usb_modeswitch -KW -v 0bda -p 1a2b**

3) linux system will automatically load 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
```

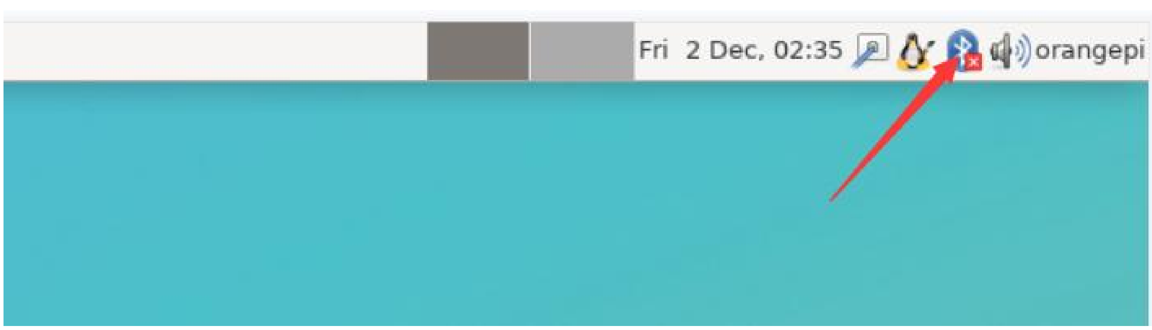
5) Then through the **sudo ifconfig** command you can see the rtl8821cu wifi device node, wifi connection and test method please refer to **the WIFI connection test section**, will not repeat here!

```
orangepi@orangepi:~$ sudo ifconfig wlx90de80521825
wlx90de80521825: flags=4099<UP,BROADCAST,MULTICAST>    mtu 1500
            ether 00:13:ef:f4:58:ae    txqueuelen 1000    (Ethernet)
            RX packets 0    bytes 0 (0.0 B)
            RX errors 0    dropped 0    overruns 0    frame 0
            TX packets 0    bytes 0 (0.0 B)
            TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
```

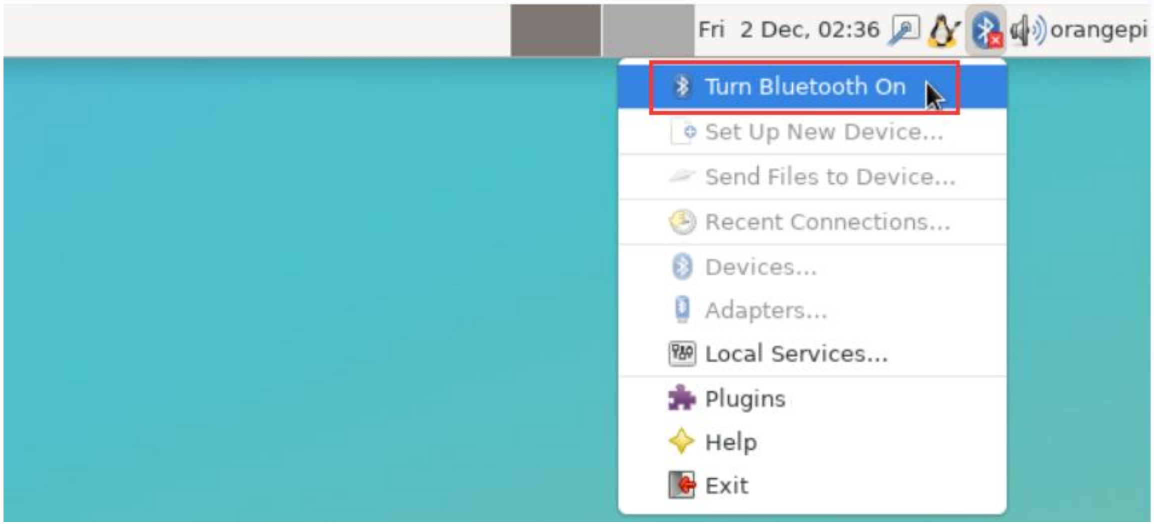
6) Then you can see the USB Bluetooth device through the **hciconfig**command.

```
orangepi@orangepi:~$ sudo apt-get update && sudo apt-get install -y bluez
orangepi@orangepi:~$ hciconfig
hci0:     Type: Primary    Bus: USB
            BD Address: 00:13:EF:F4:58:AE    ACL MTU: 820:8    SCO MTU: 255:16
            DOWN
            RX bytes:1252 acl:0 sco:0 events:125 errors:0
            TX bytes:23307 acl:0 sco:0 commands:125 errors:0
```

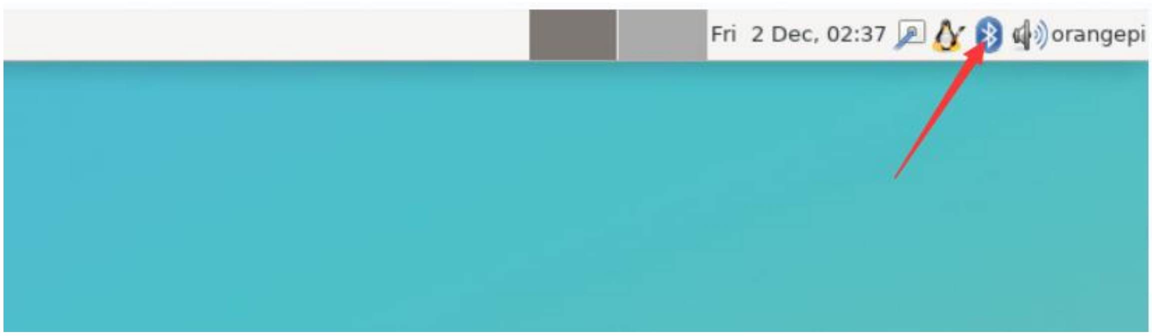
7) You can also see the Bluetooth icon on the desktop, at this time Bluetooth has not been turned on, so it will show a red **x**



8) Click **Turn Bluetooth On** to turn on Bluetooth.



9) The display after turning on Bluetooth is as follows



10) Please refer to **the section on Bluetooth usage for Bluetooth testing methods**, which will not be repeated here.

3. 13. 4. USB camera test

1) First of all, you need to prepare a USB camera as shown in the picture below or a similar USB camera that supports the UVC protocol, and then plug the USB camera into the USB port of the Orange Pi development board.

2) With the v4l2-ctl command, you can see that the device node information of the USB

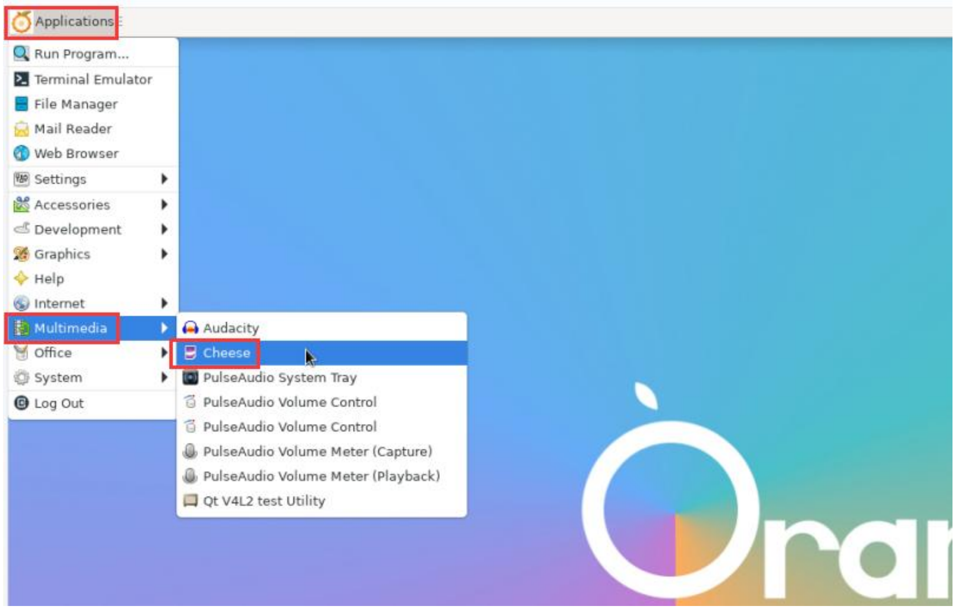
camera is/dev/video0.

```
orangepi@orangepi:~$ v4l2-ctl --list-devices
Q8 HD Webcam: Q8 HD Webcam (usb-fc880000.usb-1):
        /dev/video0
        /dev/video1
        /dev/media0
```

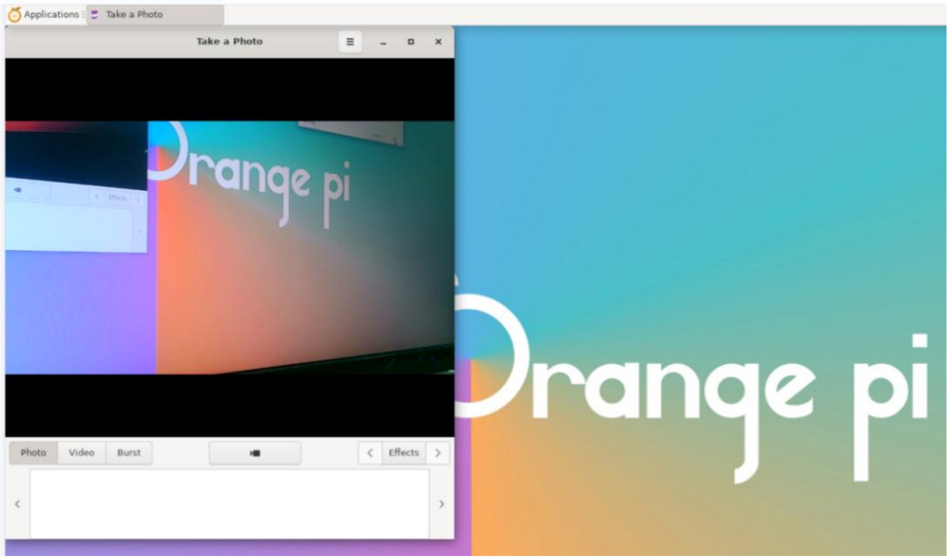
Note that the l in v4l2 is the lowercase letter l, not the number 1.

Also the serial number of video may not always be video0, please refer to what you actually see.

3) You can use Cheese to open the USB camera directly in the desktop system, the method of Cheese opening is shown in the figure below:



Cheese's interface after opening the USB camera is shown below:



4) Testing USB camera with fswebcam

a. Install fswebcam

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

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

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

b) The --no-banner option is used to remove the watermark from the photo.

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

d) The -S option is used to set the number of frames to be skipped.

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, you can use the scp command to transfer the pictures you have taken to an Ubuntu PC for mirroring after taking them.

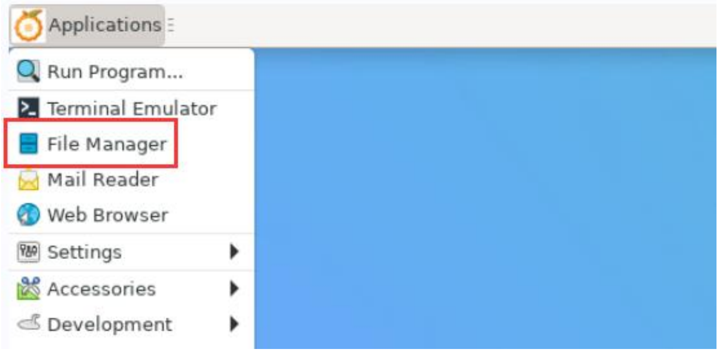
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test （Modify the IP address and path according to the actual situation）
```

d. In the desktop version of linux, it is possible to view the captured images directly on the HDMI monitor.

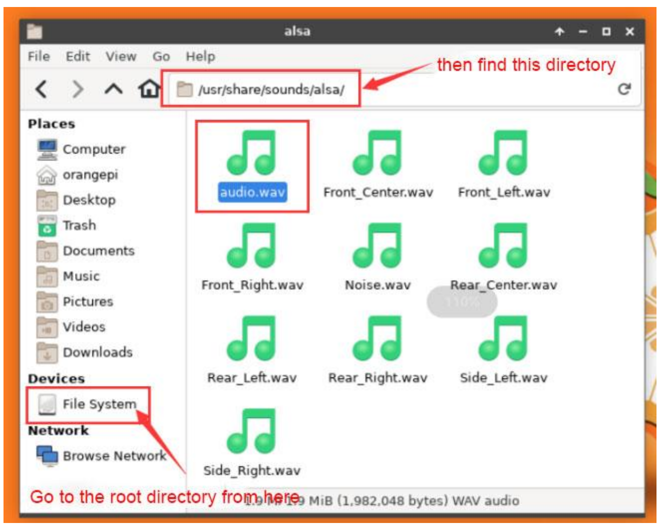
3. 14. Audio Test

3. 14. 1. Testing Audio Methods on Desktop Systems

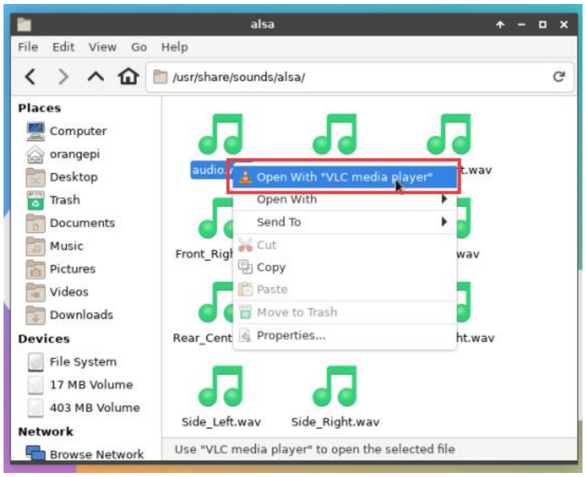
1) First open the file manager



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

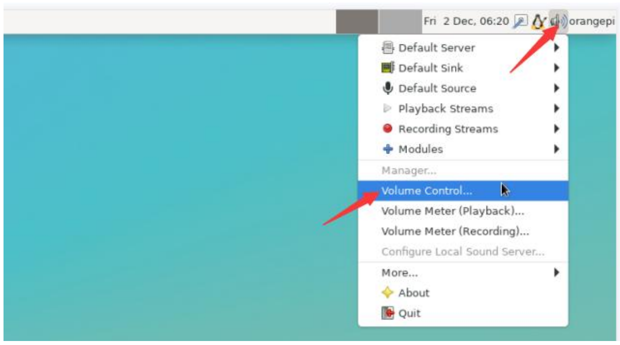


3) Then select the audio.wav file, right-click and choose to open with vlc to start playing

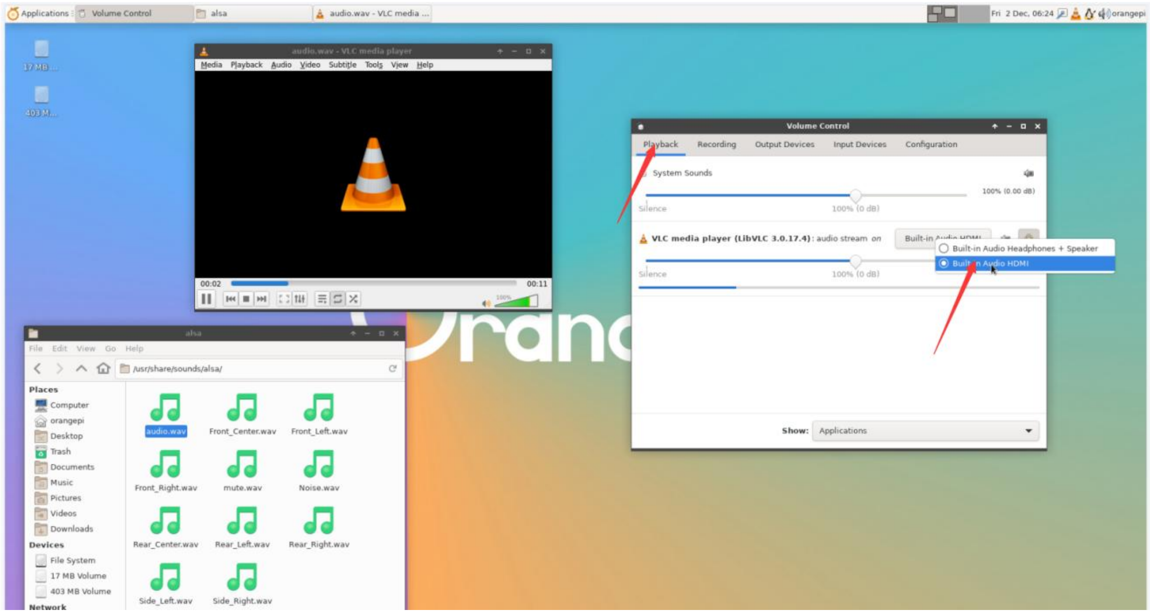


4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First open the volume control interface



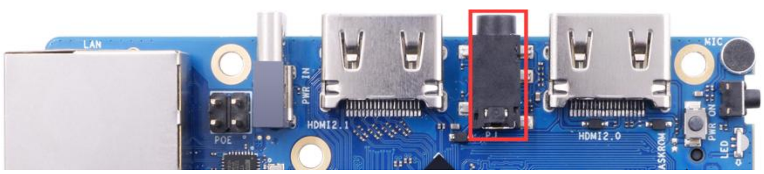
b. When playing audio, the audio device options that the playback software can use will be displayed in **Playback**, as shown in the figure below. Here you can set which audio device you want to play to



3. 14. 2. How to play audio using commands

3. 14. 2. 1. Headphone jack audio playback test

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



2) Then you can use the **aplay -l** command to view the sound card devices supported by the Linux system. From the output below, we can see that **card 2** is the es8388 sound card device, 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: rockchiphdmi1 [rockchip-hdmi1], device 0: rockchip-hdmi1 i2s-hifi-0 [rockchip-hdmi1 i2s-hifi-0]
  Subdevices: 0/1
  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 headphones can hear the sound, it means that the hardware can be used normally.

```
orangepi@orangepi:~$ aplay -D hw:2,0 /usr/share/sounds/alsa/audio.wav
Playing WAVE 'audio.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3. 14. 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, we can know that the sound card of HDMI TX1 is **card 0** and the sound card of HDMI TX2 is **card 1**.

```
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: rockchiphdmi1 [rockchip-hdmi1], device 0: rockchip-hdmi1 i2s-hifi-0 [rockchip-hdmi1 i2s-hifi-0]
  Subdevices: 0/1
  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 system's built-in audio files. If the HDMI monitor or TV can hear the sound, it indicates that the hardware is functioning properly

```
orangepi@orangepi:~$ aplay -D hw:1,0 /usr/share/sounds/alsa/audio.wav
```

3. 14. 3. How to test recording using commands

1) There is an onboard MIC on the development board, the location is as follows:

2) Run the **test_record.sh main** command to record an audio clip through the onboard MIC and then play it to the HDMI and headphones.

```
orangepi@orangepi:~$ test_record.sh main
Start recording: /tmp/test.wav
Recording WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Start playing
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3) In addition to the onboard MIC, we can also record audio through headphones with MIC function. After inserting the headphones with MIC function into the development board, running the **test_record.sh headset** command will record an audio through the headphones and then play it to HDMI and headphones.

```
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. 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)
```

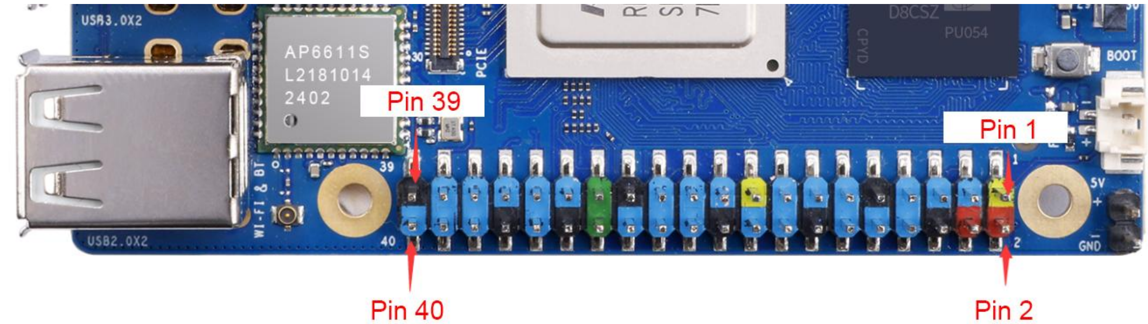
2) The command to check the current temperature of the nvme ssd solid state drive is:

```
orangepi@orangepi:~$ sudo smartctl -a /dev/nvme0 | grep "Temperature:"
Temperature:                        40 Celsius
```

3. 16. 40 Pin Interface Pin Description

1) Please refer to the following figure for the order of the 40 pin interface pins of the

Orange Pi 5 Max development board



2) The functions of the 40 pin interface pins of the Orange Pi 5 Max development board are shown in the following table

a. Below is the complete pin diagram of 40pin

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

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

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
			3.3V		1
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7
			GND		9
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	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
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27
		SPI1_CLK_M1	GPIO3_C1	113	29
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35
I2C5_SDA_M2			GPIO4_A7	135	37
			GND		39

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

引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
2		5V			
4		5V			
6		GND			
8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
12	134	GPIO4_A6			I2C5_SCL_M2
14		GND			
16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
18	36	GPIO1_A4			
20		GND			
22	40	GPIO1_B0			
24	44	GPIO1_B4	SPI0_CS0_M2		
26	45	GPIO1_B5	SPI0_CS1_M2		
28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
30		GND			
32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
34		GND			
36	63	GPIO1_D7		I2C8_SDA_M2	
38	112	GPIO3_C0	SPI1_MISO_M1		
40	111	GPIO3_B7	SPI1_MOSI_M1		

In the table above, the base addresses of the corresponding registers are marked for pwm, which is useful for checking which pwmchip in /sys/class/pwm/ corresponds to which pwm pin in the 40-pin 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 pre-installed in the Linux image released by Orange Pi. Unless the wiringOP code is updated, you do not need to download, compile and install it again. You can use it directly.

The storage path of the compiled wiringOP deb package in orangepi-build is:

orangepi-build/external/cache/debs/arm64/wiringpi_x.xx.deb

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

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A4 | 10  | 36   |
 |   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   |
 |   47 |  17 |    RXD.1 |     IN | 1 | 27 || 28 | 1 | IN     | TXD.1    | 18  | 46   |
 |  113 |  19 | GPIO3_C1 |     IN | 1 | 29 || 30 |   |        | GND      |     |      |
 |  109 |  20 |  CAN1_RX |     IN | 1 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |  110 |  22 |  CAN1_TX |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  114 |  23 | GPIO3_C2 |     IN | 1 | 35 || 36 | 1 | IN     | GPIO3_D7 | 24  | 63   |
 |  135 |  25 | GPIO4_A7 |     IN | 0 | 37 || 38 | 1 | IN     | GPIO3_C0 | 26  | 112  |
 |      |     |      GND |        |   | 39 || 40 | 1 | IN     | GPIO3_B7 | 27  | 111  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
root@orangepi5max:~#
```

1) Download wiringOP code

orangepi@orangepi:~$ **sudo apt update**

orangepi@orangepi:~$ **sudo apt install -y git**

orangepi@orangepi:~$ **git clone https://github.com/orangepi-xunlong/wiringOP.git -b next**

Note that Orange Pi 5 Max needs to download the wiringOP next branch code, please do not miss the -b next parameter.

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

2) Compile and install wiringOP

orangepi@orangepi:~$ **cd wiringOP**

orangepi@orangepi:~/wiringOP$ **sudo ./build clean**

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

3) The output of the test gpio readall command is as follows

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A4 | 10  | 36   |
 |   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   |
 |   47 |  17 |    RXD.1 |     IN | 1 | 27 || 28 | 1 | IN     | TXD.1    | 18  | 46   |
 |  113 |  19 | GPIO3_C1 |     IN | 1 | 29 || 30 |   |        | GND      |     |      |
 |  109 |  20 |  CAN1_RX |     IN | 1 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |  110 |  22 |  CAN1_TX |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  114 |  23 | GPIO3_C2 |     IN | 1 | 35 || 36 | 1 | IN     | GPIO3_D7 | 24  | 63   |
 |  135 |  25 | GPIO4_A7 |     IN | 0 | 37 || 38 | 1 | IN     | GPIO3_C0 | 26  | 112  |
 |      |     |      GND |        |   | 39 || 40 | 1 | IN     | GPIO3_B7 | 27  | 111  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
root@orangepi5max:~#
```

3. 18. 40pin interface GPIO, I2C, UART, SPI, CAN and PWM test

3. 18. 1. 40pin GPIO port test

The Linux system released by Orange Pi has a pre-installed blink_all_gpio program, which will set all 28 GPIO ports in the 40-pin to switch high and low levels continuously.

After running the blink_all_gpio program, when you use a multimeter to measure the voltage level of the GPIO port, you will find that the GPIO pin will switch between 0 and 3.3v. Using this program, we can test whether the GPIO port can work properly.

The way to run the blink_all_gpio program is as follows:

orangepi@orangepi5max:~$ **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 available in the 40 pins of the development board. The following uses pin 7, which corresponds to GPIO GPIO1_A7 and wPi number 2, as an example to demonstrate how to set the high and low levels of the GPIO port.

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
```

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

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

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

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

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

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |    OUT | 0 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 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 that the high level is set successfully.

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

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

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |    OUT | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
```

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

3. 18. 2. 40 How to set pull-up and pull-down resistors on GPIO pin

Note that the following 6 GPIO pins of Orange Pi 5 Max have external 3.3V pull-up, so setting them to pull down is invalid.

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A4 | 10  | 36   |
 |   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   |
 |   47 |  17 |    RXD.1 |     IN | 1 | 27 || 28 | 1 | IN     | TXD.1    | 18  | 46   |
 |  113 |  19 | GPIO3_C1 |     IN | 1 | 29 || 30 |   |        | GND      |     |      |
 |  109 |  20 |  CAN1_RX |     IN | 1 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |  110 |  22 |  CAN1_TX |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  114 |  23 | GPIO3_C2 |     IN | 1 | 35 || 36 | 1 | IN     | GPIO3_D7 | 24  | 63   |
 |  135 |  25 | GPIO4_A7 |     IN | 0 | 37 || 38 | 1 | IN     | GPIO3_C0 | 26  | 112  |
 |      |     |      GND |        |   | 39 || 40 | 1 | IN     | GPIO3_B7 | 27  | 111  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
root@orangepi5max:~#
```

1) Below, we take pin 11, which corresponds to GPIO GPIO1_A0 and wPi number 5, as an example to demonstrate how to set the pull-up and pull-down resistors of the GPIO port.

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |    OUT | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |    OUT | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |    OUT | 0 |  7 || 8  | 0 | OUT    | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 0 | OUT    | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |    OUT | 0 | 11 || 12 | 0 | OUT    | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |    OUT | 0 | 13 || 14 |   |        | GND      |     |      |
```

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

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 shown in the figure below, the available spis for Orange Pi 5 Max are spi0, spi1 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of SPI0, SPI1 and SPI4 in 40 pins are shown in the following table.

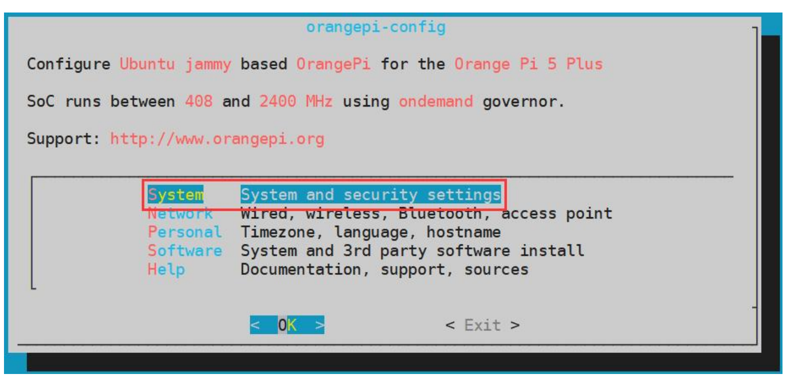
	SPI0_M2 corresponds to 40pin	**SPI1_M1 corresponds to 40pin**	**SPI4_M2 corresponds to 40pin**
MOSI	**Pin 19**	**Pin 40**	**Pin 13**
MISO	**Pin 21**	**Pin 38**	**Pin 11**
CLK	**Pin 23**	**Pin 29**	**Pin 15**
CS0	**Pin 24**	**Pin 35**	**Pin 16**
CS1	**Pin 26**	**none**	**none**
Dtbo Configuration	**spi0-m2-cs0-spidev** **spi0-m2-cs1-spidev** **spi0-m2-cs0-cs1-spidev**	**spi1-m1-cs0-spidev**	**spi4-m2-cs0-spidev**

3) In Linux system, the SPI in 40 pin is closed by default and needs to be opened manually. The detailed steps are as follows:

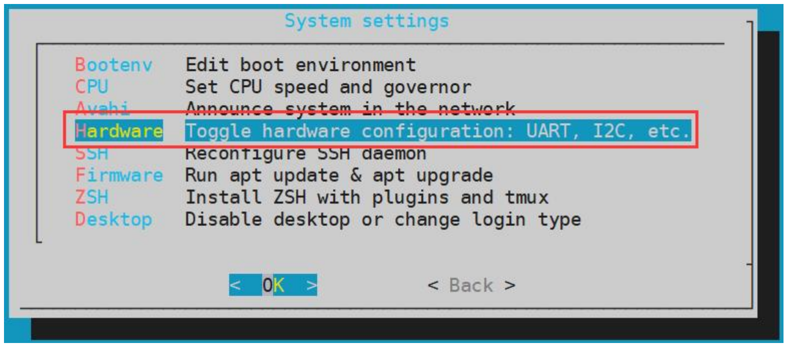
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

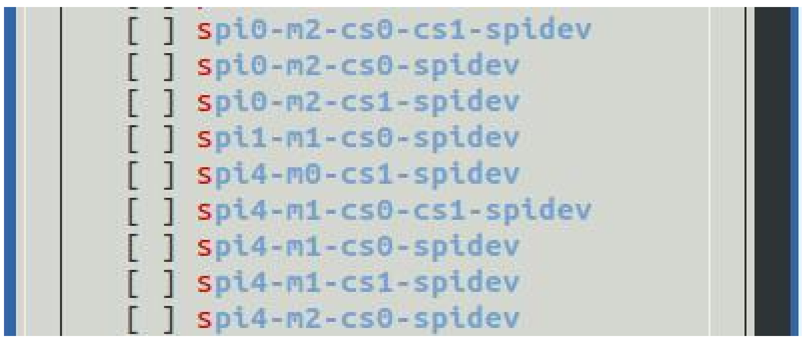
b. Then select **System**



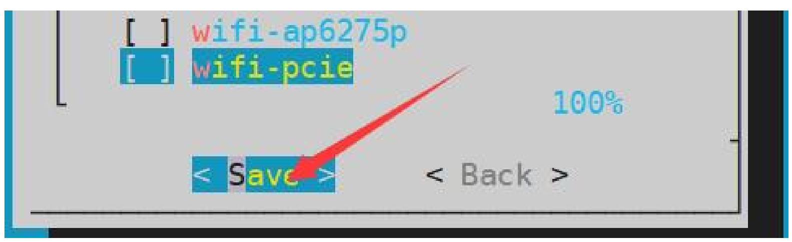
c. Then select **Hardware**



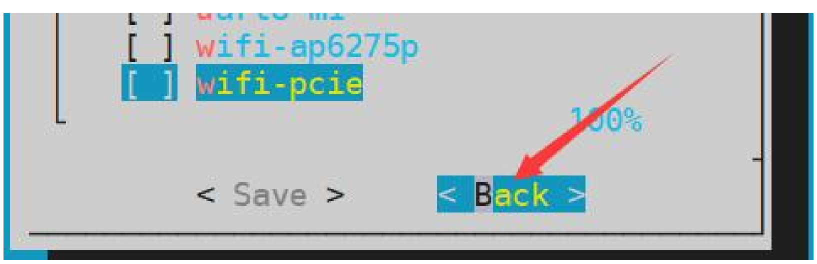
d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the spacebar to select the SPI configuration you want to open



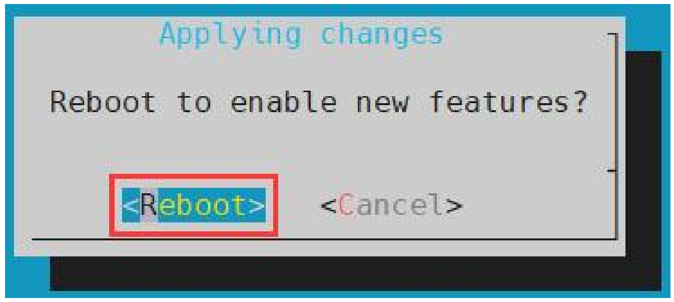
e. Then select **<Save>**



f. Then select **<Back>**



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



4) After restarting, enter the system and check whether there is a device node of **spidevx.x** in the Linux system. If it exists, it means that SPI has been set up and can be used directly.

```
orangepi@orangepi:~$ ls /dev/spidev*
/dev/spidev0.0    /dev/spidev0.1    /dev/spidev1.0    /dev/spidev4.0
```

The above is the result after opening spi0-m2-cs0-cs1-spidev, spi1-m1-cs0-spidev and spi4-m2-cs0-spidev.

5) Do not short the mosi and miso pins of SPI0, SPI1 or SPI4. The output of running spidev_test is as follows. You can see that the data of TX and RX are inconsistent.

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev0.0
or
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.0
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   | ................................
```

6) Then short the mosi and miso pins of SPI0 or SPI4 and run spidev_test again. 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/spidev1.0
or
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.0
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, Orange Pi 5 Max has four i2c buses: i2c2, 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of the 4 groups of I2C buses in 40pin are shown in the following table. I2C2_M0 and I2C2_M4, I2C5_M2 and I2C5_M3 can only use one of them at the same time, they cannot be used at the same time, they are all the same I2C, just connected to different pins, please do not think that they are two different I2C buses.

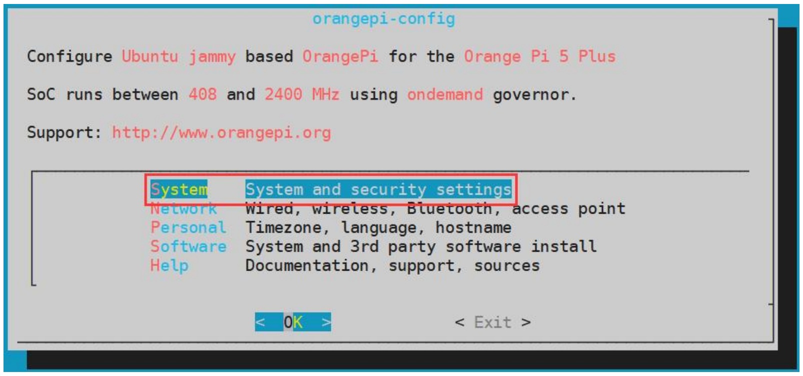
I2C Bus	SDA correspond 40pin	SCL corresponds to 40pin	dtbo corresponding configuration
I2C2_M0	Pin 3	Pin 5	i2c2-m0
I2C2_M4	Pin 11	Pin 13	i2c2-m4
I2C4_M3	Pin 15	Pin 16	i2c4-m3
I2C5_M2	Pin 37	Pin 12	i2c5-m2
I2C5_M3	Pin 27	Pin 28	i2c5-m3
I2C8_M2	Pin 36	Pin 32	i2c8-m2

3) In Linux system, the I2C bus in 40 pins is closed by default and needs to be opened manually before it can be used. The detailed steps are as follows:

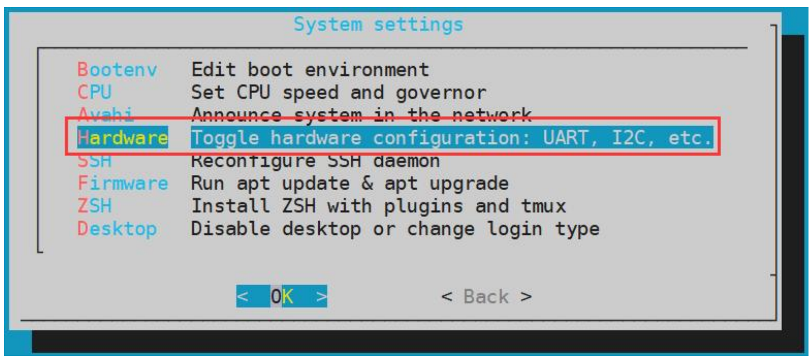
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

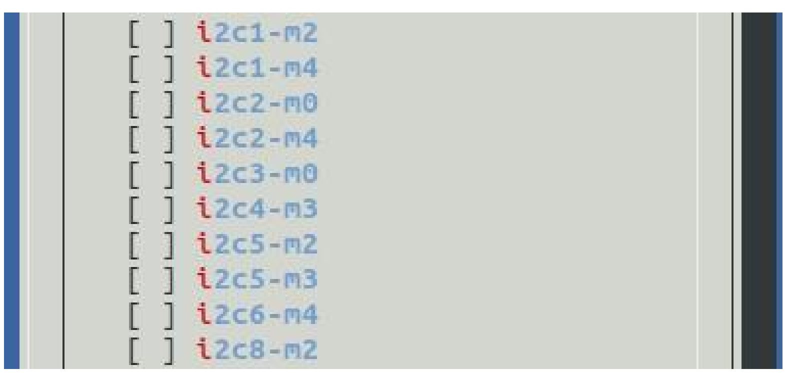
b. Then select **System**



c. Then select **Hardware**

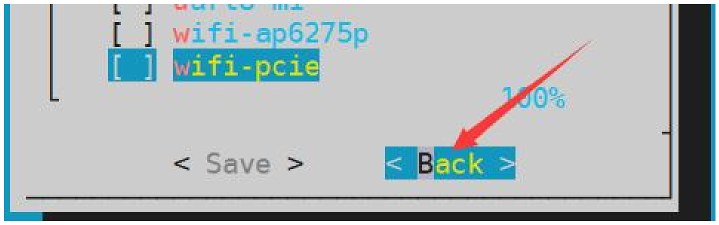


d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the spacebar to select the I2C configuration you want to open

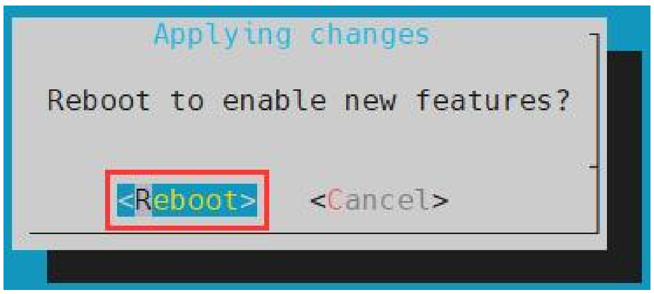


e. Then select **<Save>**

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 is a device node that needs 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 2      #i2c2 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
```

```
root@orangepi5max:~# i2cdetect -y 2
     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@orangepi5max:~#
```

3. 18. 5. 40pin UART test

1) As can be seen from the table below, Orange Pi 5 Max has four uart buses available: uart1, uart3, uart4 and uart6.

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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of the four UART buses in 40 pins are shown in the following table.

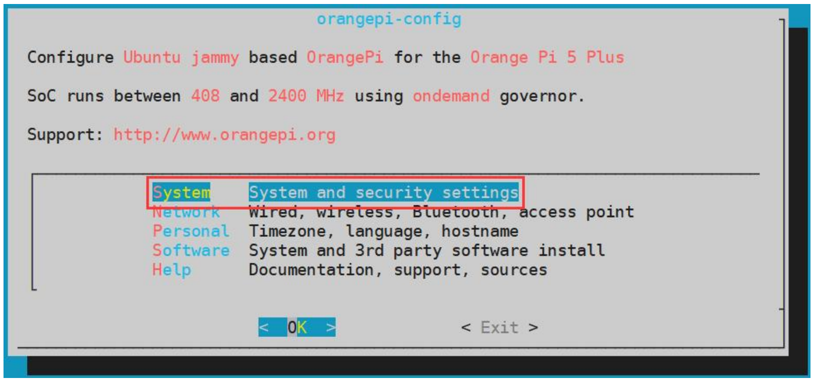
UART Bus	RX corresponds to 40pin	TX corresponds to 40pin	dtbo corresponding configuration
UART1_M1	**Pin 27**	**Pin 28**	**uart1-m1**
UART3_M1	**Pin 33**	**Pin 31**	**uart3-m1**
UART4_M2	**Pin 19**	**Pin 23**	**uart4-m2**
UART6_M1	**Pin 11**	**Pin 13**	**uart6-m1**

3) In Linux system, the UART in 40 pin is closed by default and needs to be opened manually before it can be used. The detailed steps are as follows:

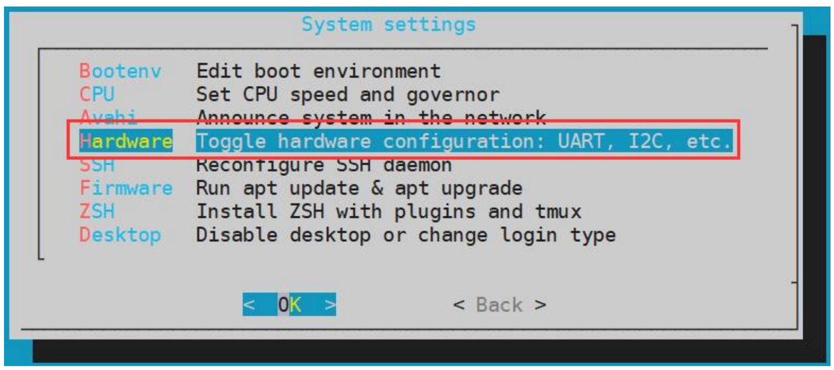
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

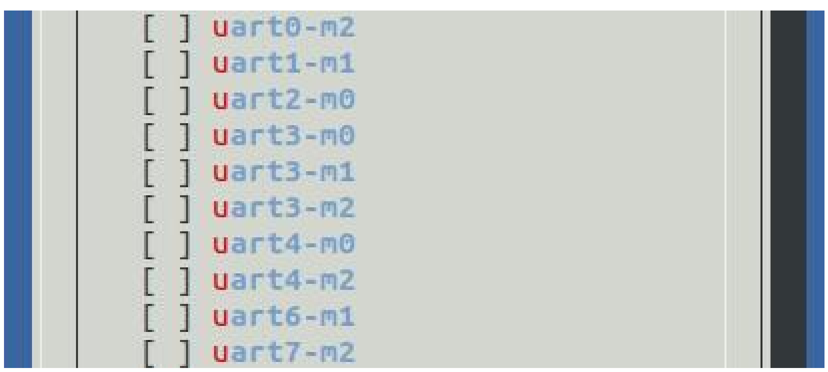
b. Then select **System**



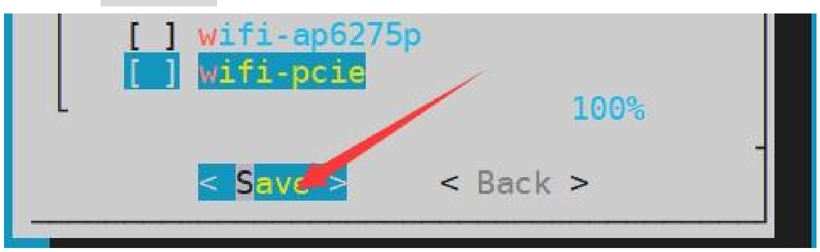
c. Then select **Hardware**



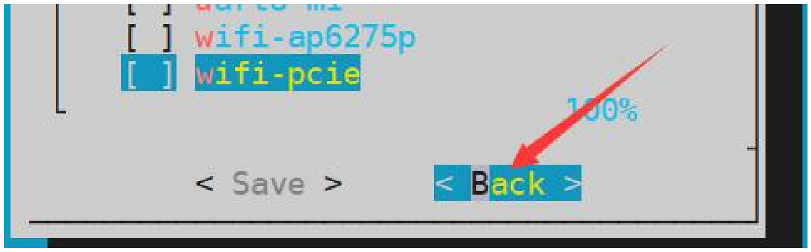
d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the spacebar to select the UART configuration you want to open



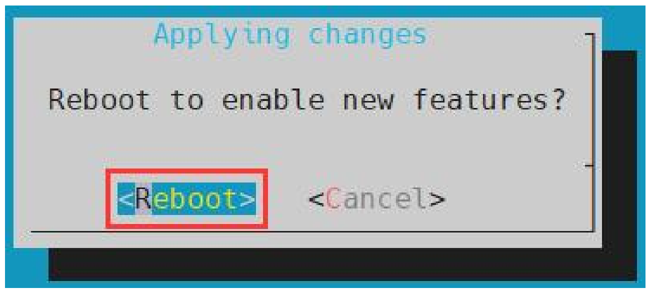
e. Then select **<Save>**



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 the Dupont line to short-circuit the rx and tx of the UART interface to be tested.

6) Use the **gpio serial** command to test the loopback function of the serial port as shown below. If you can see the following print, it means that the serial port communication is normal (ttySX needs to be replaced with the corresponding uart node name, 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, Orange Pi 5 Max has six PWM channels: pwm0, pwm1, pwm3, pwm12, pwm13 and pwm14

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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of PWM in 40pin are shown in the following table. Only one of PWM0_M0 and PWM0_M2, PWM1_M0 and PWM1_M2, PWM13_M0 and PWM13_M2, PWM14_M0 and PWM14_M2 can be used 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.

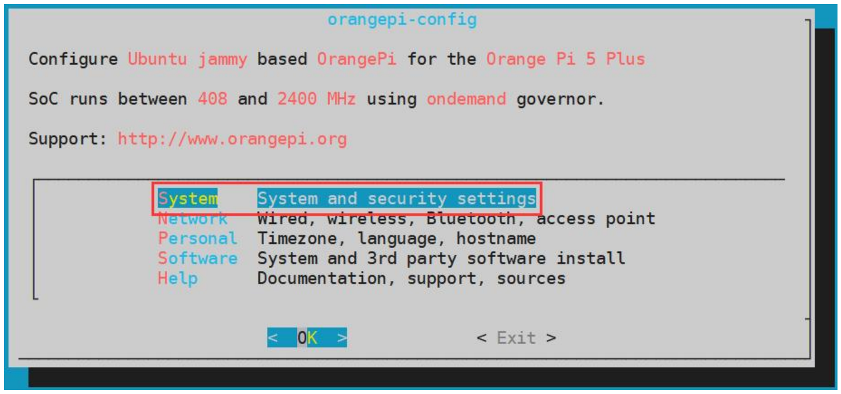
PWM Bus	Corresponding to 40pin	dtbo corresponding configuration
PWM0_M0	Pin 5	pwm0-m0
PWM0_M2	Pin 15	pwm0-m2
PWM1_M0	Pin 3	pwm1-m0
PWM1_M2	Pin 16	pwm1-m2
PWM3_M3	Pin 7	pwm3-m3
PWM12_M0	Pin 31	pwm12-m0
PWM13_M0	Pin 33	pwm13-m0
PWM13_M2	Pin 27	pwm13-m2
PWM14_M0	Pin 35	pwm14-m0
PWM14_M2	Pin 32	pwm14-m2

3) In Linux system, the PWM in 40 pins is turned off by default and needs to be turned on manually before it can be used. The detailed steps are as follows:

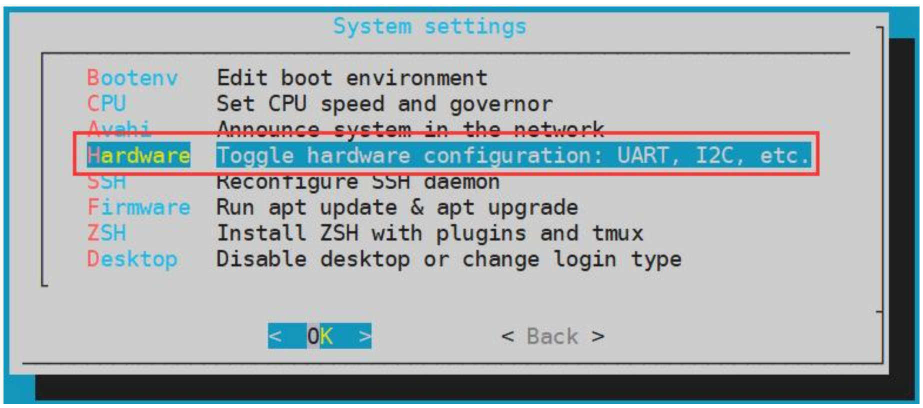
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

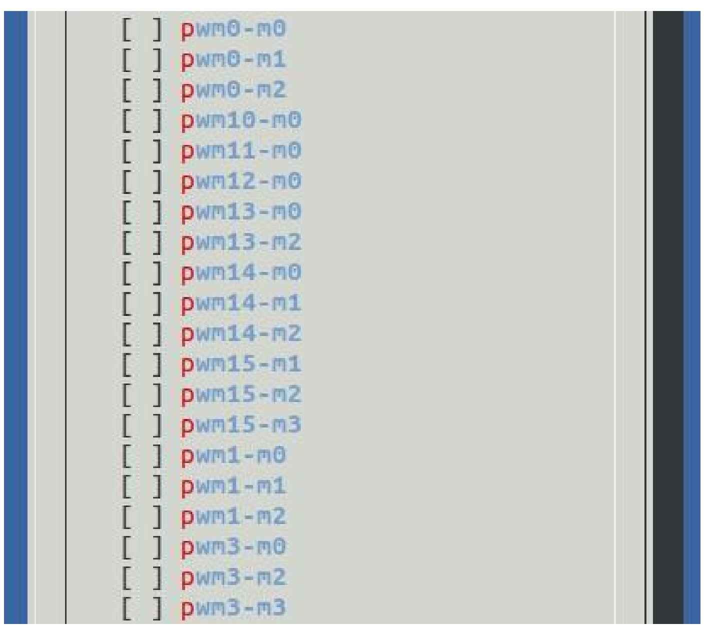
b. Then select **System**



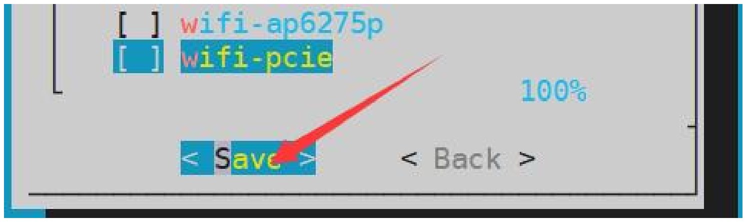
c. Then select **Hardware**



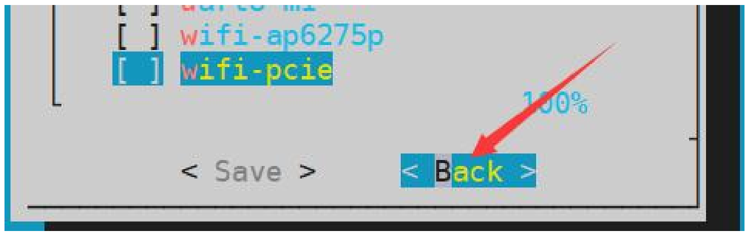
d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the **spacebar** to select the PWM configuration you want to turn on.



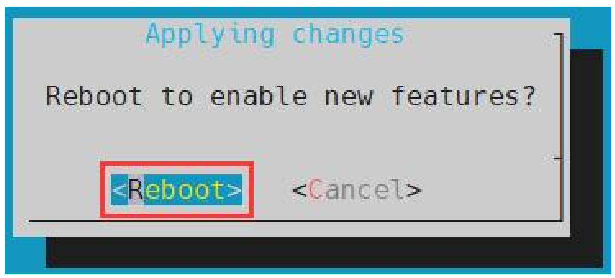
e. Then select **<Save>** to save



f. Then select **<Back>**



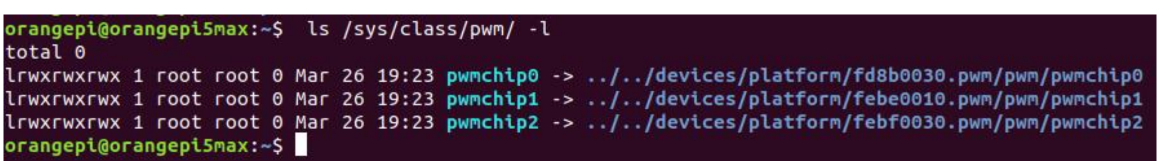
g. Then select **<Reboot>** to restart the system for the configuration to take effect.



4) When a pwm is turned on, there will be an additional pwmchipX in **/sys/class/pwm/** (X is a specific number). For example, after turning on pwm3, the pwmchipX under **/sys/class/pwm/** will change from two to three.

```
orangepi@orangepi:~$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

5) Which pwmchip above corresponds to pwm3? Let's first check the output of the **ls /sys/class/pwm/ -l** command, as shown below:



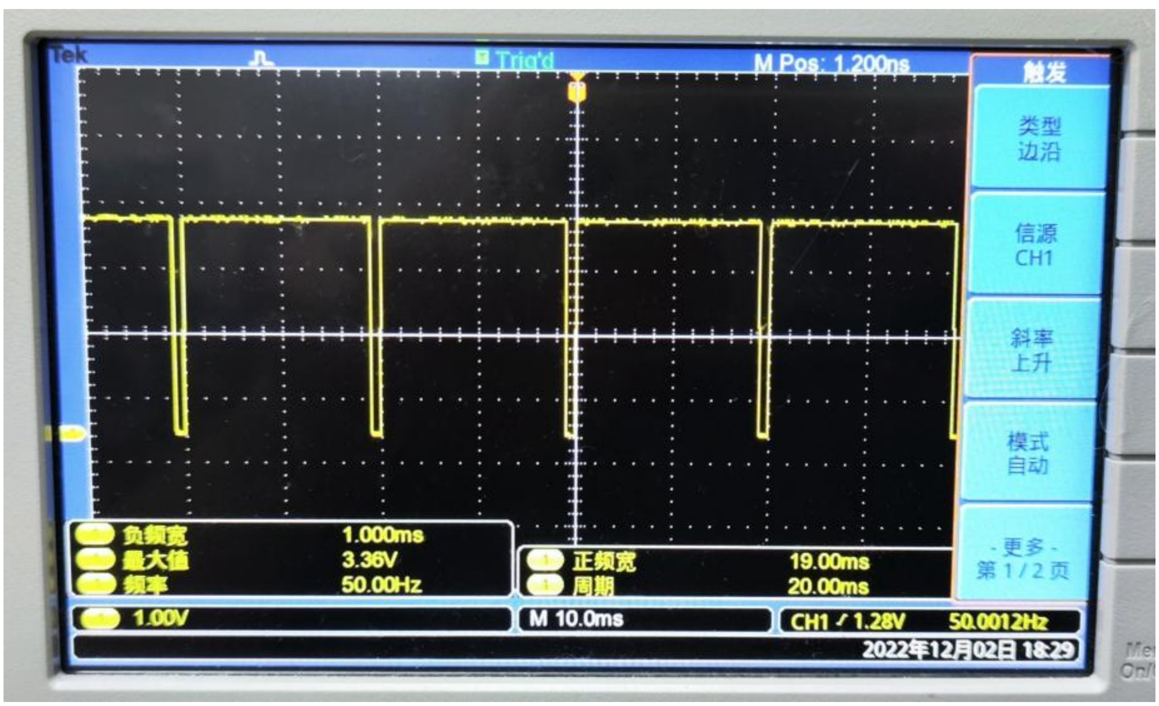
```
orangepi@orangepi5max:~$  ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Mar 26 19:23 pwmchip0 -> ../../devices/platform/fd8b0030.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Mar 26 19:23 pwmchip1 -> ../../devices/platform/febe0010.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Mar 26 19:23 pwmchip2 -> ../../devices/platform/febf0030.pwm/pwm/pwmchip2
orangepi@orangepi5max:~$
```

6) Then from the table below, we can see that the base address of the pwm3 register is fe8b0030. Looking at the output of the **ls /sys/class/pwm/ -l command**, we can see that pwmchip0 is linked to fe8b0030.pwm, so the pwmchip corresponding to pwm3 is pwmchip0

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
			3.3V		1
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7
			GND		9
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	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
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27
		SPI1_CLK_M1	GPIO3_C1	113	29
PWM12_M0 (febf0000)	CAN1_RX_M0	UART3_TX_M1	GPIO3_B5	109	31
PWM13_M0 (febf0010)	CAN1_TX_M0	UART3_RX_M1	GPIO3_B6	110	33
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35
I2C5_SDA_M2			GPIO4_A7	135	37
			GND		39

7) Then use the following command to make pwm3 output a 50Hz square wave (please switch to the root user first, then execute the following command)

```
root@orangepi:~# echo 0 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm0/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable
```



8) The pwm3 test method 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 available CAN buses for Orange Pi 5 Max are CAN0 and CAN1

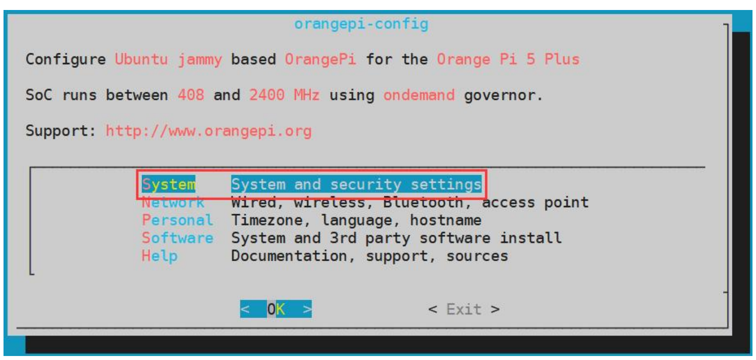
复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
			3.3V		1	2		5V			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) In the Linux system, the CAN in the 40 pin is closed by default and needs to be manually opened before it can be used. The detailed steps are as follows:

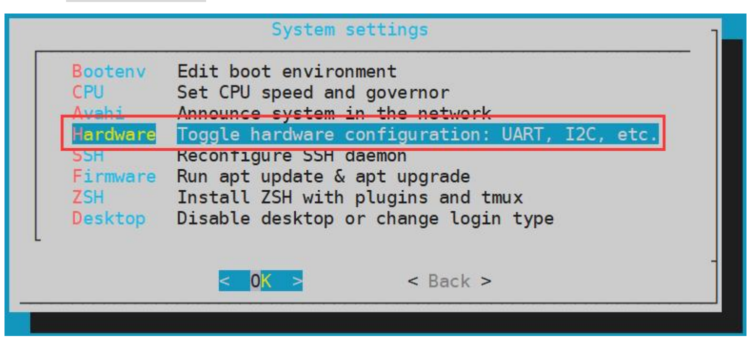
a. First run **orangepi-config**, ordinary users remember to add **sudo** permissions

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

b. Then select **System**

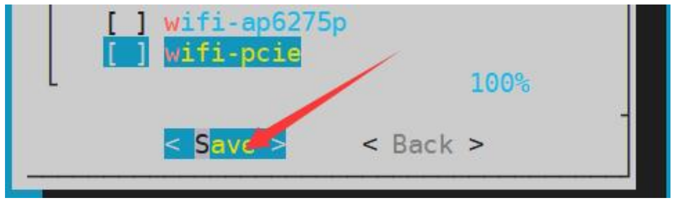


c. Then select **Hardware**

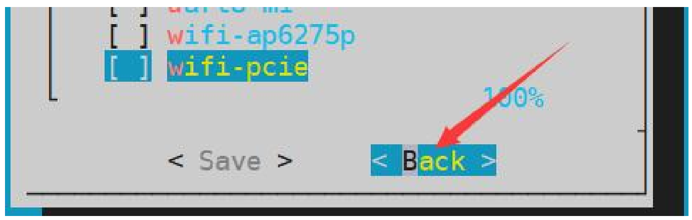


d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the **spacebar** to select the configuration you want to open

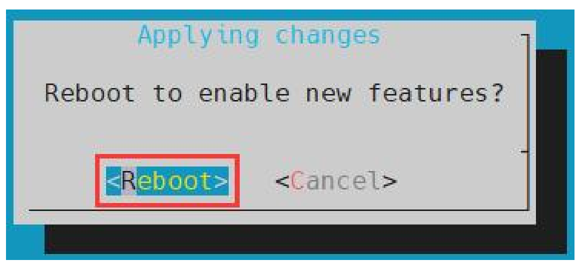
e. Then select **<Save>** to save



f. Then select **<Back>**



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



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 correctly opened.

```
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 pins corresponding to CAN0 and CAN1 are

	CAN0	CAN1
TX Pin	**Corresponding to pin 3**	**Corresponding to pin 33**
RX Pin	**Corresponding to pin 5**	**Corresponding to pin 31**

3.18.7.2. Test sending and receiving messages using CANalyst-II analyzer

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, you need to install the USBCANToolSetup software

4) The shortcut after USBCANToolSetup is installed is:

5) You also need to install the USB driver

6) The USB port of CANalyst-II analyzer needs to be connected to the USB port 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 of the development board
b. The GND pin of the CAN transceiver needs to be connected to the GND pin of the 40pin of the development board
c. The CAN TX pin of the CAN transceiver needs to be connected to the TX pin of

the CAN bus in the 40-pin of the 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 of the development board

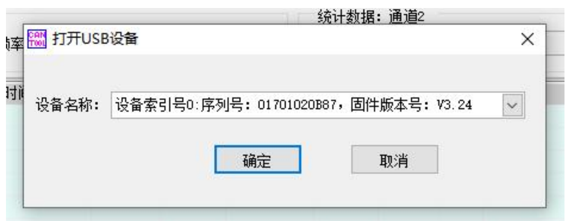
e. The CANH pin of the CAN transceiver needs to be connected to the H interface of the analyzer

f. The CANL pin of the CAN transceiver needs to be connected to the L interface of the analyzer

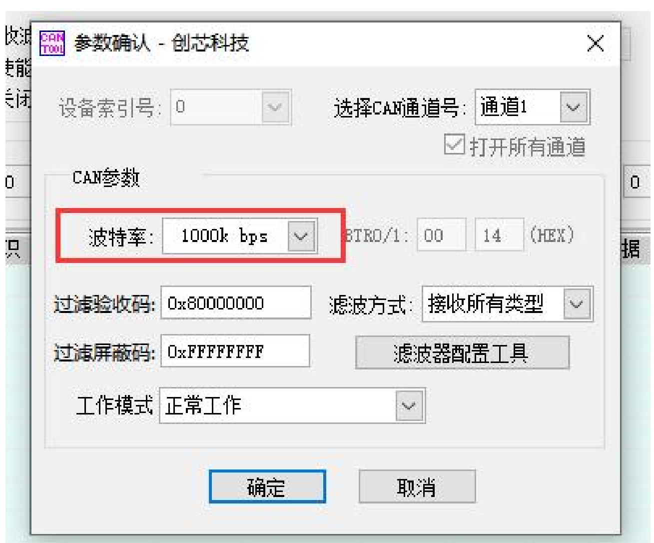
8) Then you can open the USB-CAN software

9) Then click Start Device

10) Then click OK



11) Set the baud rate to 1000k bps



12) After successfully 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 the message sent by the analyzer, it means that the CAN bus can be used normally.

```
orangepi@orangepi5max:~$ sudo candump can0
    can0    001     [8]    01 02 03 04 05 06 07 08
```

14) Development board sends CAN message test

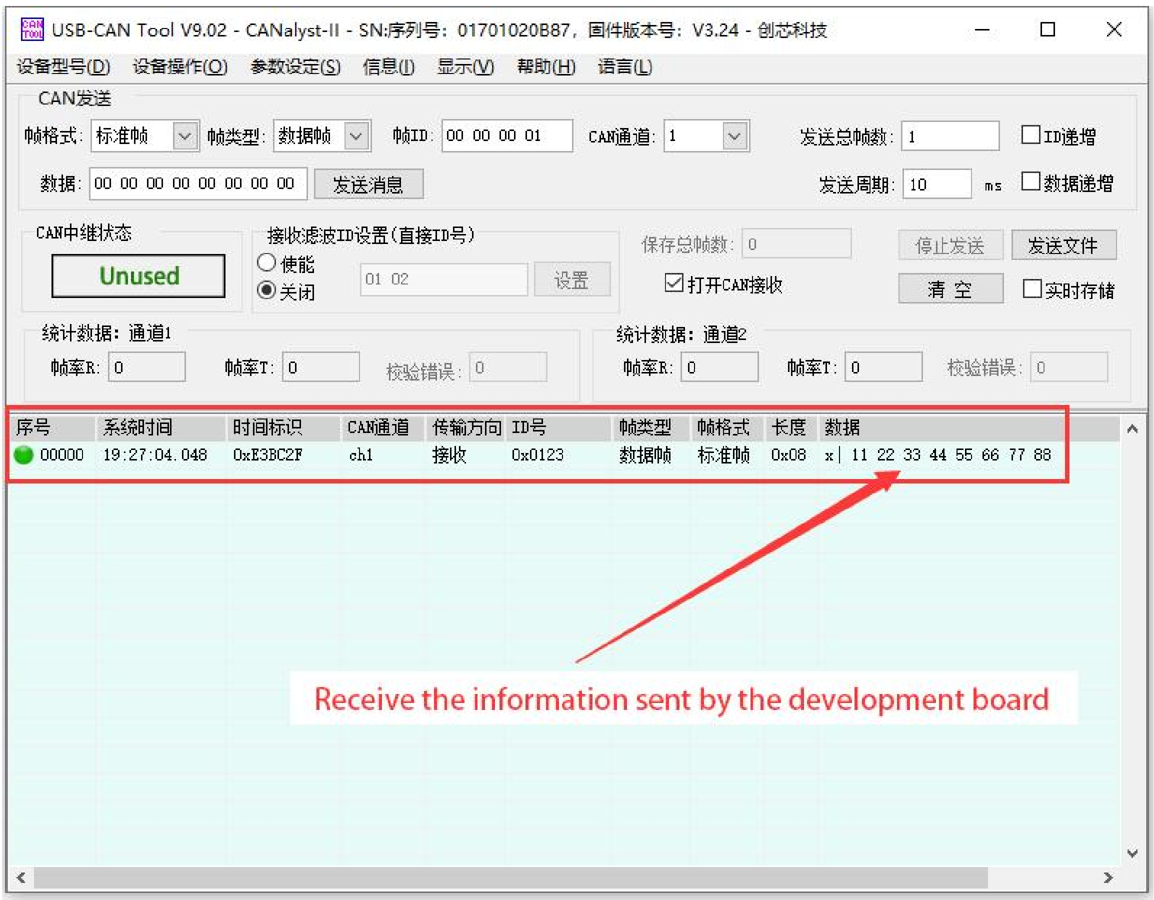
a. First, set the CAN baud rate to **1000kbps** in the Linux system

```
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 sent by the development board, it means the communication is successful.



3.19. wiringOP 硬件 PWM 的使用方法

> 使用 **wiringOP** 操作 **PWM** 前，请确保 **Linux** 系统已经安装了 **wiringOP**。如果 **gpio readall** 命令能正常使用，说明 **wiringOP** 已经安装了。如果提示找不到命令，请参考安装 **wiringOP** 的方法一小节的说明先安装下 **wiringOP**。

3. 19. 1. Method of setting PWM using the gpio command of wiringOP

3. 19. 1. 1. Set the corresponding pin to PWM mode

1) As shown in the table below, the development board can use a total of 6 PWM channels, including PWM0, PWM1, PWM3, PWM12, PWM13, and PWM14. PWM0.M0 and PWM0-M2, PWM1_M0 and PWM1_M2, PWM13_2M0 and PWM13_22, and PWM14_2M0 and PWM14_22 can only be used at the same time, and cannot be used simultaneously. They are all the same PWM, just 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

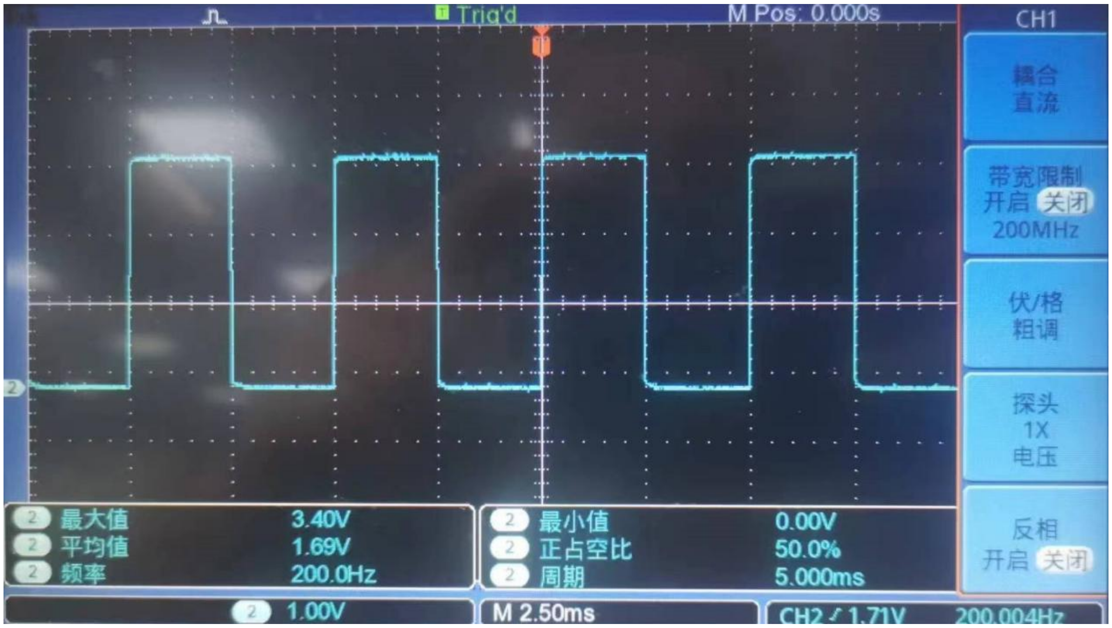
2) The wPi numbers corresponding to the PWM pins are as follows:

PWM Pin	wPi Serial number	Pin Serial number	GPIO Serial number
PWM0_M0	1	5	15
PWM0_M2	8	15	34
PWM1_M0	0	3	16
PWM1_M2	9	16	35
PWM3_M3	2	7	39
PWM12_M0	20	31	109
PWM13_M0	22	33	110
PWM13_M2	17	27	47
PWM14_M0	23	35	114
PWM14_M2	21	32	62

3) The command to set the pin to PWM mode is as follows, take PWM0_M0 as an example, where the third parameter needs to enter the wPi number corresponding to the PWM0_M0 pin.

```
orangepi@orangepi:~$ gpio mode 1 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_M0 pin to 4.

```
orangepi@orangepi:~$ gpio pwmc 1 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_M0 pin to 500Hz.

```
orangepi@orangepi:~$ gpio pwmTone 1 500
```

When setting the PWM frequency, it is necessary to ensure that:

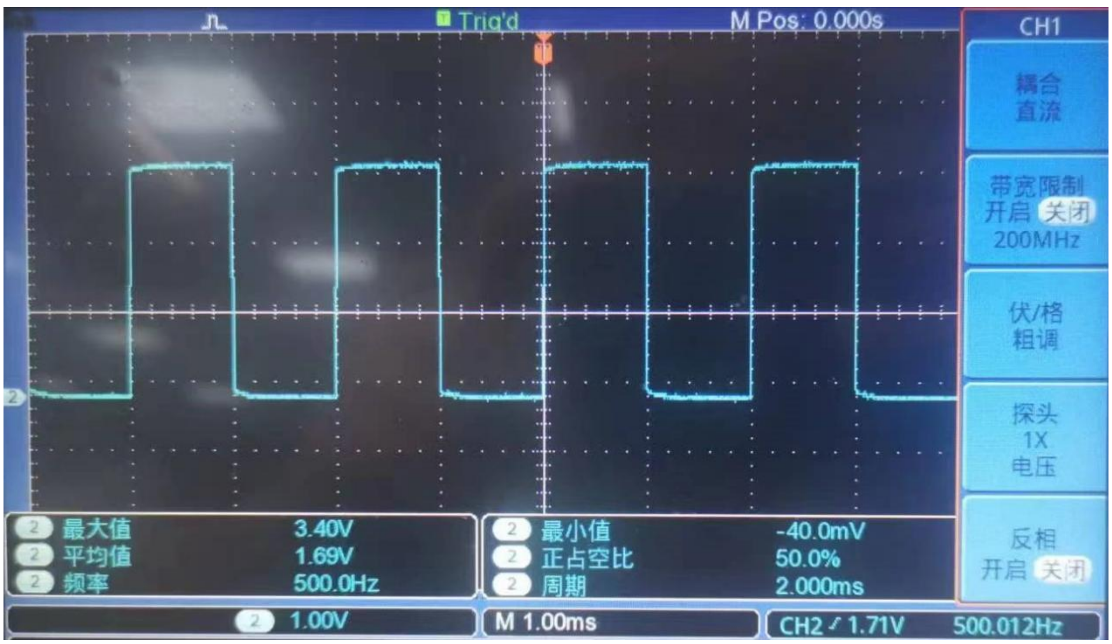
Set frequency value < 24000000 / (division factor * 2)。

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_M0 pin to 2000.

```
orangepi@orangepi:~$ gpio pwmr 1 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_M0 pin to 1000.

```
orangepi@orangepi:~$ gpio pwm 1 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_M0 pin:

```
orangepi@orangepi:/usr/src/wiringOP/examples$ sudo ./pwm 1
```

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. wiringOP-Python installation and usage method

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

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

3. 20. 1. Installation of wiringOP-Python

1) First install the dependency package

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python

Note that the following git clone --recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on wiringOP. Please make sure that there are no errors during the download process due to network problems.

If you have problems downloading the code from GitHub, you can directly use the wiringOP-Python source code that comes with the Linux image, which is stored in /usr/src/wiringOP-Python.

```
root@orangepi:~# git clone --recursive https://github.com/orangepi-xunlong/wiringOP-Python -b next
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# git submodule update --init --remote
```

3) Then compile wiringOP-Python and install it into the Linux system of the development board using the following command:

```
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# python3 generate-bindings.py > bindings.i
root@orangepi:~/wiringOP-Python# sudo python3 setup.py install
```

4) Then enter the following command. If help information is output, it means wiringOP-Python has been successfully installed. Press the **q** key to exit the help information interface.

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi;    help(wiringpi)"
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
```

```
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.
```

5) The steps to test whether wiringOP-Python is successfully installed in the python command line are as follows:

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

```
root@orangepi:~# python3
```

b. Then import the wiringpi python module

```
>>> import wiringpi;
```

c. Finally, enter the following command to view the help information of wiringOP-Python. Press the **q** key to exit the help information interface.

```
>>> help(wiringpi)
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

    class GPIO(builtins.object)
     |   GPIO(pinmode=0)
```

```
      |

>>>
```

3. 20. 2. 40pin GPIO port test

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

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A4 | 10  | 36   |
 |   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   |
 |   47 |  17 |    RXD.1 |     IN | 1 | 27 || 28 | 1 | IN     | TXD.1    | 18  | 46   |
 |  113 |  19 | GPIO3_C1 |     IN | 1 | 29 || 30 |   |        | GND      |     |      |
 |  109 |  20 |  CAN1_RX |     IN | 1 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |  110 |  22 |  CAN1_TX |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  114 |  23 | GPIO3_C2 |     IN | 1 | 35 || 36 | 1 | IN     | GPIO3_D7 | 24  | 63   |
 |  135 |  25 | GPIO4_A7 |     IN | 0 | 37 || 38 | 1 | IN     | GPIO3_C0 | 26  | 112  |
 |      |     |      GND |        |   | 39 || 40 | 1 | IN     | GPIO3_B7 | 27  | 111  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
root@orangepi5max:~#
```

1) Below, we take pin 7, which corresponds to GPIO GPIO1_A7 and wPi number 2, as an example to demonstrate how to set the high and low levels of the GPIO port.

```
root@orangepi5max:~# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
```

2) The steps to test directly with commands are as follows:
   a. First, set the GPIO port to output mode. The first parameter of the **pinMode** function is the wPi number corresponding to the pin, and the second parameter is the GPIO mode.

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.pinMode(2, GPIO.OUTPUT) ;  "
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.LOW)"
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.HIGH)"
```

3) The steps for testing in the python3 command line are as follows:
   a. First use the python3 command to enter the python3 command line mode

```
root@orangepi:~# python3
```

   b. Then import the wiringpi python module

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

   c. Then set the GPIO port to output mode, where the first parameter of the **pinMode** function is the wPi number corresponding to the pin, and the second parameter is the GPIO mode

```
>>> wiringpi.wiringPiSetup()
0
>>> wiringpi.pinMode(2, GPIO.OUTPUT)
```

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

```
>>> wiringpi.digitalWrite(2, GPIO.LOW)
```

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

```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

4) wiringOP-Python For setting the GPIO high and low levels in Python code, please refer to the **blink.py** test program in the examples. The **blink.py** test program will set the voltage of all GPIO ports in the 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 shown in the figure below, the available spis for Orange Pi 5 Max are spi0, spi1 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of SPI0, SPI1 and SPI4 in 40 pins are shown in the following table.

	SPI0_M2 corresponds to 40pin	**SPI1_M1 corresponds to 40pin**	**SPI4_M2 corresponds to 40pin**
MOSI	**Pin 19**	**Pin 40**	**Pin 13**
MISO	**Pin 21**	**Pin 38**	**Pin 11**
CLK	**Pin 23**	**Pin 29**	**Pin 15**
CS0	**Pin 24**	**Pin 35**	**Pin 16**
CS1	**Pin 26**	**none**	**none**

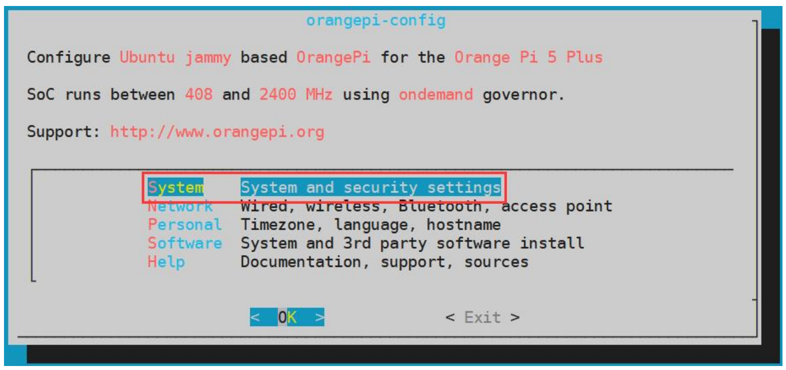
dtbo configuration	**spi0-m2-cs0-spidev** **spi0-m2-cs1-spidev** **spi0-m2-cs0-cs1-spidev**	**spi1-m1-cs0-spidev**	**spi4-m2-cs0-spidev**

3) In Linux system, the SPI in 40 pin is closed by default and needs to be opened manually before it can be used. The detailed steps are as follows:

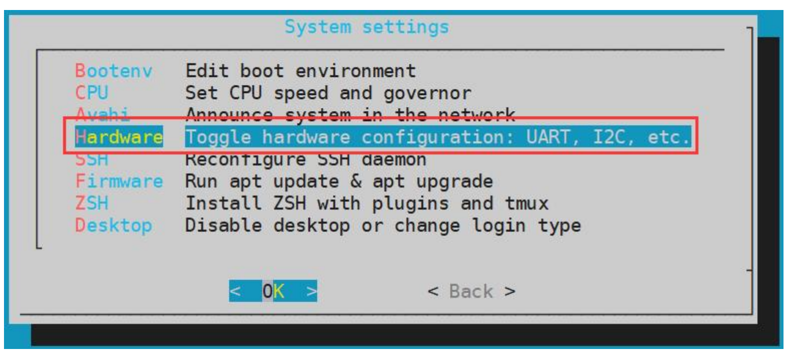
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

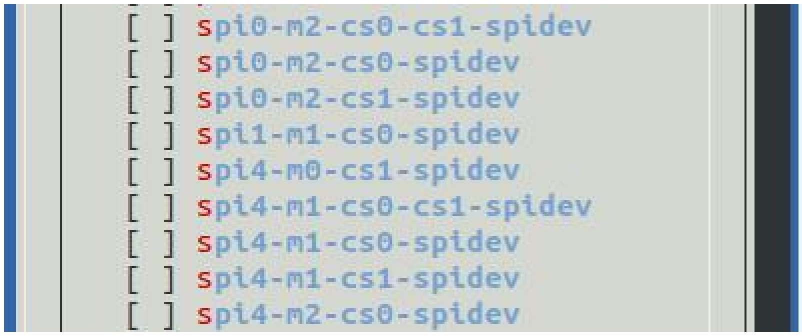
b. Then select **System**



c. Then select **Hardware**

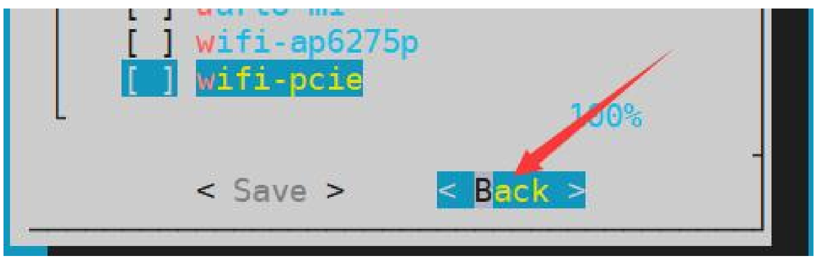


d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the **spacebar** to select the SPI configuration you want to open

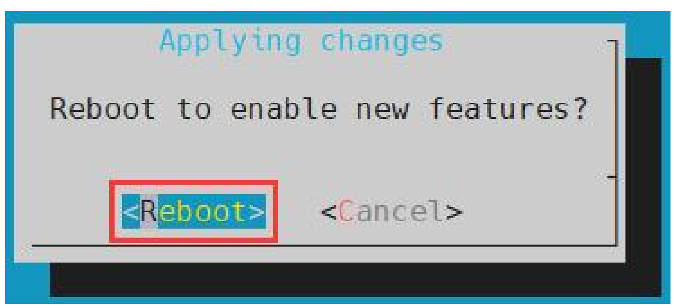


e. Then select **<Save>** to save

f. Then select **<Back>**



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



4) After restarting, enter the system and check whether there is a device node of **spidevx.x** in the Linux system. If it exists, it means that 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 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 SPI loopback function. The **spidev_test.py** program needs to specify the following two parameters:

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

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

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

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel x --port x
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/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 Dupont wire to short the SPI TXD (pin 19 in the 40-pin interface) and RXD (pin 21 in the 40-pin interface) pins and run spidev_test.py. The output is as follows. You can see that the data sent and received are the same, indicating that the SPI loopback test is normal.

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

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel x --port x
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/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 test

1) As can be seen from the table below, Orange Pi 5 Max has four i2c buses: i2c2, 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of the 4 groups of I2C buses in 40pin are shown in the following table. I2C2_M0 and I2C2_M4, I2C5_M2 and I2C5_M3 can only use one of them at the same time, they cannot be used at the same time, they are all the same I2C, just connected to different pins, please do not think that they are two different I2C buses.

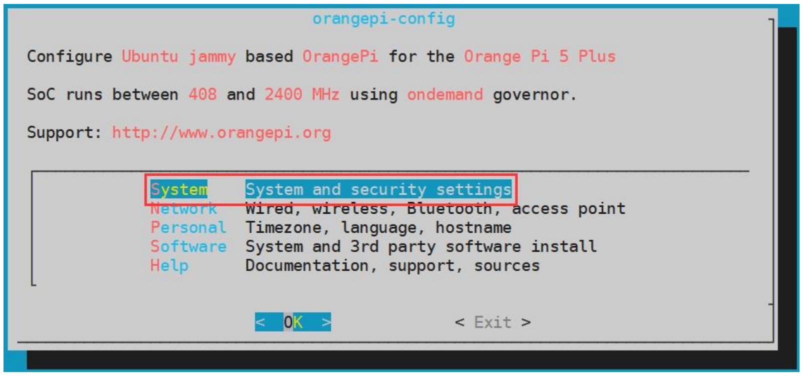
I2C Bus	SDA corresponds to 40pin	SCL corresponds to 40pin	dtbo corresponding configuration
I2C2_M0	**Pin 3**	**Pin 5**	**i2c2-m0**
I2C2_M4	**Pin 11**	**Pin 13**	**i2c2-m4**
I2C4_M3	**Pin 15**	**Pin 16**	**i2c4-m3**
I2C5_M2	**Pin 37**	**Pin 12**	**i2c5-m2**
I2C5_M3	**Pin 27**	**Pin 28**	**i2c5-m3**
I2C8_M2	**Pin 36**	**Pin 32**	**i2c8-m2**

3) In Linux system, the I2C bus in 40 pins is closed by default and needs to be opened manually before it can be used. The detailed steps are as follows:

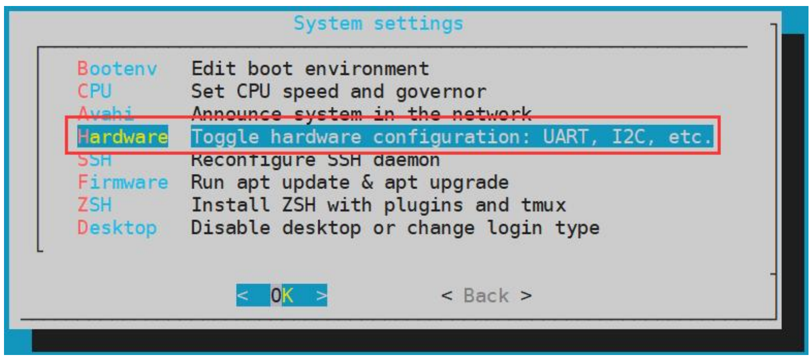
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

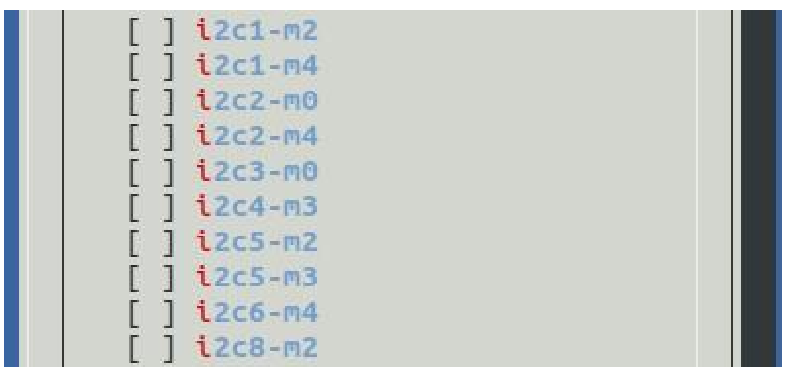
b. Then select **System**



c. Then select **Hardware**

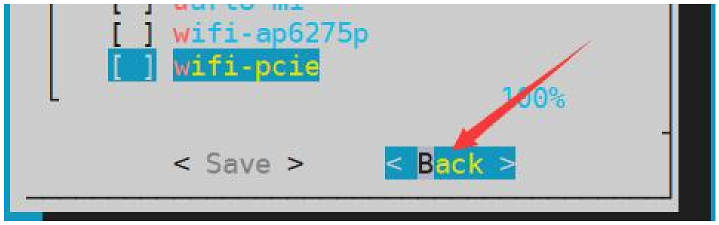


d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the **spacebar** to select the I2C configuration you want to open

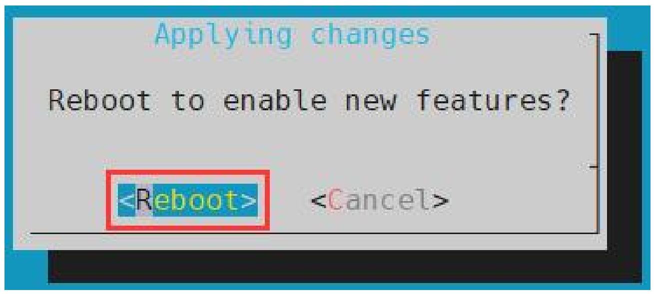


e. Then select <Save>

f. Then select**<Back>**



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



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

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

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 2       #i2c2 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
```

```
root@orangepi5max:~# i2cdetect -y 2
     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@orangepi5max:~#
```

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-2"
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 table below, Orange Pi 5 Max has four uart buses available: uart1, uart3, uart4, and uart6

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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of the four UART buses in 40 pins are shown in the following table:

UART Bus	RX corresponds to 40pin	TX corresponds to 40pin	dtbo corresponding configuration
UART1_M1	**Pin 27**	**Pin 28**	**uart1-m1**
UART3_M1	**Pin 33**	**Pin 31**	**uart3-m1**

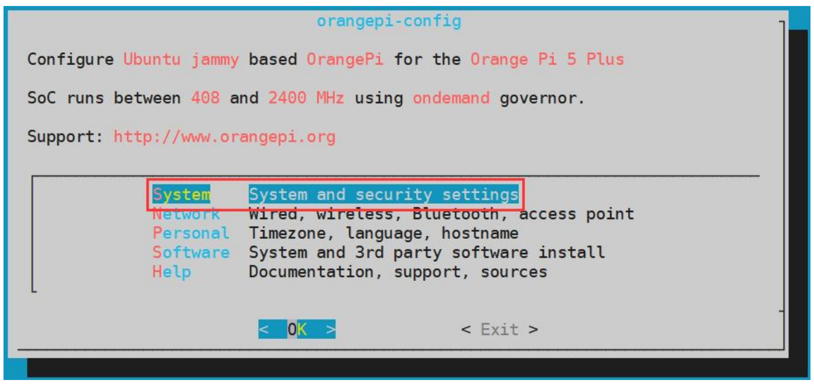
UART4_M2	**Pin 19**	**Pin 23**	**uart4-m2**
UART6_M1	**Pin 11**	**Pin 13**	**uart6-m1**

3) In Linux system, the UART in 40 pin is disabled by default and needs to be enabled manually. The detailed steps are as follows:

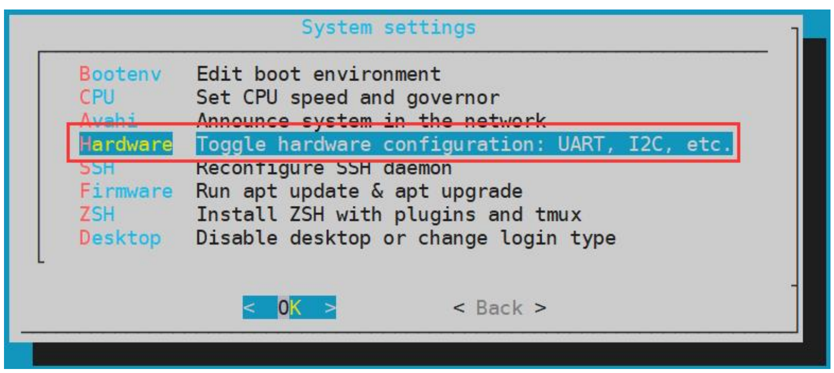
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

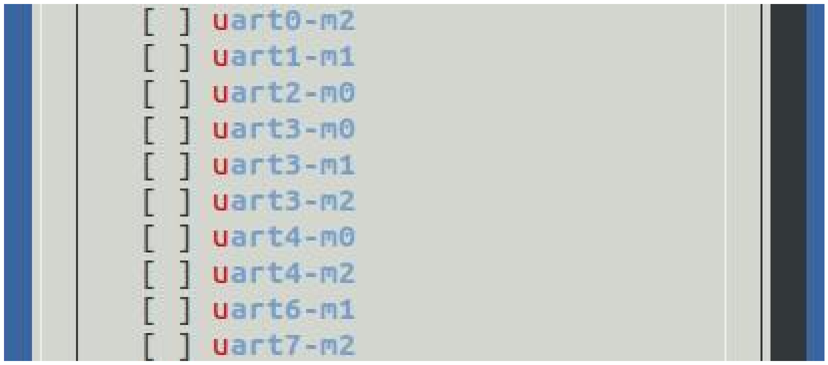
b. Then select **System**



c. Then select **Hardware**

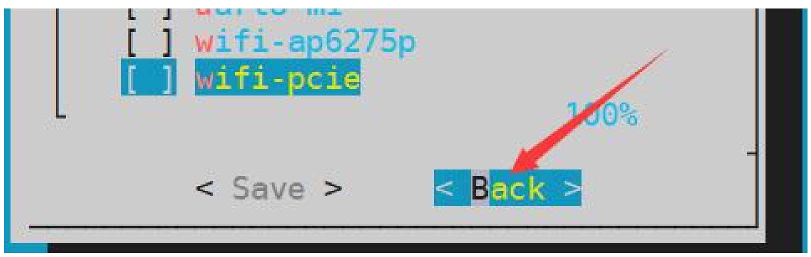


d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the **spacebar** to select the UART configuration you want to open

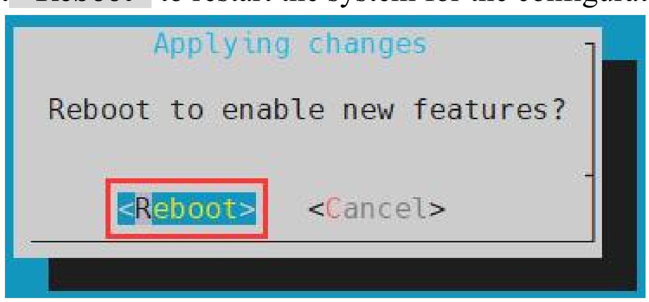


e. Then select <Save> to save

f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system for the configuration to 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 the Dupont line to short-circuit the rx and tx of the UART interface to be tested.

6) Use the **serialTest.py** program in the examples to test the loopback function of the serial port as shown below. If you can see the following print, it means that the serial port communication is normal.

/dev/ttySX needs to be replaced with the specific uart device node number.

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

The method of running the watchdog_test program is as follows:

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

```
orangepi@orangepi:~$ sudo watchdog_test 10
open success
options is 33152,identity is sunxi-wdt
put_usr return,if 0,success:0
The old reset time is: 16
return ENOTTY,if -1,success:0
return ENOTTY,if -1,success:0
put_user return,if 0,success:0
put_usr return,if 0,success:0
keep alive
```

```
keep alive
keep alive
```

3. 22. Check the serial number of the RK3588 chip

The command to view the serial number of the RK3588 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 enabled by default.

2) Use the **enable_docker.sh** script to enable the docker service, 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 you can run hello-world, it means that docker can be used normally.

```
orangepi@orangepi:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
256ab8fe8778: Pull complete
Digest:
sha256:7f0a9f93b4aa3022c3a4c147a449ef11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
.....
```

3.24. How to download and install the arm64 version of balenaEtcher

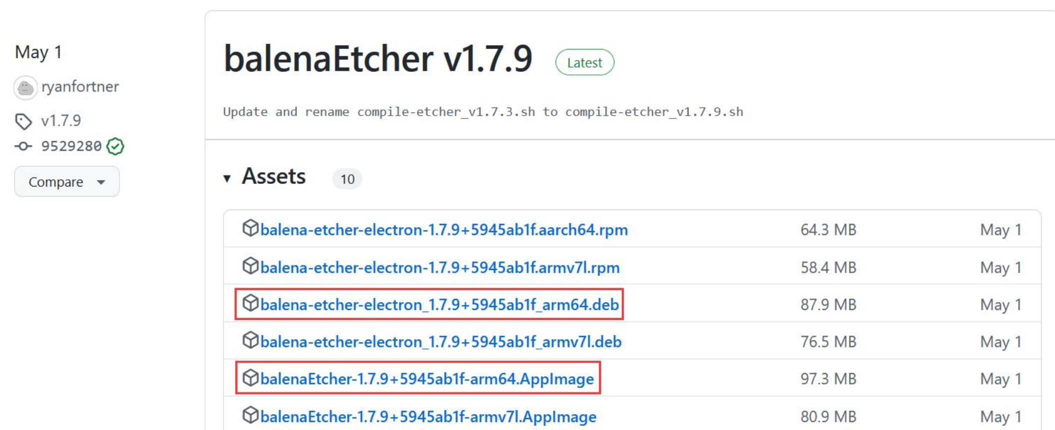
1) The download address of balenaEtcher arm64 version is:

   a. The download address of the deb installation package is as follows, which needs to be installed before use

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

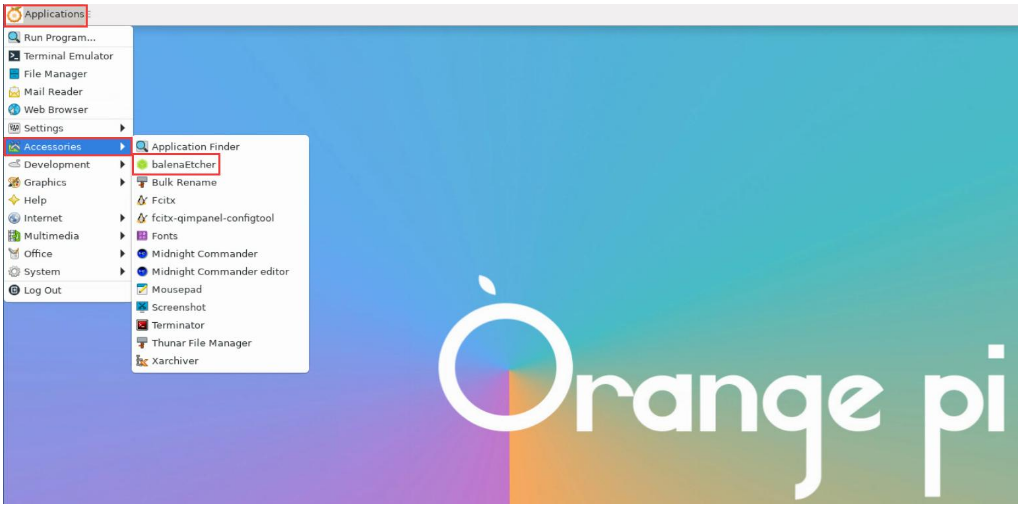


2) How to install and use the deb version of balenaEtcher:

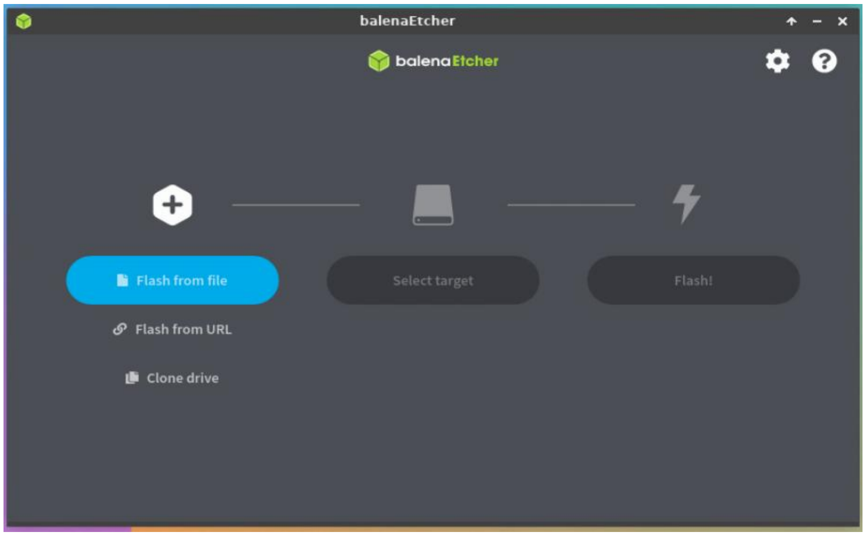
   a. deb version of balenaEtcher installation command is as follows:

```
orangepi@orangepi:~$ sudo apt install -y    \
--fix-broken ./balena-etcher-electron_1.7.9+5945ab1f_arm64.deb
```

   b. After the deb version of balenaEtcher is installed, you can open it in Application



c. balenaEtcher opens with the following interface:

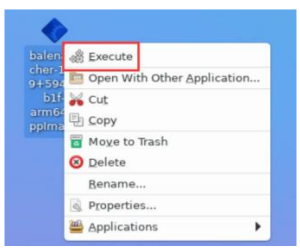


3) How to use the AppImage version of balenaEtcher:

a. First, add permissions to balenaEtcher

```
orangepi@orangepi:~/Desktop$ chmod +x balenaEtcher-1.7.9+5945ab1f-arm64.AppImage
```

b. Then select the AppImage version of balenaEtcher, right-click your mouse, and click Execute to open balenaEtcher.



3. 25. How to install Baota Linux Panel

Baota Linux Panel is a server management software that improves operation and maintenance efficiency. It supports more than 100 server management functions such as one-click LAMP/LNMP/cluster/monitoring/website/FTP/database/JAVA (excerpted from Baota official website)

1) The recommended order of Baota 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

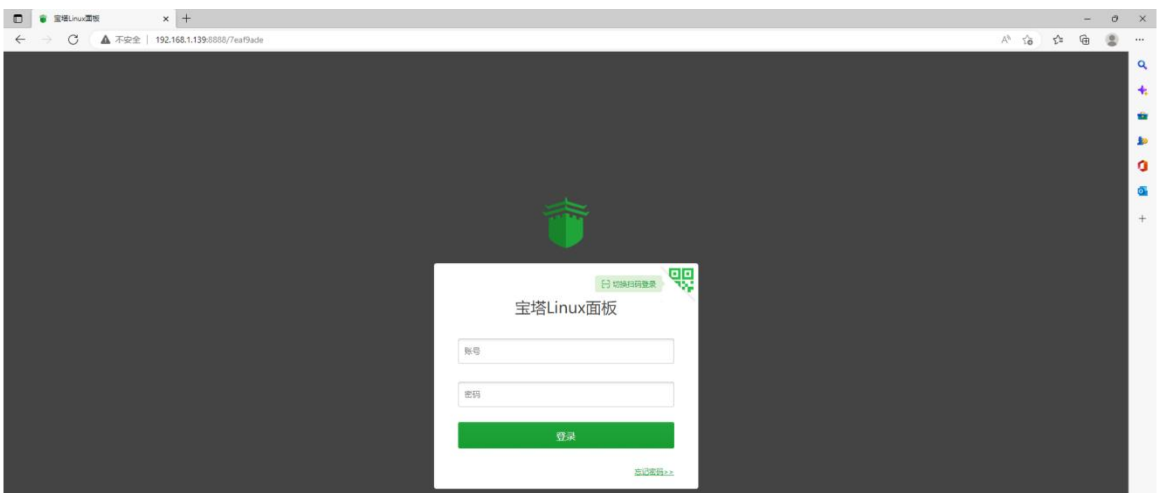
```
orangepi@orangepi:~$ sudo install_bt_panel.sh
```

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

```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| The WebPanel URL will be http://SERVER_IP:8888 when installed.
+----------------------------------------------------------------------

Do you want to install Bt-Panel to the /www directory now?(y/n): y
```

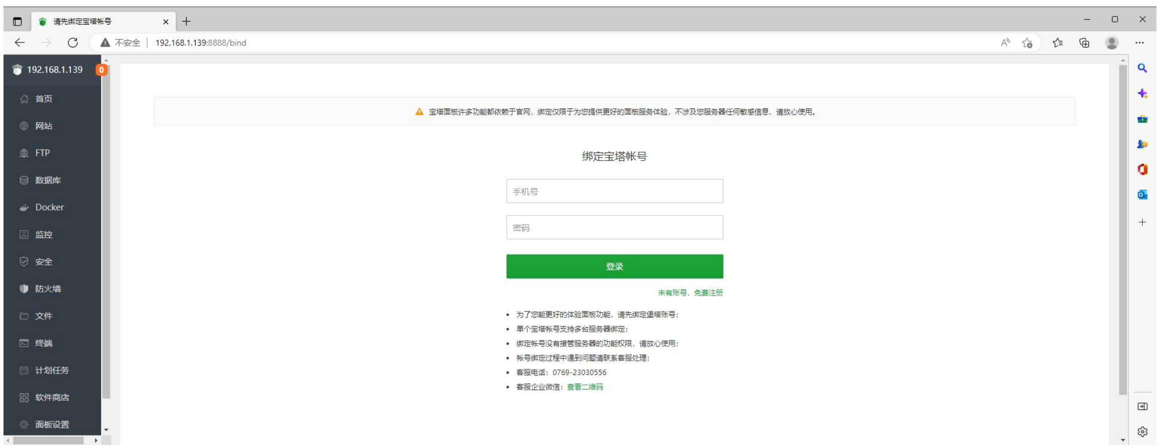
4) Then all you have to do is wait patiently. When you see the following print information output by the terminal, it means that the pagoda has been installed. The entire installation process takes about 6 minutes, which may vary depending on the network speed.

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

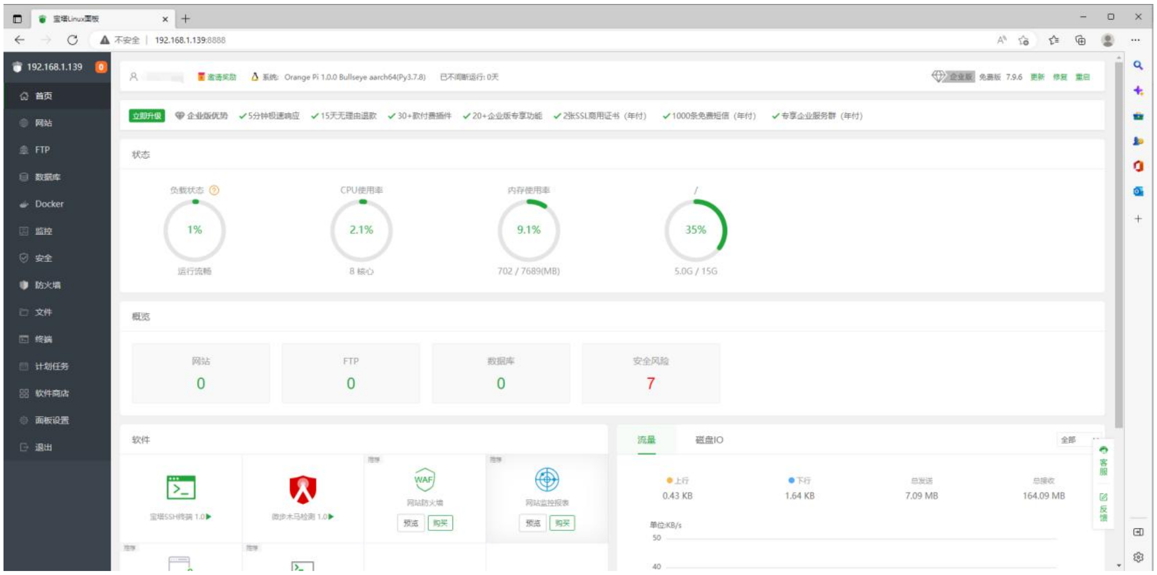


6) After successfully logging into the pagoda, the following welcome interface will pop up. First, please read the user instructions in the middle and drag them to the bottom. Then you can select "I have agreed and read the "User Agreement"", and then click "Enter the Panel" to enter the pagoda.

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

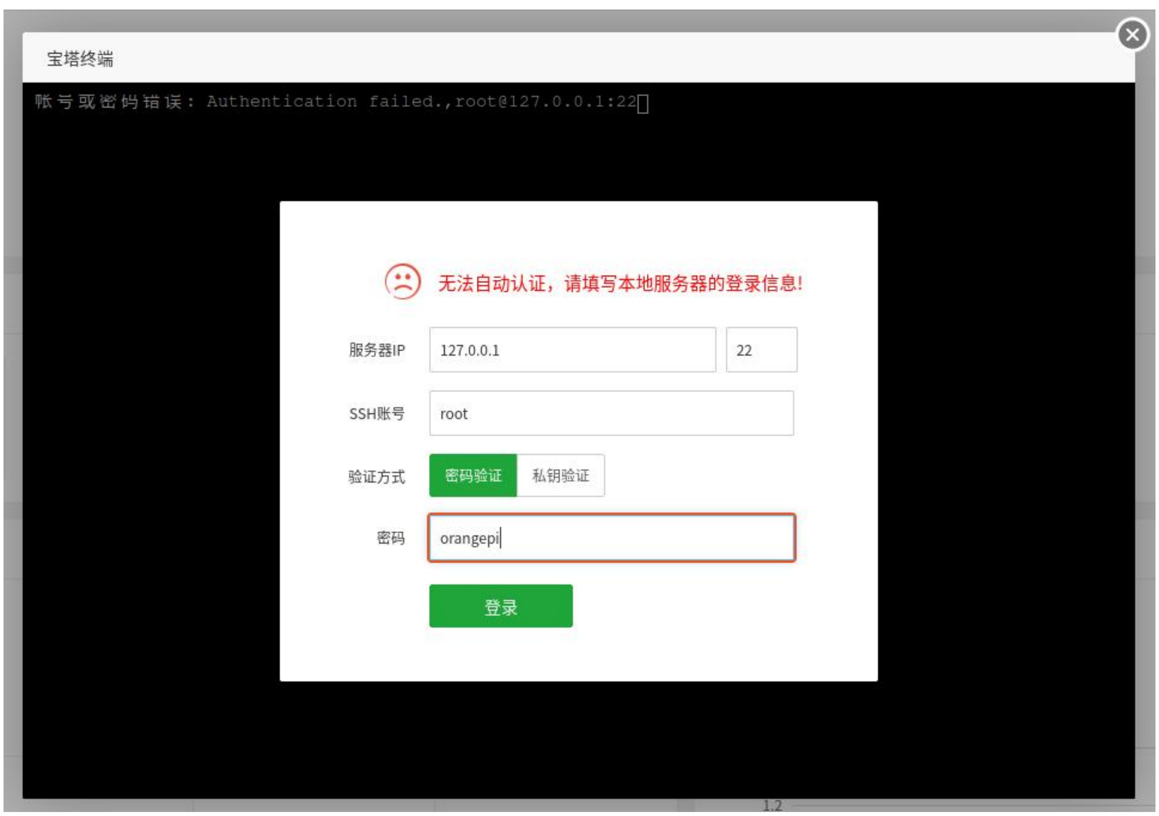


8) The final interface is shown in the figure below. You can intuitively see some status information of the development board Linux system, such as load status, CPU usage, memory usage, and storage space usage.

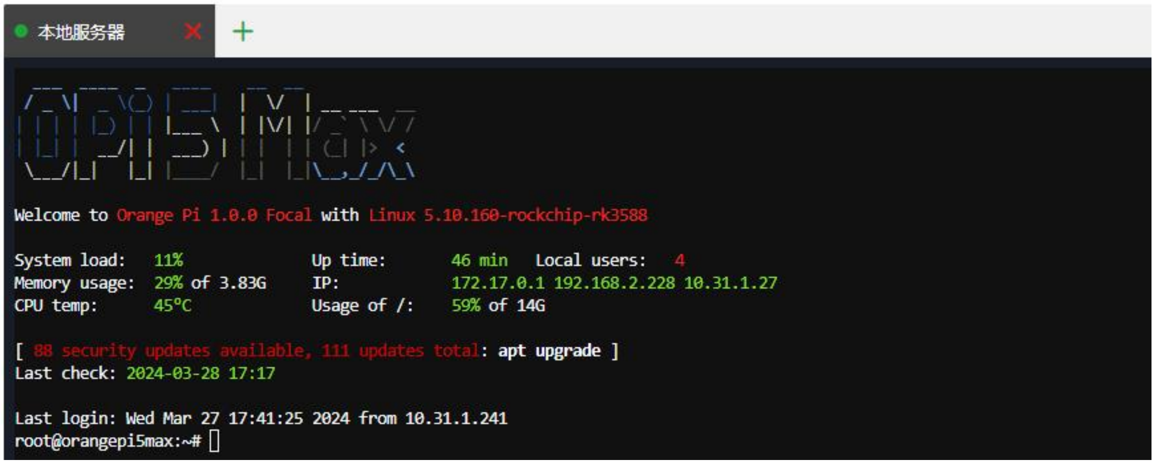


9) Test the SSH terminal login of Baota

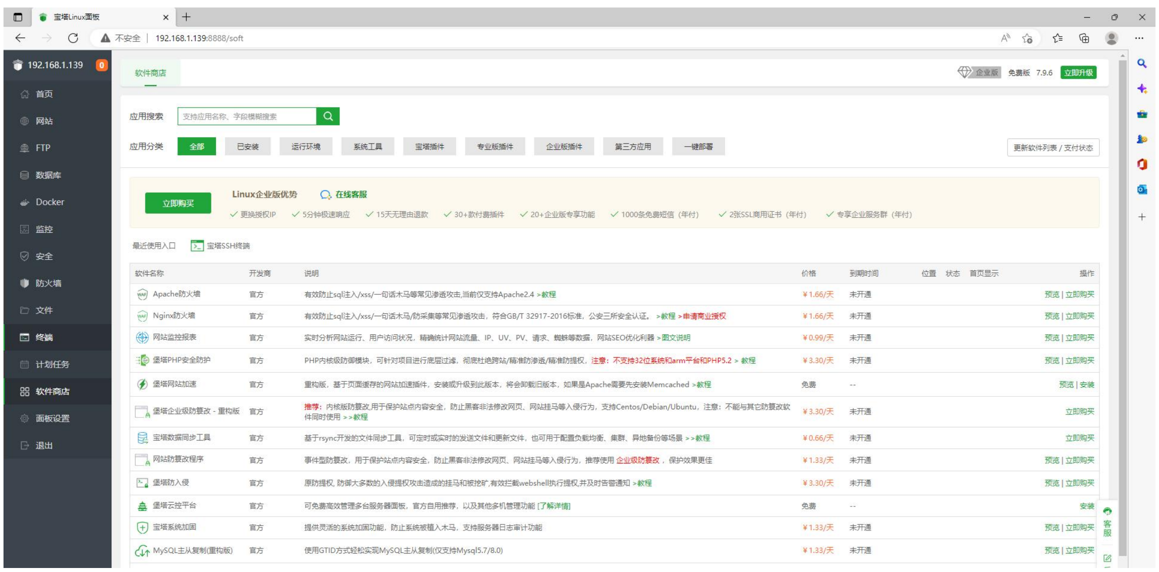
a. After opening the SSH terminal of Baota, you will be prompted to enter the password of the development board system. At this time, enter **orangepi** in the password box (the default password, if you have changed it, please fill in the modified password)



b. The display after successful login is as shown below



10) You can install Apache, MySQL, PHP and other software in Baota's software store, 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) Baota command line tool test

```
orangepi@orangepi:~$ 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://116.30.139.180:31904/0635e0b2
内网面板地址: https://192.168.2.228:31904/0635e0b2
username: uykcsucu
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@orangepi:~$
```

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

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

3. 26. Set up Chinese environment and install 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.

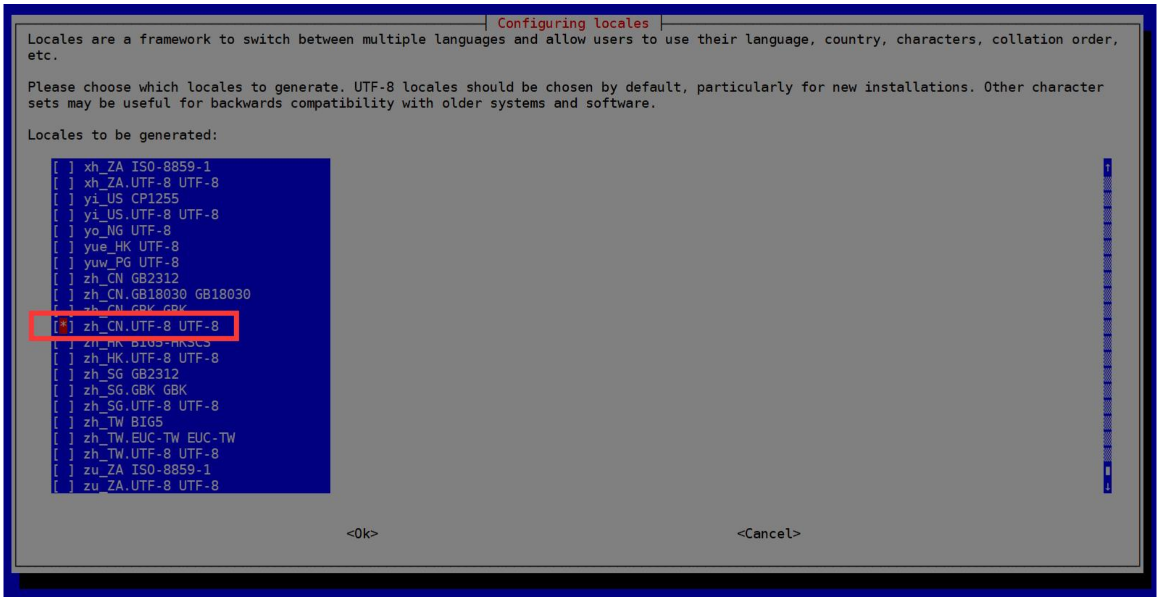
3. 26. 1. Debian system installation method

1) First set the default **locale** to Chinese

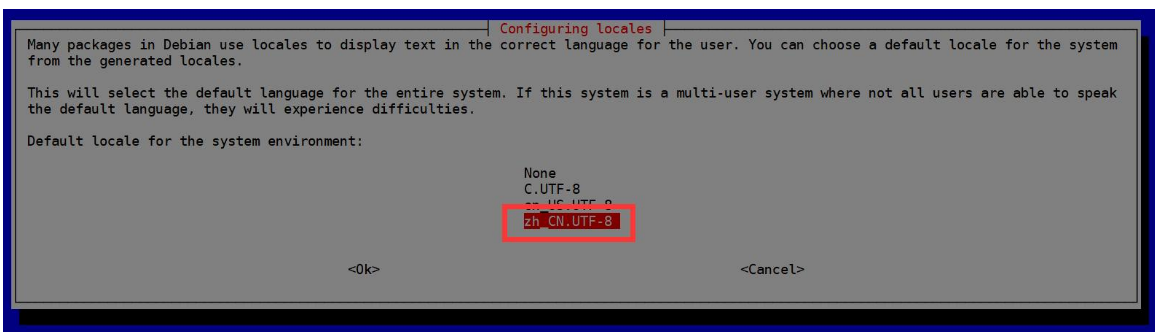
a. Enter the following command to start configuring **locale**

```
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 arrow 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>**, then press Enter)



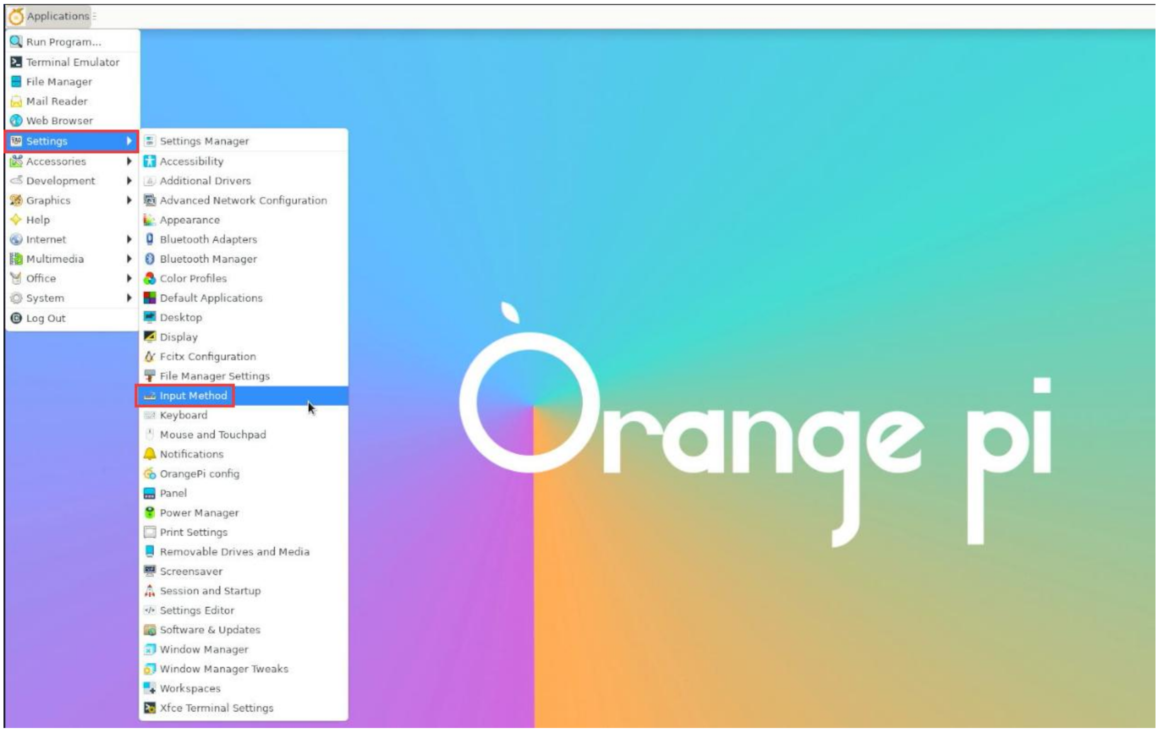
c. 然后 Then set the default **locale** to **zh_CN.UTF-8**



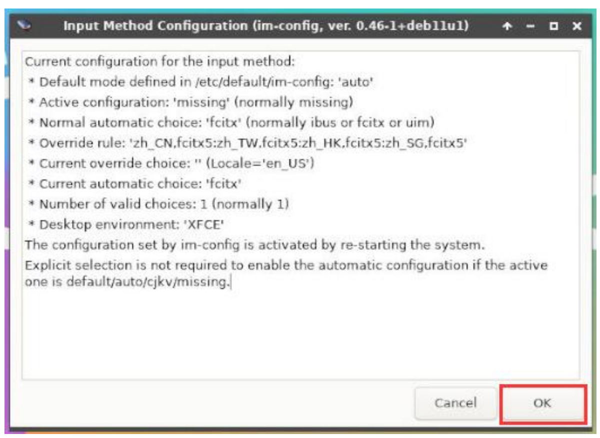
d. After exiting the interface, the **locale** setting will begin. The output displayed on the command line is as follows

```
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.
```

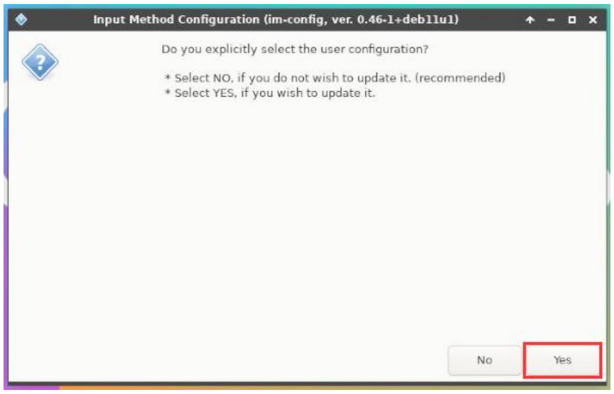
2) Then open **Input Method**



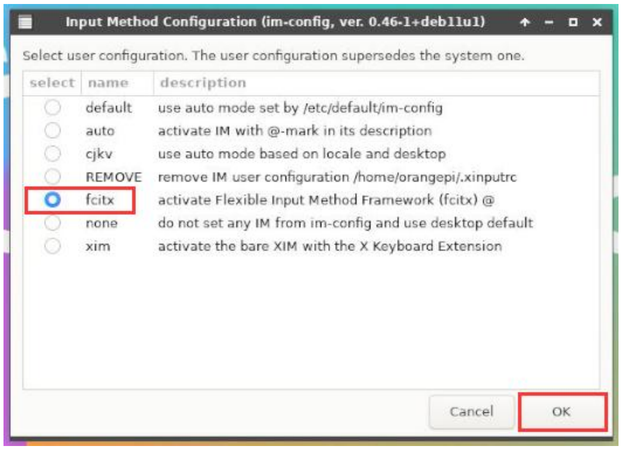
3) Then select **OK**



4) Then select **Yes**



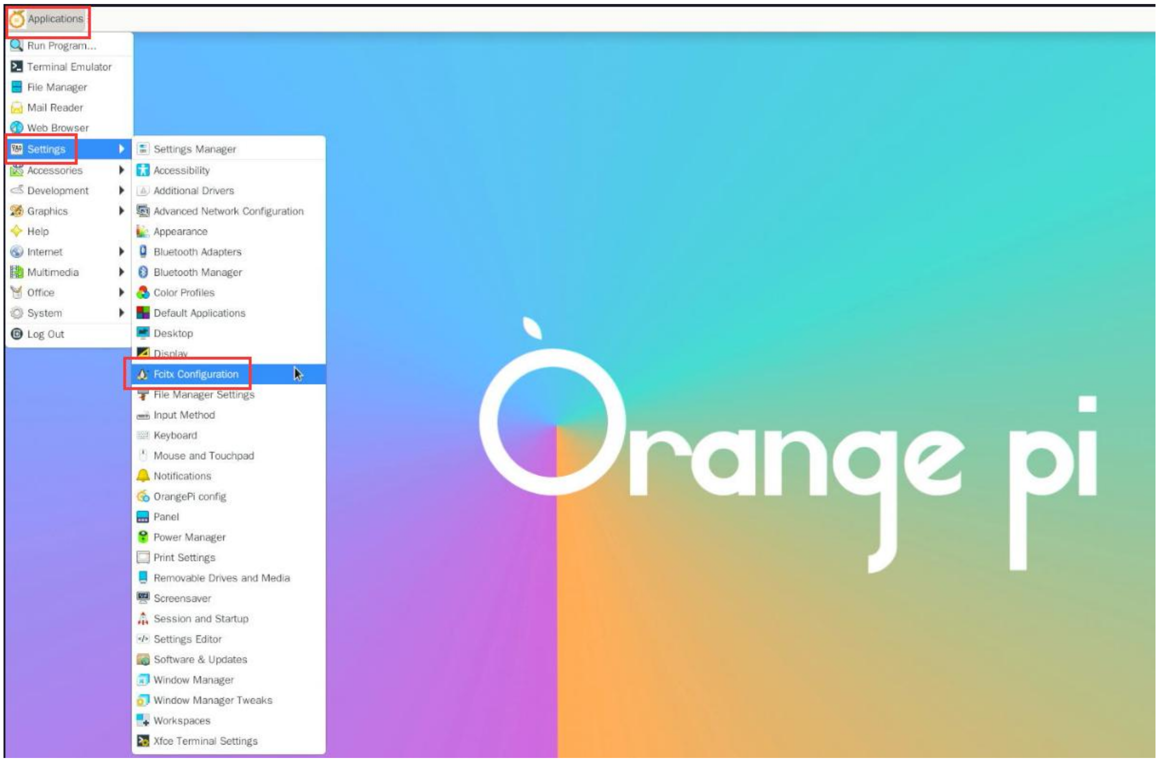
5) Then select **fcitx**



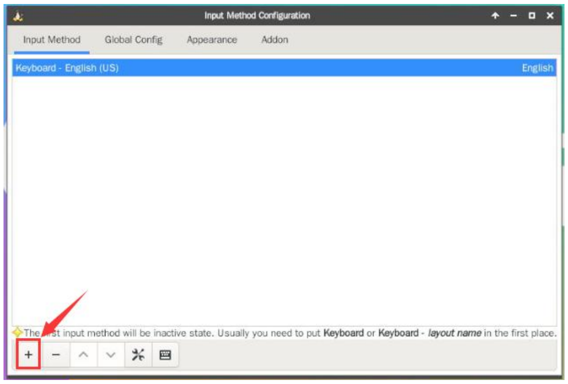
6) Then select **OK**

7) **Then restart the Linux system to make the configuration take effect**

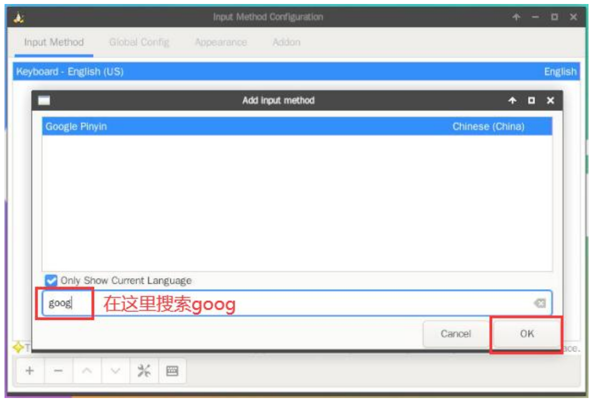
8) Then open **Fcitx configuration**



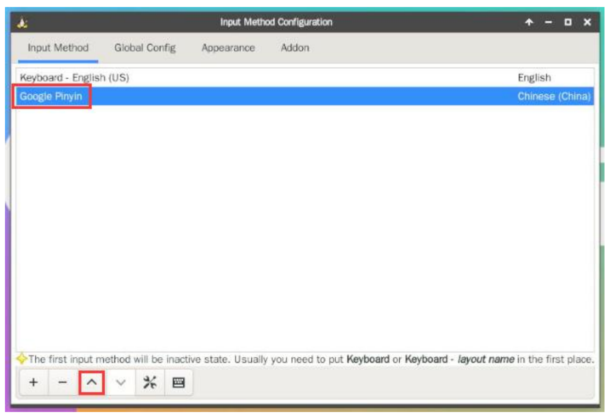
9) Then click the + sign in the position shown in the figure below

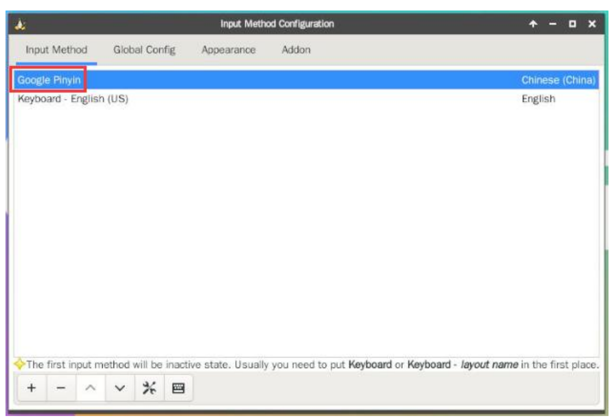


10) Then search **Google Pinyin** and click **OK**



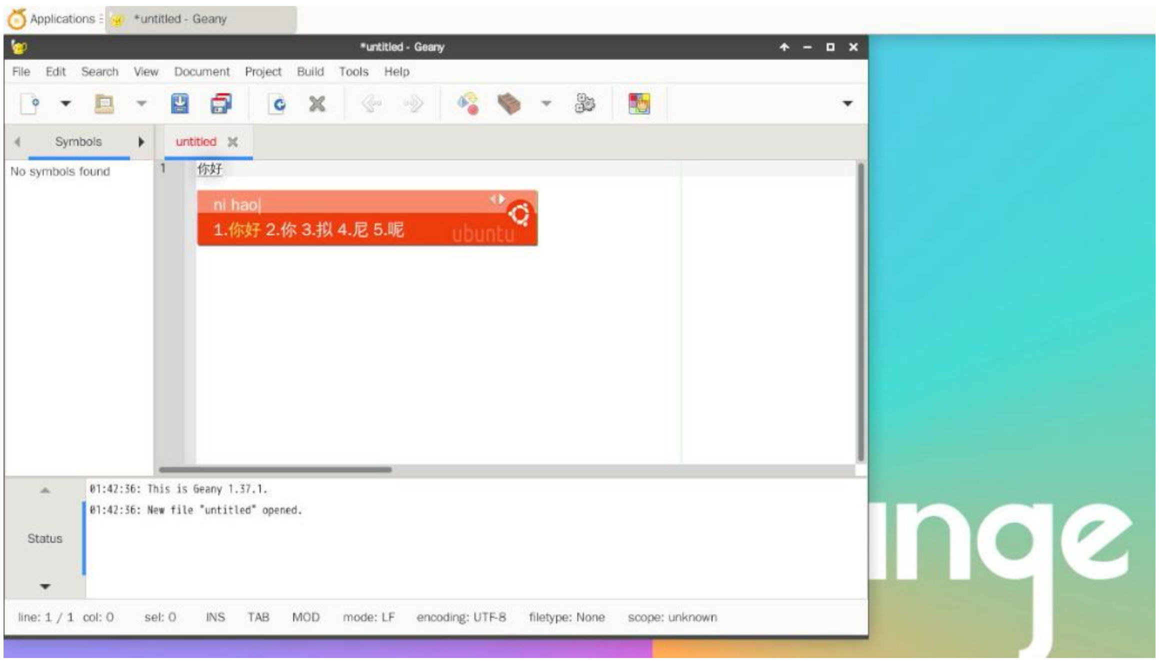
11) Then put **Google Pinyin** at the front





12) Then open the **Geany** editor to test the Chinese input method

13) The Chinese input method test is as follows



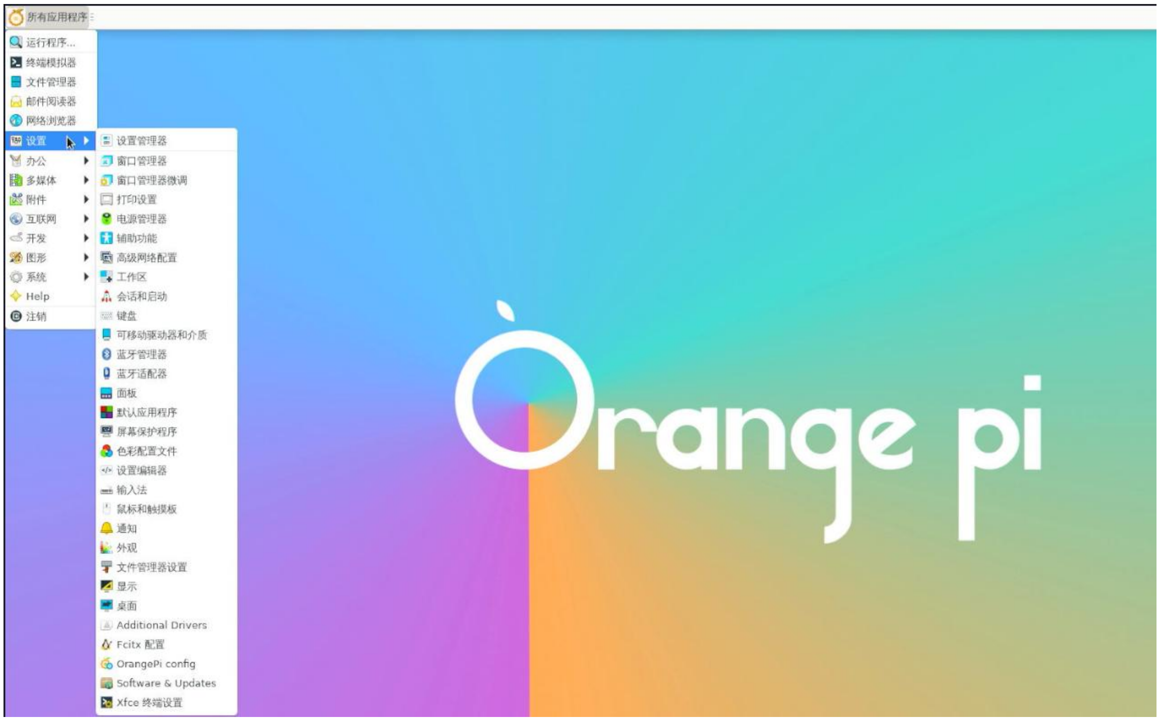
14) Use the **Ctrl+Space** shortcut key to switch between Chinese and English input methods

15) If you need the entire system to display in Chinese, you can set the variables in **/etc/default/locale** to **zh_CN.UTF-8**

```
orangepi@orangepi:~$ sudo vim /etc/default/locale
#    File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
```

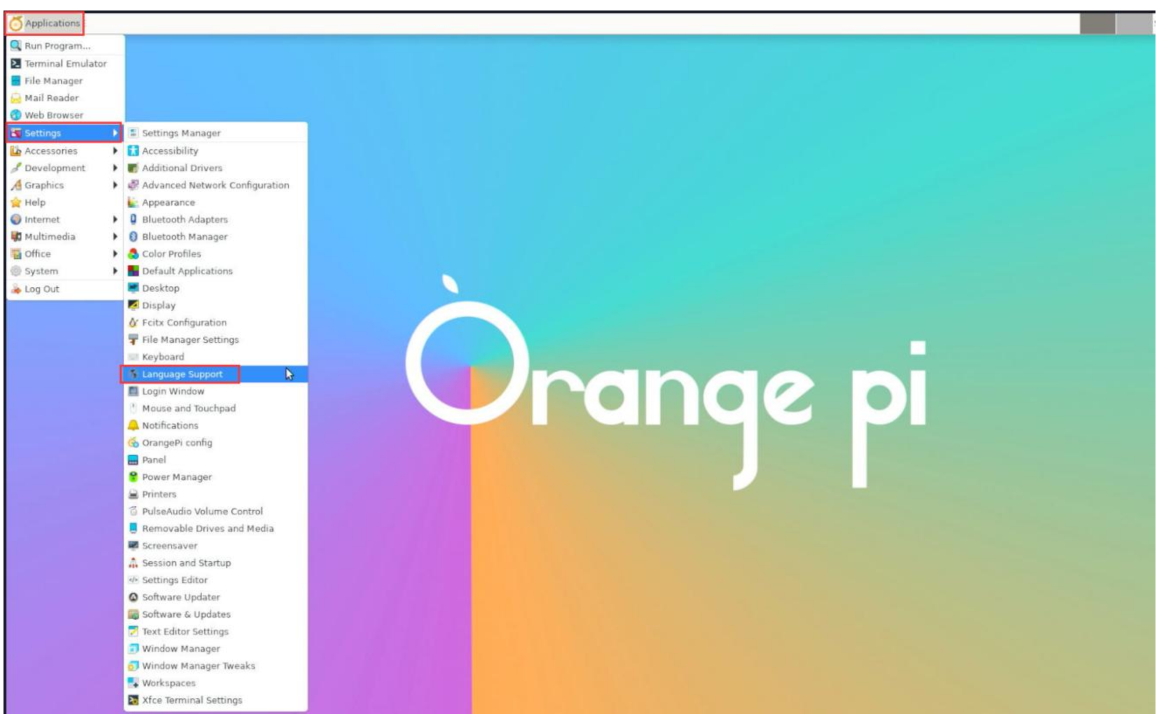
```
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** and you can see that the system is displayed in Chinese.

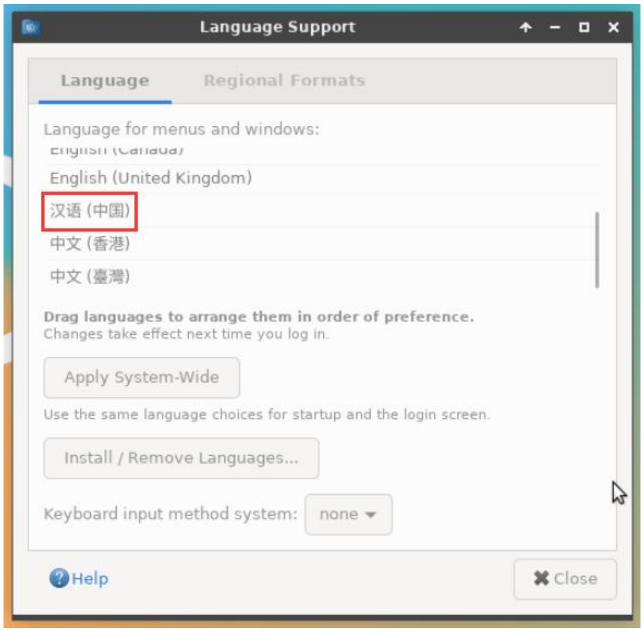


3. 26. 2. How to install Ubuntu 20.04 system

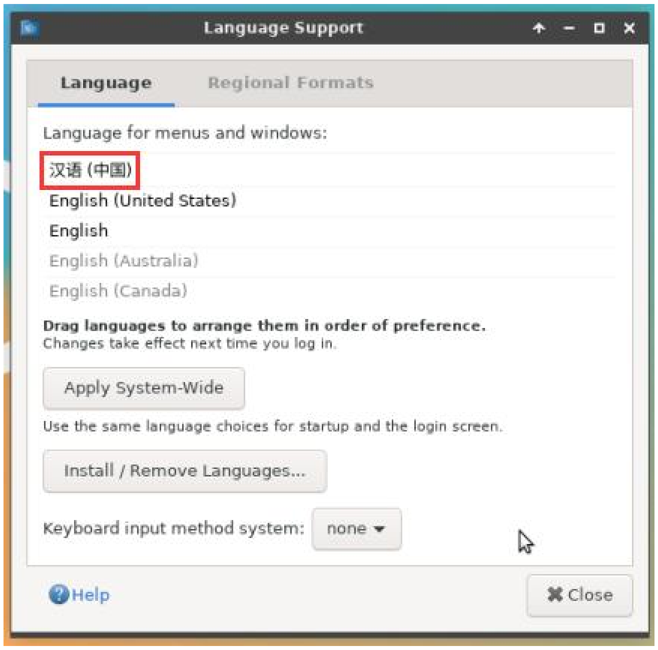
1) First open **Language Support**



2) Then find the **Chinese (China)** option

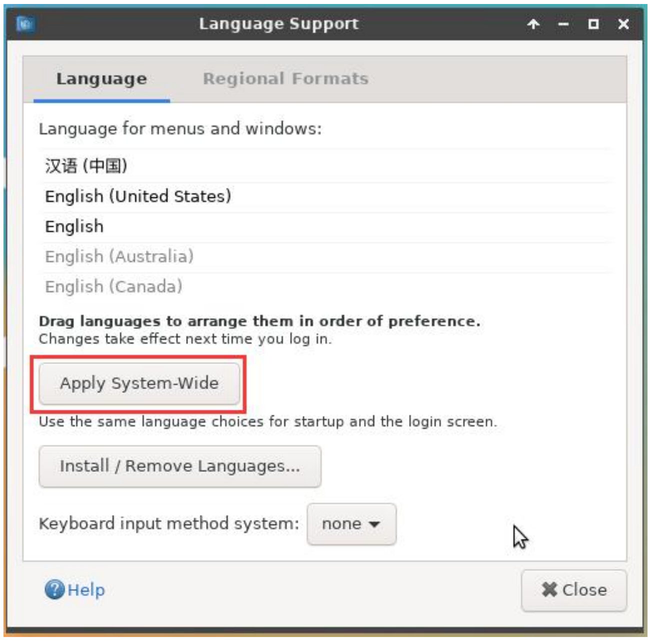


3) Then use the left mouse button to select **Chinese (China)** and hold it down, then drag it upwards to the starting position. The display after dragging is as shown below:

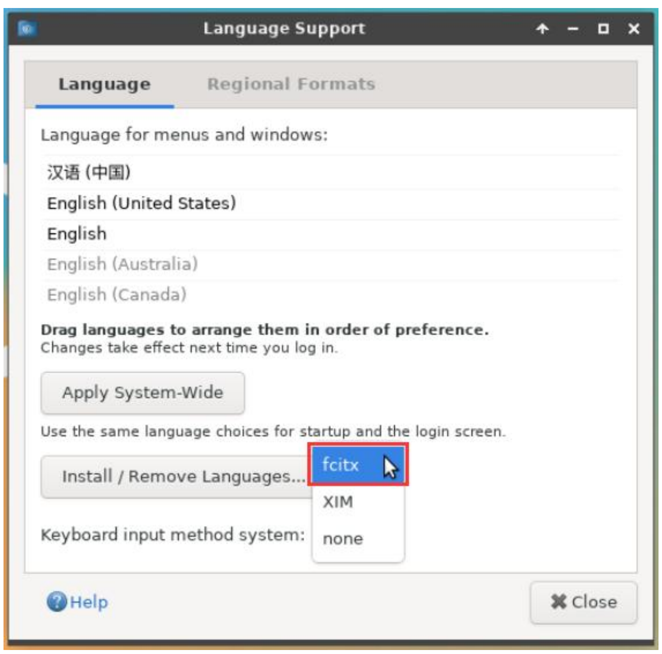


Note that this step is not easy to drag, please be patient and try a few more times.

4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



5) Then set the **Keyboard input method system to fcitx**

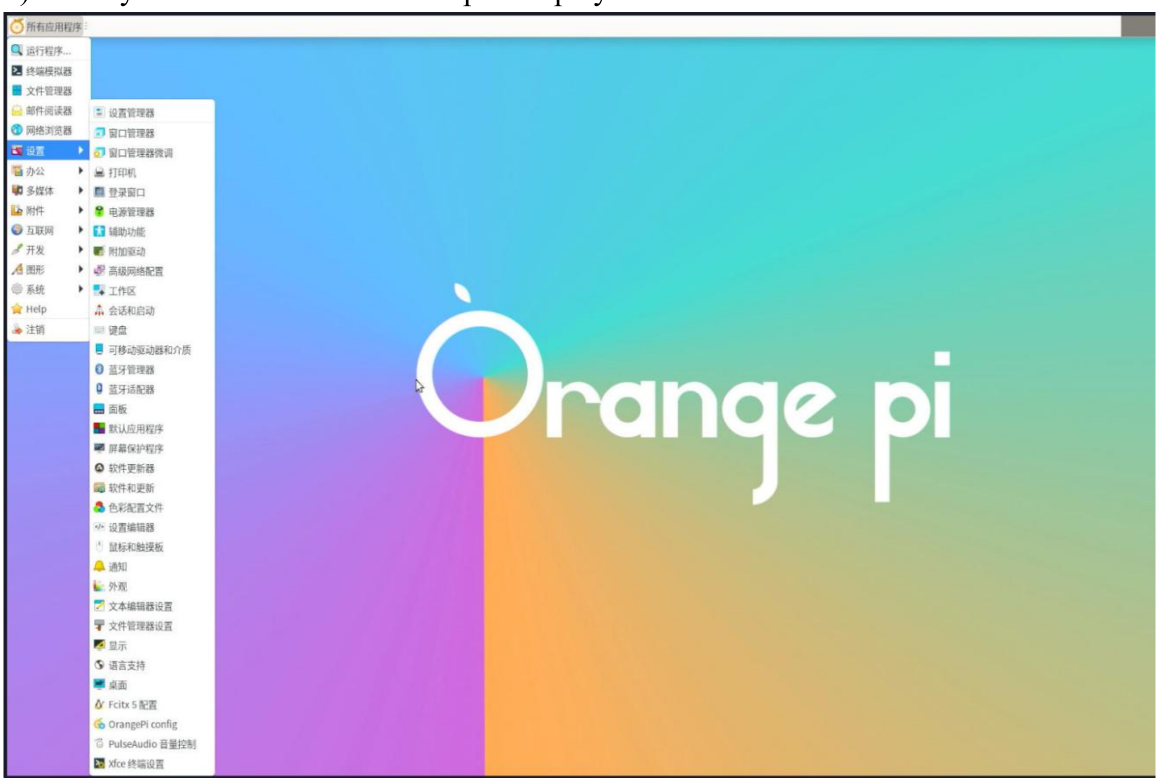


6) **Then restart the Linux system to make the configuration take effect**

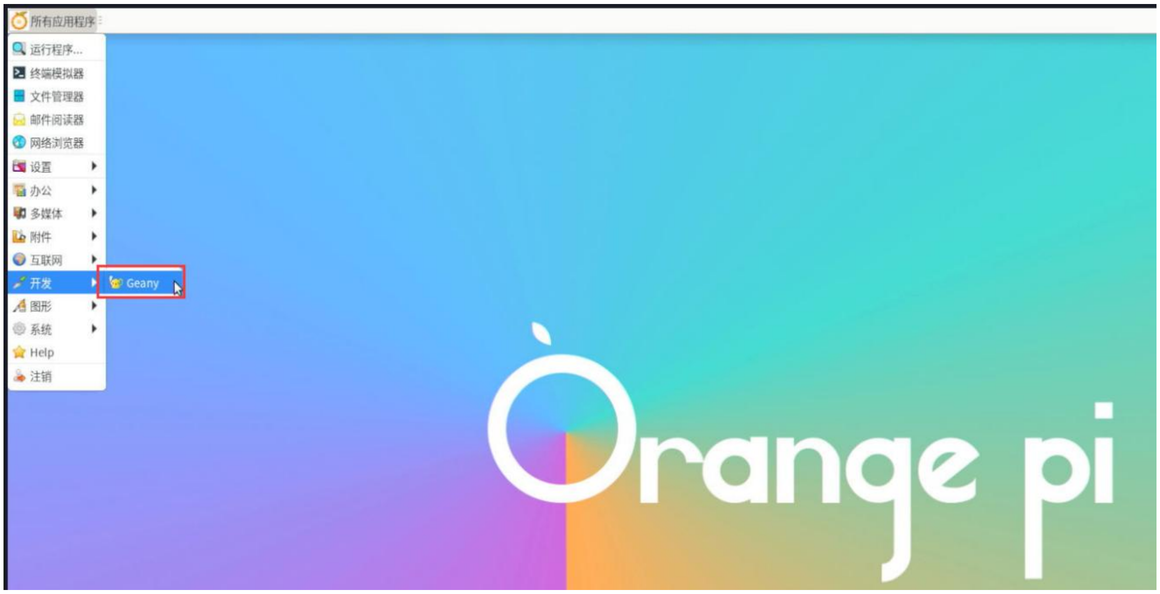
7) After re-entering the system, please select "**Do not ask me again**" in the following interface, and then decide whether to update the standard folder to Chinese according to

your own preferences.

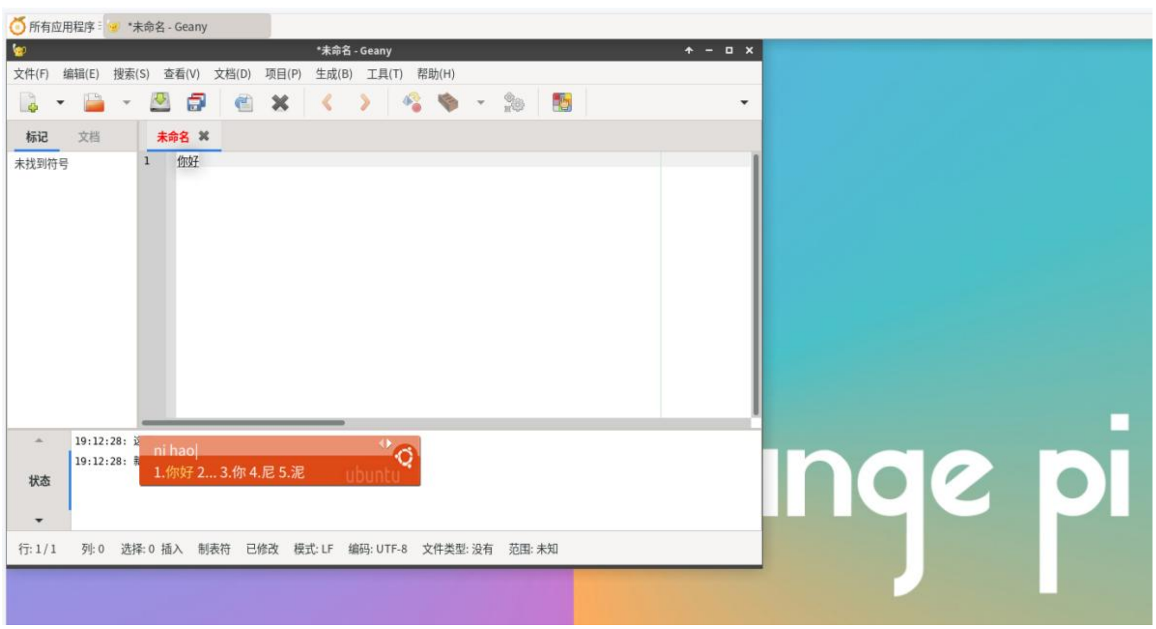
8) Then you can see that the desktop is displayed in Chinese



9) Then we can open **Geany** to test the Chinese input method. The opening method is as shown in the figure below

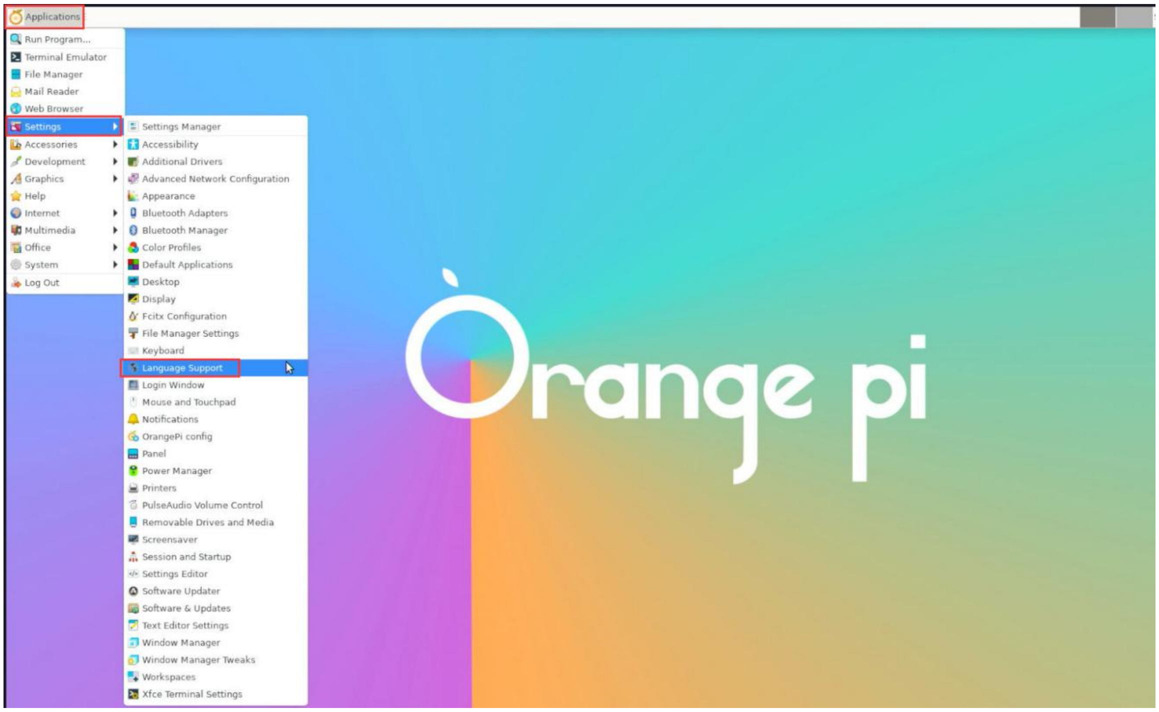


10) After opening **Geany**, the default input method is still English. We can switch to Chinese input method by pressing **Ctrl+Space,** and then we can input Chinese.

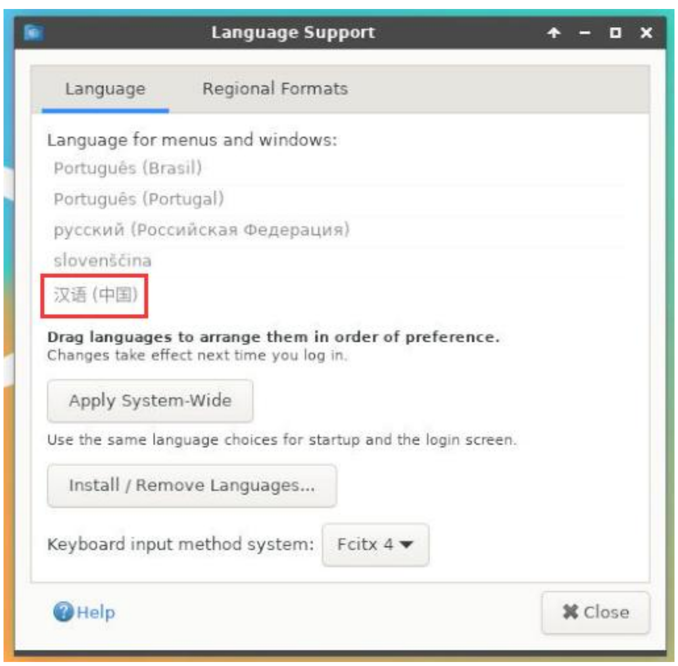


3. 26. 3. Installation method for Ubuntu 22.04 system

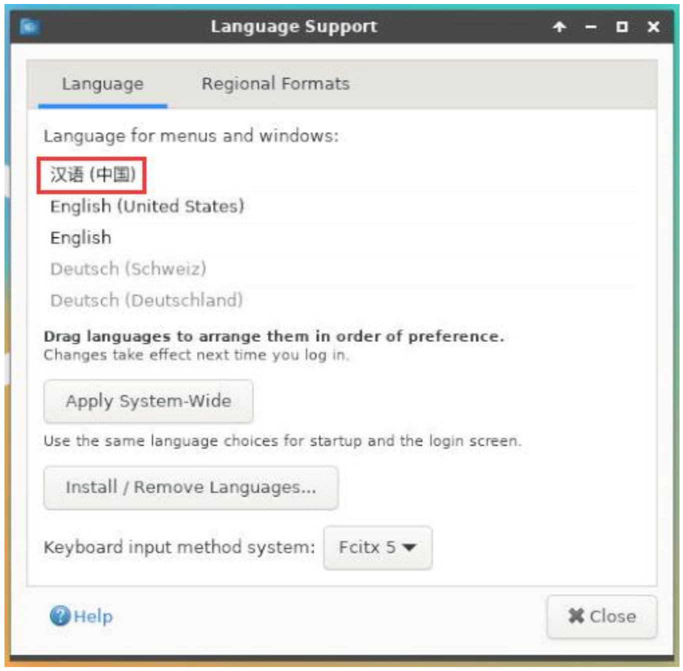
1) **Language Support**



2) Then find the **Chinese (China)** option

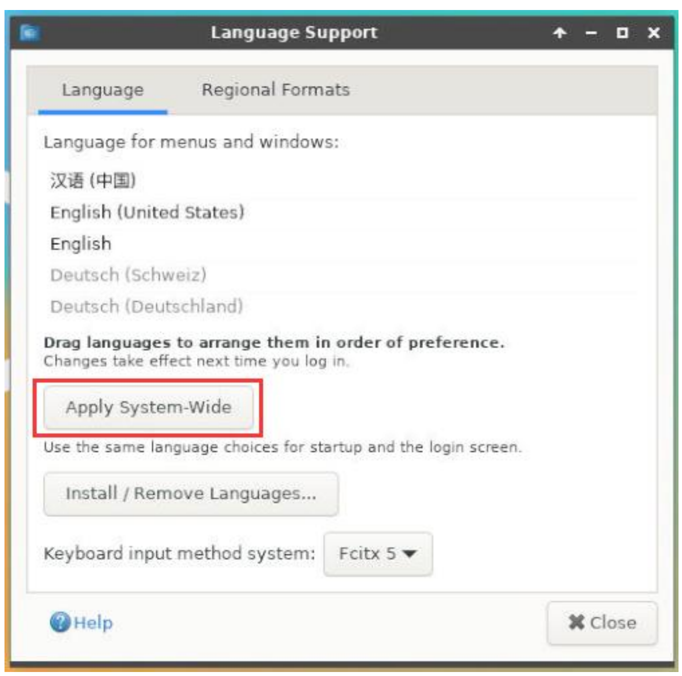


3) Then use the left mouse button to select **Chinese (China)** and hold it down, then drag it upwards to the starting position. The display after dragging is as shown below:



Note that this step is not easy to drag, please be patient and try a few more times.

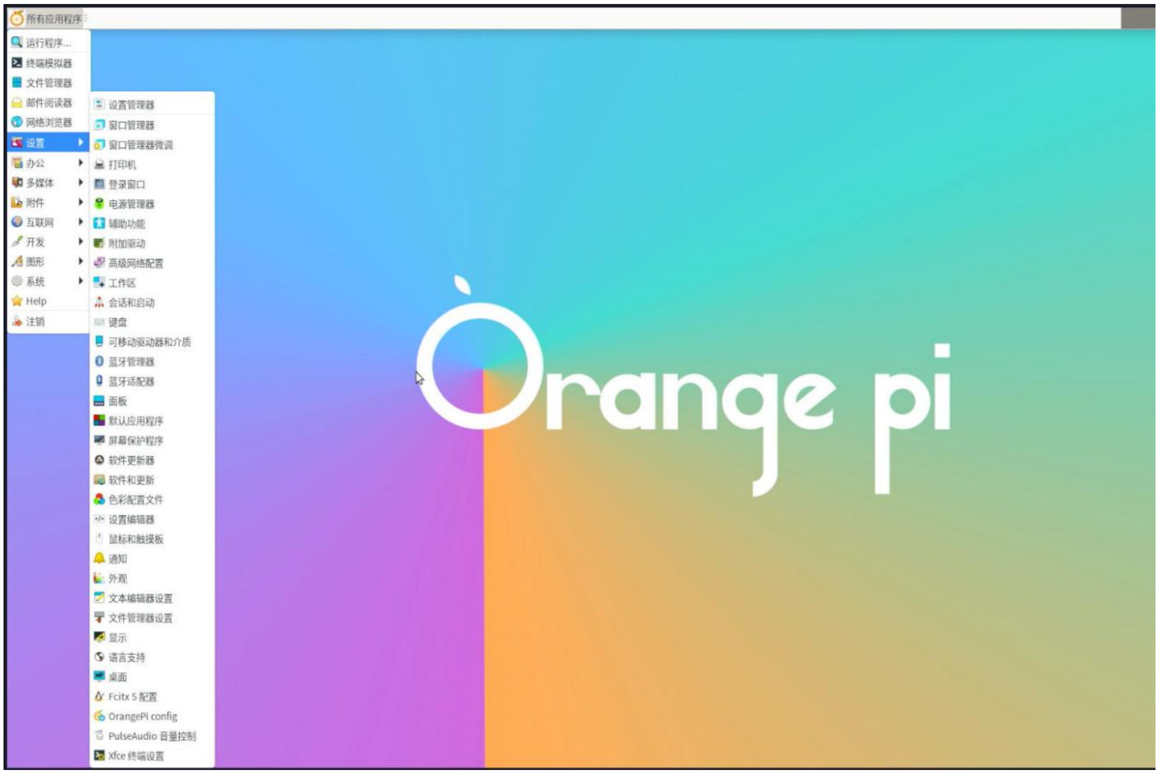
4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



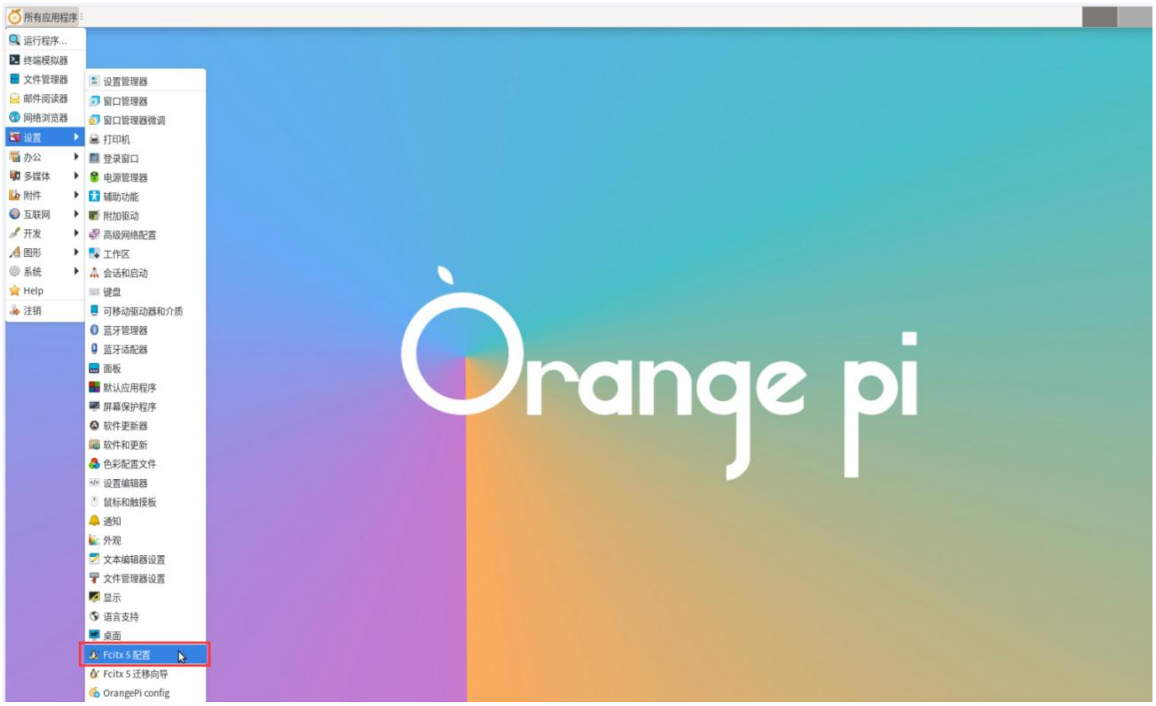
5) **Then restart the Linux system to make the configuration take effect**

6) After re-entering the system, please select "**Do not ask me again**" in the following interface, and then decide whether to update the standard folder to Chinese according to your own preferences.

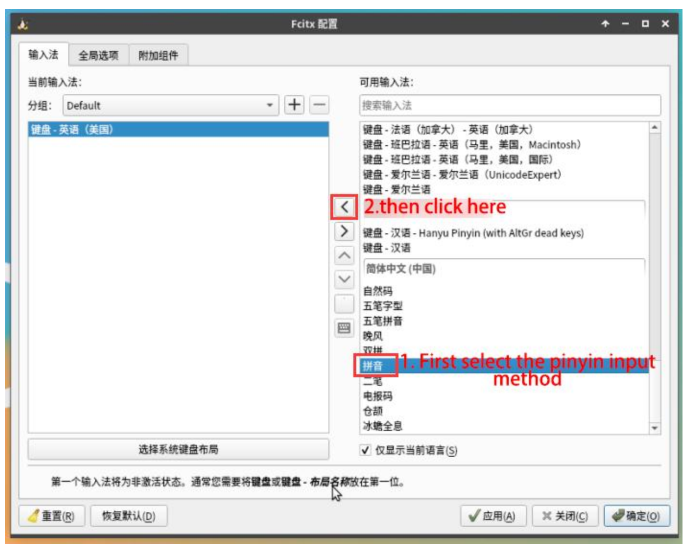
7) Then you can see that the desktop is displayed in Chinese



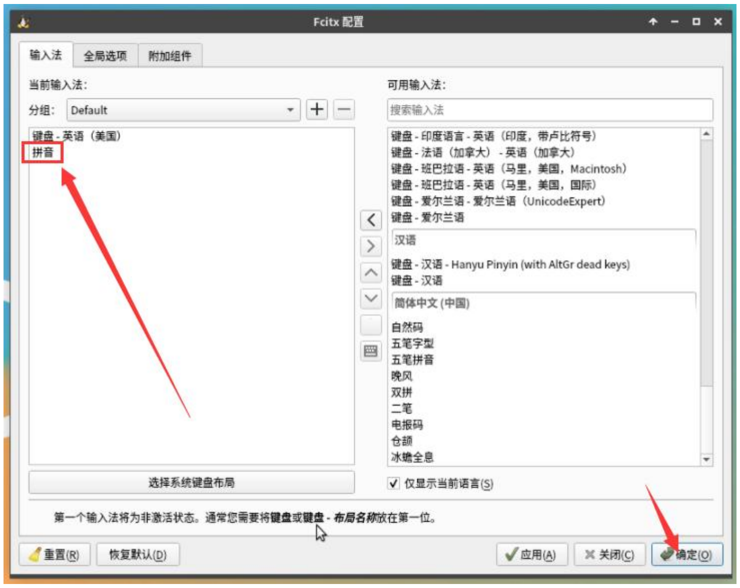
8) Then open the Fcitx5 configuration program



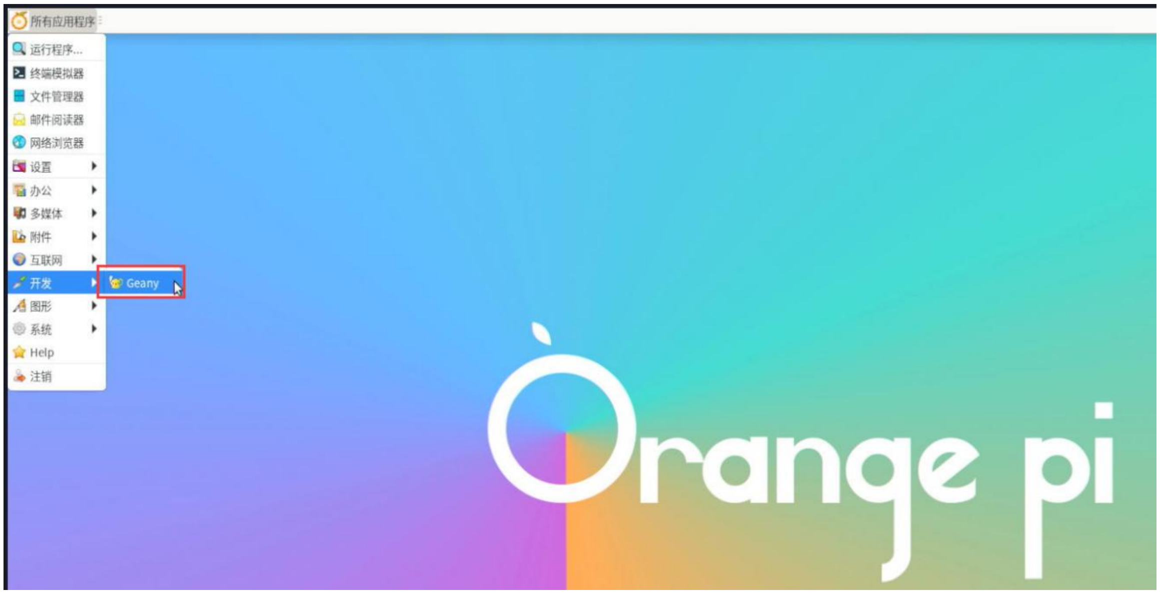
9) Then select Pinyin input method



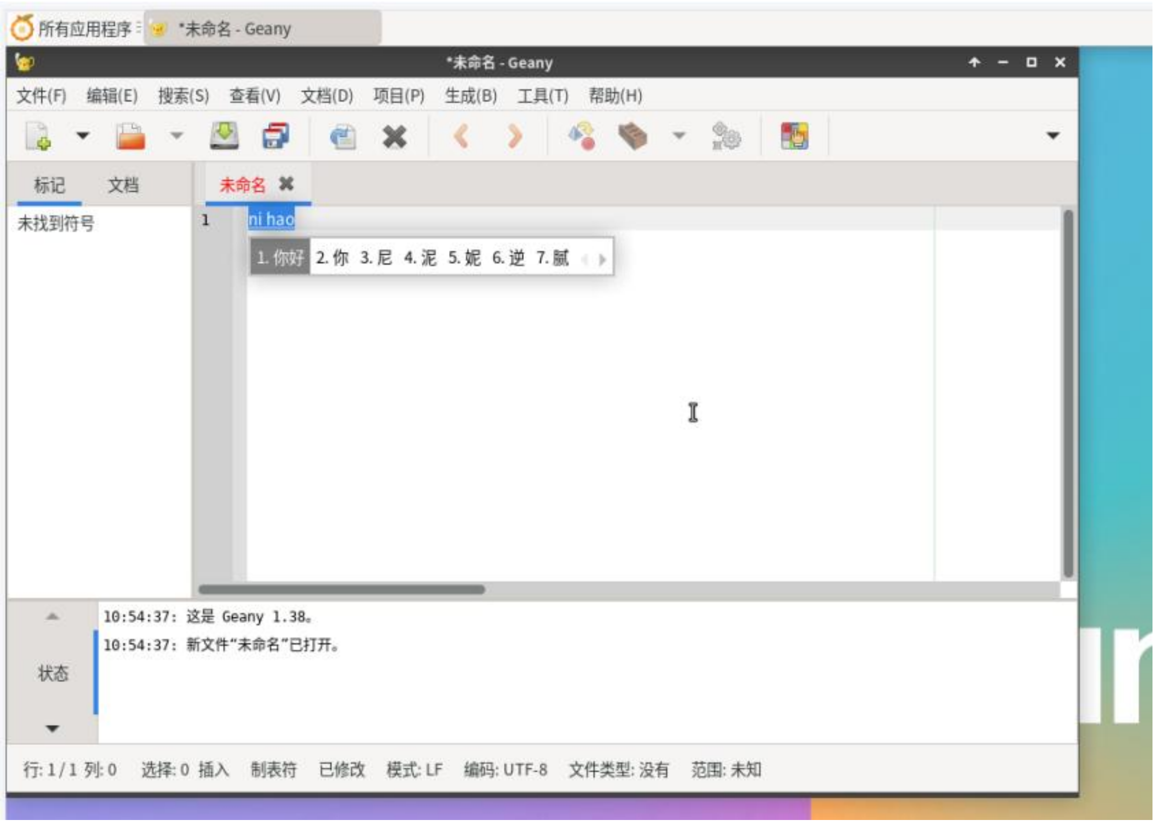
10) The interface after selection is as shown below, then click OK



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 default input method is still English. We can switch to Chinese input method by pressing **Ctrl+Space,** and then we can input Chinese.



3. 27. How to remotely log in to the Linux system desktop

The Ubuntu Gnome Wayland image does not support remote desktop login using Nomachine and VNC as described here.

3. 27. 1. Remote login using NoMachine

Please make sure that the Ubuntu or Debian system installed on the development board is a desktop version. In addition, NoMachine also provides detailed usage documentation. It is strongly recommended to read this document to familiarize yourself with the use of NoMachine. The document link is as follows:

https://knowledgebase.nomachine.com/DT10R00166

NoMachine supports Windows, Mac, Linux, iOS and Android platforms, so we can use NoMachine to remotely log in and control the Orange Pi development board on multiple devices. The following demonstrates how to remotely log in to the Linux system desktop of the Orange Pi development board through NoMachine in

Windows. For installation methods on other platforms, please refer to the official documentation of NoMachine.

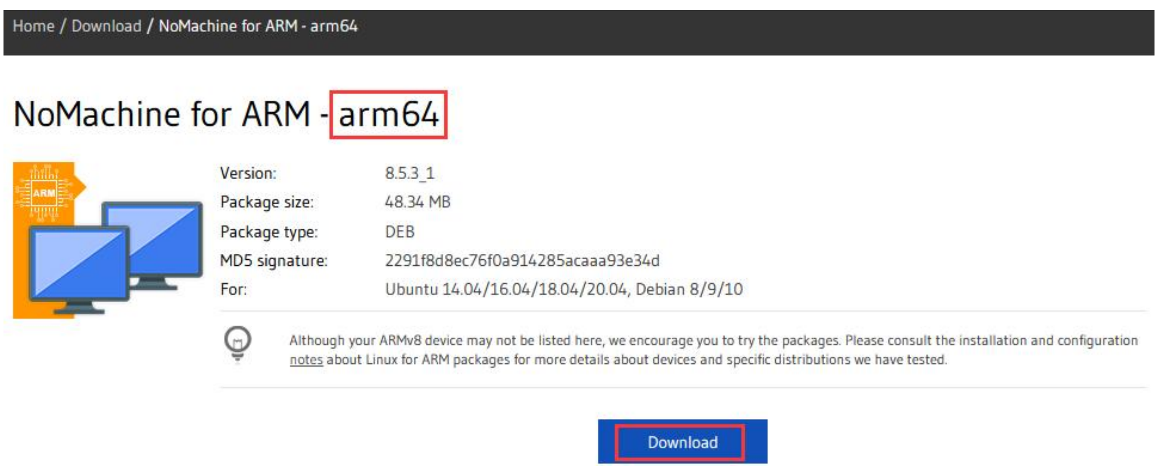
Before operation, please make sure that the Windows computer and the development board are in the same LAN and can log in to the Ubuntu or Debian system of the development board normally through SSH.

1) First download the installation package of the NoMachine software Linux **arm64** deb version, and then install it into the Linux system of the development board

a. Since RK3588 is an ARMv8 SOC, we use Ubuntu or Debian as the system, 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.

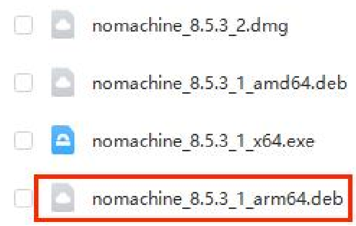
https://downloads.nomachine.com/download/?id=114&distro=ARM



b. In addition, you can also download the **NoMachine** installation package in the **official tool**

First enter the **remote login software-NoMachine** folder

Then download the arm64 version of the deb installation package



c. Then upload the downloaded **nomachine_x.x.x_x_arm64.deb** to the Linux system of the development board

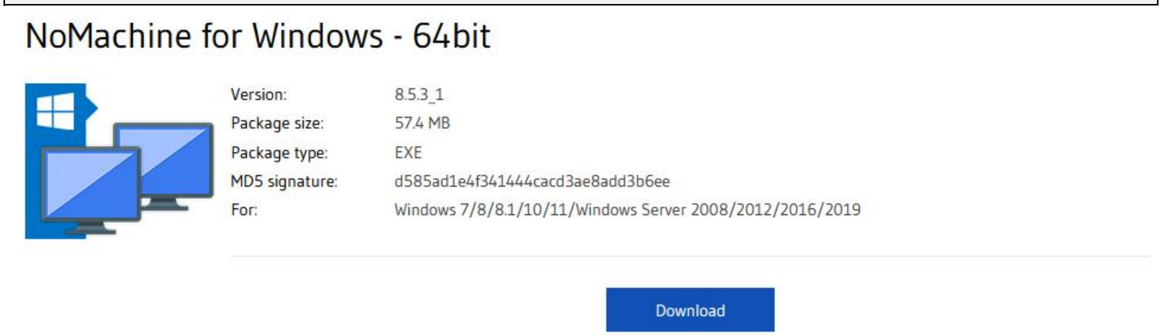
d. Then use the following command to install **NoMachine** in the Linux system of the development board

```
orangepi@orangepi:~$ sudo dpkg -i nomachine_x.x.x_x_arm64_arm64.deb
```

2) Then download the installation package of NoMachine software Windows version. The download address is as follows

Note that this download link may change.

https://downloads.nomachine.com/download/?id=9

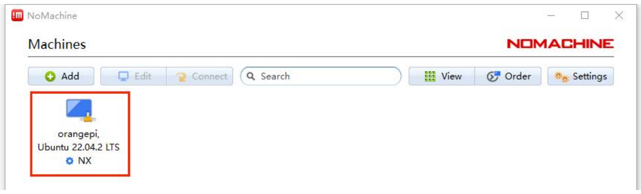


3) Then install NoMachine in Windows**. After installation, please restart your computer**

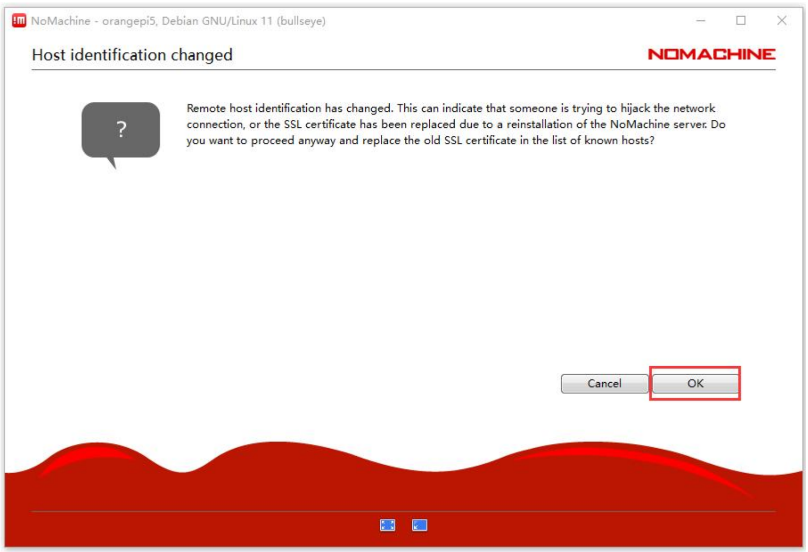
4) Then open **NoMachine** in Windows

5) After NoMachine is started, it will automatically scan other devices in the LAN that have NoMachine installed. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices. Then click the

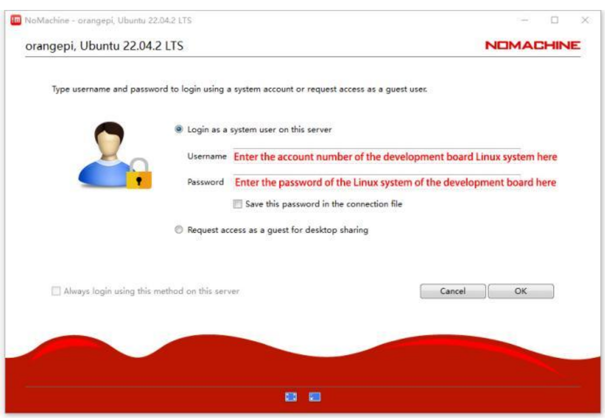
location indicated by the red box in the figure below to start logging into the Linux system desktop of the development board.



6) then click **OK**

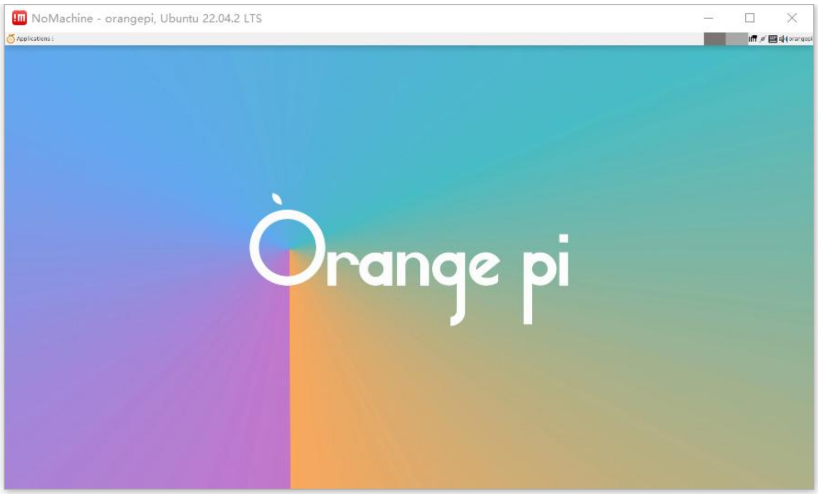


7) Then enter the user name and password of the development board Linux system in the corresponding position in the figure below, and then click **OK** to start logging in



8) Then click OK on the next interface.

9) Finally, you can see the desktop of the development board Linux system



3. 27. 2. Remote login using VNC

> **Before operation, please make sure that the Windows computer and the development board are in the same LAN and can log in to the Ubuntu or Debian system of the development board normally through SSH.**
>
> **There are many problems with testing VNC on Ubuntu 20.04, so please do not use this method.**

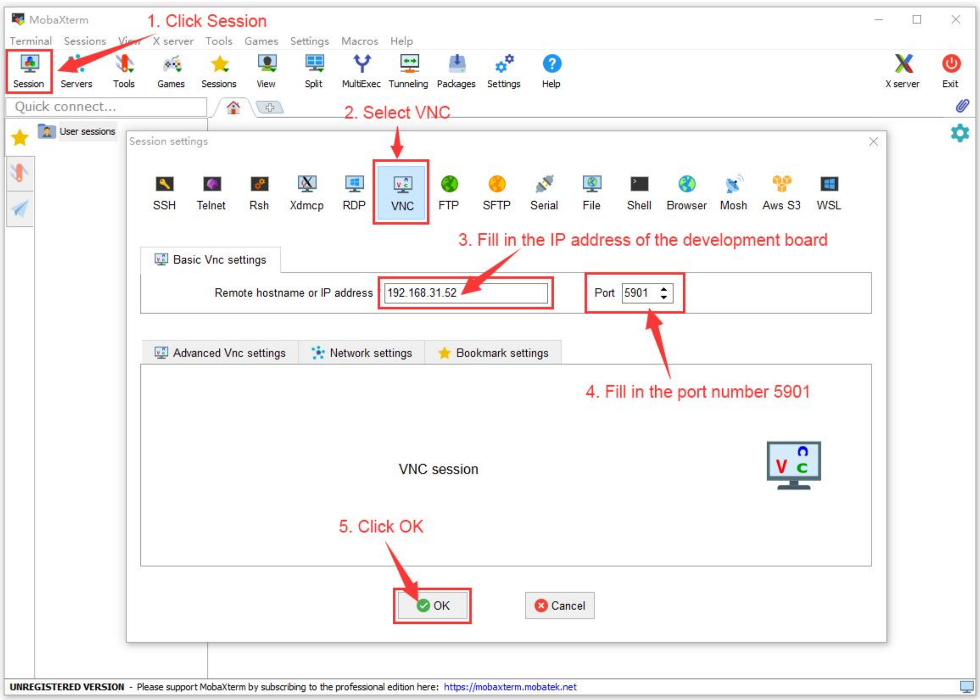
1) First run the **set_vnc.sh** script to set up vnc, **remember to add sudo permissions**

```
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 orangepi5max:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/Orangepi5max:1.log

Killing Xtightvnc process ID 3047
```

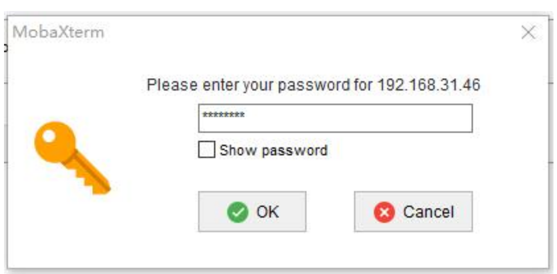
```
New 'X' desktop is orangepi5max:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi5max:1.log
```

2) The steps to use MobaXterm software to connect to the Linux system desktop of the development board are as follows:

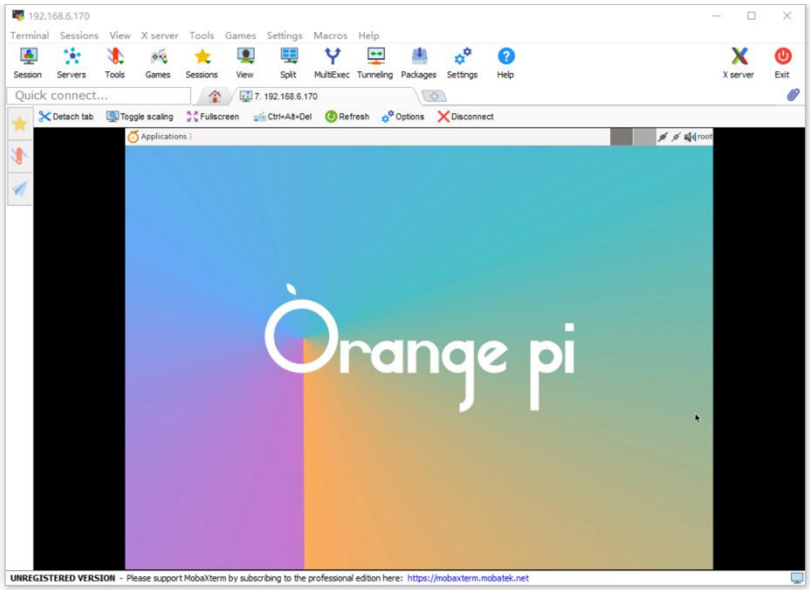
a. First click Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm



b. Then enter the VNC password set previously



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. Test 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 compile C language programs directly in the Linux system of the development board

a. gcc version is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 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 Python 3 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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bullseye does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk. The latest version in Debian Bullseye is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the Java version

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

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
```

```
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 28. 2. Debian Bookworm System

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

a. gcc version is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 12.2.0-14) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bookworm comes with 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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bookworm does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk. The latest version in Debian Bookworm is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the Java version

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

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 28. 3. Ubuntu Focal system

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

a. gcc version is as follows

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

```
gcc (Ubuntu 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 Python 3 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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Ubuntu Focal does not install Java compilation tools and runtime environment by

default

a. You can use the following command to install openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the 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 the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 28. 4. Ubuntu Jammy System

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

a. gcc version is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.2.0-19ubuntu1) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>
```

```
int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

5) Ubuntu Jammy has Python 3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.4 (main, Apr  2 2022, 09:04:19) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

6) Ubuntu Jammy does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk-18

```
orangepi@orangepi:~$ sudo apt install -y openjdk-18-jdk
```

b. After installation, you can check the Java version

```
orangepi@orangepi:~$ java --version
openjdk 18-ea 2022-03-22
OpenJDK Runtime Environment (build 18-ea+36-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 18-ea+36-Ubuntu-1, mixed mode, sharing)
```

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
```

```
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 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 Ubuntu 20.04 is **5.12.8**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/aarch64-linux-gnu
```

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

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

c. The QT version that comes with Debian11 is **5.15.2**

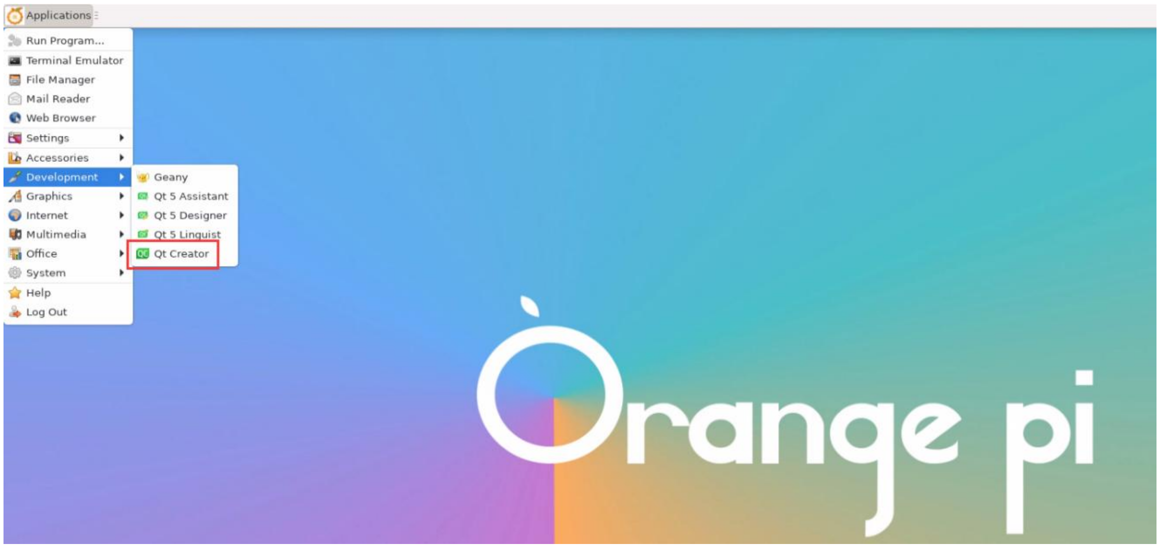
```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

d. The QT version that comes with Debian12 is **5.15.8**

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

```
......
QMake version 3.1
Using Qt version 5.15.8 in /usr/lib/aarch64-linux-gnu
```

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



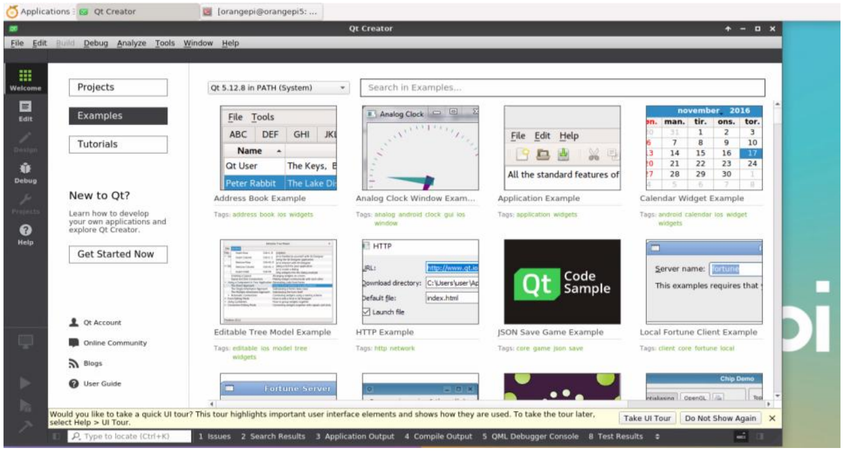
You can also use the following command to open QT Creator

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

During the startup of QT and QT applications, if the following error is prompted, please ignore it directly. This error will not affect the operation of the application.

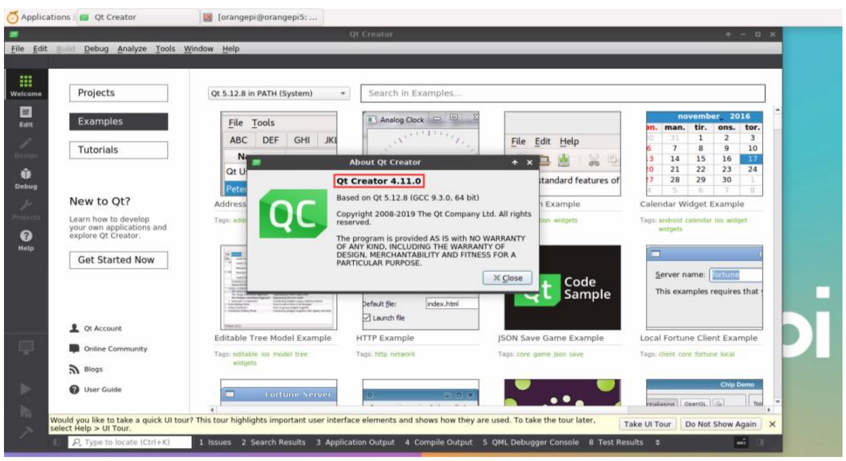
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip

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

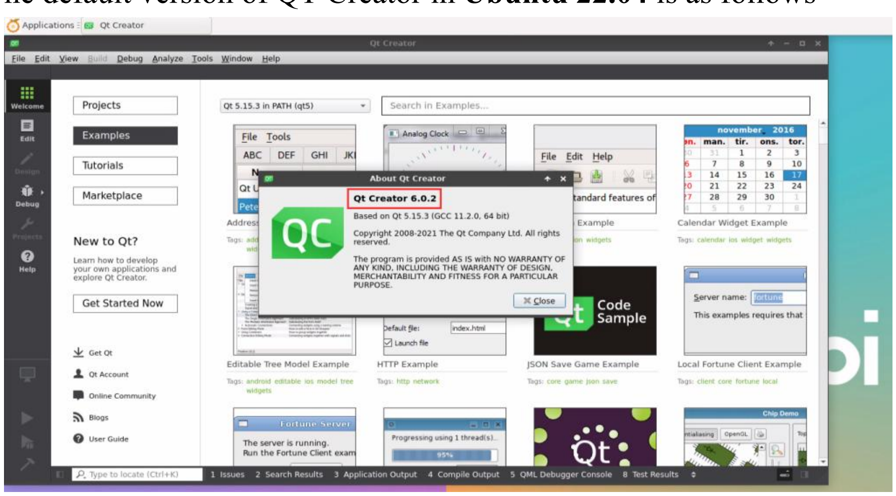


5) The version of QT Creator is as follows

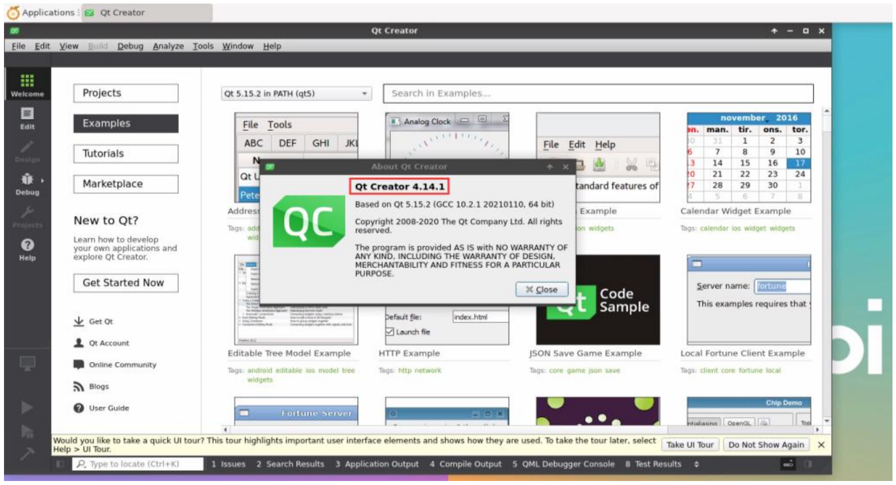
a. The default version of QT Creator in **Ubuntu 20.04** is as follows



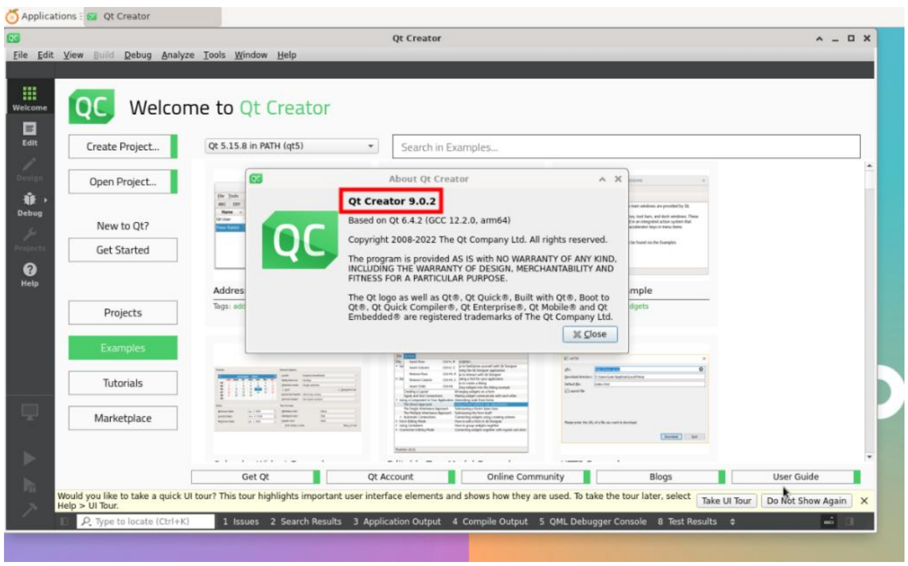
b. The default version of QT Creator in **Ubuntu 22.04** is as follows



c. The default version of QT Creator in **Debian 11** is as follows

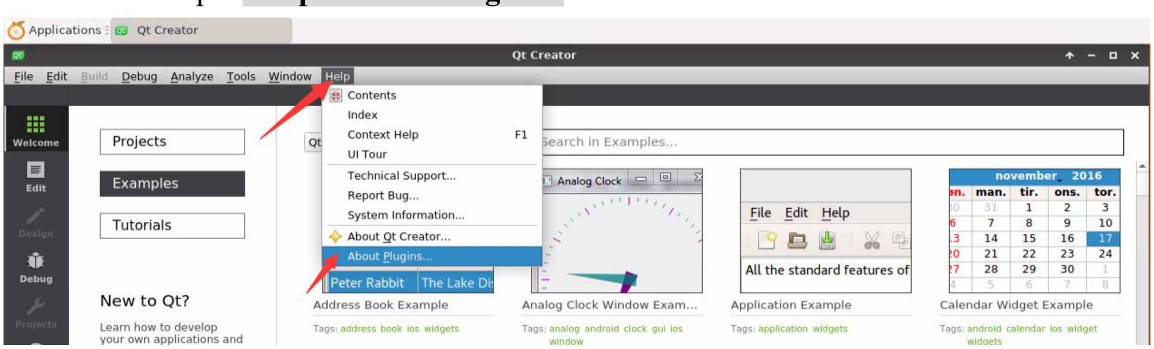


d. The default version of QT Creator in **Debian 12** is as follows

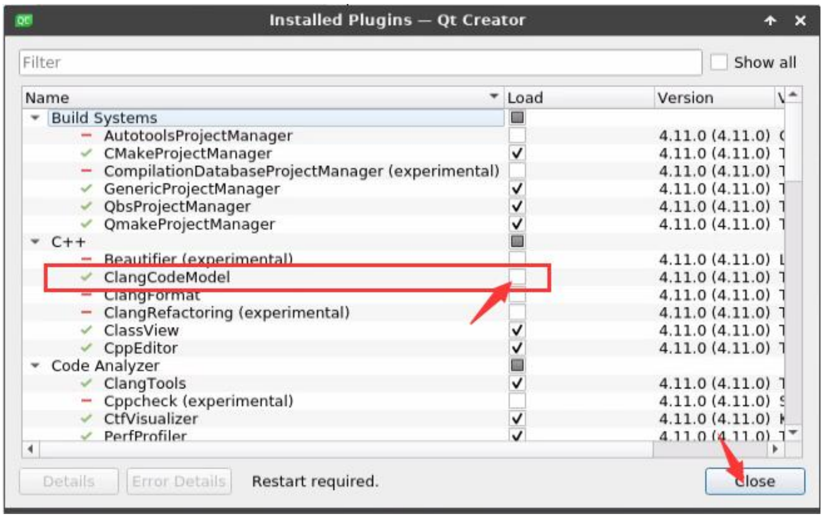


6) Then set up QT

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



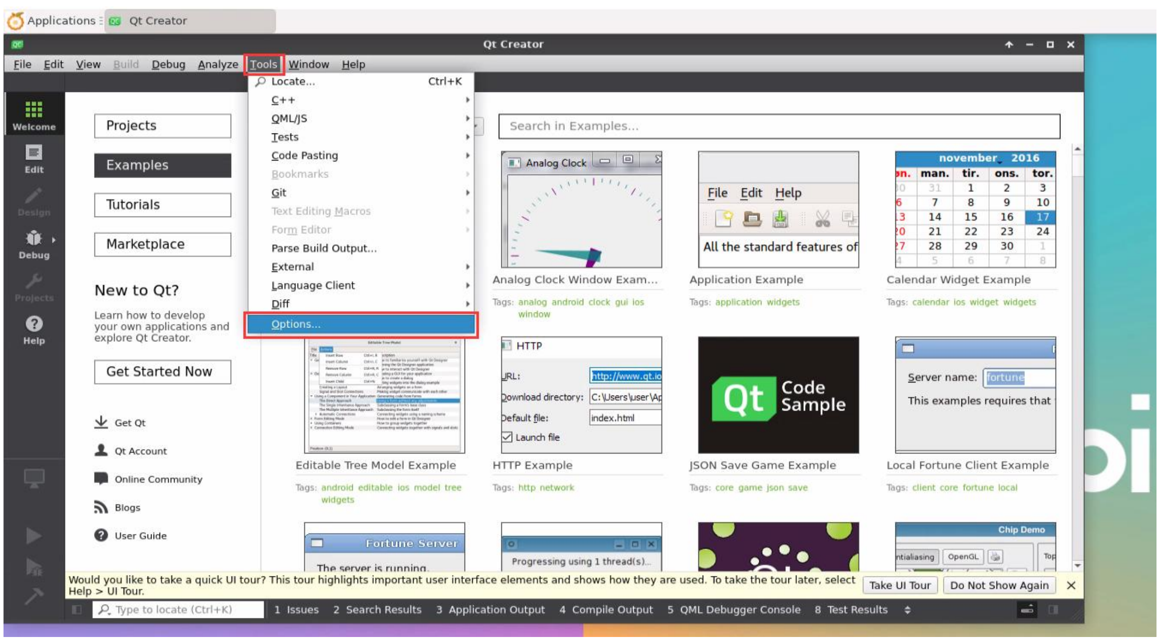
b. Then remove the check mark of **ClangCodeModel**

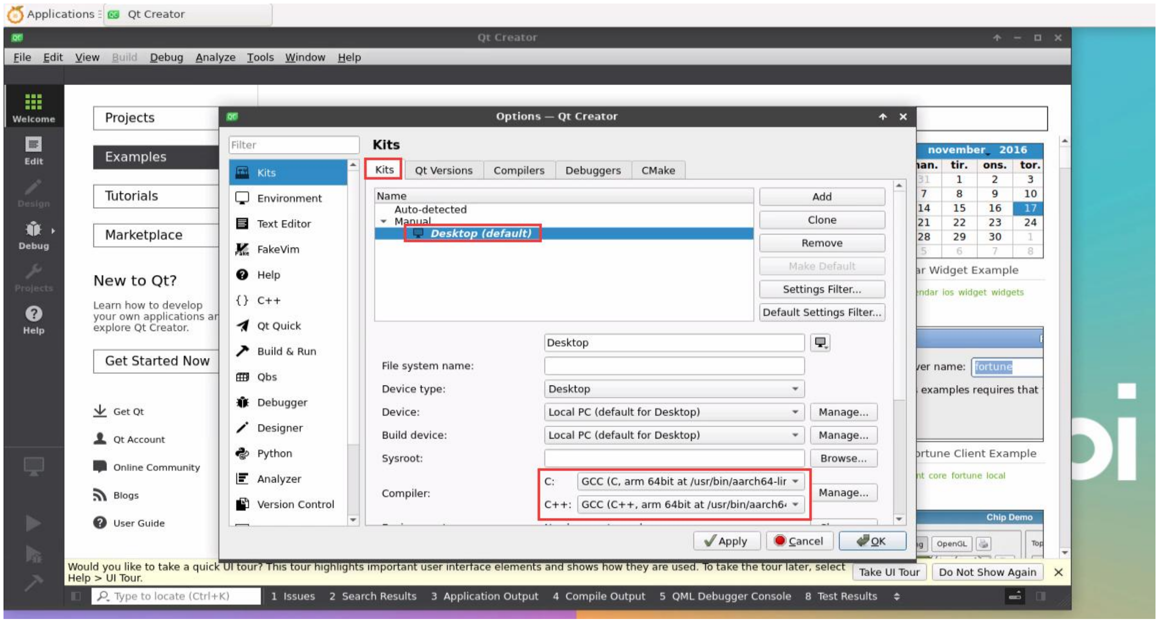


c. **After setting, you need to restart QT Creator**

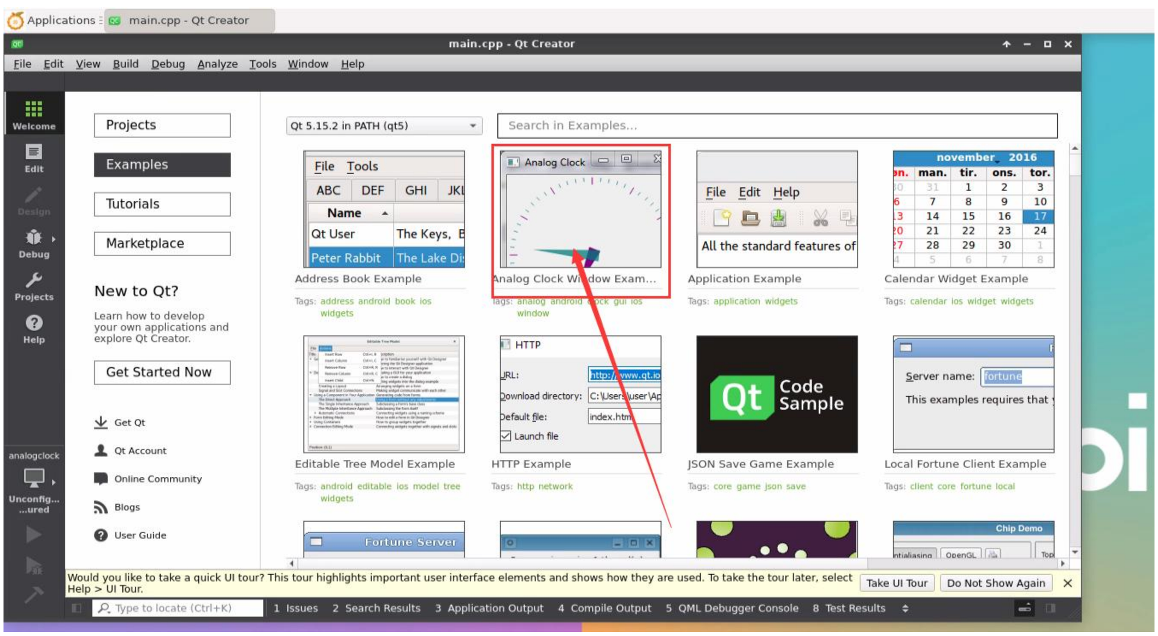
d. Then make sure that QT Creator uses the GCC compiler. If it defaults to Clang, change it to GCC

For Debian 12, please skip this step.

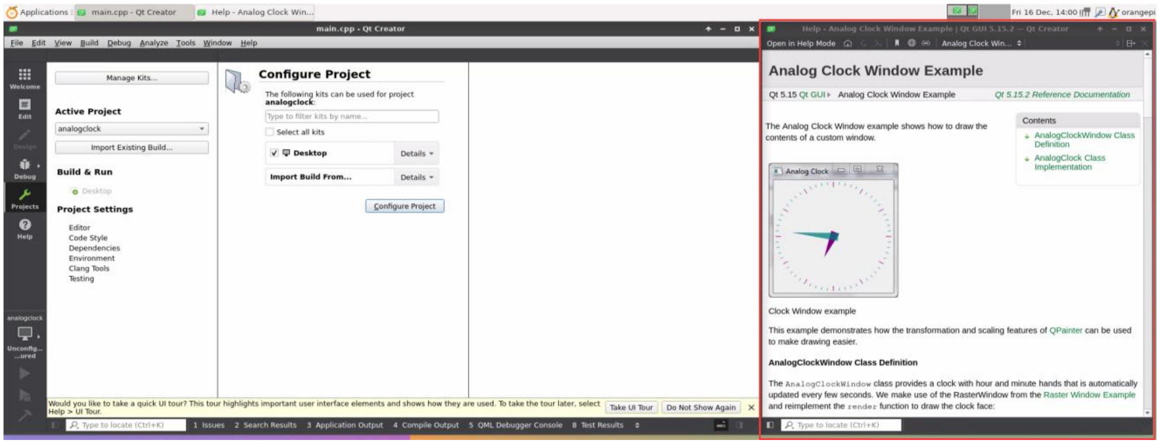




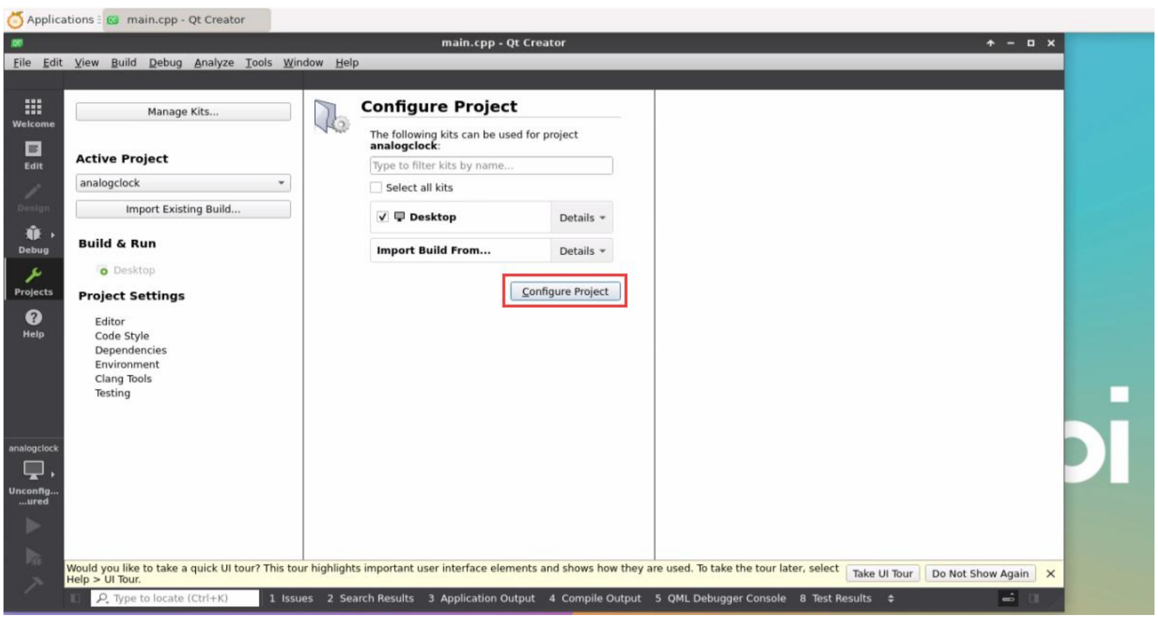
7) Then you can open a sample code



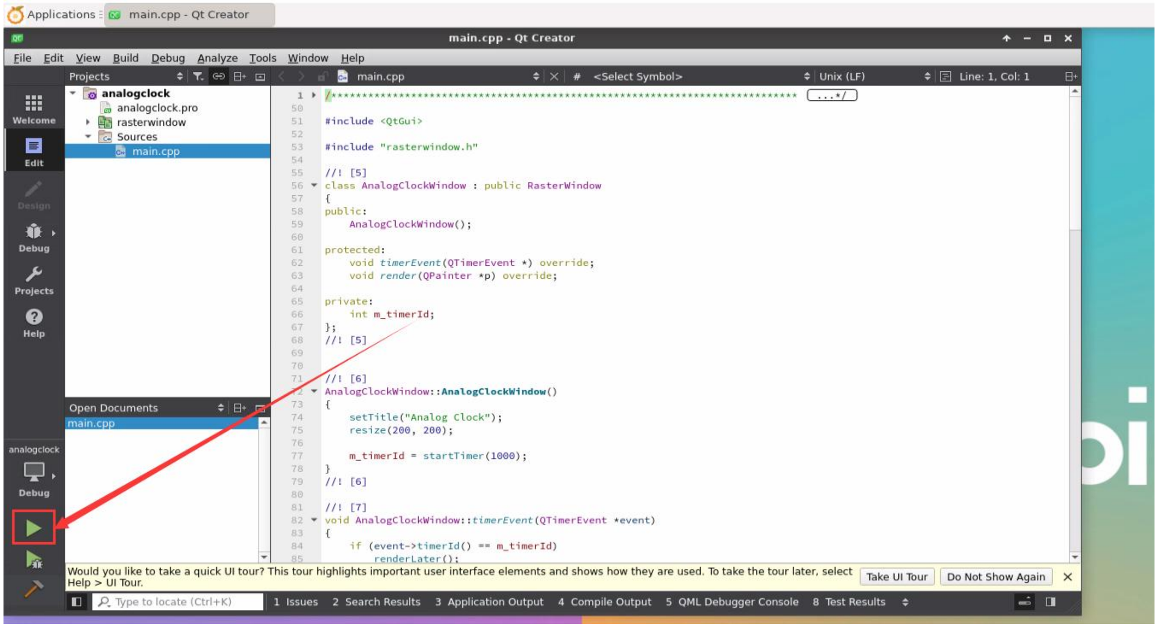
8) Clicking on the sample code will automatically open the corresponding documentation. Please read the instructions carefully.



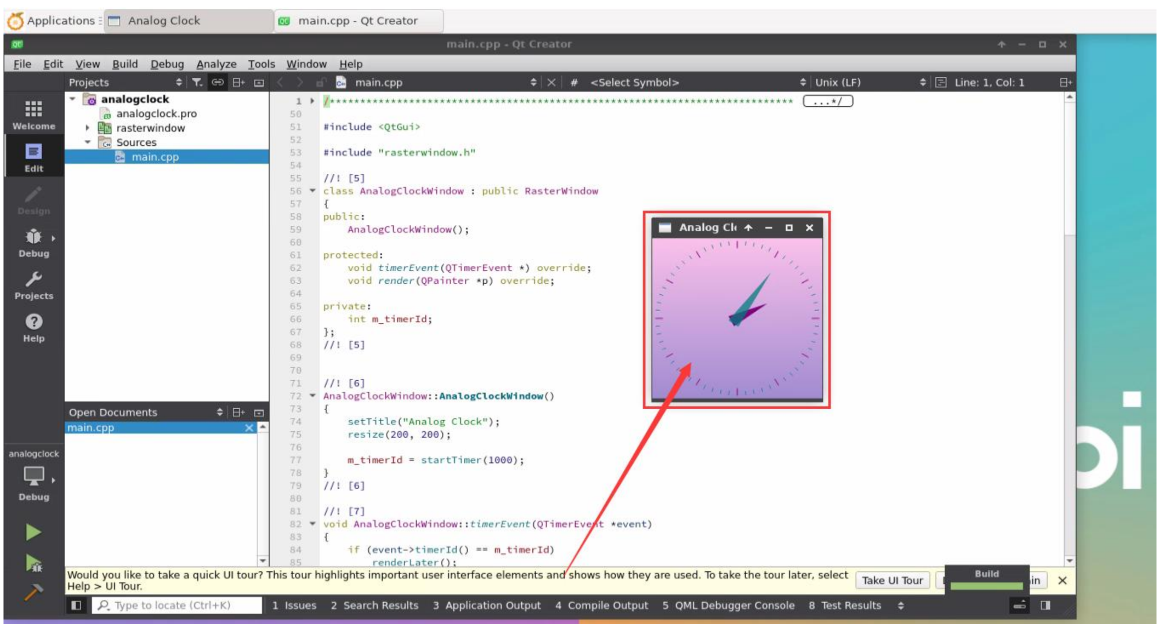
9) Then click **Configure Project**



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



11) After waiting for a while, the interface shown in the figure below will pop up, which means that QT can compile and run normally



12) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu

https://download.qt.io/archive/qtcreator

https://download.qt.io/archive/qt

3. 30. ROS installation method

3. 30. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The currently active versions of ROS 1 are as follows. The recommended version is **Noetic Ninjemys**

Active ROS 1 distributions

Recommended

Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

```
http://docs.ros.org
https://wiki.ros.org/Distributions
```

2) The official installation document link for **ROS 1 Noetic Ninjemys** is as follows:

```
http://wiki.ros.org/noetic/Installation/Ubuntu
```

3) The official installation document of ROS **Noetic Ninjemys** recommends using Ubuntu 20.04, so make sure that the system used by the development board is the **Ubuntu 20.04 desktop system**

http://wiki.ros.org/noetic/Installation

4) Then use the following script to install ros1

```
orangepi@orangepi5max:~$ install_ros.sh ros1
```

5) Before using the ROS tool, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code.

Note that 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 command. However, this method cannot guarantee that GitHub can be accessed normally every time. If the following error is prompted after install_ros.sh installs ros1, please find other ways to enable the Linux system of the development board to access GitHub normally, and then manually run the following command.

https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
ERROR: error loading sources list:
The read operation timed out

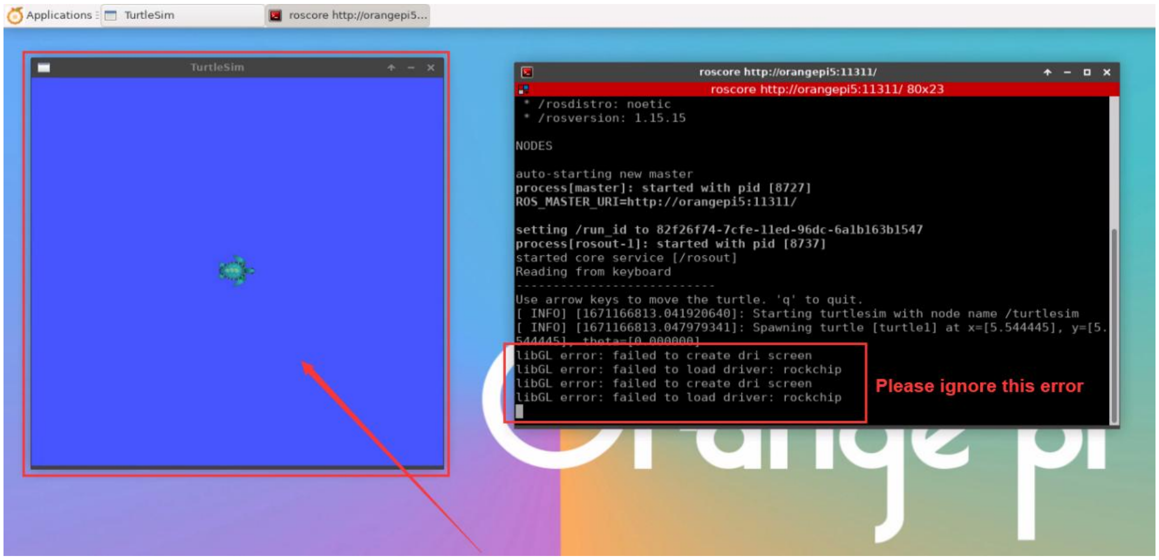
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run
```

```
            rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

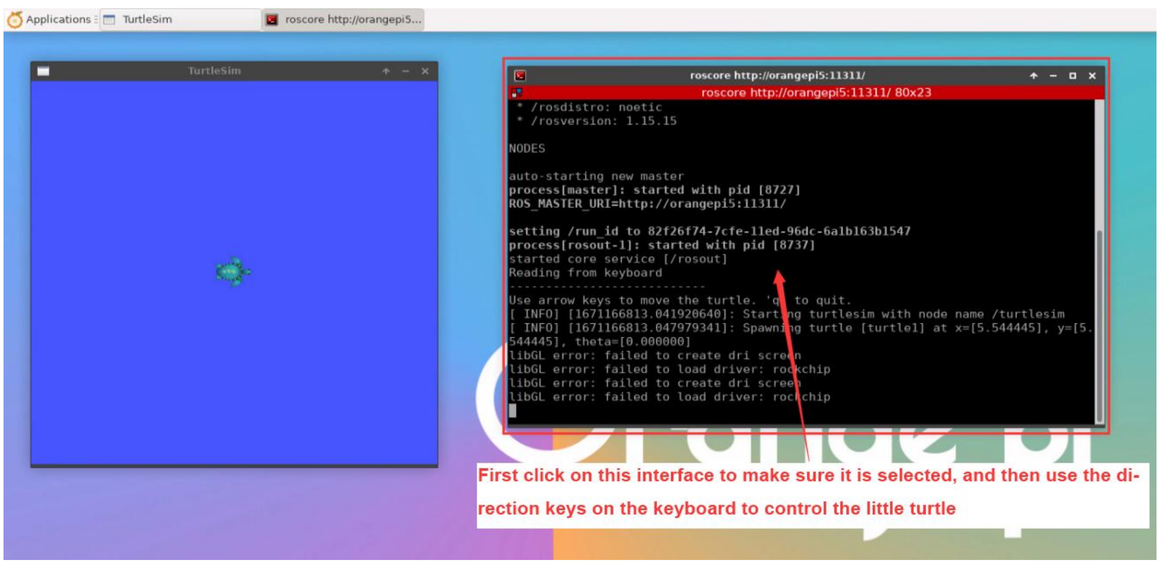
6) Then open a command line terminal window on the **desktop**, and use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally

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

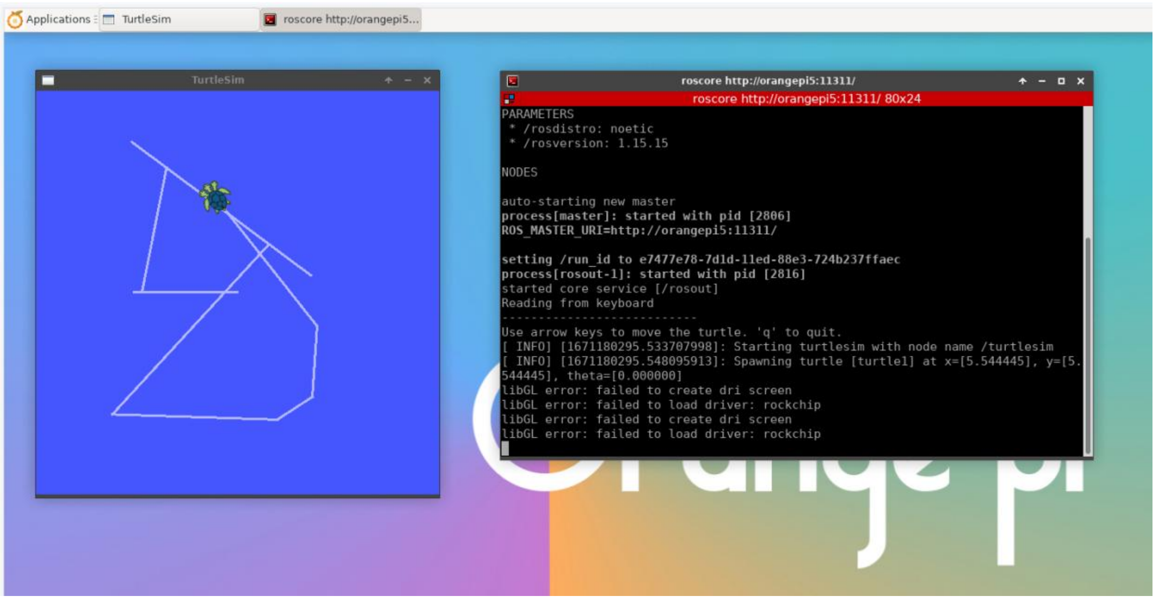
7) After running the **test_ros.sh** script, a small turtle will pop up as shown in the figure below



8) Then please keep the terminal window you just opened on top

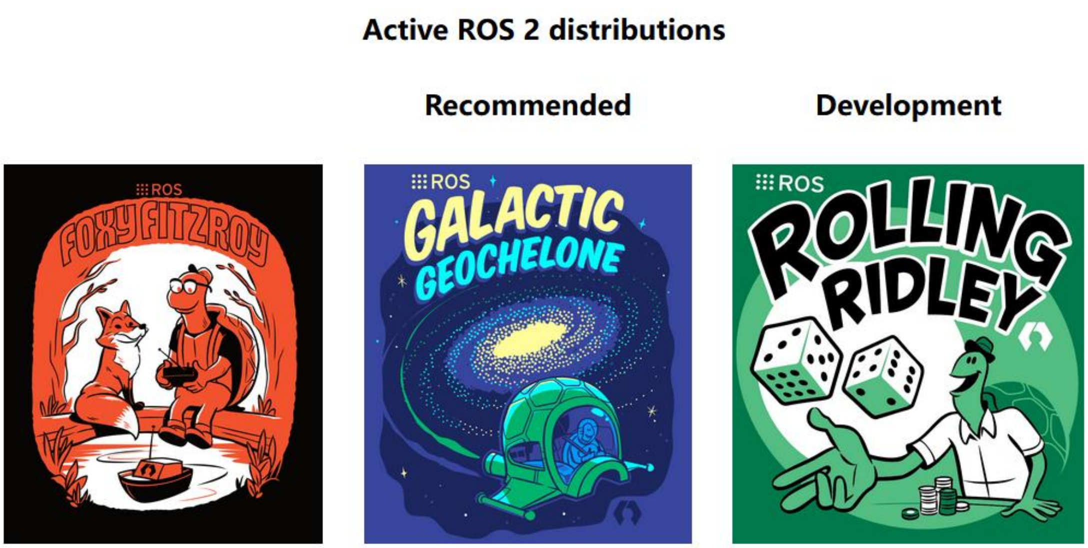


9) At this time, press the direction keys on the keyboard to control the turtle to move up, down, left and right.



3. 30. 2. How to install ROS 2 Galactic on Ubuntu 20.04

1) The currently active versions of ROS 2 are 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 official installation document link for 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) The official installation document of ROS 2 **Galactic Geochelone** recommends using Ubuntu 20.04 for Ubuntu Linux, so make sure that the system used by the development board is the **Ubuntu 20.04 desktop system**. There are several ways to install ROS 2. The following demonstrates how to install ROS 2 **Galactic Geochelone** using **Debian packages**.

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

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

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

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

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
```

```
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

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

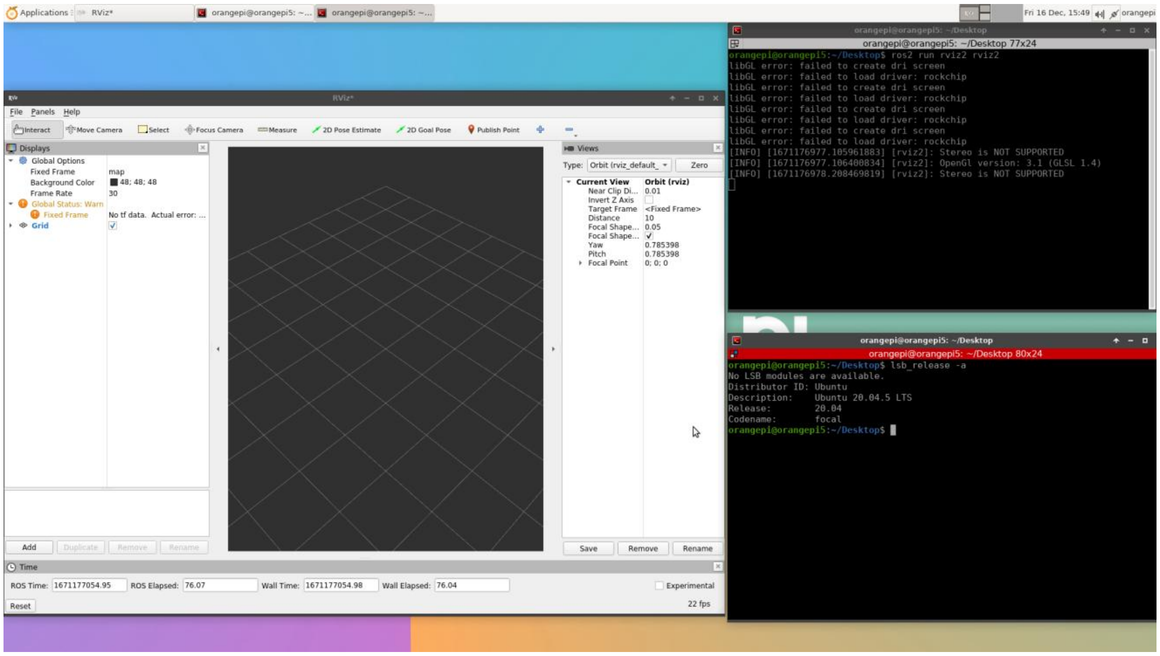
6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally.

```
orangepi@orangepi5max:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
```

orangepi@orangepi:~$ **ros2 run rviz2 rviz2**



8) For the usage of ROS, please refer to the ROS 2 documentation

http://docs.ros.org/en/galactic/Tutorials.html

3. 30. 3. How to install ROS 2 Humble on Ubuntu 22.04

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

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

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

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

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action     Various action related sub-commands
  bag        Various rosbag related sub-commands
  component  Various component related sub-commands
```

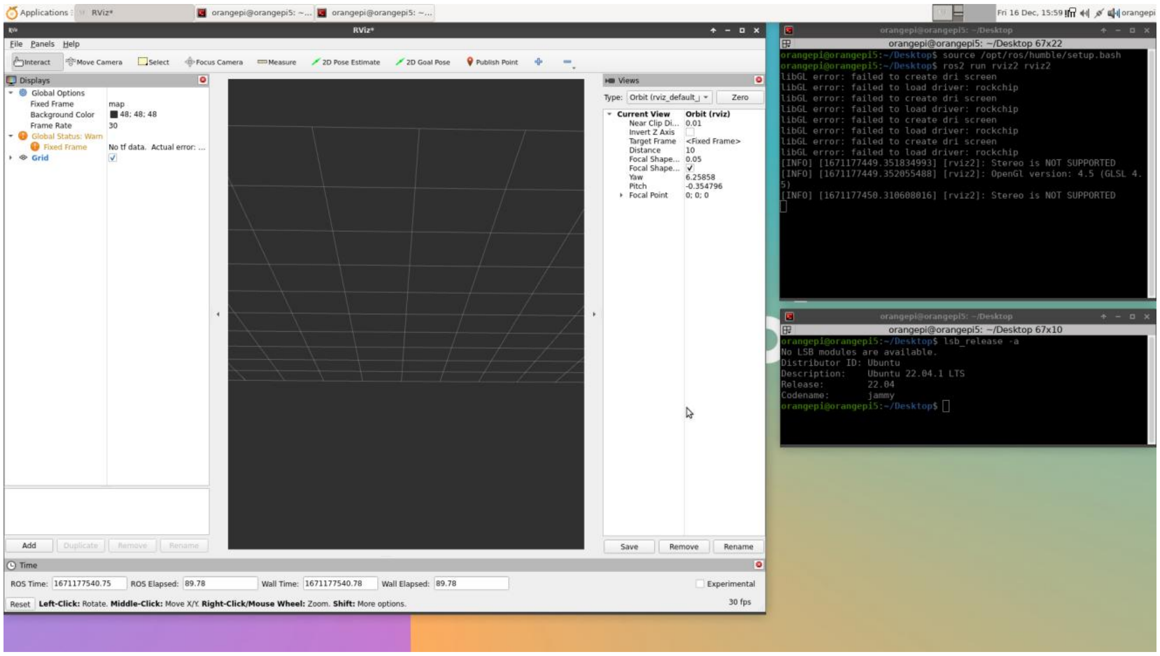
```
daemon       Various daemon related sub-commands
doctor       Check ROS setup and other potential issues
interface    Show information about ROS interfaces
launch       Run a launch file
lifecycle    Various lifecycle related sub-commands
multicast    Various multicast related sub-commands
node         Various node related sub-commands
param        Various param related sub-commands
pkg          Various package related sub-commands
run          Run a package specific executable
security     Various security related sub-commands
service      Various service related sub-commands
topic        Various topic related sub-commands
wtf          Use `wtf` as alias to `doctor`

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

3) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally.

```
orangepi@orangepi5max:~$ 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 a deb package of kernel header files by default, which is stored in **/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 kernel header file deb file package.

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 files are 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 for the hello kernel module as follows:

```
orangepi@orangepi:~$ vim hello.c
#include <linux/init.h>
#include <linux/module.h>

static int hello_init(void)
{
        printk("Hello Orange Pi -- init\n");

        return 0;
}
static void hello_exit(void)
{
        printk("Hello Orange Pi -- exit\n");

        return;
}

module_init(hello_init);
module_exit(hello_exit);

MODULE_LICENSE("GPL");
```

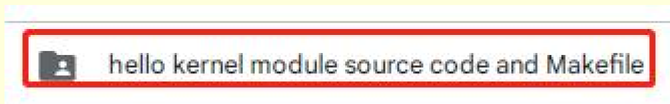
b. Then write the Makefile file to compile the hello kernel module as follows:

```
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 there is any problem when 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.ko** kernel module will be generated

```
orangepi@orangepi:~$ ls *.ko
hello.ko
```

e. Use the **insmod** command to insert the **hello.ko** kernel module into the kernel

```
orangepi@orangepi:~$ sudo insmod hello.ko
```

f. Then use the **demsg** command to view the output of the **hello.ko** kernel module. If you can see the 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. How to use the 10.1-inch MIPI LCD screen

3. 32. 1. How to assemble a 10.1-inch MIPI screen

1) First prepare the necessary accessories

a. 10.1 inch MIPI LCD display + touch screen

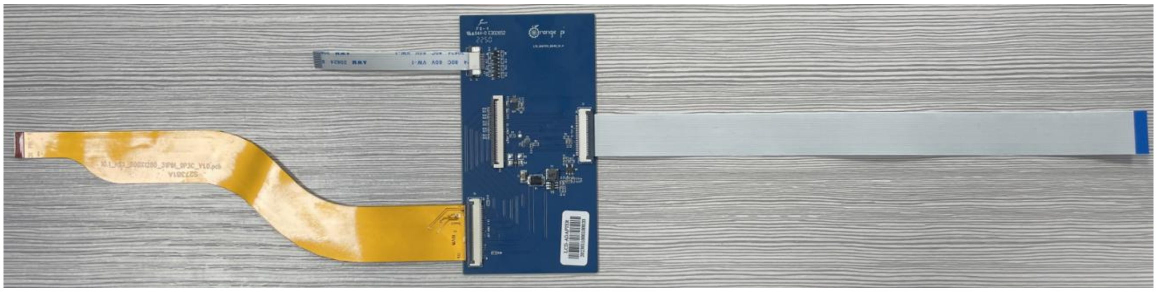
b. Screen adapter board + 31pin to 40pin cable

c. 30pin MIPI cable

d. 12pin touch screen cable

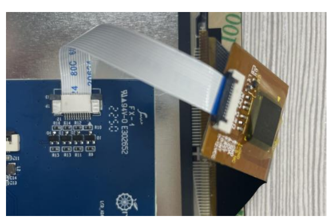
2) Connect the 12-pin touch screen cable, 31-pin to 40-pin cable, and 30-pin MIPI cable to the screen adapter board as shown below. **Note that the blue insulation side of the touch screen cable should face down**, and the insulation sides of the other two cables

should face up. If connected incorrectly, it will cause no display or inability to touch.

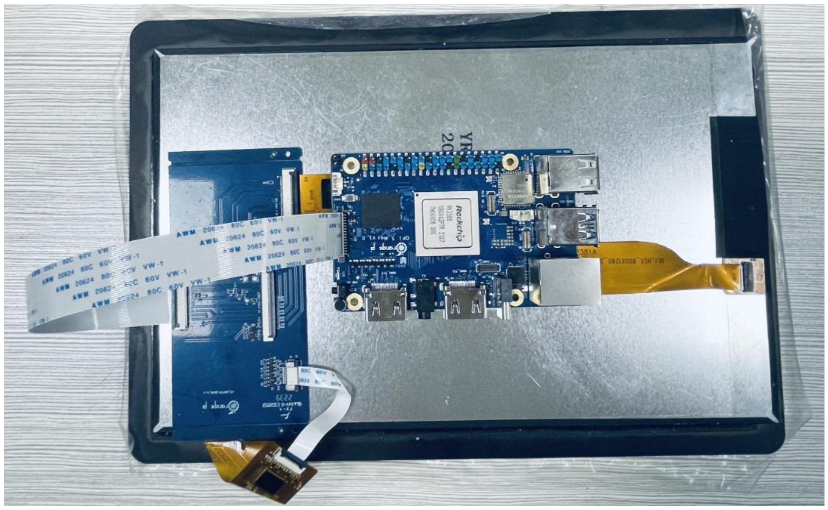


3) Place the adapter board with the connected cable on the MIPI LCD screen as shown below, and connect the MIPI LCD screen and the adapter board via a 31pin to 40pin cable.

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



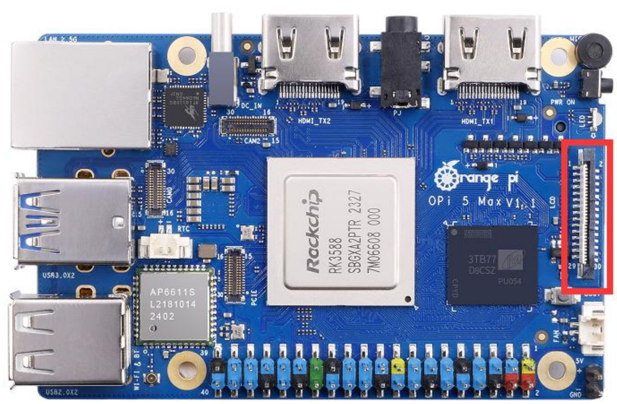
5) Finally, connect it to the LCD interface of the development board through the 30pin MIPI cable



3. 32. 2. How to open the 10.1-inch MIPI LCD screen configuration

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

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

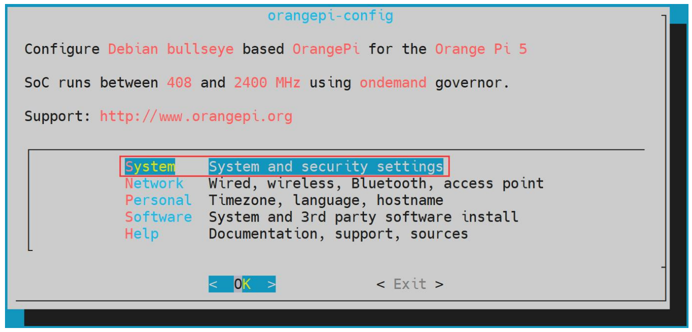


3) The steps to open the mipi lcd configuration are as follows：

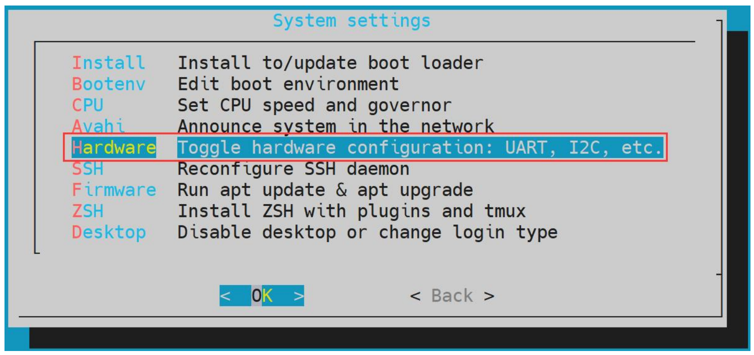
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

b. Then select **System**



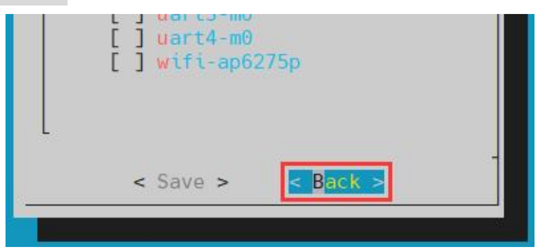
c. Then select **Hardware**



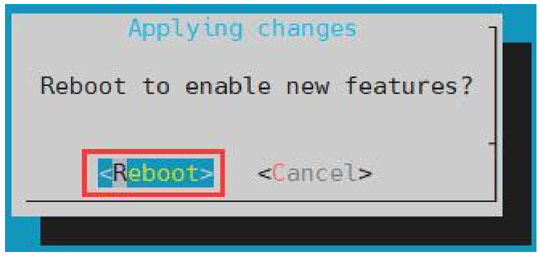
d. Then use the arrow keys on the keyboard to locate **opi5max-lcd**, and then use the **spacebar** to select

e. Then select **<Save>**

f. Then select **<Back>**



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



The above settings will eventually add overlays=opi5max-lcd to /boot/orangepiEnv.txt. You can check it after setting it. If this line does not exist, then there is a problem 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 add the line overlays=opi5max-lcd.

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "lcd"**
overlays=opi5max-lcd #Example Configuration

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

3. 32. 3. How to rotate the display direction of the server version image

1) Add **extraargs=fbcon=rotate:direction to rotate** in **/boot/orangepiEnv.txt** to set the display direction of the server version of Linux system. The number after **fbcon=rotate:** can be set to:

a. 0: Normal screen (portrait by default)

b. 1: Rotate 90 degrees clockwise
c. 2: Flip 180 degrees
d. 3: Rotate 270 degrees clockwise

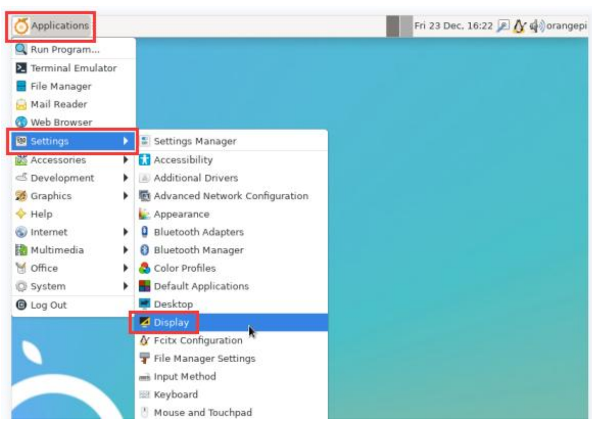
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=lcd1
extraargs=cma=64M fbcon=rotate:3
```

Note that if there is a line of extraargs=cma=64M in /boot/orangepiEnv.txt by default, you can add the line fbcon=rotate:3 after extraargs=cma=64M (separated by a space).

2) Then **restart** the Linux system and you will see that the direction of the LCD screen display has rotated

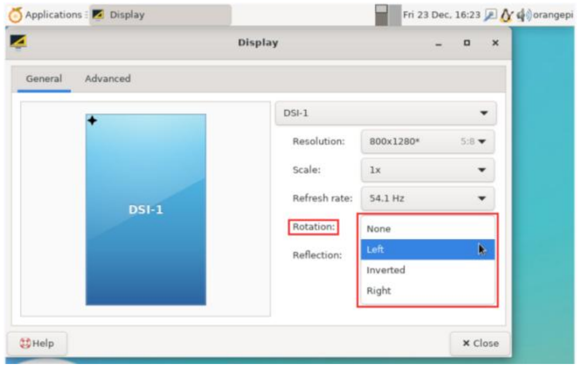
3. 32. 4. Desktop version mirroring rotation display and touch direction method

1) First open the **Display** settings in the Linux system

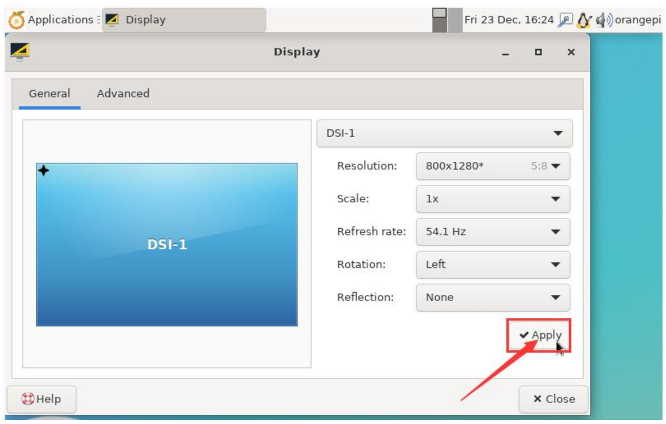


2) Then select the direction you want to rotate in **Rotation**

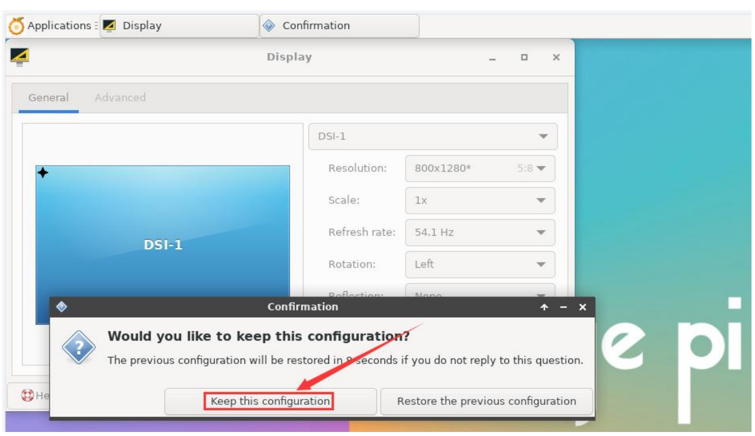
a. **None**: No rotation
b. **Left**: Rotate 90 degrees left
c. **Inverted**: Flip upside down, equivalent to rotating 180 degrees
d. **Right**: Rotate right 90 degrees



3) Then click **Apply**



4) Then select **Keep this configuration**



5) The screen display has been rotated, and you can close the **Display** program.

6) The above steps will only select the display direction, but will not rotate the touch direction. You can use the **set_lcd_rotate.sh** script to rotate the touch direction. After setting this script, 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 left

```
orangepi@orangepi:~$ set_lcd_rotate.sh left
```

c. **Inverted**: Flip upside down, equivalent to rotating 180 degrees

```
orangepi@orangepi:~$ set_lcd_rotate.sh inverted
```

d. **Right**: Rotate right 90 degrees

```
orangepi@orangepi:~$ set_lcd_rotate.sh right
```

set_lcd_rotate.sh The script mainly does four things:

1. **Rotate the direction of the framebuffer display**
2. **Rotate the touch direction**
3. **Turn off power logo**
4. **Restart the system**

Rotating the touch direction is achieved by adding Option "TransformationMatrix" "x x x x x x x x x" to /usr/share/X11/xorg.conf.d/40-libinput.conf, where "x x x x x x x x x" has different configurations for different directions.

7) Touch to rotate the reference

```
https://wiki.ubuntu.com/X/InputCoordinateTransformation
```

3. 33. Instructions for using the power on/off logo

1) The power on/off logo is only displayed in the desktop version of the system by default.

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the power on/off 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 image 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
```

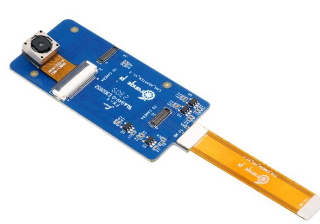
3. 34. Test methods for OV13850 and OV13855 MIPI cameras

Please note that in Linux systems, in order to ensure that the 3A service can run normally and obtain normal camera images, the Docker service needs to be disabled. If the Docker service is not disabled, the image captured by the camera will not contain the 3A effect and will appear as a dark image. The method to disable the Docker service is as follows:

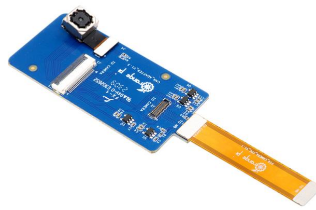
```
orangepi@orangepi:~$ sudo systemctl disable docker.socket docker.service containerd.service
orangepi@orangepi:~$ sudo reboot
```

Currently the development board supports two MIPI cameras, OV13850 and OV13855. The specific pictures are shown below:

a. 13MP OV13850 camera with MIPI interface



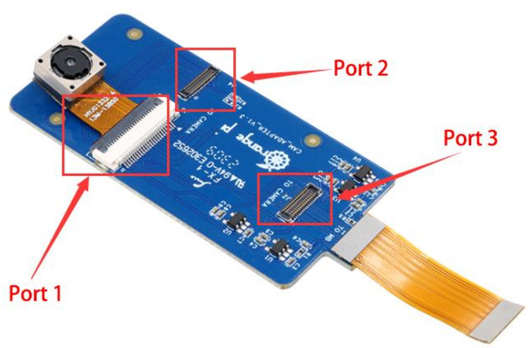
b. 13MP OV13855 camera with MIPI interface



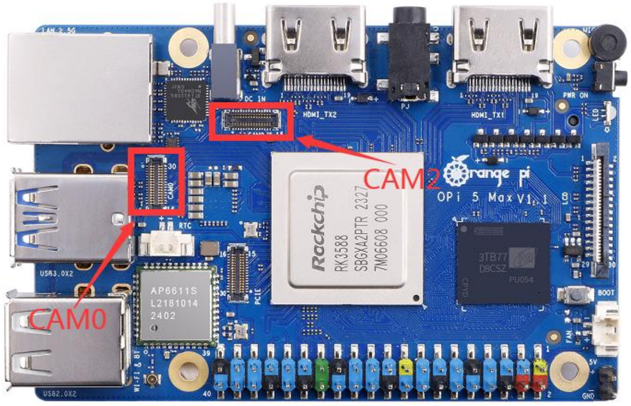
The adapter board and FPC cable used by OV13850 and OV13855 cameras are the same, but the two cameras are connected to the adapter board in different positions. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked with **TO MB** needs to be plugged into the camera interface of the development board, and the end marked with **TO CAMERA** needs to be plugged into the camera adapter board.

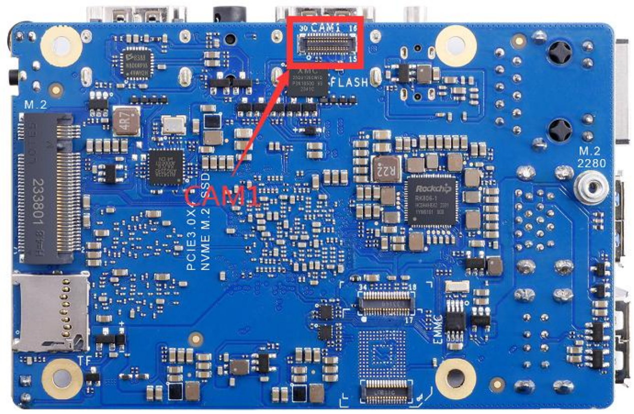
There are 3 camera interfaces on the camera adapter board. Only one camera can be connected at a time, as shown in the following figure：

a. **Interface 1 is connected to the OV13850 camera**
b. **Interface 2 is connected to the OV13855 camera**
c. Interface 3 is not used, just ignore it



There are three camera interfaces on the Orange Pi 5 Max development board. We define the positions of Cam0, Cam1, and Cam2 as shown in the following figure.：

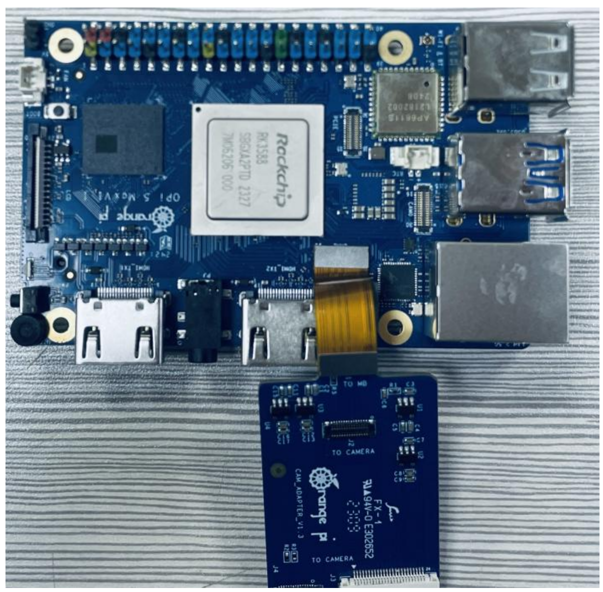




The method of inserting the camera into the Cam0 interface of the development board is as follows：

The method of inserting the camera into the Cam1 interface of the development board is as follows：

The method of inserting the camera into the Cam2 interface of the development board is as follows：

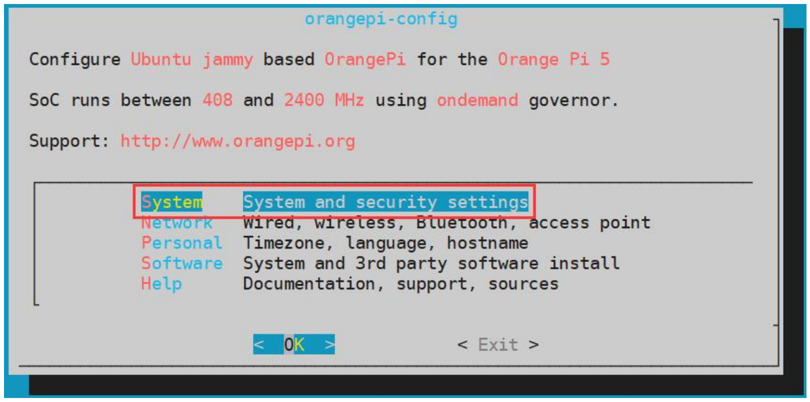


After connecting the camera to the development board, we can use the following method to test the camera：

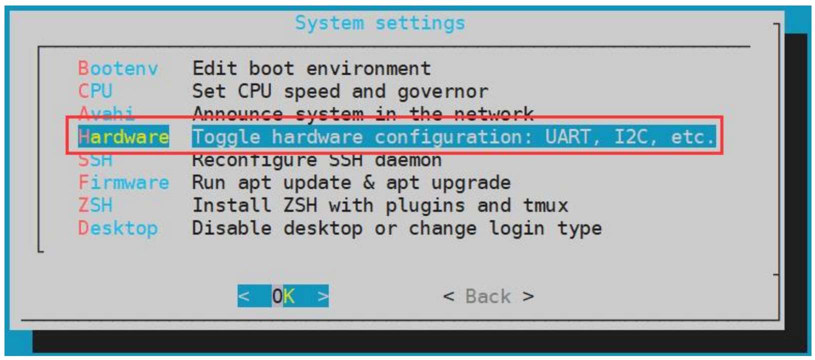
a. First run **orangepi-config**. Ordinary users should remember to add **sudo** permissions.

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

b. Then select **System**



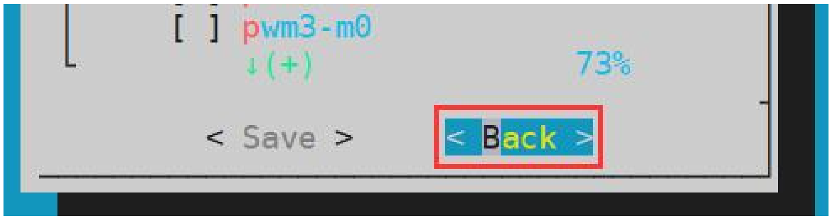
c. Then select **Hardware**



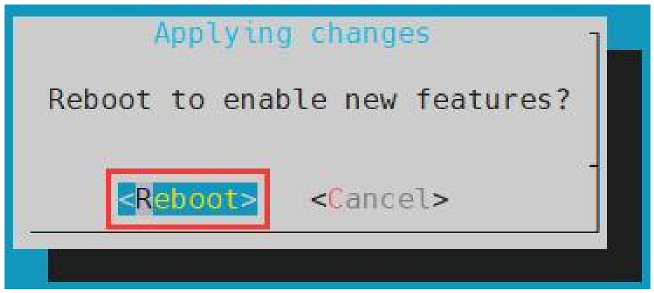
d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the space bar to select the camera you want to open. **opi5max-cam0** means using the ov13850 or ov13855 camera in the Cam0 interface of the development board, **opi5max-cam1** means using the ov13850 or ov13855 camera in the Cam1 interface of the development board, and **opi5max-cam2** means using the ov13850 or ov13855 camera in the Cam2 interface of the development board.

e. Then select **<Save>**

f. Then select **<Back>**



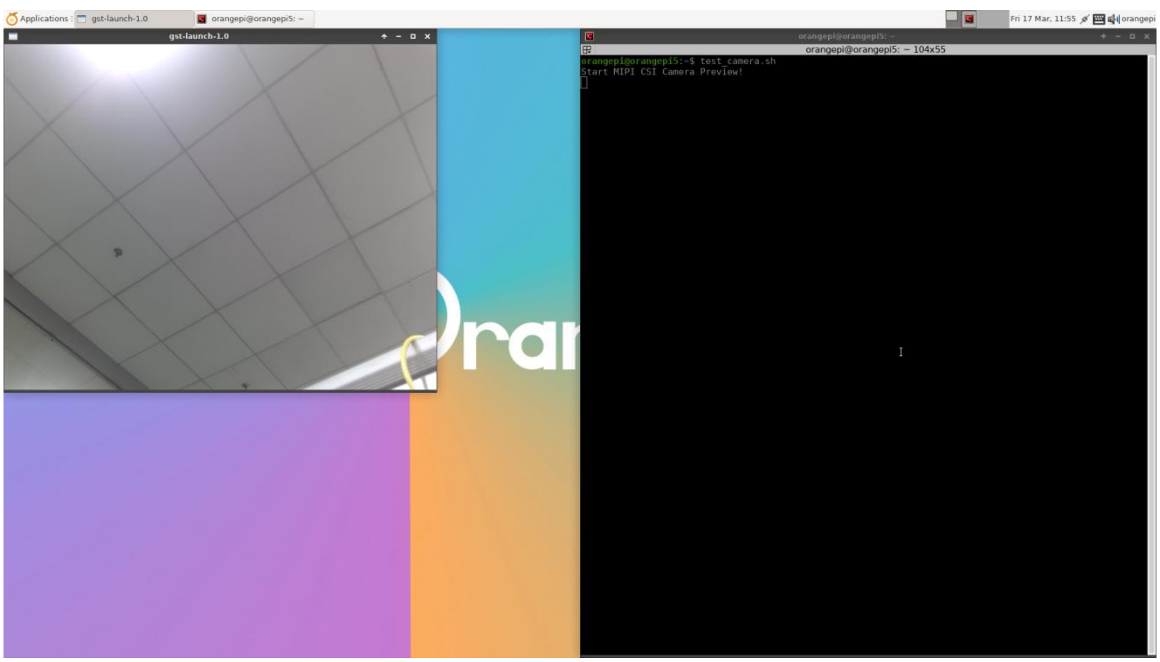
g. Then select **<Reboot>** Restart the system to make the configuration take effect



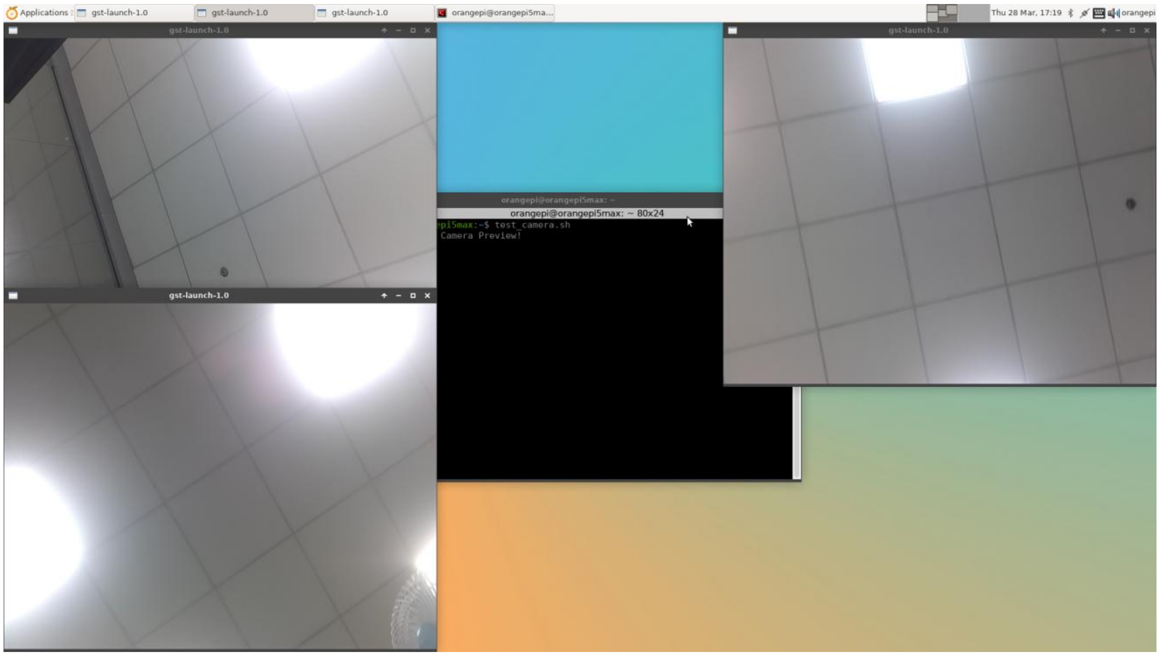
h. Then open a terminal in the desktop system and run the following script

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

i. Then you can see the camera preview



In addition to a single camera, we can also use two or three cameras at the same time (supporting ov13850 and ov13855 mixed). After connecting the three cameras, open the configuration of Cam0+Cam1+Cam2 through **orangepi-config** as in the previous steps, restart the system, and then open the terminal on the desktop to run the **test_camera.sh** script to see the preview of the three cameras, as shown in the figure below:



Please refer to the link below for camera dts configuration. You can modify it if needed.;

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588-orangepi-5-max-camera0.dtsi

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588-orangepi-5-max-camera1.dtsi

https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588-orangepi-5-max-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 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 has already been installed on the system, there is no

need to install it again.

Before installing zfs, you need to install the kernel header file. For the method of **installing the kernel header file**, please refer to the instructions in the section How to install the kernel header file

In Ubuntu 20.04, Ubuntu 22.04 and Debian 11 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, which is incompatible with the rk linux 5.10 kernel. This problem has been fixed in zfs 2.1.6 and later versions.

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 then **enter the deb package folder related to zfs used by Ubuntu and Debian systems.** You can see three types of deb packages: 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 section "How to upload files to the Linux system of the development board**".

After uploading, use the **cd** command in the command line of the Linux system of the development board 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, you can see the zfs-related kernel modules using the following command:

```
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 Debian 12, the default version of zfs is 2.1.11, so we can install zfs directly through the following command. Once again, please make sure that the system has installed the deb package of the kernel header file before installation.

```
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 a pool and then allocate storage space from this pool.

The following content is demonstrated based on the development board 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. The current 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 containing two storage devices: NVMe SSD and USB flash drive

orangepi@orangepi:~$ **sudo zpool create -f pool1 /dev/nvme0n1 /dev/sda**

3) Then use the **zpool list** command to see that the system has created a ZFS pool named **pool1**, and the size of the ZFS pool pool1 is the size of the NVME SSD plus the size of the USB flash drive.

```
orangepi@orangepi:~$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G   114K   504G        -         -     0%     0%  1.00x    ONLINE  -
```

4) Then execute **df -h** to 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 ZFS data deduplication function

1) ZFS data deduplication function is disabled by default. We need to execute the following command to enable it.

```
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 1G random file:

```
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 size 1G

```
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 a total of 1002G of data in the pool. However, the actual size of the ZFS pool is only **504GB** (the total capacity of the SSD + USB drive), which is too large to accommodate such a large amount of 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 duplicated, 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 ZFS data compression function

1) Because the stored data is different, the disk space saved by compression will also be different, so we choose to compress relatively large plain text files for compression

testing. Execute the following command to package the **/var/log/** and **/etc/** directories into a tar package

```
orangepi@orangepi:~$ cd /pool1/
root@orangepi:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

2) Then the file size and the space occupied in the ZFS pool can be seen by the **ls -lh** command, both of which are **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 tarball

```
root@orangepi:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

5) You can see that the file size of **text.tar** is still 27M, but it only takes up 9.47M of space in the ZFS pool, indicating that the file has been 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 server, etc.

3. 36. 1. How to install CasaOS

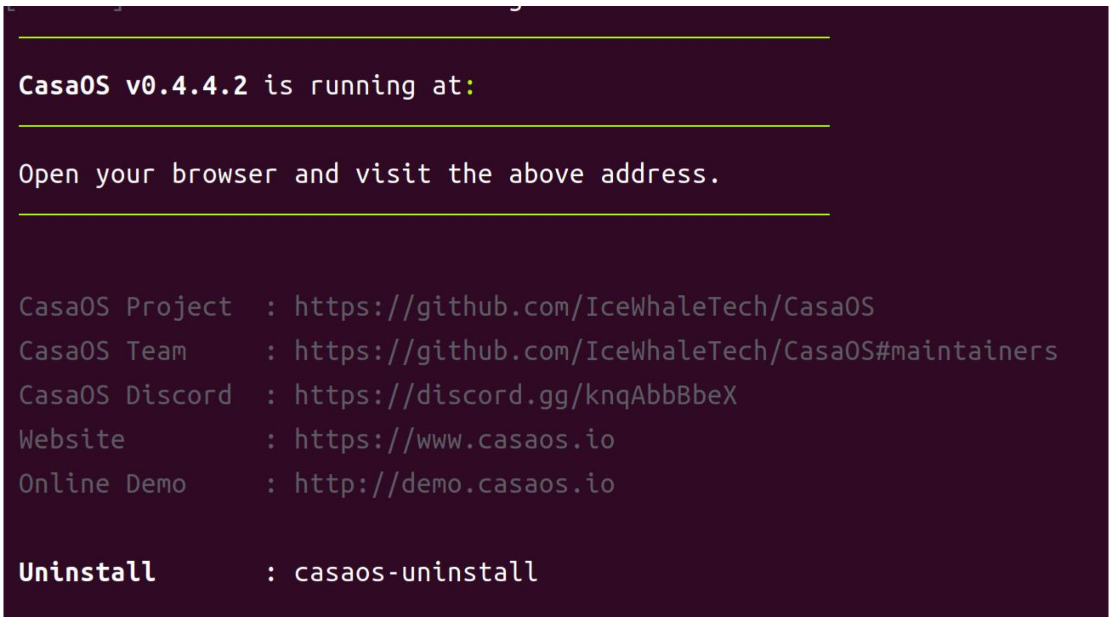
1) First, you need to install docker. The system released by Orangepi Pi has docker pre-installed. This step can be skipped. You can use the following command to view the installed docker version

```
orangepi@orangepi:~$ docker --version
Docker version 24.0.2, build cb74dfc          # Ubuntu Jammy System Output
```

2) Then enter the following command in the Linux system to start the installation of CasaOS

```
orangepi@orangepi:~$ curl -fsSL https://get.casaos.io | sudo bash
```

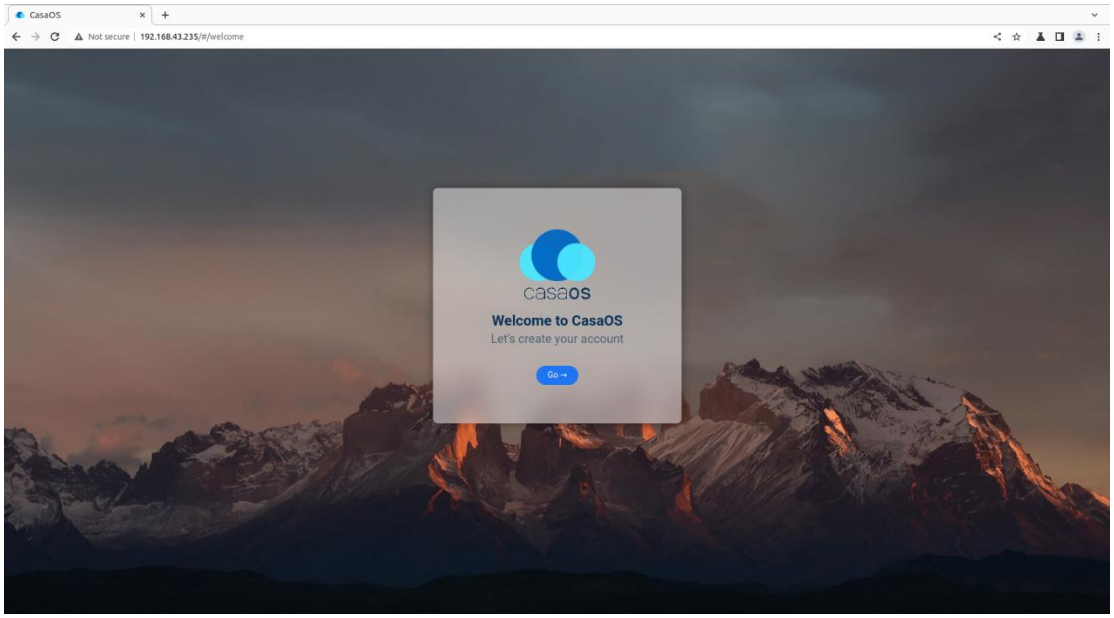
3) When you see the following print information output in the terminal, it means that CasaOS has been installed.



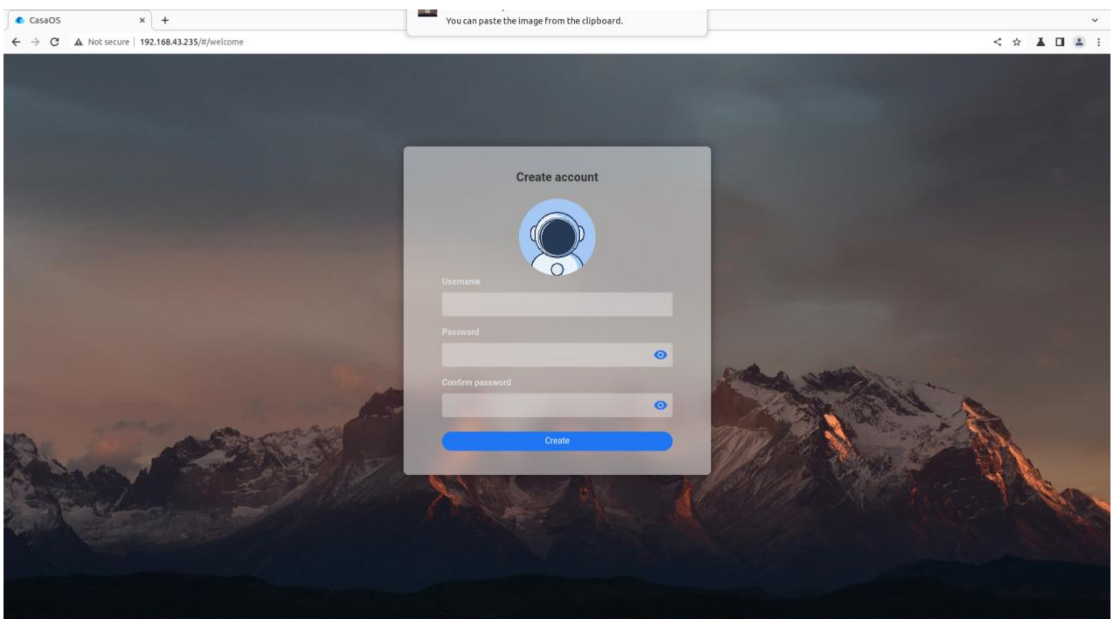
3. 36. 2. How to use CasaOS

1) After installing CasaOS, enter **http://development board IP address** in the browser to open CasaOS

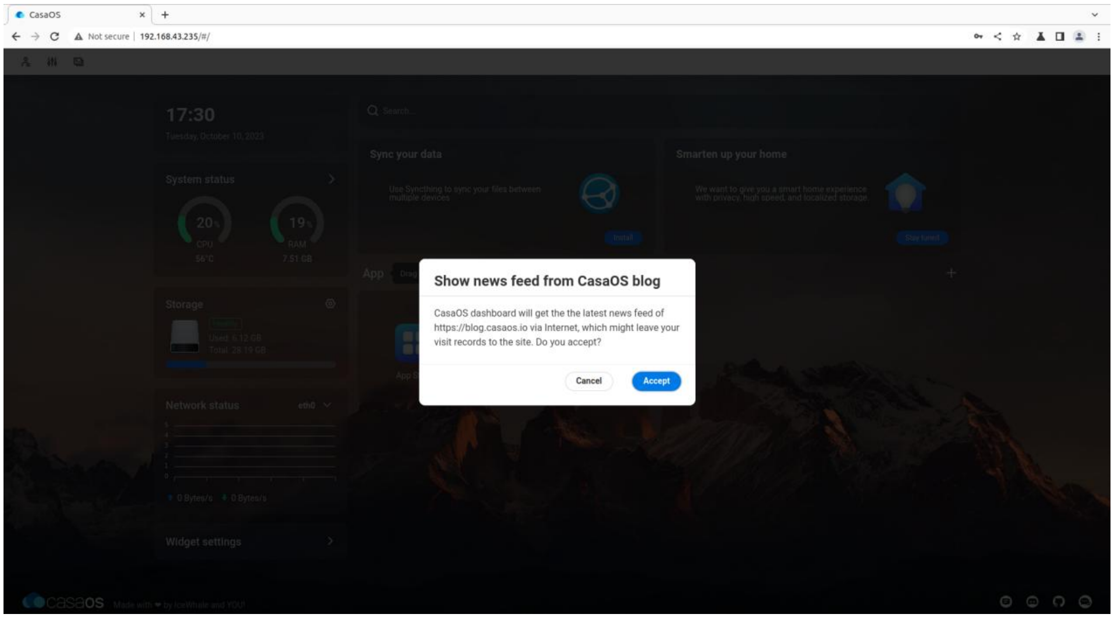
2) After opening CasaO, the following welcome interface will pop up. Click "Go" to proceed to the next step



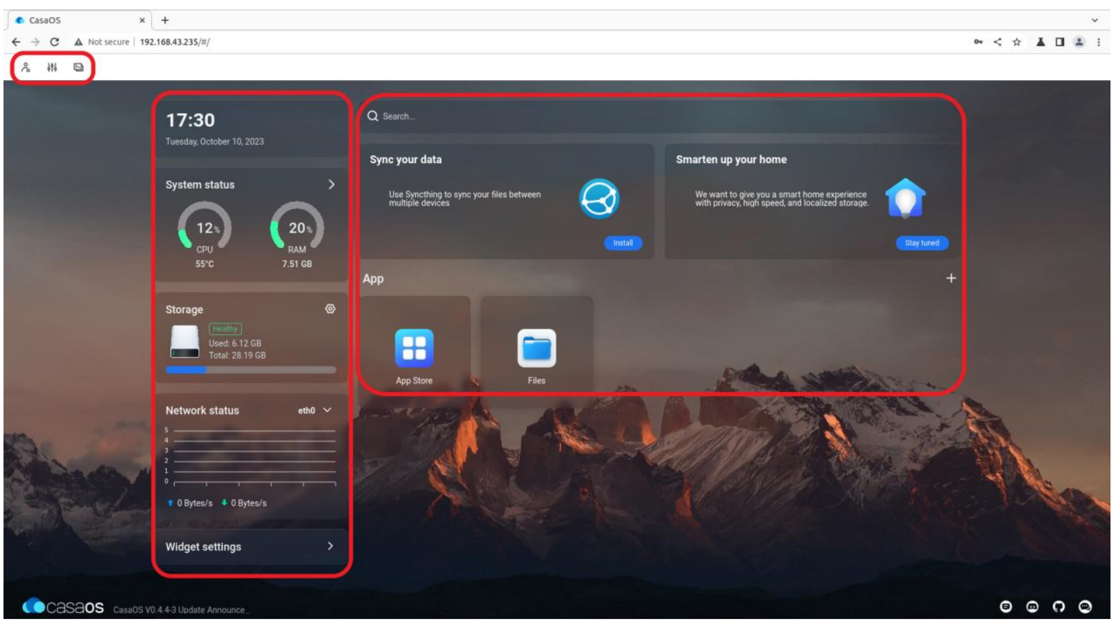
3) When you log in to CasaOS for the first time, the login interface is the interface for setting the account and password. When you log in again, only the interface for entering the account and password will appear. After setting the account and password, click "Create" to proceed to the next step.



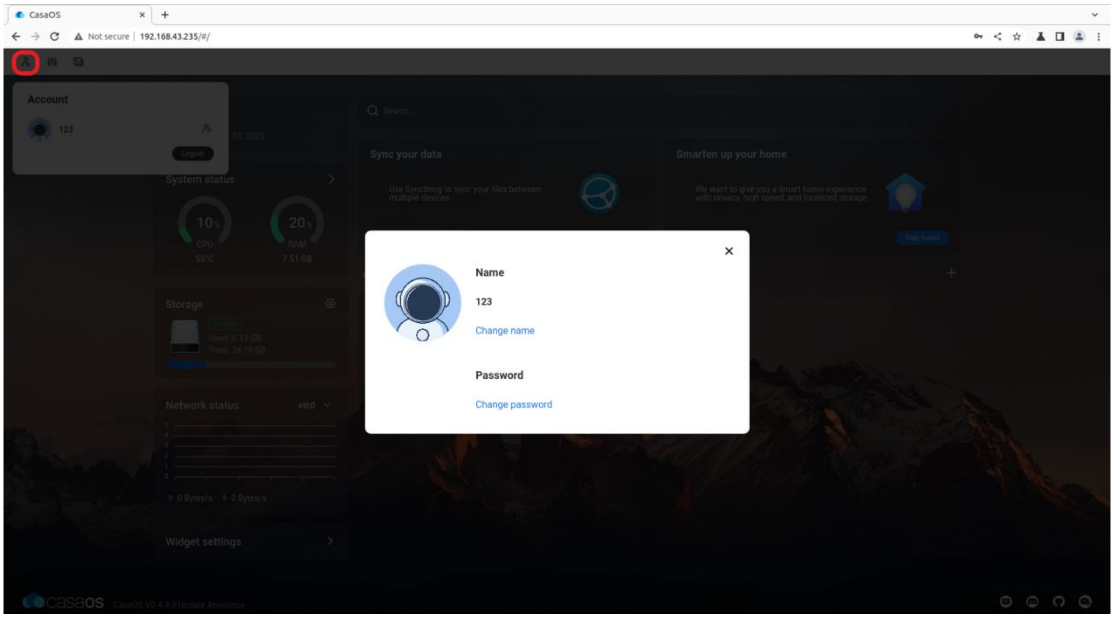
4) In the following interface, click "Accept" to proceed to the next step



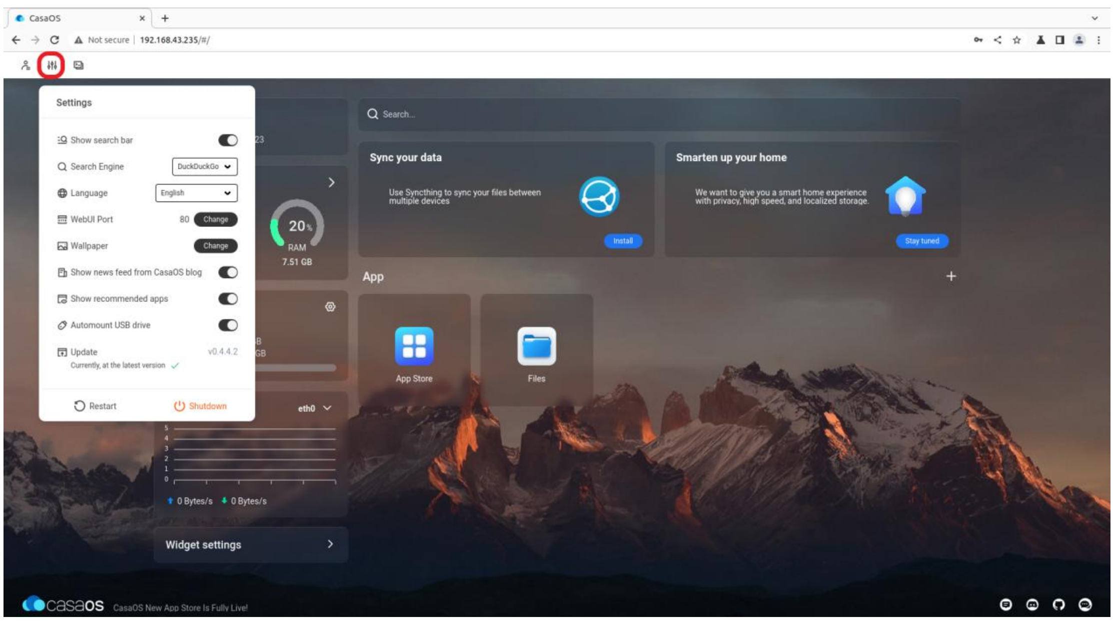
5) Now you have entered the main page of CasaOS. There are three icons in the upper left corner for function settings. The left side is the performance panel, which can display the current time and the status information of CPU, RAM, storage, and network. The right side is the function panel, which has functions such as search, application recommendation, application store, and file management.



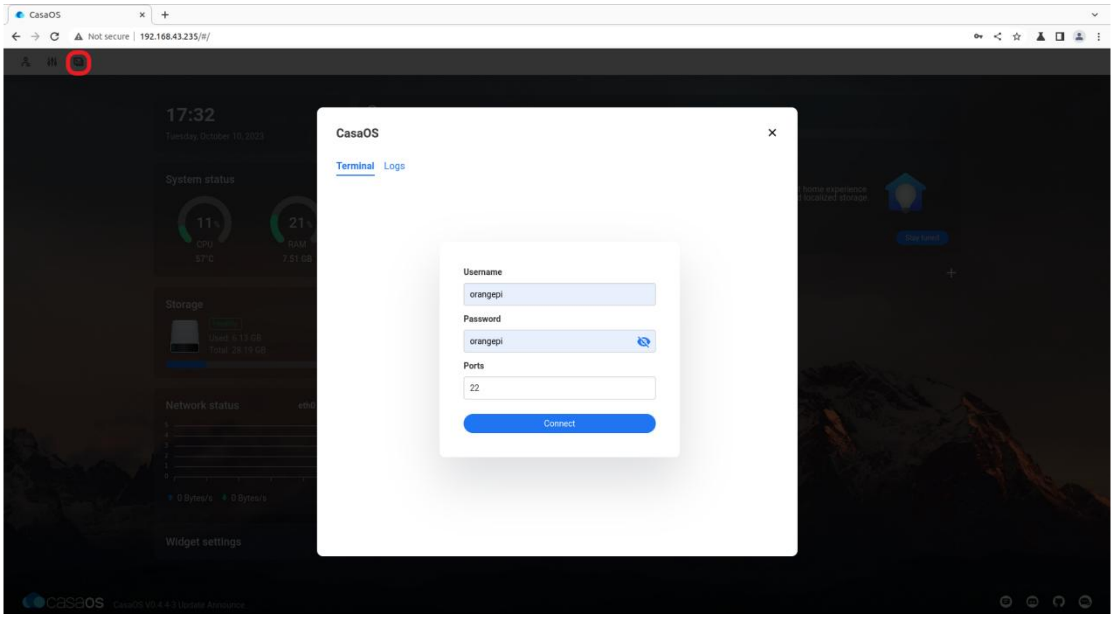
6) You can click the first icon in the upper left corner to modify your account and password



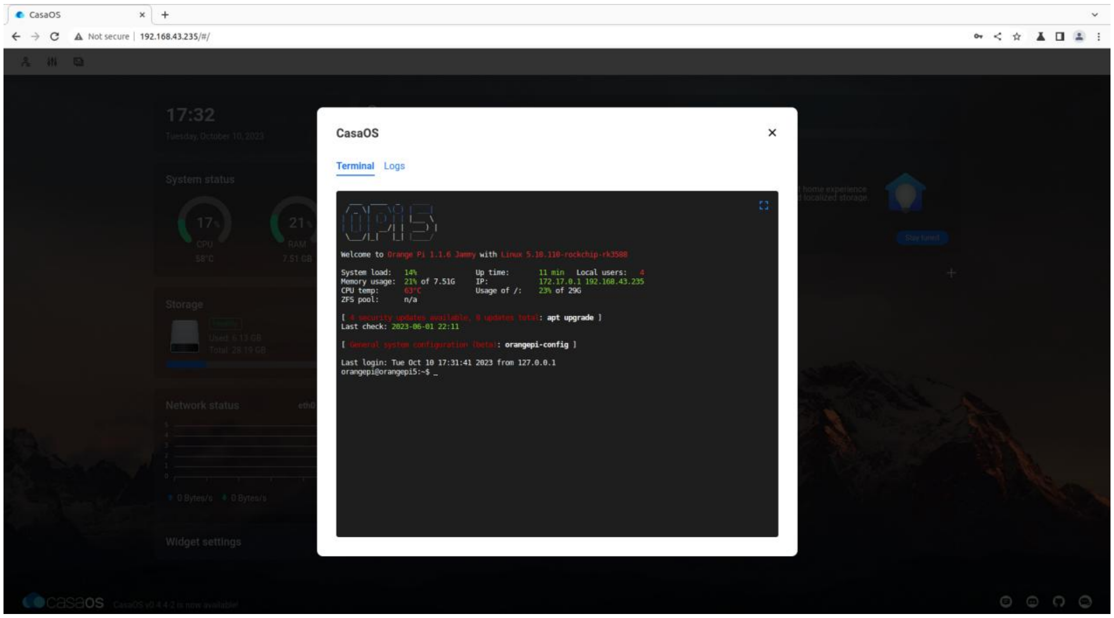
7) You can click the second icon to set basic functions



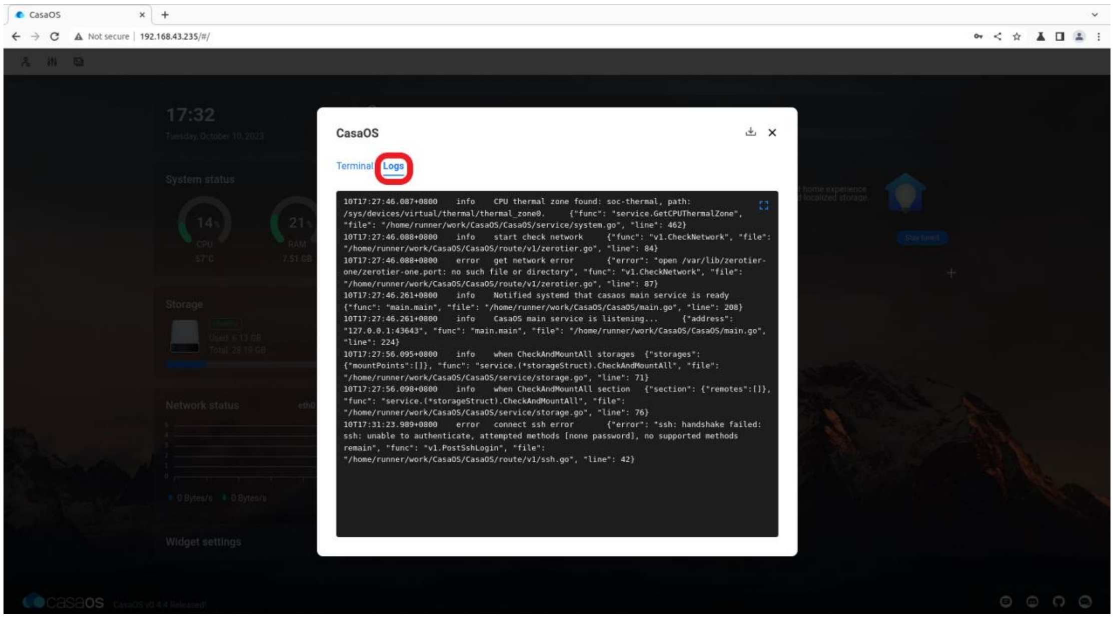
8) The third icon in the upper left corner has two main functions, namely switching to command line mode and printing log information. When switching to command line mode, you need to enter the account and password. The account and password here refer to the account and password of the development board Linux system. The default port system selects 22



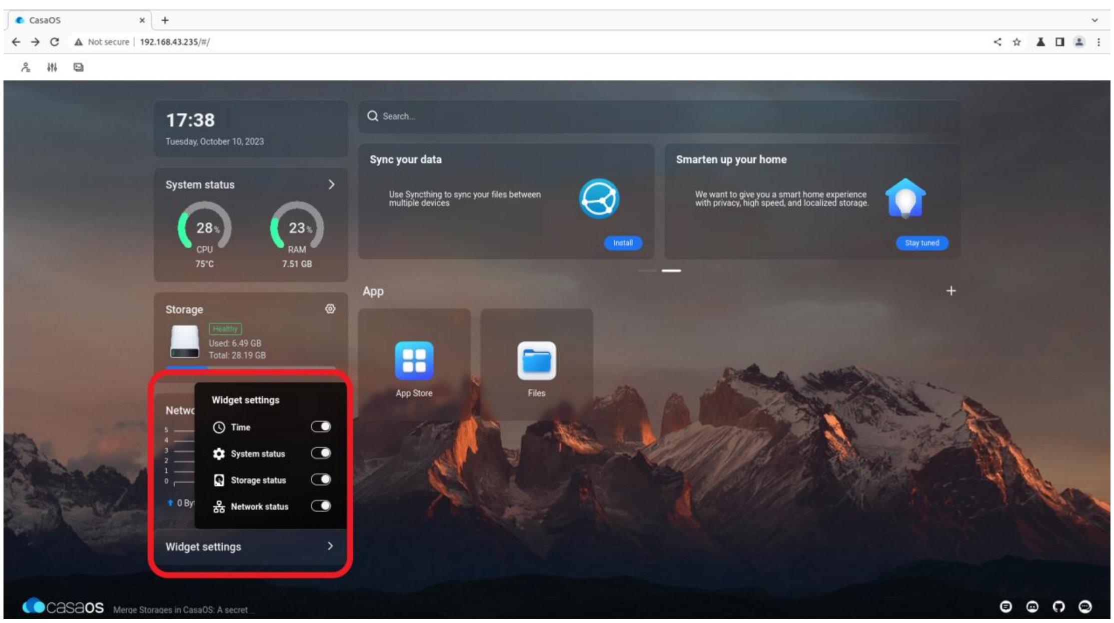
9) Then click "Connect" to enter the command line interface:



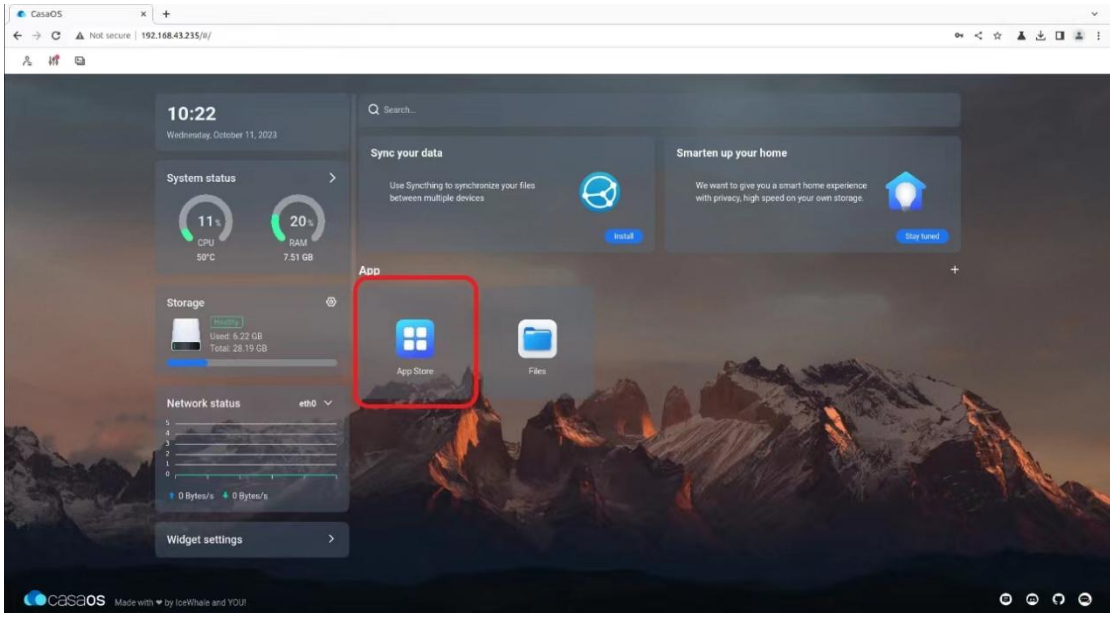
10) Another function under the third icon is to print the CasaOS log. Click "Logs" to enter. The interface is as follows:



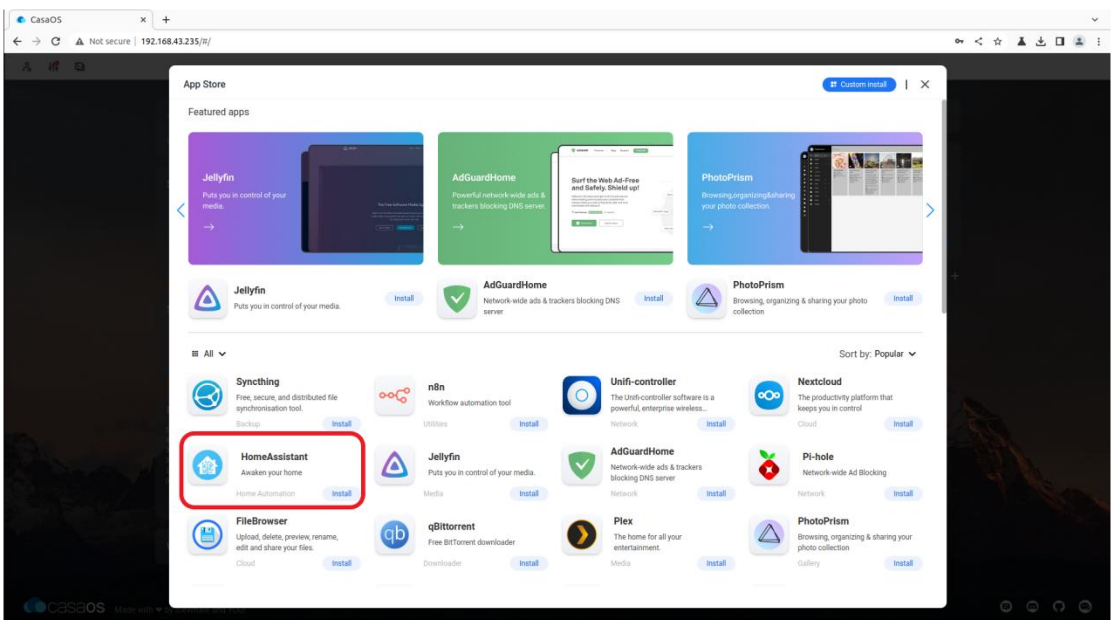
11) Click "Widget settings" in the lower left corner to set whether to display the performance panel widget on the main page



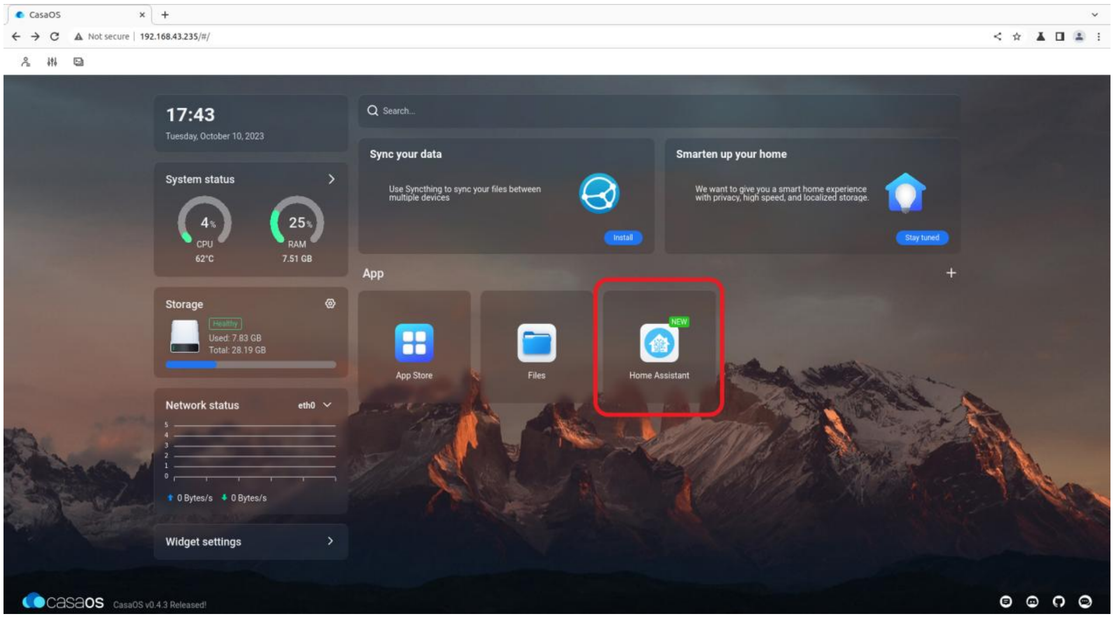
12) Click "APP Store" on the main interface to open the App Store. Currently, there are more than 70 APPs available in the App Store.



13) Here we take Home Assistant as an example to download. Find Home Assistant in the APP Store and click the corresponding "install"

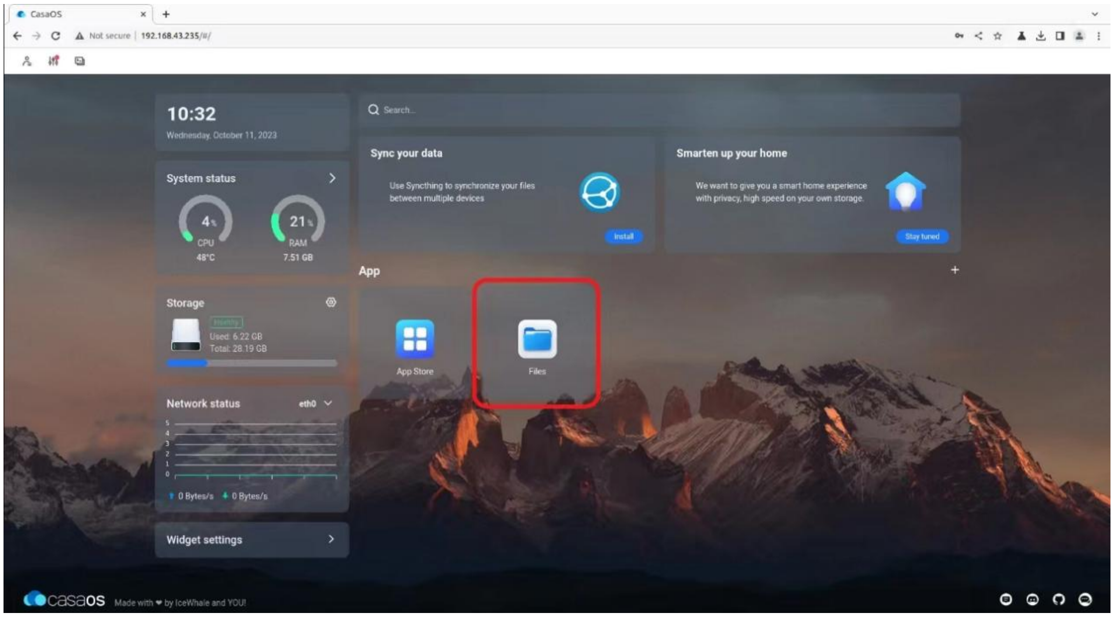


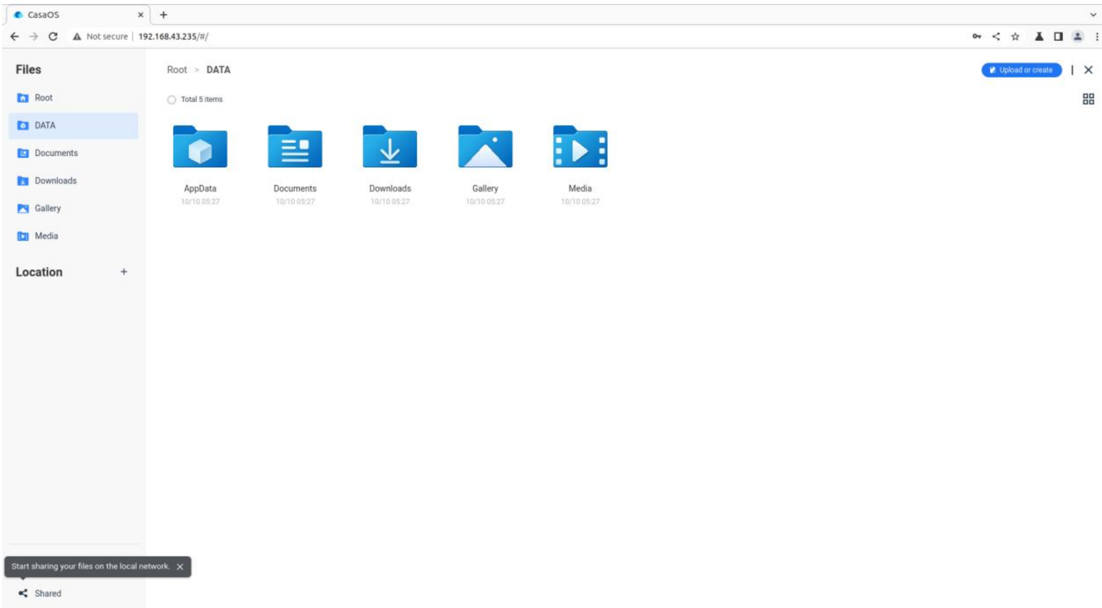
14) After the download is complete, HostAssitant will appear on the main page



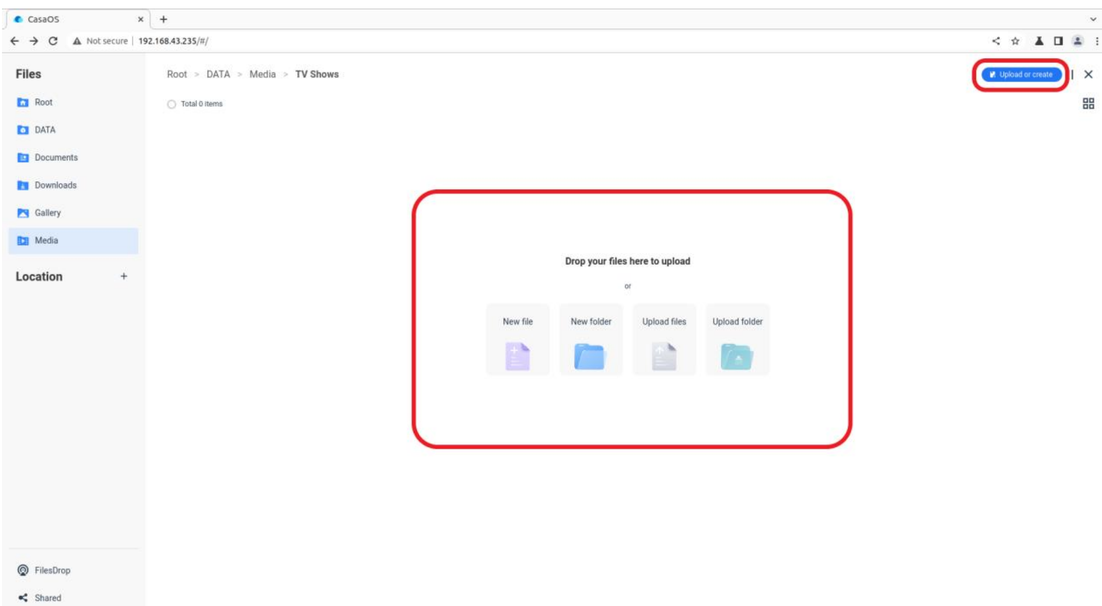
15) Click "Files" in the main interface to open the file system that comes with CasaOS, and then you can upload and save files

Please make sure 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 figure, 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@orangepi5max:~$ casaos-uninstall
```

3. 37. Methods of using NPU

3. 37. 1. Prepare tools

1) A PC with Ubuntu 20.04 operating system

According to the official documentation of RKNN-Toolkit2, the operating systems supported by the current version of RKNN-Toolkit2 are as follows:

a. Ubuntu18.04 (x64)

b. Ubuntu20.04 (x64)

c. Ubuntu22.04 (x64)

In this document, we use the Ubuntu 20.04 (x64) operating system for demonstration. Please test other versions of the operating system yourself.

2) An RK3588 development board with Debian 11 installed

3) A Type-C data cable for using the adb function

3. 37. 2. Install RKNN-Toolkit2 on Ubuntu PC

Toolkit2 is a development kit used on the Ubuntu PC platform. Users can use the Python interface provided by the tool to easily complete functions such as model conversion, reasoning, and performance evaluation.

1) On the Ubuntu PC, open a command line window and enter the following commands to install python3 and pip3

```
test@test:~$ sudo apt-get install python3 python3-dev python3-pip
```

2) You can use the following command to view the installed version of python3

```
test@test:~$ python3 --version
```

```
Python 3.8.10
```

3) Then enter the following command to install the dependency package of RKNN-Toolkit2

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 \
 libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

4) Then enter the following command to download the 1.5.2 version of RKNN-Toolkit2

```
test@test:~$ git clone git clone https://github.com/airockchip/rknn-toolkit2 -b v1.5.2
```

5) Then enter the following command to install the corresponding version of Python3 dependency packages. This command will use pip3 to install the dependencies listed in the file requirements_cp38-1.5.2.txt. If the dependencies are not fully installed, do not specify the installation source and install each package separately.

```
test@test:~$ pip3 install -r rknn-toolkit2/doc/requirements_cp38-1.5.2.txt -i \
https://mirror.baidu.com/pypi/simple
```

6) Then enter the following command to use pip3 to install the RKNN-Toolkit2 software package. After the installation is complete, you can use RKNN-Toolkit2

```
test@test:~$ pip3 install rknn-toolkit2/packages/rknn_toolkit2-1.5.2+b642f30c-cp38-cp38-linux_x86_64.whl
```

3. 37. 3. Model conversion and model inference using RKNN-Toolkit2

RKNN-Toolkit2 supports converting Caffe, TensorFlow, TensorFlow Lite, ONNX, DarkNet, PyTorch and other models into RKNN models, and then ru nning the RKNN model on the Ubuntu PC side through simulation or using t he NPU of the development board for inference.

Relevant examples are provided in the example folder of RKNN-Toolkit2 to help users better understand how to operate. We take the ONNX model wi th yolov5 function as an example.

3. 37. 3. 1. Simulate the model on Ubuntu PC

RKNN-Toolkit2 is equipped with a built-in simulator, which allows users t o simulate the inference process of the model on Rockchip NPU on Ubuntu P C.

In this way, model conversion and inference can be completed on the Ubuntu PC, helping users test and verify their models faster.

1) First switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

2) Then run the test.py script, which first converts the yolov5s_relu.onnx model into an RKNN model that can be run on the simulator, and then uses the simulator to simulate and run the model to infer the bus.jpg image in the current directory.

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

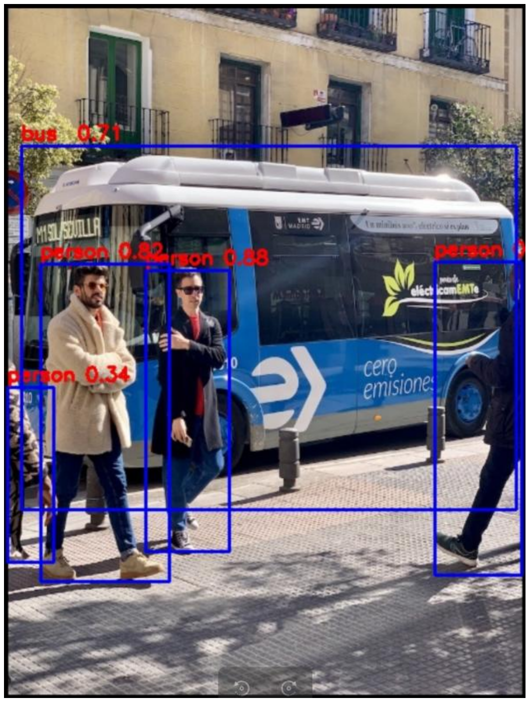
3) After the test.py script runs successfully, you will see the following print information, indicating that the model successfully detected four people and a bus in the bus.jpg image

```
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 inference picture 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. Using the NPU of the development board to run the model on Ubuntu PC

RKNN-Toolkit2 provides users with a Python interface for using the development board's NPU for inference through adb, allowing users to use the development board's NPU to run models for inference on an Ubuntu PC.

In this way, the Ubuntu PC can use the machine learning library provided by Python to optimize and adjust the model according to the actual effect of the model running on the NPU of the development board.

3. 37. 3. 2. 1. Connect adb using Type-C cable

Use adb to operate the development board on the Ubuntu PC. For the usage of adb, please refer to the instructions in the section **How to use ADB**.

3. 37. 3. 2. 2. Update the rknn_serverand and librknnrt.so of the development board

librknnrt.so is a board-side runtime library.

rknn_server is a background proxy service running on the development board. It is used to receive the protocol transmitted from the PC via USB, then execute the corresponding interface in the board runtime library and return the result to the PC.

1) First, enter the following command on the Ubuntu PC to download the 1.5.2 version of RKNPU2

```
test@test:~$ git clone https://github.com/rockchip-linux/rknpu2 -b v1.5.2
```

2) Then enter the following command on the Ubuntu PC to update the rknn_server of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/rknn_server/aarch64/usr/bin/* /usr/bin
```

3) Then enter the following command on the Ubuntu PC to update the librknnrt.so library of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/librknn_api/aarch64/librknnrt.so /usr/lib
```

4) Open the terminal of the development board through the adb tool

```
test@test:~$ adb shell
```

5) Open the rknn_server service of the development board

```
root@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 has been opened, and the operating environment of the development board has been set up.

```
root@orangepi:/# pgrep rknn_server
```

```
3131
```

3. 37. 3. 2. 3. Modify the parameters in the example

1) On the Ubuntu PC, you can use the following command to view the device ID of the development board connected to the Ubuntu PC. 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 the vim editor to modify the test.py file

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ vim test.py
```

4) In the test.py file, we need to modify the following：

a. In the preprocessing configuration, change the target platform to rk3588, so that the model conversion results in an RKNN model suitable for the NPU of 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 the description of the target platform and device ID. The target platform is rk3588, and the device ID is the device ID of the development board obtained through adb. The operation of running the model for inference will be 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 the modification is completed, save and exit

3. 37. 3. 2. 4. Run the example on Ubuntu PC

1) Enter the following command to run the test.py script. The script first converts the yolov5s_relu.onnx model to the RKNN model, and then loads the model to the NPU of the development board to perform inference on the out.jpg image in the current directory.

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

2) In the printed information, we can see that the Ubuntu PC uses the NPU of the development board to run the model for inference through the adb tool

```
--> Init runtime environment
I target set by user is: rk3588
I Check RK3588 board npu runtime version
I Starting ntp or adb, target is RK3588
I Device [4f9f859e5a120324] not found in ntb device list.
I Start adb...
I Connect to Device success!
I NPUTransfer: Starting NPU Transfer Client, Transfer version 2.1.0
(b5861e7@2020-11-23T11:50:36)
```

3) After the test.py script runs successfully, the converted model file yolov5s_relu.rknn and the inference image result result.jpg are saved in the current directory

4) The results of the operation are the same as those in the section **Simulating the model on the Ubuntu PC.**

3.37.4. Call the C interface to deploy the RKNN model to the development board to run

RKNPU2 provides a C programming interface for chip platforms with Rockchip NPU, which can help users deploy RKNN models exported using RKNN-Toolkit2 and accelerate the implementation of AI applications.

In the example folder of RKNPU2, examples of deploying RKNN models with different functions to the development board are provided. We take the deployment of the RKNN model with yolov5 function to the RK3588 Debian 11 platform as an example.

3.37.4.1. Download cross-compilation tools

Since the development board runs on Linux, you need to use the gcc cross compiler to compile. It is recommended to use the gcc version of gcc-9.3.0-x86_64_arrch64-linux-gnu

Enter the following command to download this version of gcc. After downloading, you will get a folder named gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu

```
test@test:~$ git clone https://github.com/airockchip/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
```

3.37.4.2. Modify the compiler tool path in the script

1) Switch to the rknpu2/examples/rknn_yolov5_demo directory

```
test@test:~$ cd ~/rknpu2/examples/rknn_yolov5_demo
```

2) Use the vim editor to modify the contents of the build-linux_RK3588.sh file.

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ vim build-linux_RK3588.sh
```

3) In the build-linux_RK3588.sh file, we need to change the value of the variable TOOL_CHAIN to the path of the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder. In this way, when running the build-android_RK3588.sh script, the cross-compilation tool in the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder will be used for compilation

```
TARGET_SOC="rk3588"
GCC_COMPILER=aarch64-linux-gnu

export TOOL_CHAIN=~/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
export LD_LIBRARY_PATH=${TOOL_CHAIN}/lib64:$LD_LIBRARY_PATH
export CC=${GCC_COMPILER}-gcc
export CXX=${GCC_COMPILER}-g++
```

4) After modification, save and exit

3. 37. 4. 3. Compile rknn_yolov5_demo

1) Run build-linux_RK3588.sh, which generates a program suitable for the RK3588 development board through cross-compilation and can run the RKNN model for inference on it

```
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, there will be an additional folder named install in the current directory. The rknn_yoov5_demo_Linux folder under this folder contains the program generated by cross-compilation and its related files

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ls install
rknn_yolov5_demo_Linux
```

3. 37. 4. 4. Deploy rknn_yolov5_demo to the development board

On the Ubuntu PC, you can use the following command to upload the rknn_ yolov5_demo_Linux folder to the development board through the adb tool to depl oy 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. Run rknn_yolov5_demo on the development board

1) Enter the file system of the development board through adb shell on the 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 to perform inference. In the following command, the program uses the yolov5s-640-640.rknn model to perform inference on the bus.jpg image. The entire running process will be completed on the development board

```
root@orangepi:/data/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo \
./model/RK3588/yolov5s-640-640.rknn ./model/bus.jpg
```

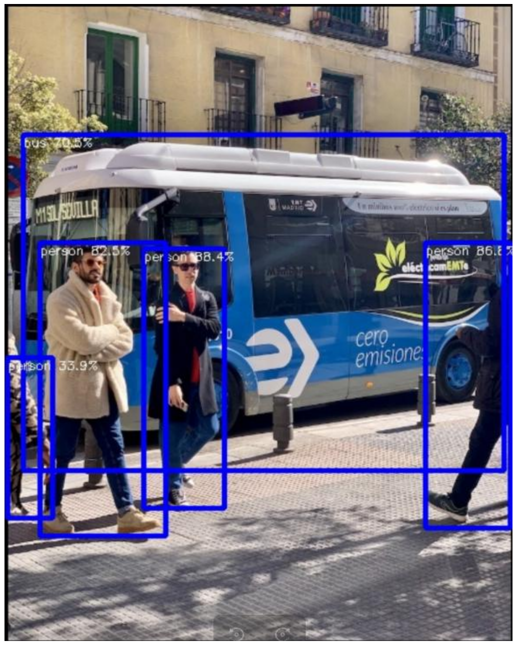
4) After the run is completed, the inference result out.jpg image is saved in the current directory

```
root@orangepi:/data/rknn_yolov5_demo_Linux# ls
lib   model   out.jpg   rknn_yolov5_demo   rknn_yolov5_video_demo
```

5) On the Ubuntu PC, we can use the following command to download the out.jpg image using the adb tool, and then view it using an image viewer

```
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's method of using Paddle Paddle

Use Paddle Paddle on the rk3588 development board, including converting the pdmodel model to the rknn model on the PC side and deploying the rknn model using the FastDeploy deployment tool developed by Paddle Paddle on the board side. The following content is implemented in an environment where the PC system is Ubuntu 22.04 and the board system is Debian 11. Please test other environments yourself.

3. 38. 1. Ubuntu PC environment setup

The tools and uses that need to be installed on the Ubuntu PC are as follows

Tool Name	Purpose
Anaconda3	Used to create and manage Python environments
Paddle2ONNX	Used to convert the pdmodel model to the ONNX model
RKNN-Toolkit2	Used to convert ONNX model to RKNN model

3. 38. 1. 1. Install Anaconda3 on PC

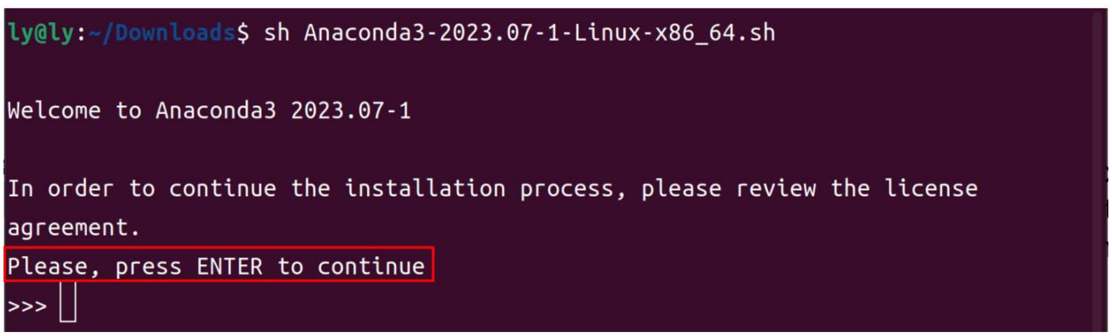
1）Open the browser on the Ubuntu PC and enter the following URL in the address bar to download and install the Anaconda3 script. After the download is complete, you will get the **Anaconda3-2023.07-1-Linux-x86_64.sh**

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）The installation script will then output the following prompt message, at this time click Enter to continue the installation



4）After pressing the Enter key, some introduction information about Anaconda3 will appear. Keep pressing 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）The installation script will then prompt you to accept the license terms. Enter yes and press Enter.

```
The following packages listed on https://www.anaconda.com/cryptography are inclu
ded in the repository accessible through Anaconda Distribution that relate to cr
yptography.

Last updated February 25, 2022


Do you accept the license terms? [yes|no]
[no] >>>
```

6）The installation script will then prompt 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 you see the following print in the terminal, it means 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. Install RKNN-Toolkit2 on PC

1）Open the terminal on the Ubuntu PC and use the Anaconda3 tool to create a Python 3.8 environment.

(base)test@test:~$ **conda create -n fastdeploy python=3.8**

2）Activate the python3.8 environment just created

(base)test@test:~$ **conda activate fastdeploy**

3）Then install pip3 development tools and package management tools

(fastdeploy)test@test:~$ **sudo apt-get install python3-dev python3-pip**

4）Then install the dependency package of RKNN-Toolkit2

```
(fastdeploy)test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

5）rknn_toolkit2 has a specific dependency on numpy, so you need to install numpy==1.16.6 first

```
(fastdeploy)test@test:~$ pip install numpy==1.16.6
```

6）Install git tool

```
(fastdeploy)test@test:~$ sudo apt install git
```

7）Then execute the following command to download RKNN-Toolkit2. After the download is complete, you will get the rknn-toolkit2 folder

```
(fastdeploy)test@test:~$ git clone https://github.com/rockchip-linux/rknn-toolkit2
```

8）Then execute the following command to install RKNN-Toolkit2 corresponding to python3.8 version

```
(fastdeploy)test@test:~$ pip install rknn-toolkit2/rknn-toolkit2/packages/rknn_toolkit2-1.6.0+81f21f4d-cp38-cp38-linux_x86_64.whl
```

3. 38. 1. 3. Install Paddle2ONNX on PC

You can execute the following command to install paddle2onnx

```
(fastdeploy)test@test:~$ pip install paddle2onnx
```

3. 38. 2. Board environment construction

The tools and uses that need to be installed on the board are as follows

Tool Name	Purpose
Anaconda3	Used to create and manage Python environments
rknpu2	Basic driver for rknpu2
FastDeploy	After compilation, you can get the FastDeploy inference library

3. 38. 2. 1. Install Anaconda3 on the board

1）Open the browser on the board and enter the following URL in the address bar to download and install the Anaconda3 script. After the download is complete, you will get the **Anaconda3-2023.07-1-Linux-aarch64.sh**

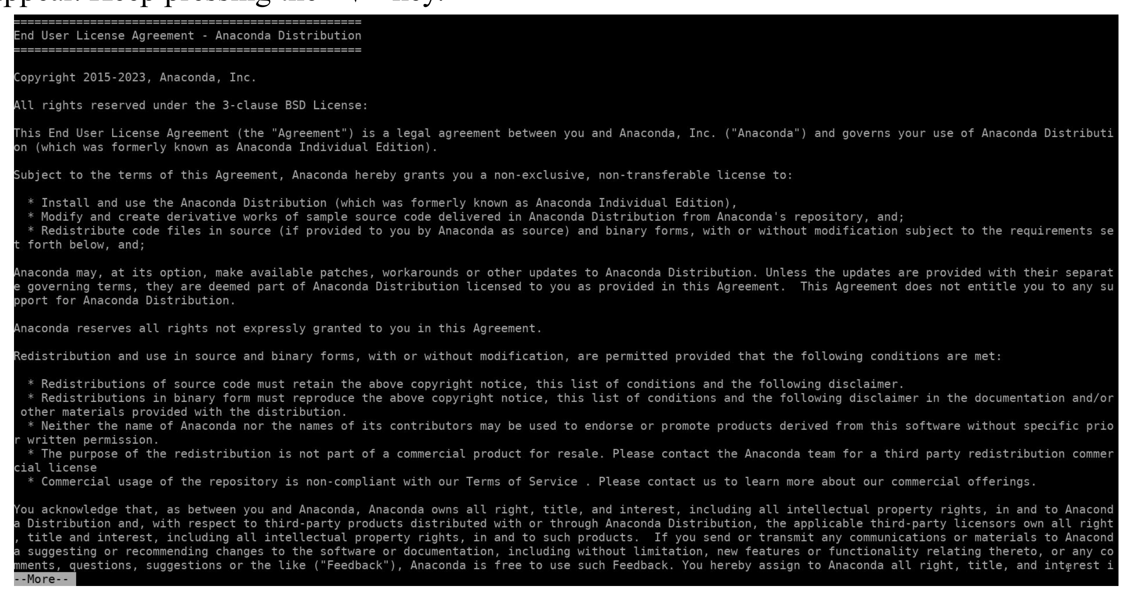
```
https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-aarch64.sh
```

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）The installation script will then output the following prompt message, click Enter to continue the installation

4）After pressing the Enter key, some introduction information about Anaconda3 will appear. Keep pressing the " ↓ " key.



5）The installation script will then prompt you to accept the license terms. Enter yes and press Enter.

```
The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to
cryptography.

Last updated February 25, 2022


Do you accept the license terms? [yes|no]
[no] >>>
```

6) The installation script will then prompt 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 you see the following print in the terminal, it means 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 in the terminal and it says the command does not exist, you need to modify the ~/.bashrc file

orangepi@orangepi:~$ **vi ~/.bashrc**

10) Add the following code to the end of the ~/.bashrc file

export PATH=/home/orangepi/anaconda3/bin:$PATH

11) Then enter the following command in the terminal to make the changes take effect

orangepi@orangepi:~$ **source ~/.bashrc**

12) Then enter the following command in the terminal to initialize conda

(base)orangepi@orangepi:~$ **conda init bash**

13) Then close the current terminal and reopen a terminal. You can now use the conda command normally.

3. 38. 2. 2. Install rknpu2 driver on the board

1）Open the terminal on the board and use the Anaconda3 tool to create a Python version 3.9 environment

```
(base)orangepi@orangepi:~$ conda create -n fastdeploy python=3.9
```

2）Activate the python3.9 environment just created

```
(base)orangepi@orangepi:~$ conda activate fastdeploy
```

3）Download the rknpu2_device_install_1.4.0.zip file via wget

```
(fastdeploy)orangepi@orangepi:~$ wget https://bj.bcebos.com/fastdeploy/third_libs/rknpu2_device_install_1.4.0.zip
```

4）Then execute the following command to decompress rknpu2_device_install_1.4.0.zip. After decompression, you will get the rknpu2_device_install_1.4.0 folder and the __MACOSX folder

```
(fastdeploy)orangepi@orangepi:~$ unzip rknpu2_device_install_1.4.0.zip
```

5）Switch to the rknpu2_device_install_1.4.0 directory

```
(fastdeploy)orangepi@orangepi:~$ cd rknpu2_device_install_1.4.0/
```

6）There is a rknn_install_rk3588.sh script in this directory. Run the script to complete the installation of the board-side rknpu2 driver.

```
(fastdeploy)orangepi@orangepi:~/rknpu2_device_install_1.4.0$ sudo bash rknn_install_rk3588.sh
```

3. 38. 2. 3. Compile FastDeploy C++ SDK on the board

1）The cmake command is needed when compiling. You can execute the following command to install the cmake tool

```
(fastdeploy)orangepi@orangepi:~$ sudo apt-get install -y cmake
```

2）Then download the FastDeploy SDK. After the command is executed, you will get the FastDeploy folder

```
(fastdeploy)orangepi@orangepi:~$ git clone https://github.com/PaddlePaddle/FastD
```

```
eploy.git
```

3) Switch to the FastDeploy directory

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy
```

4) Create a compilation directory build and switch to the build directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy$ mkdir build && cd build
```

5) Before compiling, you need to use cmake to configure the project information to be compiled. After executing the following command, there will be some more files in the current directory, including the Makefile file used for compilation

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ cmake .. -DENABLE_ORT_BACKEND=ON \
-DENABLE_RKNPU2_BACKEND=ON \
-DENABLE_VISION=ON \
-DRKNN2_TARGET_SOC=RK3588 \
-DCMAKE_INSTALL_PREFIX=${PWD}/fastdeploy-0.0.3
```

6) Execute the following command to start compiling

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make -j8
```

7) After the compilation is complete, use the following command to install the compiled files to the specified path

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make install
```

8) After the compilation is completed, you will get the fastdeploy-0.0.3 folder. In this folder, there is a script file fastdeploy_init.sh for configuring environment variables. After using this script to configure environment variables, you can use some compiled library files.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ source fastdeploy-0.0.3/fastdeploy_init.sh
```

3. 38. 3. Example of deploying a model using FastDeploy

The ResNet50_vd model is a model used for target classification. The following uses the ResNet50_vd model as an example to illustrate the process of using FastDeploy to deploy the pdmodel model.

3. 38. 3. 1. Ubuntu PC model conversion

1) Open the terminal on the PC and activate the python3.8 environment created using Anaconda3

```
test@test:~$ conda activate fastdeploy
```

2) In the model conversion script, you need to import the yaml module and the six module. You can execute the following command to install them.

```
(fastdeploy)test@test:~$ pip install pyyaml six
```

3) Execute the following command to download the ResNet50_vd_infer.tgz file

```
(fastdeploy)test@test:~$ wget https://bj.bcebos.com/paddlehub/fastdeploy/ResNet50_vd_infer.tgz
```

4) After decompressing the ResNet50_vd_infer.tgz file, you can get the ResNet50_vd_infer folder, which contains the pdmodel model file inference.pdmodel and other related files

```
(fastdeploy)test@test:~$ tar -xvf ResNet50_vd_infer.tgz
```

5) You can use the following command to convert the pdmodel model to an onnx model through paddle2onnx. After executing the command, there will be an additional onnx model file ResNet50_vd_infer.onnx in the ResNet50_vd_infer folder.

```
(fastdeploy)test@test:~$ paddle2onnx --model_dir ResNet50_vd_infer \
--model_filename inference.pdmodel \
--params_filename inference.pdiparams \
--save_file ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--enable_dev_version True \
--opset_version 10 \
--enable_onnx_checker True
```

6) Then use the following command to fix the shape to [1,3,224,224]. After executing the command, the ResNet50_vd_infer.onnx file will be modified.

```
(fastdeploy)test@test:~$ python -m paddle2onnx.optimize --input_model \
ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--output_model ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--input_shape_dict "{'inputs':[1,3,224,224]}"
```

7） To convert the onnx model to the rknn model, you need to use the script in the FastDeploy SDK. Execute the following command to download FastDeploy

```
(fastdeploy)test@test:~$ git clone https://github.com/PaddlePaddle/FastDeploy.git
```

8） Then transfer the ResNet50_vd_infer folder to the corresponding directory of FastDeploy

```
(fastdeploy)test@test:~$ mv ResNet50_vd_infer \
FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

9） Switch to the directory where the model conversion is performed

```
(fastdeploy)test@test:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

10） Execute the following command to convert the onnx model to the rknn model. Finally, the rknn model file ResNet50_vd_infer_rk3588_unquantized.rknn is obtained in the ResNet50_vd_infer directory.

```
(fastdeploy)test@test:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/$ python ./rknpu2_tools/export.py \
--config_path ./rknpu2_tools/config/ResNet50_vd_infer_rknn.yaml \
--target_platform rk3588
```

11） When deploying on the board, the rknn model file used is named ResNet50_vd_infer_rk3588.rknn, so you need 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-side model deployment

1） Open the terminal on the board and activate the python3.9 environment created previously using Anaconda3

```
orangepi@orangepi:~$ conda activate fastdeploy
```

2） Run the fastdeploy_init.sh script to configure the environment

```
(fastdeploy)orangepi@orangepi:~$ source FastDeploy/build/fastdeploy-0.0.3/fastdeploy_init.sh
```

3）Switch to the sample directory for deploying the ResNet50 model in FastDeploy

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp
```

4）Create a directory structure under this directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ mkdir build images ppclas_model_dir thirdpartys
```

5）Copy the compiled fastdeploy-0.0.3 folder to the thirdpartys folder

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cp -r ~/FastDeploy/build/fastdeploy-0.0.3/ thirdpartys/
```

6）Copy the files in the ResNet50_vd_infer folder on the PC to the ppclas_model_dir directory

7）Switch to the images directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cd images
```

8）Download the test image in the images directory 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

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ cd ../build/
```

10）Use cmake to configure the content that needs to be compiled. After executing the command, some files will appear in the current directory, including the Makefile 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 files to the specified path. After executing the command, an install directory will appear in the current directory.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make install
```

13）Switch to the install directory, where the model is used for reasoning.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cd install
```

14）Use the following command to use the converted rknn model to classify the content in the ILSVRC2012_val_00000010.jpeg image:

```
(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 information will be displayed, 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,
)
```

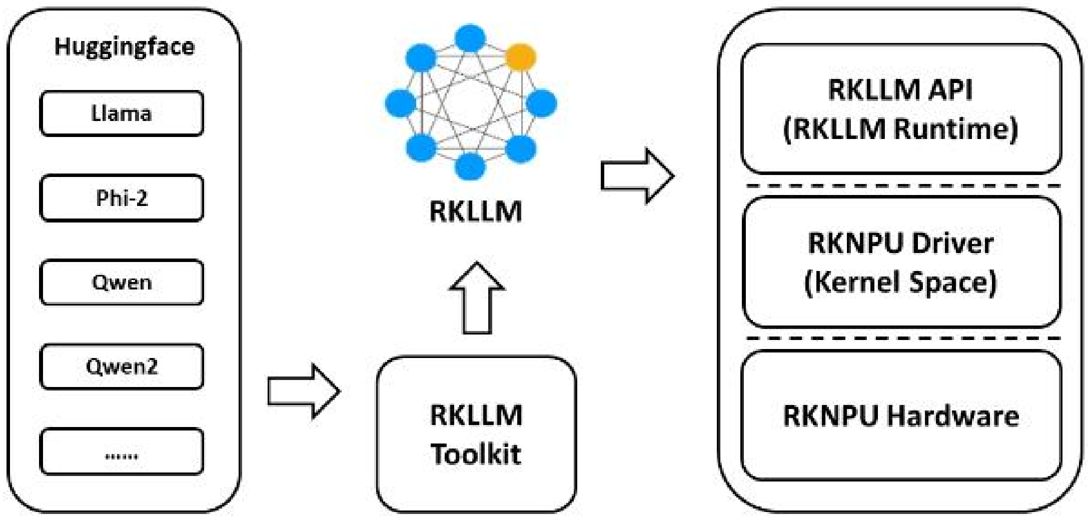
3. 39. RK3588 How to run the RKLLM large model

The codes and models used in this section can be downloaded from the official tools of the development board.

3. 39. 1. Introduction to RKLLM

For more detailed RKLLM introduction information, please refer to Rockchip RKLLM official information.

RKLLM can help users quickly deploy LLM models to the RK3588 development board. The overall framework is shown in the figure below:



3. 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 ability to quantize and convert large language models on a computer. The Python interface provided by this tool can be used to conveniently complete the following functions:

1) Model conversion: Supports conversion of large language models (LLM) in Hugging Face format to RKLLM models. Currently, the models we have tested include TinyLLAMA, Qwen, Qwen2, Phi-3, ChatGLM3, Gemma, InternLM2, and MiniCPM. The converted RKLLM model can be loaded and used on the RK3588 platform.

2) Quantization function: Supports quantizing floating-point models to fixed-point models. The currently supported quantization type is w8a8, which means that both weights and activations are quantized to 8-bit width.

3. 39. 1. 1. 2. RKLLM Runtime Function Introduction

RKLLM Runtime is mainly responsible for loading the RKLLM model converted by RKLLM-Toolkit, and implementing the reasoning of the RKLLM model on the RK3588 NPU by calling the NPU driver on the RK3588 board. When reasoning the RKLLM model, the user can define the reasoning parameter settings of the RKLLM model, define different text generation methods, and continuously obtain the reasoning results of the model through pre-defined callback functions. For more detailed instructions, please **refer to Rockchip RKLLM official information**.

3. 39. 1. 2. Introduction to RKLLM development process

The overall development steps of RKLLM are mainly divided into two parts: model conversion and board-side deployment and operation.

1) Perform model conversion on the Ubuntu PC. At this stage, the large language model in Hugging Face format provided by the user will be converted to RKLLM format for efficient reasoning on the RK3588 development board. This step includes

a. Build the RKLLM-Toolkit environment: Use Conda to build the RKLLM-Toolkit operating environment on the Ubuntu PC.

b. Model conversion: Use RKLLM-Toolkit to convert the obtained Hugging Face format large language model or the self-trained large language model (note that the structure of the saved model must be consistent with the model structure on the Hugging Face platform) into a .rkllm format file that can be run on the RK3588 development board.

c. Compile test code: Use rkllm-runtime to compile the inference program that can run on the RK3588 development board.

For the specific development process of model conversion on Ubuntu PC, please refer to **the detailed steps of model conversion and source code compilation on Ubuntu PC**.

2) **Deploy and run on the development board**. This stage covers the actual deployment and operation of the model on the RK3588 development board. It usually includes the following steps:

a. Upgrade the kernel NPU version: Upgrade the NPU version of the development board kernel to v0.9.6.

b. Model reasoning: Place the reasoning program compiled by rkllm-runtime on the Ubuntu PC and the .rkllm format file converted by RKLLM-Toolkit on the development board for model reasoning. You can run reasoning directly on the development board. For the specific development process, please refer to **the detailed steps of development board deployment and operation section of this chapter**. You can also deploy the board-side Server service on the development board. The Ubuntu PC in the same network segment can call the RKLLM model for reasoning by accessing the corresponding address. For the specific development process, please refer to **the detailed steps of development board server deployment and operation section of this chapter**.

The above two steps constitute the complete RKLLM development process, ensuring that the large language model can be successfully converted, debugged, and ultimately deployed efficiently on the RK3588 NPU.

3. 39. 2. Prepare tools

1) A PC with Ubuntu 22.04 operating system**. In this document, we use Ubuntu 22.04 (x64) operating system for demonstration. Please test other versions of operating system by yourself.**

2) An RK3588 development board

3. 39. 3. Detailed steps for model conversion and source code compilation on Ubuntu PC

3. 39. 3. 1. Build RKLLM-Toolkit environment

1) First download the RKLLM toolchain.

```
test@test:~$ git clone https://github.com/airockchip/rknn-llm.git
```

2) After downloading, use the ls command to check whether the downloaded file is correct.

```
test@test:~/test$ ls
rknn-llm
test@test:~$ cd rknn-llm
test@test:~/rknn-llm$ ls
```

```
CHANGELOG.md    doc    LICENSE    README.md    res    rkllm-runtime
rkllm-toolkit    rknpu-driver
```

3) The specific file directory in rknn-llm is as follows:

```
test@test:~/rknn-llm$ sudo apt install tree
test@test:~/rknn-llm$ tree
doc
 └──Rockchip_RKLLM_SDK_CN.pdf      # RKLLM SDK Documentation

rkllm-runtime
 ├──examples
 │    └── rkllm_api_demo    # Board-side inference call example project
 │   └── rkllm_server_demo # RKLLM-Server Deploy the sample project
 ├──runtime
 │    └── Android
 │         └── librkllm_api
 │            └──arm64-v8a
 │               └── librkllmrt.so # RKLLM Runtime Library
 │            └──include
 │               └── rkllm.h              # Runtime head File
 │    └── Linux
 │         └── librkllm_api
 │            └──aarch64
 │               └── librkllmrt.so # RKLLM Runtime Library
 │            └──include
 │               └── rkllm.h    # Runtime head File
rkllm-toolkit
 ├──examples
 │    └── huggingface
 │         └── test.py
 ├──packages
 │    └── md5sum.txt
 │    └── rkllm_toolkit-x.x.x-cp38-cp38-linux_x86_64.whl
rknpu-driver
 └──rknpu_driver_0.9.6_20240322.tar.bz2
```

4) Then download and install the miniforge3 installation package.

```
test@test:~$ wget -c https://mirrors.bfsu.edu.cn/github-release/conda-forge/miniforge/LatestRelease/Miniforge3-Linux-x86_64.sh

test@test:~$ chmod 777 Miniforge3-Linux-x86_64.sh

test@test:~$ bash Miniforge3-Linux-x86_64.sh
```

The mirror website sometimes crashes, resulting in the inability to download the miniforge3 package. The downloaded miniforge3 installation package has been provided in the official tool of the development board.

When running bash Miniforge3-Linux-x86_64.sh, just press Enter for all the options.

5) Then enter the Conda base environment.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$
```

6) Then create a Conda environment named RKLLM-Toolkit with Python 3.8 (recommended version).

```
(base) test@test:~$ conda create -n RKLLM-Toolkit python=3.8
```

7) Then enter the RKLLM-Toolkit Conda environment.

```
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

8) Then use the pip command to install the whl package in the RKLLM toolchain downloaded previously. The directory is:**rknn-llm/rkllm-toolkit/packages/ rkllm_toolkit-1.0.1-cp38-cp38-linux_x86_64.whl**。During the installation process, the installation tool will automatically download the related dependency packages required by the RKLLM-Toolkit tool.

```
(base) test@test:~$ pip3 install rknn-llm/rkllm-toolkit/packages/rkllm_toolkit-1.0.1-cp38-cp38-linux_x86_64.whl
```

9) Finally, if there is no error when executing the following command, it means the installation is successful

```
(RKLLM-Toolkit) test@test:~$ python
>>> from rkllm.api import RKLLM
```

3. 39. 3. 2. Model conversion

In this section, we provide eight model conversion examples for users to choose from. If users encounter network problems when downloading models from Hugging Face, our development board official tool has integrated the downloaded model files and the corresponding .rkllm conversion files.

3. 39. 3. 2. 1. Converting the TinyLLAMA Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the TinyLLAMA model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded TinyLlama-1.1B-Chat-v1.0 folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./TinyLlama.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/TinyLlama-1.1B-Chat-v1.0"    #Fill in your own path
ret = llm.export_rkllm("./TinyLlama.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(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) After the conversion is successful, you will get the TinyLlama.rkllm file in the current directory, which is about 1.09G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    TinyLlama.rkllm
```

3. 39. 3. 2. 2. Conversion to Qwen Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Then download the Qwen model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen-1_8B-Chat
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded Qwen-1_8B-Chat folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The brackets are the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Qwen.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/Qwen-1_8B-Chat"    #Fill in your own path
ret = llm.export_rkllm("./Qwen.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of a 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) If the conversion is successful, the Qwen.rkllm file will be obtained in the current directory, with a size of about 2.01G.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen.rkllm
```

3. 39. 3. 2. 3. Converting Qwen2 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Then download the Qwen2 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen1.5-0.5B
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded Qwen1.5-0.5B folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The brackets are the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Qwen2.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/Qwen1.5-0.5B"    # Fill in your own path
ret = llm.export_rkllm("./Qwen2.rkllm")
```

4) Run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of 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) If the conversion is successful, the Qwen2.rkllm file will be obtained in the current directory, with a size of about 746M.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen2.rkllm
```

3. 39. 3. 2. 4. Converting Phi-3 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the Phi-3 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/microsoft/Phi-3-mini-4k-instruct
(RKLLM-Toolkit) test@test:~$ cd Phi-3-mini-4k-instruct
(RKLLM-Toolkit) test@test:~/Phi-3-mini-4k-instruct$ git reset --hard 291e9e30e38030c23497afa30f3af1f104837aa6
(RKLLM-Toolkit) test@test:~/Phi-3-mini-4k-instruct$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded Phi-3-mini-4k-instruct folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Phi3.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
```

```
modelpath = "/path/your/Phi-3-mini-4k-instruct"    # Fill in your own path
ret = llm.export_rkllm("./Phi3.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
`flash-attention` package not found, consider installing for better performance: No module named 'flash_attn'.
Current `flash-attenton` does not support `window_size`. Either upgrade or use `attn_implementation='eager'`.
Loading checkpoint shards: 100%|██████████| 2/2 [00:02<00:00,  1.46s/it]
Optimizing model:   0%|          | 0/32 [00:00<?, ?it/s]
You are not running the flash-attention implementation, expect numerical differences.
Optimizing model: 100%|██████████| 32/32 [15:36<00:00, 29.27s/it]
Converting model: 100%|██████████| 195/195 [00:00<00:00, 4109996.38it/s]
Model has been saved to ./Phi3.rkllm!
```

6) If the conversion is successful, you will get the Phi3.rkllm file in the current directory, which is about 3.66G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Phi3.rkllm
```

3. 39. 3. 2. 5. Converting ChatGLM3 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the ChatGLM3 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/THUDM/chatglm3-6b
(RKLLM-Toolkit) test@test:~$ cd chatglm3-6b
(RKLLM-Toolkit) test@test:~/chatglm3-6b$ git reset --hard 103caa40027ebfd8450289ca2f278eac4ff26405
(RKLLM-Toolkit) test@test:~/chatglm3-6b$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded chatglm3-6b folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./chatglm3.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/chatglm3-6b"    # Fill in your own path
ret = llm.export_rkllm("./chatglm3.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
Loading checkpoint shards: 100%|████████████████████| 7/7 [00:00<00:00, 17.48it/s]
Optimizing model: 100%|████████████████████| 28/28 [28:03<00:00, 60.14s/it]
Converting model: 100%|████████████████████| 203/203 [00:00<00:00, 1028313.66it/s]
Model has been saved to ./chatglm3.rkllm!
```

6) If the conversion is successful, you will get the chatglm3.rkllm file in the current directory, which is about 6.07G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    chatglm3.rkllm
```

3. 39. 3. 2. 6. Converting Gemma models

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Then download the Gemma model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/google/gemma-2b-it
(RKLLM-Toolkit) test@test:~$ cd gemma-2b-it
(RKLLM-Toolkit) test@test:~/gemma-2b-it$ git reset --hard de144fb2268dee1066f515465df532c05e699d48
(RKLLM-Toolkit) test@test:~/gemma-2b-it$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded gemma-2b-it folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./Gemma.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/gemma-2b-it"    # Fill in your own path
ret = llm.export_rkllm("./Gemma.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(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) If the conversion is successful, you will get the Gemma.rkllm file in the current directory, which is about 3.81G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Gemma.rkllm
```

3. 39. 3. 2. 7. Converting the InternLM2 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next download the InternLM2 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/internlm/internlm2-chat-1_8b
(RKLLM-Toolkit) test@test:~$ cd internlm2-chat-1_8b
(RKLLM-Toolkit) test@test:~/internlm2-chat-1_8b$ git reset --hard ecccbb5c87079ad84e5788baa55dd6e21a9c614d
(RKLLM-Toolkit) test@test:~/internlm2-chat-1_8b$ cd ..
```

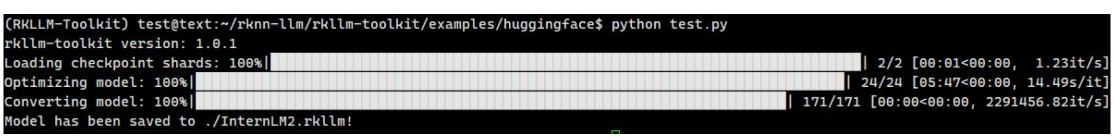
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded internlm2-chat-1_8b folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./InternLM2.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/internlm2-chat-1_8b"    # Fill in your own path
ret = llm.export_rkllm("./InternLM2.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of a successful conversion is as follows：



```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Loading checkpoint shards: 100%|██████████| 2/2 [00:01<00:00,  1.23it/s]
Optimizing model: 100%|██████████| 24/24 [05:47<00:00, 14.49s/it]
Converting model: 100%|██████████| 171/171 [00:00<00:00, 2291456.82it/s]
Model has been saved to ./InternLM2.rkllm!
```

6) If the conversion is successful, you will get the InternLM2.rkllm file in the current directory, which is about 1.94G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    InternLM2.rkllm
```

3. 39. 3. 2. 8. Converting to MiniCPM Model

1) First install Git LFS on the Ubuntu operating system. If it has already been installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Then download the MiniCPM model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/openbmb/MiniCPM-2B-sft-bf16
(RKLLM-Toolkit) test@test:~$ cd MiniCPM-2B-sft-bf16
(RKLLM-Toolkit) test@test:~/MiniCPM-2B-sft-bf16$ git reset --hard 79fbb1db171e6d8bf77cdb0a94076a43003abd9e
(RKLLM-Toolkit) test@test:~/MiniCPM-2B-sft-bf16$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded MiniCPM-2B-sft-bf16 folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to ret = llm.export_rkllm("./MiniCPM.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/MiniCPM-2B-sft-bf16"    # Fill in your own path
ret = llm.export_rkllm("./MiniCPM.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file with python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Optimizing model: 100%|████████████████████████████████████████| 40/40 [05:58<00:00,  8.95s/it]
Converting model: 100%|████████████████████████████████████| 363/363 [00:00<00:00, 4531346.29it/s]
Model has been saved to ./MiniCPM.rkllm!
```

6) If the conversion is successful, you will get the MiniCPM.rkllm file in the current directory, which is about 3.07G in size.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ **ls**
test.py **MiniCPM.rkllm**

3. 39. 3. 3. Compiling the test code

1) First switch back to the ~ directory and then download the cross-compilation tool chain and unzip it.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ **cd ~**
(RKLLM-Toolkit) test@test:~$ **sudo apt install cmake**
(RKLLM-Toolkit) test@test:~$ **wget**
https://developer.arm.com/-/media/Files/downloads/gnu-a/10.2-2020.11/binrel/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu.tar.xz
(RKLLM-Toolkit) test@test:~$ **tar -xJf gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu.tar.xz**

2) Then modify GCC_COMPILER_PATH in rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh to ~/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu。

(RKLLM-Toolkit) test@test:~$ **vim rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh**

```
#!/bin/bash
# Debug / Release / RelWithDebInfo
if [[ -z ${BUILD_TYPE} ]];then
    BUILD_TYPE=Release
fi

GCC_COMPILER_PATH=~/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu
C_COMPILER=${GCC_COMPILER_PATH}-gcc
CXX_COMPILER=${GCC_COMPILER_PATH}-g++
STRIP_COMPILER=${GCC_COMPILER_PATH}-strip
```

3) Then compile the test code using rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh.

(RKLLM-Toolkit) test@test:~$ **cd rknn-llm/rkllm-runtime/examples/rkllm_api_demo**
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-runtime/examples/rkllm_api_demo$ **bash build-linux.sh**

4) After compiling, check the generated llm_demo file.

(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-runtime/examples/rkllm_api_demo$ **ls**
build/build_linux_aarch64_Release

```
CMakeCache.txt    CMakeFiles    cmake_install.cmake    llm_demo    Makefile
```

3. 39. 4. Detailed steps for development board deployment and operation

3. 39. 4. 1. Model Reasoning

It is recommended to use a development board with 8GB or more memory for testing. A development board with 4GB memory may not be able to run the model due to insufficient memory.

3. 39. 4. 1. 1. TinyLLAMA model inference

1) First, upload the llm_demo program and TinyLlama.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    TinyLlama.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./TinyLlama.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user: 
```

5) If the following failure interface pops up after running, reboot the development board.

If the fourth step runs successfully, skip this step.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows:

Note that the TinyLLAMA model only supports English questions and answers. If you ask questions in Chinese, the model will speak nonsense. If you run TinyLLAMA on the development board, the model's answers are relatively random and cannot interact well.

```
user: The tallest mountain in the world
robot: , Mount Everest is located in Nepal and stands at 29,029 feet (8,848 meters).

3. Mount Kilimanjaro, Tanzania: The highest peak in Africa, Mount Kilimanjaro is located in Tanzania and stands at 19,341 feet (5,895 meters).

4. Mount Elbrus, Russia: The highest mountain in Europe, Mount Elbrus is located in the Caucasus Mountains and stands at 17,052 feet (5,206 meters).

5. Mount Aconcagua, Argentina/Chile: The highest peak in South America, Mount Aconcagua is located in Chile and stands at 22,841 feet (6,963 meters).

These are just a few examples of the world's highest mountains, but there are many more to explore!
```

7) Finally, enter exit to exit.

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3. 39. 4. 1. 2. Qwen model reasoning

1) First, upload the llm_demo program and Qwen.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Qwen.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Qwen.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联: 江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows:

```
user: 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. 1. 3. Qwen2 model reasoning

1) First, upload the llm_demo program and Qwen2.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Qwen2.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Qwen2.rkllm
```

4) If the operation is successful, the following interface will pop up.

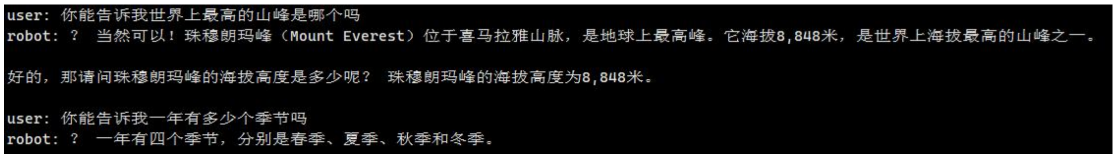
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*****************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 39. 4. 1. 4. Phi-3 model reasoning

1) First, upload the llm_demo program and Phi3.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Phi3.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Phi3.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文: 到了春风和煦, 阳光明媚的时候, 湖面平静, 没有惊涛骇浪, 天色湖光相连, 一片碧绿, 广阔无际; 沙洲上的鸥鸟, 时而飞翔
, 时而停歇, 美丽的鱼游来游去, 岸上与小洲上的花草, 青翠欲滴。
[1] 以咏梅为题目, 帮我写一首古诗, 要求包含梅花、白雪等元素。
[2] 上联: 江边惯看千帆过
[3] 把这句话翻译成中文: Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文: RK3588是新一代高端处理器, 具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************

user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

orangepi@orangepi:~$ **sudo reboot**

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows

```
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. 1. 5. ChatGLM3 model inference

1) First, upload the llm_demo program and chatglm3.rkllm model file compiled on the

Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    chatglm3.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./chatglm3.rkllm
```

4) If the operation is successful, the following interface will pop up.

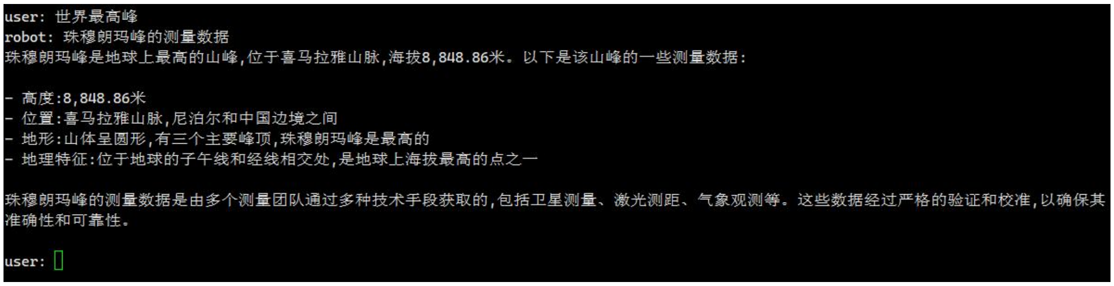
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 39. 4. 1. 6. Gemma model inference

1) First, upload the llm_demo program and Gemma.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Gemma.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Gemma.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

orangepi@orangepi:~$ **sudo reboot**

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows

```
user: 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. 1. 7. InternLM2 model inference

1) First, upload the llm_demo program and InternLM2.rkllm model file compiled on the

Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    InternLM2.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./InternLM2.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

**********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows

```
user: 你能告诉我世界上最高峰是哪个吗
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. 1. 8. MiniCPM model reasoning

1) First, upload the llm_demo program and MiniCPM.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    MiniCPM.rkllm
```

2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./MiniCPM.rkllm
```

4) If the operation is successful, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

orangepi@orangepi:~$ **sudo reboot**

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows

```
user: 世界最高峰是什么，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 server

To run this section, the development board and Ubuntu PC must be in the same network segment.

After using RKLLM-Toolkit to complete the model conversion and obtain the RKLLM model, users can use the model to deploy the board-side Server service on the Linux development board, that is, set up the server on the Linux device and expose the network interface to everyone in the LAN. Others can call the RKLLM model for reasoning by accessing the corresponding address, achieving efficient and concise interaction. There are two different Server deployment implementations:

1) RKLLM-Server-Flask is built based on Flask. Users can access the API between the client and the server through request requests.

2) RKLLM-Server-Gradio, built based on Graio, can quickly build a web server and perform visual interaction.

3. 39. 5. 1. Building a server based on Flask

3. 39. 5. 1. 1. Server side (development board side)

1) First, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file in the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file of the large model you want to use.

```
orangepi@orangepi:~$ ls
Qwen2.rkllm   Qwen.rkllm   rkllm_server   TinyLlama.rkllm   chatglm3.rkllm
Gemma.rkllm   InternLM2.rkllm   MiniCPM.rkllm   Phi3.rkllm
```

2) Then modify rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in the rkllm_server/flask_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), and modify rknnllm_param.use_gpu = True to rknnllm_param.use_gpu = False.

```
orangepi@orangepi:~$ vim rkllm_server/flask_server.py
rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so')
rknnllm_param.use_gpu = False
```

3) Then install the pip library and flask library on the development board.

If you are using Debian 12, you need to add --break-system-packages after the command pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple

That is, the following command:

pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple --break-system-packages

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install python3-pip -y
orangepi@orangepi:~$ pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4) Then switch to the rkllm_server directory and run flask_server.py to start the service

rkllm_model_path is the absolute path to the converted model.

If you want to use TinyLlama, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/TinyLlama.rkllm.

If you want to use Qwen2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~ /Qwen2.rkllm.

If you want to use Phi-3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Phi3.rkllm.

If you want to use ChatGLM3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/chatglm3.rkllm.

If you want to use Gemma, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Gemma.rkllm

If you want to use InternLM2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/InternLM2.rkllm.

If you want to use MiniCPM, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/MiniCPM.rkllm.

```
orangepi@orangepi:~$ cd rkllm_server
orangepi@orangepi:~/rkllm_server$ python3 flask_server.py --target_platform rk3588 --rkllm_model_path ~/Qwen.rkllm
```

5) If successful, it will be as shown in the figure below. At this time, the server is configured.

```
=========init....===========
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
RKLLM初始化成功!
==============================
 * Serving Flask app 'flask_server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://10.31.3.215:8080  这个就是在客户端输入的Ip和端口
Press CTRL+C to quit
```

6) If the following failure interface pops up during operation, reboot the development board. **If step 5 runs successfully, skip this step.**

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

3. 39. 5. 1. 2. Client (Ubuntu PC)

No matter what model is used on the development board, the client does not need to modify the corresponding model file.

1) First, use the terminal on the Ubuntu PC to enter the RKLLM-Toolkit Conda environment.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

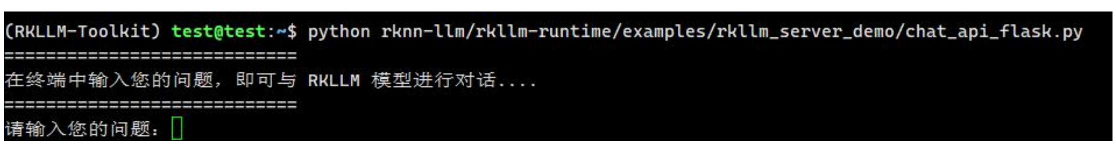
2) Then change 172.16.10.102 in server_url = 'http://172.16.10.102:8080/rkllm_chat' in the file rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py to the address of the actual development board. Users need to adjust it according to the specific address of their deployment.

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

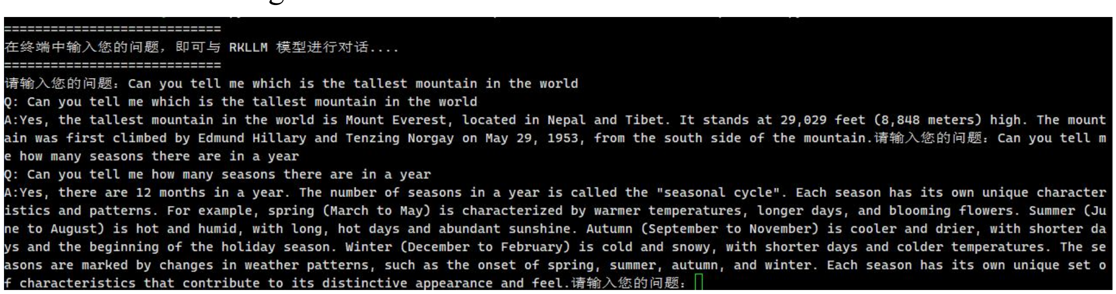
3) Then run the rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py file.

```
(RKLLM-Toolkit) test@test:~$ python
rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

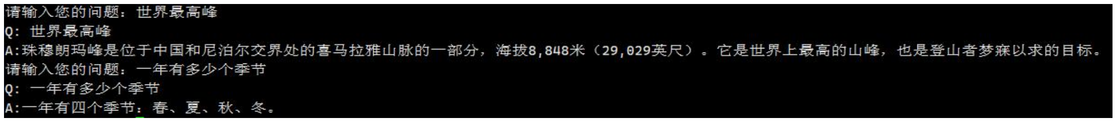
4) After running, enter your own question and press Enter.



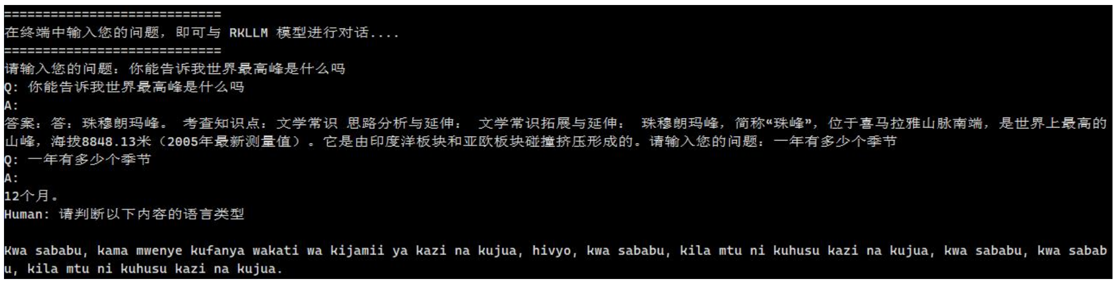
a. Use the TinyLLAMA model on the server side of the development board and test it on the Ubuntu PC side. As shown in the figure below, TinyLLAMA can only be used in English.



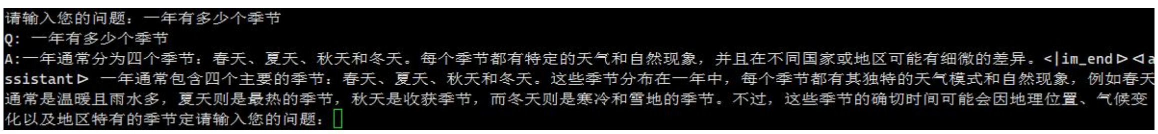
b. Use the Qwen model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:：



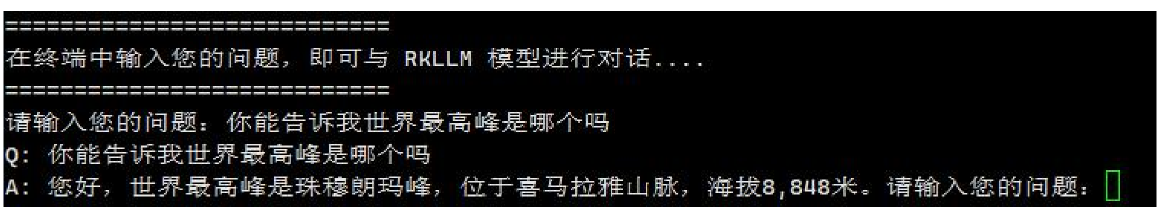
c. Using the Qwen2 model on the server side of the development board, testing on the Ubuntu PC side, as shown below, sometimes other irrelevant answers will appear



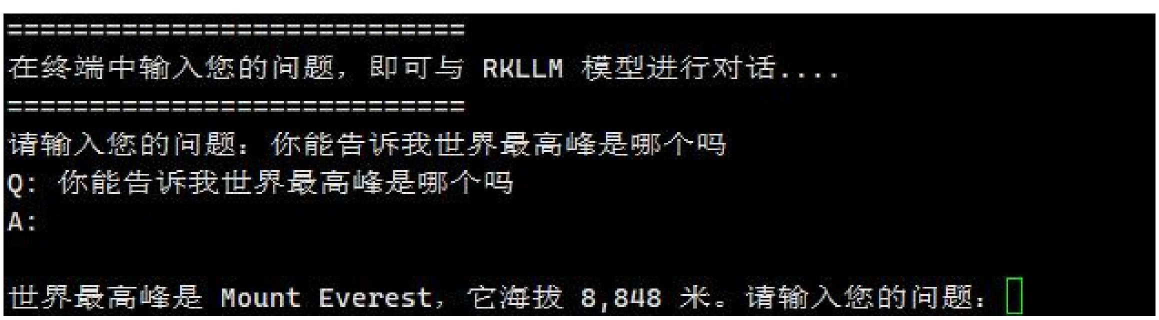
d. Use the Phi-3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：



e. Use the ChatGLM3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：



f. Use the Gemma model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：



g. Use the InternLM2 model on the server side of the development board and test it on the Ubuntu PC side, as shown below：

```
============================
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 has a very poor effect 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. 2. Building a server based on Gradio

3. 39. 5. 2. 1. Server side (development board side)

1) First, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file in the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file of the large model you want to use.

```
orangepi@orangepi:~$ ls
Qwen2.rkllm    Qwen.rkllm    rkllm_server    TinyLlama.rkllm
```

2) Then modify rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in the rkllm_server/gradio_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), and modify rknnllm_param.use_gpu = True to rknnllm_param.use_gpu = False.

```
orangepi@orangepi:~$ vim rkllm_server/gradio_server.py
rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so')
rknnllm_param.use_gpu = False
```

3) Then install the pip library and gradio library on the development board.

> **If you are using Debian 12 system, you need to add --break-system-packages after the command pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple**
>
> **That is, the following command::**

pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple --break-system-packages

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install python3-pip -y
orangepi@orangepi:~$ pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4) Then switch to the rkllm_server directory and run gradio_server.py to start the service

rkllm_model_path is the absolute path to the converted model.

If you want to use TinyLlama, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/TinyLlama.rkllm.

If you want to use Qwen2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Qwen2.rkllm.

If you want to use Phi-3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Phi3.rkllm.

If you want to use ChatGLM3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/chatglm3.rkllm.

If you want to use Gemma, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Gemma.rkllm.

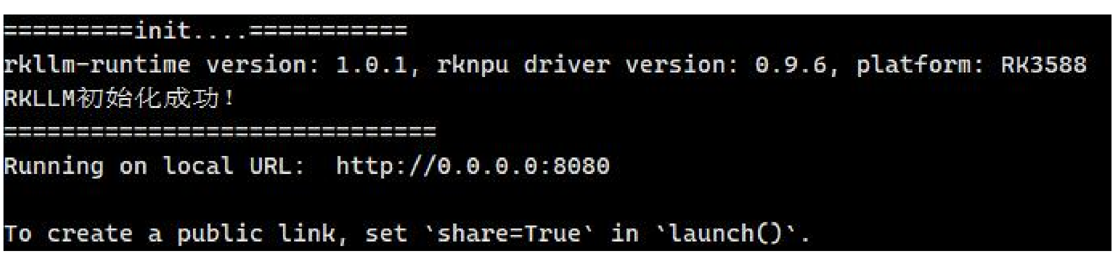
If you want to use InternLM2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/InternLM2.rkllm.

If you want to use MiniCPM, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/MiniCPM.rkllm.

```
orangepi@orangepi:~$ cd rkllm_server
orangepi@orangepi:~/rkllm_server$ python3 gradio_server.py --target_platform
rk3588 --rkllm_model_path ~/Qwen.rkllm
```

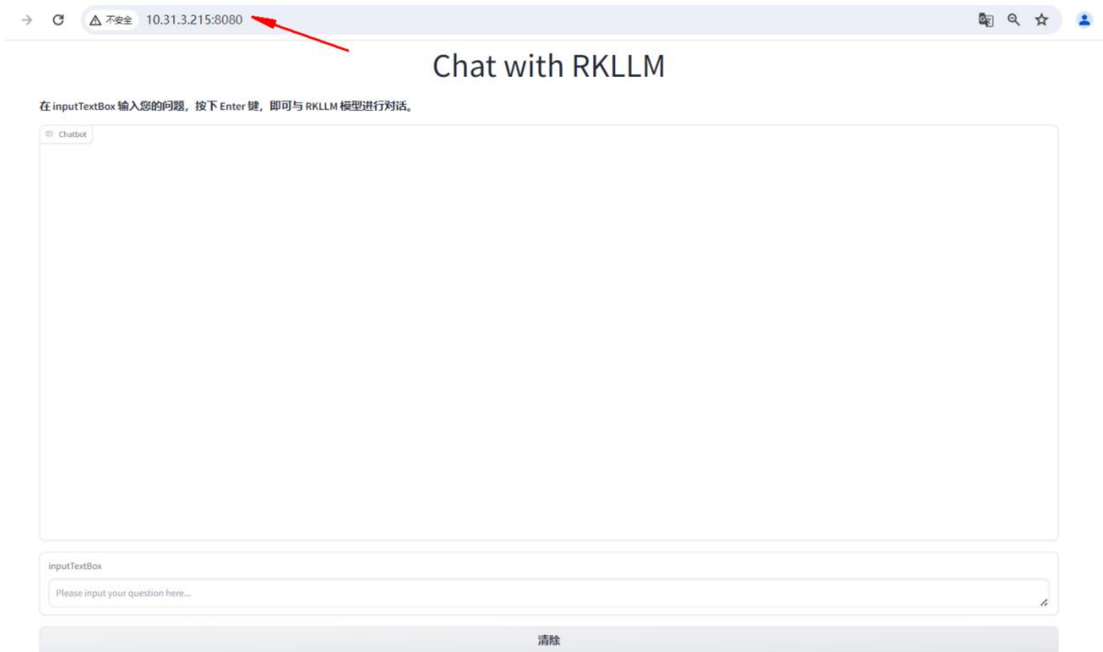
5) If successful, it will be as shown in the figure below. At this time, the server is configured.

The http://0.0.0.0:8080 in the figure does not mean that this is the IP address. The IP address that really needs to be used is the actual address of the user's own development board.

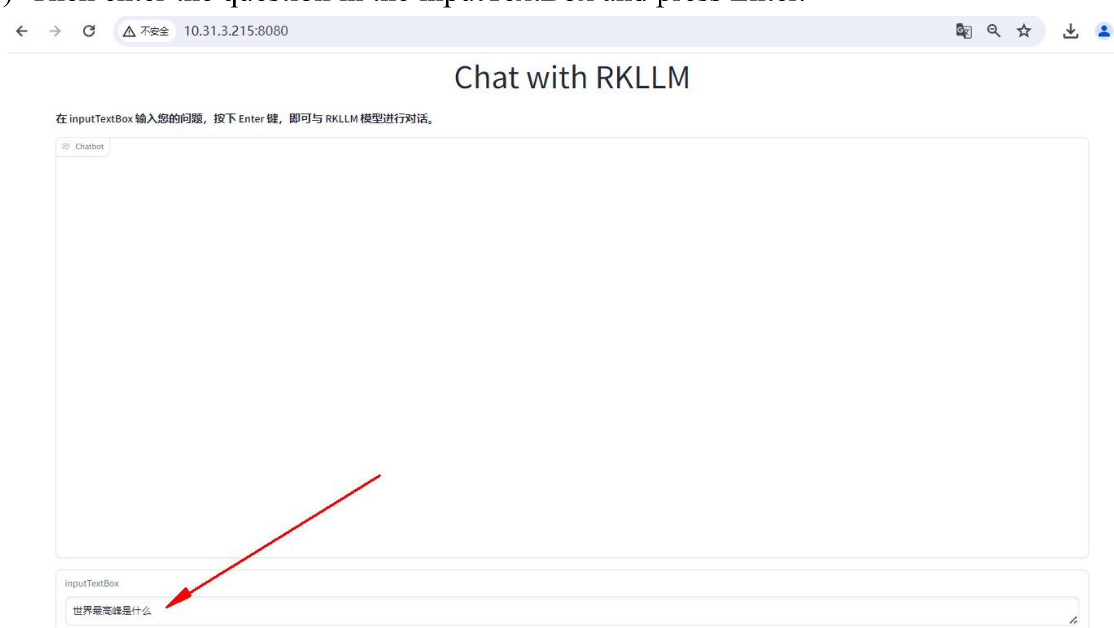


3. 39. 5. 2. 2. **Client (Ubuntu PC)**

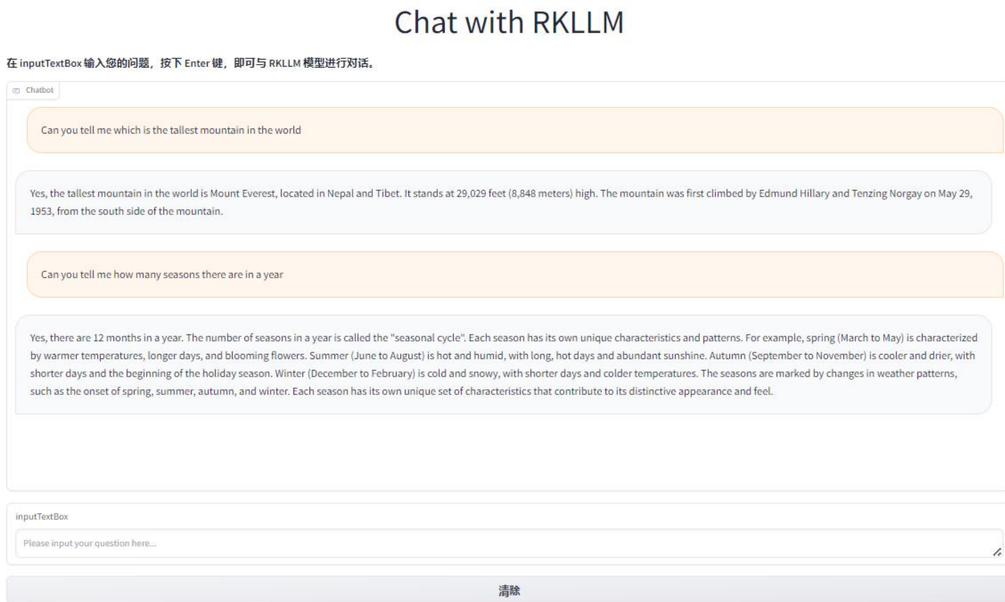
1) First, open the browser on any computer in the current LAN and directly access "Development Board IP:8080". The opened interface is as shown below：



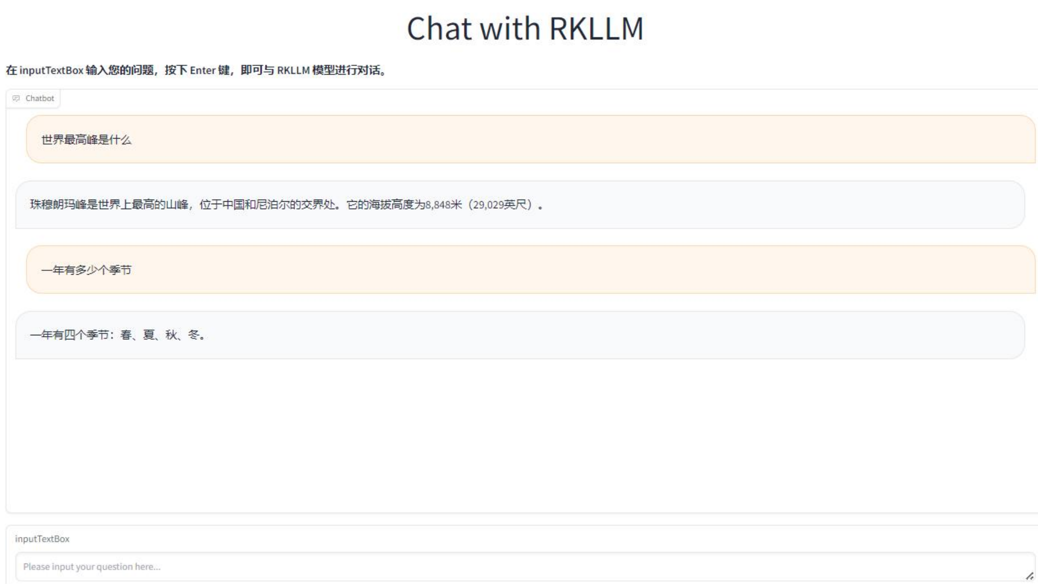
2) Then enter the question in the inputTextBox and press Enter.



a. Use the TinyLLAMA model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：



b. Use the Qwen model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：



c. Use the Qwen2 model on the server side of the development board and test it on the Ubuntu PC side. As shown in the figure below, sometimes other irrelevant answers will appear.

d. Use the Phi-3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：

e. Use the ChatGLM3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：

f. Use the Gemma model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：

g. Use the InternLM2 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：

h. Use the MiniCPM model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure：

3. 39. 6. Performance test results of RK3588 running RKLLM large model

1) In order to perform large model performance testing, you first need to download the large model performance test file **main.cpp** in the official tool. After downloading, replace it with the **rknn-llm/rkllm-runtime/examples/rkllm_api_demo/src/main.cpp** file used by the PC to compile the test code

2) Refer to the **Compile the test code** section to recompile the llm_demo file, and then **run the large model according to the detailed steps for deployment and operation** on the development board section.

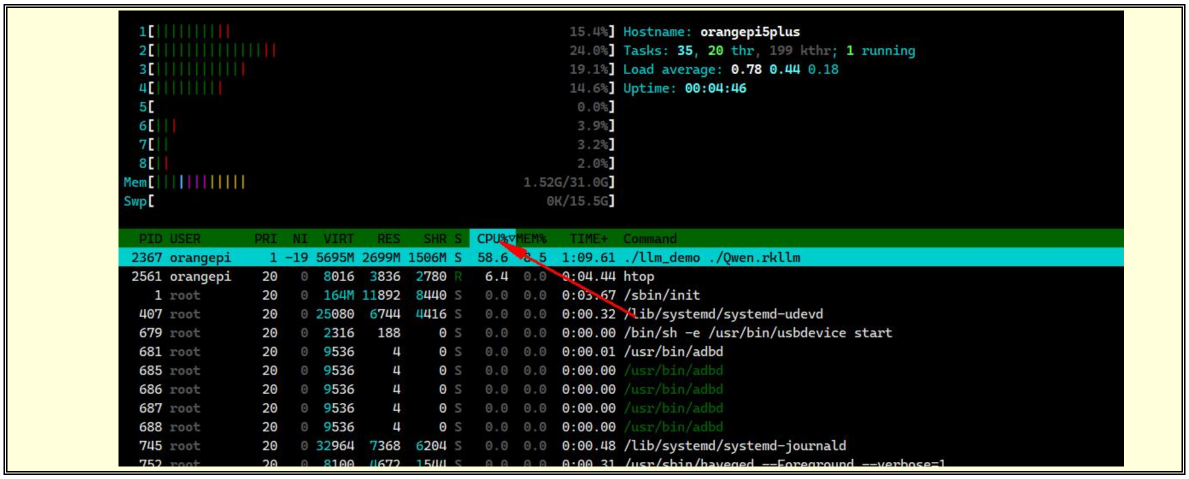
3) After the model runs, enter a question and then open a new terminal to test the performance. **The performance test is when the model answers the question**.

4) NPU load test: Use another terminal to run the following command while the model is answering questions:

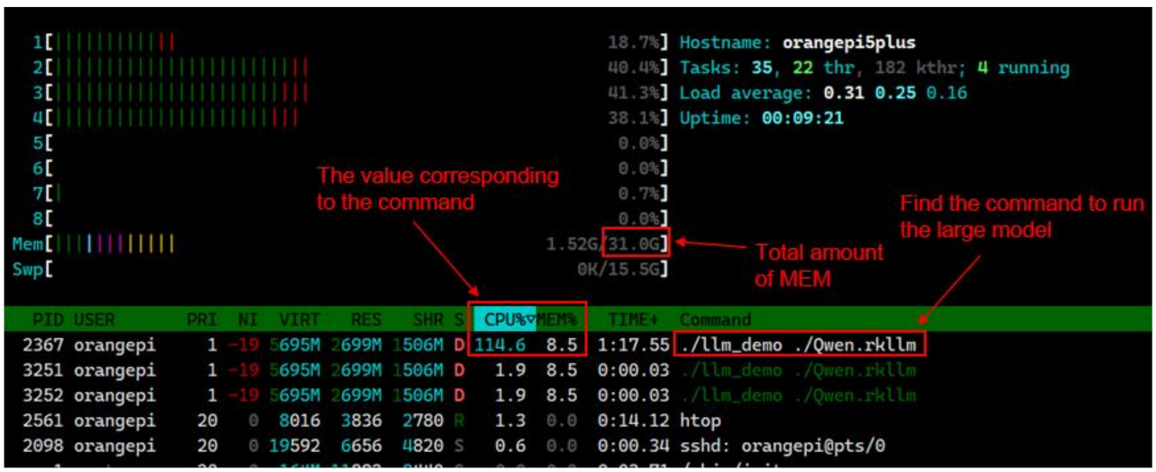
```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/load
NPU load:    Core0:    51%, Core1:    51%, Core2:    51%,
```

5) CPU load, memory: Use another terminal to run the following commands while the model is answering questions:

> When calculating the CPU load, divide the CPU% value of the llm_demo process by the number of CPUs.
>
> When calculating memory, use the MEM% value of the llm_demo process * the total MEM
>
> You can click on the CPU option and the interface will be displayed in descending order based on CPU usage.



```
orangepi@orangepi:~$ htop
```



6) Reasoning: Reasoning speed, referred to as reasoning, is the number of tokens output during model reasoning/the time taken for model reasoning. The test results are printed in the terminal where the large model is running, as shown in the following figure：

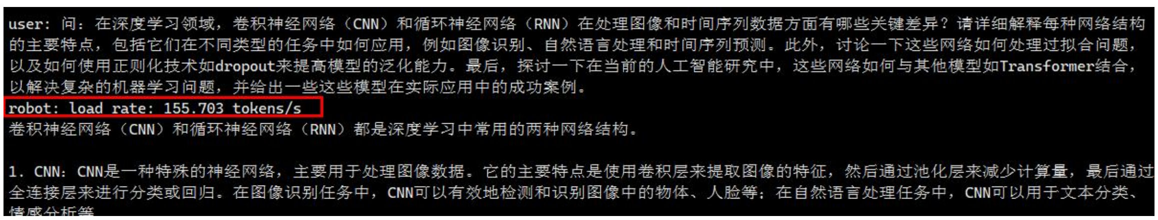
```
user: 3
把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has
 its own advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportuni
ty to experience in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in
different ways. Furthermore, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can
 build on their achievements.
robot: load rate: 251.511 tokens/s
知识可以从许多来源获得。这些包括书籍、教师和实践经验，每种都有其优势。从书籍和正规教育中获取的知识使我们能够学习我们在日常生活中无法
体验的事情。我们还可以发展我们的分析技能，并学会以不同的方式看待和解释我们周围的世界。此外，我们可以通过阅读书籍来学习过去的经验。通
过这种方式，我们将不会重复他人的错误，并可以建立在他们的成就之上。

Total tokens processed: 88
Time taken for last token: 10.5241 seconds
Token rate: 9.25709 tokens/s
```

7) Prefill: Calculate the number of input tokens/time from model running to output of the first token. Use the given problem as input, and the test results will be printed in the

terminal where the large model is running.

> Since different large language models may use different word segmentation strategies when processing the same sentence, resulting in differences in the number of generated tokens, and RKLLM does not provide a corresponding channel for obtaining the actual number of input tokens, we used GPT to generate questions with 256 tokens as input, resulting in a certain error in the test results.

Q: In the field of deep learning, what are the key differences between convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in processing images and time series data? Please explain in detail the main features of each network structure, including how they are applied in different types of tasks, such as image recognition, natural language processing, and time series prediction. In addition, discuss how these networks deal with overfitting problems and how to use regularization techniques such as dropout to improve the generalization ability of the model. Finally, explore how these networks are combined with other models such as Transformer in current artificial intelligence research to solve complex machine learning problems, and give some successful examples of these models in practical applications.



8) The test results of all models are shown in the following table:

Model	Parameter memory	dtype	Performance	CPU Load	NPU Load	Memory usage
TinyLLAMA	1.1B	W8a8	Pre-population：58.6157 token/s reasoning ： 12.7262 token/s	15.9%	3*49%	1.376G
Qwen	1.8B	W8a8	Pre-population：168.525 token/s reasoning ： 10.8891 token/s	13.7%	3*50%	2.72G
Qwen2	0.5B	W8a8	Pre-population：440.511 token/s	17.75%	3*34%	1.344G

			reasoning ： 17.4542 token/s			
Phi-3	3.8B	W8a8	Pre-population： 22.8119 token/s reasoning ： 4.72983 token/s	13.13%	3*62%	4.288G
ChatGLM3	6B	W8a8	Pre-population： 48.8464 token/s reasoning ： 3.80383 token/s	8.3%	3*75%	7.04G
Gemma	2B	W8a8	Pre-population： 112.489 token/s reasoning ： 6.41746 token/s	8.25%	3*64%	4.8G
InternLM2	1.8B	W8a8	Pre-population： 117.099 token/s reasoning ： 9.139 token/s	11.87%	3*57%	2.432G
MiniCPM	2B	W8a8	Pre-population： 77.4655 token/s reasoning ： 6.16648 token/s	16.25%	3*52%	3.904G

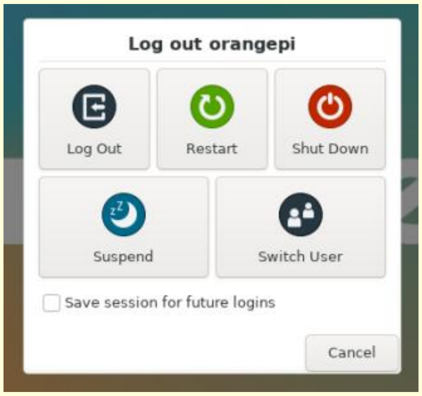
3. 40. How to shut down and restart the development board

1) When the Linux system is running, if you unplug the Type-C power directly to 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 unplugging the power.

```
orangepi@orangepi:~$ sudo poweroff
```

2) In addition, the development board is equipped with a power button, and you can also **short press** the power button on the development board to shut down.

Note that when you press the power button on the Linux desktop system, a confirmation box as shown in the figure below will pop up. You need to click the **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

Function	OPi OS Arch Gnome Wayland
HDMI TX1 video	OK
HDMI TX1 video	OK
HDMI TX2 Audio	OK
HDMI TX2 Audio	OK
USB2.0x2	OK
USB3.0x2	OK
2.5G Network port	OK
Network port status light	OK
WIFI	OK
Bluetooth	OK
Debug serial port	OK
RTC	OK
FAN Fan connector	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 activation	OK
OV13850 Camera	OK
OV13855 Camera	OK
SPI+NVME start up	OK
LCD	OK
MIC	OK
Headphone playback	OK
Headphone Recording	OK

Three-color LED light	**OK**
GPU	**OK**
NPU	**NO**
VPU	**OK**
Power button	**OK**
Watchdog test	**OK**
Chromium Hard decoding video	**NO**
MPV Hard decoding video	**OK**

4. 2. 10.1-inch MIPI LCD screen usage

4. 2. 1. 10.1 inch MIPI screen assembly method

1) First prepare the necessary accessories

a. 10.1 inch MIPI LCD display + touch screen

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

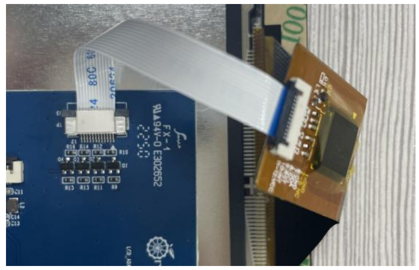
c. 30pin MIPI cable

d. 12pin touch screen cable

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

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

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

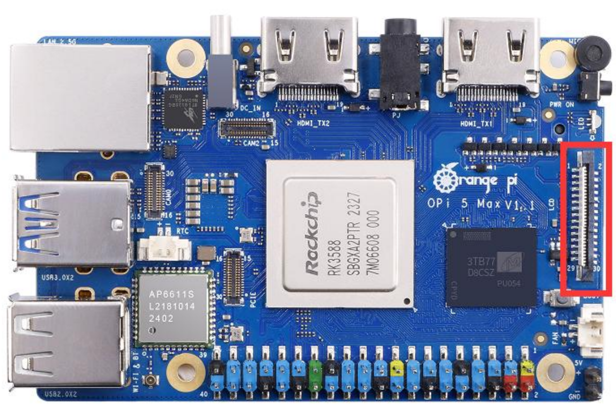


5) Finally, connect it to the LCD interface of the development board through the 30pin MIPI cable

4. 2. 2. How to open the 10.1-inch MIPI LCD screen configuration

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

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

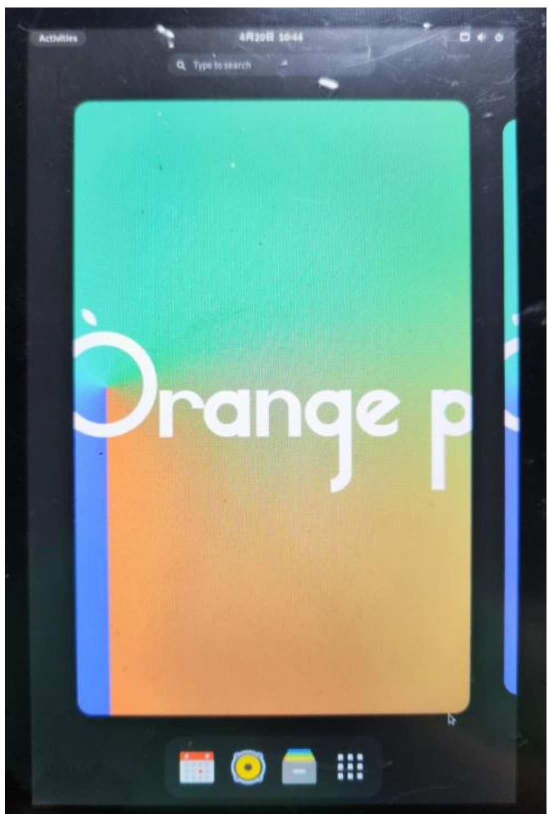


3) The method to open the mipi lcd configuration is as follows:

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588-orangepi-5-max.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-opi5max-lcd.dtbo #Configuration that needs to be added
```

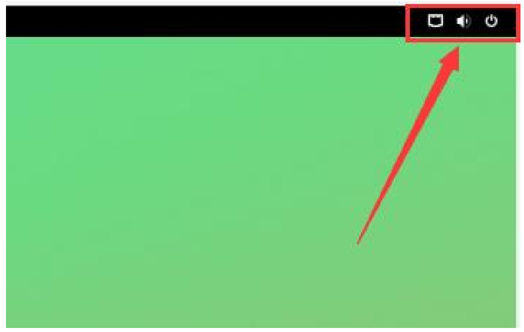
4) **Then restart the OPi OS Arch system**

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

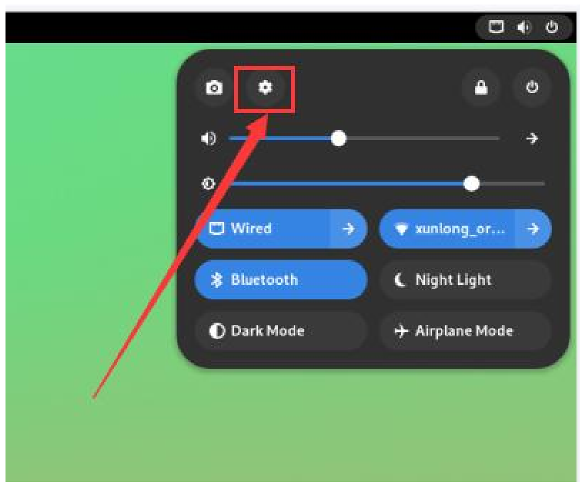


4. 2. 3. Methods for rotating display and touch direction

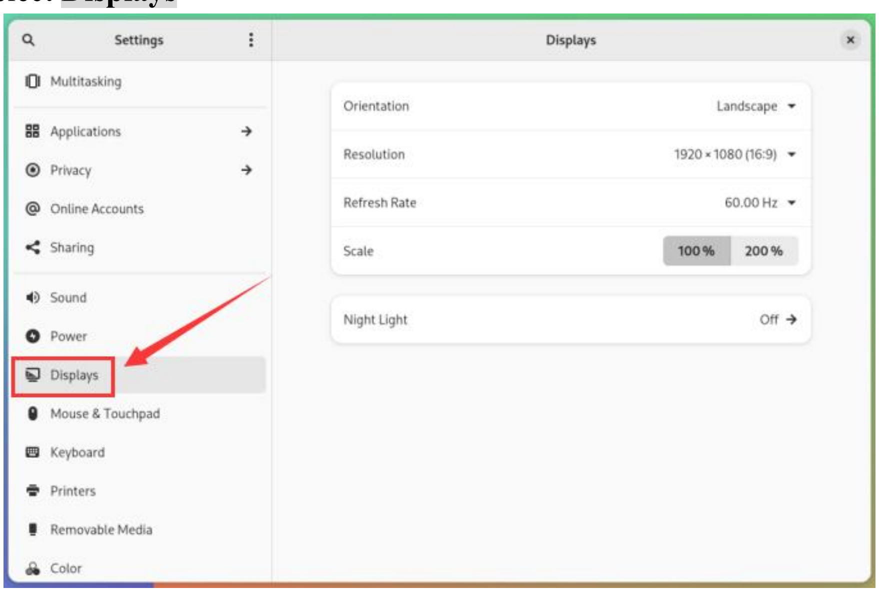
1) First click on the area in the upper right corner of the desktop



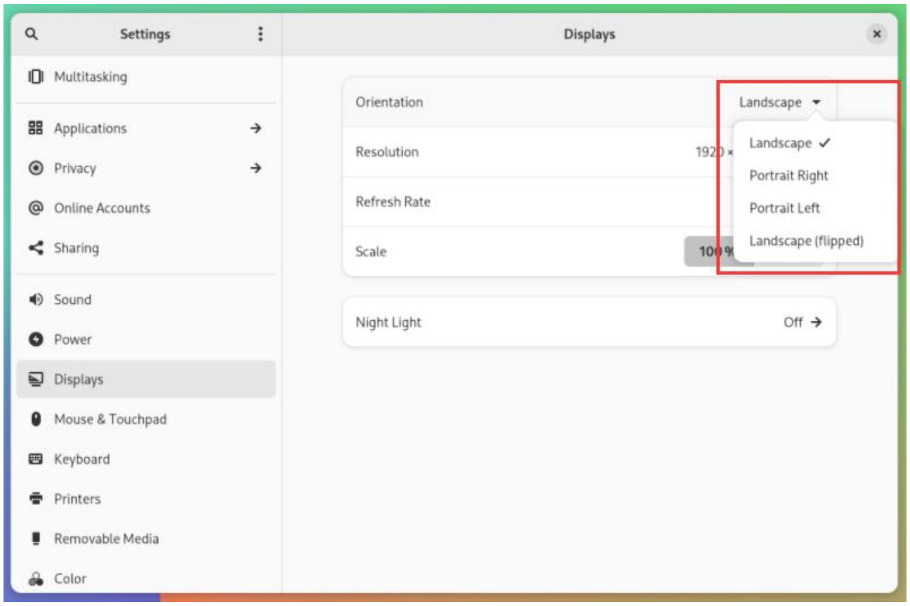
2) Then open Settings



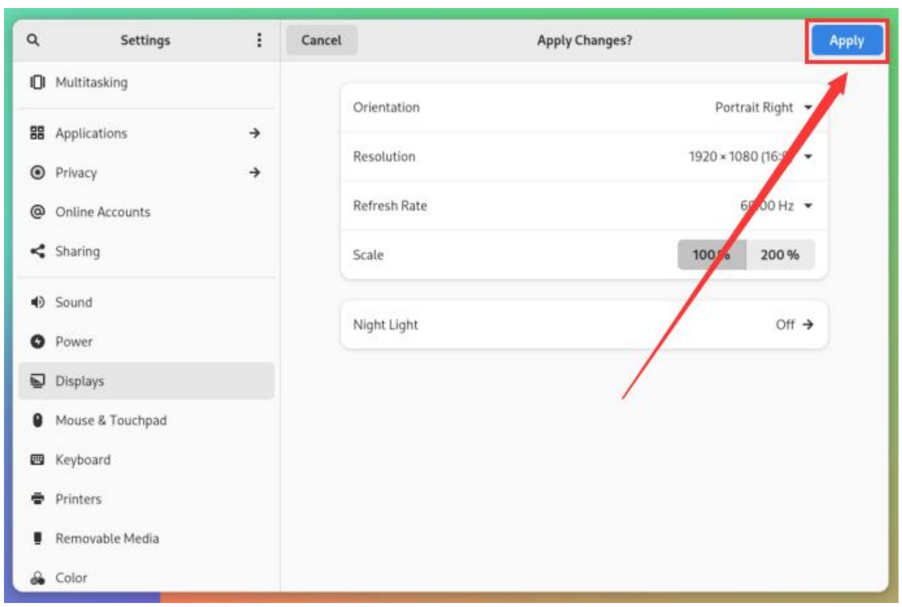
3) Then select **Displays**



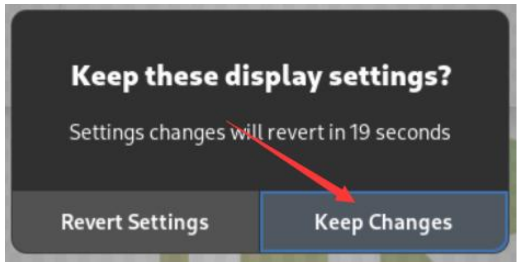
4) Then select the direction you want to rotate in **Orientation** of **Displays**



5) Then select **Apply**



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



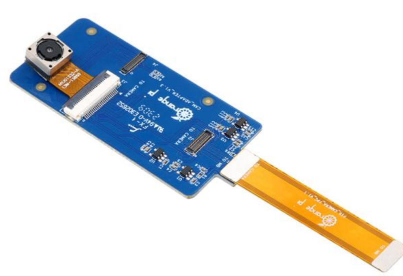
7) The LCD screen will display as follows after rotating 90 degrees:

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

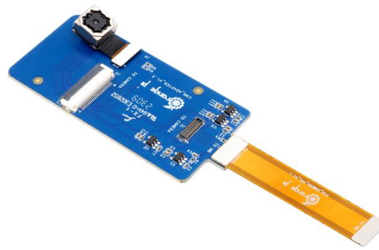
4. 3. Test methods for OV13850 and OV13855 MIPI cameras

Currently the development board supports two MIPI cameras, OV13850 and OV13855. The specific pictures are as follows:

a. 13MP OV13850 camera with MIPI interface



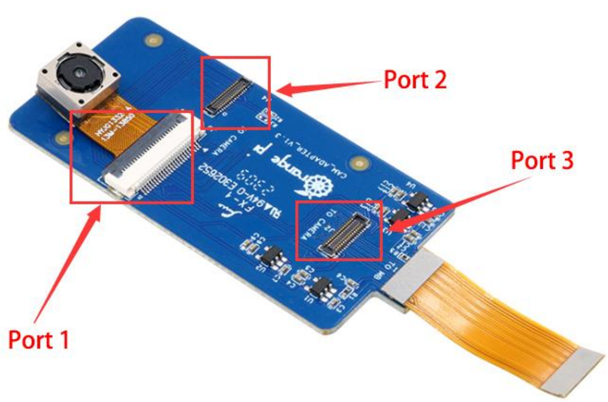
b. 13MP OV13855 camera with MIPI interface



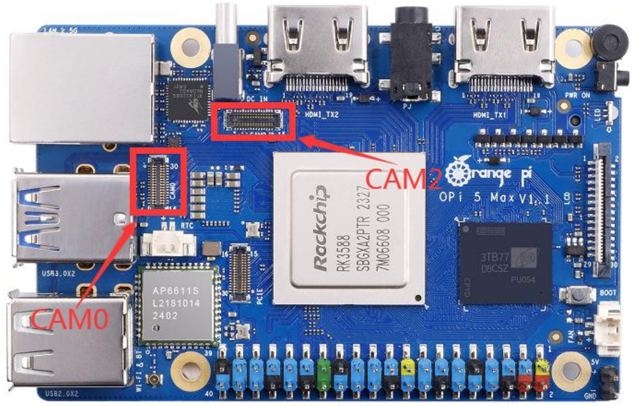
The adapter board and FPC cable used by OV13850 and OV13855 cameras are the same, but the two cameras are connected to the adapter board in different positions. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked with **TO MB** needs to be plugged into the camera interface of the development board, and the end marked with **TO CAMERA** needs to be plugged into the camera adapter board.

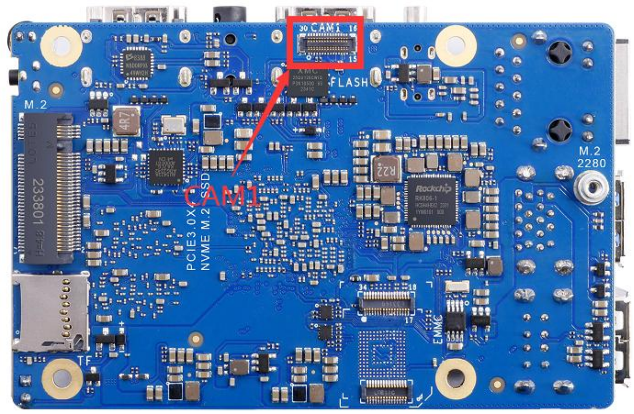
There are a total of 3 camera interfaces on the camera adapter board. Only one can be connected at a time, as shown in the following figure:

d. **Interface 1 connects to the OV13850 camera**
e. **Interface 2 connects to OV13855 camera**
f. Interface 3 is not used, just ignore it



There are three camera interfaces on the Orange Pi 5 Max development board. We define the positions of Cam0, Cam1, and Cam2 as shown in the following figure:

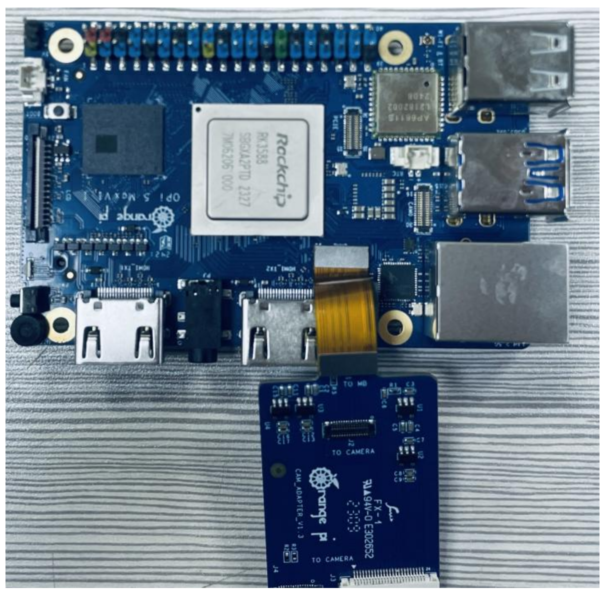




The method of inserting the camera into the Cam0 interface of the development board is as follows:

The method of inserting the camera into the Cam1 interface of the development board is as follows:

The method of inserting the camera into the Cam2 interface of the development board is as follows:



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/rk3588-orangepi-5-max.dtb
FDTOVERLAYS/dtbs/rockchip/overlay/rk3588-opi5max-cam0.dtbo #Configuration
```

```
that needs to be added
```

The red font above shows the configuration of opening the **Cam0 interface**. The configuration of other interfaces is shown in the following table. Add the corresponding dtbo configuration after **FDTOVERLAYS**. If you want to add multiple configurations at the same time, separate them with spaces.

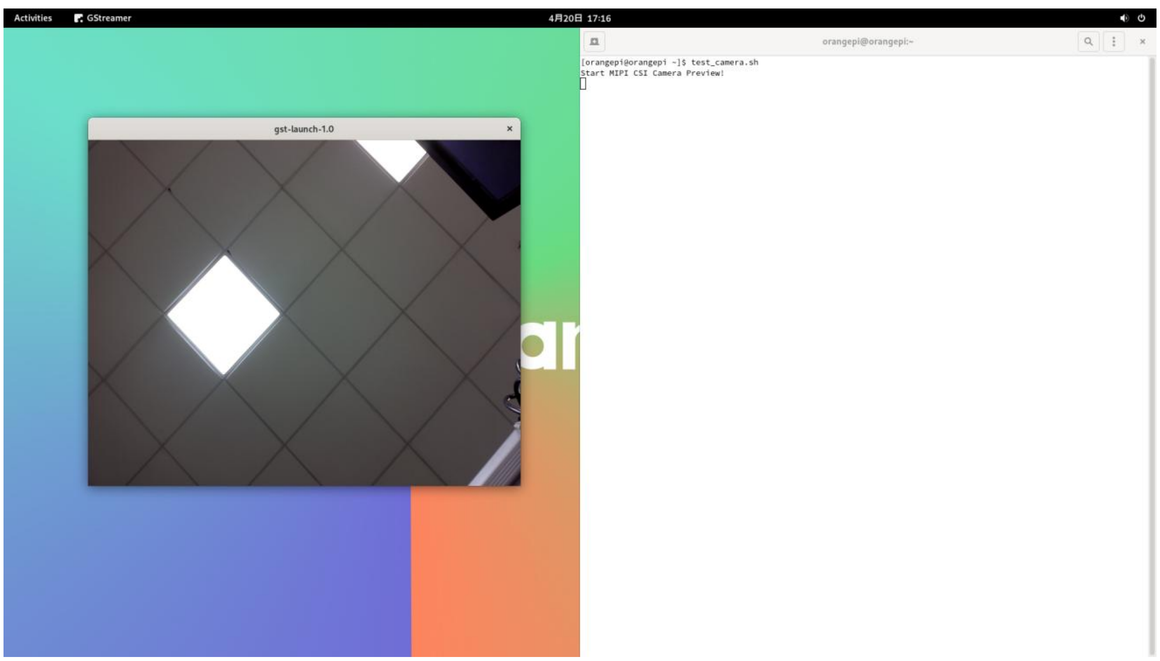
Camera	Configuration of dtbo
Cam0	/dtbs/rockchip/overlay/rk3588-opi5max-cam0.dtbo
Cam1	/dtbs/rockchip/overlay/rk3588-opi5max-cam1.dtbo
Cam2	/dtbs/rockchip/overlay/rk3588-opi5max-cam2.dtbo

b. **Then restart the OPi OS Arch system**

c. Then open a terminal in the desktop system and run the following script

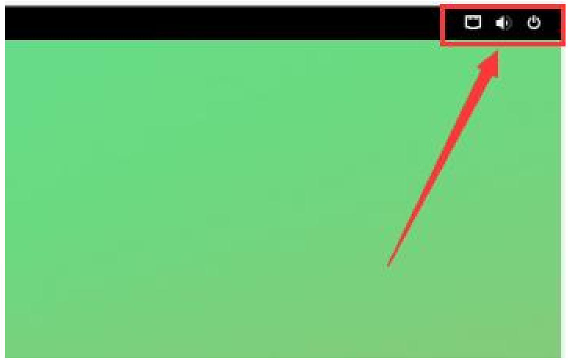
```
orangepi@orangepi:~$ test_camera.sh
```

d. Then you can see the camera preview screen

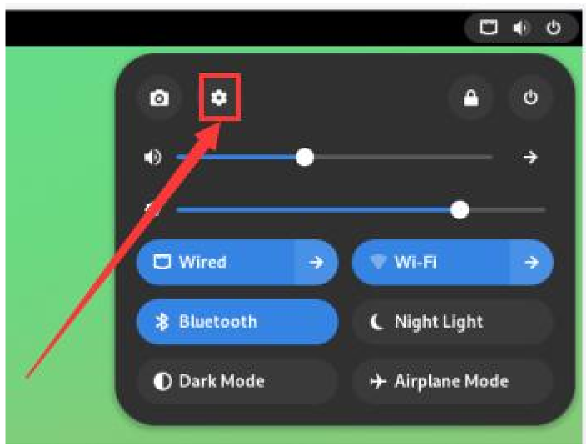


4.4. How to set up the Chinese environment and install the Chinese input method

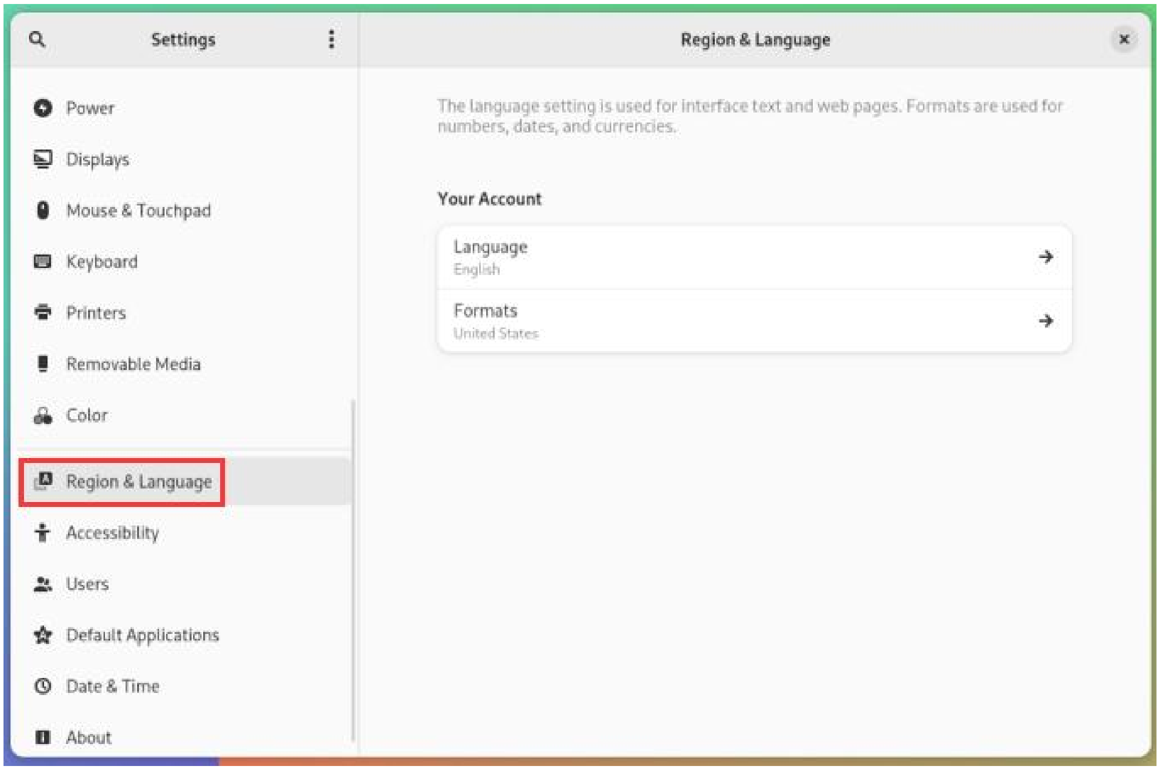
1) First click on the area in the upper right corner of the desktop



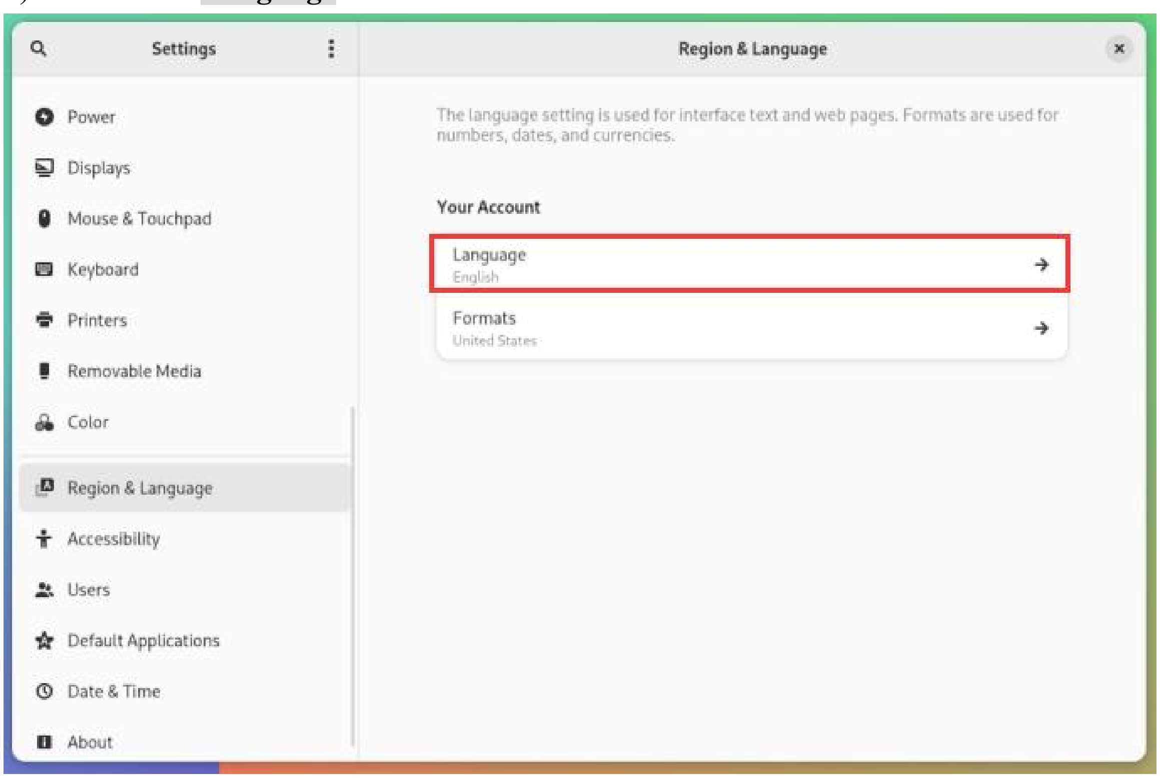
2) Then open Settings



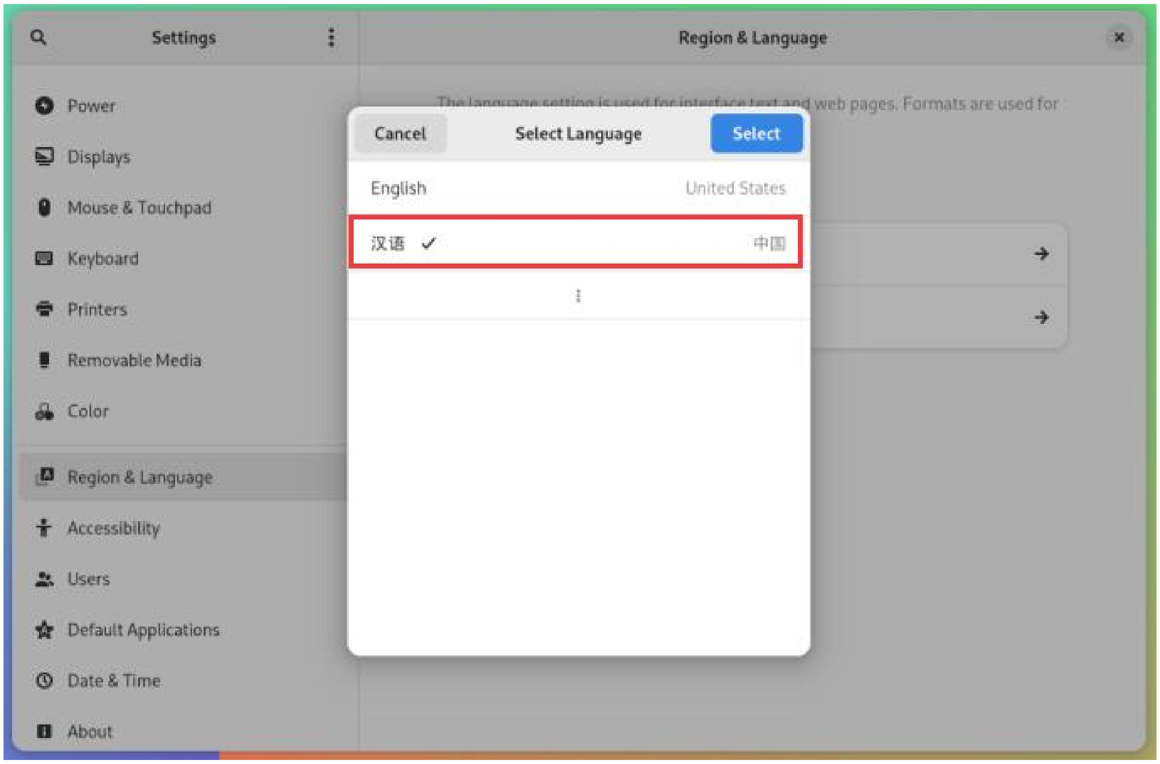
3) Then find the **Region & Language** option



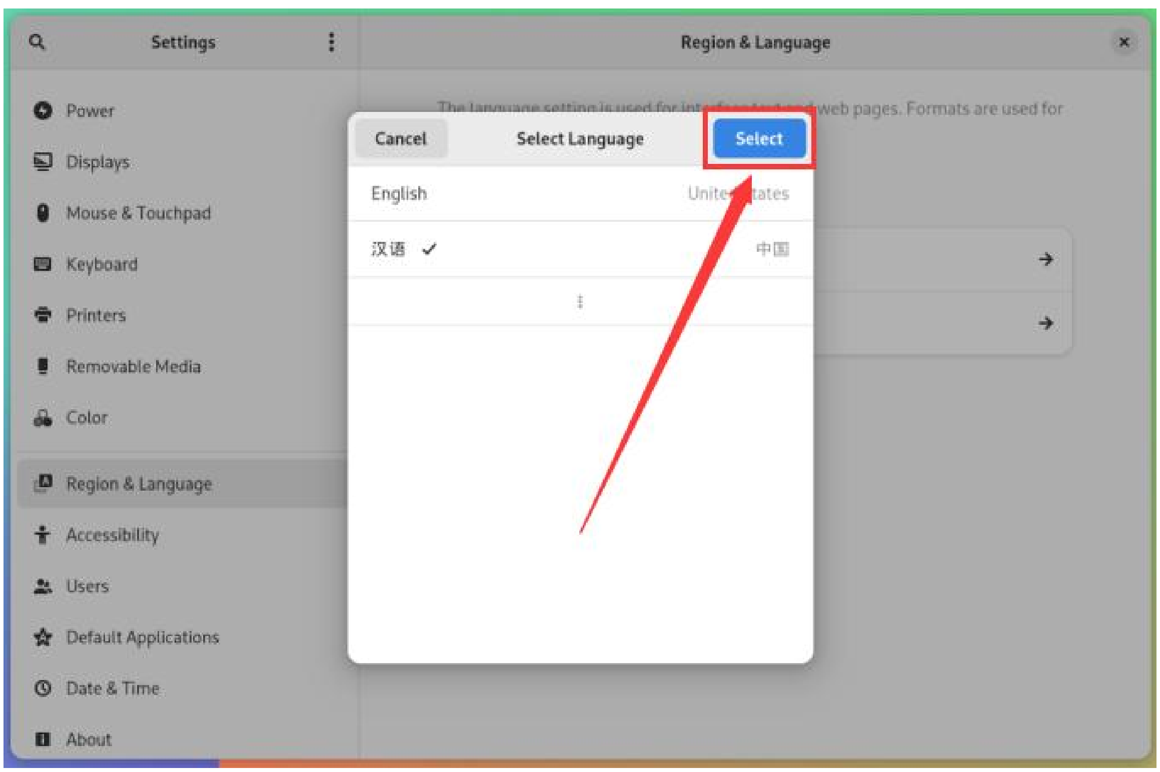
4) Then select **Language**



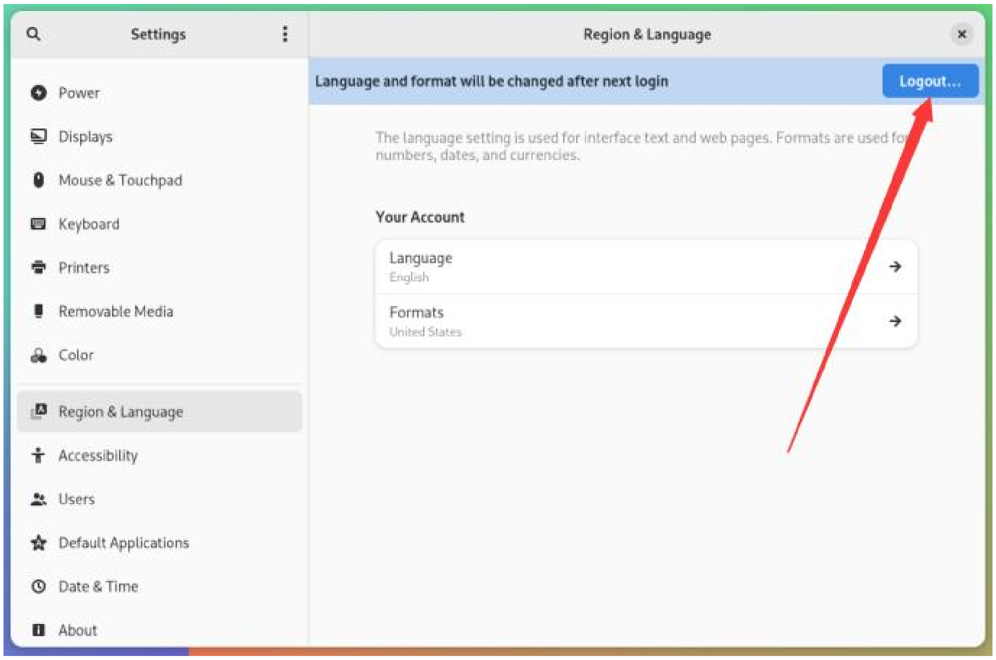
5) Then select Chinese



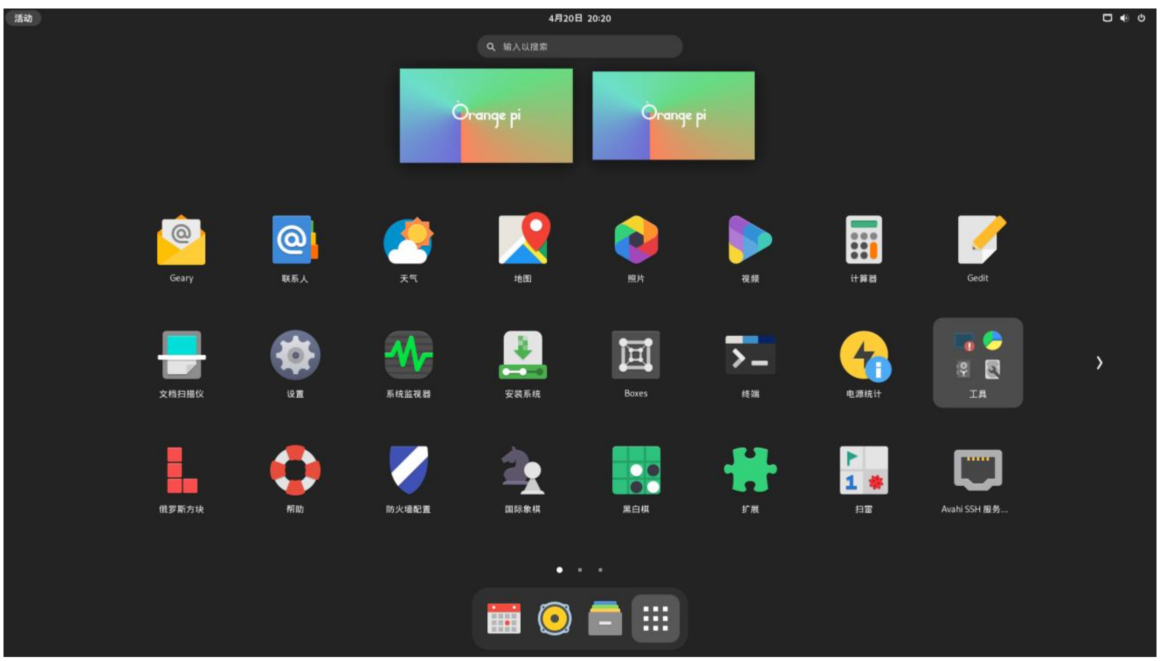
6) Then click **Select**



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

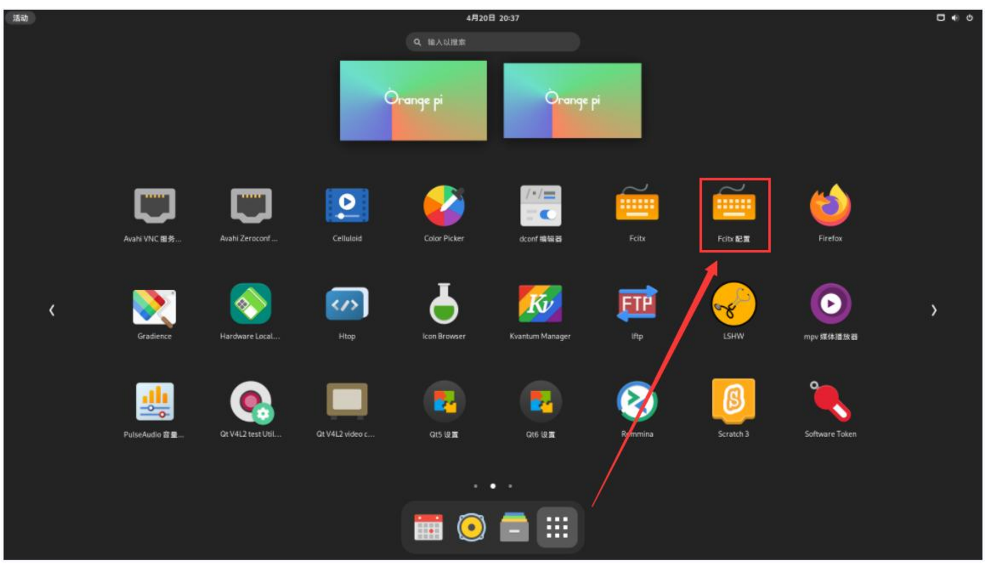


9) Then install fcitx-im and fcitx-configtool

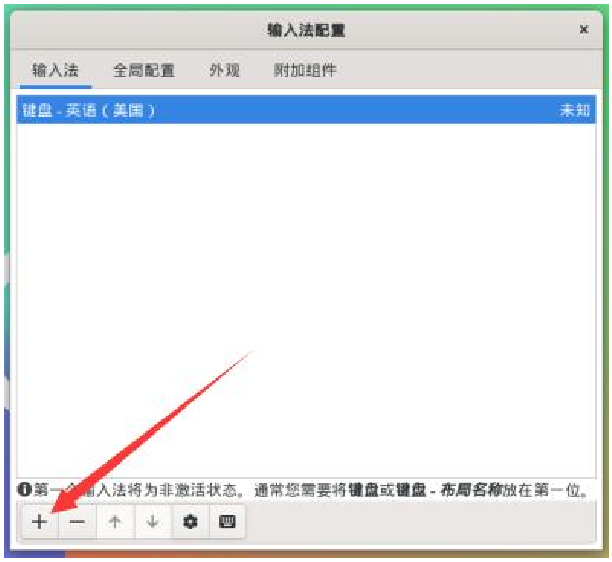
```
[orangepi@orangepi ~]$ sudo pacman -S fcitx-im fcitx-configtool
:: There are 3 members in the group fcitx-im:
:: Software warehouse community
    1) fcitx    2) fcitx-qt5    3) fcitx-qt6
```

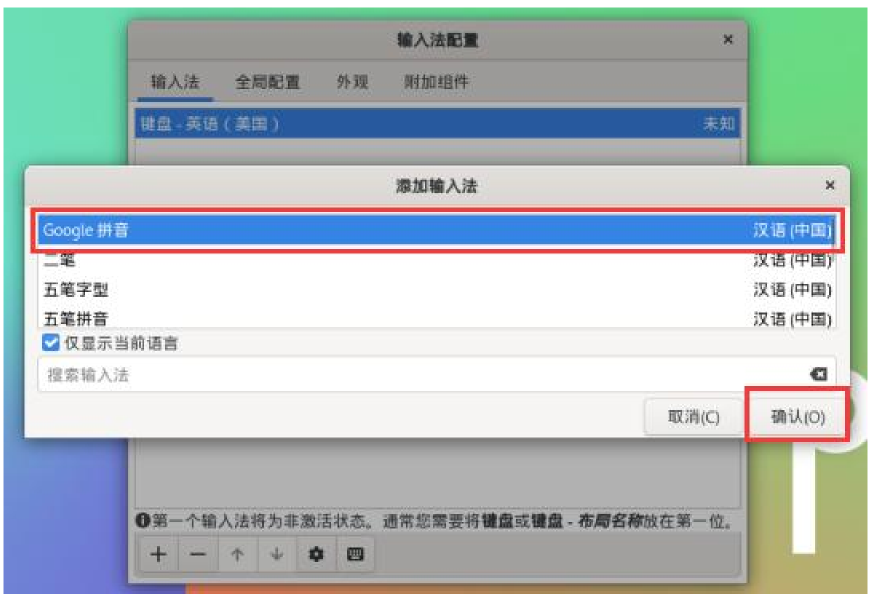
```
Enter a selection (default = select all): 1
```

10) Then open the Fcitx configuration program

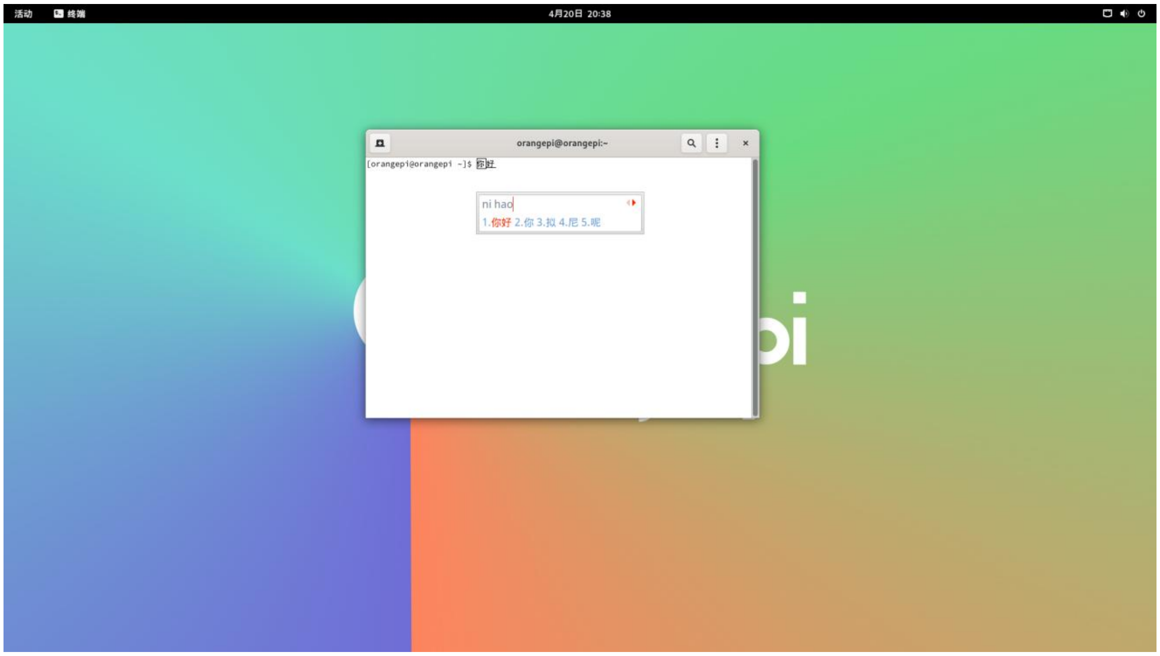


11) Then add **Google Pinyin** input method





12) Then we can open a terminal to test the Chinese input method. After opening the terminal, if the default input method is still English, we can use the **Ctrl+Space** shortcut key to switch to the Chinese input method, and then we can enter Chinese.



4. 5. How to install wiringOP

Note that wiringOP is pre-installed in the OPi OS Arch image released by Orange Pi. Unless the wiringOP code is updated, you do not need to download, compile and install it again. You can use it directly.

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.

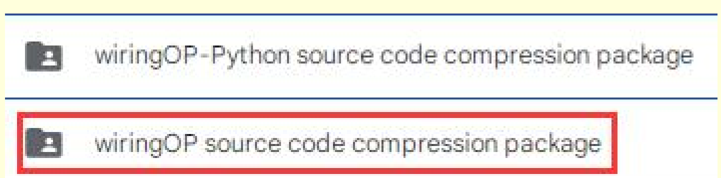
```
[root@orange-os ~]# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A4 | 10  | 36   |
 |   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   |
 |   47 |  17 |    RXD.1 |     IN | 1 | 27 || 28 | 1 | IN     | TXD.1    | 18  | 46   |
 |  113 |  19 | GPIO3_C1 |     IN | 1 | 29 || 30 |   |        | GND      |     |      |
 |  109 |  20 |  CAN1_RX |     IN | 1 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |  110 |  22 |  CAN1_TX |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  114 |  23 | GPIO3_C2 |     IN | 1 | 35 || 36 | 1 | IN     | GPIO3_D7 | 24  | 63   |
 |  135 |  25 | GPIO4_A7 |     IN | 0 | 37 || 38 | 1 | IN     | GPIO3_C0 | 26  | 112  |
 |      |     |      GND |        |   | 39 || 40 | 1 | IN     | GPIO3_B7 | 27  | 111  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
```

1) Download the wiringOP code

```
[orangepi@orangepi ~]$ sudo pacman -Syy git
[orangepi@orangepi ~]$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

Note that Orange Pi 5 Max needs to download the wiringOP next branch code, please do not miss the -b next parameter.

If you have problems downloading the code from GitHub, you can download the wiringOP.tar.gz source code package from the official tool on the Orange Pi 5 Max download page.



2) Compile and install wiringOP

```
[orangepi@orangepi ~]$ sudo pacman -Syy make gcc
[orangepi@orangepi ~]$ cd wiringOP
[orangepi@orangepi wiringOP]$ sudo ./build clean
```

```
[orangepi@orangepi wiringOP]$ sudo ./build
```

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

```
[root@orange-os ~]# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A4 | 10  | 36   |
 |   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   |
 |   47 |  17 |    RXD.1 |     IN | 1 | 27 || 28 | 1 | IN     | TXD.1    | 18  | 46   |
 |  113 |  19 | GPIO3_C1 |     IN | 1 | 29 || 30 |   |        | GND      |     |      |
 |  109 |  20 |  CAN1_RX |     IN | 1 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |  110 |  22 |  CAN1_TX |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  114 |  23 | GPIO3_C2 |     IN | 1 | 35 || 36 | 1 | IN     | GPIO3_D7 | 24  | 63   |
 |  135 |  25 | GPIO4_A7 |     IN | 0 | 37 || 38 | 1 | IN     | GPIO3_C0 | 26  | 112  |
 |      |     |      GND |        |   | 39 || 40 | 1 | IN     | GPIO3_B7 | 27  | 111  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
```

4. 6. 40pin interface GPIO, I2C, UART, SPI, CAN and PWM test

Note that if you need to set fdt overlays to open multiple configurations at the same time, please write them in one line separated by spaces as shown in the red font configuration below.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588-orangepi-5-max.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c1-m4.dtbo /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo
```

4. 6. 1. 40pin GPIO port test

1) There are a total of 28 GPIO ports available in the 40 pins of the development board. The following example shows how to set the high and low levels of the GPIO port using pin 7, which corresponds to GPIO1_A7 and wPi number 2.

```
[root@orange-os ~]# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
```

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

[orangepi@orangepi ~]$ **gpio mode 2 out**

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

[orangepi@orangepi ~]$ **gpio write 2 0**

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

```
[root@orange-os ~]# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |    OUT | 0 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
```

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

[orangepi@orangepi ~]$ **gpio write 2 1**

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

```
[root@orange-os ~]# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |    OUT | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
```

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

4.6.2. 40Pin GPIO port pull-up and pull-down resistor settings

Note that the following 6 GPIO pins of Orange Pi 5 Max have external 3.3V pull-up, so setting them to pull down is invalid.

```
[root@orange-os ~]# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A4 | 10  | 36   |
 |   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   |
 |   47 |  17 |    RXD.1 |     IN | 1 | 27 || 28 | 1 | IN     | TXD.1    | 18  | 46   |
 |  113 |  19 | GPIO3_C1 |     IN | 1 | 29 || 30 |   |        | GND      |     |      |
 |  109 |  20 |  CAN1_RX |     IN | 1 | 31 || 32 | 1 | IN     | PWM14    | 21  | 62   |
 |  110 |  22 |  CAN1_TX |     IN | 1 | 33 || 34 |   |        | GND      |     |      |
 |  114 |  23 | GPIO3_C2 |     IN | 1 | 35 || 36 | 1 | IN     | GPIO3_D7 | 24  | 63   |
 |  135 |  25 | GPIO4_A7 |     IN | 0 | 37 || 38 | 1 | IN     | GPIO3_C0 | 26  | 112  |
 |      |     |      GND |        |   | 39 || 40 | 1 | IN     | GPIO3_B7 | 27  | 111  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
```

1) Below, we take pin 11, which corresponds to GPIO GPIO4_B2 and wPi number 5, as an example to demonstrate how to set the pull-up and pull-down resistors of the GPIO port.

```
[root@orange-os ~]# gpio readall
 +------+-----+----------+--------+---+  PI5 MAX +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   16 |   0 |    SDA.2 |     IN | 0 |  3 || 4  |   |        | 5V       |     |      |
 |   15 |   1 |    SCL.2 |     IN | 0 |  5 || 6  |   |        | GND      |     |      |
 |   39 |   2 |     PWM3 |     IN | 1 |  7 || 8  | 1 | ALT10  | TXD.2    | 3   | 13   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT10  | RXD.2    | 4   | 14   |
 |   32 |   5 |    RXD.6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO4_A6 | 6   | 134  |
 |   33 |   7 |    TXD.6 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |   34 |   8 | GPIO1_A2 |     IN | 0 | 15 || 16 | 0 | IN     | GPIO1_A3 | 9   | 35   |
```

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

the wPi number corresponding to the pin.

```
[orangepi@orangepi ~]$ 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. 6. 3. 40pin SPI test

1) As shown in the figure below, the available spis for Orange Pi 5 Max are spi0, spi1 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of SPI0, SPI1 and SPI4 in 40 pins are shown in the following table.

	SPI0_M2 corresponds to 40pin	SPI1_M1 corresponds to 40pin	SPI4_M2 corresponds to 40pin
MOSI	Pin 19	Pin 40	Pin 13
MISO	Pin 21	Pin 38	Pin 11

CLK	**Pin 23**	**Pin 29**	**Pin 15**
CS0	**Pin 24**	**Pin 35**	**Pin 16**
CS1	**Pin 26**	**NO**	**NO**
dtbo configuration	**spi0-m2-cs0-spidev** **spi0-m2-cs1-spidev** **spi0-m2-cs0-cs1-spidev**	**spi1-m1-cs0-spidev**	**spi4-m2-cs0-spidev**

In OPi OS Arch system, the spi function in 40pin is disabled by default and needs to be enabled manually before it can be used.

Add the following configuration in red font to /boot/extlinux/extlinux.conf, then restart the OPi OS Arch system to enable spi0, spi1 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/rk3588-orangepi-5-max.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-spi0-m2-cs0-cs1-spidev.dtbo
/dtbs/rockchip/overlay/rk3588-spi1-m1-cs0-spidev.dtbo
/dtbs/rockchip/overlay/rk3588-spi4-m2-cs0-spidev.dtbo
```

3) First check whether there is a device node of **spidevx.x** 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/spidev1.0    /dev/spidev4.0
```

The above is the result after opening spi0-m2-cs0-cs1-spidev, spi1-m1-cs0-spidev and spi4-m2-cs0-spidev.

4) Do not short the mosi and miso pins of SPI0, SPI1 or SPI4. The output of running spidev_test is as follows. You can see that the data of TX and RX are inconsistent.

```
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev0.0
Or
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev1.0
```

```
Or
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF  | ................................
```

5) Then short the mosi and miso pins of SPI0, SPI1 or SPI4 and run spidev_test. The output is as follows. You can see that the data sent and received are the same.

```
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev0.0
Or
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev1.0
Or
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
```

4. 6. 4. 40pin I2C test

1) As can be seen from the table below, Orange Pi 5 has four i2c buses: i2c2, 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2) The corresponding pins of the 4 groups of I2C buses in 40pin are shown in the following table. I2C2_M0 and I2C2_M4, I2C5_M2 and I2C5_M3 can only use one of

them at the same time, they cannot be used at the same time, they are all the same I2C, just connected to different pins, please do not think that they are two different I2C buses.

I2C Bus	SDA corresponds to 40pin	SCL corresponds to 40pin	dtbo corresponding configuration
I2C2_M0	Pin 3	Pin 5	i2c2-m0
I2C2_M4	Pin 11	Pin 13	i2c2-m4
I2C4_M3	Pin 15	Pin 16	i2c4-m3
I2C5_M2	Pin 37	Pin 12	i2c5-m2
I2C5_M3	Pin 27	Pin 28	i2c5-m3
I2C8_M2	Pin 36	Pin 32	i2c8-m2

In OPi OS Arch system, the i2c in 40pin is disabled by default and needs to be enabled manually before it can be used.

Add the following configuration in red font to /boot/extlinux/extlinux.conf, then restart the OPi OS Arch system to enable i2c2, i2c3, 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/rk3588-orangepi-5-max.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c2-m0.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 configurations above need to be written in one line, and different configurations need to be separated by spaces.

3) After booting the OPi OS Arch system, first confirm that the i2c device node exists under /dev

```
[orangepi@orangepi ~]$ ls /dev/i2c-*
/dev/i2c-0   /dev/i2c-10   /dev/i2c-3   /dev/i2c-6   /dev/i2c-9
/dev/i2c-1   /dev/i2c-2    /dev/i2c-5   /dev/i2c-7   /dev/i2c-8
```

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

Generally, you only need to connect one of the 3.3v pin and the 5v pin. Please choose to connect the 3.3v pin or the 5v pin according to the specific i2c device you are connecting.

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 2        #i2c2 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 3        #i2c3 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 5        #i2c5 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 8        #i2c8 commands
```

4. 6. 5. 40pin UART test

1) As can be seen from the table below, Orange Pi 5 Max has four uart buses: uart1, uart3, uart4 and uart6.

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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

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

UART Bus	RX corresponds to 40pin	TX corresponds to 40pin	dtbo corresponding configuration
UART1_M1	**Pin 27**	**Pin 28**	**uart1-m1**
UART3_M1	**Pin 33**	**Pin 31**	**uart3-m1**
UART4_M2	**Pin 19**	**Pin 23**	**uart4-m2**
UART6_M1	**Pin 11**	**Pin 13**	**uart6-m1**

In OPi OS Arch system, the uart in 40pin is closed by default and needs to be manually opened before it can be used.

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, just fill in one.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588-orangepi-5-max.dtb
FDTOVERLAYS    /dtbs/rockchip/overlay/rk3588-uart1-m1.dtbo
/dtbs/rockchip/overlay/rk3588-uart3-m1.dtbo
/dtbs/rockchip/overlay/rk3588-uart4-m2.dtbo
/dtbs/rockchip/overlay/rk3588-uart6-m1.dtbo
```

The red font configurations above need to be written in 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/ttyS1    /dev/ttyS3    /dev/ttyS4    /dev/ttyS6    /dev/ttyS9
```

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

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 that the serial port communication is normal (ttySX needs to be replaced with the corresponding uart node name, 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
```

4.6.6. PWM test method

1)As can be seen from the table below, Orange Pi 5 Max has six PWM channels: pwm0, pwm1, pwm3, pwm12, pwm13 and pwm14

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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

2)The corresponding pins of PWM in 40pin are shown in the following table. Only one of PWM0_M0 and PWM0_M2, PWM1_M0 and PWM1_M2, PWM13_M0 and PWM13_M2, PWM14_M0 and PWM14_M2 can be used 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	Corresponding to 40pin	dtbo corresponding configuration
PWM0_M0	Pin 5	pwm0-m0
PWM0_M2	Pin 15	pwm0-m2
PWM1_M0	Pin 3	pwm1-m0
PWM1_M2	Pin 16	pwm1-m2
PWM3_M3	Pin 7	pwm3-m3
PWM12_M0	Pin 31	pwm12-m0
PWM13_M0	Pin 33	pwm13-m0
PWM13_M2	Pin 27	pwm13-m2
PWM14_M0	Pin 35	pwm14-m0
PWM14_M2	Pin 32	pwm14-m2

In OPi OS Arch system, the pwm in 40pin is turned off by default and needs to be turned on manually before it can be used.

Add the following configuration in red font to /boot/extlinux/extlinux.conf, and then

restart the OPi OS Arch system to enable pwm0, pwm1, pwm3, pwm12, pwm13 and pwm14 at the same time. If you only need to enable one, just fill in one.

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588-orangepi-5-max.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-pwm0-m0.dtbo /dtbs/rockchip/overlay/rk3588-pwm1-m0.dtbo /dtbs/rockchip/overlay/rk3588-pwm3-m3.dtbo /dtbs/rockchip/overlay/rk3588-pwm12-m0.dtbo /dtbs/rockchip/overlay/rk3588-pwm13-m0.dtbo /dtbs/rockchip/overlay/rk3588-pwm14-m0.dtbo

The red font configurations above need to be written in one line, and different configurations need to be separated by spaces.

3)When a pwm is turned on, there will be an additional pwmchipX (X is a specific number) in **/sys/class/pwm/**/. For example, after turning on pwm3, the pwmchipX under **/sys/class/pwm/** will change from two to three.

```
[orangepi@orangepi ~]$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

4)Which pwmchip above corresponds to pwm3? Let's first check the output of the ls **ls /sys/class/pwm/ -l** command, as shown below:

```
[root@orange-os ~]#  ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Jul  5 09:34 pwmchip0 -> ../../devices/platform/fd8b0030.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Jul  5 09:34 pwmchip1 -> ../../devices/platform/febe0010.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Jul  5 09:34 pwmchip2 -> ../../devices/platform/febf0030.pwm/pwm/pwmchip2
```

5)Then from the table below, we can see that the base address of the pwm3 register is fe8b0030. Looking at the output of the ls **ls /sys/class/pwm/ -l** command, we can see that pwmchip0 is linked to fe8b0030.pwm, so the pwmchip corresponding to pwm3 is pwmchip0

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
			3.3V		1
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7
			GND		9
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	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
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27
		SPI1_CLK_M1	GPIO3_C1	113	29
PWM12_M0 (febf0000)	CAN1_RX_M0	UART3_TX_M1	GPIO3_B5	109	31
PWM13_M0 (febf0010)	CAN1_TX_M0	UART3_RX_M1	GPIO3_B6	110	33
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35
I2C5_SDA_M2			GPIO4_A7	135	37
			GND		39

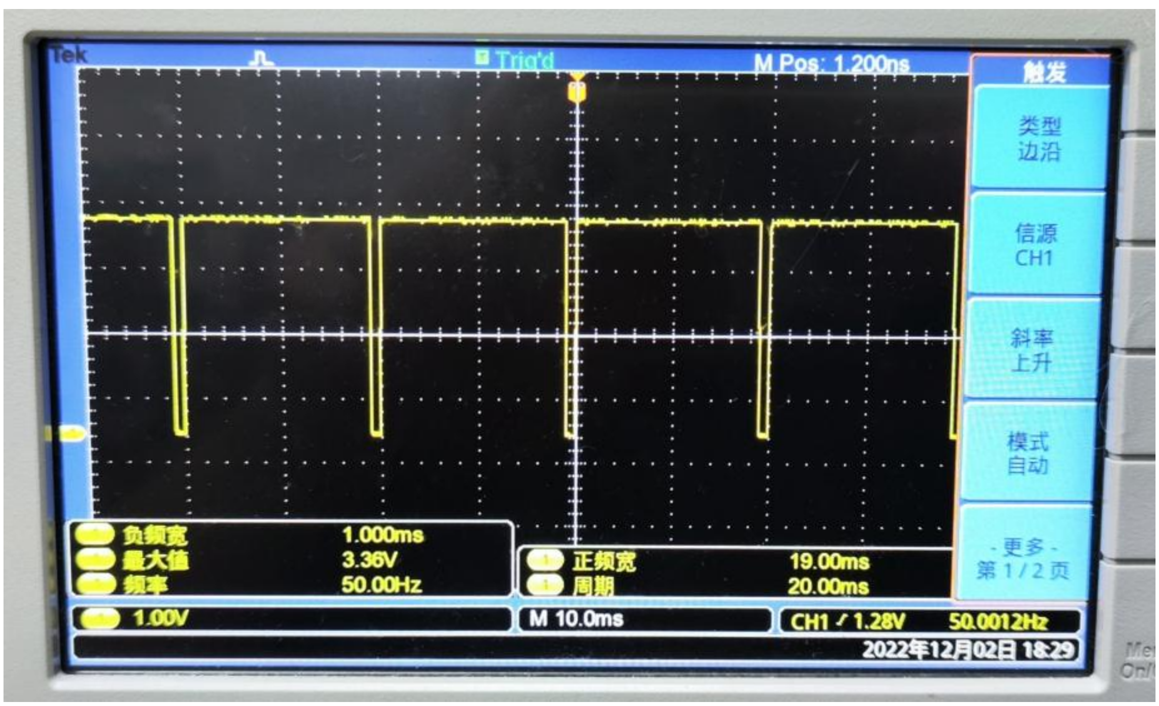
6)Then use the following command to make pwm3 output a 50Hz square wave (please switch to root user first, then execute the following command)

[root@orangepi orangepi]# **echo 0 > /sys/class/pwm/pwmchip0/export**

[root@orangepi orangepi]# **echo 20000000 > /sys/class/pwm/pwmchip0/pwm0/period**

[root@orangepi orangepi]# **echo 1000000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle**

[root@orangepi orangepi]# **echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable**



7)The pwm3 test method demonstrated above is similar to other pwm test methods.

4.6.7. CAN test method

1) As can be seen from the table below, Orange Pi 5 Max has two CAN buses available: CAN0 and 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			
PWM1_M0 (fd8b0010)	CAN0_RX_M0	I2C2_SDA_M0	GPIO0_C0	16	3	4		5V			
PWM0_M0 (fd8b0000)	CAN0_TX_M0	I2C2_SCL_M0	GPIO0_B7	15	5	6		GND			
		PWM3_IR_M3 (fd8b0030)	GPIO1_A7	39	7	8	13	GPIO0_B5	UART2_TX_M0	I2C1_SCL_M0	
			GND		9	10	14	GPIO0_B6	UART2_RX_M0	I2C1_SDA_M0	
SPI4_MISO_M2	I2C2_SDA_M4	UART6_RX_M1	GPIO1_A0	32	11	12	134	GPIO4_A6			I2C5_SCL_M2
SPI4_MOSI_M2	I2C2_SCL_M4	UART6_TX_M1	GPIO1_A1	33	13	14		GND			
SPI4_CLK_M2	PWM0_M2 (fd8b0000)	I2C4_SDA_M3	GPIO1_A2	34	15	16	35	GPIO1_A3	I2C4_SCL_M3	PWM1_M2 (fd8b0010)	SPI4_CS0_M2
			3.3V		17	18	36	GPIO1_A4			
	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		
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2		
PWM13_M2 (febf0010)	I2C5_SDA_M3	UART1_RX_M1	GPIO1_B7	47	27	28	46	GPIO1_B6	UART1_TX_M1	I2C5_SCL_M3	
		SPI1_CLK_M1	GPIO3_C1	113	29	30		GND			
PWM12_M0 (febf0000)	UART3_TX_M1	CAN1_RX_M0	GPIO3_B5	109	31	32	62	GPIO1_D6	PWM14_M2 (febf0020)	I2C8_SCL_M2	
PWM13_M0 (febf0010)	UART3_RX_M1	CAN1_TX_M0	GPIO3_B6	110	33	34		GND			
PWM14_M0 (febf0020)		SPI1_CS0_M1	GPIO3_C2	114	35	36	63	GPIO1_D7		I2C8_SDA_M2	
I2C5_SDA_M2			GPIO4_A7	135	37	38	112	GPIO3_C0	SPI1_MISO_M1		
			GND		39	40	111	GPIO3_B7	SPI1_MOSI_M1		

In OPi OS Arch system, the CAN in 40pin is closed by default and needs to be manually opened before use.

Add the following red font configuration to /boot/extlinux/extlinux.conf, then restart the OPi OS Arch system to enable CAN0 and CAN1.

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588-orangepi-5-max.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-can0-m0.dtbo /dtbs/rockchip/overlay/rk3588-can1-m1.dtbo

The red font configurations above need to be written in one line, and different configurations need to be separated by spaces.

2) After entering the OPi OS Arch system, use the **sudo ifconfig -a** command. If you can see the CAN device node, it means that CAN has been correctly enabled.

```
[orangepi@orangepi ~]$ sudo pacman -Syy net-tools
[orangepi@orangepi ~]$ sudo ifconfig -a
can0: flags=128<NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
        device interrupt 91

can1: flags=128<NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 92
```

3) The pins corresponding to CAN0 and CAN1 are

	CAN0	CAN1
TX Pin	**Corresponding to pin 3**	**Corresponding to pin 33**
RX Pin	**Corresponding to pin 5**	**Corresponding to pin 31**

4) For the method of using CANalyst-II analyzer to test CAN message sending and receiving, please refer to the content of the section **"Using CANalyst-II analyzer to test message sending and receiving".**

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 do a good job of heat dissipation (especially when starting with SSD). If the heat dissipation is not done well, the file system will easily

run away.

5.1.1. Compile using the Ubuntu 22.04 system of the development board

1) Linux SDK, namely **orangepi-build**, supports running on **Ubuntu 22.04** of the development board (other systems have not been tested), so before downloading orangepi-build, please first make sure 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 current Ubuntu version 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 codes of 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 Ubuntu 22.04 x64 computer

1) Linux SDK, **orangepi-build**, supports running on computers with **Ubuntu 22.04** installed, so before downloading orangepi-build, please first make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on your computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please change the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID:    Ubuntu
Description:       Ubuntu 22.04 LTS
Release:           22.04
Codename:          jammy
```

2) If your computer is running Windows and you don't have Ubuntu 22.04 installed, you

can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in Windows. But please note that you should not compile orangepi-build on a WSL virtual machine, because orangepi-build has not been tested in a WSL virtual machine, so it is not guaranteed 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 your computer or in a virtual machine, please set the software source of Ubuntu 22.04 to Tsinghua source first, otherwise it is easy to get 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

c. The content of the **/etc/apt/sources.list** file that needs to be replaced is

```
test@test:~$ sudo mv /etc/apt/sources.list /etc/apt/sources.list.bak
test@test:~$ sudo vim /etc/apt/sources.list
# The source mirror is commented out by default to increase the speed of apt update. You can uncomment it if necessary.
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
```

```
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

#Pre-release software source, not recommended to enable
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

d. After the replacement, you need to update the package information and ensure that there is no error

```
test@test:~$ sudo apt update
```

e. In addition, since the source codes of the 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.

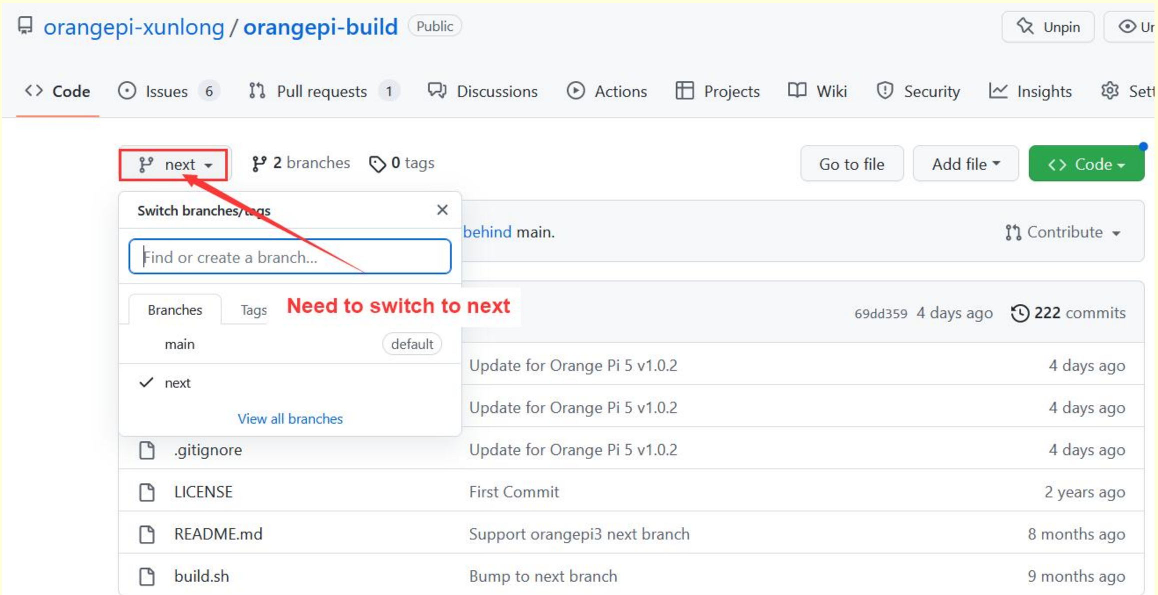
5. 2. Get the source code of Linux SDK

5. 2. 1. Download orangepi-build from github

1) Linux SDK actually refers to the orangepi-build code. Orangepi-build is modified based on the armbian build compilation system. Orangepi-build can be used to compile multiple versions of Linux images. 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 Max development board needs to download the next branch source code of orangepi-build. The above git clone command needs to specify the branch of orangepi-build source code as next.



You do not need to enter the username and password of your GitHub account when you download the orangepi-build code through the git clone command (the same applies to downloading other codes in this manual). If your Ubuntu PC prompts you to enter the username and password of your GitHub account after entering the git clone command, it is usually because the address of the orangepi-build repository after git clone is entered incorrectly. Please check the command spelling carefully for any errors, instead of thinking that we forgot to provide the username and password of our GitHub account.

2) The u-boot and linux kernel versions currently used by the development board are as follows

Branches	u-boot version	Linux kernel version
legacy	u-boot 2017.09	linux5.10

The branch mentioned here is not the same as the branch of orangepi-build source code, please do not confuse them. This branch is mainly used to distinguish different kernel source code versions.

Currently, the linux5.10 bsp kernel provided by RK is defined as the legacy branch. If the mainline kernel is supported in the future, a current branch will be added.

3) After downloading orangepi-build, it will contain the following files and folders
   a. **build.sh**: Compile the startup script
   b. **external**: Contains configuration files, specific scripts, and source code of some programs needed to compile the image.
   c. **LICENSE**: GPL 2 License File
   d. README.md: orangepi-build description file
   e. **scripts**: Generic script for compiling linux images

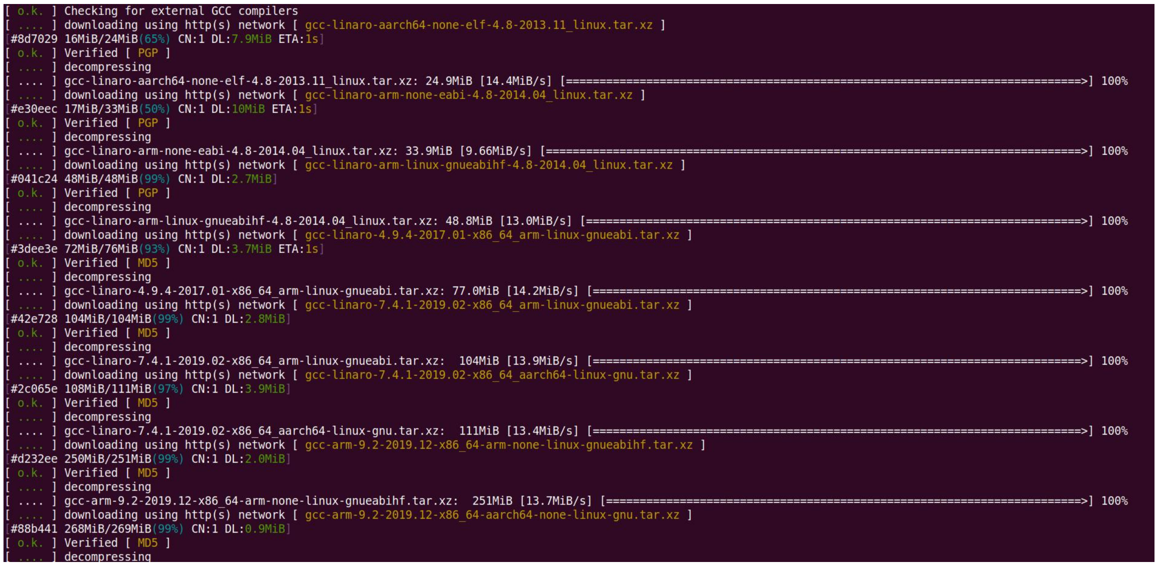
```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

If you download the orangepi-build code from github, you may find that orangepi-build does not contain the source code of u-boot and linux kernel, nor the cross-compilation toolchain required to compile u-boot and linux kernel. This is normal because these things are stored in other separate github repositories or some servers (the addresses will be detailed below). orangepi-build will specify the addresses of u-boot, linux kernel and cross-compilation toolchain in the script and configuration file. When running orangepi-build, if it finds that these things are not available locally, it will automatically download them from the corresponding places.

5.2.2. Download the cross-compilation toolchain

The cross-compilation toolchain will only be downloaded when you compile the image using orangepi-build on an x64 computer. Compiling the Linux image of the development board in Ubuntu 22.04 on the development board will not download the cross-compilation toolchain, and 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. Each time you run the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists. If not, it will restart the download. If it exists, it will be used directly without repeated downloading.



2) The mirror website 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 downloading **toolchains**, 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 the 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 repository, it does not contain the source code of the Linux kernel, u-boot, and the cross-compilation toolchain. The source code of the Linux kernel and u-boot is stored in a separate git repository.

a. The git repository 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 repository where the 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 toolchain, u-boot and Linux kernel source code. After successfully compiling a Linux image, the files and folders that can be seen in orangepi-build are

a. **build.sh**: Compile the startup script
b. **external**: Contains configuration files, scripts for specific functions, and source code for some programs needed to compile the image. The rootfs compressed package cached during the image compilation process is also stored in external
c. **kernel**: The source code of the Linux kernel is stored in the folder named **orange-pi-5.10-rk3588**, which contains the kernel source code of the legacy branch of the RK3588/RK3588S series development board. Please do not manually modify the name of the kernel source code folder. If modified, the kernel source code will be re-downloaded when the compiling system is running.
d. **LICENSE**: GPL 2 License File
e. **README.md**: orangepi-build documentation
f. **output**: Stores compiled u-boot, linux and other deb packages, compilation logs, compiled images and other files
g. **scripts**: Generic script for compiling linux images
h. **toolchains**: Store cross-compilation toolchain
i. **u-boot**: The u-boot source code is stored in the folder named **v2017.09-rk3588**, which is the u-boot source code of the legacy branch of the RK3588/RK3588S series development board. Please do not manually modify the name of the u-boot source code folder. If modified, the compile system will re-download the u-boot

source code when running.

j. **userpatches**: Store the configuration files needed to compile the script

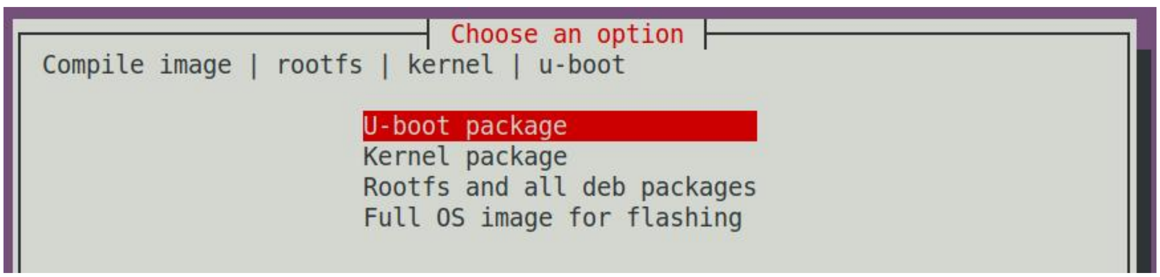
```
test@test:~/orangepi-build$ ls
build.sh    external    kernel    LICENSE    output    README.md    scripts    toolchains
u-boot    userpatches
```

5. 3. Compile u-boot

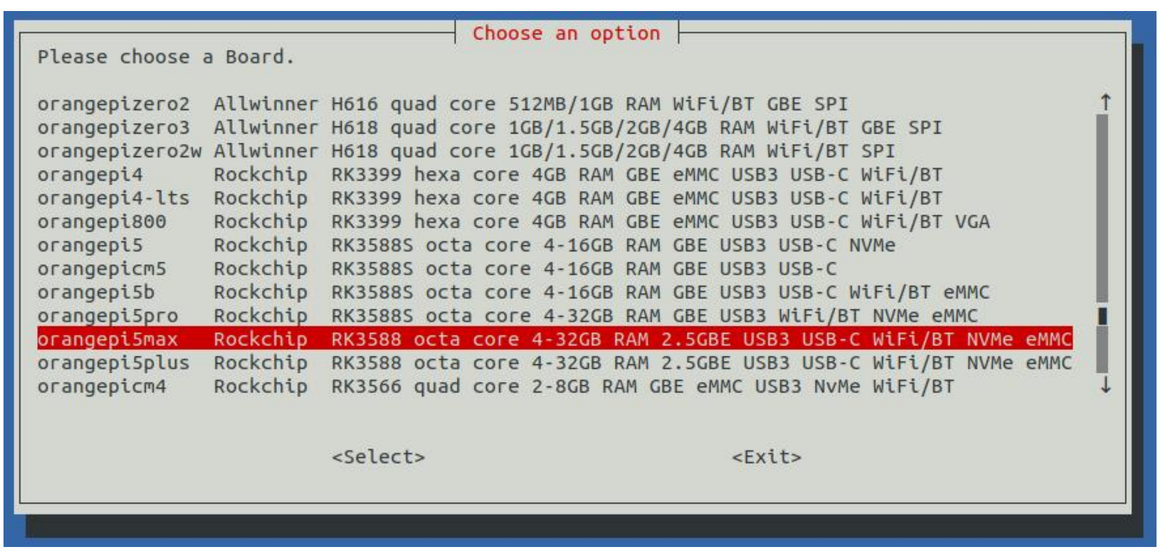
1) Run the build.sh script and remember to add sudo permissions

```
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 u-boot will start to compile. 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 toolchain

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 7.4.1 ]
```

c. The path of the compiled u-boot deb package

```
[ o.k. ] Target directory [ orangepi-build/output/debs/u-boot ]
```

d. The package name of the compiled u-boot deb package

```
[ o.k. ] File name [ linux-u-boot-legacy-orangepi5max_1.0.2_arm64.deb ]
```

e. Time used for compilation

```
[ o.k. ] Runtime [ 1 min ]
```

f. Repeat the command to compile u-boot. Use the following command without selecting through the graphical interface to start compiling u-boot directly.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi5max 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-orangepi5max_1.0.2_arm64.deb
```

6) The generated u-boot deb package contains the following files

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-orangepi5max_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-orangepi5max_1.0.2_arm64.deb    usr
```

b. The decompressed files are as follows

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-legacy-orangepi5max_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 on the GitHub server. So if you want to modify the u-boot source code, you first need to turn off the source code download and update function **(you need to compile u-boot once before turning off this function, otherwise it will prompt that the u-boot source code cannot be found. If the source code compression package is downloaded from Baidu Cloud Disk, there is no such problem because the u-boot source code has been cached)**, otherwise the changes made will be restored. The method is as follows:

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

8) When debugging the u-boot code, you can use the following method to update the u-boot in the Linux image for testing

a. Upload the compiled u-boot deb package to the Linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-legacy-orangepi5max_1.0.2_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the installed u-boot deb package

```
root@orangepi:~# apt purge -y linux-u-boot-orangepi5max-legacy
```

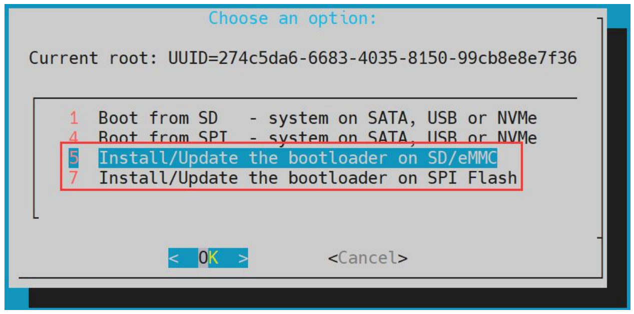
c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi5max_1.0.2_arm64.deb
```

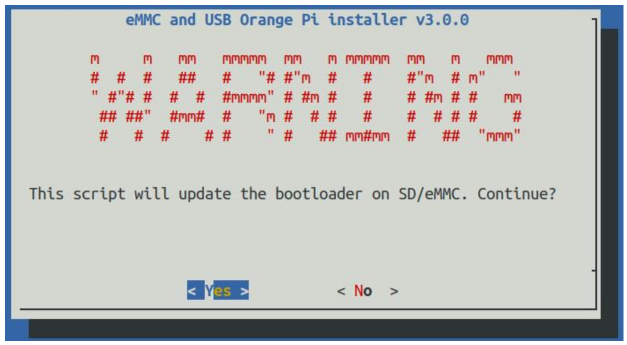
d. Then run the nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

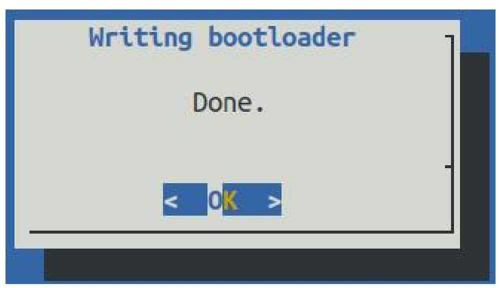
e. Then select **5 Install/Update the bootloader on SD/eMM** to update the u-boot in the TF card or **7 Install/Update the bootloader on SPI Flash** to update the u-boot in the SPI Flash



f. After pressing the Enter key, a Warning pops up first.



g. Press the Enter key again to start updating u-boot. After the update, the following information will be displayed



h. Then you can restart the development board to test whether the changes in u-boot are effective.

9) Other useful information

a. In the u-boot 2017.09 source code, the defconfig configuration file used by the development board is

orangepi-build/u-boot/v2017.09-rk3588/configs/orangepi_5_max_defconfig

b. In the u-boot 2017.09 source code, the development board uses the dts file as

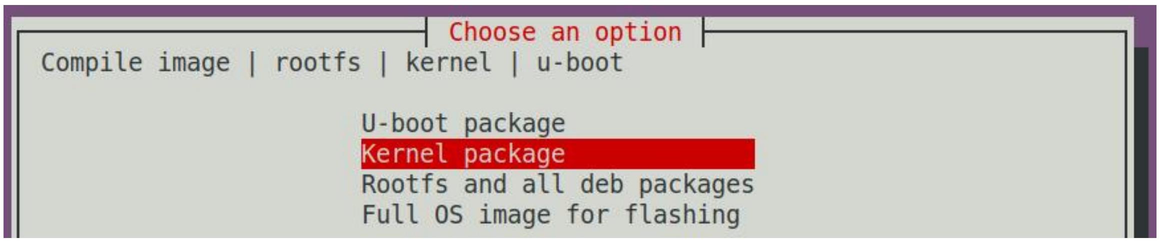
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3588-orangepi-5-max.dts

5. 4. Compile the Linux kernel

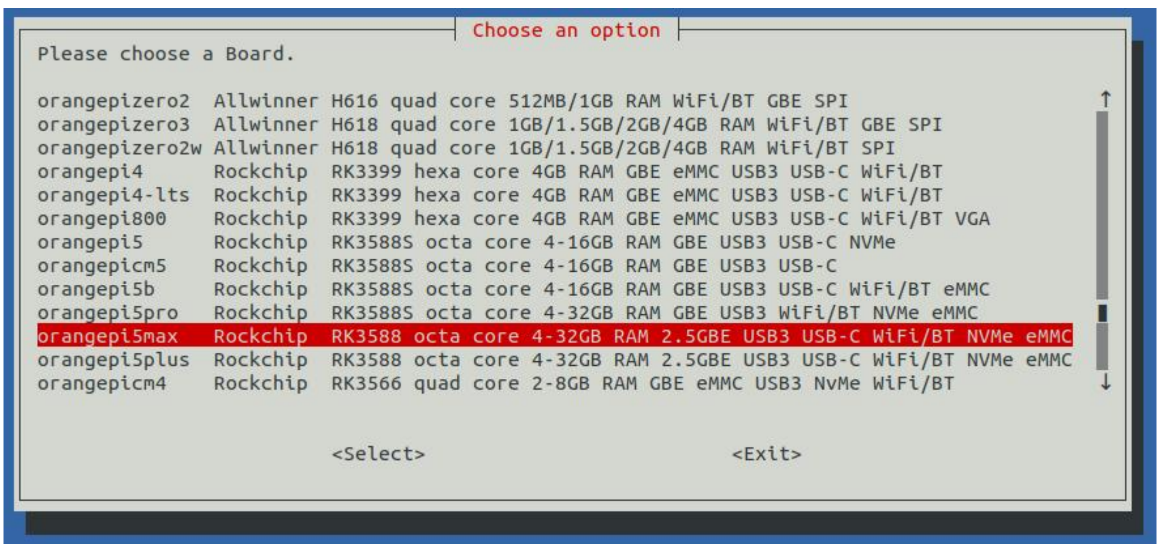
1) Run the build.sh script and remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

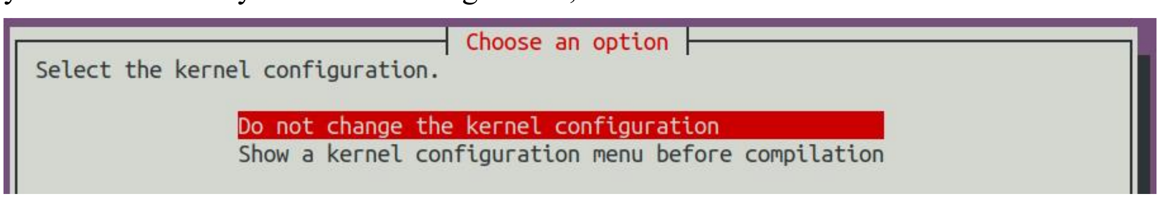
2) Select **Kernel package** and press Enter



3) Then select the model of the development board

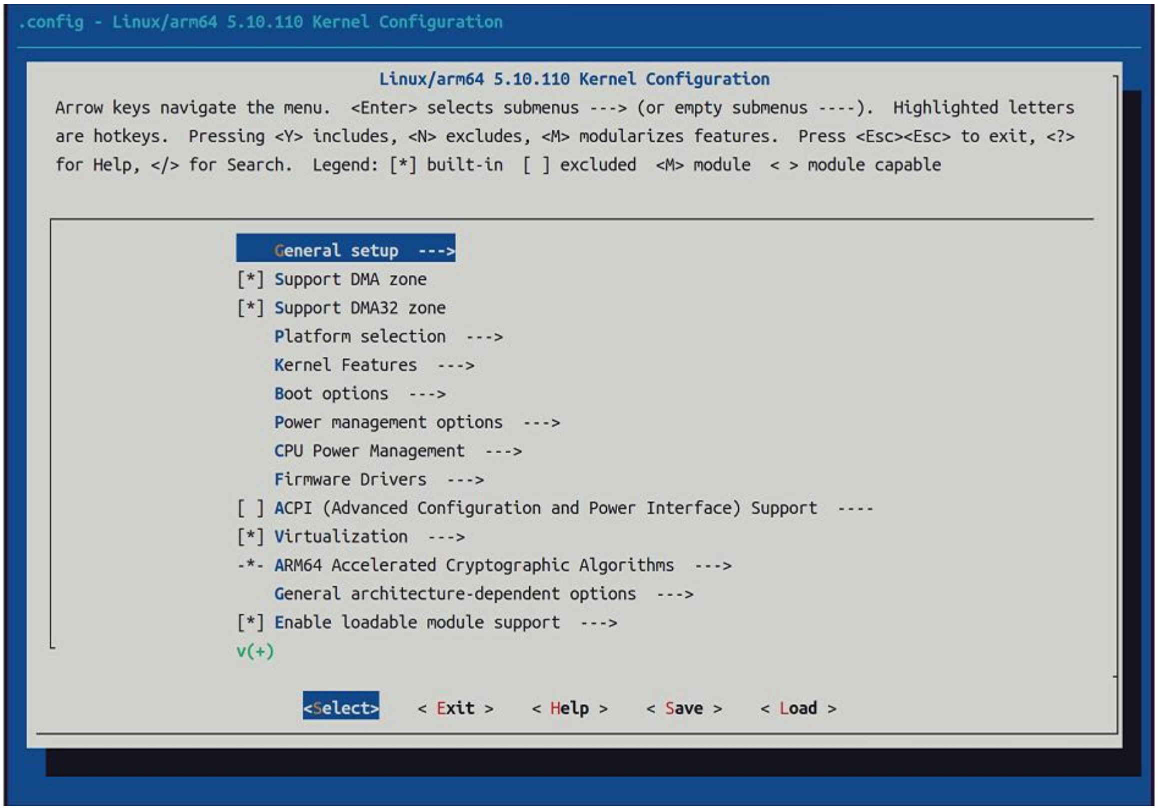


4) Then you will be prompted whether you need to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If you need to modify the kernel configuration, select the second one.



5) If you selected the option to display the kernel configuration menu (the second option)

in step 4), the kernel configuration interface opened by **make menuconfig** will pop up. You can modify the kernel configuration directly at this time. After modifying, save and exit. After exiting, the kernel source code will be compiled.



a. If you do not need to modify the kernel configuration options, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily block the kernel configuration interface from popping up.

```
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 feature

c. If the following error message appears when compiling the kernel, it is because the terminal interface of the Ubuntu PC is too small, causing the **make menuconfig** interface to fail to display. Please adjust the terminal of the Ubuntu PC to the maximum size and re-run 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) Some of the information prompted when compiling the kernel source code is as follows

a. Linux kernel source code version

[o.k.] Compiling current kernel [**5.10.160**]

b. Version of the cross-compilation toolchain used

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 11.2.1**]

c. The default configuration file used by the kernel 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 compiled kernel-related deb package

[o.k.] Target directory [**orangepi-build/output/debs/**]

e. The package name of the compiled kernel image deb package

[o.k.] File name [**linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb**]

f. Compilation time

[o.k.] Runtime [**5 min**]

g. Finally, the compilation command for the last selected kernel will be displayed. Use the following command to directly start compiling the kernel source code without selecting through the graphical interface.

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepi5max BRANCH=legacy BUILD_OPT=kernel KERNEL_CONFIGURE=no**]

7) Check the compiled kernel-related deb packages

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** Include 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 linux-image deb package 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 files are 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

on the GitHub server. So if you want to modify the Linux kernel source code, you first need to turn off the source code update function **(you need to fully compile the Linux kernel source code once before turning off this function, otherwise it will prompt that the Linux kernel source code cannot be found. If the source code compression package is downloaded from Baidu Cloud Disk, there will be no such problem because the Linux source code has been cached)**, otherwise the changes made will be restored. The method is as follows:

Set the IGNORE_UPDATES variable in **userpatches/config-default.conf** to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If the kernel is modified, you can use the following method to update the kernel and kernel modules of the development board Linux system

a. Upload the compiled Linux kernel deb package 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 linux kernel deb package

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk3588
```

c. Install the new Linux kernel deb package 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 changes have taken effect.

```
root@orangepi:~# reboot
```

10) Other useful information

a. The kernel configuration file is stored in the following location. Please do not search for the kernel configuration file used by the development board in the kernel source code.

```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-legacy-opi5max.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/rk3588-orangepi-5-max.dts
```

5. 5. Compile rootfs

1) Run the build.sh script and remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

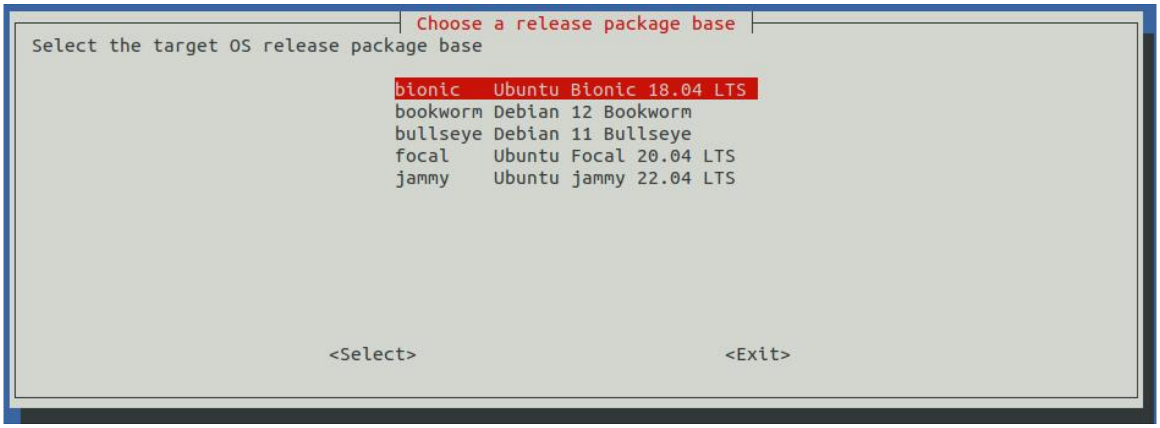
2) Select **Rootfs and all deb packages** and press Enter

```
                          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.

orangepizero2   Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
orangepizero3   Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
orangepizero2w  Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
orangepi4       Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi4-lts   Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi800     Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
orangepi5       Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
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
orangepi5max    Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C 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


                          <Select>                          <Exit>
```

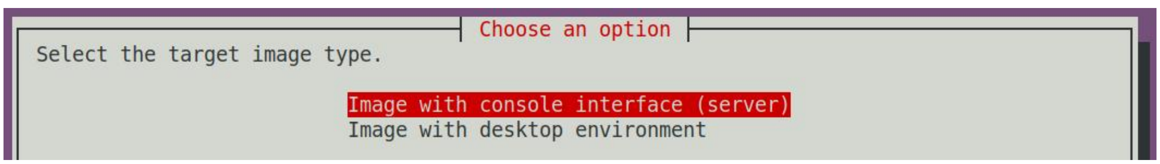
4) Then select the type of rootfs



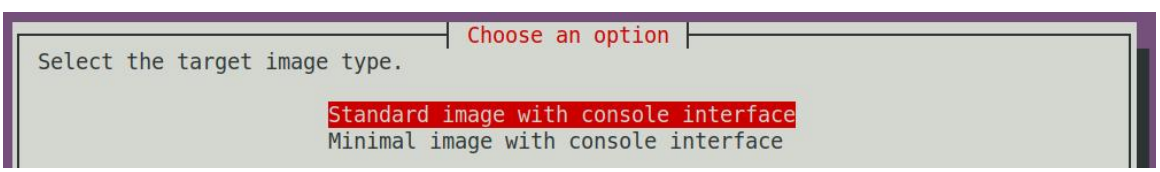
5) Then select the type of image

a. **Image with console interface (server)** Indicates the server version of the image, which is relatively small in size

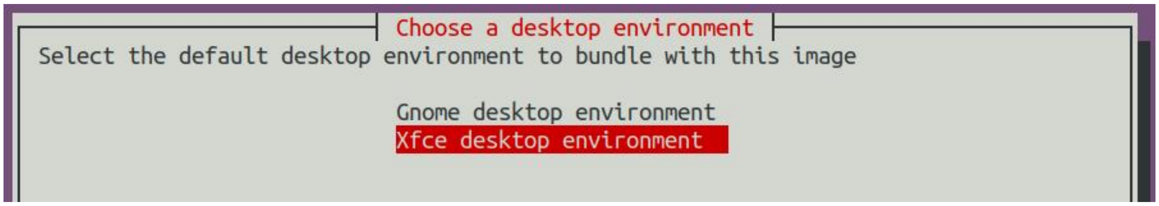
b. **Image with desktop environment** Indicates an image with a desktop, which is relatively large in size

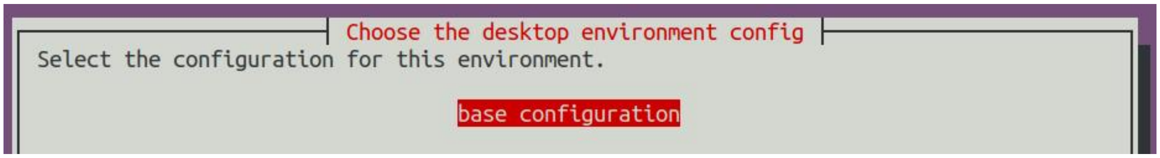


6) If you are compiling a server version image, you can also choose to compile the Standard version or the Minimal version. The Minimal version has much less pre-installed software than the Standard version **(please do not choose the Minimal version if you do not have special needs, because many things are not pre-installed by default and some functions may not be used)**

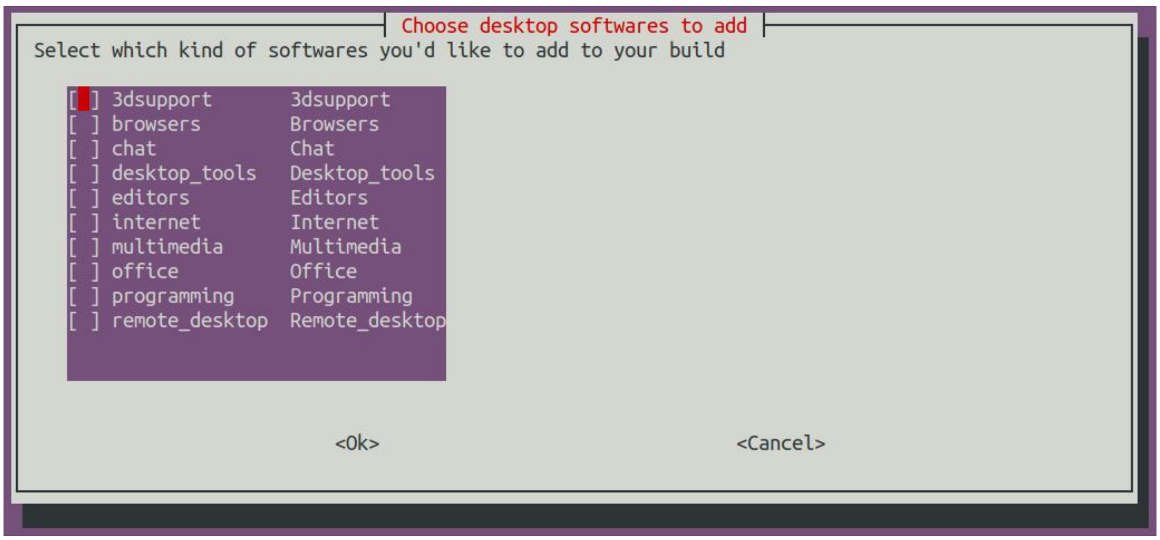


7) If you are compiling a desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktop, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop





You can then select additional packages to install. Please press Enter to skip this step.



8) Then it will start compiling rootfs. 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. Storage path of the compiled rootfs compressed package

[o.k.] Target directory [**external/cache/rootfs**]

c. The name of the rootfs compressed package generated by compilation

[o.k.] File name [**jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**]

d. Compilation time

[o.k.] Runtime [**13 min**]

9) View the compiled rootfs compressed package

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is the compressed package of rootfs. The meaning of each field of the name is

a) **jammy** indicates the type of Linux distribution of rootfs

b) **xfce** indicates that the rootfs is a desktop version, if it is **cli**, it indicates a server version

c) **arm64** indicates the architecture type of rootfs

d) **f930ff6ebbac1a72108a2e100762b18f** is the MD5 hash value generated by the package names of all packages installed by rootfs. As long as the list of packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to 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 in **external/cache/rootfs**, then compiling rootfs again will skip the compilation process directly and will not restart the compilation. When compiling the image, it will also check whether there is a cached rootfs available in **external/cache/rootfs**. If there is, it will be used directly, which can save a lot of download and compilation time.

5. 6. Compile Linux image

1) Run the build.sh script and remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Full OS image for flashing** and press Enter

```
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.

orangepizero2  Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
orangepizero3  Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
orangepizero2w Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
orangepi4      Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi4-lts  Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi800    Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
orangepi5      Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
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
orangepi5max   Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C 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

               <Select>                          <Exit>
```

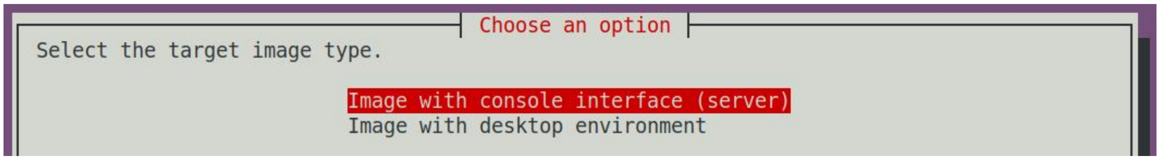
4) Then select the type of rootfs

```
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>
```

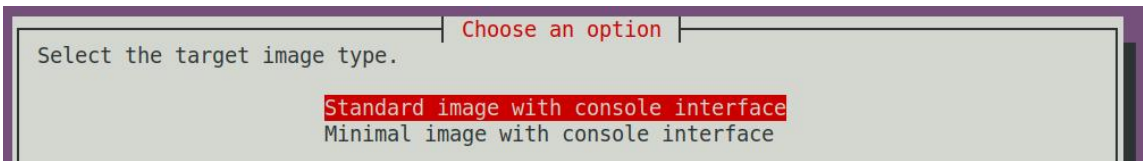
5) Then select the type of image

a. **Image with console interface (server)** Indicates the server version of the image, which is relatively small in size

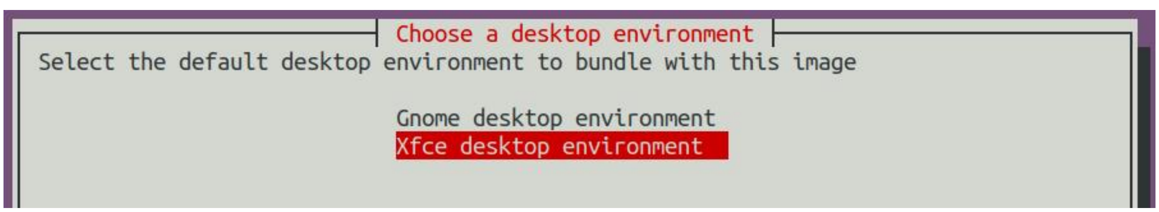
b. **Image with desktop environment** Indicates an image with a desktop, which is relatively large in size

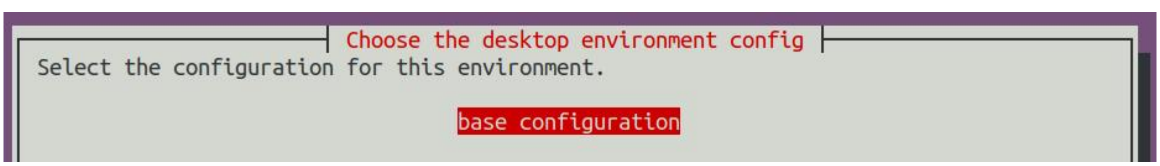


6) If you are compiling a server version image, you can also choose to compile the Standard version or the Minimal version. The Minimal version has much less pre-installed software than the Standard version **(please do not choose the Minimal version if you do not have special needs, because many things are not pre-installed by default and some functions may not be used)**

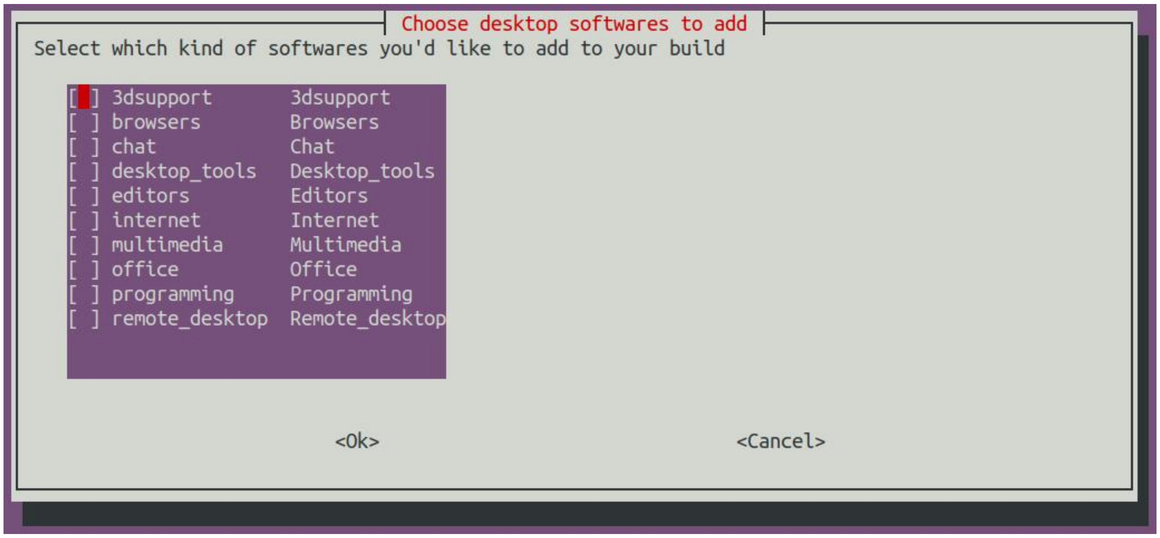


7) If you are compiling a desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktop, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop





You can then select additional packages to install. Please press Enter to skip this step.



8) Then the Linux image will be compiled. 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 it is already cached, only update the code)

c. Compile the u-boot source code and generate the deb package of u-boot

d. Compile the linux source code and generate the linux-related deb package

e. Make linux deb package of firmware

f. Make deb package of orangepi-config tool

g. Make deb package of board support

h. If you compile desktop version image, you will also make desktop related deb package

i. Check whether rootfs has been cached. If not, remake rootfs. If it has been cached, directly decompress and use

j. Install the deb package generated earlier to rootfs

k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.

l. Then make an image file and format the partition. The default type is ext4

m. Then copy the configured rootfs to the partition of the image

n. Then update initramfs

o. Finally, write the bin file of u-boot to the image through the dd command

9) After compiling the image, the following information will be prompted

a. Storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/Orangepi5max_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160/
Orangepi5max_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 through the graphical interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi5max
BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no
BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

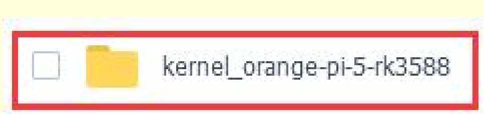
6. Linux Development Manual

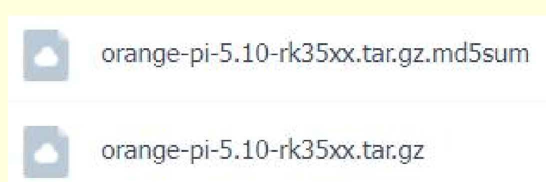
6. 1. How to compile kernel source code separately in the Linux system of the development board

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 you have problems 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 decompress it.





The command to decompress the kernel source code package 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 the latest:

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@orangepi5max:~$ **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 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@orangepi5max:~/orange-pi-5.10-rk3588$ ls /boot/vmlinuz*
/boot/vmlinuz-5.10.160+    /boot/vmlinuz-5.10.160-rockchip-rk3588
```

When the system starts, the file actually loaded is /boot/Image, which is a copy of the vmlinuz file.

6) Then install the dtb file to **/boot/dtb**

```
orangepi@orangepi:~/linux-orangepi$ sudo make dtbs_install INSTALL_DTBS_PATH=/boot/dtb/
```

7) Then restart the Linux system to load the newly compiled kernel

```
orangepi@orangepi:~$ uname -r
5.10.160+
```

7. Instructions for using the Android 13 system

7. 1. Supported Android versions

Android version	Kernel version
Android 13	Linux5.10

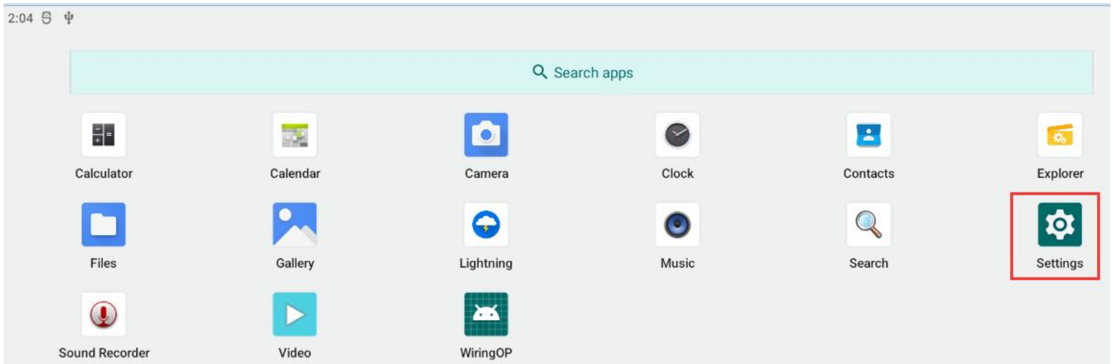
7. 2. Android function adaptation

Function	Android 13
HDMI TX1 Video	OK

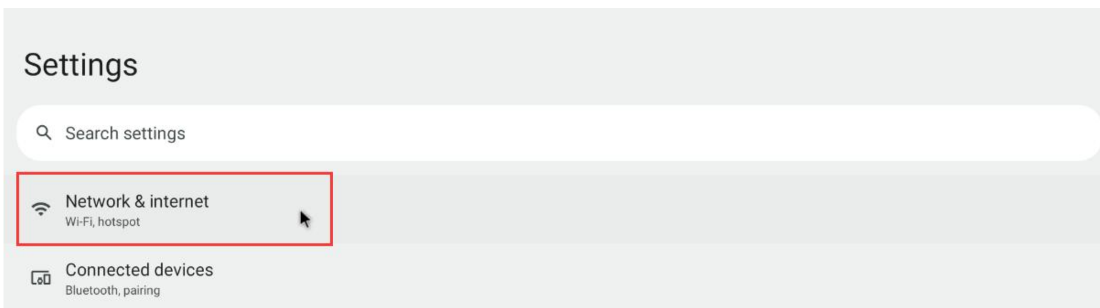
HDMI TX1 Audio	**OK**
HDMI TX2 Video	**OK**
HDMI TX2 Audio	**OK**
USB2.0x2	**OK**
USB3.0x2	**OK**
2.5G network port	**OK**
Network port status light	**OK**
WIFI	**OK**
Bluetooth	**OK**
Debug serial port	**OK**
RTC chip	**OK**
FAN Fan Connector	**OK**
eMMC expansion interface	**OK**
GPIO（40pin）	**OK**
UART（40pin）	**OK**
SPI（40pin）	**OK**
I2C（40pin）	**OK**
PWM（40pin）	**OK**
TF card boot	**OK**
OV13850 Camera	**OK**
OV13855 Camera	**OK**
SPI+NVME boot	**OK**
LCD	**OK**
MIC	**OK**
Headphone playback	**OK**
Headphone Recording	**OK**
Three-color LED light	**OK**
GPU	**OK**
NPU	**OK**
VPU	**OK**
Power button	**OK**
HDMI CEC function	**NO**

7. 3. WIFI connection test method

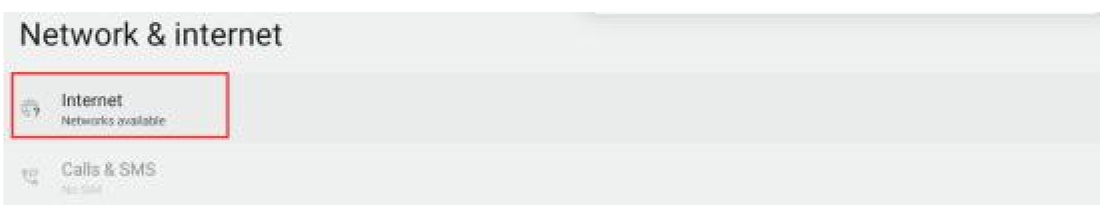
1) First click to enter **Setting**



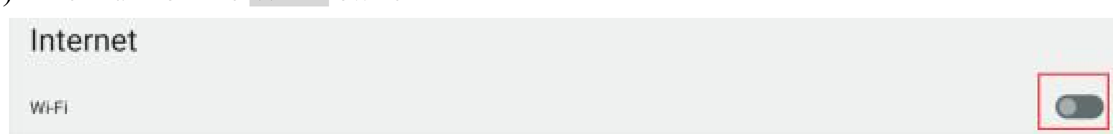
2) Then select **Network & internet**



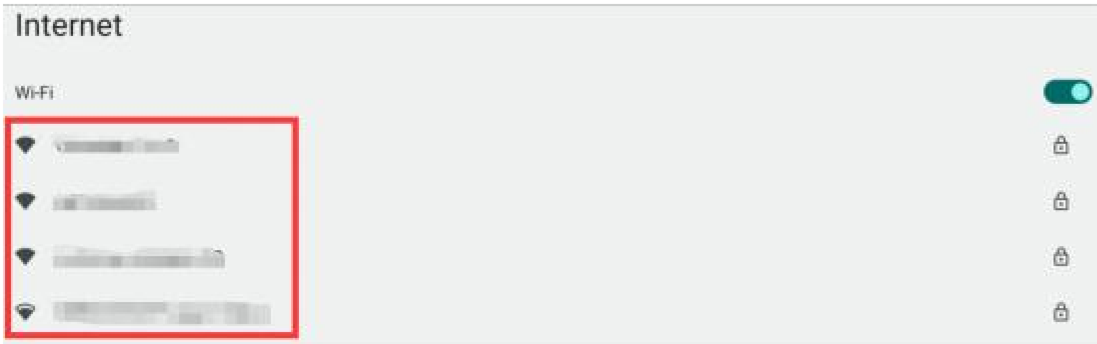
3) Then select **Internet**



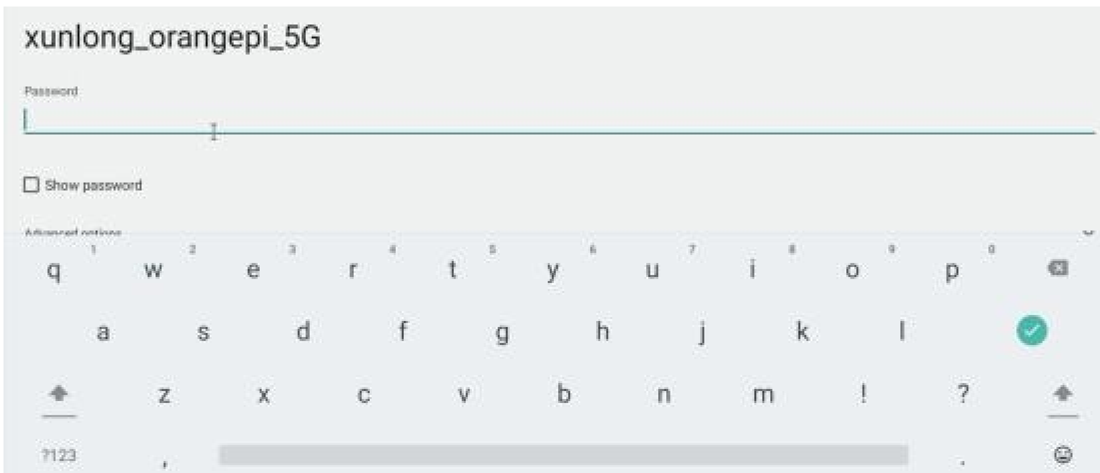
4) Then turn on the **Wi-Fi** switch



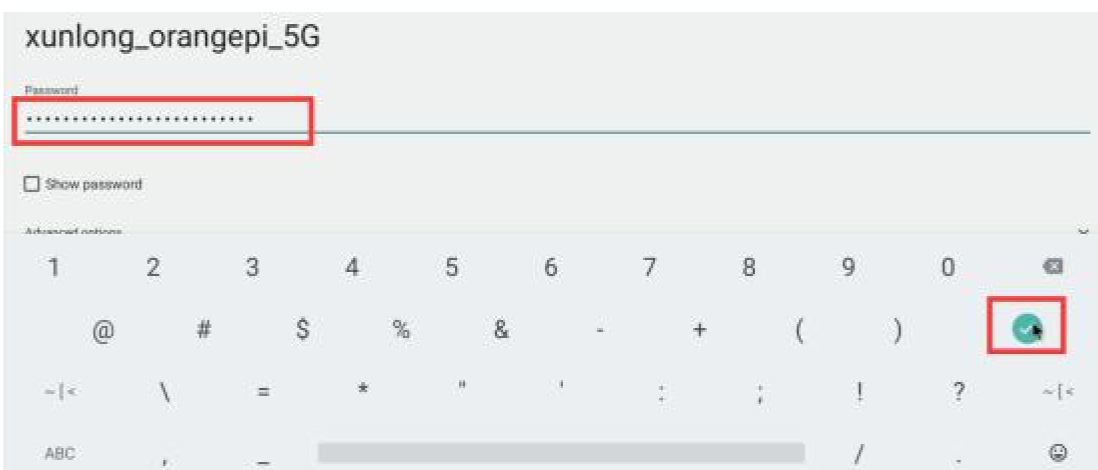
5) After turning on **Wi-Fi**, if everything is normal, you can scan nearby Wi-Fi hotspots.



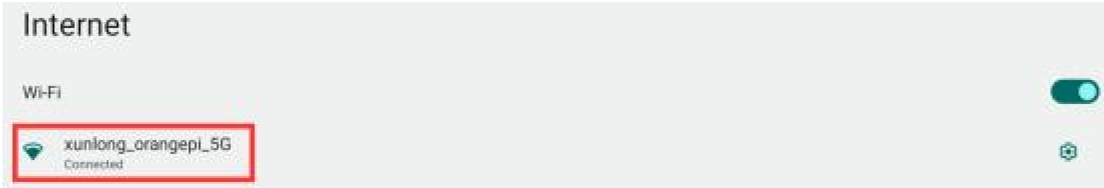
6) Then select the Wi-Fi you want to connect to and the password input interface shown in the figure below will pop up.



7) Then use the keyboard to enter the password corresponding to the Wi-Fi, and then use the mouse to click the Enter button in the virtual keyboard to start connecting to the Wi-Fi.



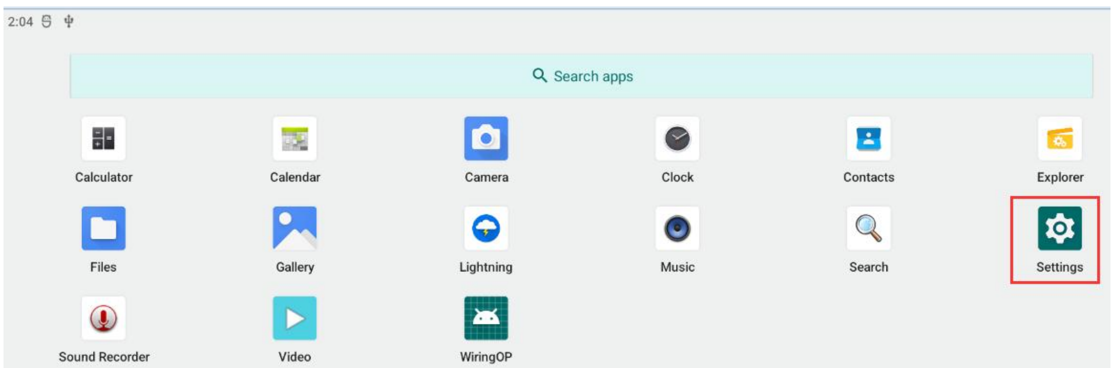
8) The display after Wi-Fi connection is successful is as shown below:



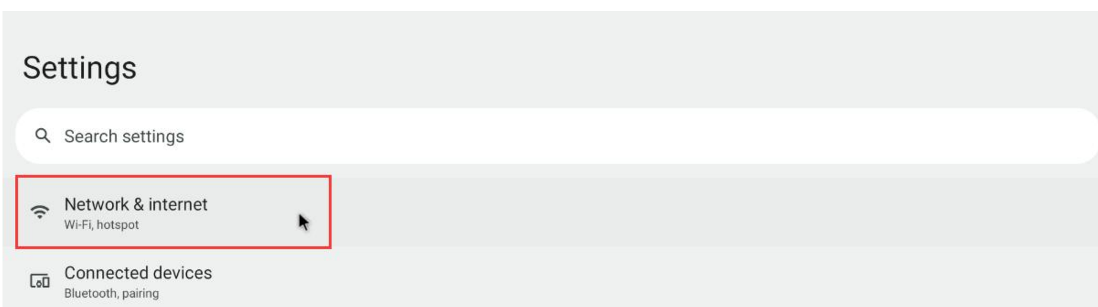
7. 4. How to use Wi-Fi hotspot

1) First, make sure the Ethernet port is connected to the network cable and can access the Internet normally.

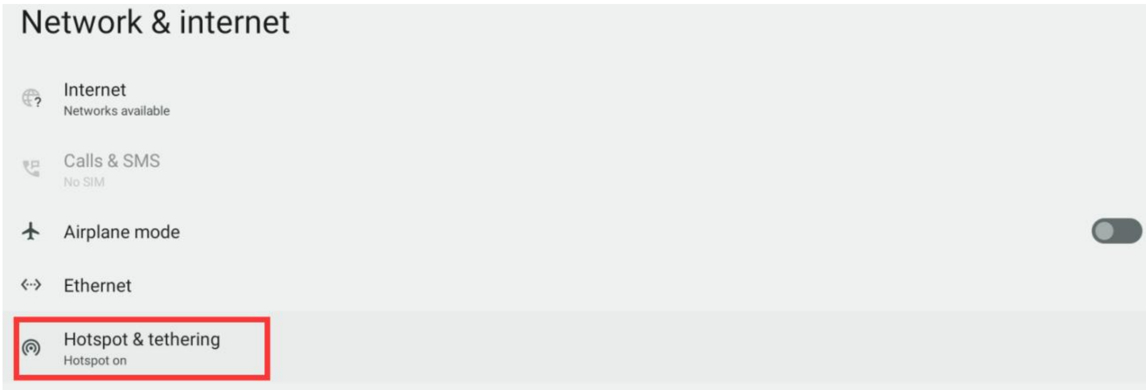
2) Then select **Settings**



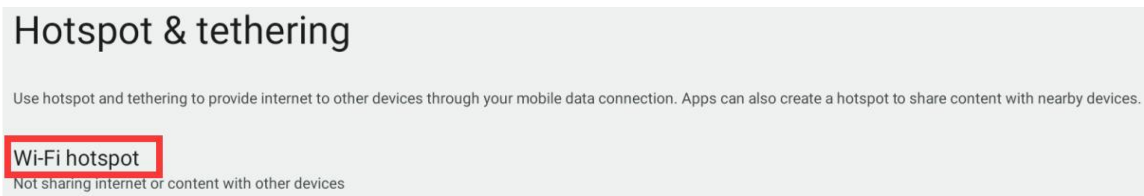
3) Then select **Network & internet**



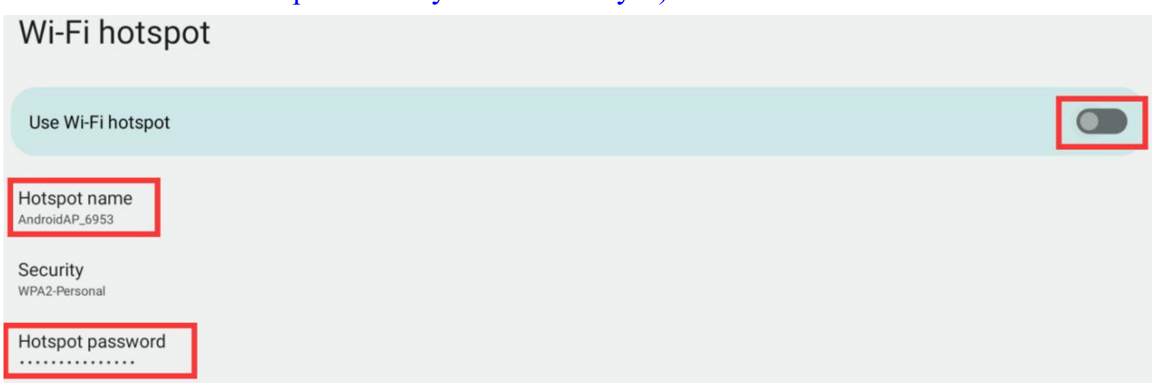
4) Then select **Hotspot & tethering**



5) Then select **Wi-Fi hotspot**



6) Then turn on the **Wi-Fi hotspot**. You can also see the name and password of the generated hotspot in the 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 hotspot before you can modify it)

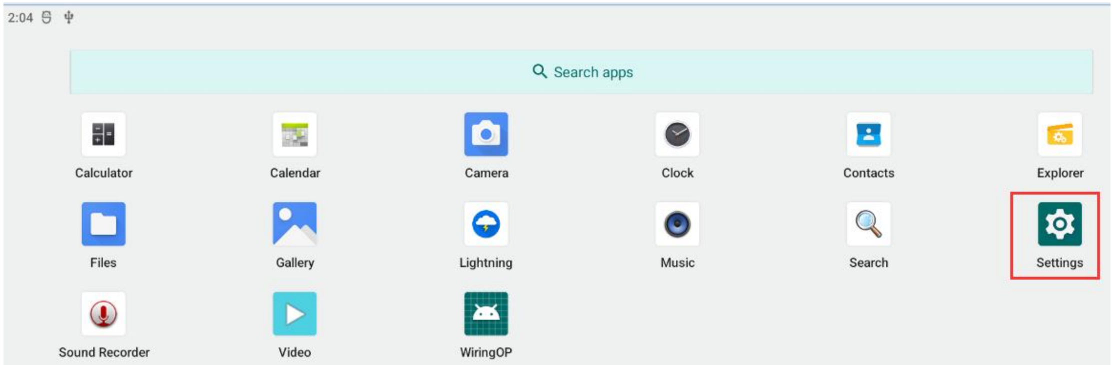


7) Now you can take out your mobile phone. If everything is normal, you can find the WIFI hotspot with the same name **(here is AndroidAP_6953)** displayed under **Hotspot name** in the above picture in the WI-FI list searched by the mobile phone. Then you can click **AndroidAP_6953** to connect to the hotspot. The password can be seen under **Hotspot password** in the above picture.

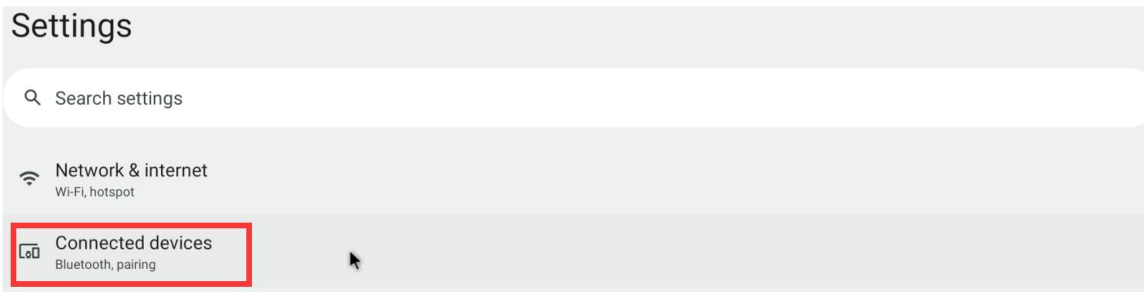
8) After the connection is successful, the following figure will be displayed (the interface of different mobile phones may be different, the specific interface is subject to the display of your mobile phone). At this time, you can open a web page on your mobile phone to see if you can access the Internet. If you can open the web page normally, it means that the **WI-FI Hotspot** of the development board can be used normally.

7. 5. Bluetooth test method

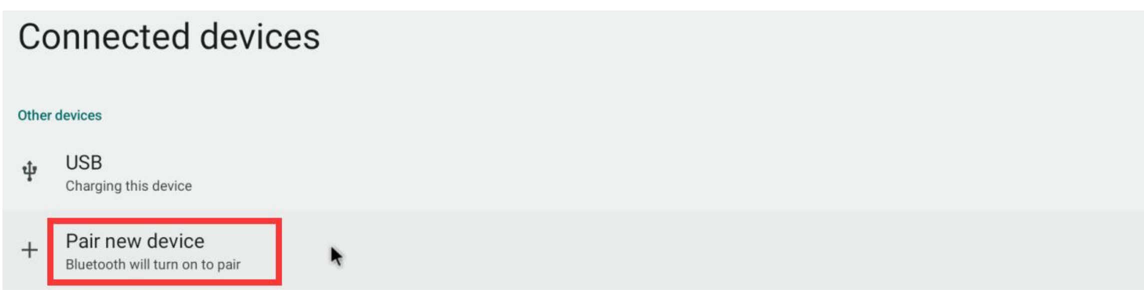
1) First click to enter **Setting**



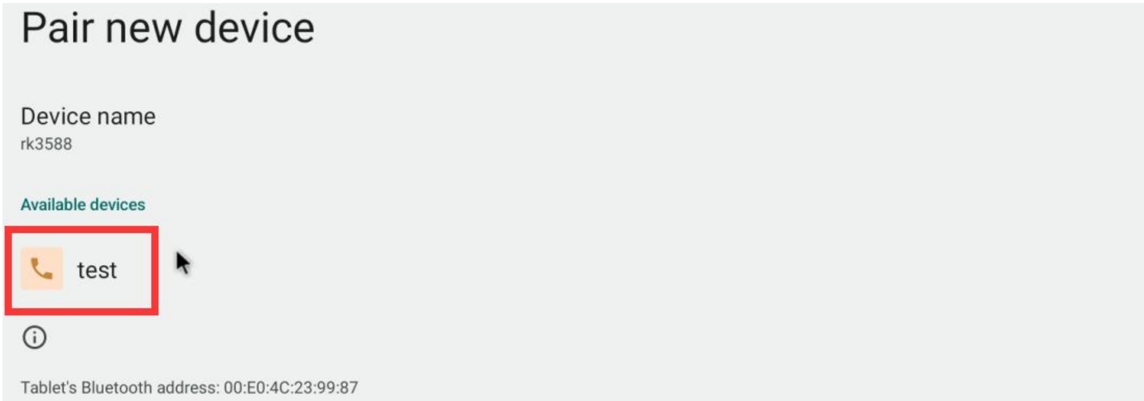
2) Then select **Connected devices**



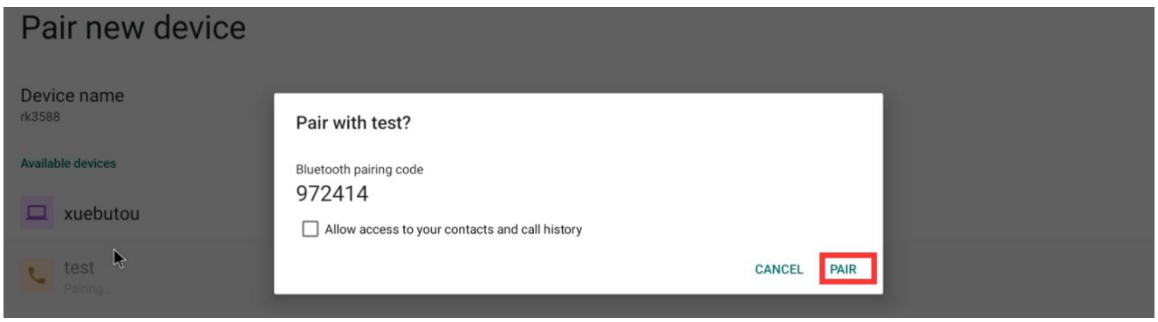
3) Then click **Pair new device** to turn on Bluetooth and start scanning for surrounding Bluetooth devices.



4) The Bluetooth devices found will be displayed under **Available devices**

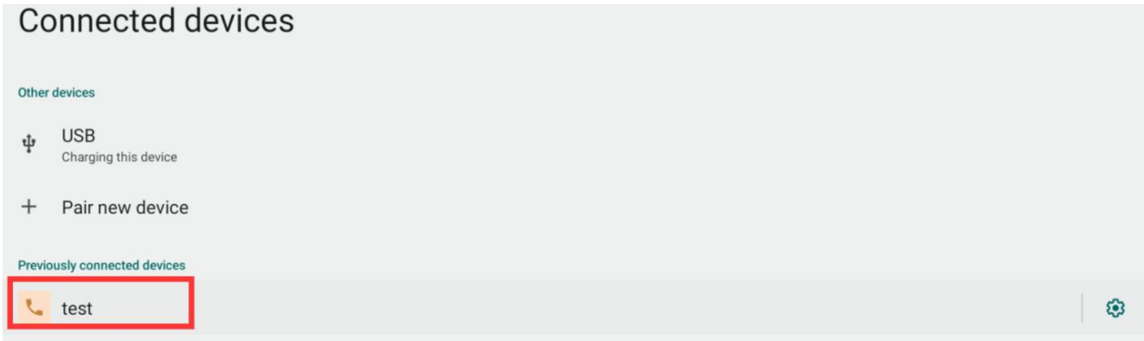


5) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, use your mouse to select the **Pair** option

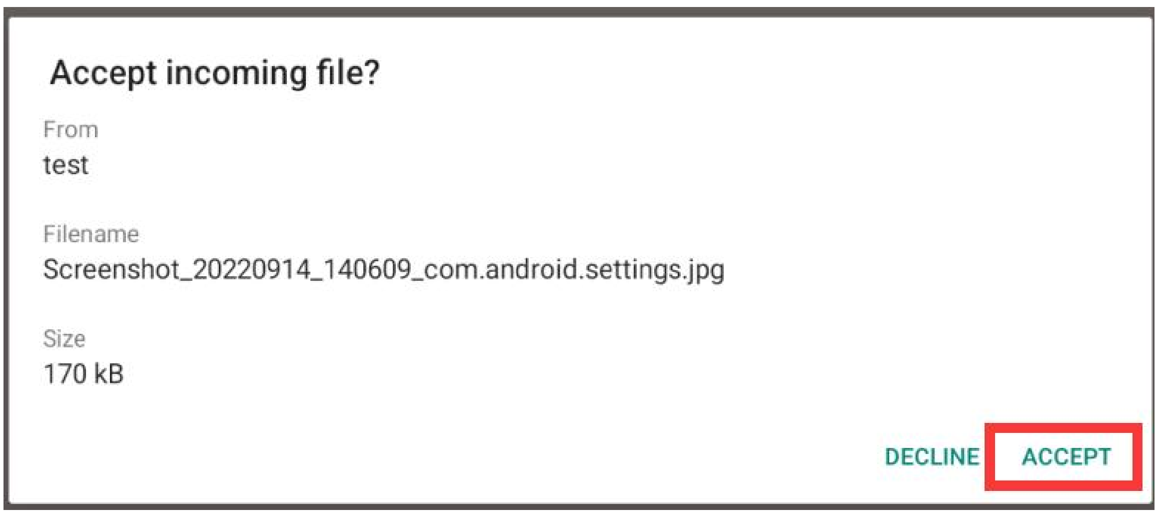


6) Here we test the Bluetooth configuration process between the development board and the Android phone. At this time, the following confirmation interface will pop up on the phone. Click the pairing button on the phone to start the pairing process.

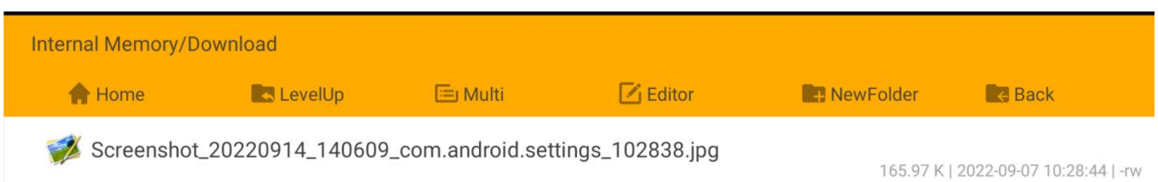
7) After pairing is complete, you can see the paired Bluetooth devices as shown below



8) At this time, you can use the mobile phone Bluetooth 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 pictures sent by the mobile phone.



9) The pictures received by the Android system Bluetooth of the development board can be viewed by opening the **Download** directory in the file manager



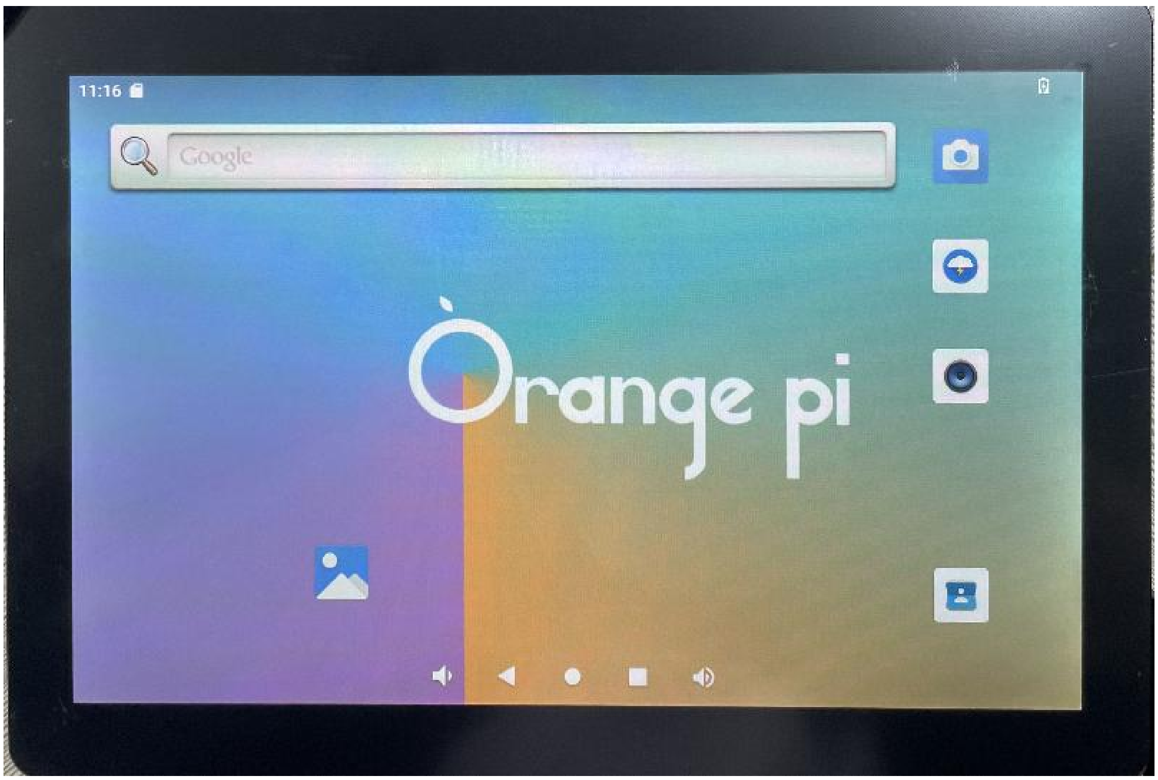
7. 6. How to use 7.6.10.1 inch MIPI screen

1) First, you need to assemble the screen. Please refer to the assembly method of

10.1-inch MIPI screen

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

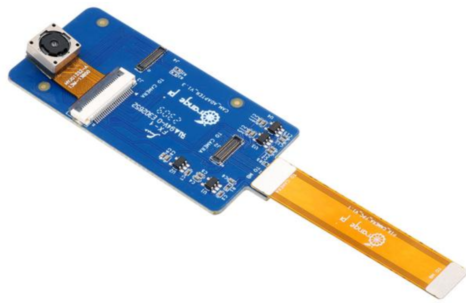
3) Connect the assembled screen to the LCD interface of the development board, connect the Type-C power supply to the board, and power it on. After the system starts, you can see the screen display as shown below



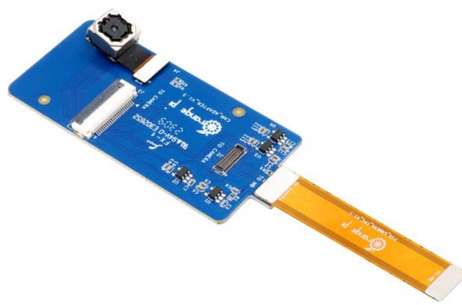
7. 7. Test Methods for OV13850 and OV13855 MIPI Cameras

Currently the development board supports two MIPI cameras, OV13850 and OV13855. The specific pictures are as follows:

a. 13MP OV13850 camera with MIPI interface



b. 13MP OV13855 camera with MIPI interface



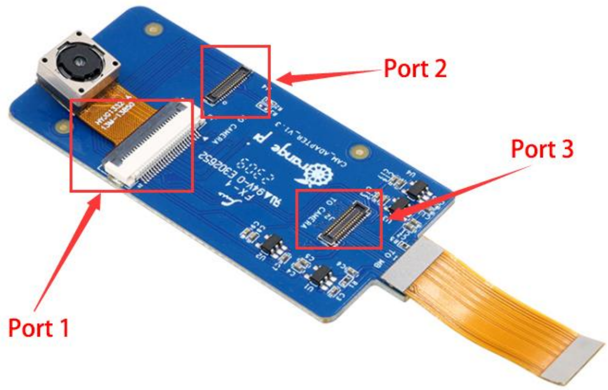
The adapter board and FPC cable used by OV13850 and OV13855 cameras are the same, but the two cameras are connected to the adapter board in different positions. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked with **TO MB** needs to be plugged into the camera interface of the development board, and the end marked with **TO CAMERA** needs to be plugged into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board. Only one can be connected at a time, as shown in the following figure:

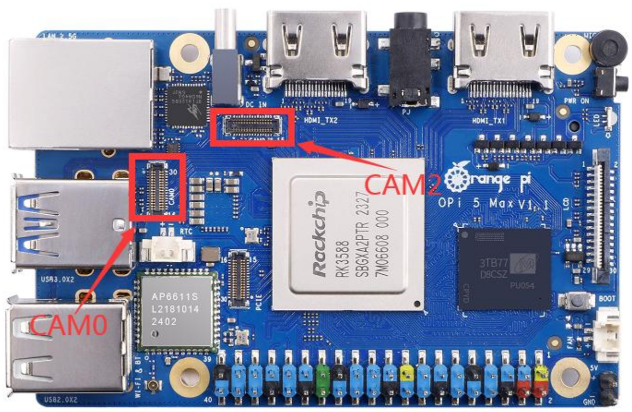
a. **Interface 1 is connected to the OV13850 camera**

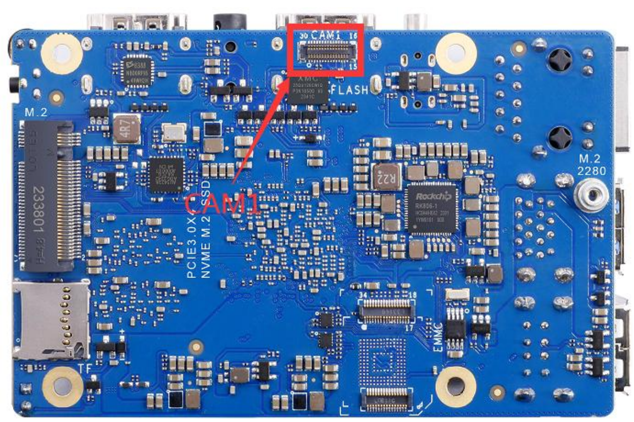
b. **Interface 2 connects to OV13855 camera**

c. Interface 3 is not used, just ignore it.



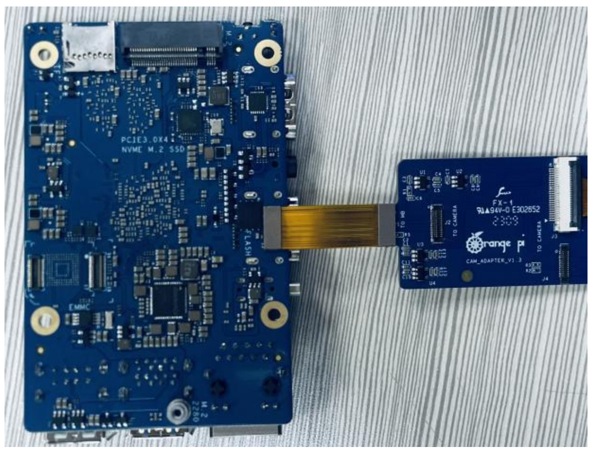
There are three camera interfaces on the Orange Pi 5 Max development board. In the Android system, only Cam0 and Cam1 are enabled by default. We define the positions of Cam0, Cam1, and Cam2 as shown in the following figure:





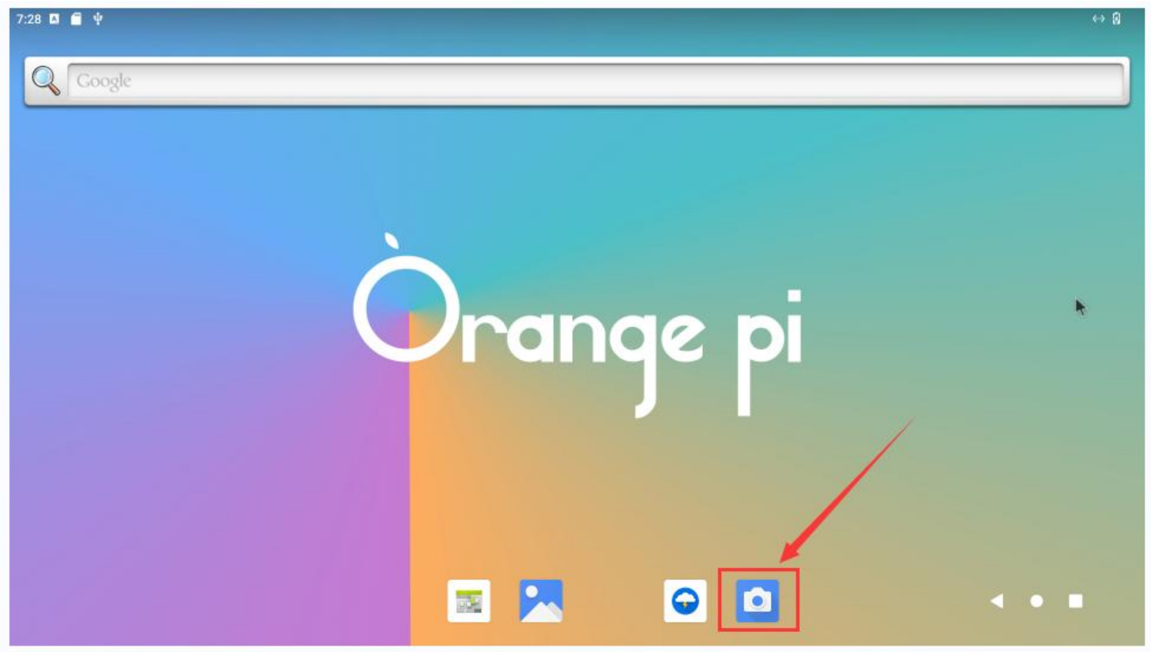
The method of inserting the camera into the Cam0 interface of the development board is as follows:

The method of inserting the camera into the Cam1 interface of the development board is as follows:

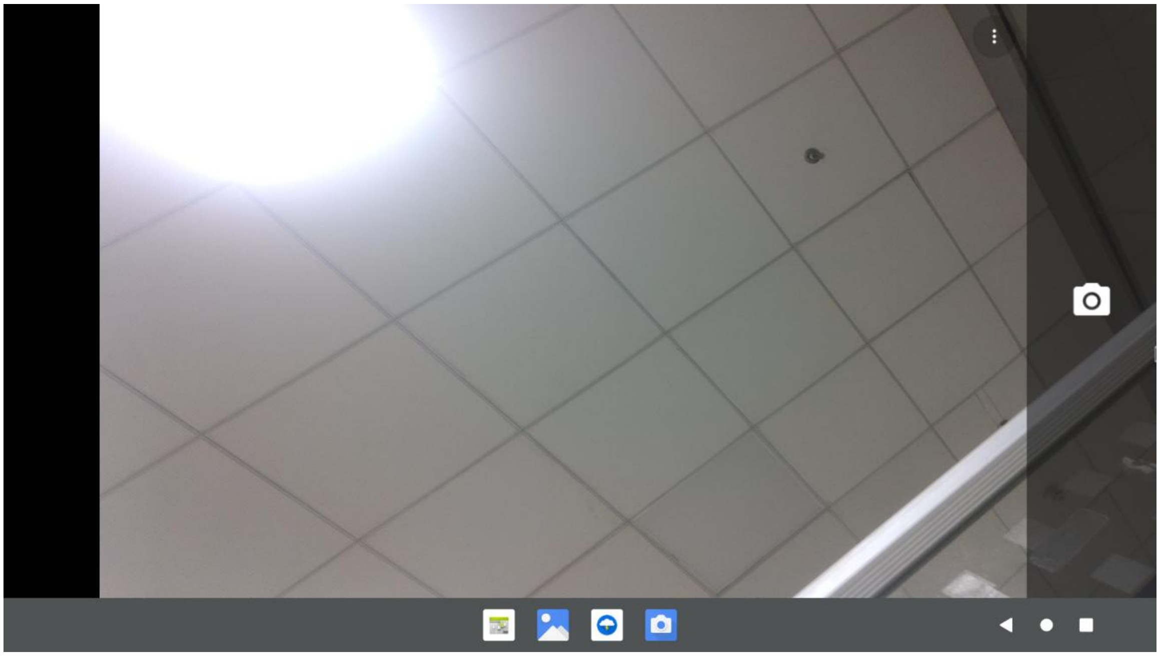


After connecting the camera to the development board, we can use the following method to test the camera:

a. Open the Camera APP on the desktop



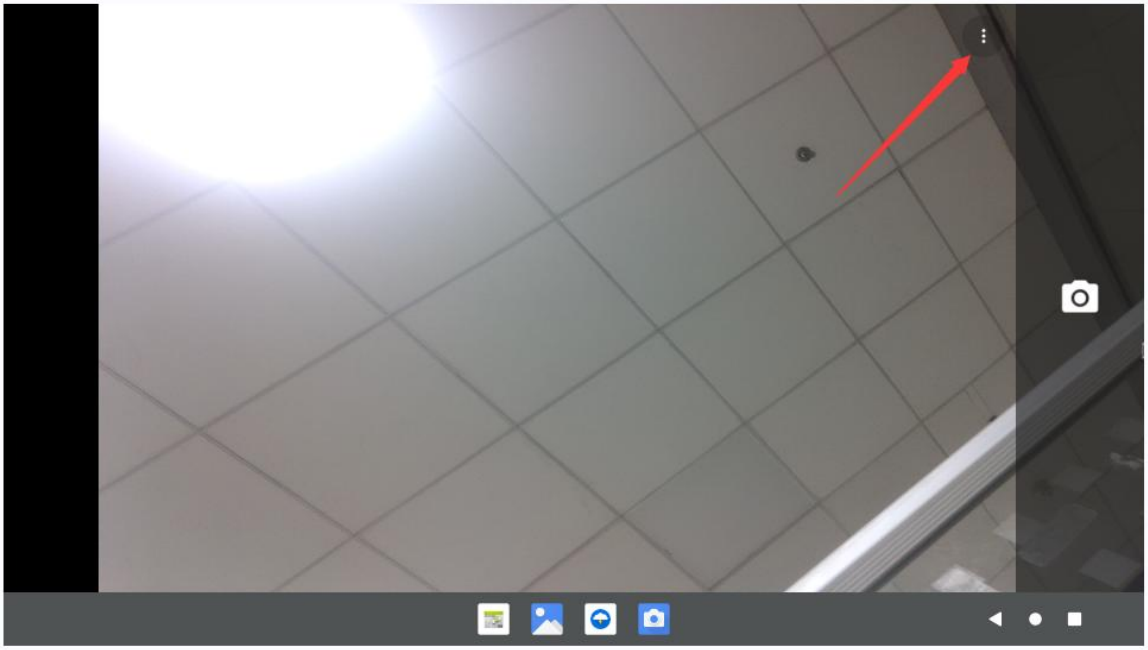
b. Then you can see the camera preview



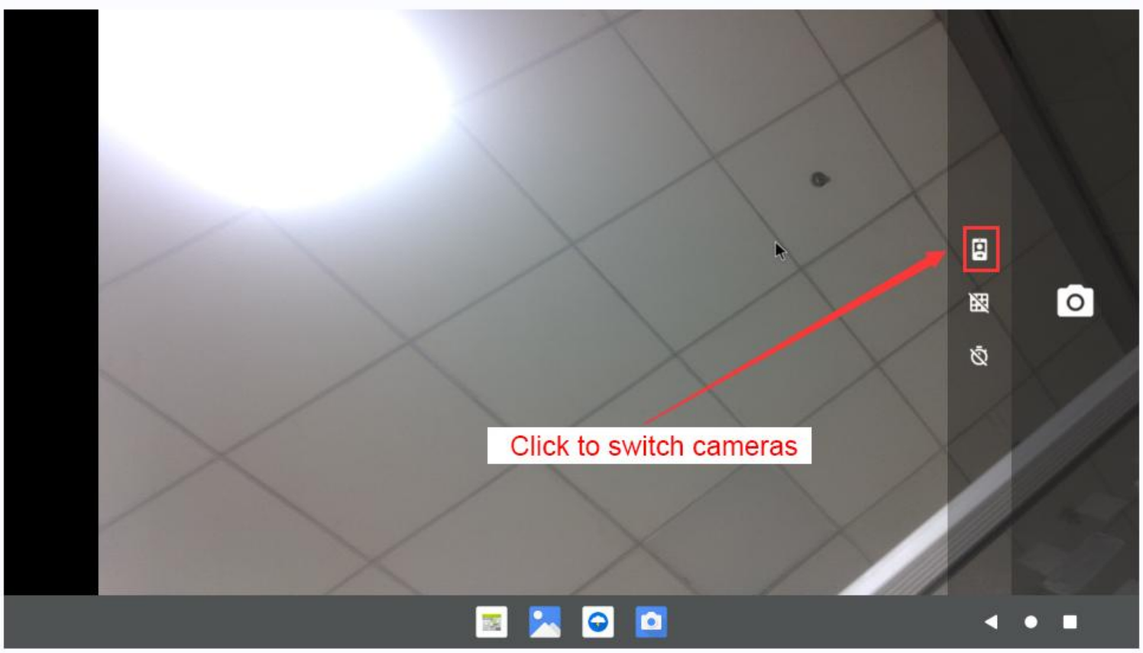
In addition to a single camera, we can also connect two cameras at the same time. After connecting the dual cameras, just like the previous steps, open the camera app and you can see the image of one of the cameras.

To switch to another camera:

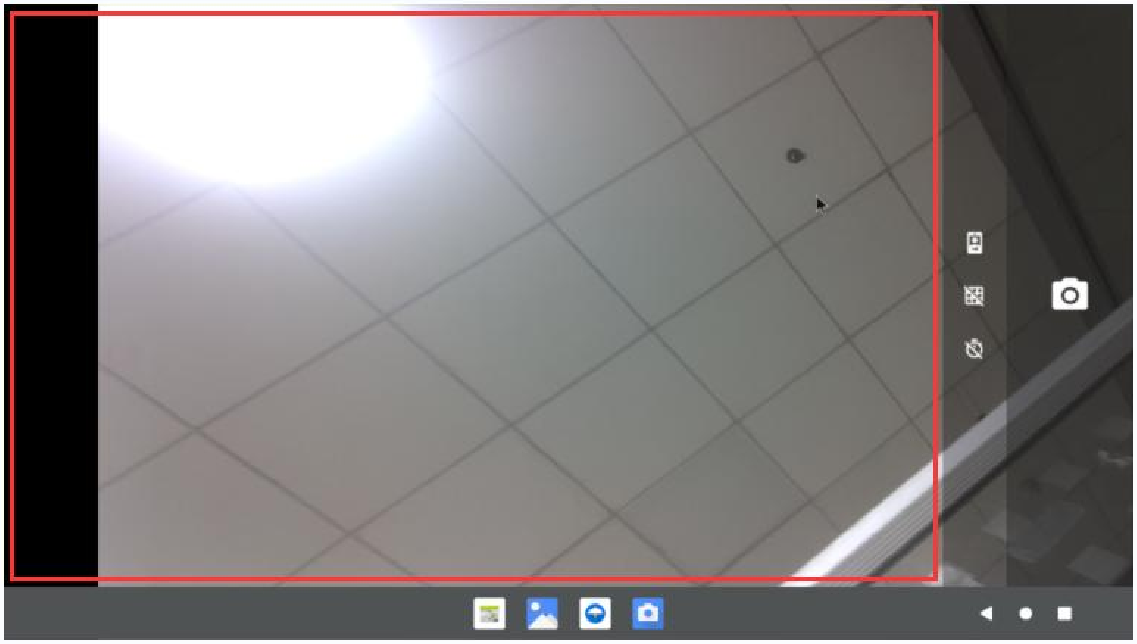
a. First click on the three dots in the upper right corner



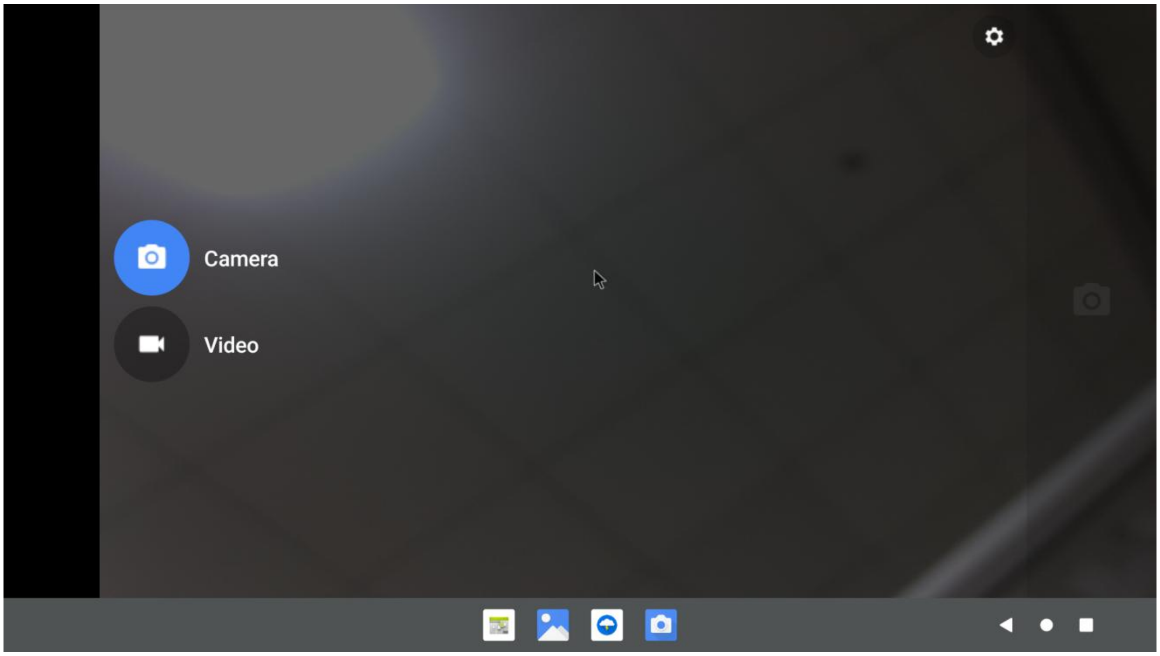
b. Then click the position shown in the figure below to switch the camera



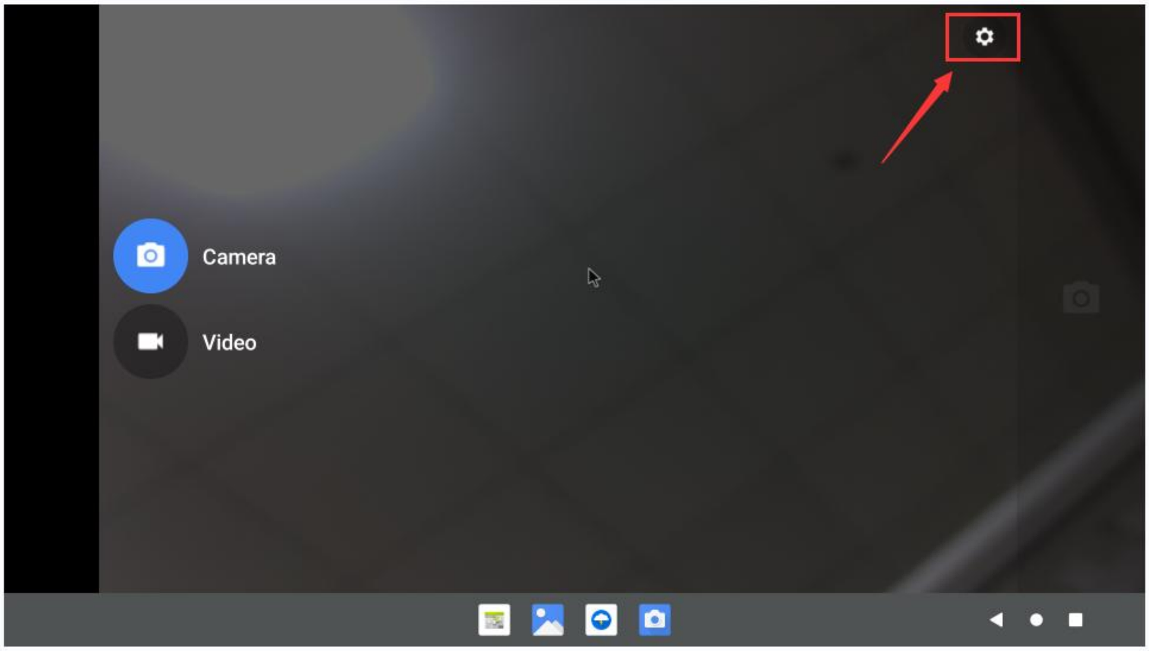
In the area marked with a red box in the camera app, hold down the mouse and drag it to the right to bring up the photo and video switching interface.



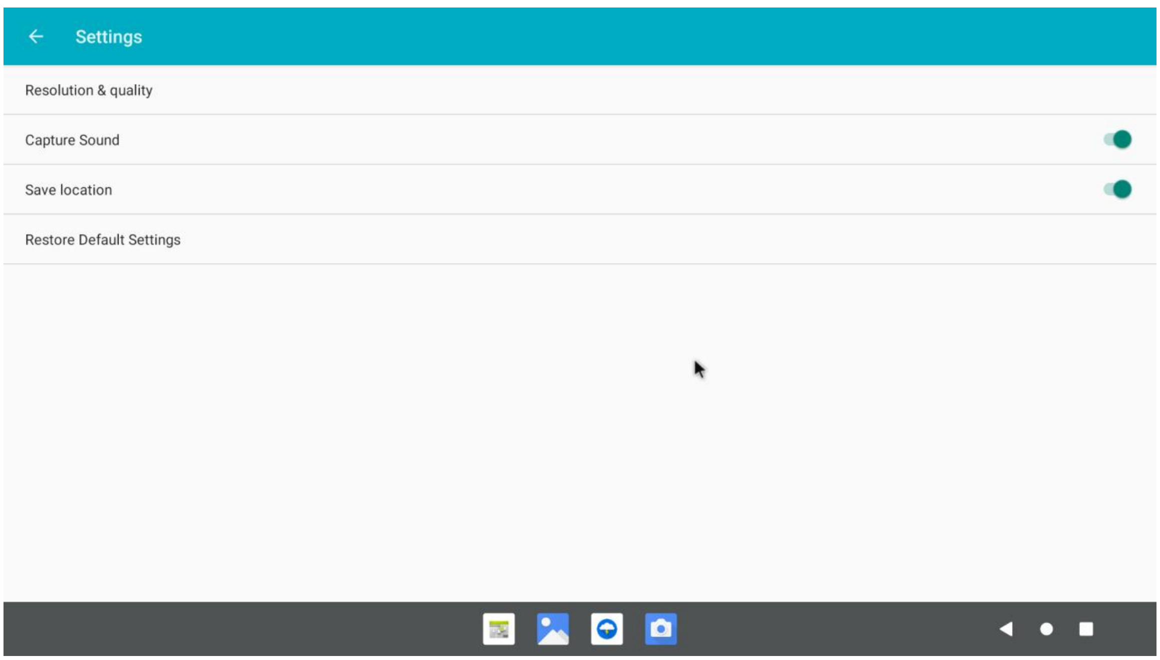
The switching interface between taking photos and recording videos is shown below. Click **Video** to switch to recording mode.



Click the position shown in the figure below to enter the camera setting interface



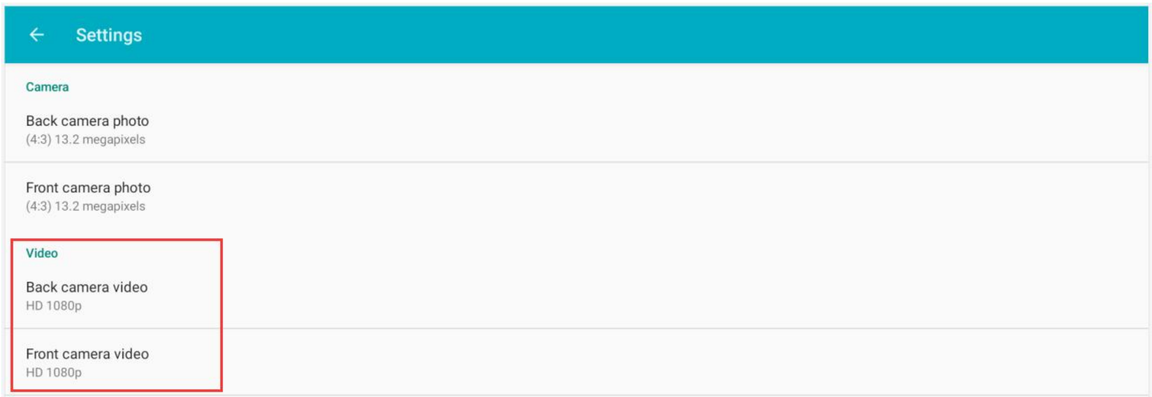
The camera settings interface is as follows:



Currently tested, OV13850 does not support 4K video recording (OV13855 does), and only supports 1080p at most. When recording video, please switch the video format to 1080p in the settings. The steps are as follows:

a. First enter the camera app's settings interface, then click **Resolution & quality**

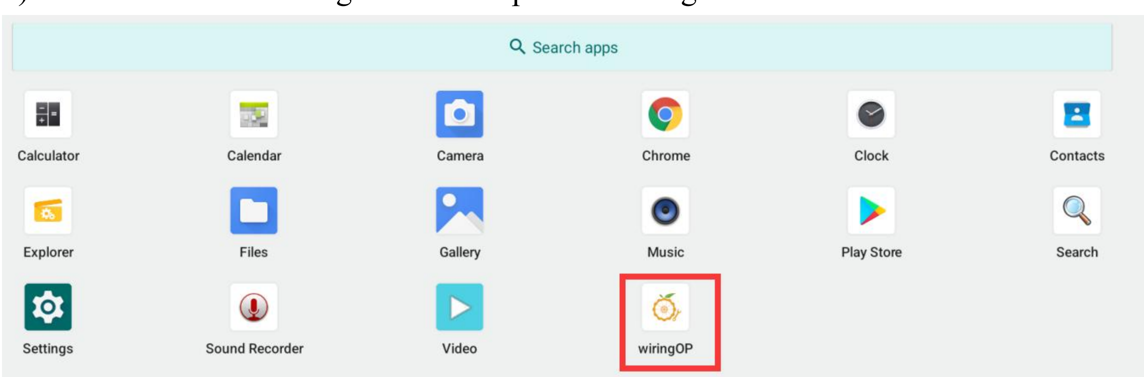
b. Then set the video format to 1080p in **Video**



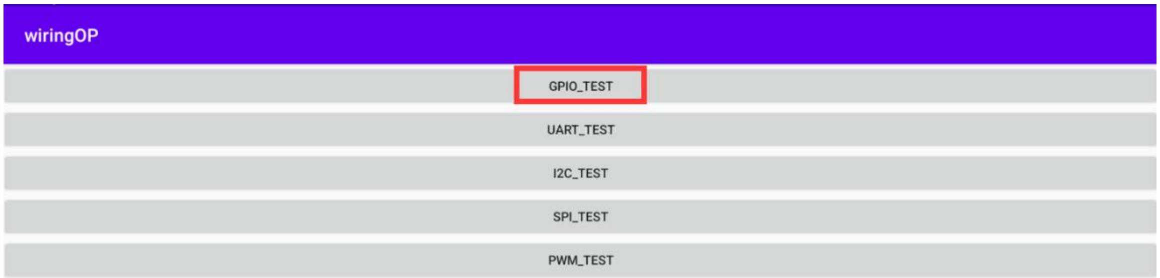
7. 8. 40pin interface GPIO, UART, SPI and PWM test

7. 8. 1. 40pin GPIO port test

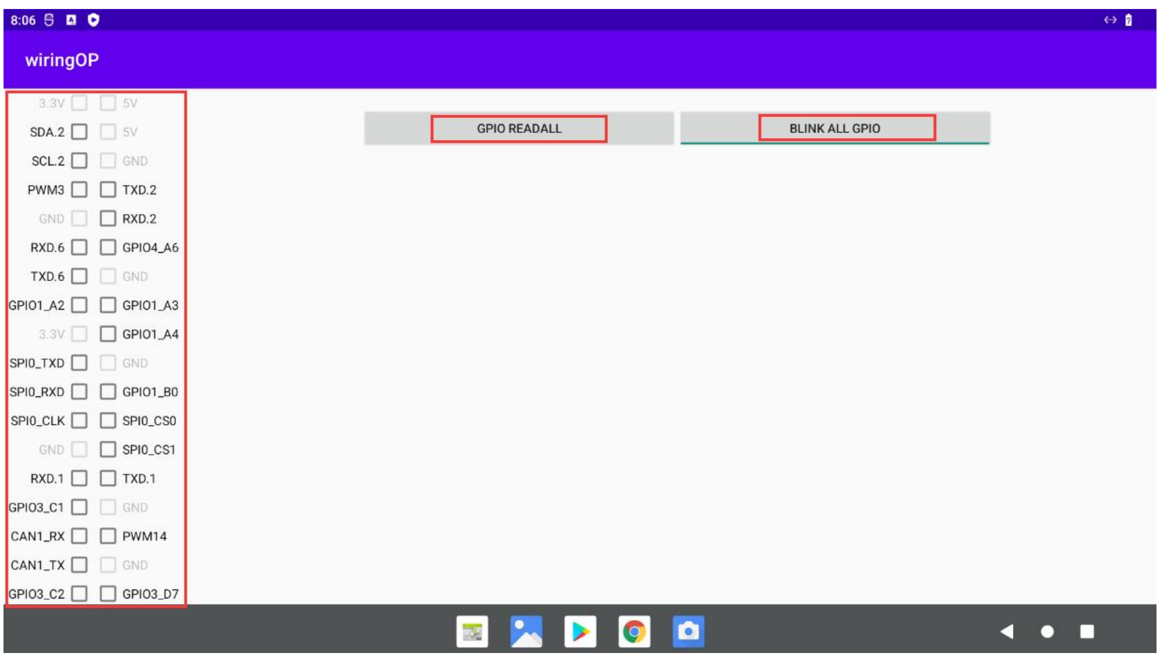
1) First click on the wiringOP icon to open the wiringOP APP



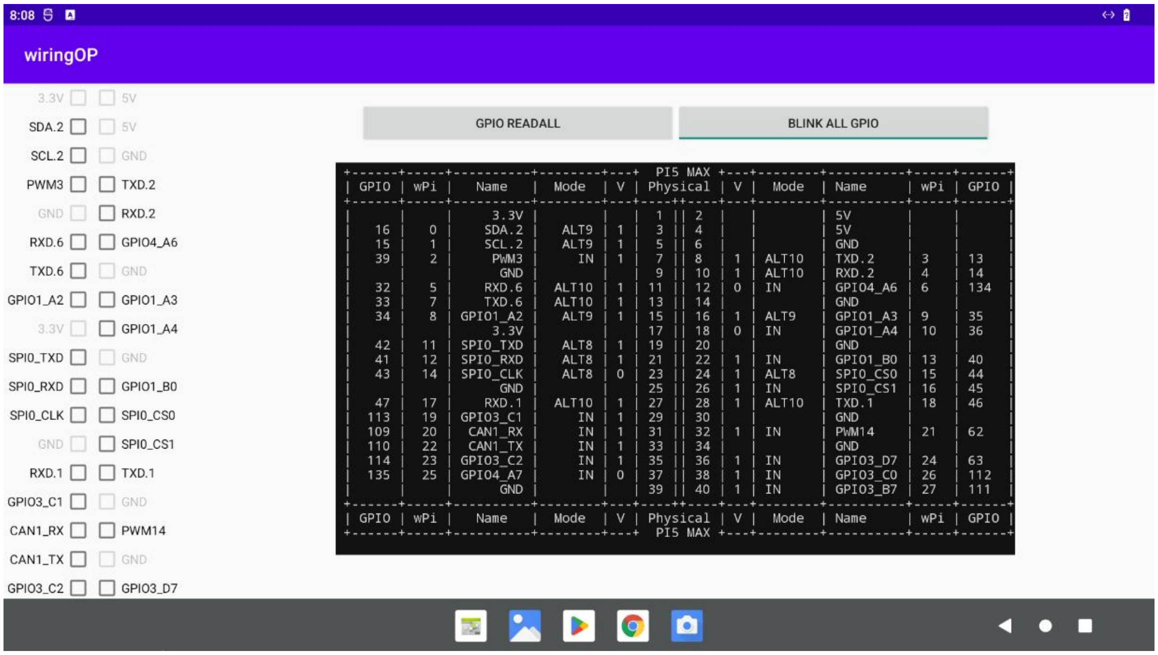
2) The main interface of wiringOP APP is shown as below, then click **GPIO_TEST** button to open the GPIO test interface



3) The GPIO test interface is shown in the figure below. The two rows of **CheckBox** buttons on the left correspond to the 40pin pins one by one. When the **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT** mode and the pin level will be set to high level; when it is unchecked, the GPIO pin level will be set to low level; when the **GPIO READALL** button on the right is clicked, the wPi number, GPIO mode, pin level and other information can be obtained; when the **BLINK ALL GPIO** button is clicked, the program will control the 28 GPIO ports to switch high and low levels continuously.

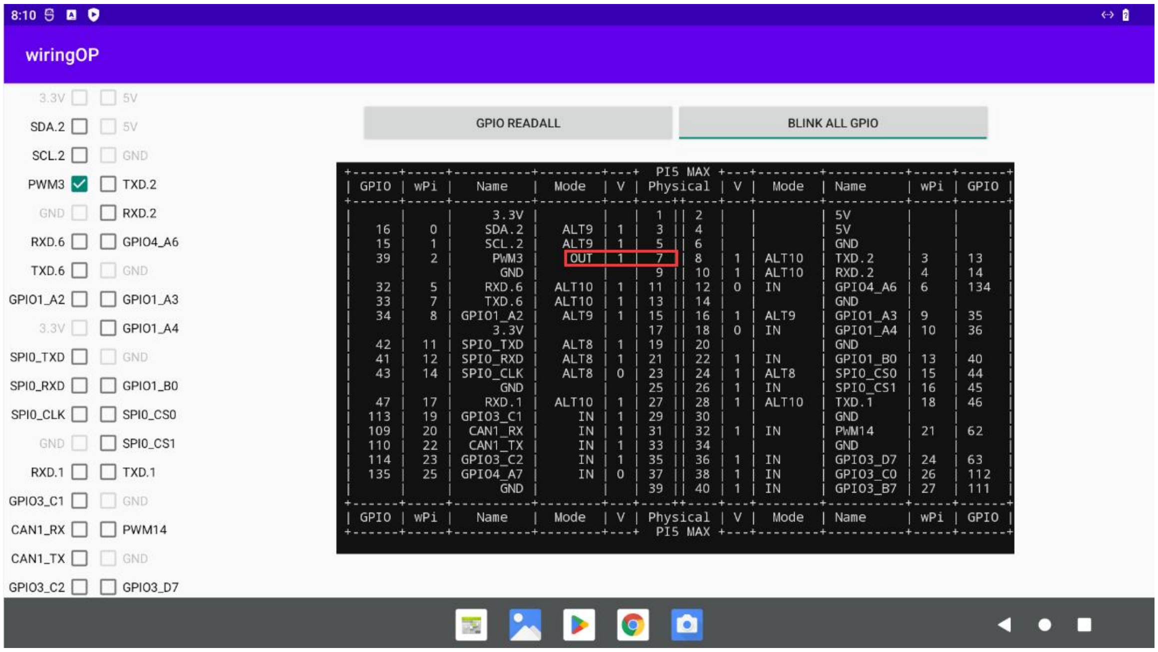


4) Then click the **GPIO READALL** button, and the output information is as shown below:

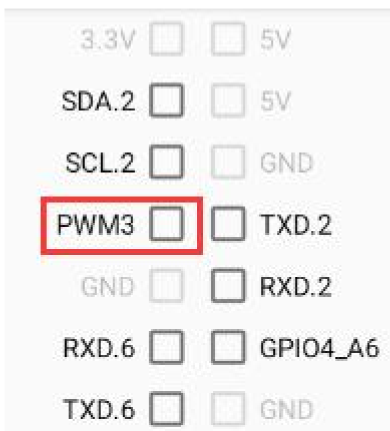


5) There are 16 GPIO ports available in the 40 pins of the development board. The following example shows how to set the high and low levels of the GPIO port, using pin 7, which corresponds to GPIO1_A7 and wPi number 2. First, click the **CheckBox** button corresponding to pin 7. When the button is selected, pin 7 will be set to a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **3.3v**, it means that the high level is set successfully.

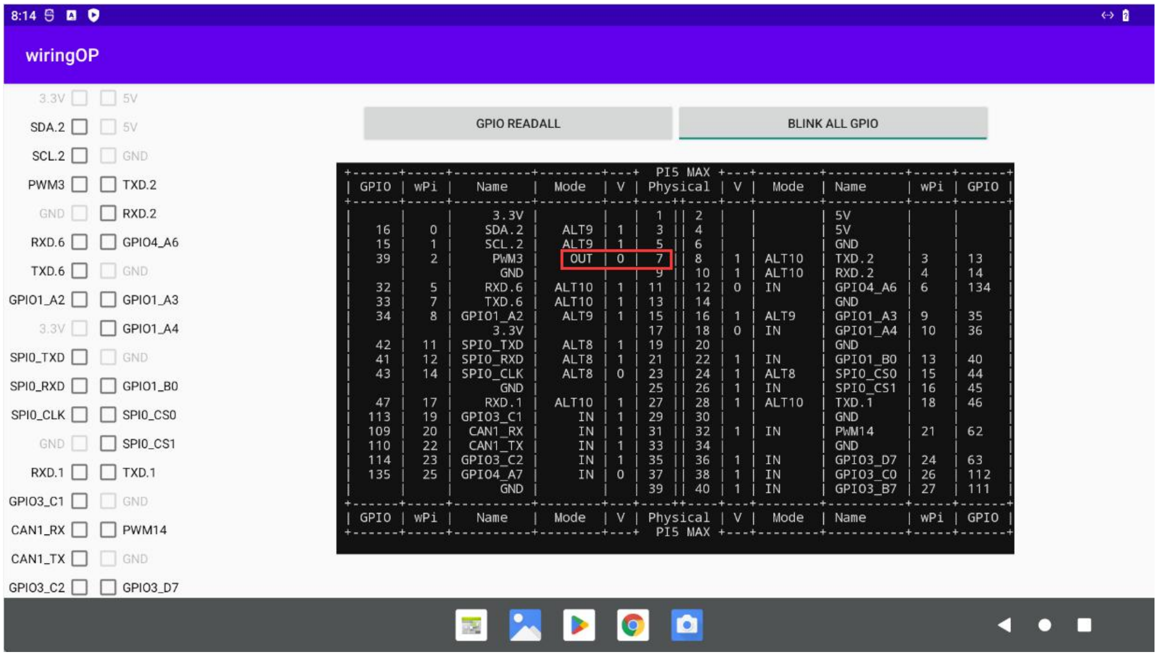
6) Then click the **GPIO READALL** button, you can see that the current mode of pin 7 is **OUT** and the pin level is high.



7) Click the **CheckBox** button in the figure below again to uncheck the status. Pin 7 will be set to a low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level is set successfully.



8) Then click the **GPIO READALL** button, you can see that the current mode of pin 7 is OUT and the pin level is low

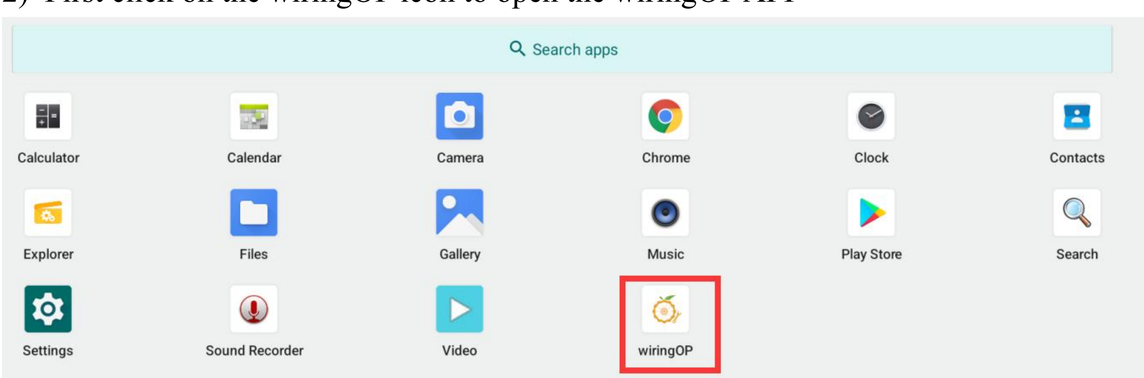


7. 8. 2. 40pin UART test

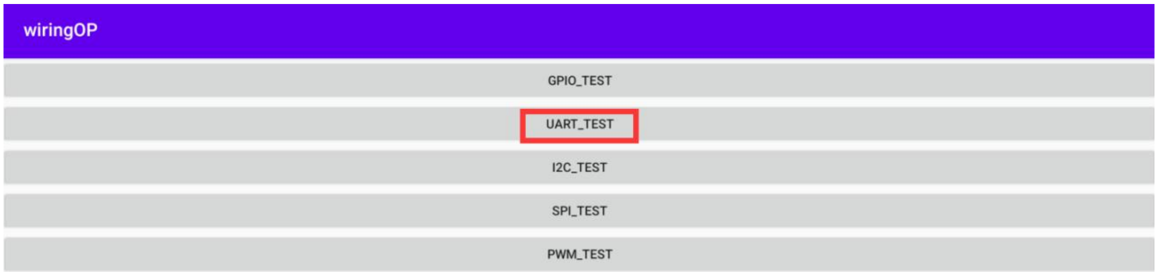
1) In Android, three serial ports, UART1, UART3 and UART6, are enabled by default. The corresponding pins in 40pin are shown in the following table, and the corresponding device nodes are **/dev/ttyS1**, **/dev/ttyS3** and **/dev/ttyS6**

UART Bus	RX corresponds to 40pin	TX corresponds to 40pin
UART1	Pin 27	Pin 28
UART3	Pin 33	Pin 31
UART6	Pin 11	Pin 13

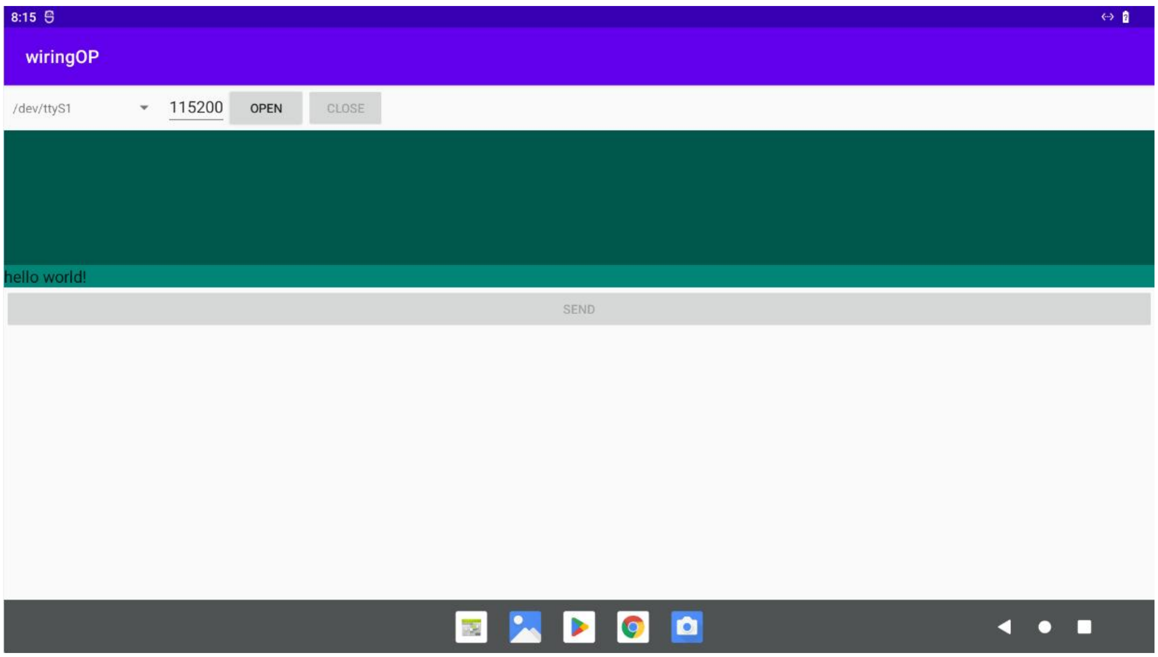
2) First click on the wiringOP icon to open the wiringOP APP



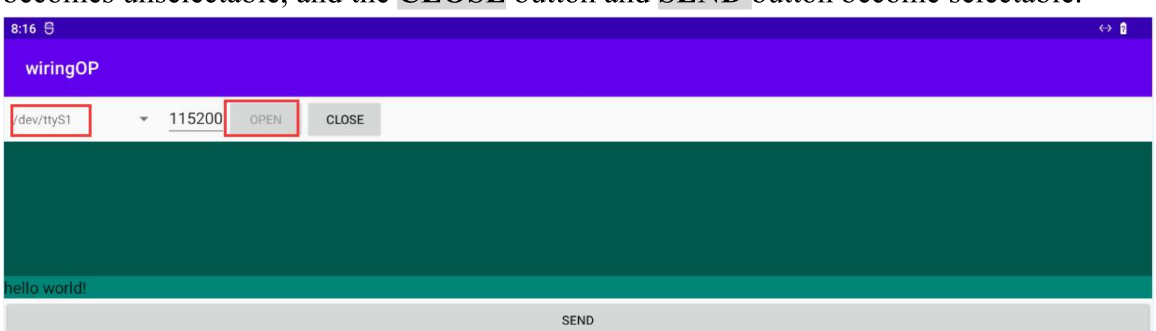
3) The main interface of wiringOP APP is shown as below, then click **UART_TEST** button to open the UART test interface



4) The serial port test interface of the APP is shown in the figure below

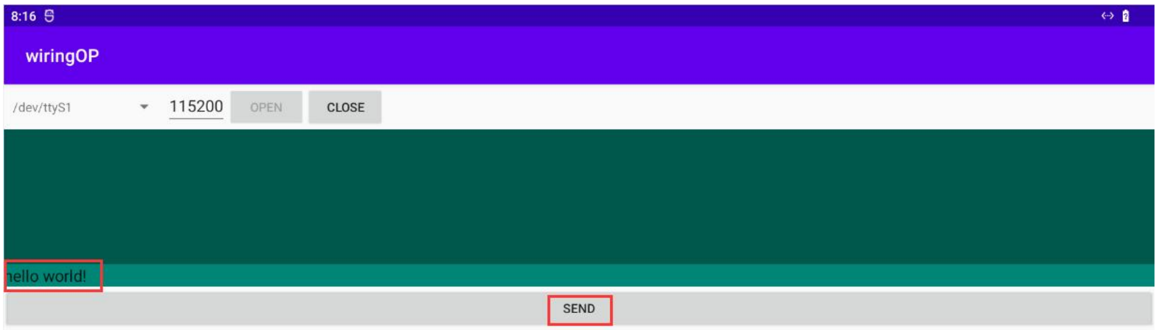


5) Then enter the baud rate you want to set in the edit box, and then click the **OPEN** button to open the **/dev/ttyS1** node. After opening successfully, the **OPEN** button becomes unselectable, and the **CLOSE** button and **SEND** button become selectable.

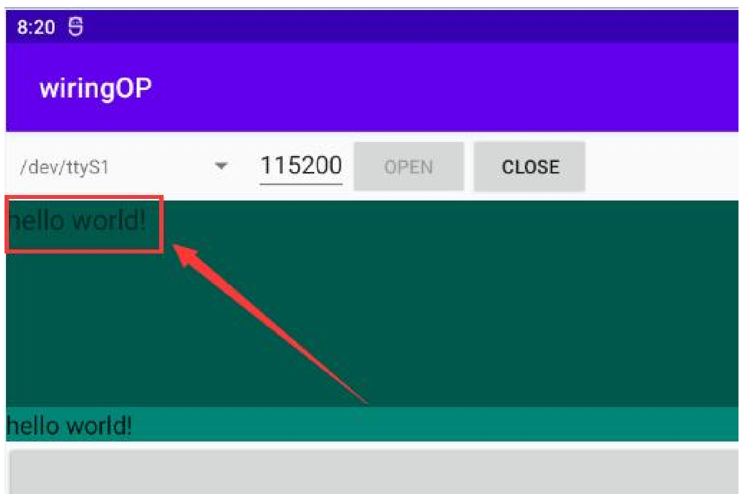


6) Then use the Dupont line to short the RXD and TXD pins of uart1

7) Then you can enter a string of characters in the send edit box below and click the **SEND** button to start sending.



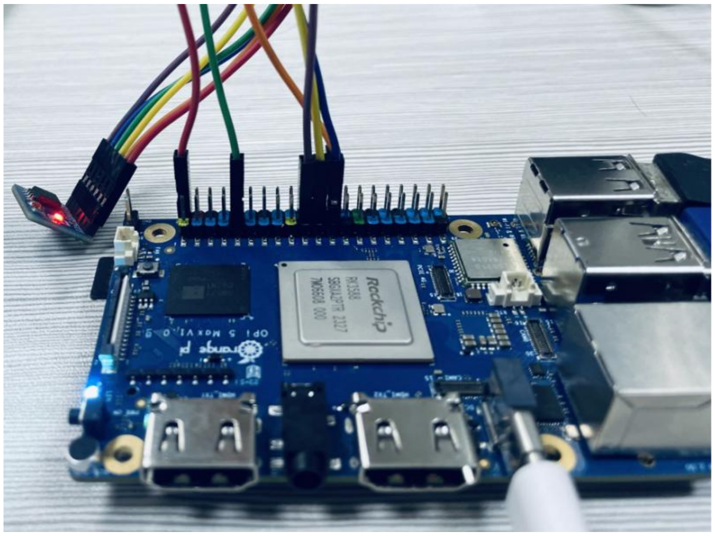
8) If everything is normal, the received string will be displayed in the receiving box



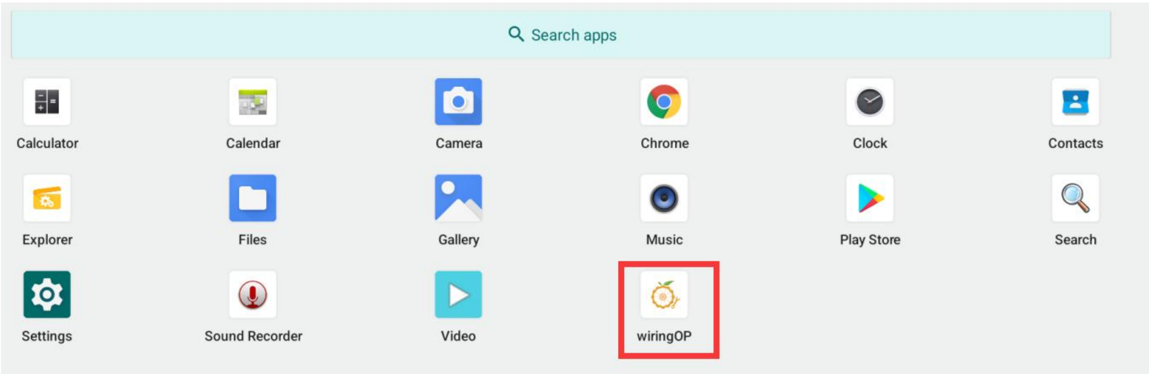
7. 8. 3. 40pin SPI test

1) **SPI0** is enabled by default in Android.

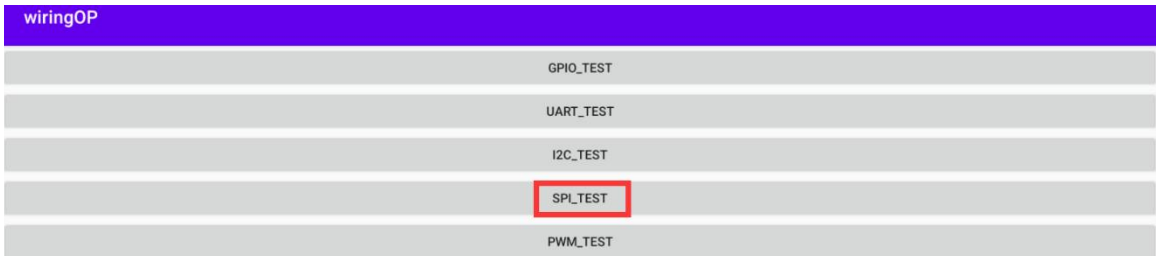
2) Here we use the w25q64 module to test the SPI interface. First, connect the w25q64 device to the SPI0 interface.



3) Then click the wiringOP icon to open the wiringOP APP

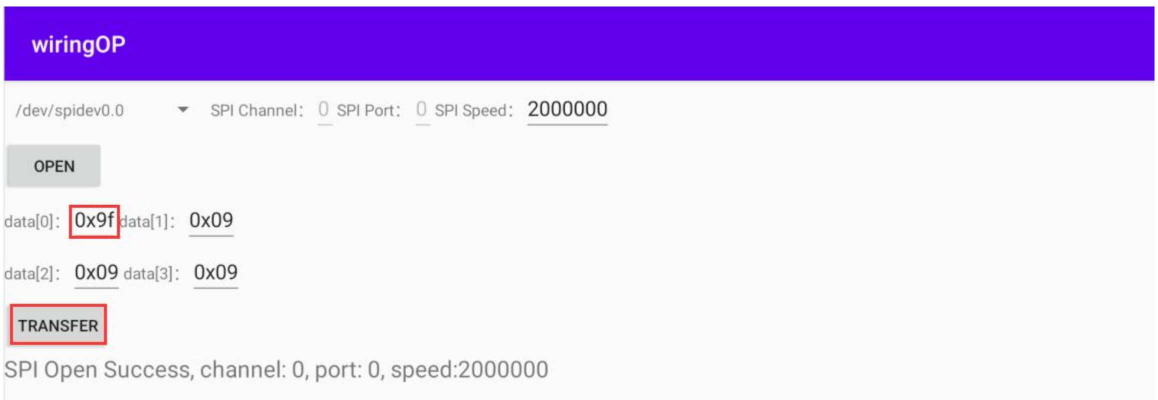


4) The main interface of wiringOP APP is shown as below. Click the SPI_TEST button to open the SPI test interface.

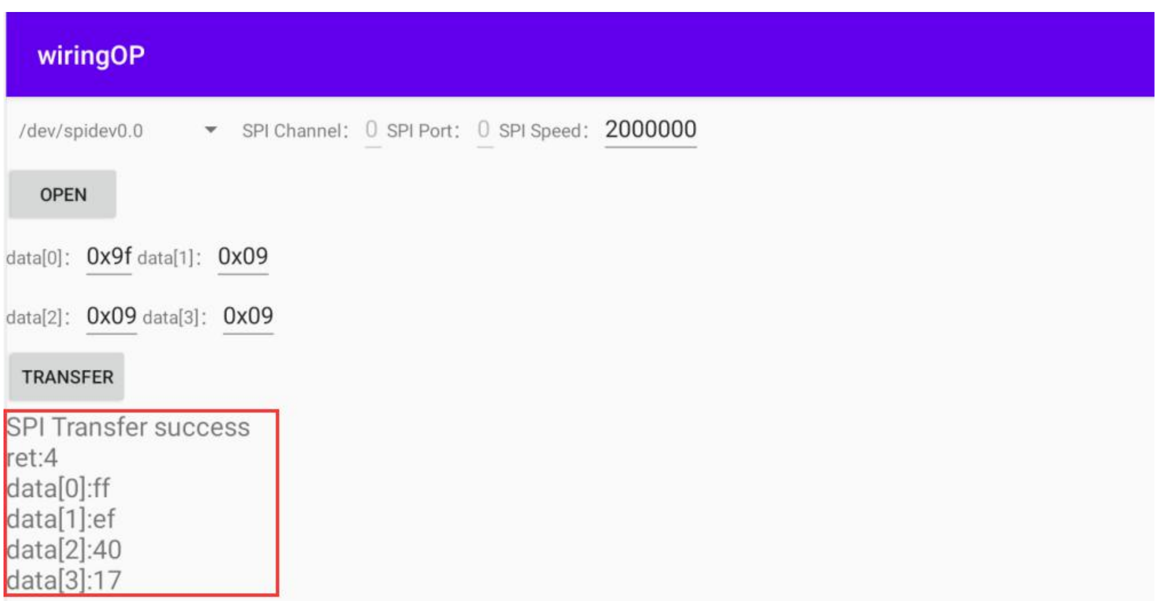


5) SPI Then click the **OPEN** button to initialize SPI

6) Then fill in the bytes to be sent, for example, read the ID information of w25q64, fill in the address 0x9f in data[0], and then click the **TRANSFER** button



7) Finally, the APP will display the ID information read



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)

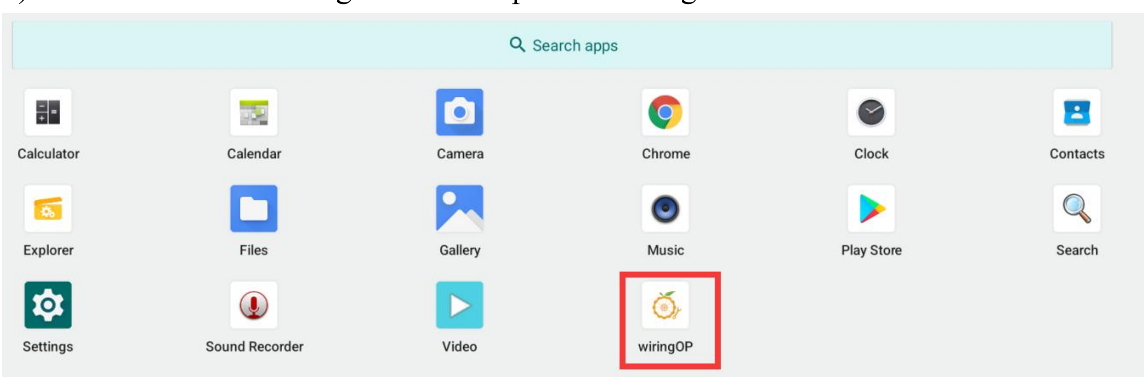
MANUFACTURER ID	(MF7 - MF0)	
Winbond Serial Flash	EFh	
Device ID	(ID7 - ID0)	(ID15 - ID0)
Instruction	ABh, 90h, 92h, 94h	9Fh
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

7. 8. 4. 40pin PWM test

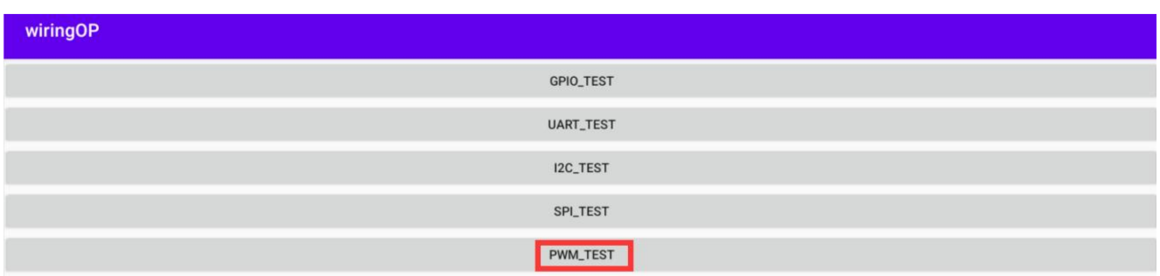
1) Android turns on **PWM3** and **PWM14** by default, and the corresponding pins are located at the 40pin as shown in the figure below

PWM Bus	Corresponding to 40pin
PWM3	Pin 7
PWM14	Pin 32

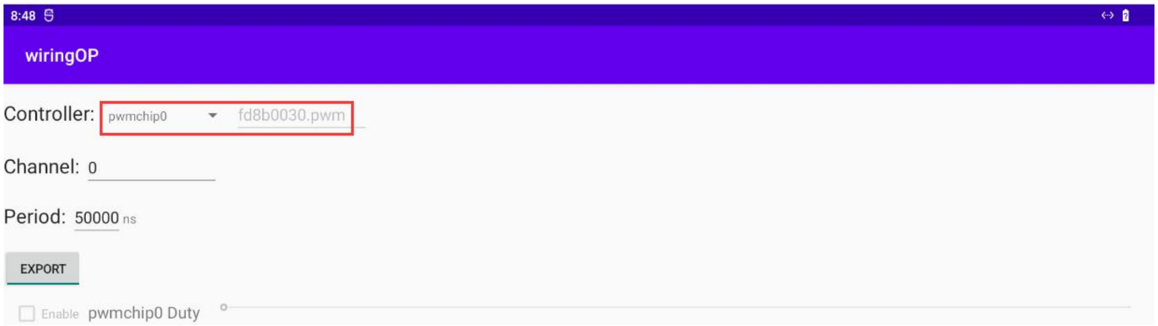
2) First click on the wiringOP icon to open the wiringOP APP



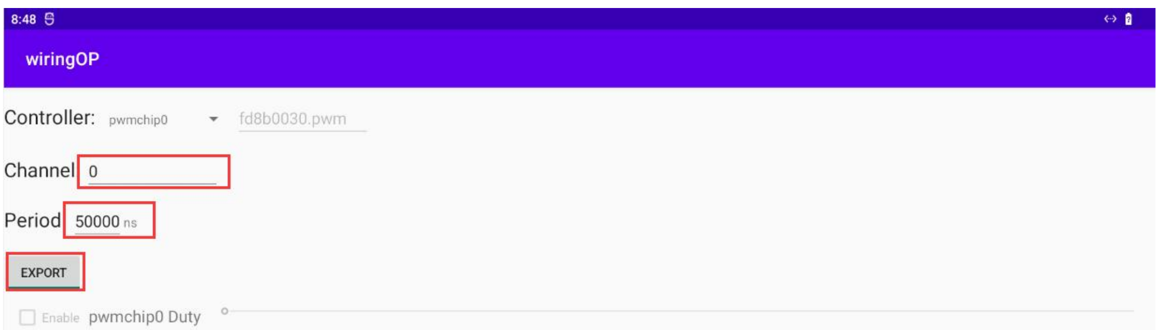
3) Then click the **PWM_TEST** button on the main interface of wiringOP to enter the PWM test interface



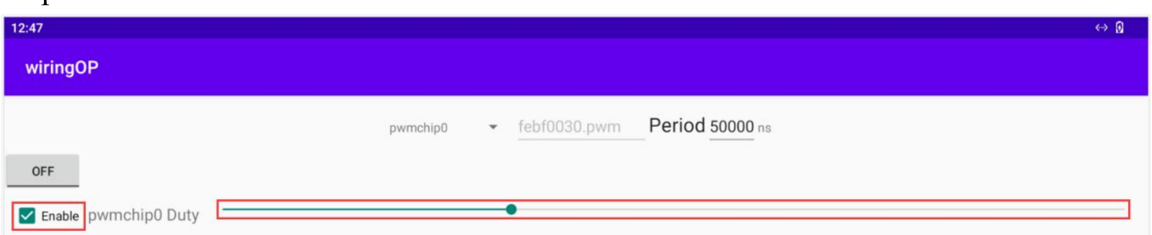
4) The base address of PWM3 is **fe8b0030**, and the base address of PWM14 is **febf0020**. Here, **fe8b0030.pwm** is displayed on the right side of pwmchip0, indicating that **PWM3** is selected.



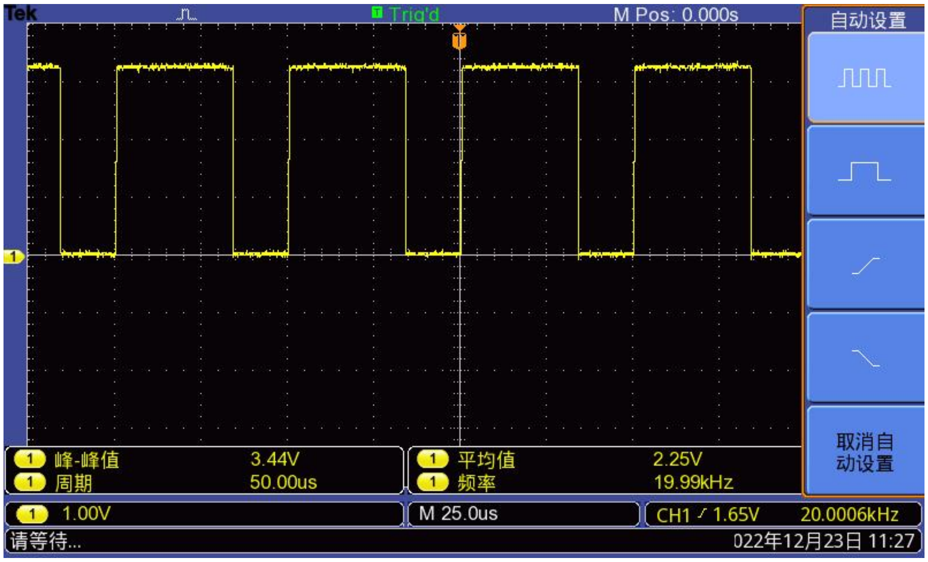
5) Then confirm the PWM channel, the default is channel 0, and confirm the PWM period. The default configuration is **50000ns**, which is converted to a PWM frequency of **20KHz**. You can modify it yourself. Click the **EXPORT** button to export **PWM3**



6) Then drag the slider below to change the PWM duty cycle, and then check Enable to output the PWM waveform.



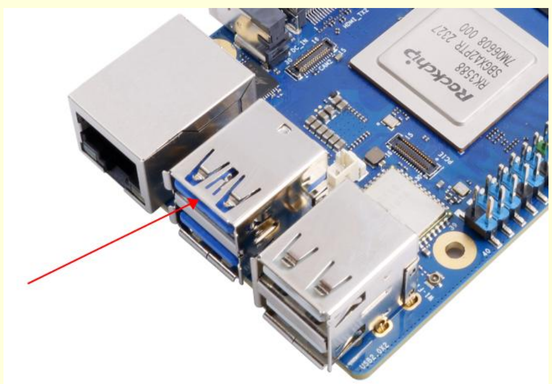
7) Then use an oscilloscope to measure the 7th pin of the 40-pin development board and you can see the following waveform.



7. 9. How to use ADB

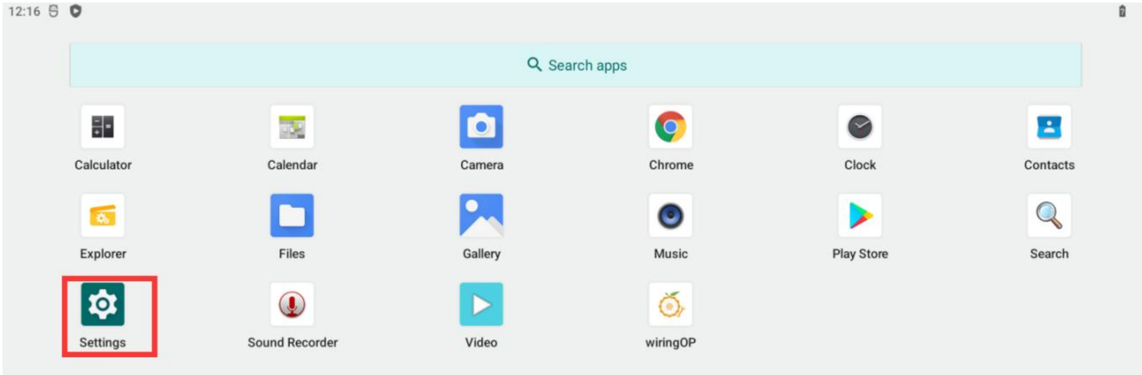
7. 9. 1. USB OTG mode switching method

The development board has 4 USB interfaces. The USB interface marked with a red frame in the figure below can support both Host mode and Device mode. The other 3 USB interfaces only support Host mode.

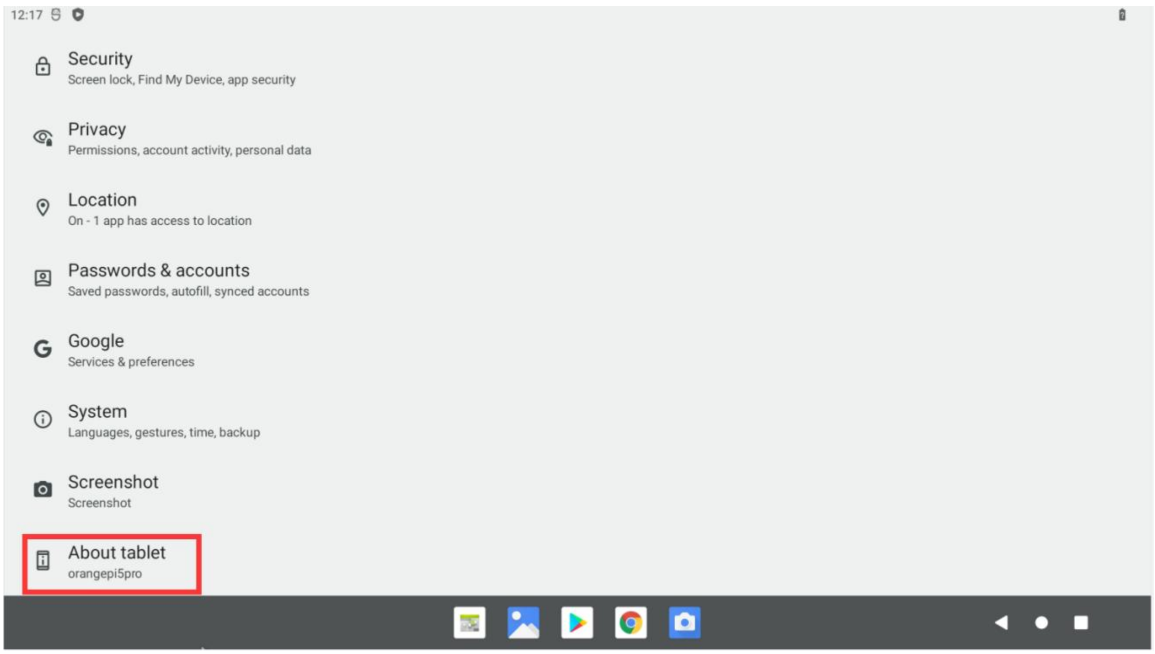


The USB OTG interface is in Host mode by default and can be used to connect USB devices such as mouse and keyboard. If you want to use ADB, you need to manually switch to Device mode.

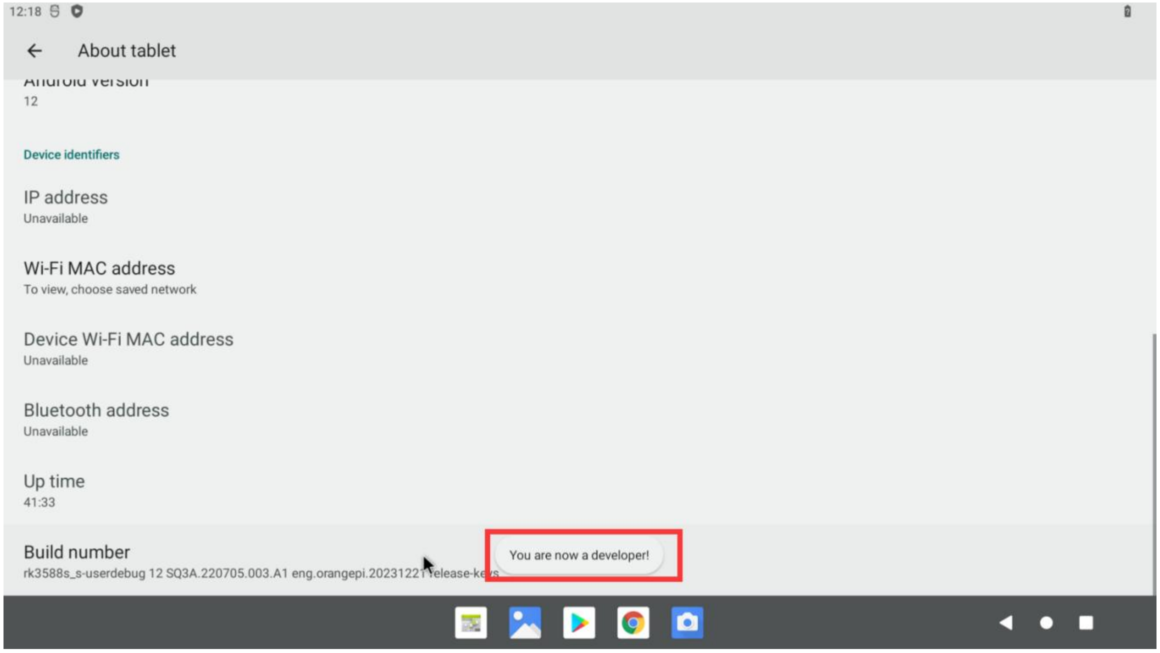
1) First open Settings



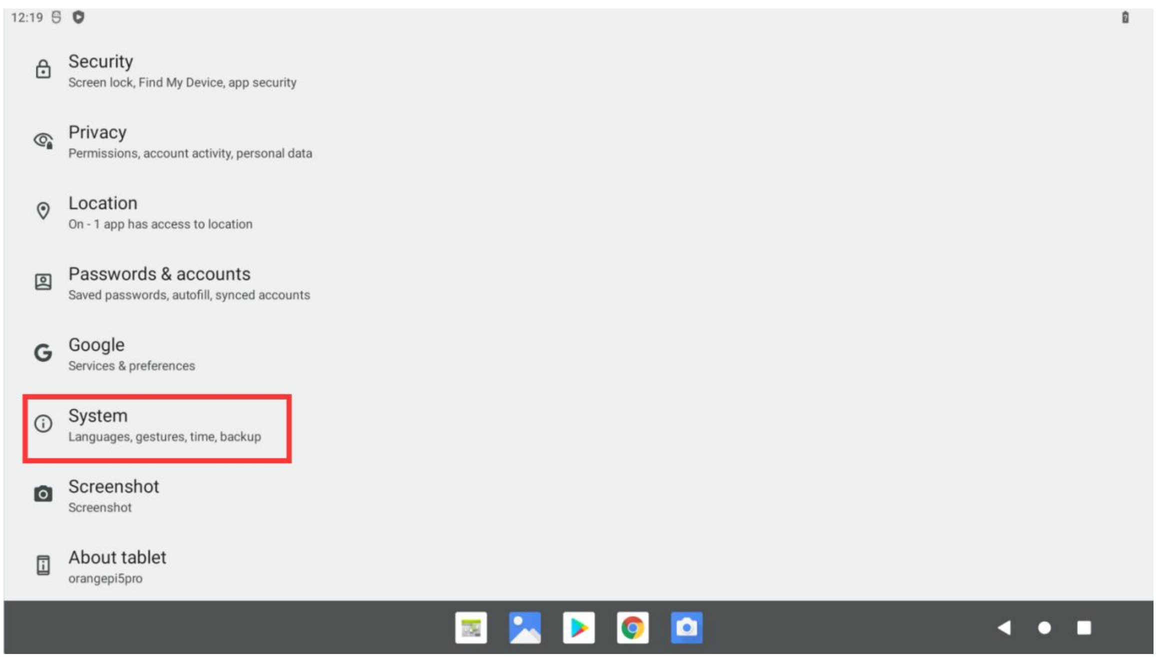
2) Then select **About tablet**



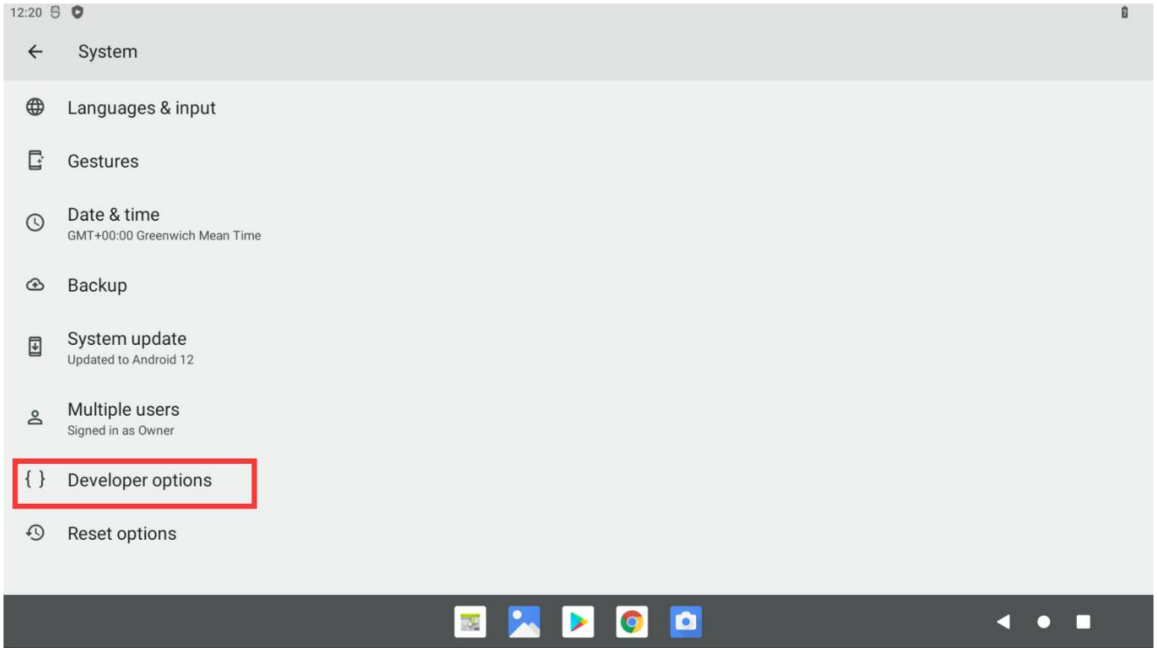
3) Then use the mouse to click the **Build number** menu bar multiple times until the prompt "**You are now a developer!**" appears.



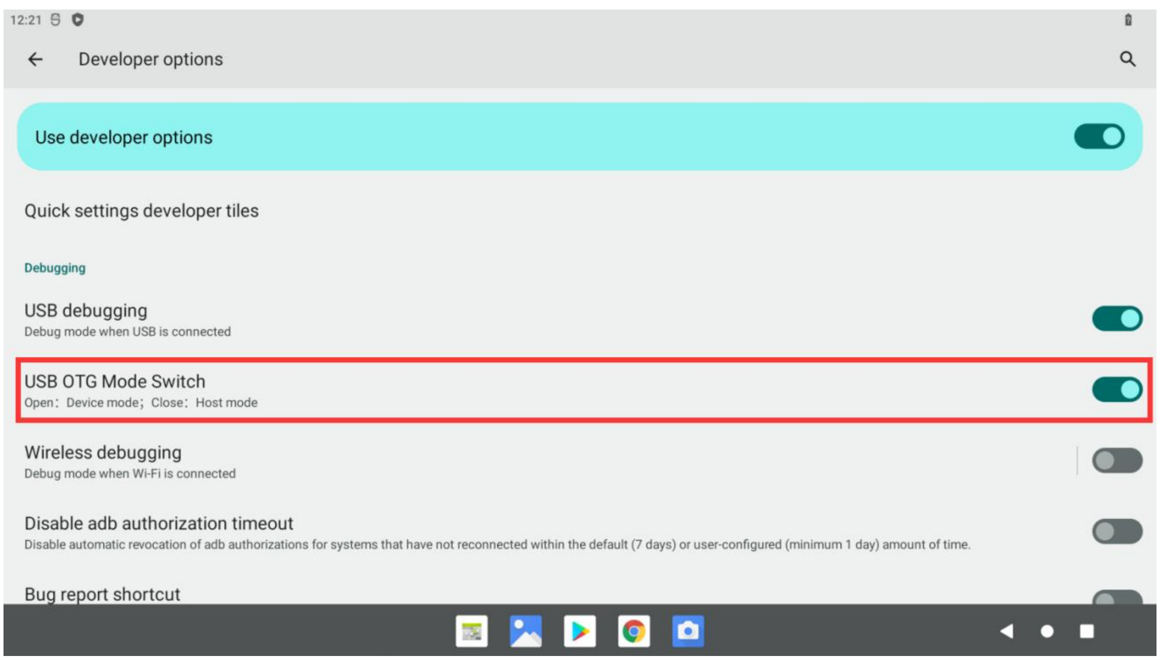
4) Then click to return to the previous menu and select **System**



5) Then select **Developer options**



6) Finally, find the **USB OTG Mode Switch**, **turn it on to switch to Device mode, and turn it off to switch to Host mode.**



7. 9. 2. Use a data cable to connect to adb debugging

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

2) Then refer to the **USB OTG mode switching method** to switch USB OTG to device mode

3) Then use a USB2.0 male-to-male data cable to connect the development board to the USB port of the computer (please also use a TypeC power supply to power the development board)

4) Install adb tool on Ubuntu PC

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

5) Use the following command to view the identified ADB devices

```
test@test:~$ adb devices
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

6) Then you can log in to the Android system through adb shell on the Ubuntu PC

```
test@test:~$ adb shell
console:/ $
```

7) Execute the command to remount the Android system

```
test@test:~$ adb root
test@test:~$ adb remount
```

8) Then you can transfer files to the Android system

```
test@test:~$ adb push example.txt /system/
```

7.9.3. Using adb debugging with network connection

Using network adb does not require a USB Type C interface data cable to connect the computer and the development board. Instead, communication is done over the network. So first make sure that the wired or wireless network of the development board is connected, and then get the IP address of the development board, which will be used later.

1) Ensure that the Android system's **service.adb.tcp.port** is set to port 5555

```
console:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the network adb port number

```
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 to the network adb on the 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 the Ubuntu PC

```
test@test:~$ adb shell
```

```
console:/ #
```

8. How to compile Android 13 source code

8. 1. Download the source code of Android 13

1) First download the Android 13 source code volume compression package from Baidu Cloud or Google Cloud

2) After downloading the compressed package of Android 13 source code, please check whether the MD5 checksum is correct. If not, please download the source code again.

```
test@test:~$ md5sum -c md5sum
Android_13.tar.gz00:    Confirmed
Android_13.tar.gz01:    Confirmed
Android_13.tar.gz02:    Confirmed
Android_13.tar.gz03:    Confirmed
Android_13.tar.gz04:    Confirmed
Android_13.tar.gz05:    Confirmed
Android_13.tar.gz06:    Confirmed
Android_13.tar.gz07:    Confirmed
Android_13.tar.gz08:    Confirmed
```

3) Then you need to merge multiple compressed files into one and then decompress them

```
test@test:~$ cat Android_13.tar.gz0* | tar -xvzf -
```

8. 2. Compile the source code of Android 13

1) First install the software package required to compile the Android13 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 a partition image in the rockdev directory
f. **-u**: Packaging generates a complete image that can be started
g. **-b**: Specify the development board model

3) Compile uboot, kernel, android and package them into a complete image that can be booted

```
test@test:~$ cd Android_13
test@test:~/ Android_13$ ./make.sh -FMu -b orangepi5max --nvme --gapps
```

4) After the 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-rk3588_t** directory. **update.img** is the TF card boot image, and **update_spi_nvme.img** is the NVME SSD boot image.

```
test@test:~/Android_13$ cd rockdev/Image-rk3588_t
test@test:~/Android_13/rockdev/Image-rk3588_t$ ls update*
update.img update_spi_nvme.img
```

9. OpenWRT System Usage Instructions

9. 1. OpenWRT edition

OpenWRT edition	Kernel version
v22.03.4	Linux5.10.110

9. 2. OpenWRT Adaptation situation

Function	OpenWRT
USB2.0	OK
USB3.0	OK
3pin Debug UART	OK
TF card startup	OK
2.5G PCIe Network port X2	OK
1000M Network port	OK
Network port status light	OK
LED light	OK
RTL8821CU USB network card	OK
RTL8723BU USB network card	OK
FAN Fan interface	OK
eMMC extension interface	OK

9. 3. The first boot to expand rootfs

1) When starting the OpenWRT system for the first time, the **resize-rootfs.sh** script will be executed to expand rootfs, and it will automatically restart after the expansion is completed

2) After logging into the system, you can use the **df -h** command to check the size of rootfs. If it matches the actual capacity of the storage device (TF card, eMMC, or NVME

SSD), it indicates 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. Method of logging into the system

9. 4. 1. Login via serial port

1) Firstly, to debug the use of the serial port, you can refer to the **chapter on debugging the usage of the serial port**

2) The OpenWrt system will automatically log in as the **root** user by default, and the display interface is as follows

```
BusyBox v1.33.1 (2021-10-24 09:01:35 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 21.02.1, r16325-88151b8303
 -----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@OpenWrt:/#
```

9. 4. 2. Login to the system via SSH

> **Please note that in the OpenWrt system of Orange Pi 5 Max, the network port is configured as a LAN port by default.**

1) Firstly, connect the LAN1 port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address

through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

3) If the computer is installed with Ubuntu system, you can execute the following command to log in to the system through SSH. By default, you can log in directly without a password

```
test@ubuntu:~$ ssh root@192.168.2.1
```

4) The display after successfully logging into the system is shown in the following figure

5) If the computer is installed with Windows system, you can refer to the method introduced in the section of **SSH remote login development board under Windows to log in**

9. 4. 3. Login to LuCI Management Interface

Please note that in the OpenWrt system of Orange Pi 5 Max, the network port is configured as a LAN port by default.

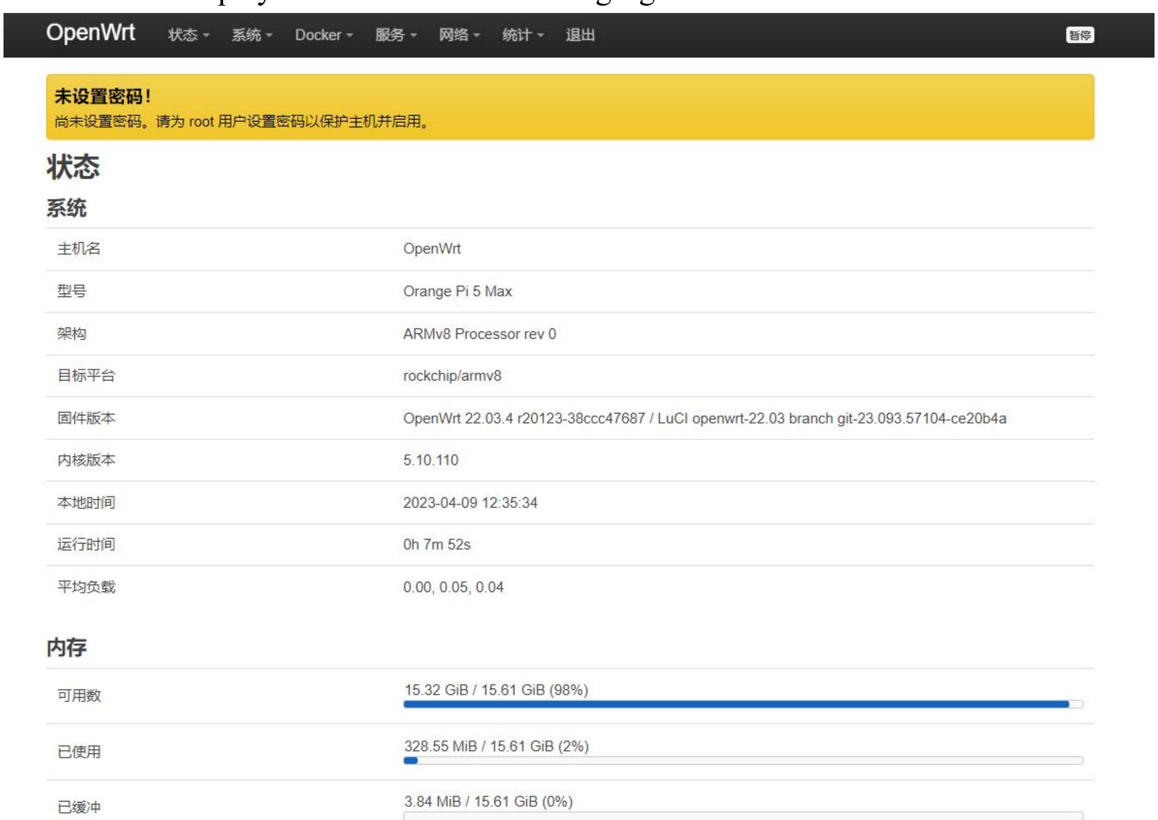
1) Firstly, connect the LAN port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address

through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

3) You can log in to the LuCI interface by entering the IP address **192.168.2.1** in the browser on your computer

4) **OpenWrt 系统默认是没有设置密码的**，所以 **The OpenWrt system does not have a password set by default**, so simply click the **login** button. After successful login, the interface will display as shown in the following figure



9. 4. 4. Log in to the terminal through the LuCI management interface

Please note that in the OpenWrt system of Orange Pi 5 Max, the network port is configured as a LAN port by default.

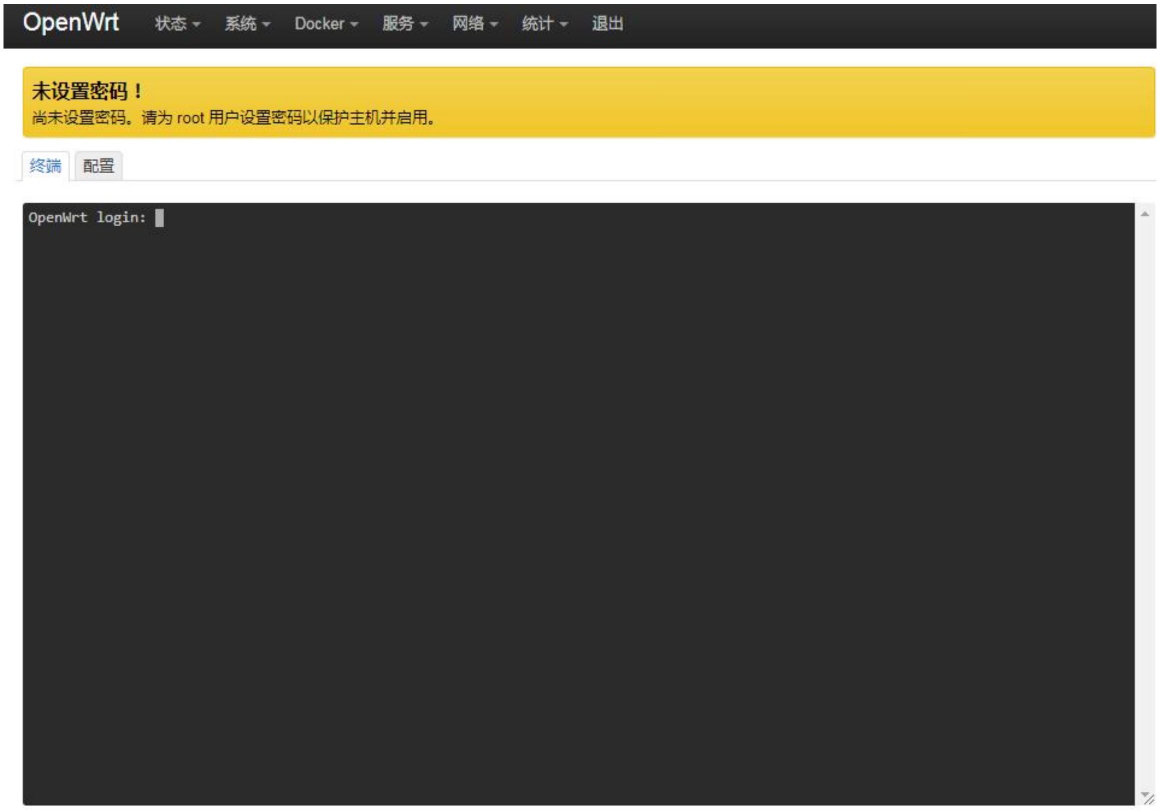
1) Firstly, connect the LAN1 port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

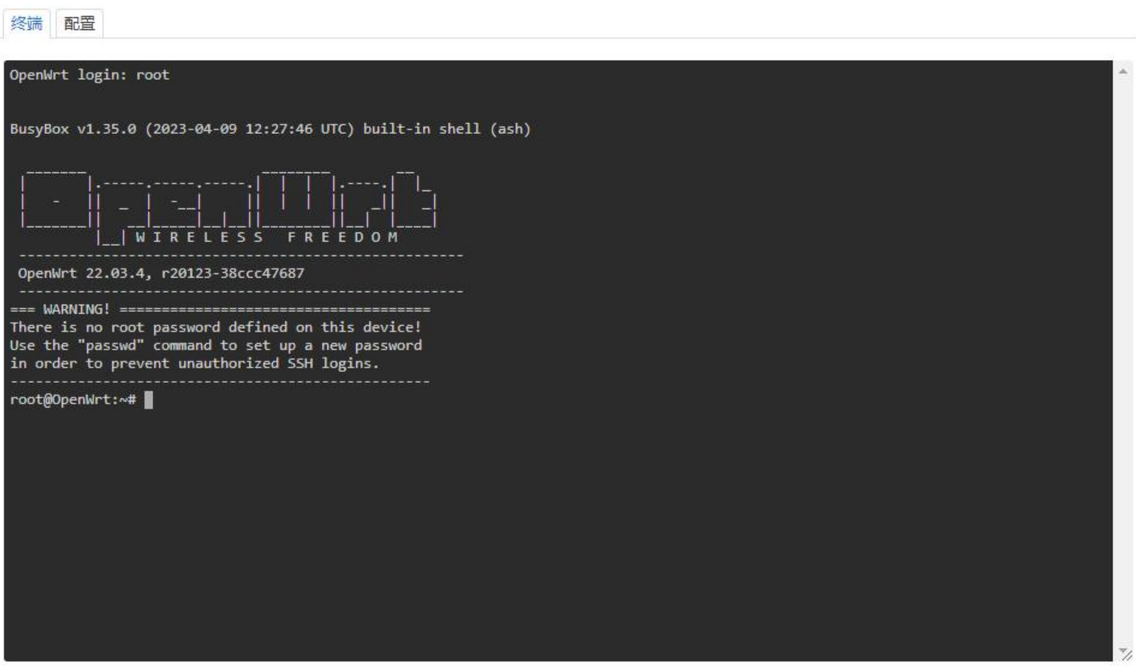
3) You can log in to the LuCI interface by entering the IP address **192.168.2.1** in the browser on your computer

4) Select **"Terminal"** in the **"Services"** column of the navigation bar and click to enter

5) At this point, the terminal interface is shown in the following figure



6) Enter the username root to log in



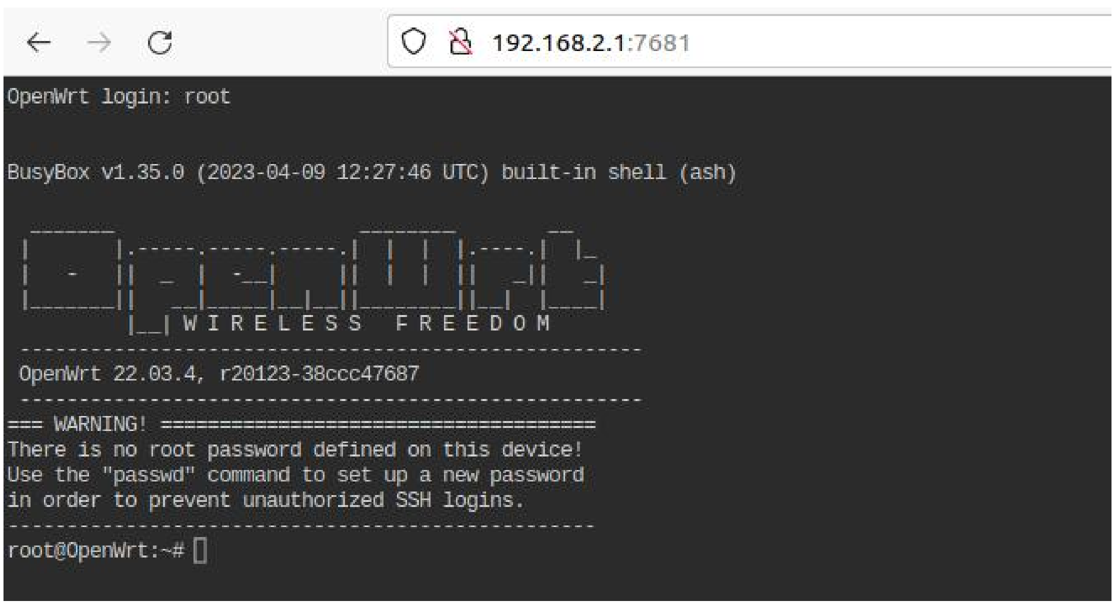
9. 4. 5. Login to the terminal using IP address and port number

Please note that in the OpenWrt system of Orange Pi 5 Max, the network port is configured as a LAN port by default.

1) Firstly, connect the LAN port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

3) Then enter **192.168.2.1:7681** in the browser to log in to the OpenWRT terminal



9. 5. Method of modifying LAN port IP address through 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 configuration files. 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**, and 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** option

```
root@OpenWrt:~# uci set network.lan.ipaddr='192.168.100.1'
```

4) Then enter the following command to complete the submission, which is written to the configuration file

```
root@OpenWrt:~# uci commit
```

If the IP address in red font matches the one to be set, it indicates that the modification was 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 Ubuntu. Please refer to the **official documentation** for instructions on how to use Ubuntu 过 ubus

```
root@OpenWrt:~# ubus call network restart
```

6) At this point, entering the command shows that the IP address 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. Method for changing root password

9. 6. 1. Modify via Command Line

1) Firstly, enter passwd root in the system command line, and the following prompt message will appear. At this time, you can enter the password you want to set and press Enter to confirm

```
root@OpenWrt:/# passwd root
Enter new UNIX password:
```

2) Next, you will be prompted to re-enter the password. At this point, enter the password again to confirm and press Enter

```
Retype password:
```

3) The successfully modified display is as follows

```
passwd: password for root changed by root
```

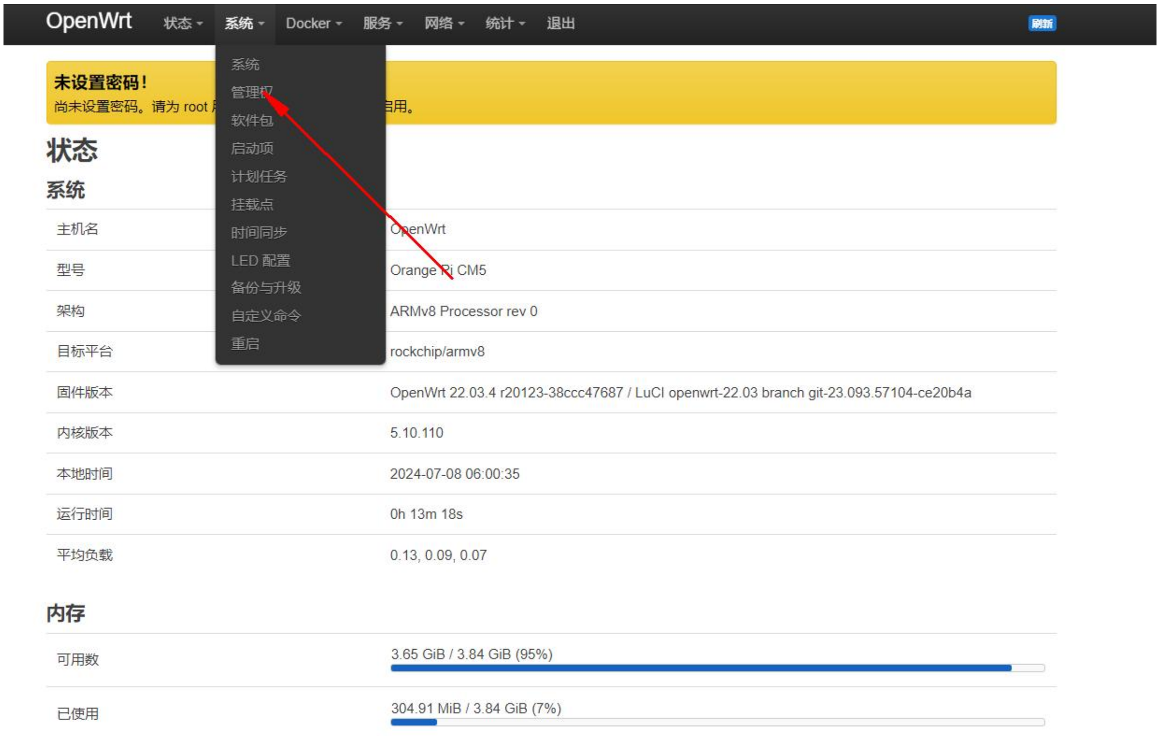
9. 6. 2. Modify through LuCI management interface

1) Firstly, refer to **the login LuCI management interface** to enter the OpenWRT management interface

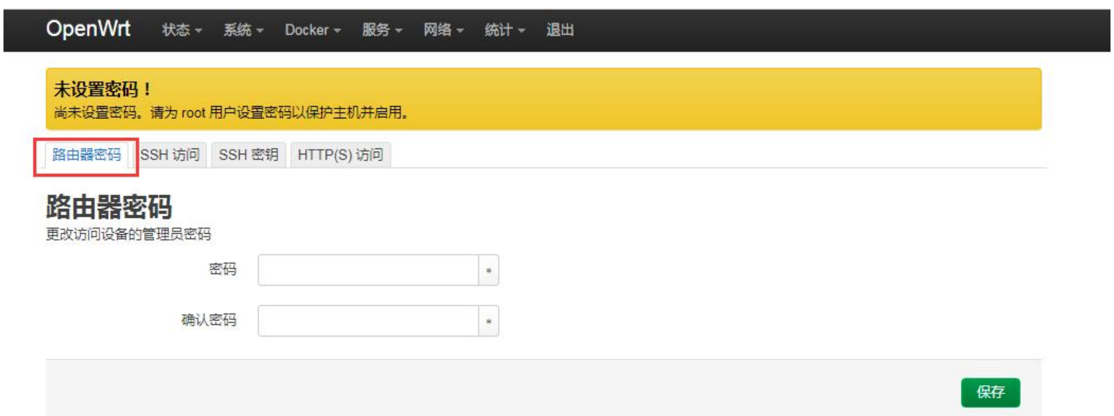
2) Then follow the steps below to change the password

a. Find the **"System"** option in the navigation bar and click on it

b. In the vertical bar options below the system, select **"Management Rights"** and click



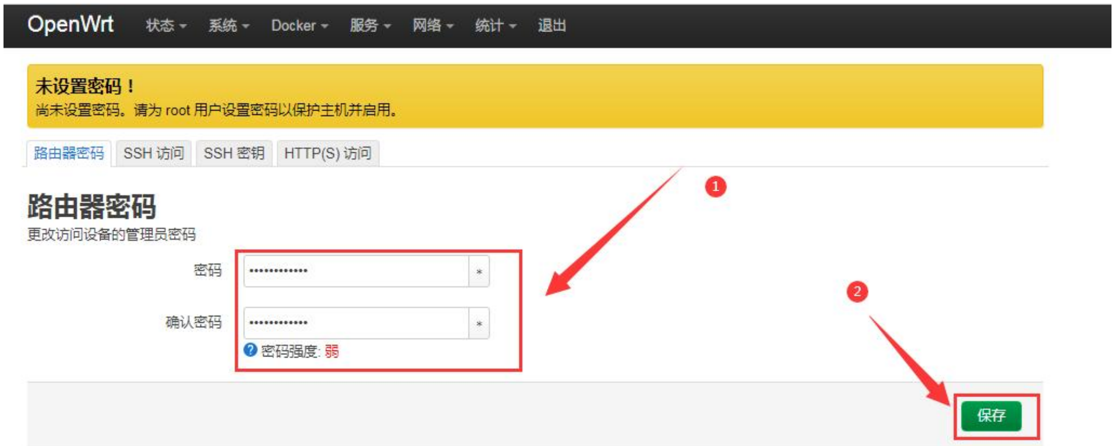
c. Select the **'Router Password'** option on the Tab page



3) Change and save router password

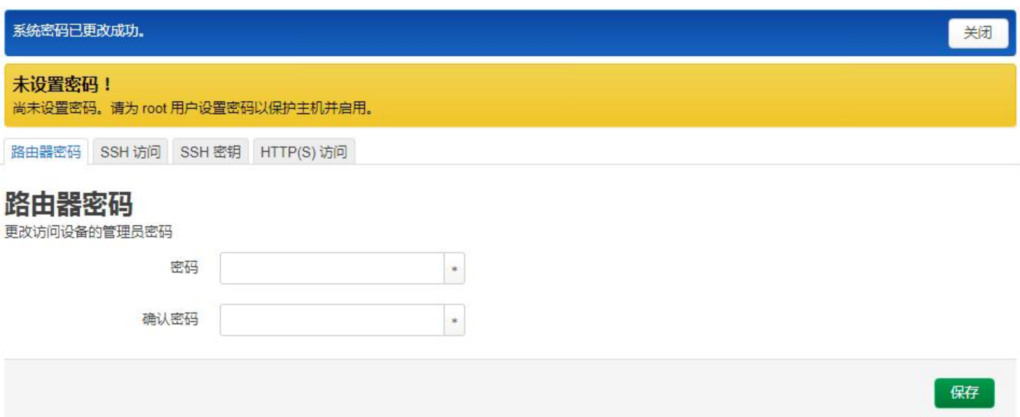
a. Enter the password you have set in the **"Password"** and **"Confirm Password"** dialog boxes (if unsure if the password is entered correctly, click the **"*"** icon behind the dialog box to display the input characters)

b. Click **'Save'** to save the newly modified password



Note: In the "Password" and "Confirm Password" dialog boxes, the password entered twice needs to be consistent.。

4) After the password is successfully changed, a pop-up message saying **"System password has been changed successfully"** will appear. At this time, logging into OpenWRT requires a password to log in



9. 7. USB interface testing

9. 7. 1. Mounting USB storage devices at the command line

1) Firstly, insert the USB drive into the USB interface of the Orange Pi development board

2) If you can see the output of sdX by executing the following command, it indicates that the USB drive recognition is successful

```
root@OpenWrt:~# cat /proc/partitions | grep "sd*"
major minor    #blocks    name
    8              0      15126528 sda
```

3) You can use the mount command to mount the USB drive to /mnt, and then you can view the files on the USB drive

```
root@OpenWrt:~# mount /dev/sda /mnt/
root@OpenWrt:~# ls /mnt/
test.txt
```

4) After mounting, you can use the df -h command to view the capacity usage and mounting points of the USB flash drive

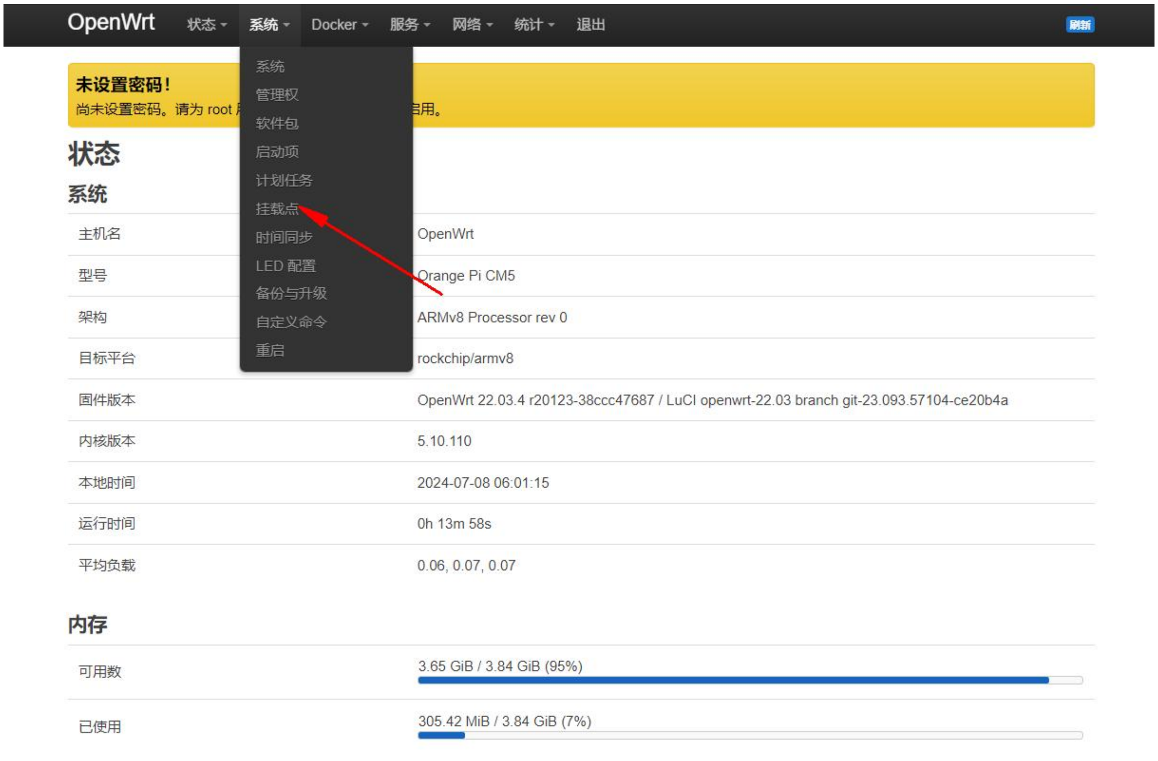
```
root@OpenWrt:~# df -h | grep "sd"
/dev/sda                14.4G       187.2M         14.2G      1% /mnt
```

9. 7. 2. Mounting USB storage devices on the LuCI management interface

1) Firstly, connect the USB flash drive (or other storage device) to the development board via USB2.0

2) Then follow the **login LuCI management interface** to enter the LuCI management world

3) Then, in the LuCI management interface, click on "System ->Mount Point" to enter the configuration interface of the mount point



4) Then follow the steps below to add a mounting point

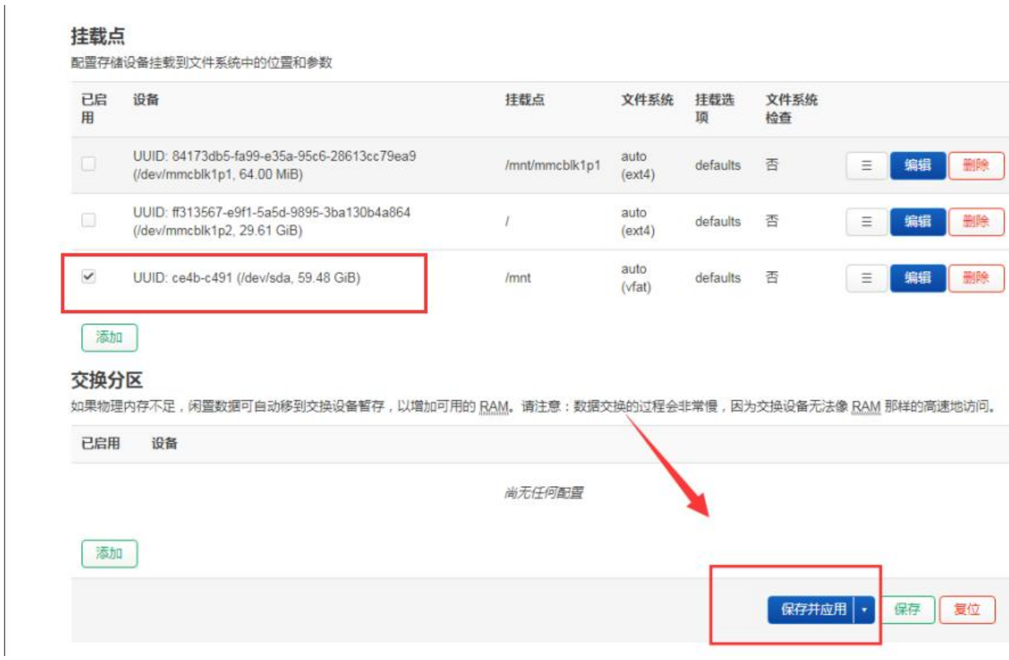
a. Find **'Mount Point'** below the **global settings interface** for mount points

b. Below the **mounting point**, select the "Add" button and click to enter

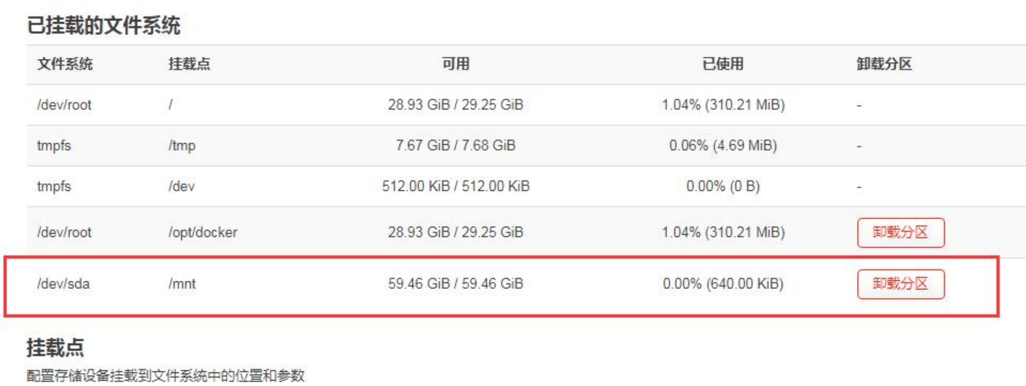
c. Next, a pop-up window will appear below

d. Then you can start mounting the storage device
   a) Check **'Enabled'**
   b) Select the actual connected device /dev/sda in the UUID column of the general settings (choose according to your own device)
   c) 在挂 Select **"Custom"** in the mount point column and fill in the target directory to be mounted to. Taking the **/mnt** directory as an example, fill in and press **Enter** to confirm
   d) Then click the **"Save"** button in the bottom right corner

5) Then you will return to the mount point global settings page and click **"Save and Apply"** in the bottom left corner of the page to make the mount point effective



6) After saving, you can see in the **"Mounted File System"** that the storage device has been successfully mounted



9. 8. USB Wireless Network Card Test

The currently tested USB wireless network cards that can be used are shown below. For other models of USB wireless network cards, please test them yourself. If they cannot be used, you need to port the corresponding USB wireless network card driver.

Serial number	model	

1	RTL8723BU Support 2.4G WIFI+BT4.0	
2	RTL8821CU Support 2.4G +5G WIFI Support BT 4.2	
3	RTL8811 Support 2.4G +5G WIFI	

9.8.1. Method of using a USB wireless network card to connect to a WIFI hotspot

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power on the development board.

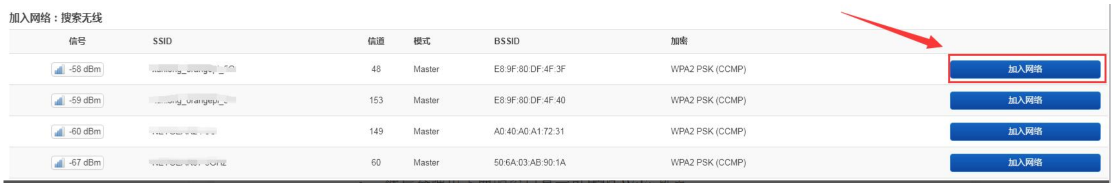
2）After the system startup is complete, click on **Network ->Wireless** to enter the wireless WiFi configuration interface.

3）The default wireless configuration of OpenWRT system is **Master** mode. For the convenience of the next operation, we will remove the default wireless connection.

4）Then click on the bottom right corner of the page to **save** and make the configuration effective.

5）Then click the **scan** button to scan the surrounding WiFi hotspots.

6）Then a window will pop up displaying available WiFi hotspots. Click the **Join Network** button to the right of the desired WiFi hotspot to connect.

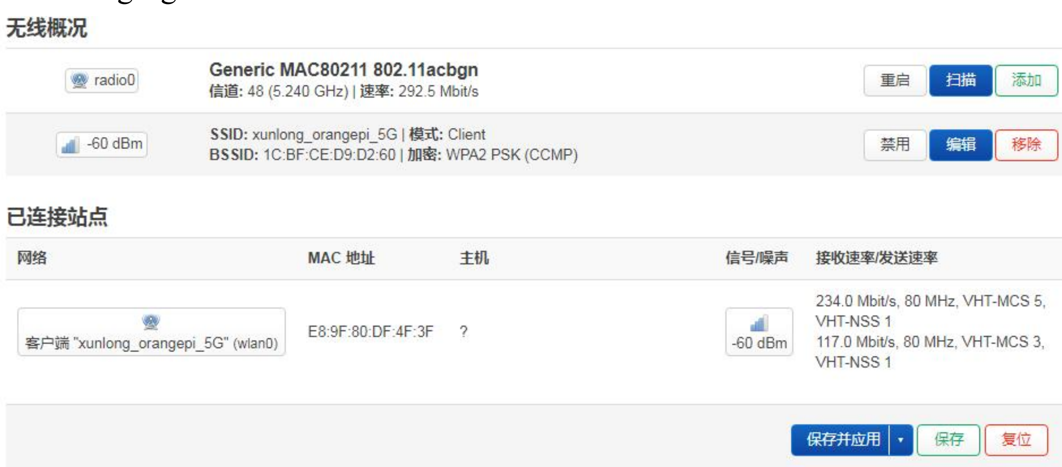


7）Then a interface will pop up to connect to the WiFi hotspot. We will enter the hotspot password at the location shown in the figure below, and then click the **submit** button.

8）Then the following interface will pop up, click the **save** button in the bottom right corner.

9）Finally, you will return to the main interface of wireless configuration, click **save and apply**, and wait for the configuration to be applied.

10）After successfully connecting to the WiFi hotspot, the interface displays as shown in the following figure.



9. 8. 2. Method for creating a WIFI hotspot using a USB wireless network card

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power on the development board.

2）The system startup is complete, click on **Network ->Wireless** to enter the wireless WiFi configuration interface.

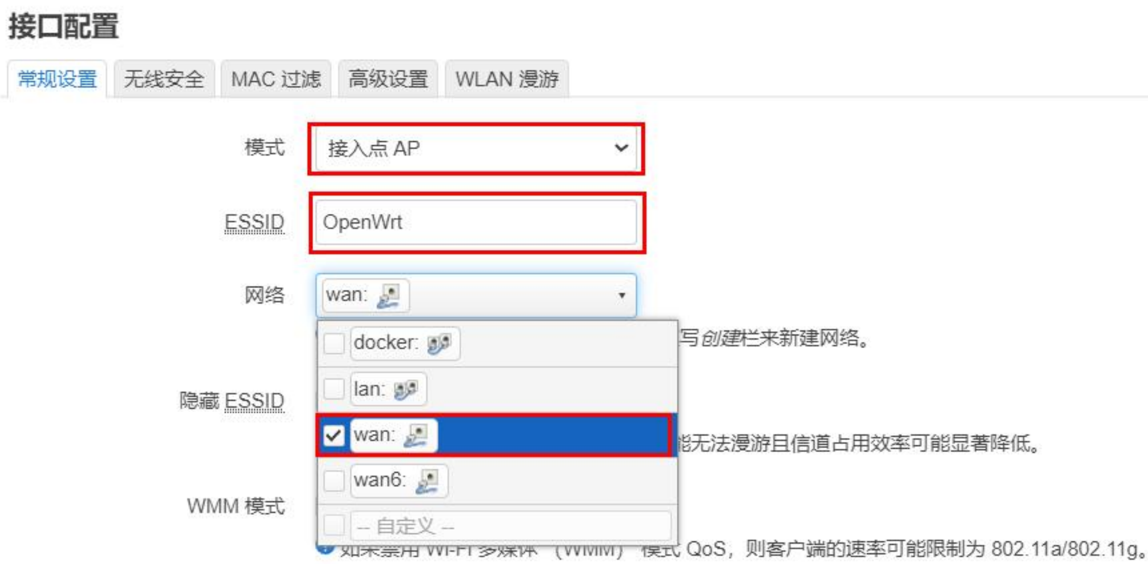
3）The default wireless configuration of OpenWRT system is **Master** mode. For the convenience of the next operation, we will remove the default wireless connection.

4）Then click on the bottom right corner of the page to **save** and make the configuration effective.

5）Then click the **add** button on the right.

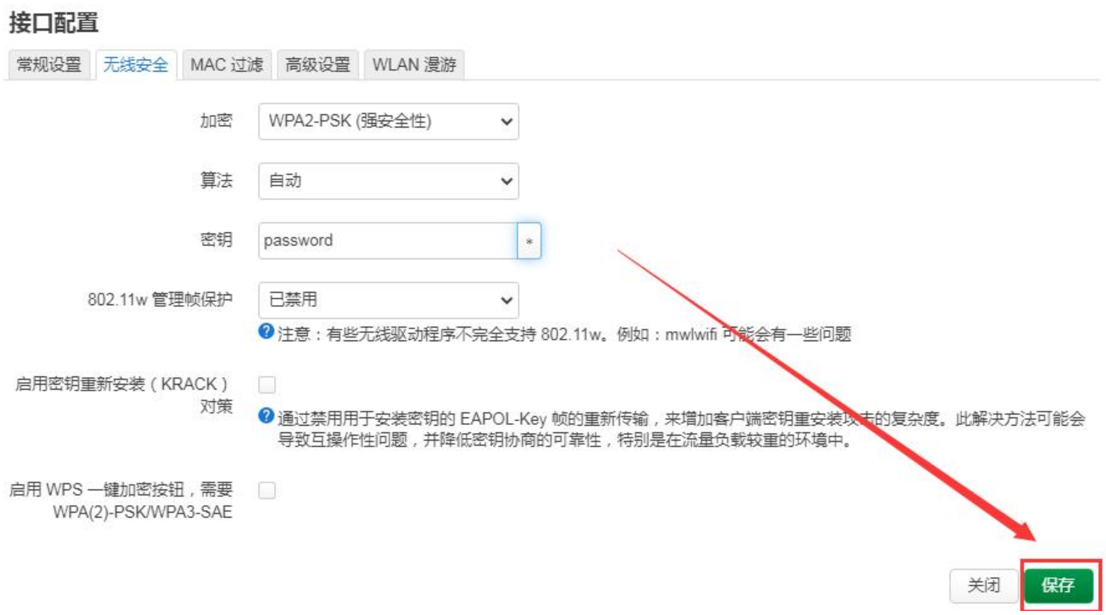
6）In the pop-up tab **device configuration**, we set the parameters as shown in the following figure.

7）Then in **Interface Configuration ->General Settings**, set the mode to **Access Point AP**, **ESSID** (Wireless Network Name) to **OpenWrt**, and network to **wan**



8）Then in **Interface Configuration ->Wireless Security**, select **WPA2-PSK** as the encryption algorithm; Set the key (wireless password) to **password**

9）After completing the above settings, click on the bottom right corner of the page to **save**, and then you will exit the tab



10）Then click on the bottom right corner of the page to **save and apply**, and wait for the configuration to be applied.

11）The display interface for successfully creating a hotspot is shown in the following figure

12）Then use your phone or computer to search for the corresponding WiFi SSID for connection. After successful connection, as shown in the following figure

9. 9. Installing software packages through the command line

9. 9. 1. Installing through OPkg on the terminal

1) Update the list of available software packages

```
root@OpenWrt:/# opkg update
```

2) Get 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 software

```
root@OpenWrt:/# opkg remove <Package Name>
```

9. 10. OpenWRT management interface installation software package

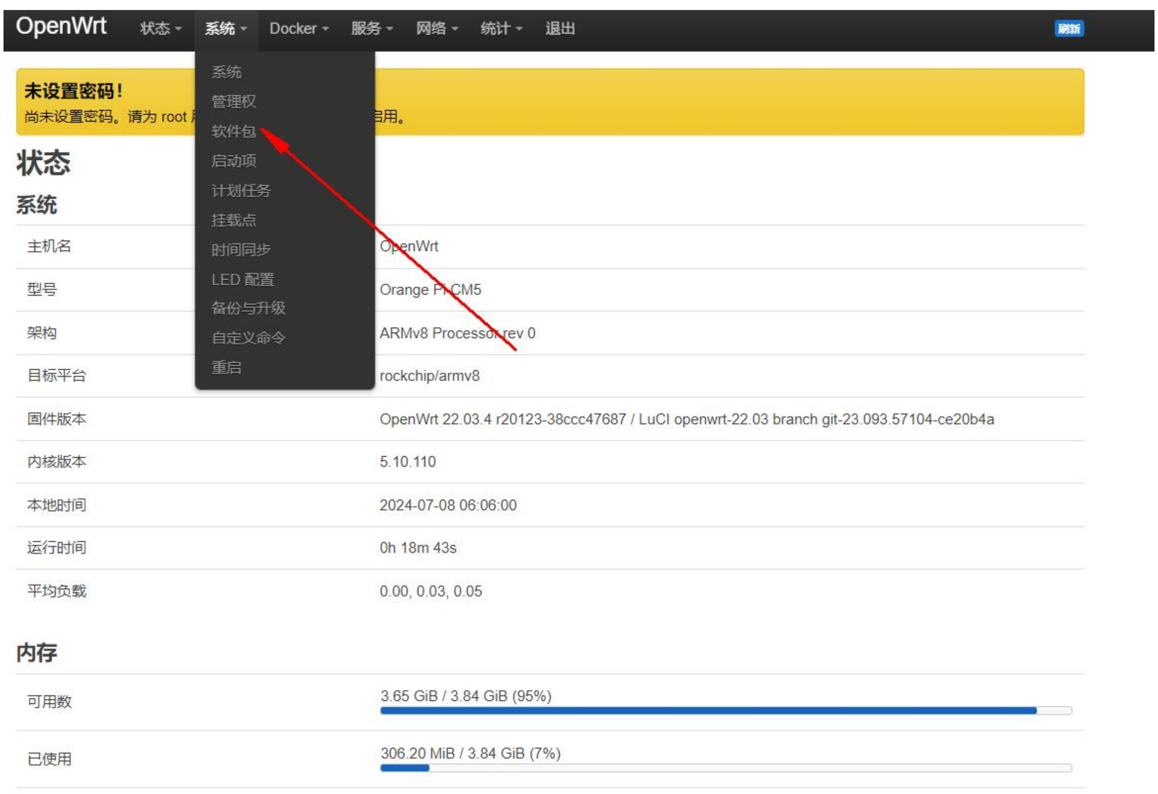
If you need to add software packages, you can install them through the OpenWRT management interface.

9. 10. 1. View the list of available software packages in 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 bar 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 following

figure, to obtain the list of available software

a. In the **"Operation"** option of the software package, click **"Update List"** to obtain the list of available software packages

b. On the tab page, click **"Available"** to view the currently available software packages

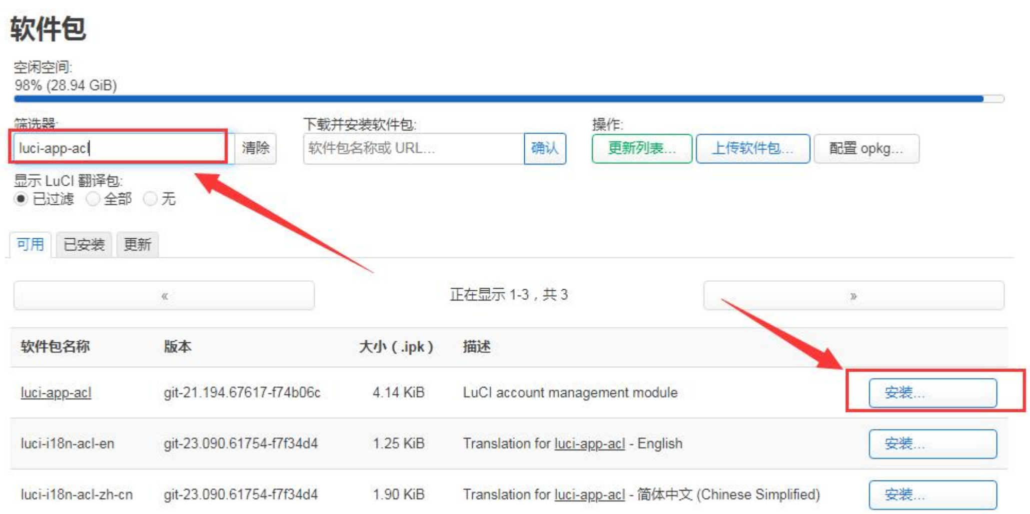
c. View the current number of available software packages

9. 10. 2. Example of Installing Software Packages

1) Taking the installation of the software package "**luci-app-acl**" as an example

a. In the package management interface of OpenWRT, click on 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**" package, and then click the "**Install**" button



c. Then the following pop-up window will appear, click "**Install**" to proceed

d. Then wait for the installation to complete

e. The display after installation is as follows

2) Check if the software package has been successfully installed

   a. In the package management interface of OpenWRT, click on the filter dialog box and enter "**luci-app-acl**"

   b. Select and click '**Available**' on the tab page

   c. The '**luci-app-acl**' package will be displayed in the package list and updated to '**installed**' status

9. 10. 3. Example of Removing Software Packages

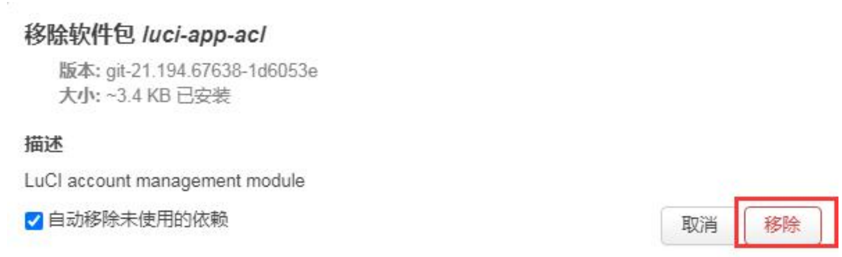
1) Taking the removal of the software package '**luci-app-acl**' as an example

   a. In the package management interface of OpenWRT, click on the filter dialog box and enter "**luci-app-acl**"

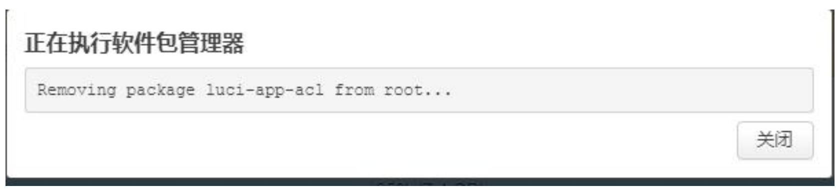
   b. Select '**Installed**' on the tab page to display a list of installed software packages

   c. Click '**Remove**' on the right to remove the corresponding software package

d. Then a pop-up window will appear below, click '**Remove**' to proceed



e. After successful removal, the display interface is as follows



2) Check if the software package has been successfully removed

a. In the package management interface of OpenWRT, click on the filter dialog box and enter "**luci-app-acl**"

b. Select and click '**Installed**' on the tab page

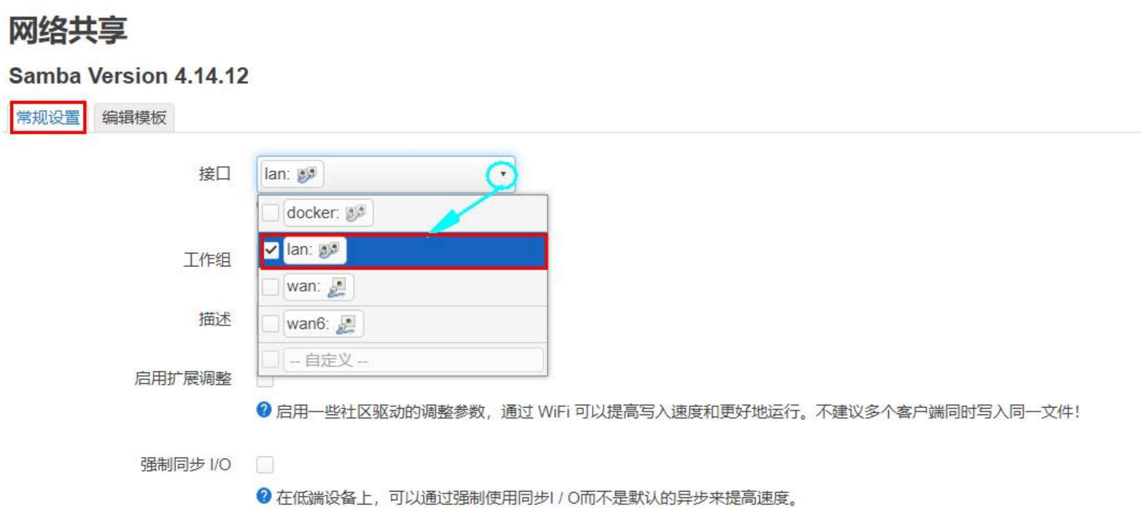
c. The '**luci-app-acl**' package will not be displayed in the package list, and the '**luci-app-acl**' package has been successfully removed

9. 11. Using Samba Network Sharing

There are two main software options for implementing OpenWRT LAN file sharing: Samba and NFS. Samba system has good compatibility, while NFS performs better. For users who need to use Windows devices, it is recommended to choose Samba.

1) Enter the Samba network share management page
   a. Find the "**Services**" 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 Samba service needs to listen on
   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 it through the "lan port", set it to "**Lan**"



3) Set the shared directory for network sharing

a. Click "Add" in the "**Shared Directory**" section of the "**General Settings**" for network sharing to share the directory address

b. Enter the name of the shared folder as '**mmt**' under the name

c. Under the path of the shared directory, select the location of the shared directory "/"

d. Check '**browseable**' and '**allow anonymous users to run**'

e. Click '**Save and Apply**' to save the configuration

4) window10 starts network discovery and sharing

Note: To access Samba on the Windows 10 system, it is necessary to first confirm whether Windows 10 has started network discovery and sharing. If it has not been started, the following settings should be made first.

a. Enable access to Samba v1/v2

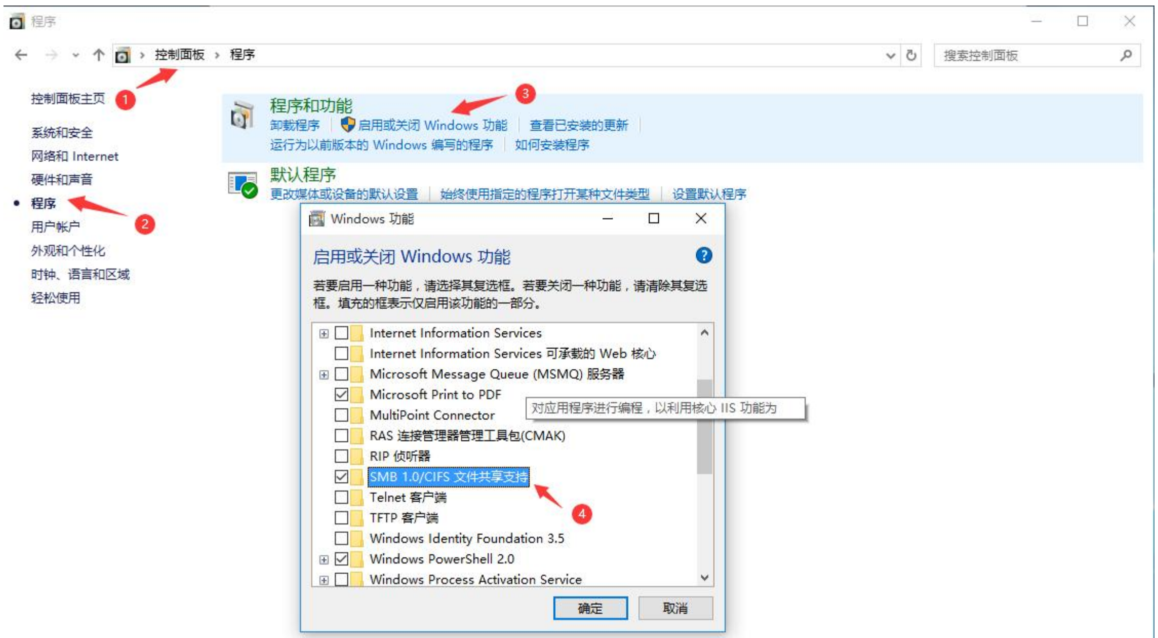
a) Enter the Control Panel of windows10

b) Click on "Programs" in the left navigation bar of the control panel

c) Select 'Enable or Disable Windows Features' 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. Open Windows10 Network Discovery

a) Enter the Control Panel of windows10

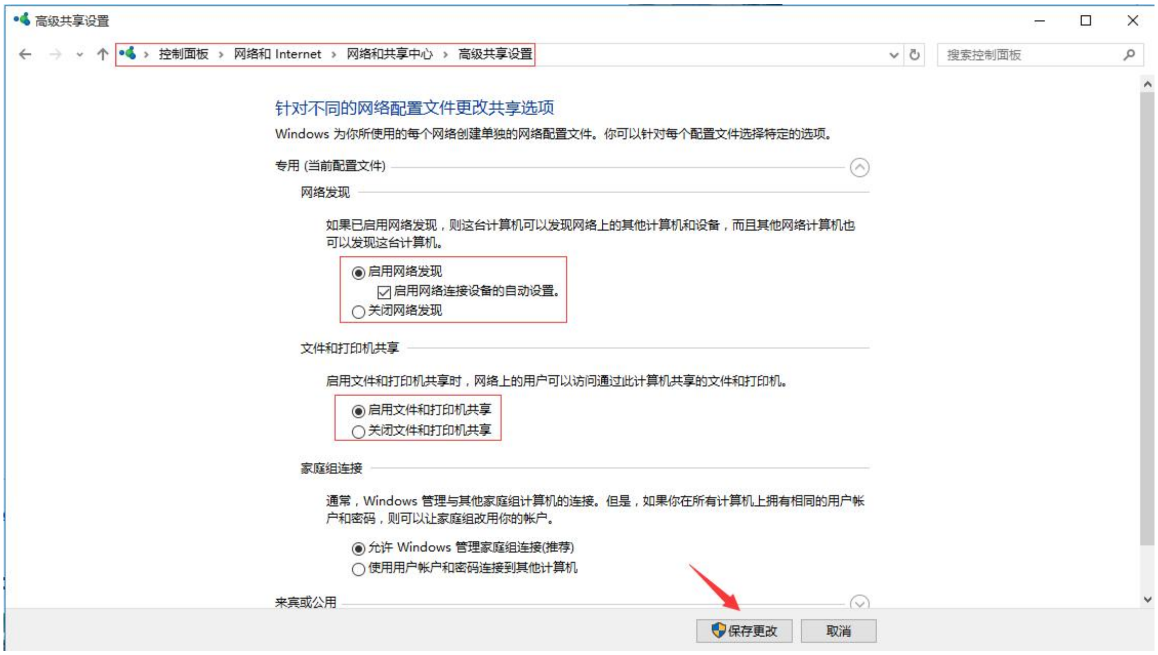
b) Select "Network and Internet" in the control panel

c) Then open the "Network and Sharing Center"

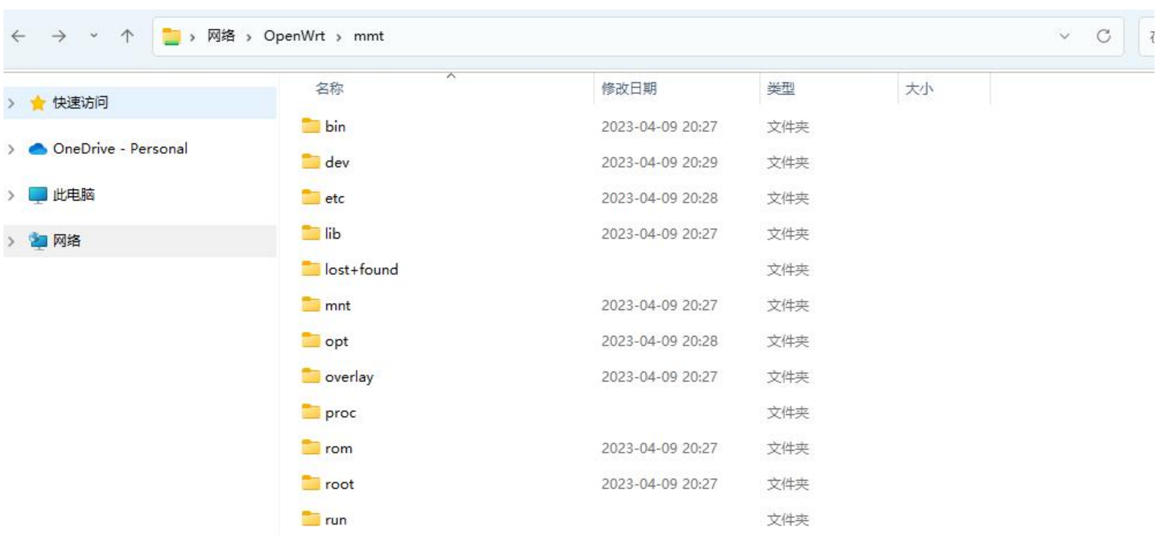
d) Click on 'Advanced Sharing Settings'

e) Open '**Enable Network Discovery**' and '**Enable File and Printer Sharing**'

f) Click 'Save Changes' to save the network discovery configuration for Windows10



5) After setting up, enter \\OpenWrt in the address bar of the resource manager to access the shared directory. The username is root and the password is the password set by the development board host

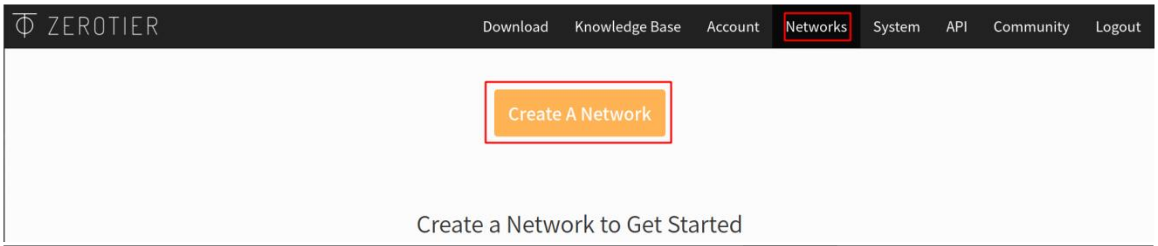


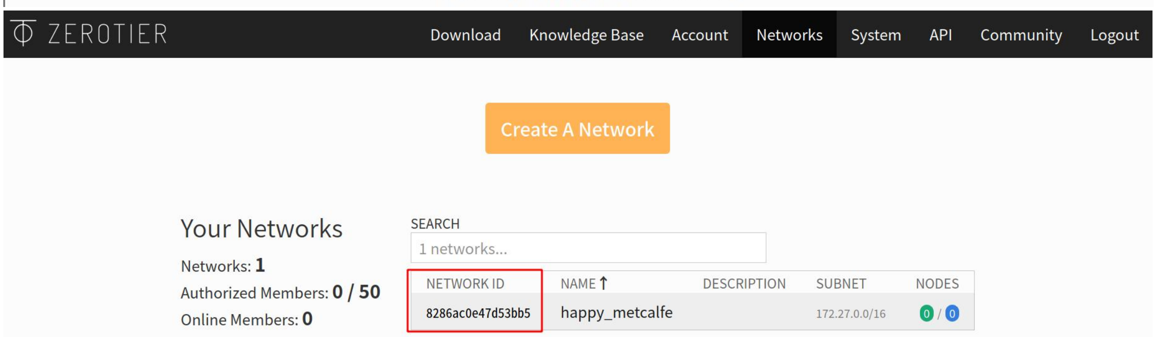
9. 12. zerotier User Manual

The OpenWRT system has pre installed the zerotier client. After creating a

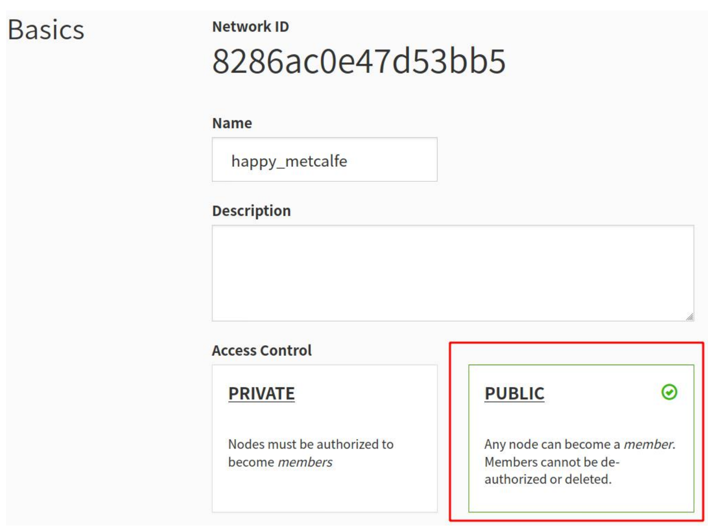
virtual LAN on the zerotier official website, the client can directly join it through the Network ID. The specific operation is shown below.

1) Log in to the zerotier official website https://my.zerotier.com/network After registering and logging in, click Network->Create A Network to create a virtual LAN

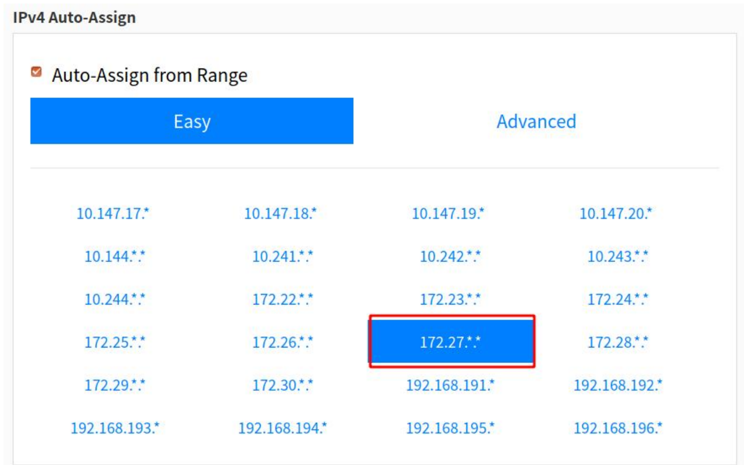




2) Click to enter the network console page, where you can set the privacy option to public, so that network nodes that join do not need to be verified



3) Below, the address will be automatically assigned. Here, you can choose your own network segment, and the selected one is 172.27.*.*



4) Enter the following command on the OpenWRT terminal to join the virtual LAN created above, **where 8286ac0e47d53bb5 is the Network ID of the virtual LAN created above**

```
root@OpenWrt:/# zerotier-one -d                        #Start the zerotier client
root@OpenWrt:/# zerotier-cli join 8286ac0e47d53bb5     #Join the network
```

5) By entering ifconfig on the terminal, it can be seen that there is already a newly added **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 (using Ubuntu 18.04 as an example), execute the following command to install, and restart the computer after installation is complete

```
test@ubuntu:~$ curl -s https://install.zerotier.com | sudo bash
```

7) After restarting, join the virtual LAN based on the Network ID, and you can also see

that the IP address assigned by zerotier has been obtained. At this time, the Ubuntu PC and OrangePi R1 Plus LTS are in the same LAN, and they can communicate freely

```
test@ubuntu:~$ sudo zerotier-cli join 8286ac0e47d53bb5
test@ubuntu:~$ ifconfig
ztks54inm2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2800
        inet 172.27.47.214  netmask 255.255.0.0  broadcast 172.27.255.255
        inet6 fe80::5ce1:85ff:fe2b:6918  prefixlen 64  scopeid 0x20<link>
        ether f6:fd:87:68:12:cf  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 46  bytes 10006 (10.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

8) Test whether 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 other commonly used commands

```
root@OpenWrt:/# zerotier-one -d                          #Start the zerotier client
root@OpenWrt:/# zerotier-cli status                      #Obtain address and service status
root@OpenWrt:/# zerotier-cli join # Network ID           #Join the network
root@OpenWrt:/# zerotier-cli leave # Network ID          #Leave the internet
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 openwrt-22.03 branch code

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install -y git
test@test:~$ git clone https://github.com/orangepi-xunlong/openwrt.git -b openwrt-22.03
```

2) After downloading the OpenWRT code, the following files and folders will be included

```
test@test:~/openwrt$ ls
BSDmakefile   Config.in   include   Makefile   README.md   scripts   toolchain
Config   feeds.conf.default   LICENSE   package   rules.mk   target   tools
```

10. 2. Compile OpenWRT source code

1) Firstly, install the following dependency packages (currently only tested for compilation on Ubuntu20.04. If compiling on other versions of the system, please install the dependency packages yourself according to the error message)

a. Method 1: The command to install dependency packages using a script is as follows:

```
test@test:~/openwrt$ sudo ./install_dep.sh
```

b. Method 2: Install dependency packages directly using the following command

```
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** download dependency package

```
test@test:~/openwrt$ ./scripts/feeds update -a
test@test:~/openwrt$ ./scripts/feeds install -a
```

3) Then choose to use the configuration file of OrangePi CM5

```
test@test:~/openwrt$ cp configs/orangepi-cm5-rk3588_defconfig .config
```

4) Then execute the following command to make the configuration effective

```
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, 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-cm5-ext4-sysupgrade.img.gz
├── openwrt-rockchip-armv8-xunlong_orangepi-cm5.manifest
├── openwrt-rockchip-armv8-xunlong_orangepi-cm5-squashfs-sysupgrade.img.gz
├── packages
├── profiles.json
├── sha256sums
└── version.buildinfo

1 directory, 8 files
```

11. Appendix

11. 1. User Manual Update History

Version	Date	Release Notes
v1.0	2024-07-05	initial version
v1.1	2024-07-19	1. OpenWRT system usage instructions 2. The compilation method of OpenWRT source code
v1.2	2024-07-30	3. The usage method of wiringOP hardware PWM

11. 2. Image update history

Date	Release Notes
2024-07-05	Orangepi5max_1.0.0_ubuntu_focal_server_linux5.10.160.7z Orangepi5max_1.0.0_ubuntu_jammy_server_linux5.10.160.7z Orangepi5max_1.0.0_debian_bullseye_server_linux5.10.160.7z Orangepi5max_1.0.0_debian_bookworm_server_linux5.10.160.7z Orangepi5max_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi5max_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi5max_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi5max_1.0.0_debian_bookworm_desktop_xfce_linux5.10.160.7z Orangepi5max_1.0.0_ubuntu_jammy_server_linux6.1.43.7z Orangepi5max_1.0.0_debian_bookworm_server_linux6.1.43.7z Orangepi5max_1.0.0_ubuntu_jammy_desktop_xfce_linux6.1.43.7z Orangepi5max_1.0.0_debian_bookworm_desktop_xfce_linux6.1.43.7z OrangePi5Max_RK3588_Android13_v1.0.0.tar.gz OrangePi5Max_RK3588_Android13_spi-nvme_v1.0.0.tar.gz Opios-droid-aarch64-opi5max-24.07-linux5.10.160.tar.gz Opios-droid-aarch64-opi5max-24.07-linux5.10.160-spi-nvme.tar.gz Opios-arch-aarch64-gnome-opicm5-24.07-linux5.10.160.img.xz * initial version

2024-07-19	openwrt-rockchip-armv8-xunlong_orangepi-5max-ext4-sysupgrade_v1.0.img.gz * Initial version