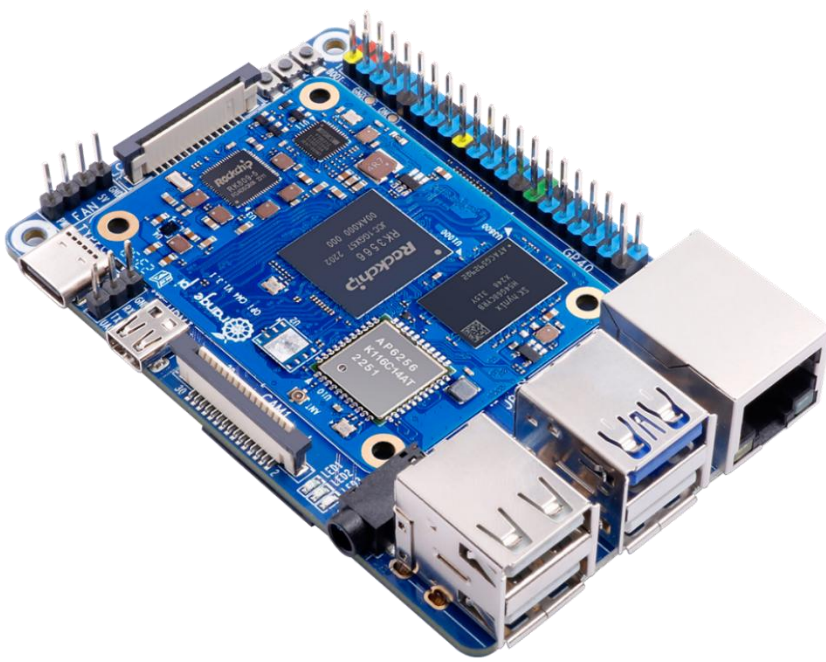
Orange Pi CM4
User Manual



Catalog

1. Basic features of Orange Pi CM4

1. 1. What is Orange Pi CM4

Orange Pi CM4 uses Rockchip RK3566 quad-core 64-bit Cortex-A55 processor, using 22nm process, with a main frequency of up to 1.8GHz, integrated ARM Mali-G52 GPU, embedded high-performance 2D image acceleration module, built-in 0.8 The AI accelerator NPU of Tops computing power can choose 1GB, 2GB, 4GB or 8GB memory, and has up to 4K display processing capability.

Orange Pi CM4 brings out quite a lot of interfaces, including Micro HDMI output, M.2 PCIe2.0x1, Gigabit Ethernet port, USB2.0, USB3.0 interface and 40pin expansion pin header, etc. It can be widely used in high-end tablet, edge computing, artificial intelligence, cloud computing, AR/VR, smart security, smart home and other fields, covering various AIoT industries.

Orange Pi CM4 supports Android11, Ubuntu22.04, Ubuntu20.04, Debian11, Debian12, open source Hongmeng 4.0 Beta1, Orange Pi OS (Arch), Orange Pi OS (OH) based on open source Hongmeng and other operating systems.

1. 2. Usage of Orange Pi CM4

We can use it to achieve:

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

Of course, there are more functions. Relying on a powerful ecosystem and a variety of expansion accessories, Orange Pi can help users easily achieve delivery from ideas to prototypes to mass production. It is an ideal choice for makers, dreamers, and hobbyists. An ideal creative platform for readers.

1. 3. Hardware specifications of Orange Pi CM4

1. 3. 1. Hardware specifications of Orange Pi CM4 core board

Hardware specifications	
Master chip	Rockchip RK3566
CPU	Quad-core 64-bit Cortex-A55 processor, 22nm advanced process, clocked at up to 1.8GHz
GPU	• ARM Mali G52 2EE graphics processor • Support OpenGL ES 1.1/2.0/3.2, OpenCL 2.0, Vulkan 1.1 • Embedded high-performance 2D acceleration hardware
NPU	• Integrated RKNN NPU AI accelerator, 0.8Tops@INT8 performance • Supports one-click conversion of Caffe/TensorFlow/TFLite/ONNX/PyTorch/Keras/Darknet architecture models
VPU	• 4K@60fps H.265/H.264/VP9 video decoding • 1080P@60fps H.265/H.264 video encoding
PMU	Rockchip RK809-5
Memory	2GB/4GB/8GB （LPDDR4/4x）
Storage	• Onboard eMMC: 16GB/32GB/64GB/128GB • SPI Flash: default blank paste
Wi-Fi+BT	Wi-Fi 5+BT 5.0，BLE（AP6256）

Ethernet transceiver	10/100/1000Mbps Ethernet (onboard PHY chip: YT8531C)
Core board interface	2 x 100PIN, 1 x 24PIN, including the following signals: • 10/100/1000Mbps Ethernet MDI signal • 1x HDMI 2.0 interface, up to 4K@60fps • 1x 4-lane MIPI DSI display interface • 1x 4-lane MIPI CSI camera interface • eDP • SATA3 or PCIe • 1 x USB3.0 • 3 x USB2.0 • RESET, MASKROM, RECOVERY • Headphone jack audio input and output signals • DC 5V input power supply, DC3.3V and 1.8V output power supply
Supported OS	Android11, Ubuntu22.04, Ubuntu20.04, Debian11, Debian12, open source Hongmeng 4.0 Beta1, Orange Pi OS (Arch), Orange Pi OS (OH) based on open source Hongmeng and other operating systems.
Introduction of Appearance Specifications	
PCB Size	55x40mm

1. 3. 2. Hardware specifications of Orange Pi CM4 baseboard

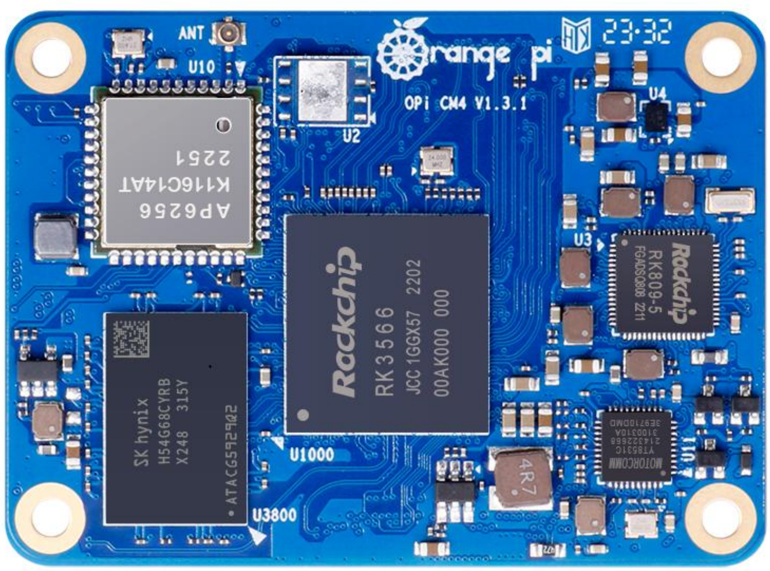
CM4 Base Board Hardware Specifications	
Storage	• M.2 M-KEY slot: SATA3 or PCIe2.0 NVME SSD • TF card slot
Ethernet interface	10/100/1000Mbps Ethernet RJ45 socket
Display	• 1x Micro HDMI TX 2.0, maximum support 4K@60FPS • 1xMIPI DSI 2 Lane • eDP1.3
Camera	2xMIPI CSI 2 Lane
USB	• 1xUSB 2.0 supports Device or HOST mode • 1xUSB 3.0 HOST • 2xUSB 2.0 HOST
Audio	3.5mm headphone jack audio input/output
Button	1xMaskROM key, 1xRECOVERY key, 1x reset key
FAN	4Pin 2.54mm 5V fan interface
40Pin	40Pin function expansion interface, supports the following interface types: GPIO、UART、I2C、SPI、PWM

Power Source	Type-C 5V3A
Supported OS	Android11, Ubuntu22.04, Ubuntu20.04, Debian11, Debian12, open source Hongmeng 4.0 Beta1, Orange Pi OS (Arch), Orange Pi OS (OH) based on open source Hongmeng and other operating systems.
Introduction of Appearance Specifications	
PCB Size	85x56mm

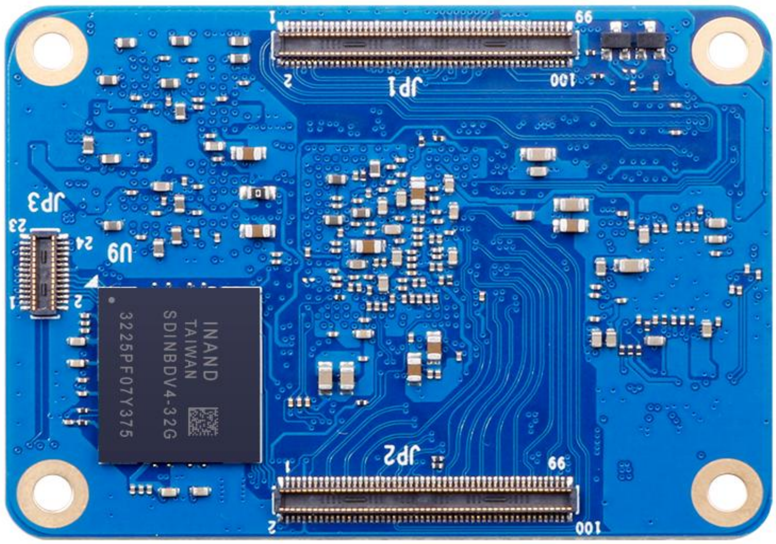
1.4. Top view and bottom view of Orange Pi CM4

1.4.1. Top view and bottom view of Orange Pi CM4 core board

Top view:

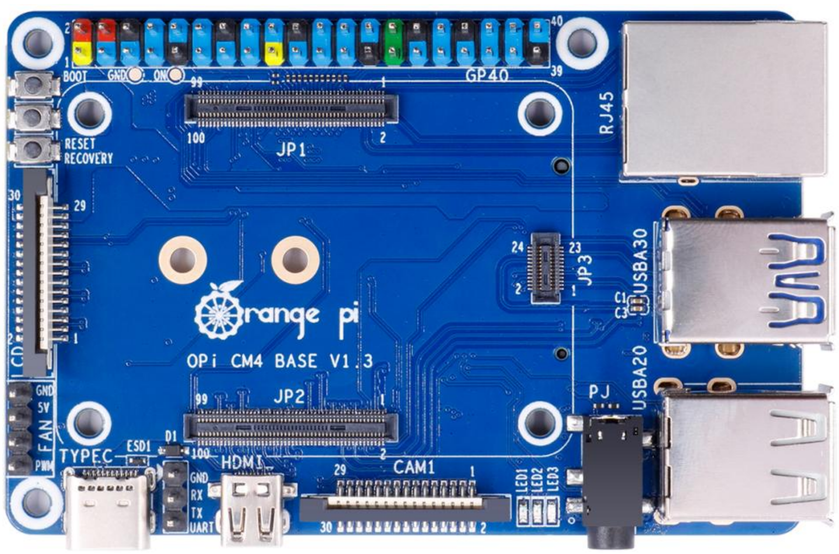


Bottom view:

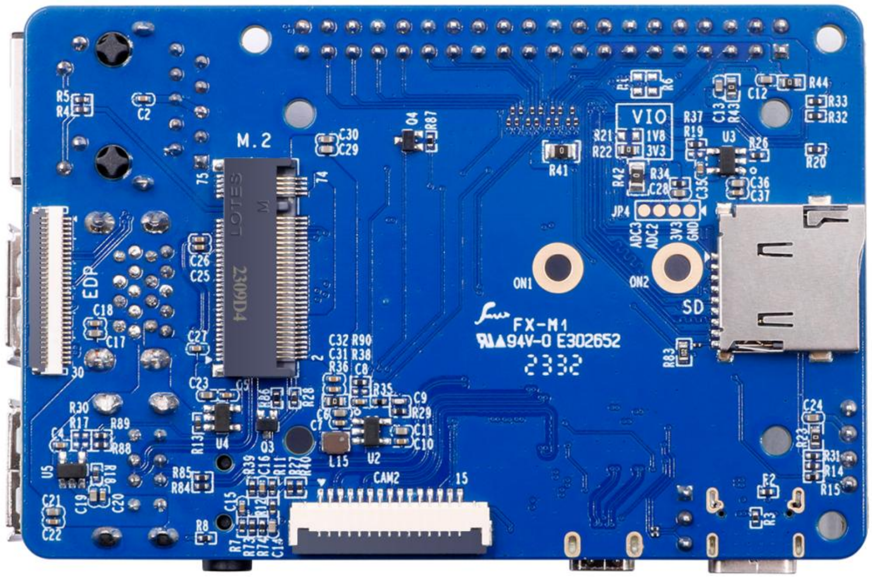


1.4.2. Top and bottom views of the Orange Pi CM4 base board

Top view:

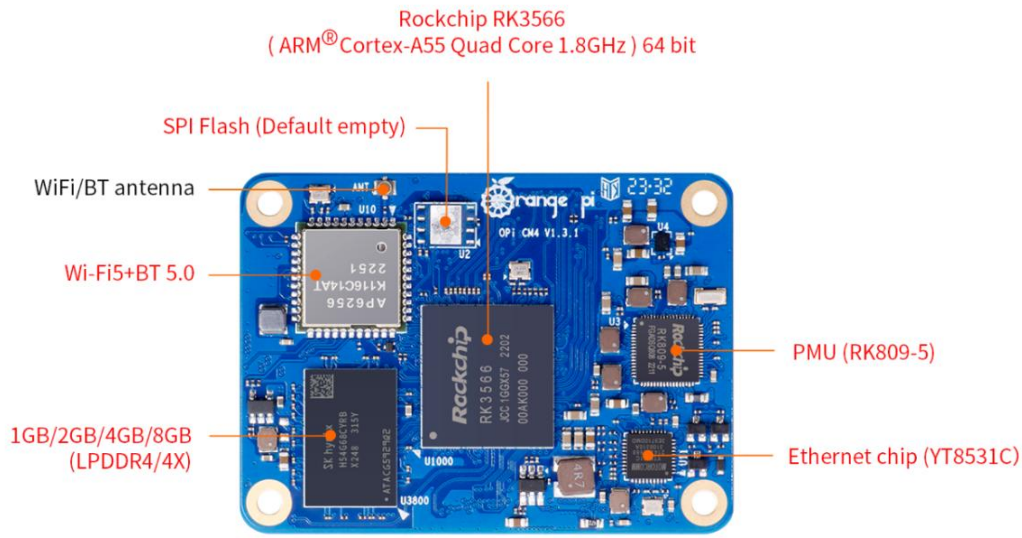


Bottom view:

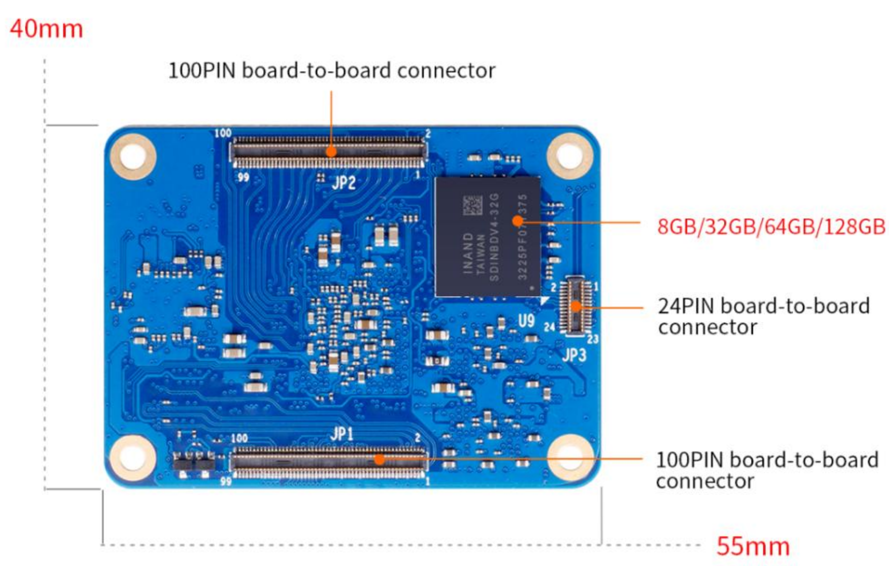

M.2
EDP
VIO
SD
FX-M1
94V-0 E302652
2332
CAM2

1.5. The interface details of Orange Pi CM4

1.5.1. Interface details of Orange Pi CM4 core board

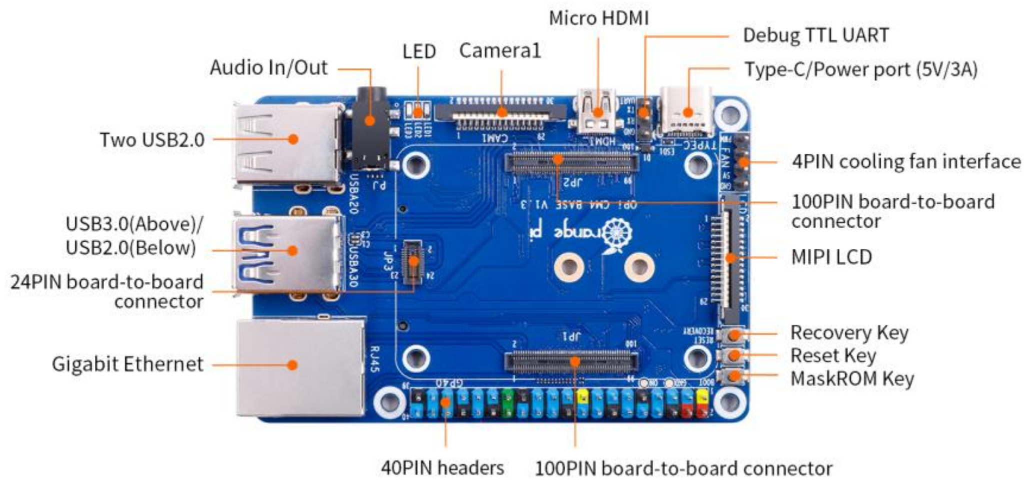


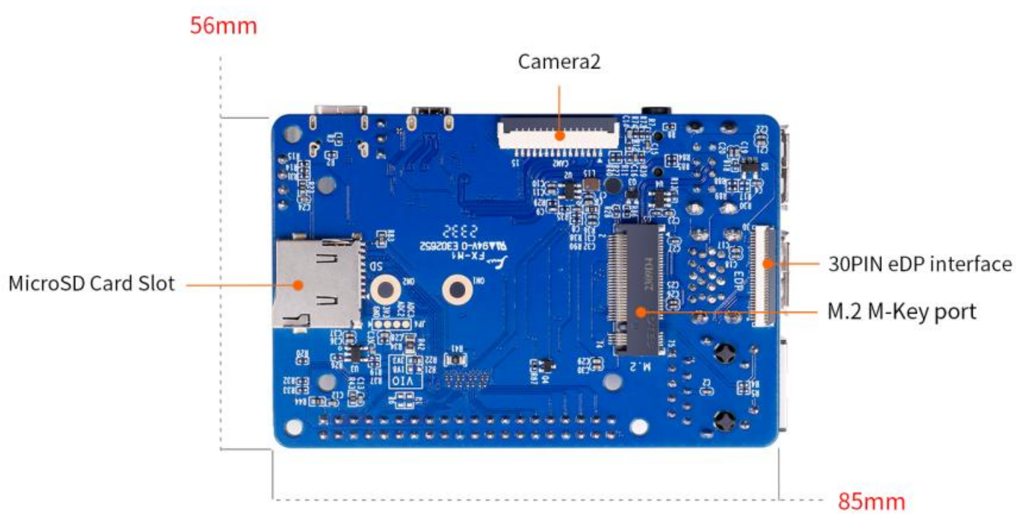
Top View

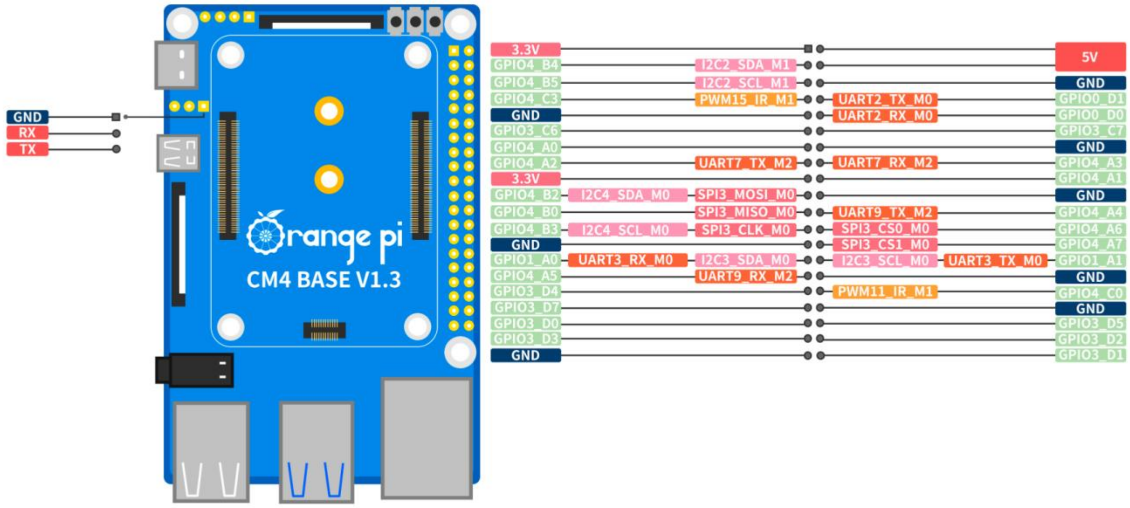


Bottom View

1.5.2. Interface details of Orange Pi CM4 base board







The diameter of the three positioning holes on the base plate is 2.6mm, the diameter of the M.2 PICE device fixing hole is 2.7mm, and the diameter of the four positioning holes on the core board is 2.7mm.

2. Introduction to the use of the development board

2.1. Prepare the required accessories

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

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

3) Display with HDMI interface

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

Note, if you want to connect a 4K monitor, make sure the HDMI cable supports 4K video output.

5) Power adapter. Orange Pi CM4 is recommended to use a 5V/3A Type-C power supply.

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

7) USB camera

8) PCIe NVMe SSD, the development board supports SSDs with M.2 2230 and M.2 2242 specifications. M.2 2280 is also supported, but it cannot be fixed with screws.

The development board does not come with an SPI Flash chip by default. You need to buy it yourself and then solder it on. The SPI Flash chip model we recommend is XM25QU128CWIQT08Q.

Please pay special attention to this point if you need to use SPIFlash+NVMe SSD to start the system.

9) 100M or 1000M network cable, used to connect the development board to the Internet

10) USB2.0 male-to-male data cable, used for burning images to eMMC, TF card, etc.

11) **3.3V** USB to TTL module and DuPont line, when using serial port debugging function, need USB to TTL module and DuPont line to connect the development board

and computer

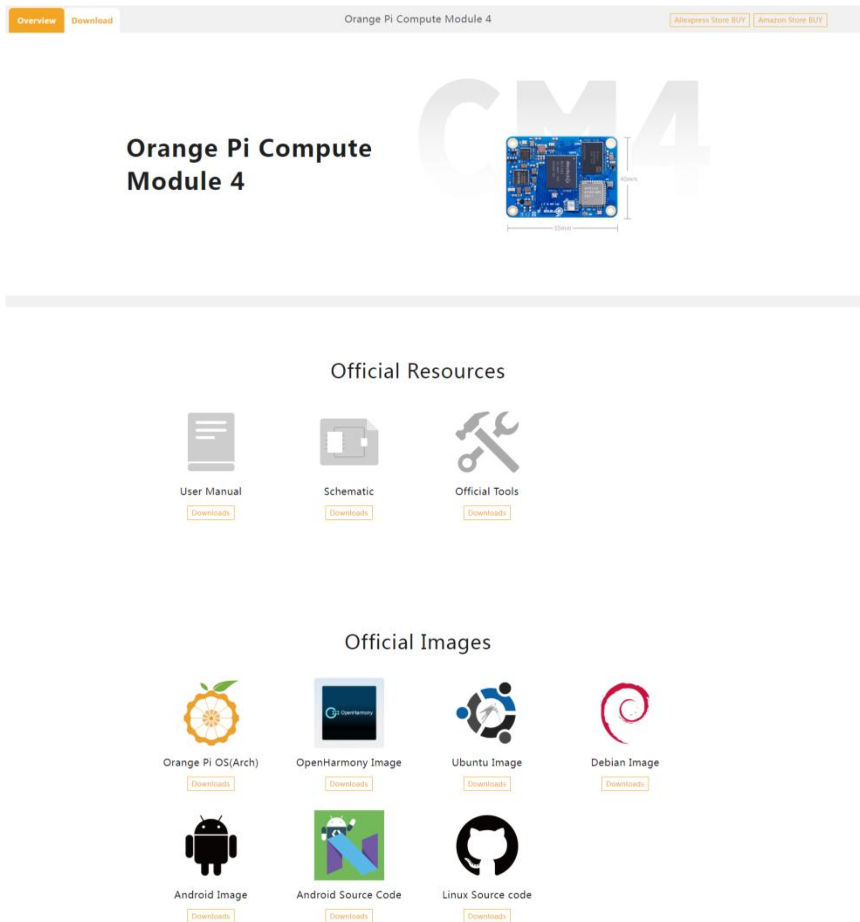
12) Personal computer with Ubuntu and Windows operating systems installed

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

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

1) The website for downloading the English version is：

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-CM4.html



2) The information mainly includes

a. **Android source code**: saved on Google Cloud Disk

b. **Linux source code:** saved on Github

c. **User manual and schematic diagram**: saved on Google Cloud Disk

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 Cloud Disk

f. **Ubuntu image**: saved on Google Cloud Disk

g. **Debian image**: saved on Google Cloud Disk

h. **Orange Pi OS Arch image**: saved on Google Cloud Disk

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

Note that the Linux images mentioned here specifically refer to Linux distribution images such as Debian, Ubuntu and OPi OS Arch downloaded from the Orange Pi's data download page**.**

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

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

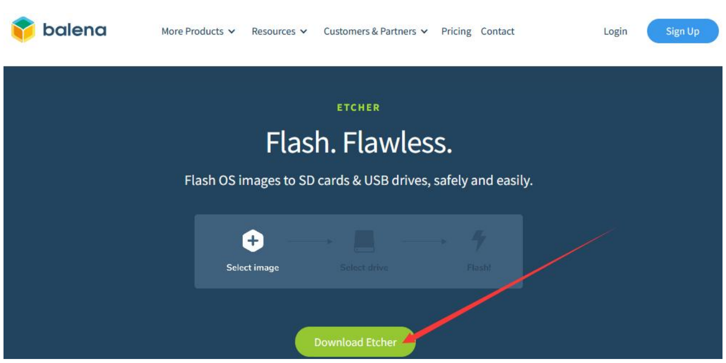
2) Then use the card reader to insert the TF card into the computer

3) Download the Linux operating system image file compression package that you want to burn from the Orange Pi's 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.

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

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

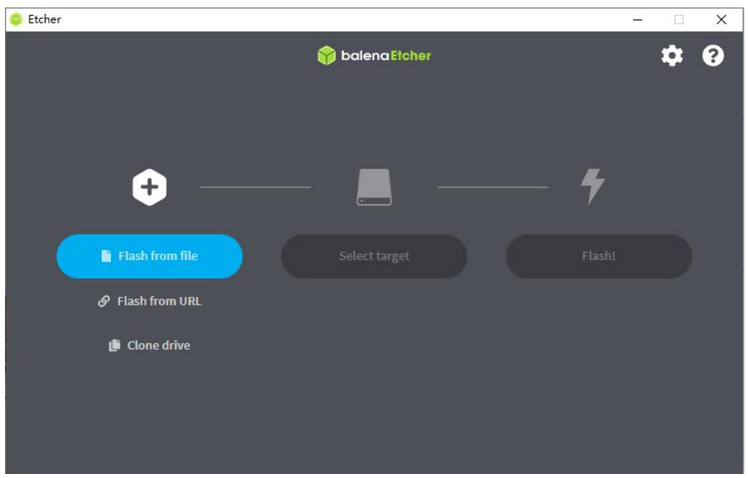
5) After entering the balenaEtcher download page, click the green download button to jump to the place where the software is downloaded



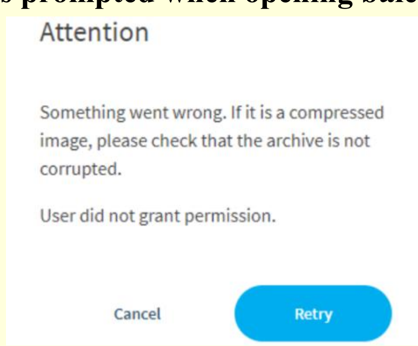
6) Then you can choose to download the Portable version of balenaEtcher software. The

Portable version does not need to be installed, and you can use it by double-clicking to open it

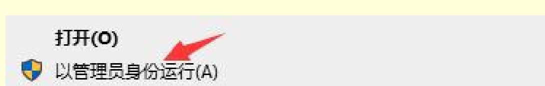
7) If you downloaded the version of balenaEtcher that needs to be installed, please install it first and then use it. If you downloaded the Portable version of balenaEtcher, just double-click to open it. The balenaEtcher interface after opening is as shown below:



If the following error is prompted when opening balenaEtcher:



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

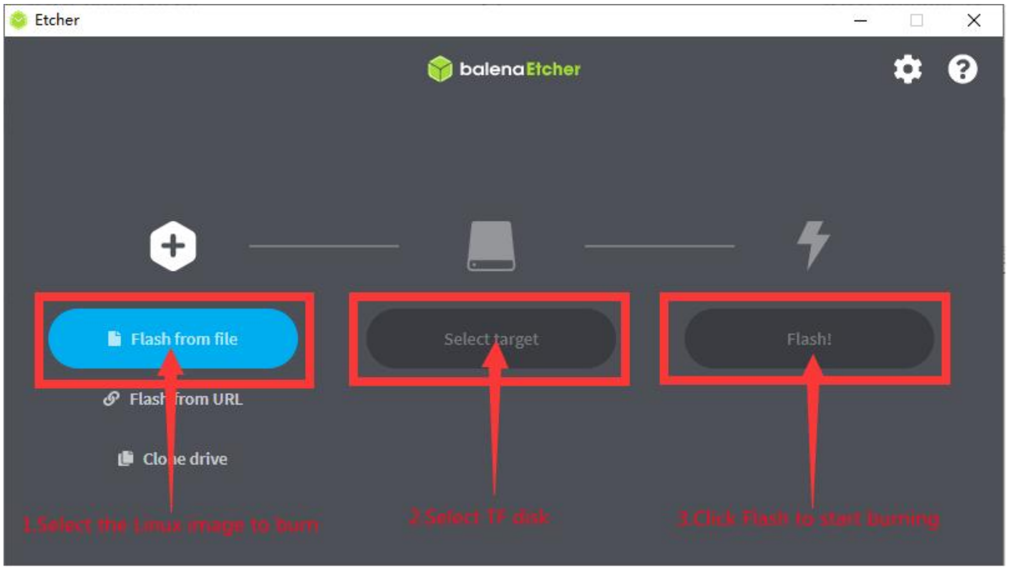


8) The specific steps to use balenaEtcher to burn a Linux image are as follows

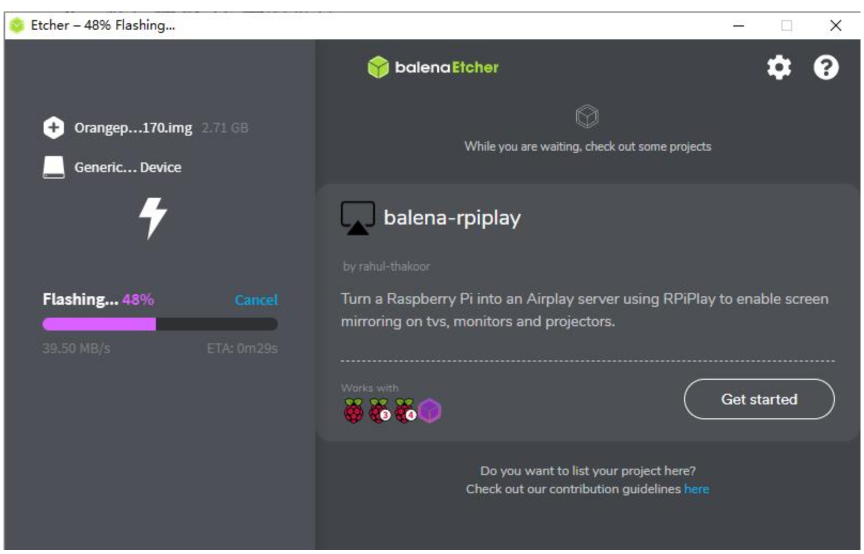
a. First select the path of the Linux image file to be burned

b. Then select the drive letter of the TF card

c. Finally click Flash and it will start burning the Linux image to the TF card.

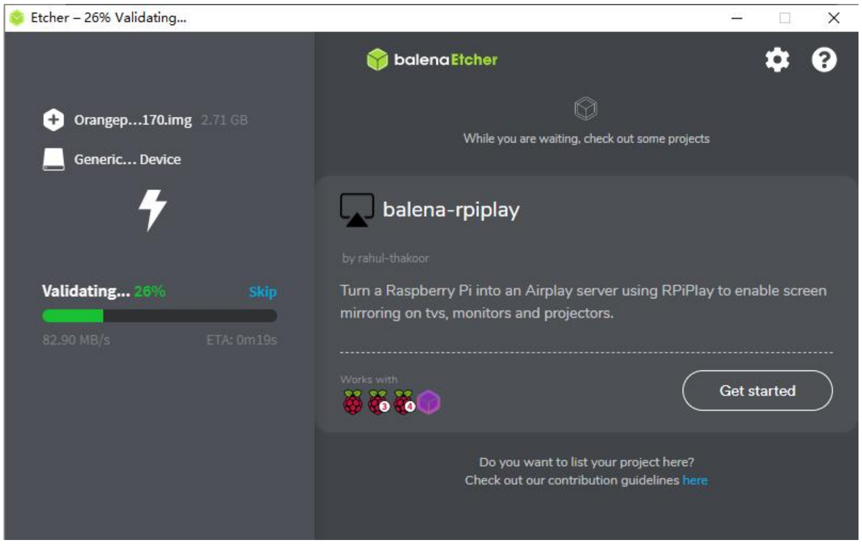


9) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card

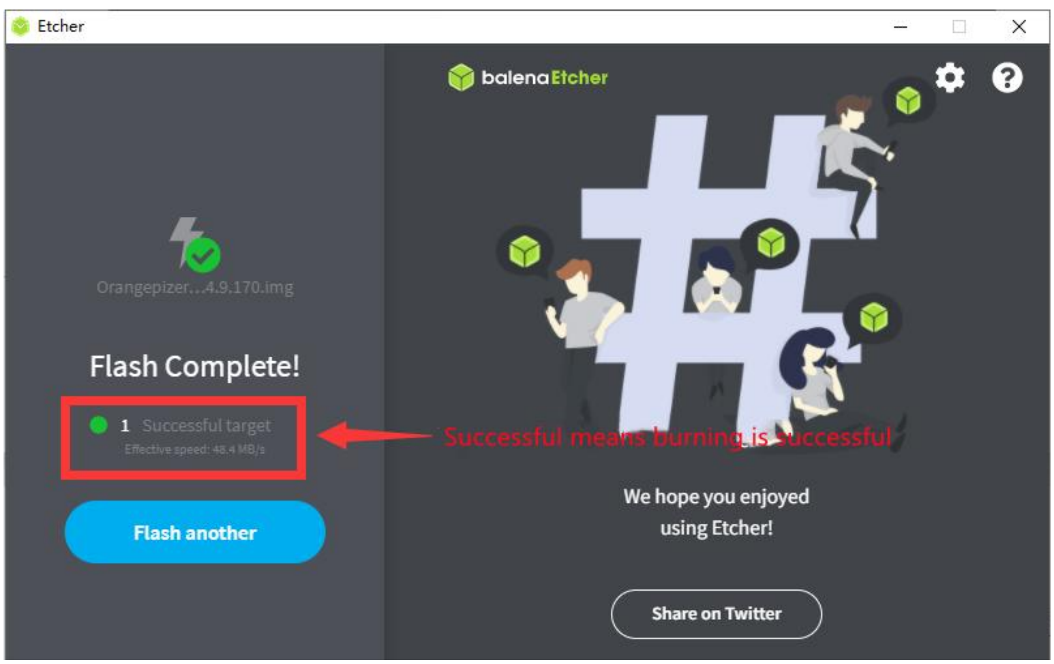


10) After burning the Linux image, balenaEtcher will also verify the image burned into the TF card by default to ensure that there is no problem in the burning process. As shown in the figure below, a green progress bar indicates that the image has been burnt,

and balenaEtcher is verifying the burnt image



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



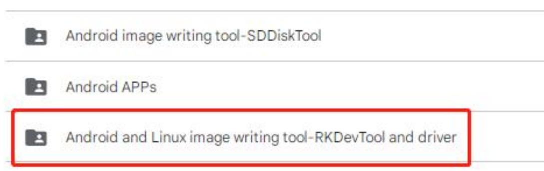
2. 3. 2. How to use RKDevTool to burn Linux image to TF card

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

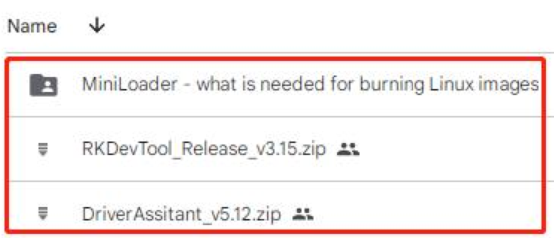
2) You also need to prepare a TF card with a capacity of 16GB or larger. The transmission speed of the TF card must be **class 10** or above. It is recommended to use TF cards from SanDisk and other brands.

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

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



b. Then download all the files below



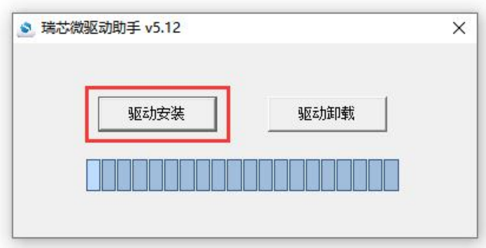
Note that the "MiniLoader-things needed to burn Linux images" folder will be referred to as the MiniLoader folder below.

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

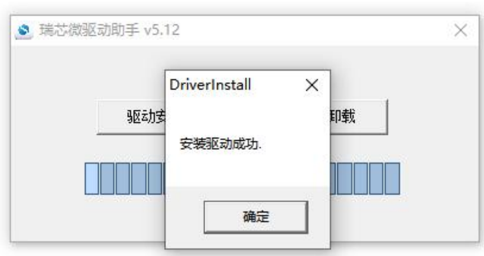
5) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

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

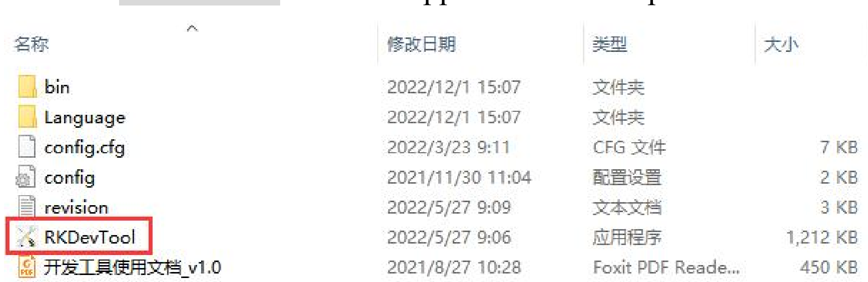
a. Click the "**Driver Installation**" button



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.

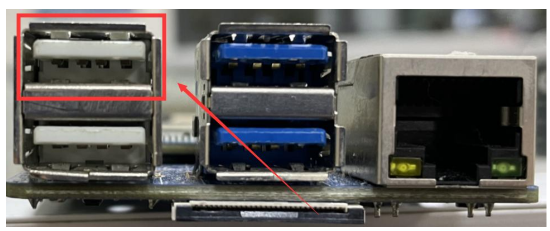


8) After opening the **RKDevTool** burning tool, because the computer has not

connected to the development board through the USB2.0 male-to-male data cable at this time, the lower left corner will prompt "**No device found**"

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

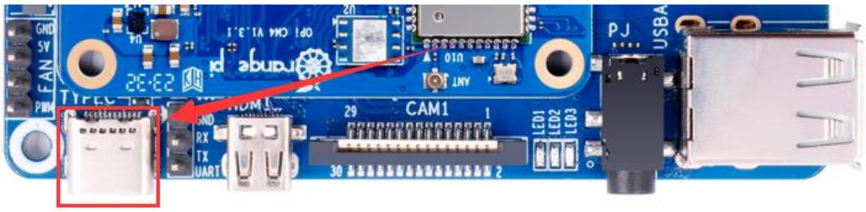
a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the USB2.0 programming port on the development board is as shown in the figure below.



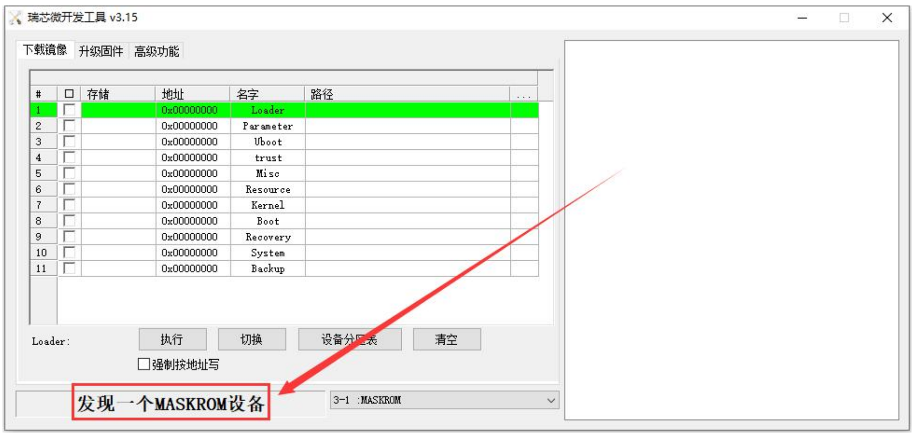
b. Make sure the TF card slot is not inserted into the TF card

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

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

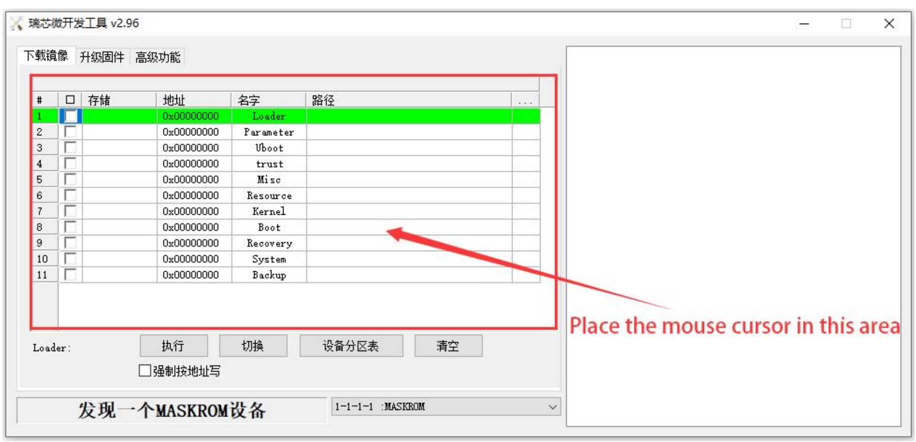


e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"

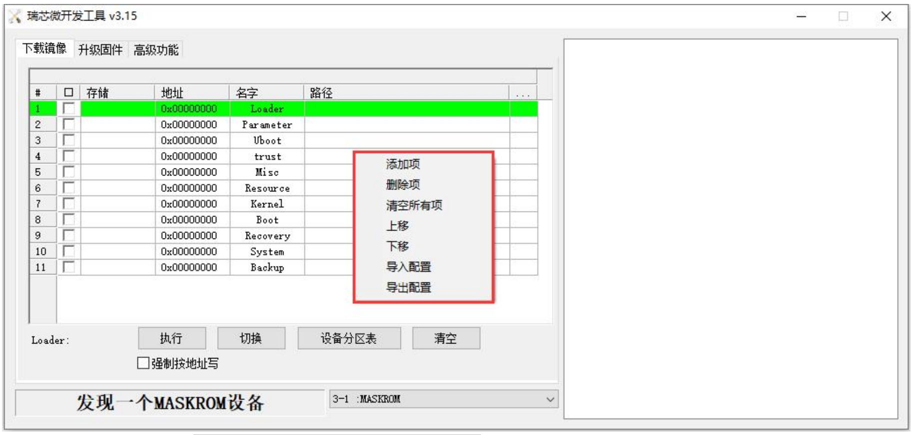


f. At this time, please insert the TF card into the TF card slot

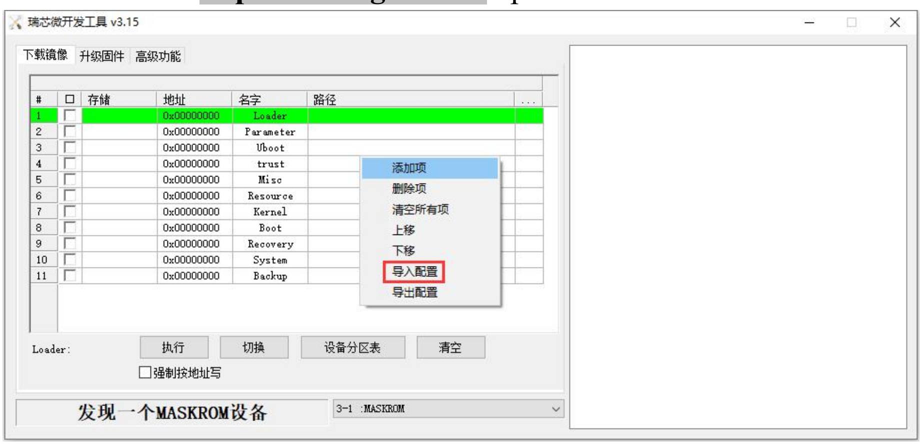
g. Then place the mouse cursor in the area below



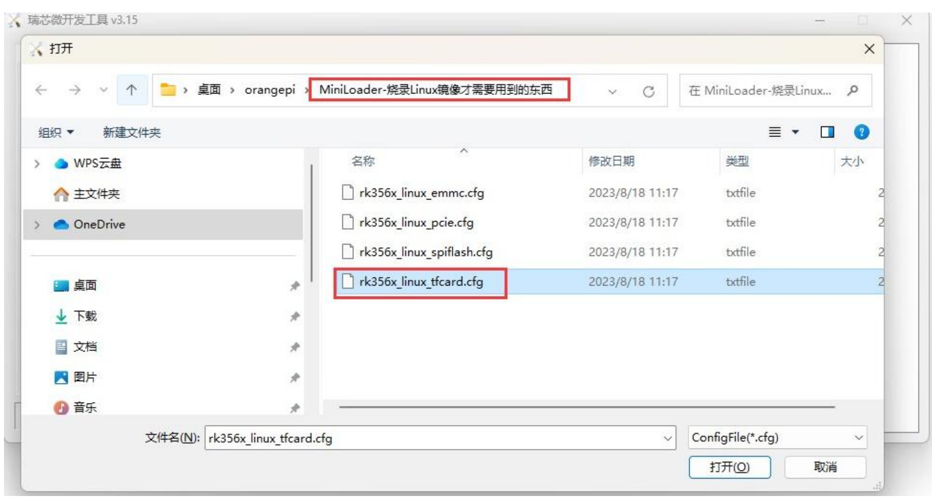
h. Then click the right mouse button and the selection interface shown in the figure below will pop up



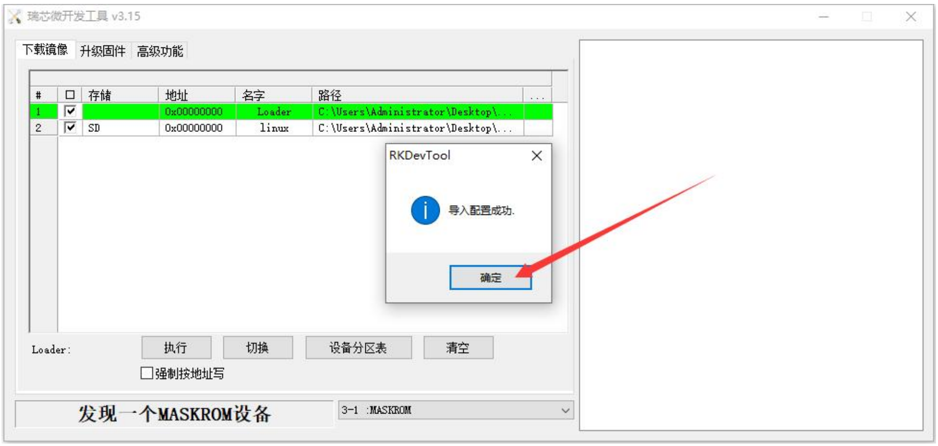
i. Then select the **import configuration** option



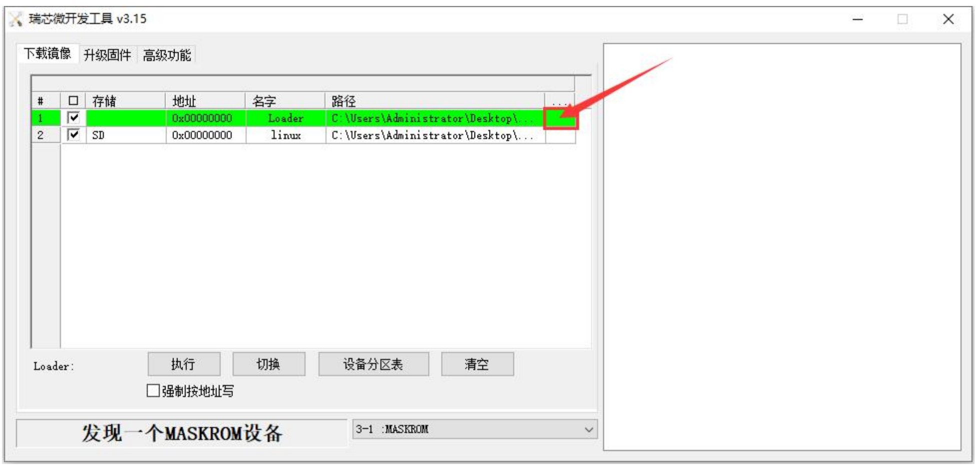
j. Then select the **rk356x_linux_tfcard.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and click **Open**



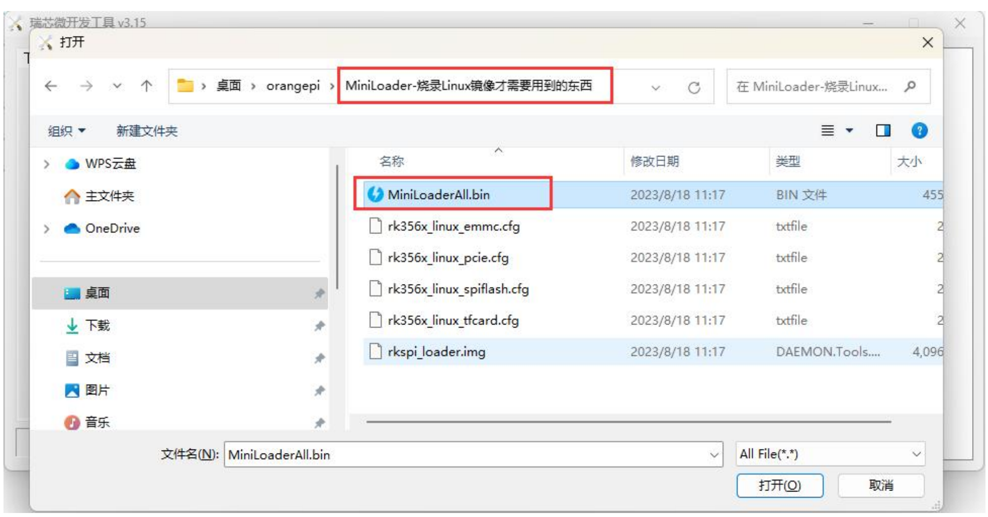
k. Then click **OK**



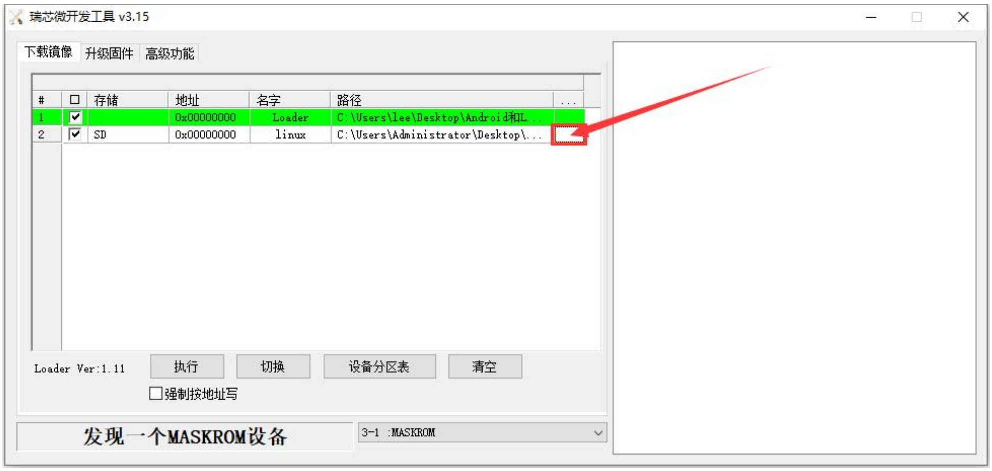
l. Then click the position shown in the figure below



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

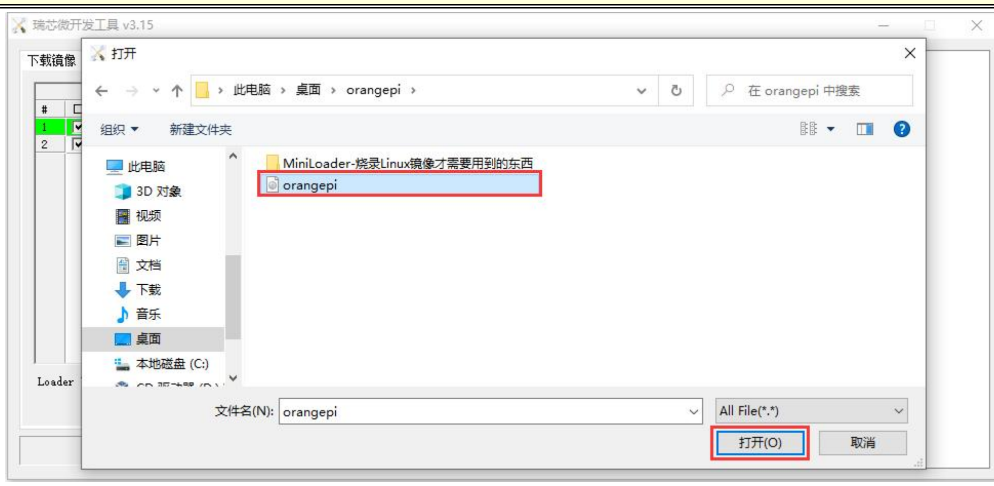


n. Then click on the location shown in the picture below

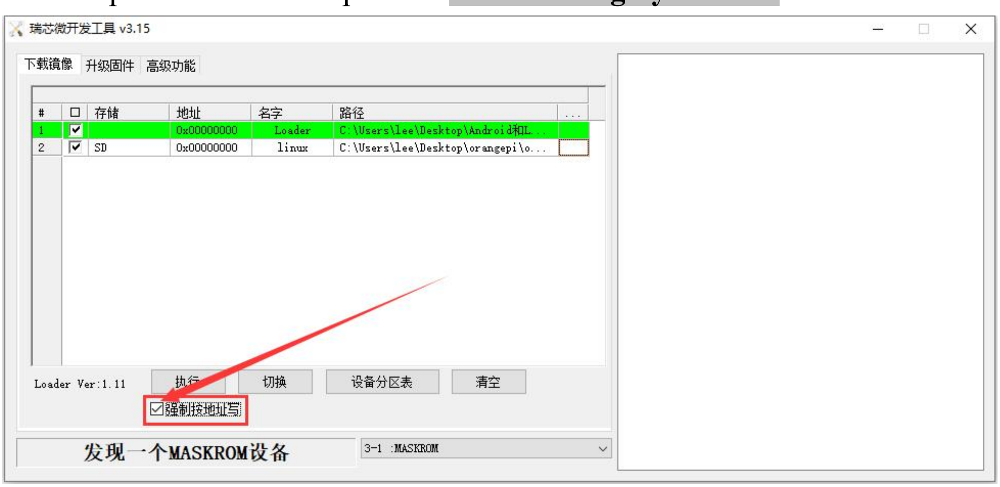


o. Then select the path of the Linux image you want to burn, and then click **Open**

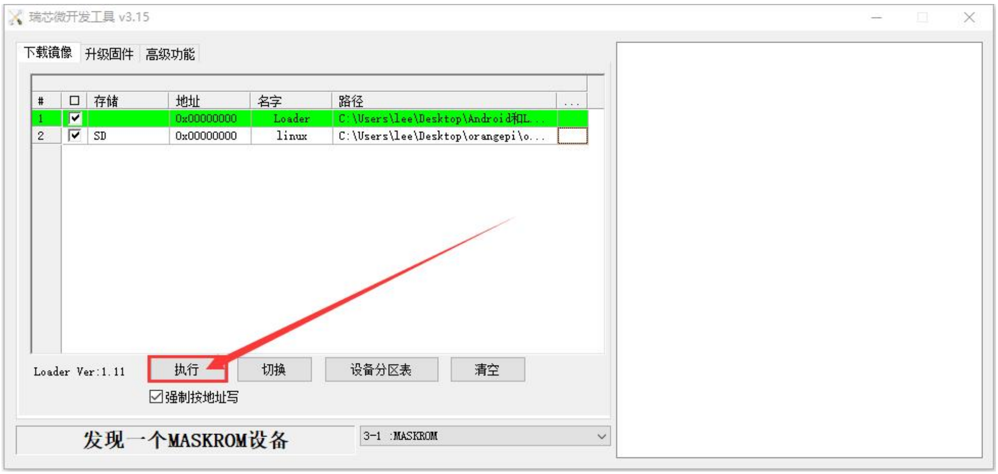
Before burning the image, it is recommended to rename the Linux image to be burned to orangepi.img or another shorter name, so that you can see the percentage of the burning progress when burning the image.



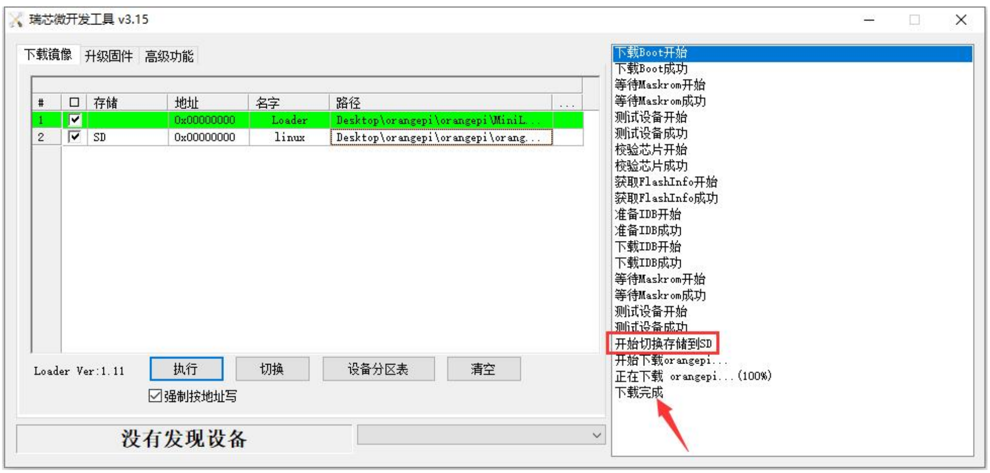
p. Then please check the option to **force writing by address**.



q. Click the execute button again and it will start burning the Linux image to the tf card of the development board.



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



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

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

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

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

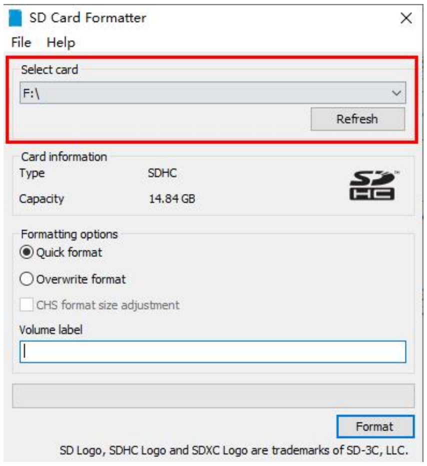
3) Then format the TF card

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

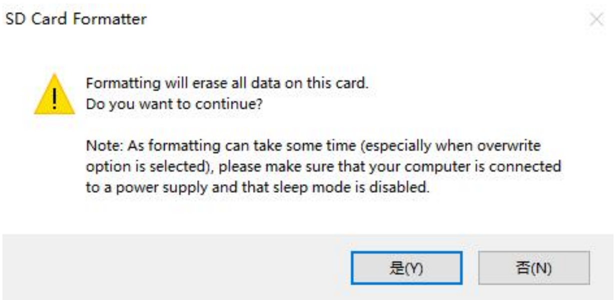
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

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

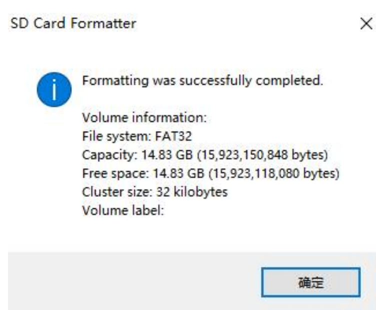
c. If only the TF card is inserted into the computer, the drive letter of the TF card will be displayed in the "**Select card**" column. If multiple USB storage devices are inserted into the computer, you can select the drive letter corresponding to the TF card through the drop-down box.



d. Then click "**Format**". A warning box will pop up before formatting. After selecting "**Yes (Y)**", formatting will begin.



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



4) Download the compressed package of the Linux operating system image file you want to burn from the **Orange Pi information download page**, and then use the decompression software to decompress it. In the decompressed file, the file ending with "**.img**" is the image file of the operating system. The size is generally above 2GB

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

a. The download page of Win32Diskimager is

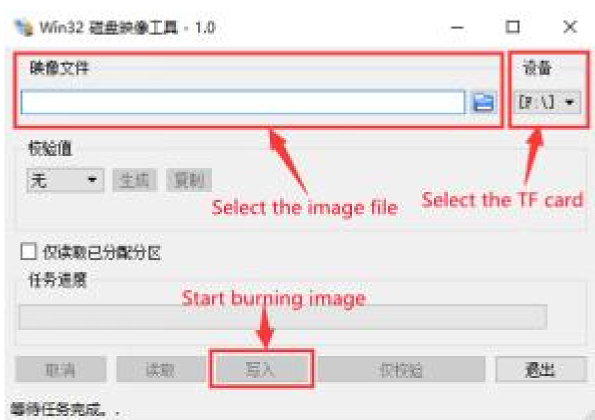
http://sourceforge.net/projects/win32diskimager/files/Archive/

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

a) First select the path to the image file

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

c) Finally click "Write" to start burning



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

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

Note that the Linux images here refer specifically to the images of Linux distributions like Debian, Ubuntu, and OPi OS Arch downloaded from the Orange Pi's data download page, and Ubuntu PC refers to a personal computer with Ubuntu installed.

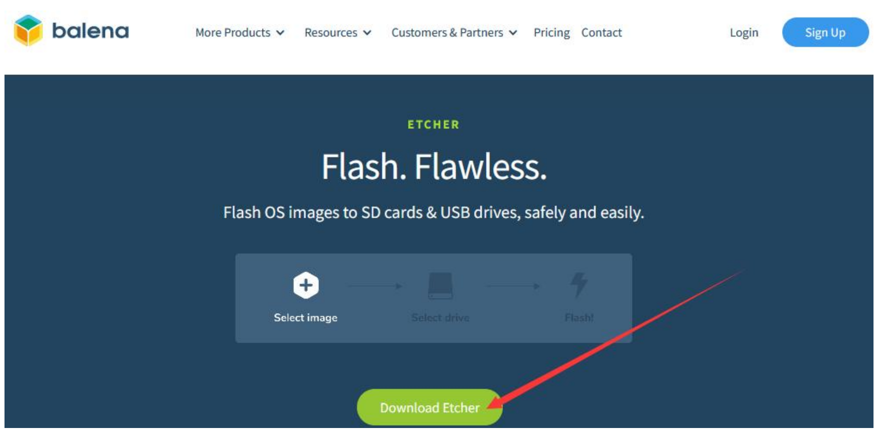
1) First prepare a 16GB or larger capacity TF card, TF card transmission speed must be class10 or above, it is recommended to use Sandisk and other brands of TF card

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

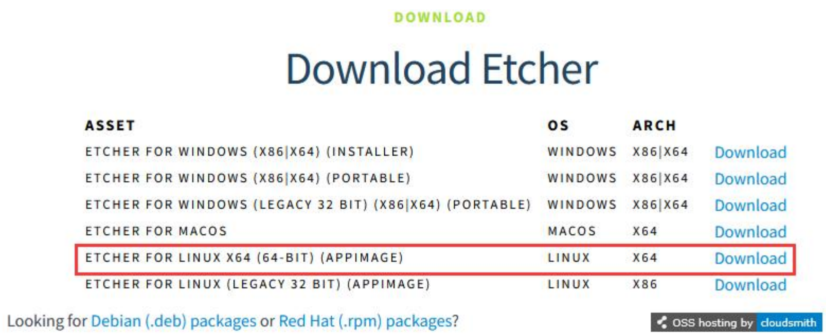
3) Download the balenaEtcher software, the download address is

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

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



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



6) Download the compressed Linux operating system image file you want to burn from **Orange Pi's data download page**, and then use the decompression software to decompress it. Files ending with "**.img** "in the decompressed files are the image files of the operating system, and the size is generally more than 2GB

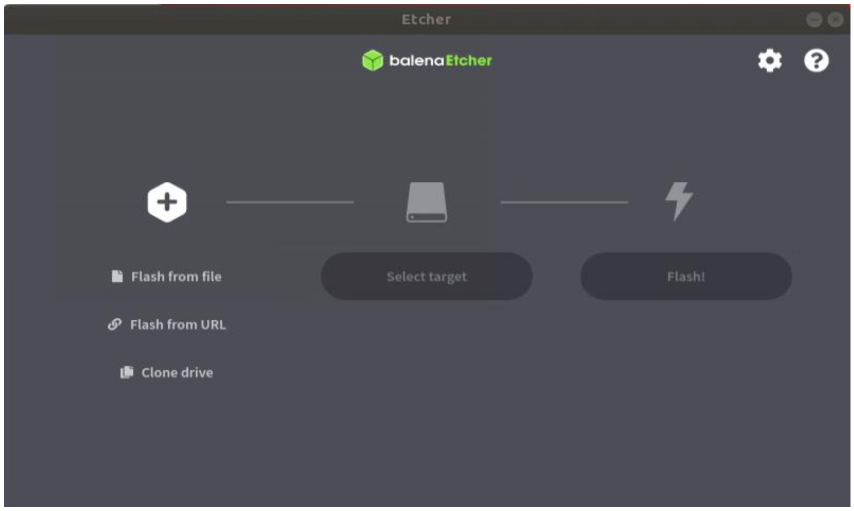
The following command is used to decompress the compressed package ending in 7z

```
test@test:~$ 7z x Orangepicm4_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
test@test:~$ ls Orangepicm4_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.*
Orangepicm4_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
Orangepicm4_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.sha        #Checksum file
Orangepicm4_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img     #Image file
```

7) After decompressing the image, you can first use the **sha256sum -c *.sha** command to calculate whether the checksum is correct, if the success indicates that the downloaded image is not wrong, you can rest assured to burn to the TF card, if the **checksum does not match** that there is a problem with the downloaded image, please try to re-download

```
test@test:~$ sha256sum -c *.sha
Orangepicm4_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img: OK
```

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

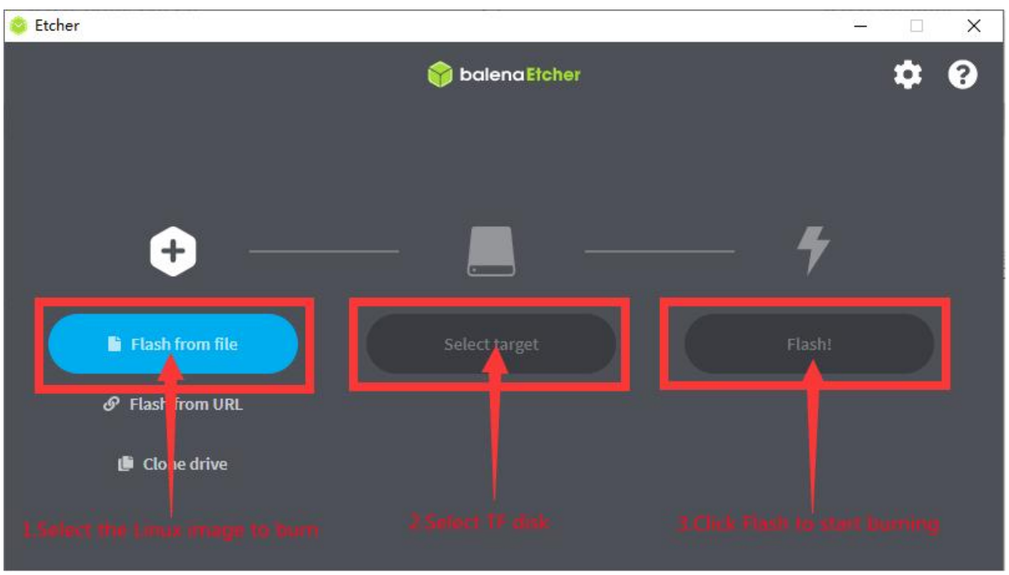


9) The specific steps to burn a Linux image using balenaEtcher are shown below

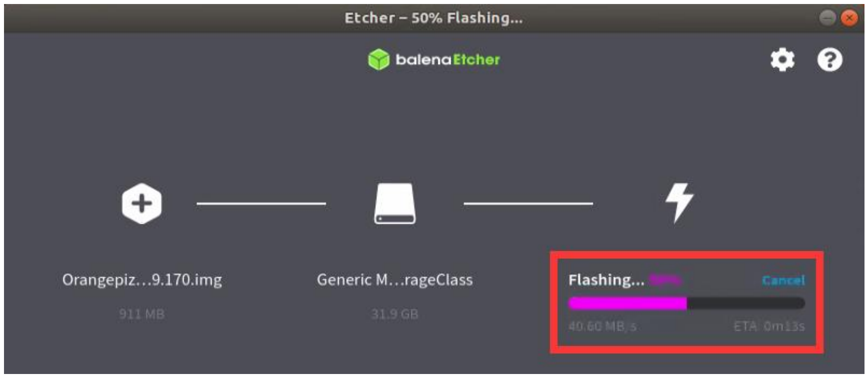
a. Select the path of the Linux image file to be burned

b. Then select the drive letter of the TF card

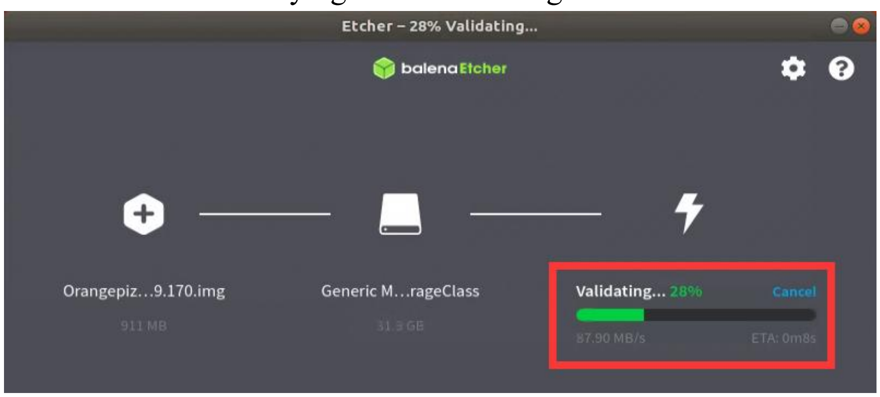
c. Finally, clicking Flash will start burning the Linux image to the TF card



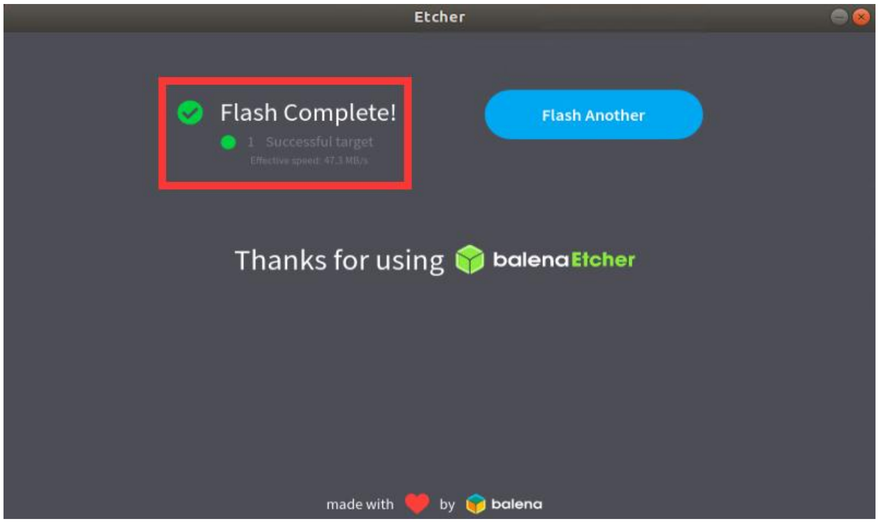
10) The interface displayed during balenaEtcher burning the Linux image is as shown in the following figure. In addition, the progress bar showing purple indicates that the Linux image is burning to the TF card



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



12) The display interface of balenaEtcher after the successful burning is shown as the following figure. If the green indicator icon is displayed indicating that the image is burned successfully, you can exit balenaEtcher at this time, and then pull out the TF card and insert it into the TF card slot of the development board



2. 5. Methods for burning Linux images to eMMC

2. 5. 1. Methods for burning Linux images to eMMC using RKDevTool

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

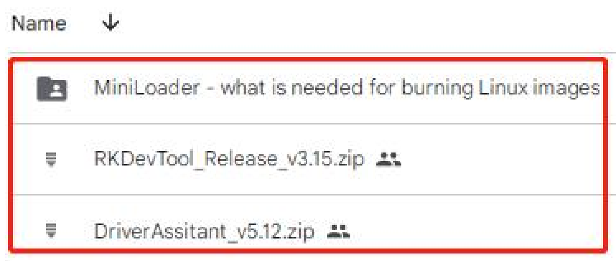
Note that the Linux images here specifically refer to the images of Linux distributions such as Debian, Ubuntu, and OPi OS Arch downloaded from the Orange Pi's data download page.

1) First of all, you need to prepare a good quality USB2.0 public-to-public data cable

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

a. On the **Orange Pi's data download page**, first select the official tool and then go to the following folder

b. Then download all the files below



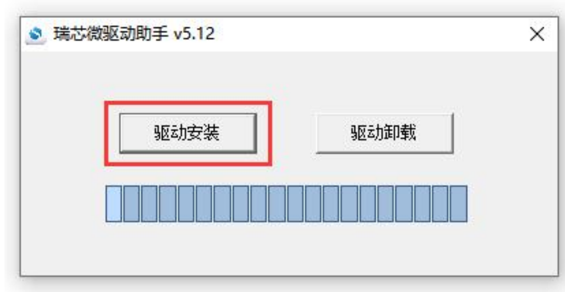
Note that the folder "MiniLoader- What you Need to burn Linux images" is hereinafter referred to as the MiniLoader folder.

3) Download the compressed Linux operating system image file you want to burn from the **Orange Pi's data download page**, and then use the decompression software to decompress it. After decompressing the file, the file ending with "**.img** "is the image file of the operating system, and the size is generally more than 2GB

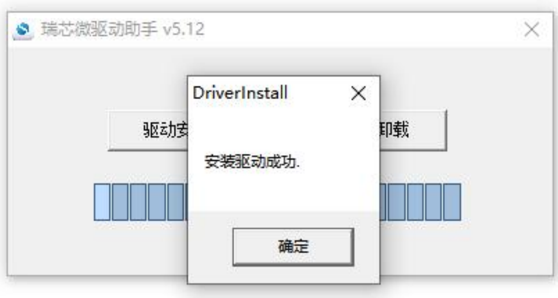
4) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

5) Open **DriverInstall.exe** and install the SWick micro driver as follows

a. Click the **“Driver Installation”** button

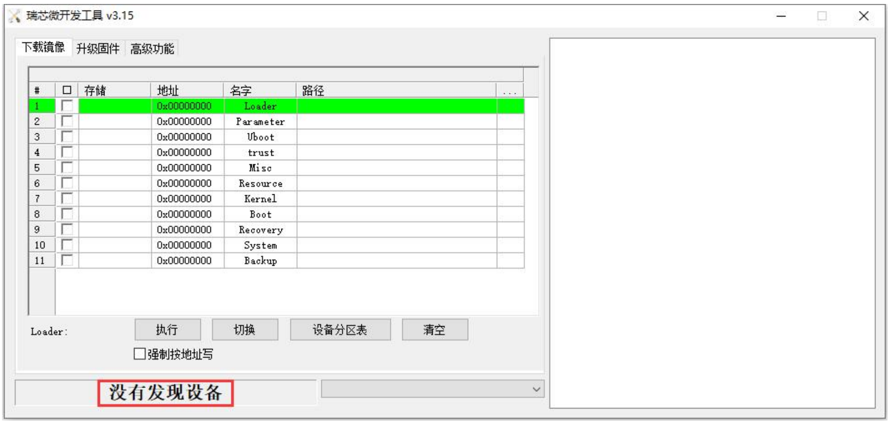


b. Wait for a period of time, the pop-up window will prompt "**Driver installation successful**", and then click the "**OK**" button



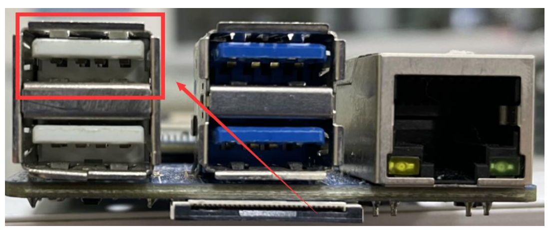
6) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, find **RKDevTool** in the decompressed folder and open it

7) After opening the **RKDevTool** burning tool, because the computer has not connected to the development board through the USB2.0 public-to-public data cable at this time, the lower left corner will indicate "**no device found**".



8) Then start burning the Linux image into eMMC

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



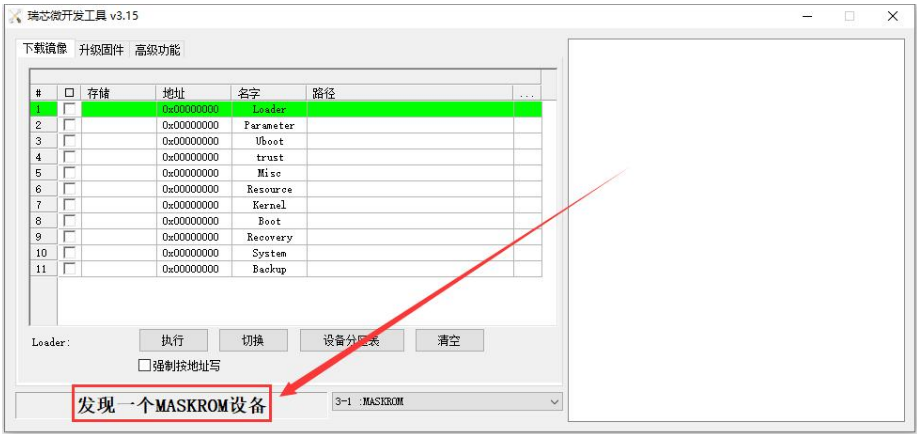
b. Ensure that no TF card is inserted into the development board and no power supply is connected

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

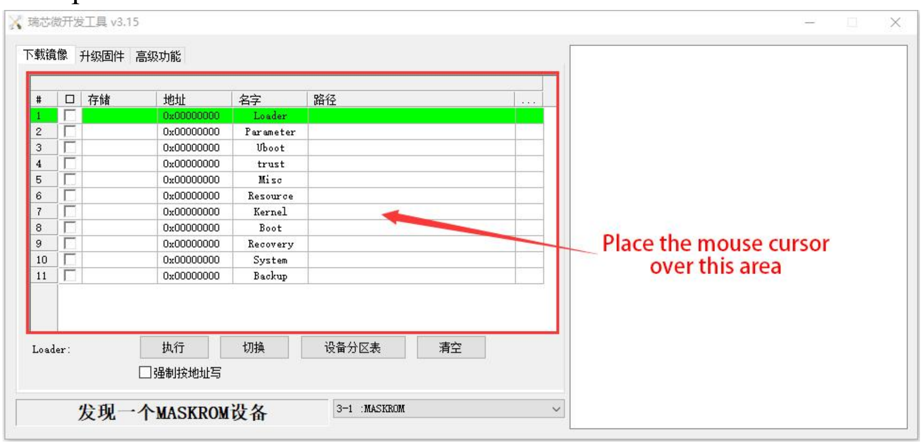
d. Then connect the Type-C interface to the development board, and power on, and then you can release the MaskROM button

e. If the previous steps are smooth, the development board will enter

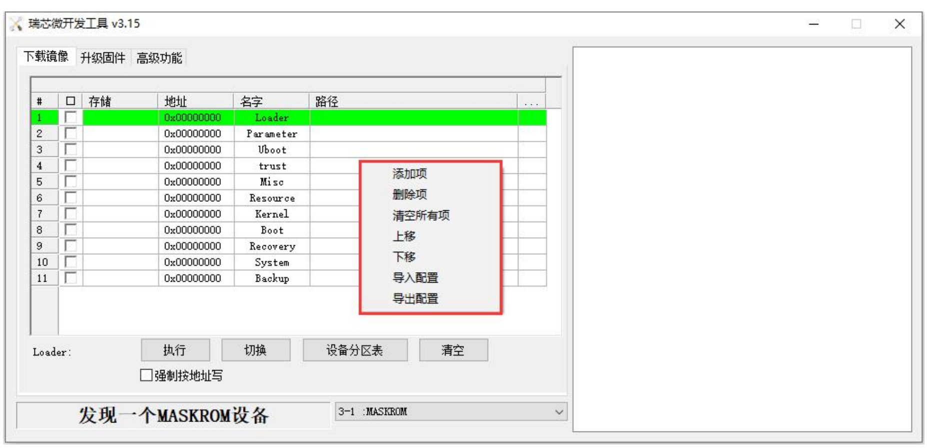
MASKROM mode at this time, and the interface of the burning tool will prompt "**Found a MASKROM device**".



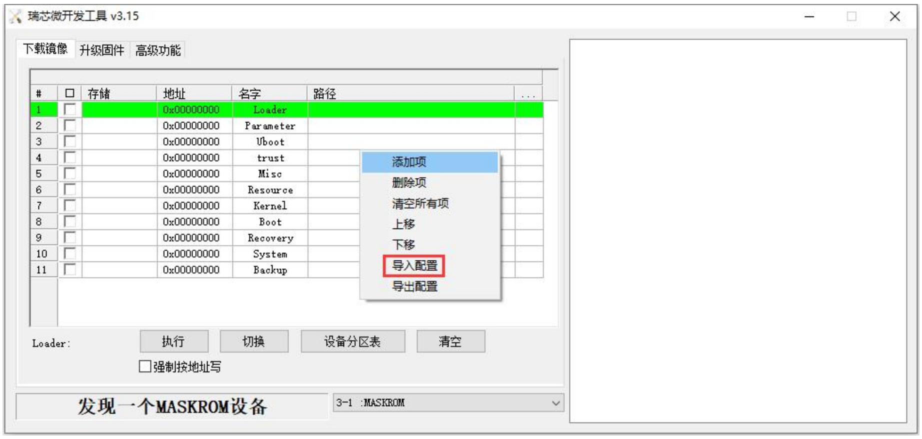
f. Then place the mouse cursor on the field below



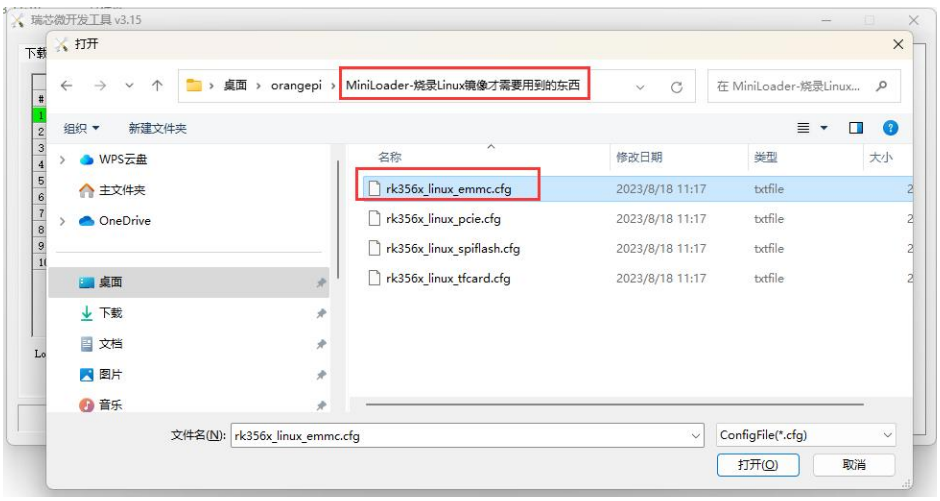
g. Then click the right mouse button to pop up the selection interface as shown in the following figure



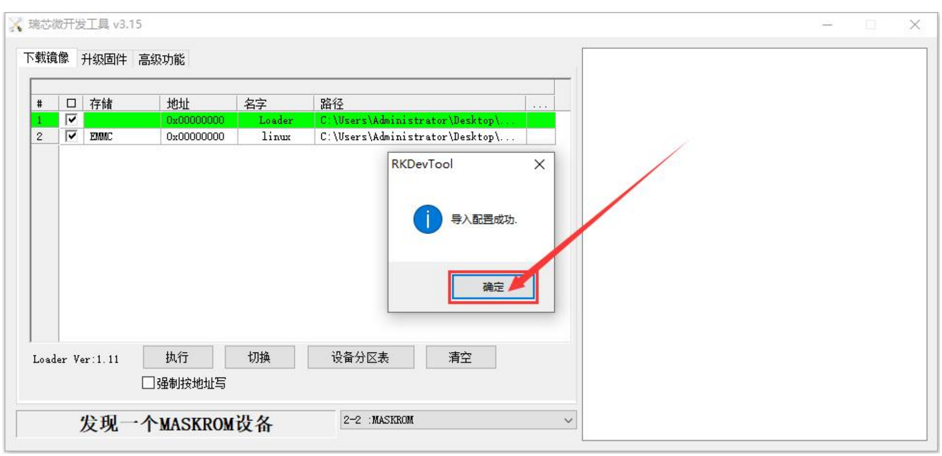
h. Then select the **Import configuration** option



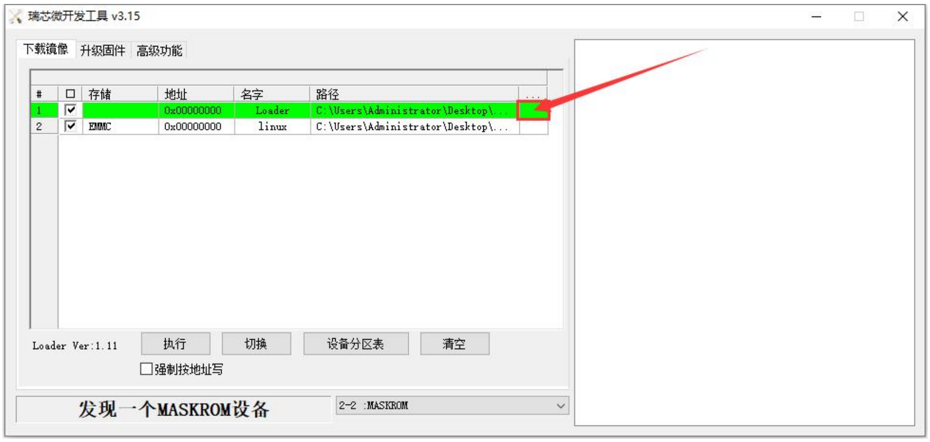
i. Then select the **rk356x_linux_emmc.cfg** configuration file in the **MiniLoader** folder downloaded earlier and click **Open**



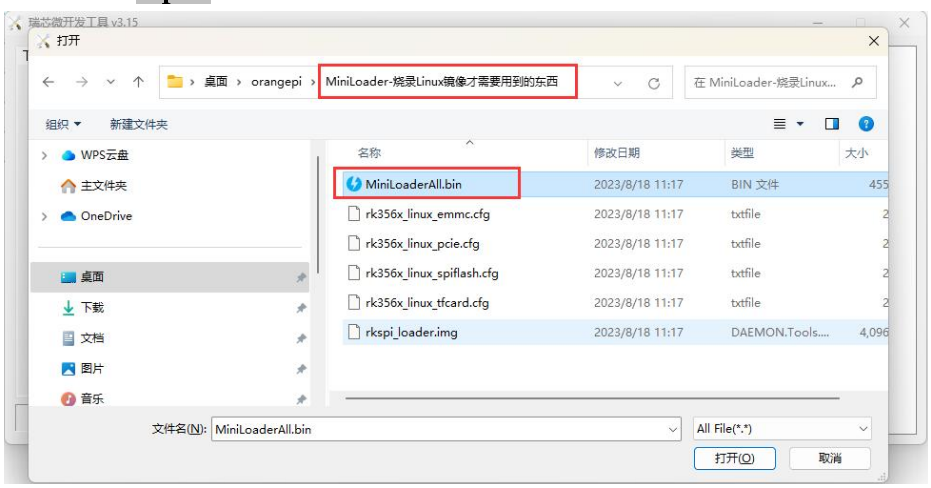
j. Then click **OK**



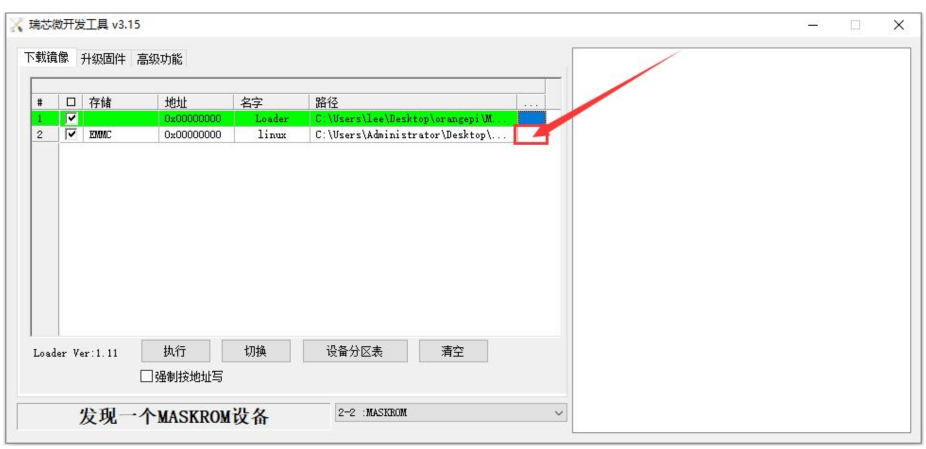
k. Then click on the location shown in the image below



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



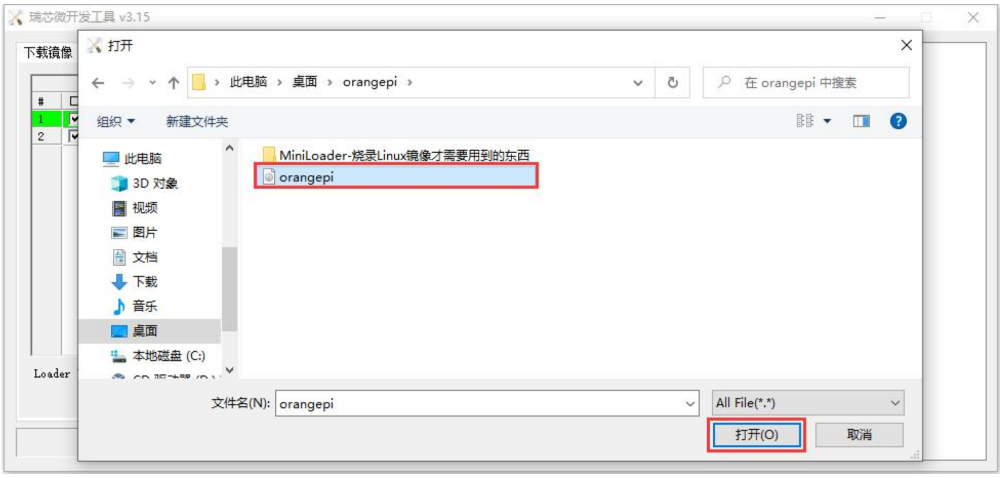
m. Then click on the location shown in the image below



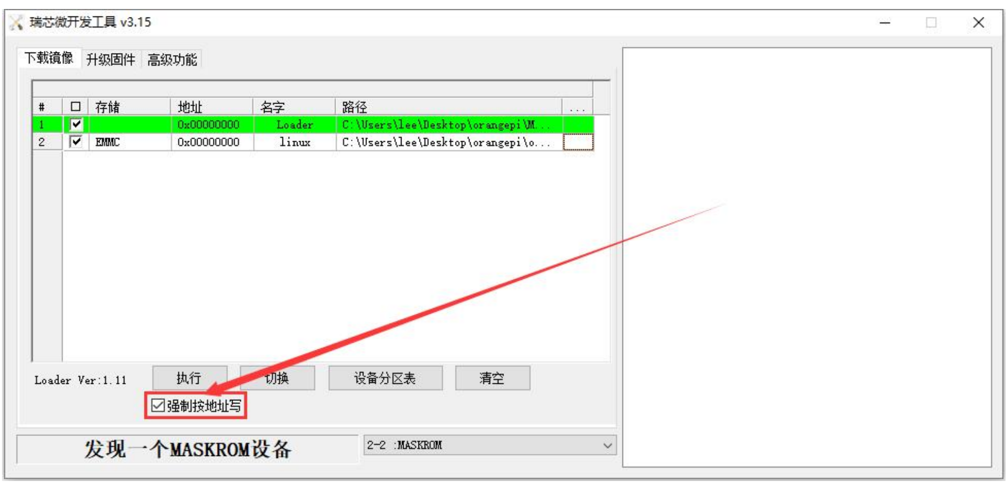
n. Then select the path of the Linux image you want to burn and click **Open**

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

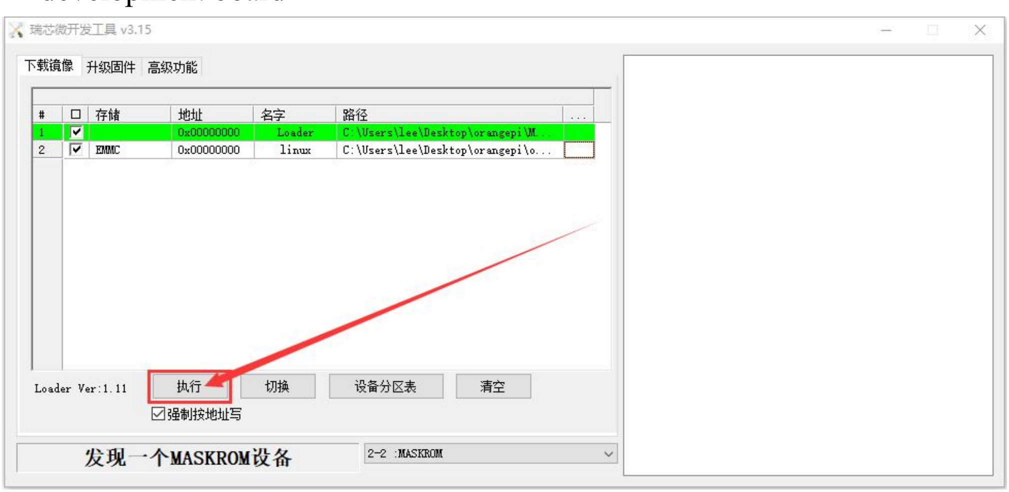
orangepi.img or something shorter so that you can see the percentage of the burn progress when burning the image.



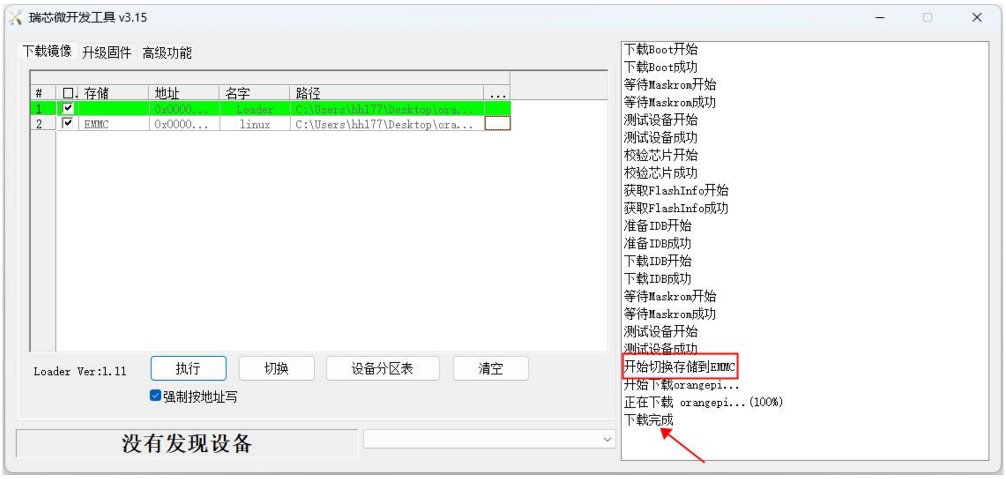
o. Then check the **Force write to address** option



p. Click the Execute button again to start burning the Linux image to eMMC on the development board



q. The following figure shows the display log after the Linux image is burned



r. After the Linux image is burned into eMMC, the Linux system automatically starts

2. 5. 2. How to burn a Linux image to eMMC using the dd command

Note that by Linux images, I specifically mean images of Linux distributions such as Debian, Ubuntu, or OPi OS Arch downloaded from the Orange Pi's data download page.

1) Using dd command to burn Linux image to eMMC needs TF card to complete, so first need to burn Linux image to TF card, and then use TF card to start the development board into the Linux system. For the method of burning a Linux image to a TF card, see the two sections for **the method of burning a Linux image to a TF Card based on a Windows PC** and **the method of burning a Linux image to a TF card based on an Ubuntu PC**.

2) After starting the Linux system with the TF card, we first upload the decompressed Linux image file (Debian, Ubuntu image or OPi OS (Arch) image downloaded from the official website) to the TF card. For details about how to upload a Linux image file to the development board, see section **Uploading Files to the Linux system on the Development Board**.

3) 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 saved the Linux image of the development board in the directory **/home/orangepi/Desktop**. Then go to the **/home/orangepi/Desktop** directory and you will see the uploaded image file.

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepicm4_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

How to enter the development board Linux system command line?

1. For details about how to log in to the terminal through the serial port, see section Debugging Serial Port.

2. For details about how to remotely log in to the Linux operating system over ssh, see section "Logging In to the Development Board over SSH".

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

4) Next, we first use the following command to identify the eMMC device node

```
orangepi@orangepi:~/Desktop$ ls /dev/mmcblk*boot0 | cut -c1-12
/dev/mmcblk0
```

5) Then we can use the dd command to clear eMMC, pay attention to the **of=** parameter after please fill in the output result of the above command

```
orangepi@orangepi:~/Desktop$ sudo dd bs=1M if=/dev/zero of=/dev/mmcblk0 count=1000 status=progress
orangepi@orangepi:~/Desktop$ sudo sync
```

6) You can then use the dd command to burn the Linux image of the development board into eMMC

a. **if** = parameter in the command is to fill in behind the Linux + Linux mirror mirror to store the full path name (such as **/home/orangepi/Desktop/Linux image name**). Since we have entered the path of the Linux image above, we only need to fill in the name of the Linux image.

b. Do not copy the Linux image name in the following command. Replace it with the actual image name (because the version of the image may be updated).

```
sudo dd bs=1M if=Orangepicm4_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/mmcblk0 status=progress

sudo sync
```

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

A detailed description of all the parameters of the dd command and more usage can be seen by running the man dd command on Linux.

7) After successfully burning the Linux image of the development board to eMMC, you can use the poweroff command to shut down. Then pull out the TF card, power it on again, and the Linux system in eMMC will be started.

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

Before starting to burn the image, it is necessary to ensure that the SPI Flash chip has been affixed to the development board, because the development board does not have SPI Flash chip when it is shipped, so it is necessary to purchase and weld it by itself. We suggest that the SPI Flash chip model be XM25QU128CWIQT08Q

eMMC has a higher boot priority than NVMe SSDS. Therefore, you need to clear the eMMC system before burning the image. For details about how to clear eMMC, see section "Clearing eMMC using RKDevTool"

Note that the Linux images here specifically refer to the images of Linux distributions such as Debian, Ubuntu, and OPi OS Arch downloaded from the Orange Pi's data download page.

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

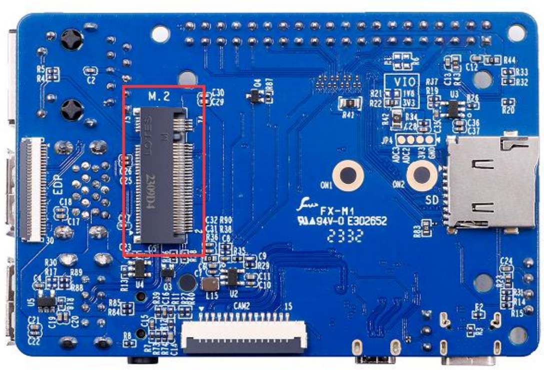
2. 6. 1. Method of burning using RKDevTool

1) First of all, you need to prepare an NVMe SSD. The PCIe supported by the M.2 slot on the development board is PCIe2.0x1, and the theoretical maximum speed is 500MB/s. Pci 3.0 and PCI 4.0 NVMe SSDS are also available, but the speed is only up to the speed of PCI 2.0x1.

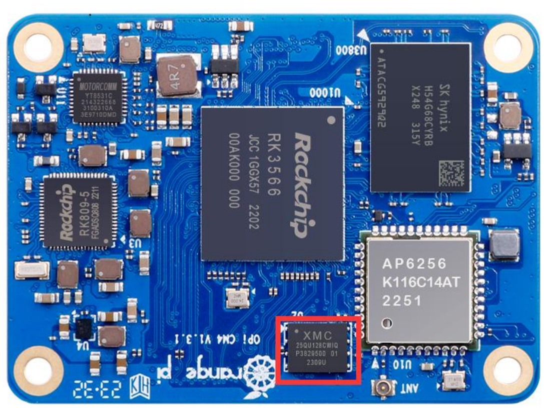
a. M.2 2230 SSD specifications are as follows

b. M.2 2242 SSDS are described as follows

2) Insert the NVMe SSD into the M.2 PCIe port on the development board and secure it



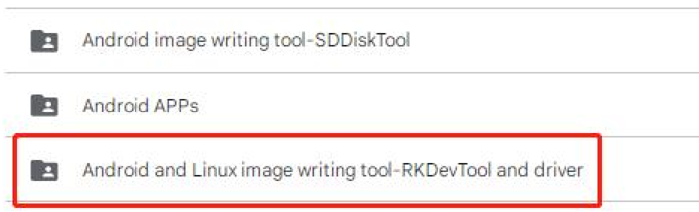
3) Please ensure that the SPI Flash is affixed to the development board. The position of SPI Flash on the development board is shown in the figure below. No other Settings are required before burning



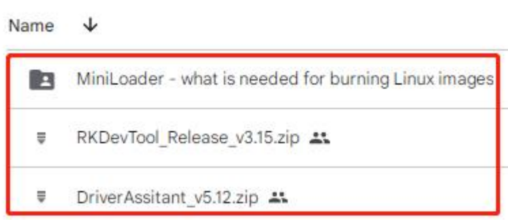
4) Then you need to prepare a good quality USB2.0 public-to-public data cable

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

a. On the **Orange Pi's data download page**, first select the **official tool** and then go to the following folder



b. Then download all the files below



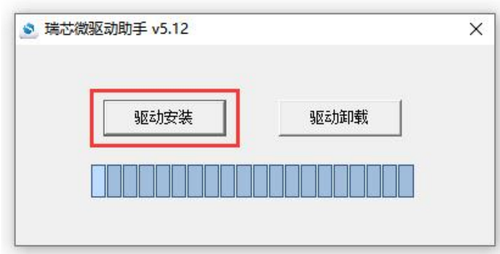
Note that the folder "MiniLoader- What you Need to burn Linux images" is hereinafter referred to as the MiniLoader folder.

6) Download the compressed Linux operating system image file you want to burn from the **Orange Pi's data download page**, and then use the decompression software to decompress it. In the decompressed file, the file ending with "**.img** "is the image file of the operating system, and the size is generally more than 2GB

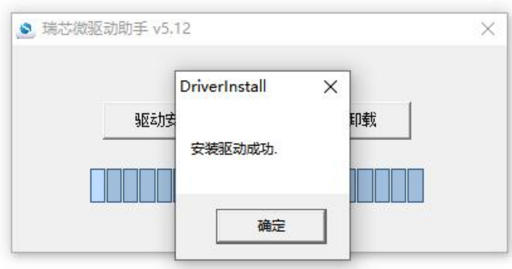
7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

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

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

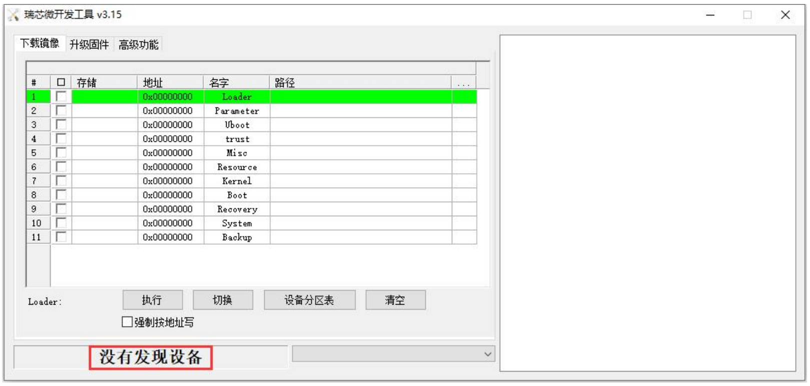


b. Wait for a period of time, the pop-up window will prompt "**Driver installation successful**", and then click the "**OK**" button



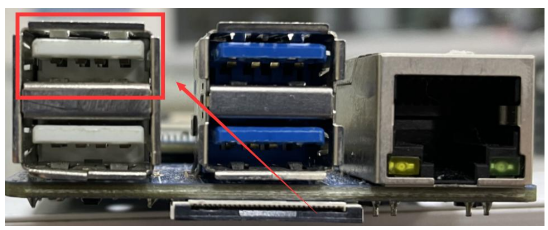
9) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, find **RKDevTool** in the decompressed folder and open it

10) After opening the **RKDevTool** burning tool, because the computer has not connected to the development board through the USB2.0 public-to-public data cable at this time, the lower left corner will indicate "**No device found**".



11) Then start burning the Linux image to the SSD

a. First, connect the development board to the Windows computer through the USB2.0 male-to-public data cable. The position of the USB2.0 burning interface on the development board is shown in the figure below



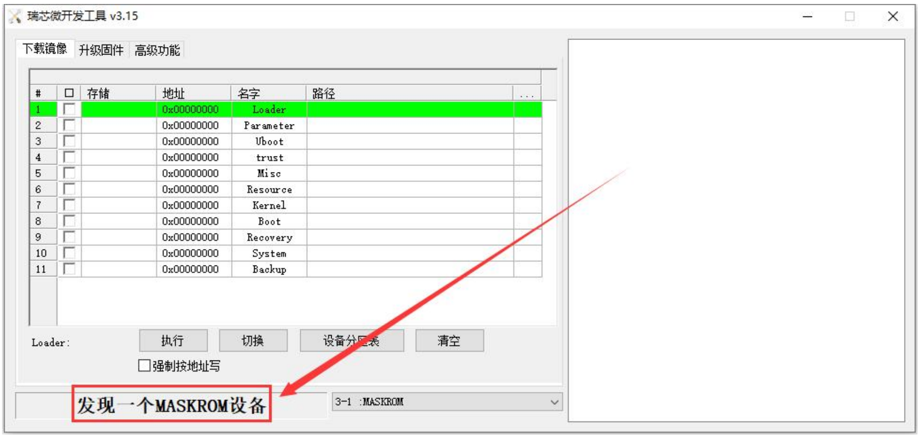
b. Ensure that no TF card is inserted into the development board and no power supply is connected

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

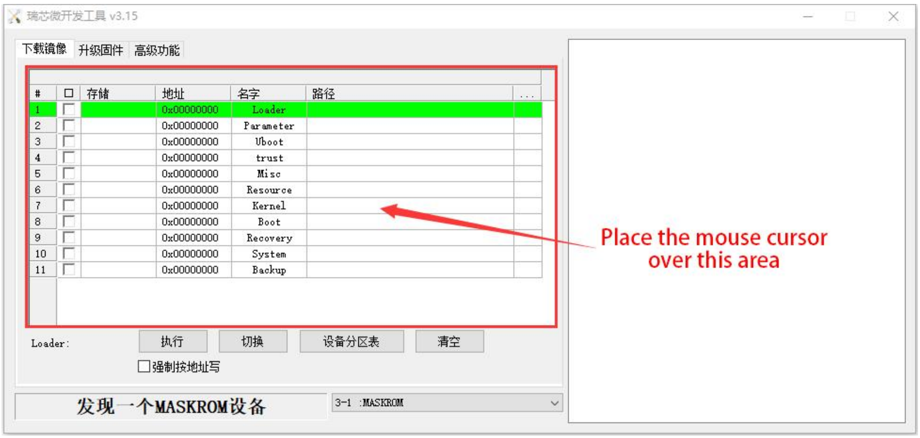
d. Then connect the Type-C interface to the development board, and power on, and then you can release the MaskROM button

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

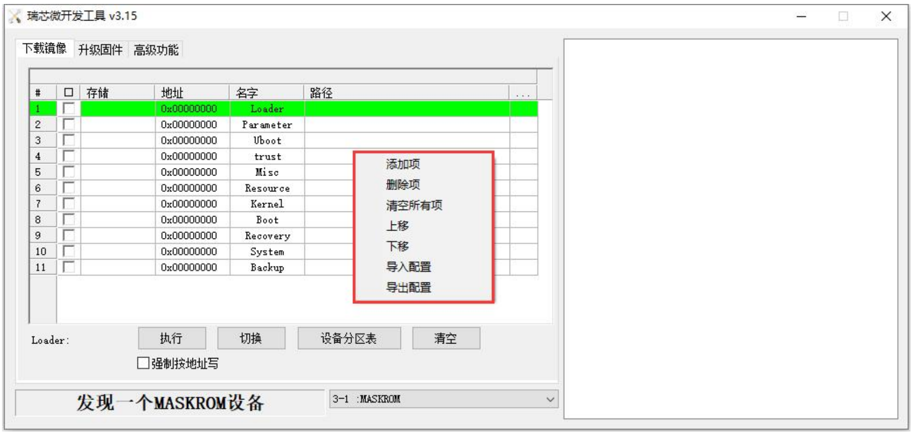
MASKROM device".



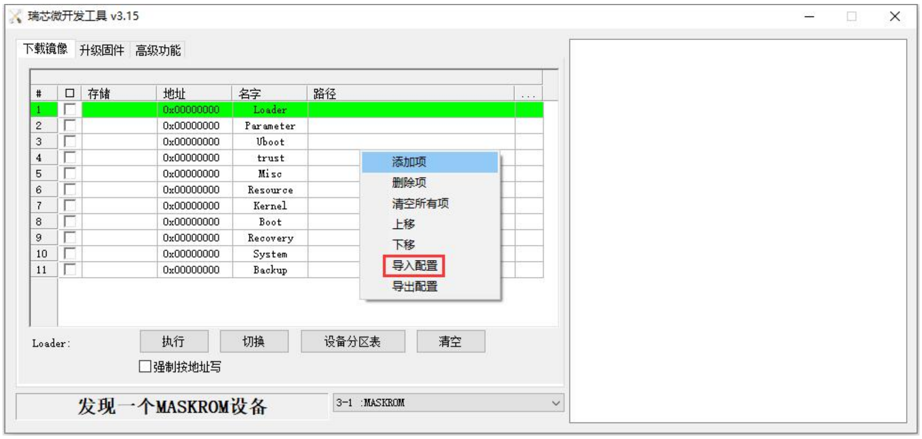
f. Then place the mouse cursor on the field below



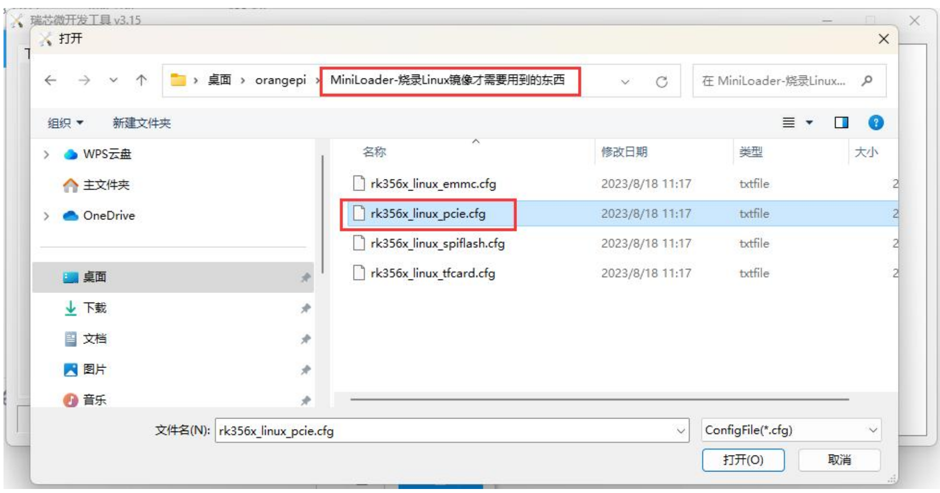
g. Then click the right mouse button to pop up the selection interface as shown in the following figure



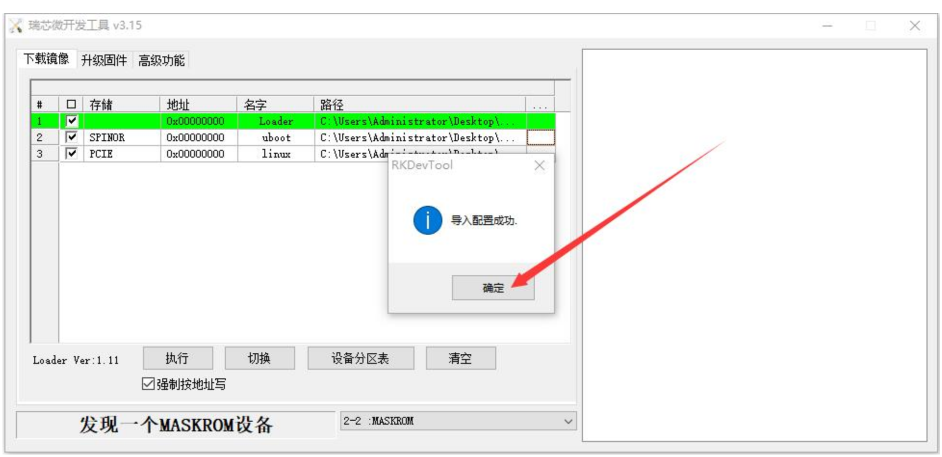
h. Then select the **Import configuration** option



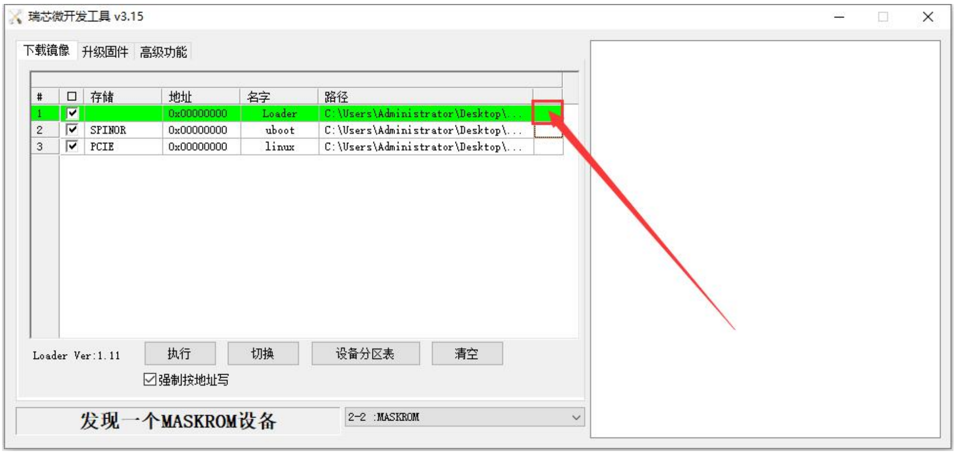
i. Then go to the previously downloaded **MiniLoader** folder, select the **rk356x_linux_pcie.cfg** configuration file, and click **Open**



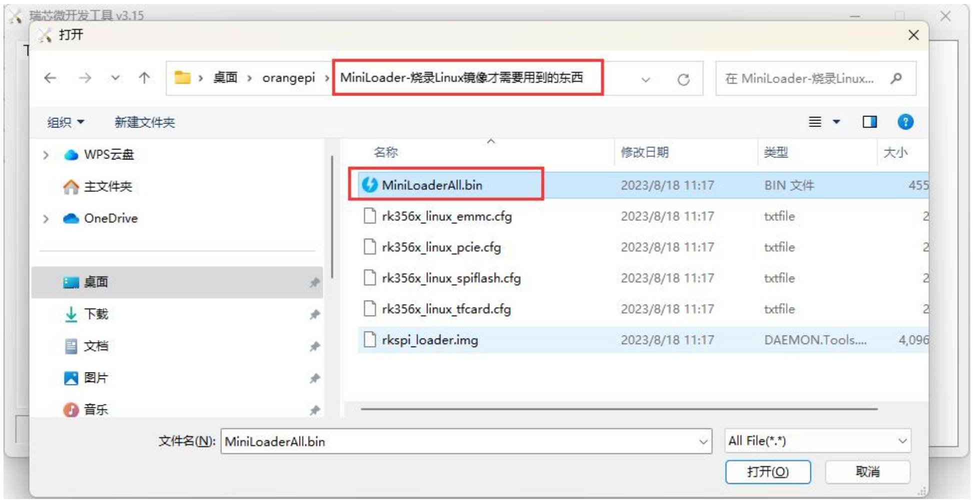
j. Then click **OK**



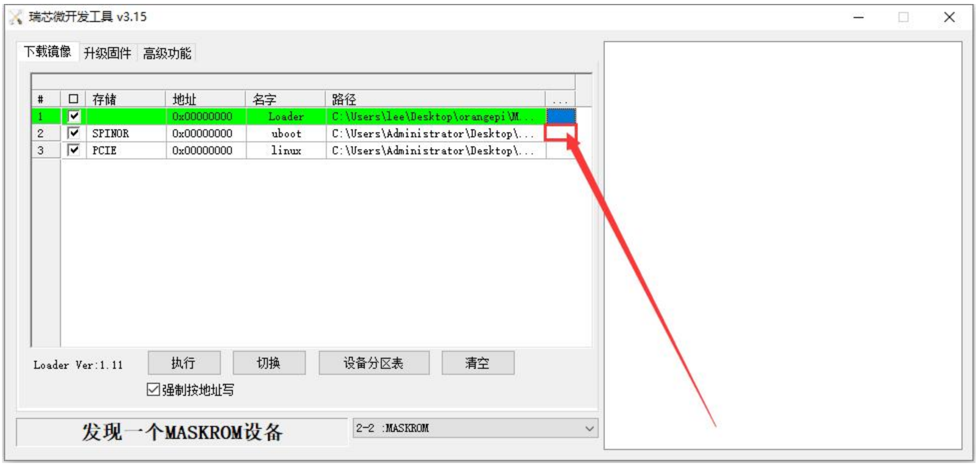
k. Then click on the location shown in the image below



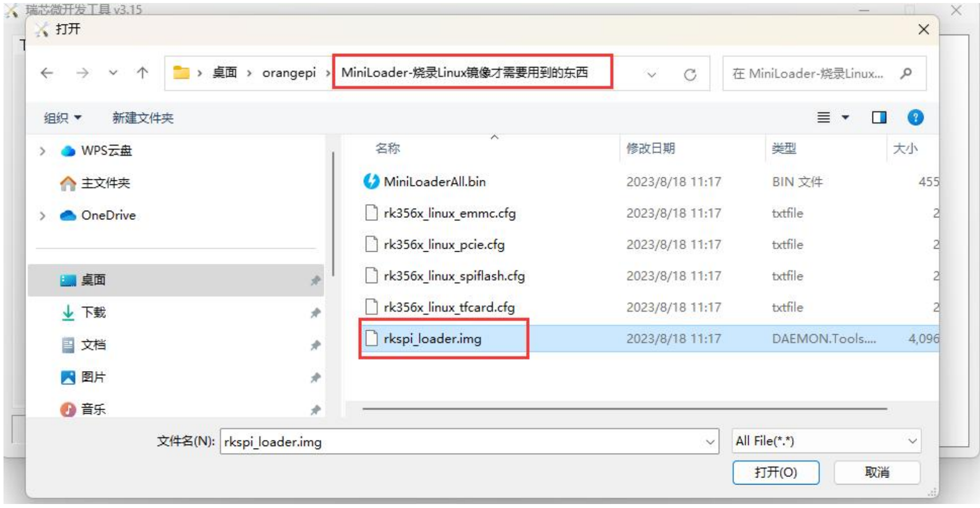
l. Select **MiniLoaderAll.bin** from the **MiniLoader** folder downloaded earlier, and then click **Open**



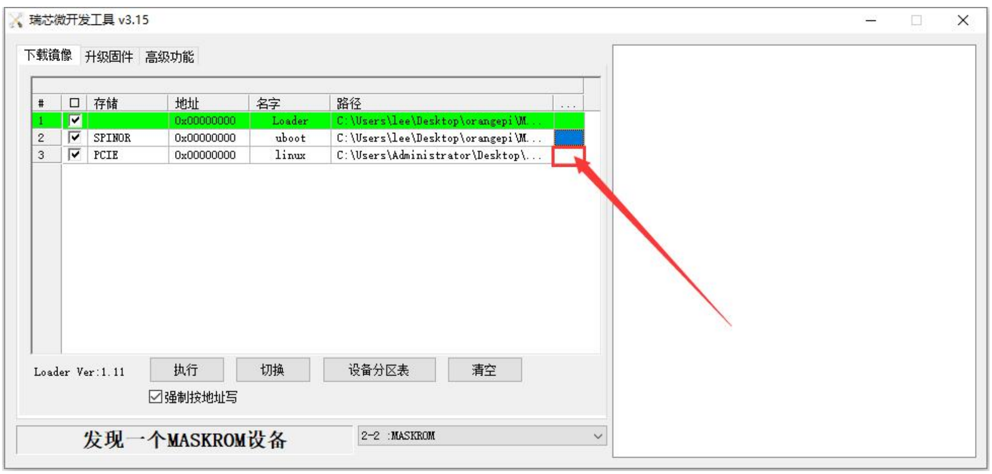
m. Then click on the location shown in the image below



n. Then go to the previously downloaded **MiniLoader** folder, select **rkspi_loader.img**, and click **Open**

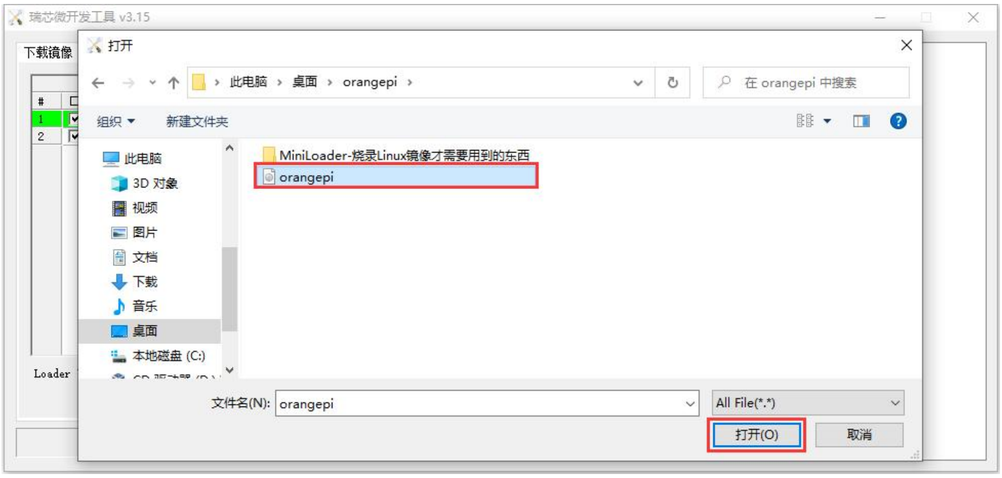


o. Then click on the location shown in the image below

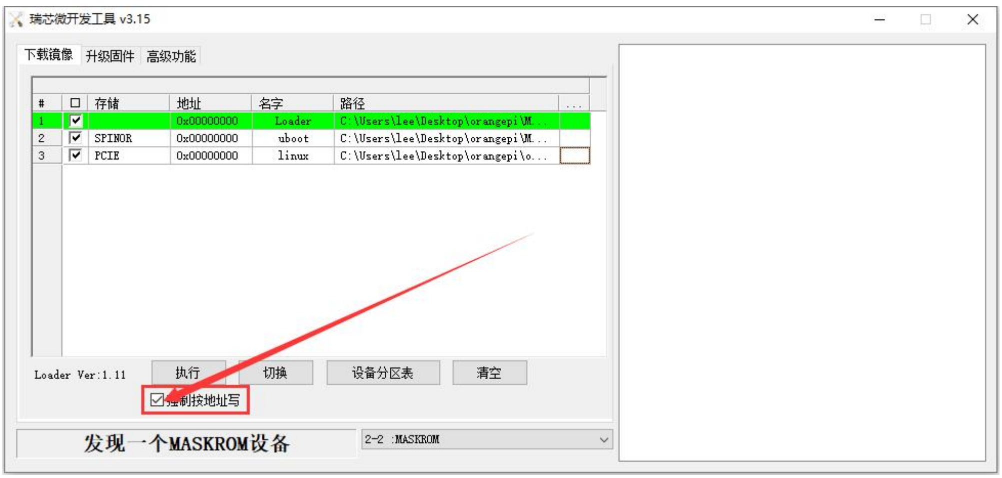


p. Then select the path of the Linux image you want to burn and click **Open**

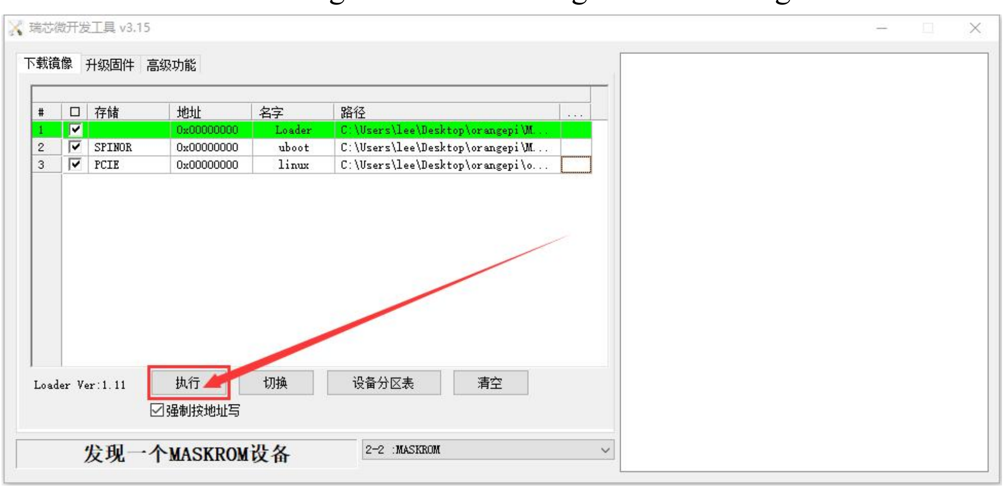
Before burning the image, it is recommended to rename the Linux image to orangepi.img or something shorter so that you can see the percentage of the burn progress when burning the image.



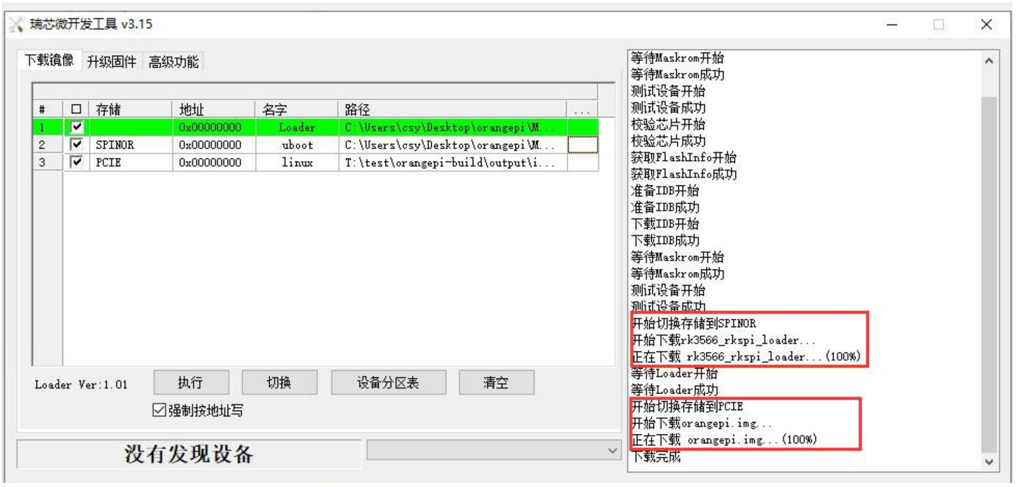
q. Then select **Force Write to address**



r. Click the Run button again to start burning the Linux image to the SSD



s. The display log of the S.lux image after burning is shown in the following figure



If you have problems with burning, please clear the SPIFlash first and then burn the test. For instructions on how to clear SPIFlash, refer to the section on how to clear SPIFlash using RKDevTool.

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

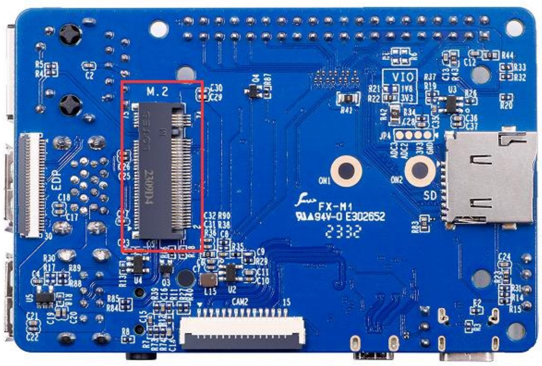
2. 6. 2. Method of burning using dd command

1) First of all, you need to prepare an NVMe SSD. The PCIe supported by the M.2 slot on the development board is PCIe2.0x1, and the theoretical maximum speed is 500MB/s. Pci 3.0 and PCI 4.0 NVMe SSDS are also available, but the speed is only up to the speed of PCI 2.0x1.

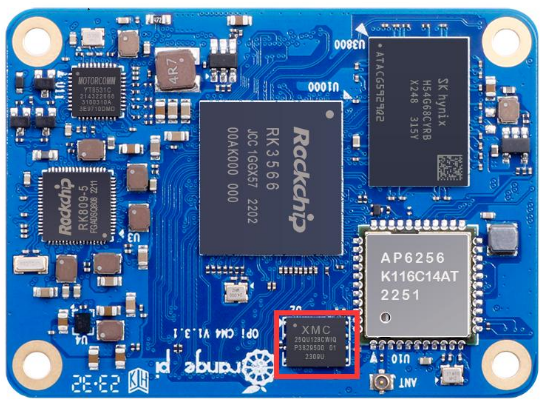
a. M.2 2230 SSDS are as follows

b. M.2 2242 SSDS are described as follows

2) Insert the NVMe SSD into the M.2 PCIe port on the development board and secure it



3) Please ensure that the SPI Flash is affixed to the development board. The position of SPI Flash on the development board is shown in the figure below. No other Settings are required before burning



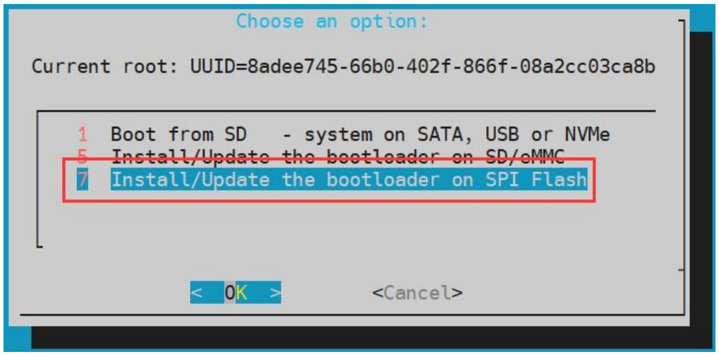
4) Burning Linux image to SPIFlash+NVMe SSD requires TF card to complete, so first need to burn Linux image to TF card, and then use the TF card to start the development board into the Linux system. For the method of burning a Linux image to a TF card, see the two sections on **the method of burning a Linux image to a TF card based on a Windows PC** and **the method of burning a Linux image to a TF card based on an Ubuntu PC.**

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

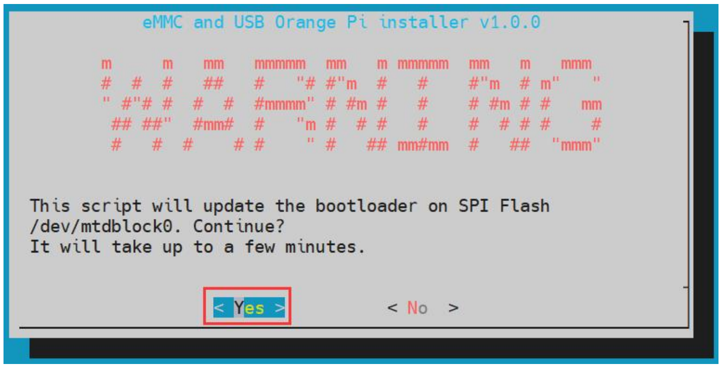
a. Run **nand-sata-install**l first. **Ordinary users remember to add sudo permission**

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

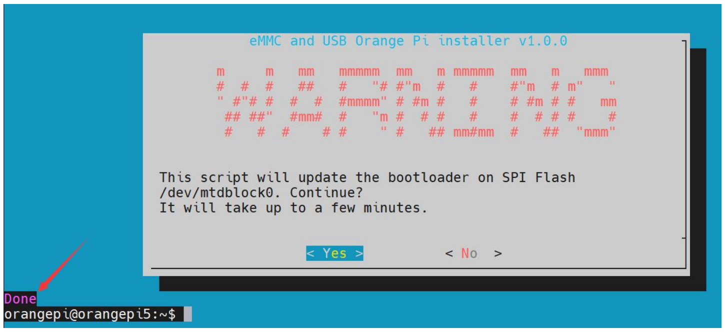
b. Then select **7 Install/Update ther bootloader on SPI Flash**



c. Then select **<Yes>**



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



6) Then upload the Linux image file (Debian or Ubuntu image downloaded from the official website) to the TF card. For details about how to upload a Linux image file to the development board, see section **Uploading Files to the Linux system on the Development Board**.

7) After uploading the image to the Linux system of the development board, we can enter the storage path of the image file in the command line of the Linux system of the development board. For example, I saved the Linux image of the development board in

the directory **/home/orangepi/Desktop**. Then go to the **/home/orangepi/Desktop** directory and you will see the uploaded image file.

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepi3b_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

How to enter the development board Linux system command line?

1. For details about how to log in to the terminal through the serial port, see section Debugging Serial Port.

2. For details about how to remotely log in to the Linux operating system over ssh, see section "Logging In to the Development Board over SSH".

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

8) Next, let's first make sure that NVMe SSD has been properly recognized by Linux development board. If the NVMe SSD is correctly identified, run the **sudo fdisk -l** command to view information about the **nvme**

```
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
00:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3566 (rev 01)
01:00.0 Non-Volatile memory controller: Realtek Semiconductor Co., Ltd. Device 5765 (rev 01)
```

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

```
orangepi@orangepi3b:~/Desktop$ sudo dd bs=1M if=/dev/zero of=/dev/nvme0n1 count=2000 status=progress
orangepi@orangepi3b:~/Desktop$ sudo sync
```

10) You can then use the dd command to burn the Linux image of the development board into the NVMe SSD

a. if = parameter in the command is to fill in behind the Linux + Linux mirror image to store the full path name (such as**/home/orangepi/Desktop/Linux image name**). Since we have entered the path of the Linux image above, we only need to fill in the name of the Linux image.

b. Do not copy the Linux image name in the following command. Replace it with the actual image name (because the version of the image may be updated).

```
sudo dd bs=1M if=Orangepi3b_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/nvme0n1 status=progress

sudo sync
```

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

A detailed description of all the parameters of the dd command and more usage can be seen by running the man dd command on Linux.

11) After successfully burning the Linux image of the development board to the NVMe SSD, you can use the **poweroff** command to shut down. Then, pull out the TF card, press the power button again, and then start the Linux system in the SPIFlash+NVMe SSD.

12) After starting the system in the NVMe SSD, run the **df -h** command to view 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
```

13) When the TF card and NVMe SSD burn exactly the same system, **if the TF card and NVMe SSD are inserted in the development board, power on the development board at this time, u-boot will give priority to start the system in the TF card.** However, since the TF card and the NVMe SSD system are identical, the UUID of the /**boot** partition and **rootfs** partition in the two storage devices are the same, which will cause the TF card may load the NVMe SSD partition when it is started. Run the following script to solve this problem.

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

An identical system means that the mirror name is exactly the same. Even if it is a Debian11 system, the version is different.

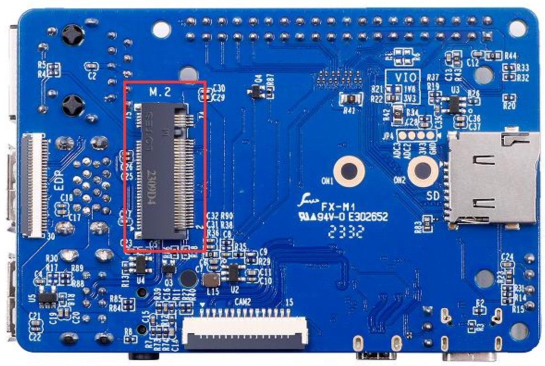
2. 6. 3. Method of burning using balenaEtcher software

1) First of all, you need to prepare an NVMe SSD. The PCIe supported by the M.2 slot on the development board is PCIe2.0x1, and the theoretical maximum speed is 500MB/s. Pci 3.0 and PCI 4.0 NVMe SSDS are also available, but the speed is only up to the speed of PCI 2.0x1.

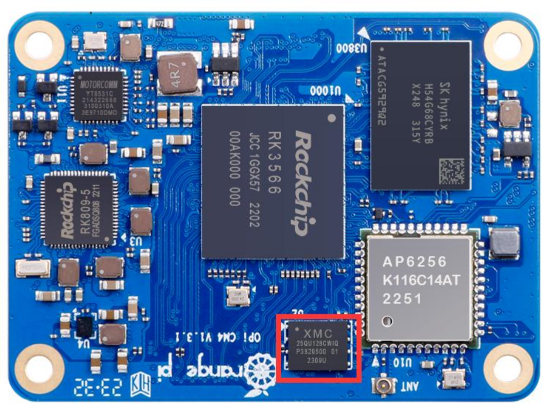
a. M.2 2230 SSDS are as follows

b. M.2 2242 SSDS are described as follows

2) Insert the NVMe SSD into the M.2 PCIe port on the development board and secure it



3) Please ensure that the SPI Flash is affixed to the development board. The position of SPI Flash on the development board is shown in the figure below. No other Settings are required before burning



4) Burning Linux image to SPIFlash+NVMe SSD requires TF card to complete, so first need to burn Linux image to TF card, and then use the TF card to start the development board into the Linux system. For the method of burning a Linux image to a TF card, see the two sections on **the method of burning a Linux image to a TF card based on a Windows PC** and **the method of burning a Linux image to a TF card based on an Ubuntu PC.**

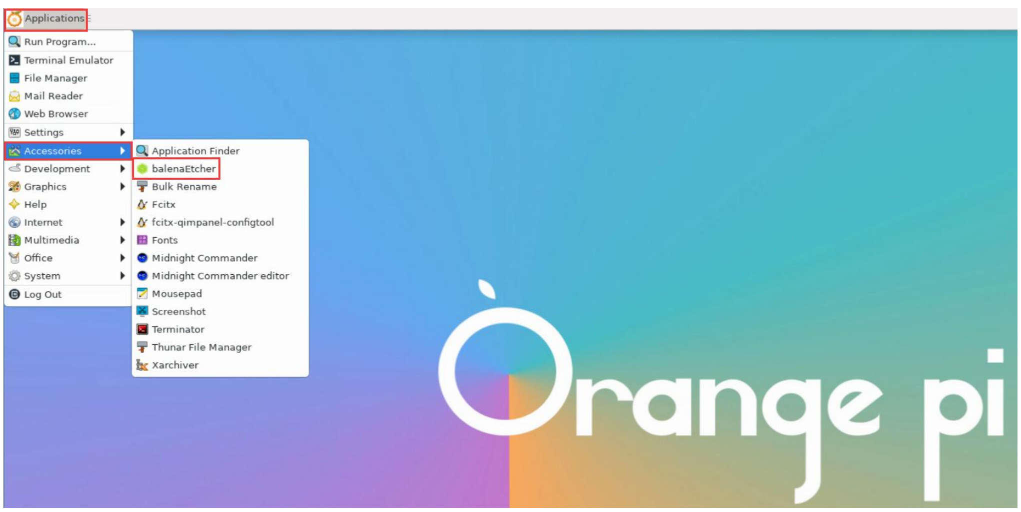
5) After starting the Linux system in the TF card, please make sure that the NVMe SSD has been properly recognized by Linux on the development board. If the NVMe SSD is correctly identified, run the **sudo fdisk -l** command to view information about the **nvme**

```
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

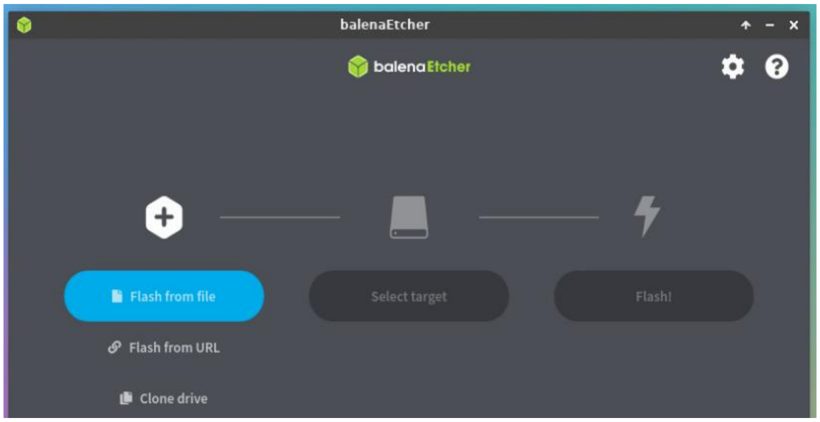
```
orangepi@orangepi:~/Desktop$ lspci
00:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3566 (rev 01)
01:00.0 Non-Volatile memory controller: Realtek Semiconductor Co., Ltd. Device 5765 (rev 01)
```

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



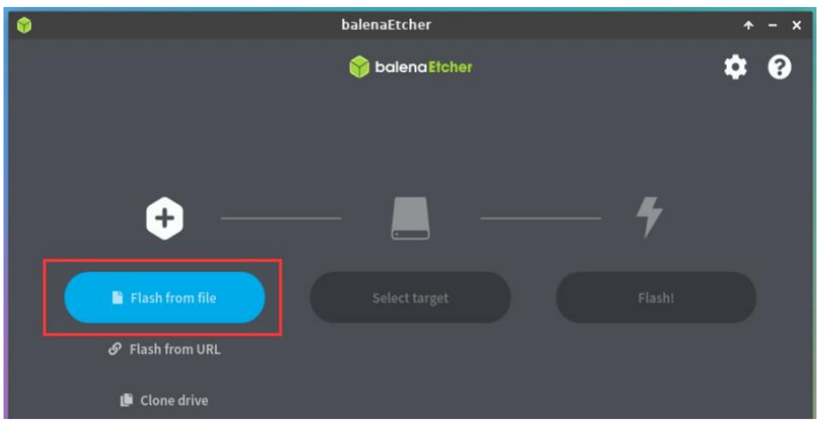
If the balenaEtcher arm64 version is not pre-installed, for details about how to download and install the balenaEtcher arm64 version, see section How to Download and Install the balenaEtcher arm64 version.

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

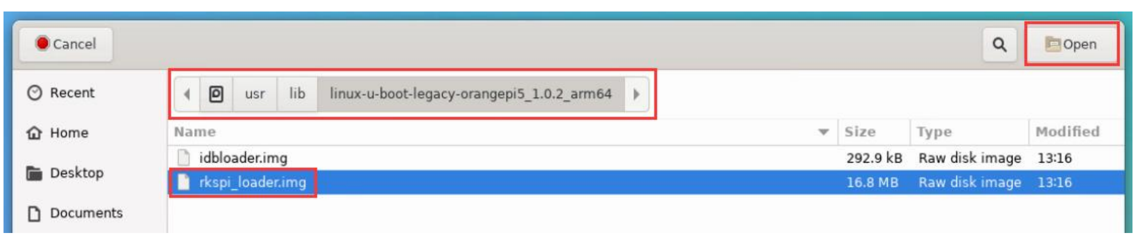


8) The method of using balenaEtcher to burn u-boot to SPI Flash on the development board is as follows:

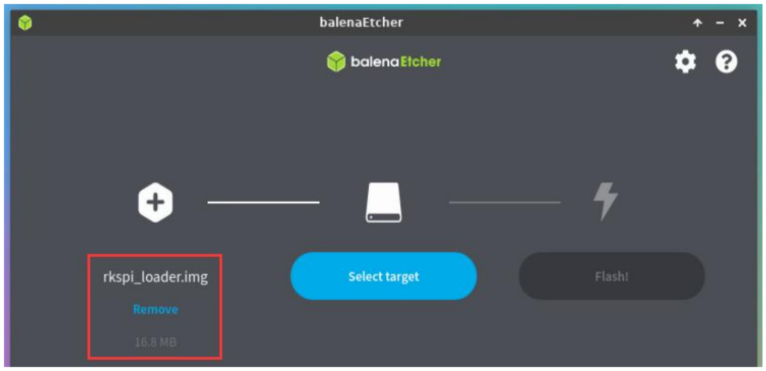
a. First click **Flash from file**



b. Go to **/usr/lib/linux-u-boot-legacy-orangepi3b_1.x.x_arm64**, select **rkspi_loader.img**, and click **Open**

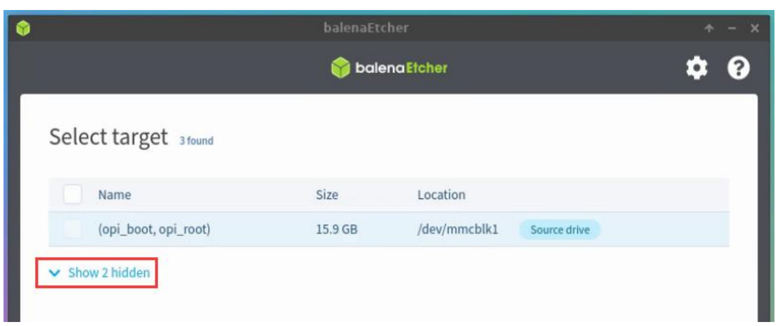


c. Open **rkspi_loader.img**. The following interface is displayed:

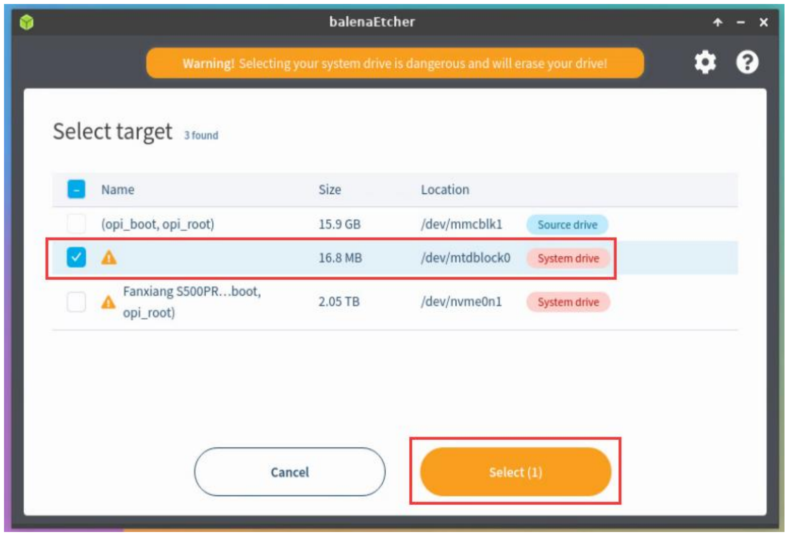


d. Then click **Select target**

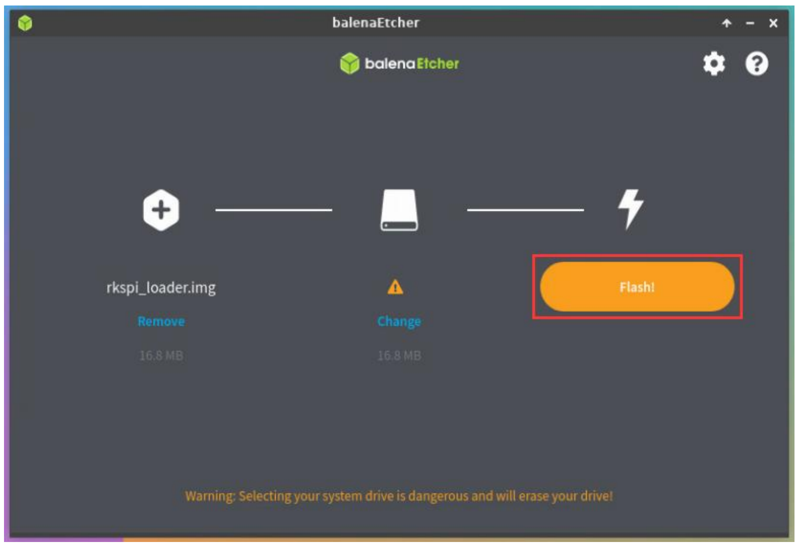
e. Then click **Show 2 hidden** to open the option for more storage devices



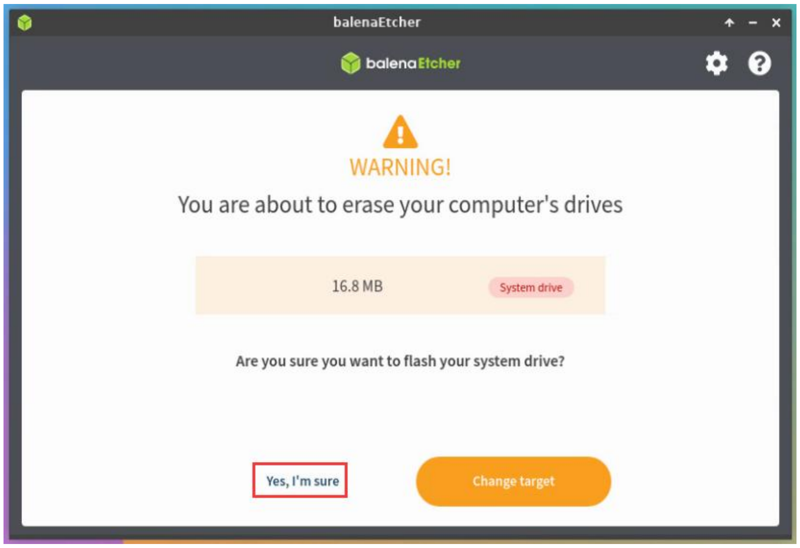
f. Then Select the SPI Flash device name **/dev/mtdblock0** and click **Select**



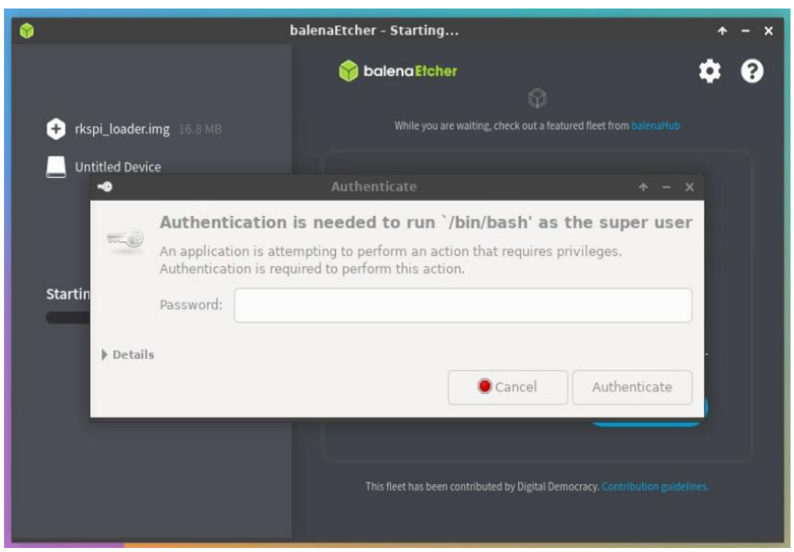
g. Then click **Flash**



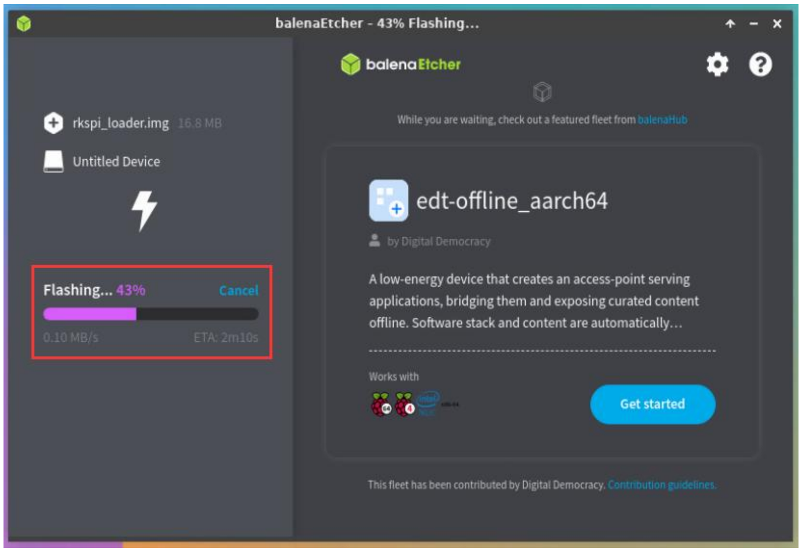
h. Then click **Yes, I'm sure**



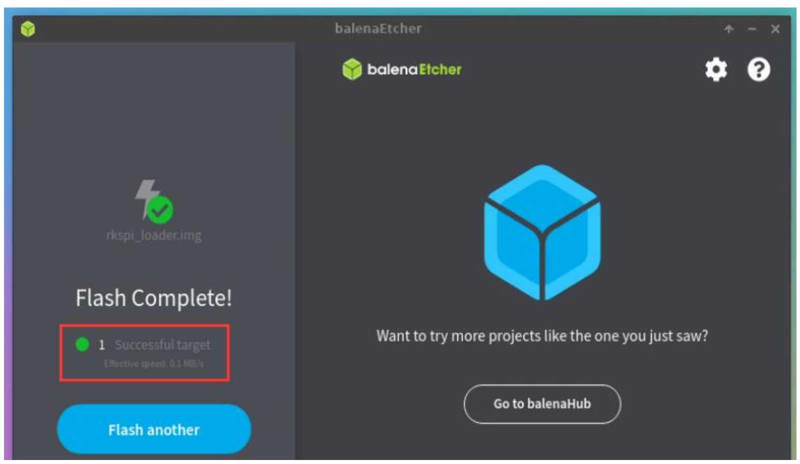
i. Then enter the password **orangepi** of the development board Linux system, and the u-boot image will be burned to the SPI Flash



j. The burning process is displayed as follows:

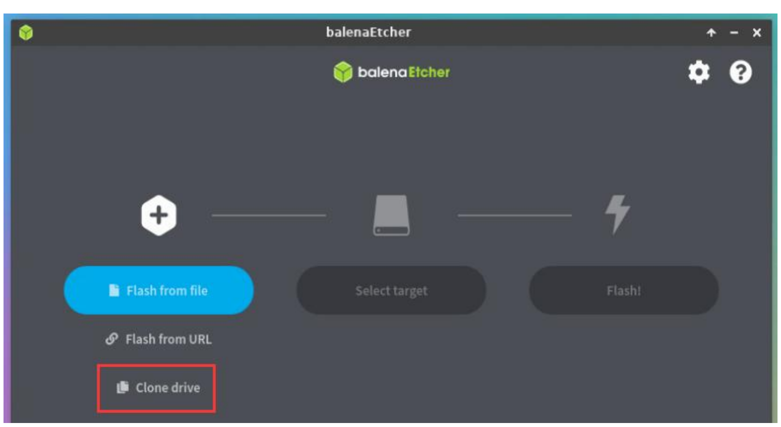


k. The display after burning is as follows:

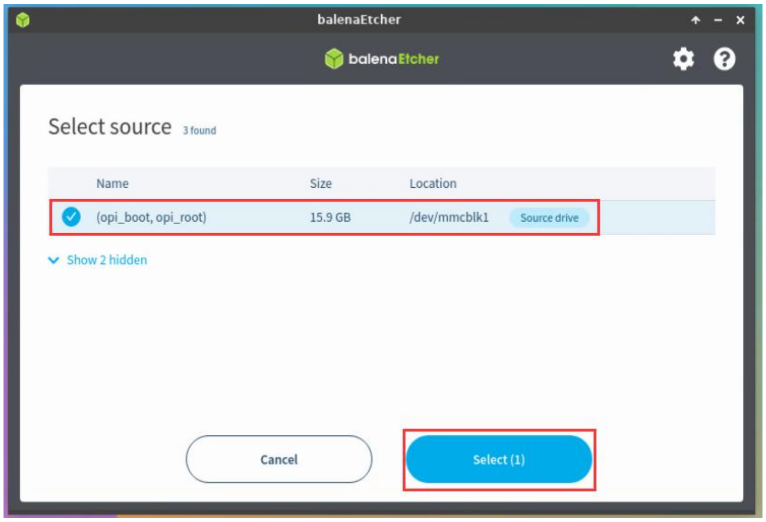


9) Method of burning the Linux system in the TF card to the NVMe SSD (this method is equivalent to cloning the system in the TF card to the NVMe SSD)

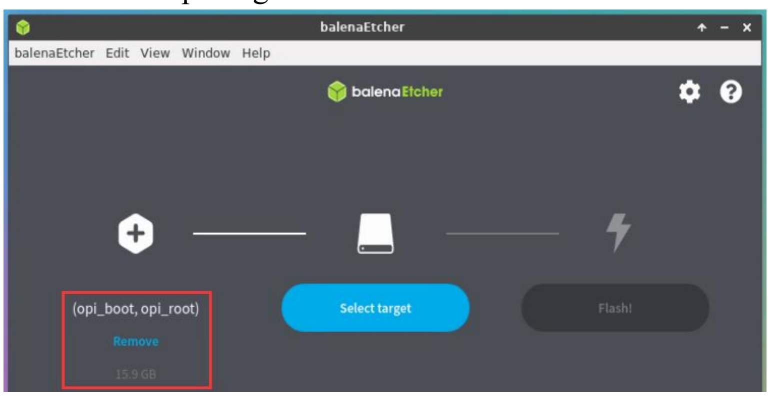
a. First click **Clone drive**



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

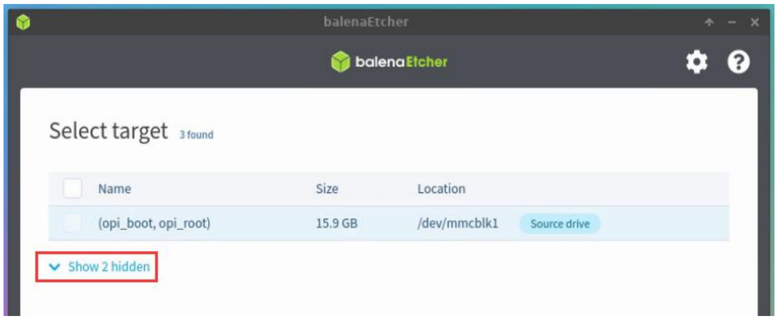


c. The interface after opening the TF card is as follows:

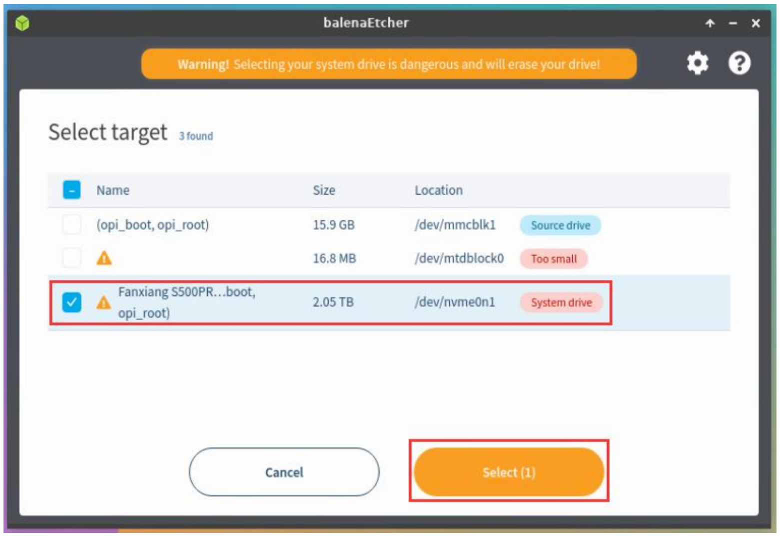


d. Then click **Select target**

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

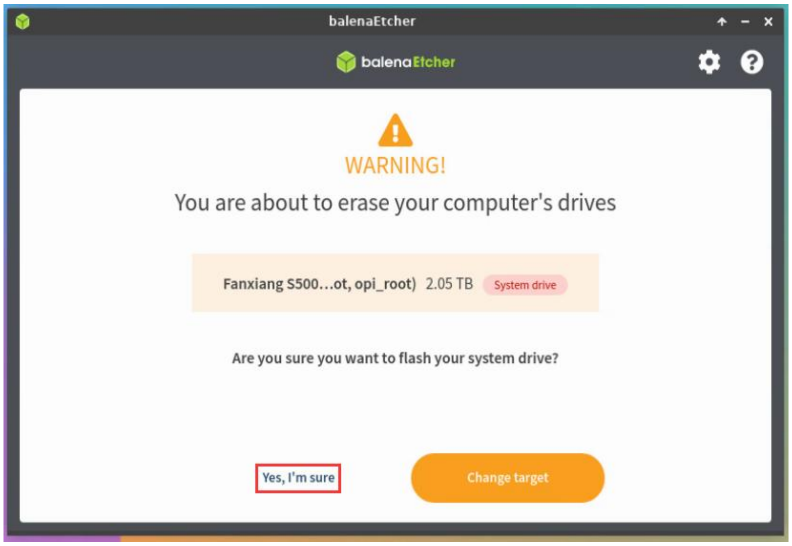


f. Then Select the device name for the NVMe SSD **/dev/nvme0n1** and click **Select**

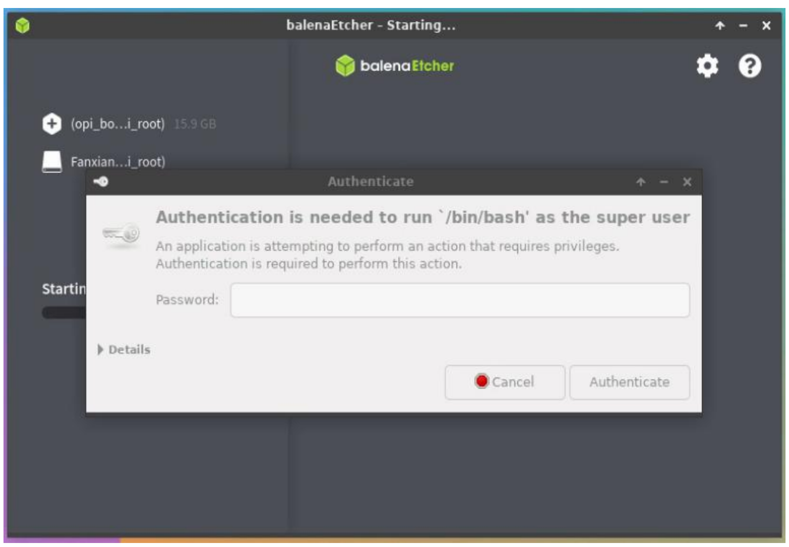


g. Then click **Flash**

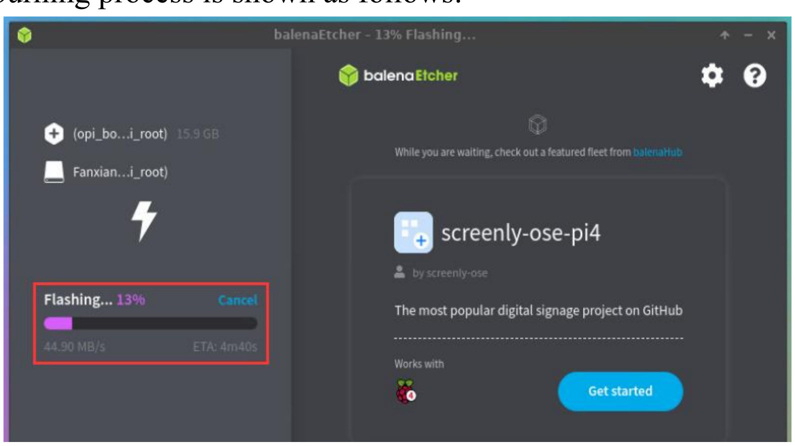
h. Then click **Yes, I'm sure**

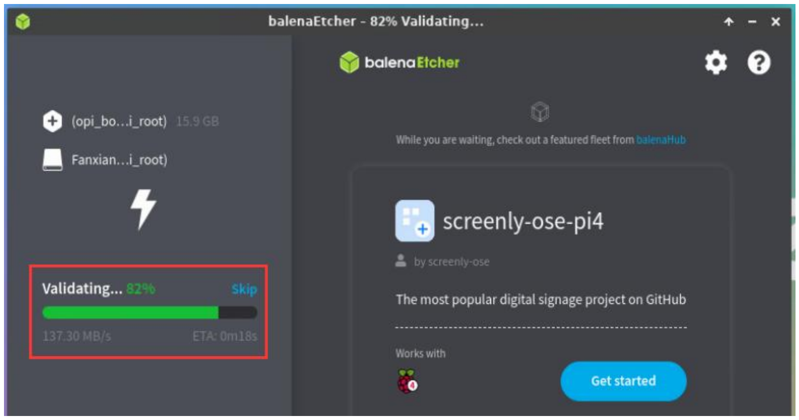


i. Then enter the password orangepi for the development board Linux system and start burning the Linux image to the SSD

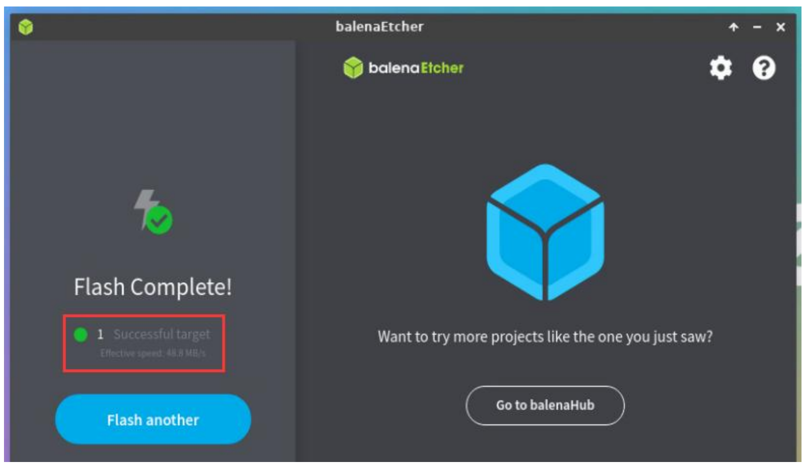


j. The burning process is shown as follows:





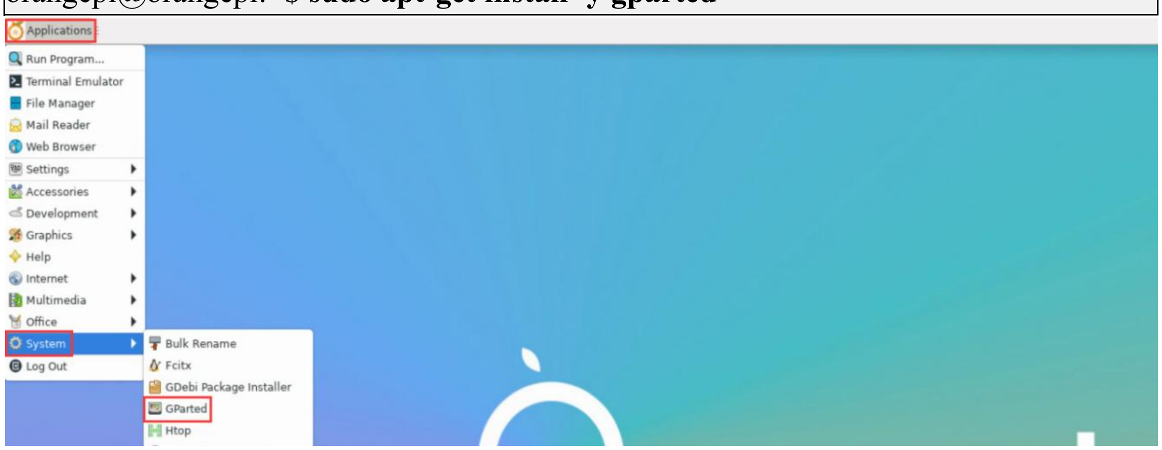
k. The display after burning is as follows:



l. To expand the capacity of the rootfs partition on the NVMe SSD, perform the following steps:

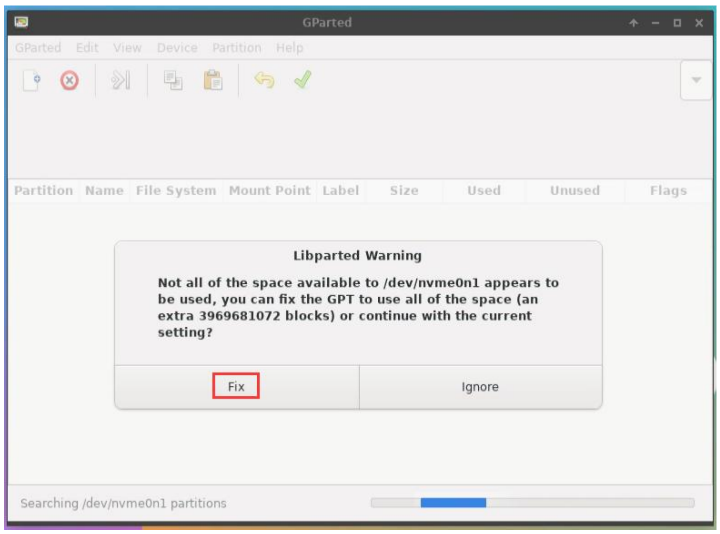
a) First, open **GParted**. If Gparted is not pre-installed, run apt command to install it

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

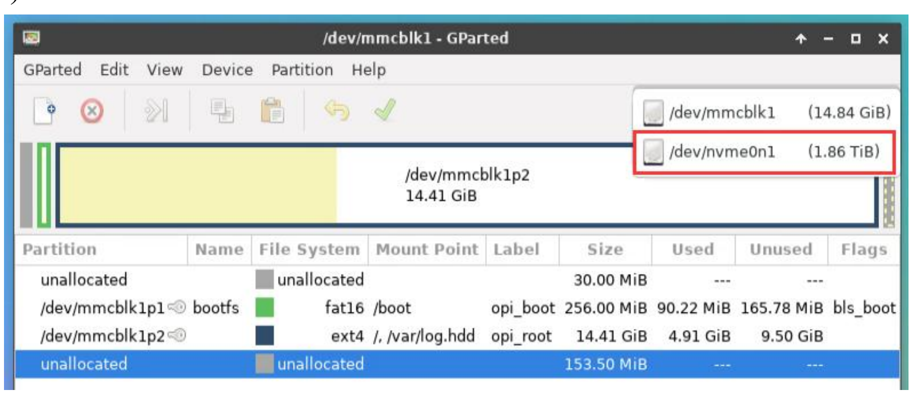


b) Then enter the password orangepi for Linux and click **Authenticate**

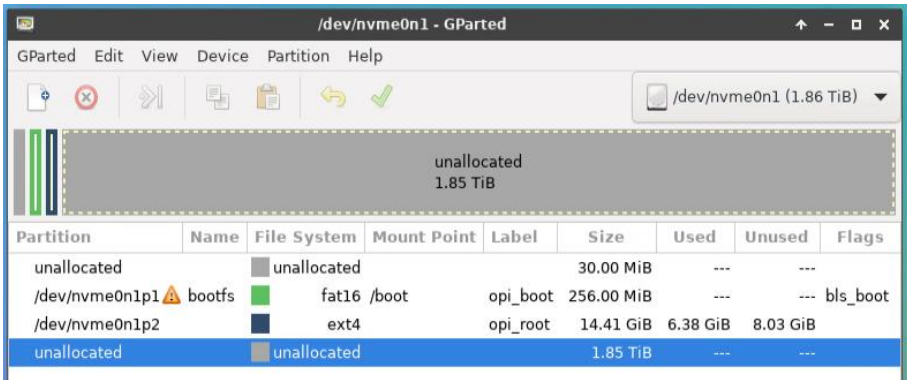
c) Then click **Fix**



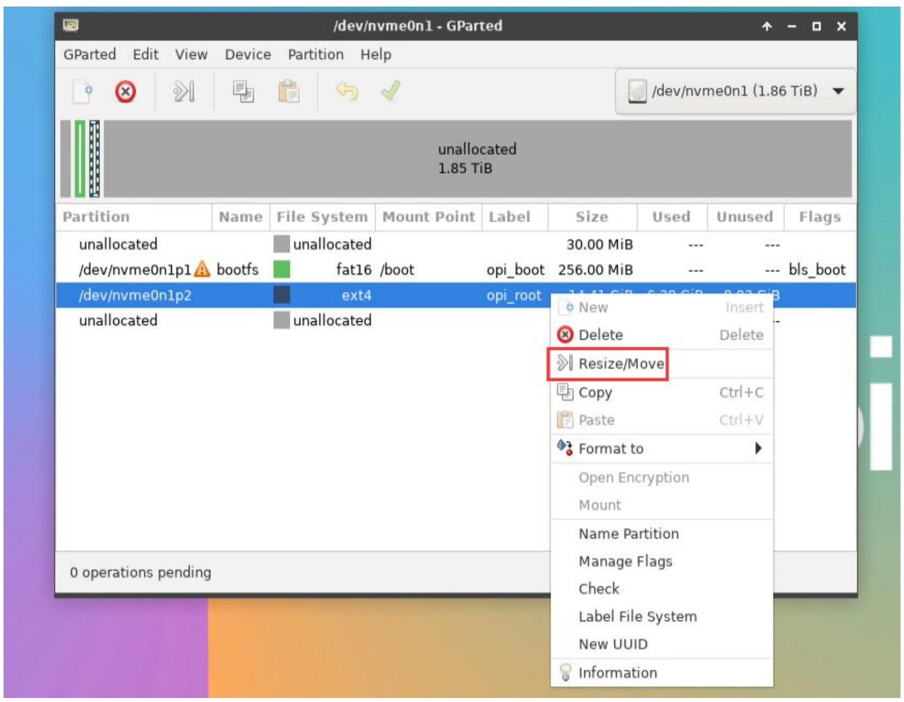
d) Then select NVMe SSD



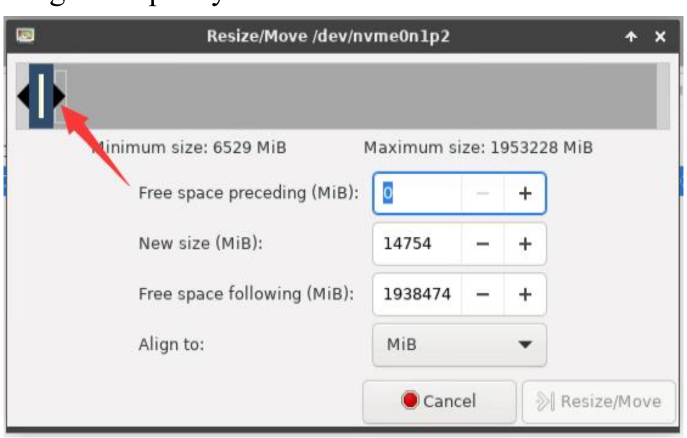
e) The following information is displayed after the NVMe SSD is selected:



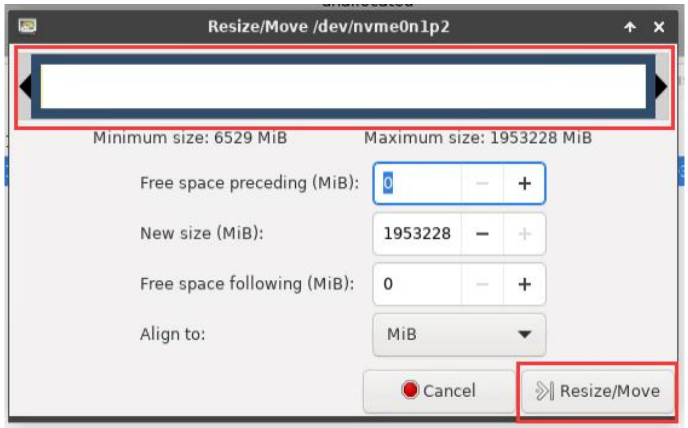
f) Then select the **/dev/nvme0n1p2** partition, right-click, and select **Resize/Move**



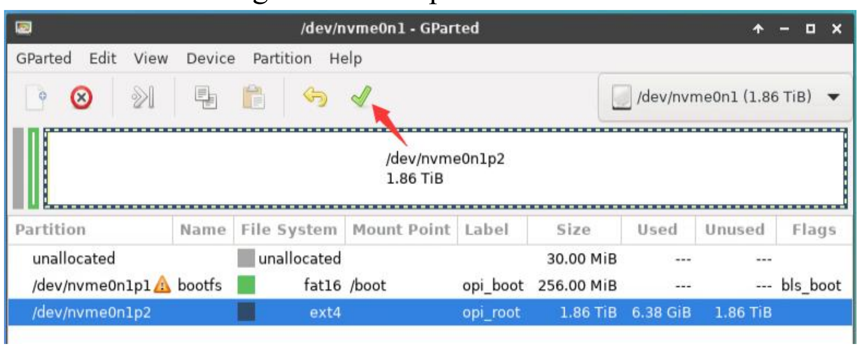
g) Then drag the capacity to the maximum as shown in the image below



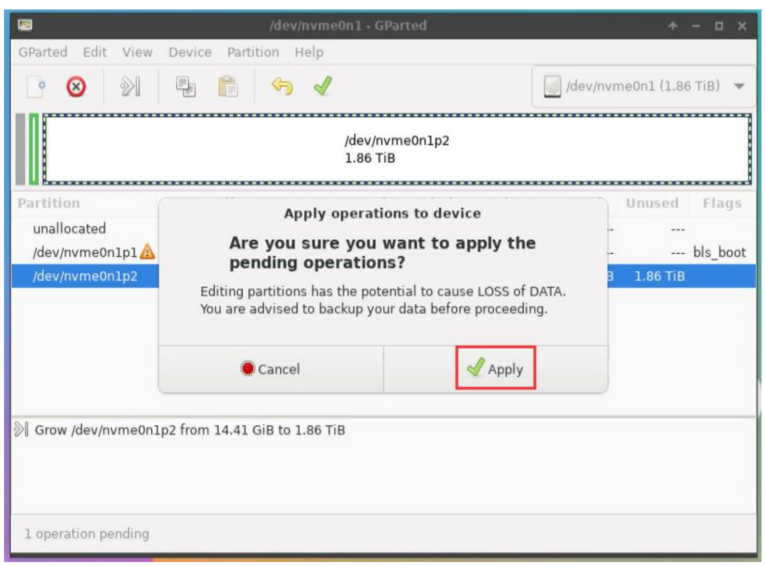
h) Then click **Resize/Move**



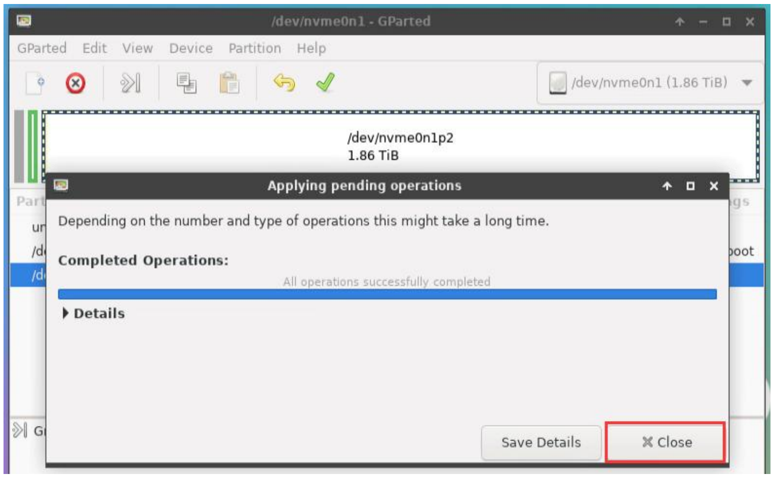
i) Then click on the green√ in the picture below



j) Then click **Apply**



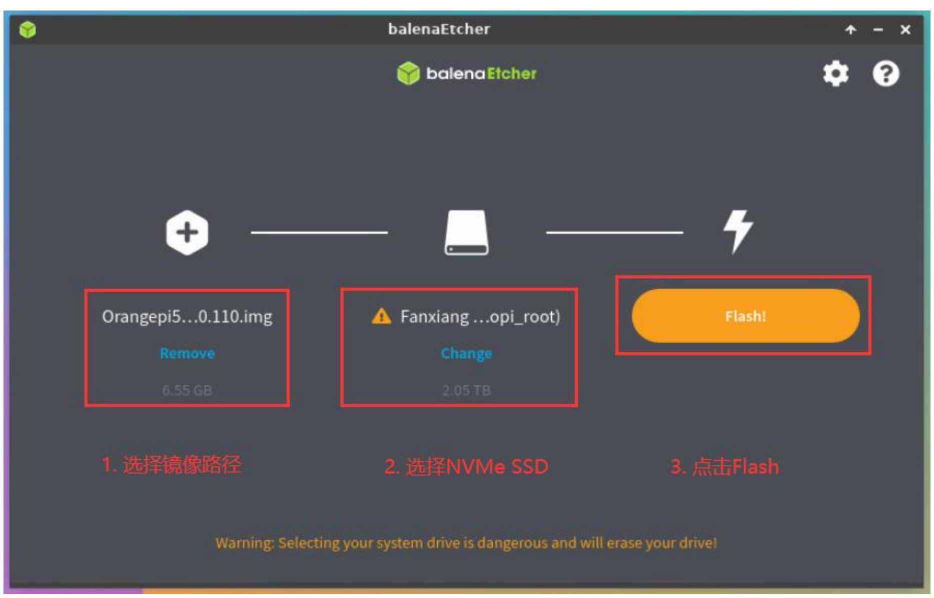
k) Then click **Close** to close it



m. At this point, you can shut it down using the **sudo poweroff** command. Then, pull out the TF card, press the power button again, and then start the Linux system in the SPIFlash+NVMe SSD.

10) The **9)** step is to clone the system in the TF card to the NMVe SSD, we can also directly burn the Linux image file to the NVMe SSD, here roughly say the following steps:

a. Upload the Linux image file to the Linux operating system on the development board
b. Then use balenaEtcher to burn



c. **There is no need to manually expand the capacity after burning the image using this method. The first startup will automatically expand the capacity.**

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

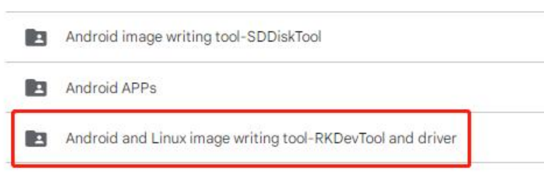
2. 7. 1. Method of burning Android image to TF card through USB2.0 burning port

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

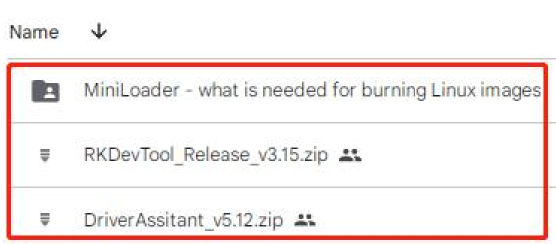
2) Also need to prepare a good quality USB2.0 public to public data cable

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

a. On the **Orange Pi's Data download page**, first select the official tool and then go to the following folder

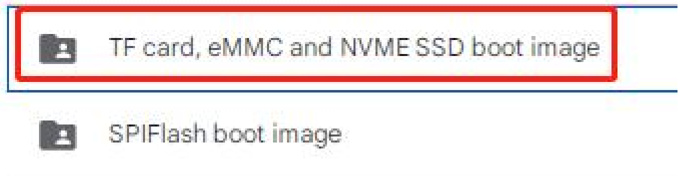


b. Then download all the files below



4) Then download the Android image from **Orange Pi's Data download page**.

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



b. After entering the **TF card and eMMC to start the image** folder, you can see the following two images, their differences are:

a) The first image is dedicated to HDMI display and supports 4K display, if you do not use LCD screen, please download the image without lcd

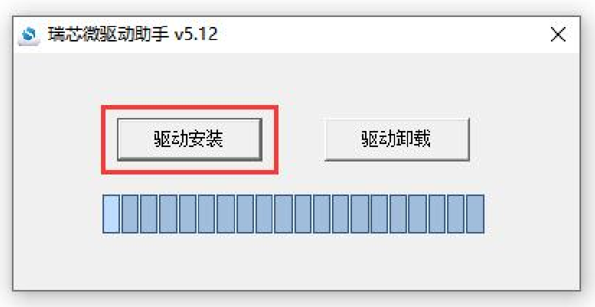
b) If you want to use an lcd screen, select an image with lcd

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

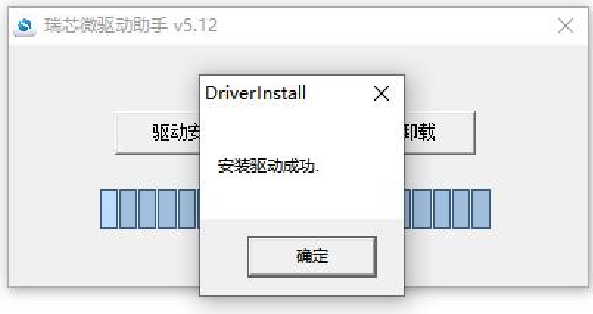
6) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

7) Open **DriverInstall.exe** and install the SWick micro driver as follows

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

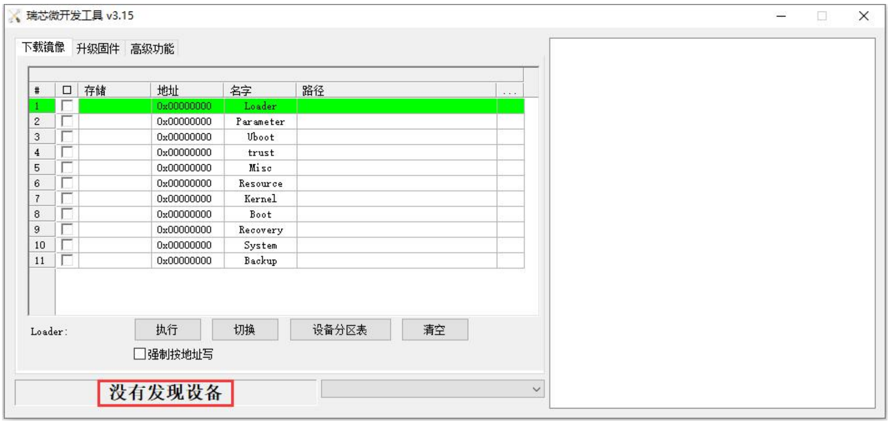


b. Wait for a period of time, the pop-up window will prompt "**Driver installation successful**", and then click the "**OK**" button



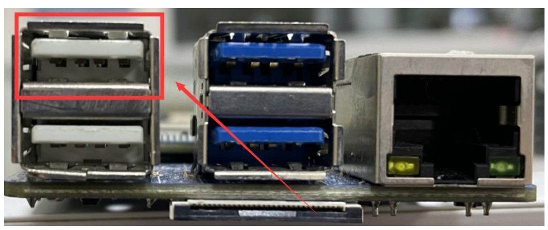
8) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, find **RKDevTool** in the decompressed folder and open it

9) After opening the **RKDevTool** burning tool, because the computer has not connected to the development board through the USB2.0 public-to-public data cable at this time, the lower left corner will display "**No device found**".



10) Then start burning Android image to TF card

a. First, connect the development board to the Windows computer through the USB2.0 male-to-public data cable. The position of the USB2.0 burning interface on the development board is shown in the figure below

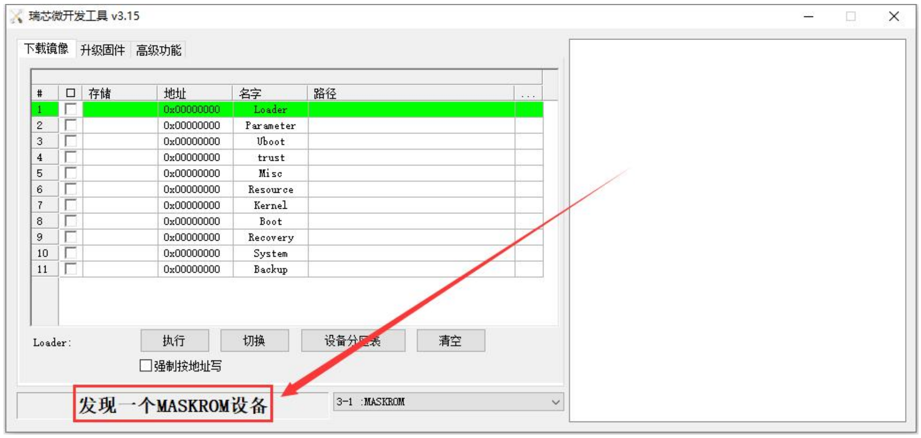


b. Then make sure that no TF card is inserted into the development board and no power is connected

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

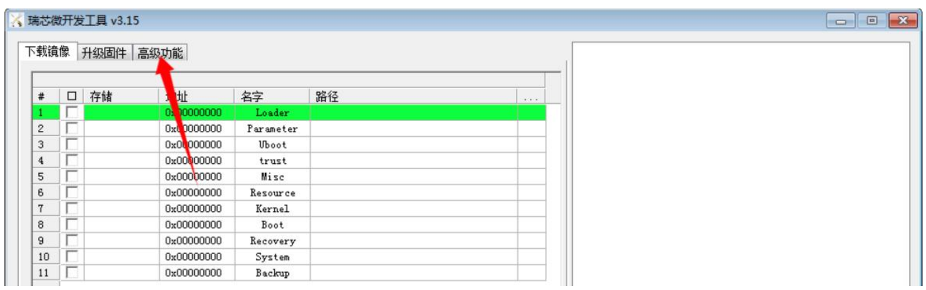
d. Connect the development board to the Type-C port, and power on the development board

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

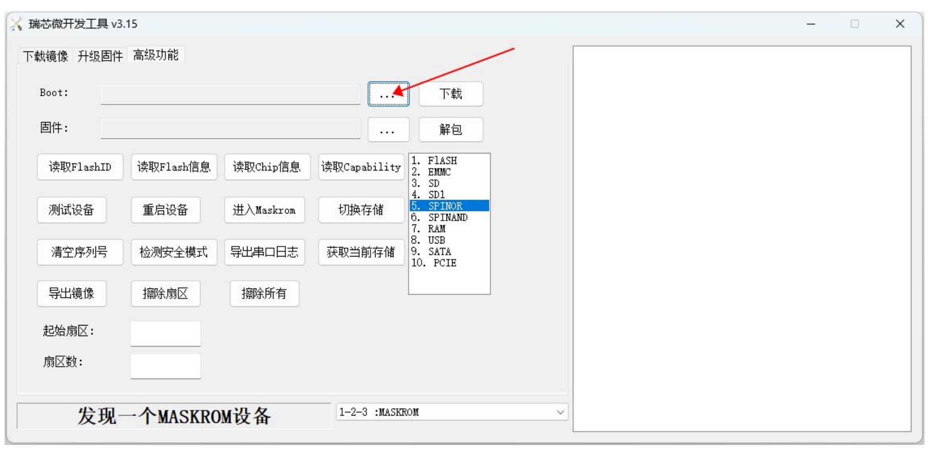


f. Then insert the TF card into the development board

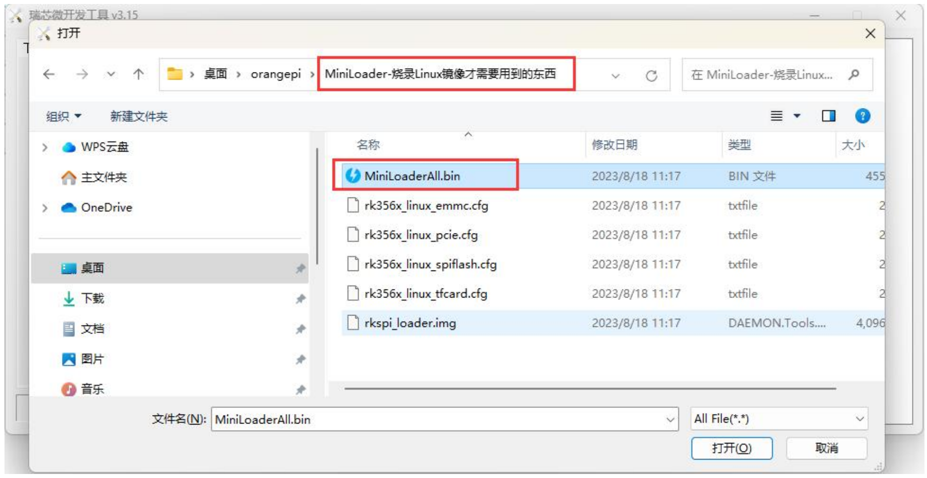
g. Then select **Advanced features**



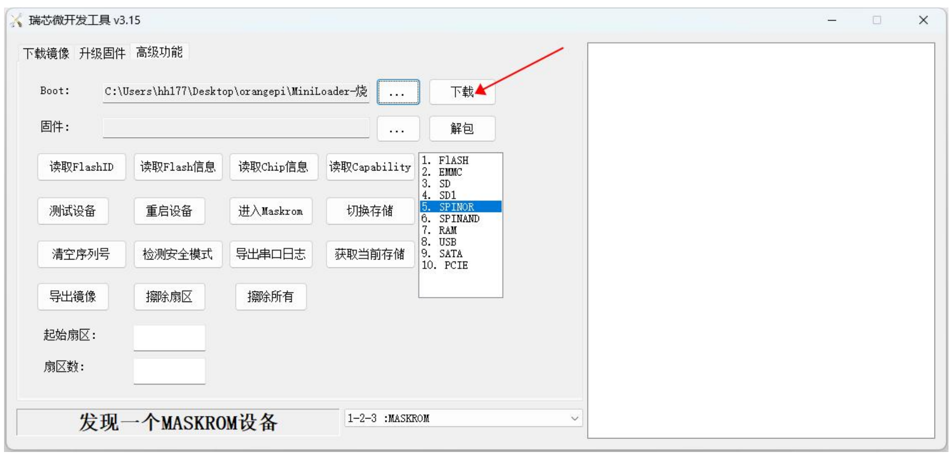
h. Then click on the location shown in the image below



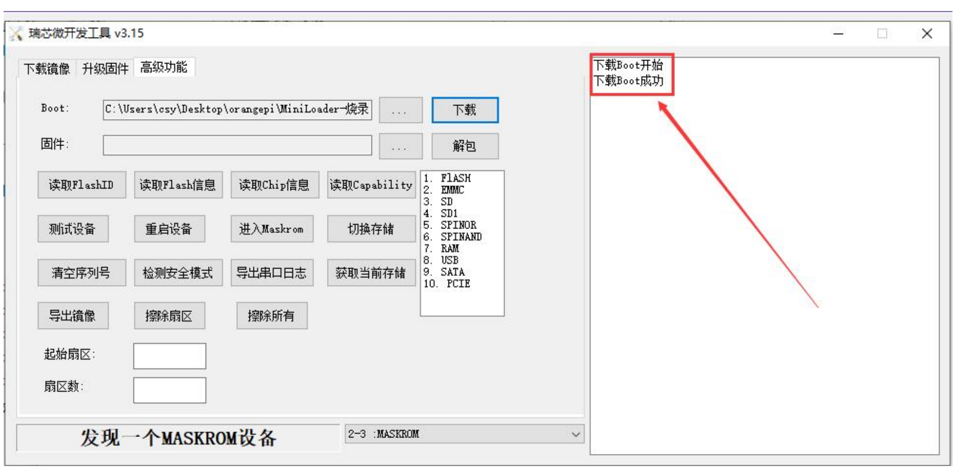
i. Select **MiniLoaderAll.bin** from the **MiniLoader** folder downloaded earlier and click Open



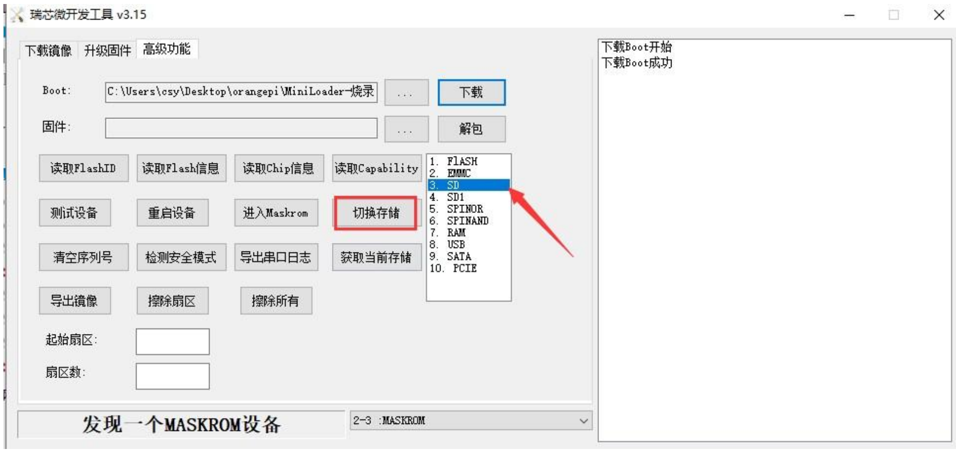
j. Then click **Download**



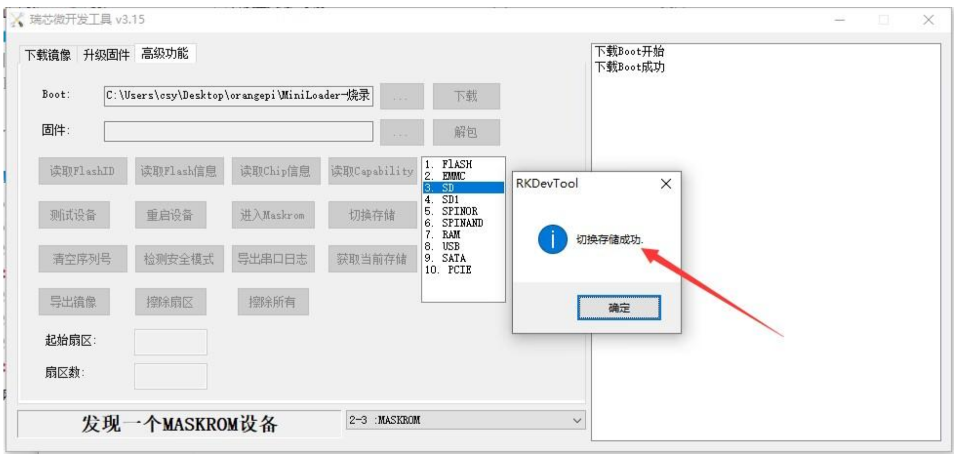
k. The following figure is displayed after **MiniLoaderAll.bin** is downloaded



l. Then select the storage device as **SD** and click **Switch storage**



m. If the switchover is successful, the following figure is displayed



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

o. Then click the "**Firmware**" button to select the path of the Android image that you want to burn

p. Finally, click the "**Upgrade**" button to start burning. The log in the burning process is shown below. The Android system will start automatically after the burning is complete.

2.8. Method of burning Android image into eMMC

2.8.1. Method of burning Android image into eMMC through USB2.0 burning port

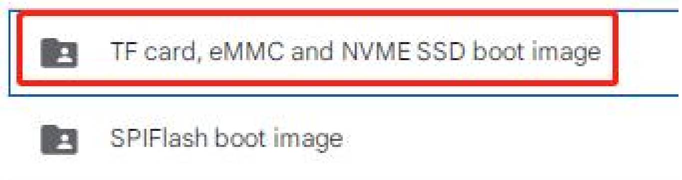
Note that all operations below are performed on a Windows computer.

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

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

3) Then download the Android image from the **Orange Pi download page**.

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



b. After entering the **TF card and eMMC boot image** folders, you can see the following two images. The difference between them is:

a) The first image is dedicated to HDMI display and supports 4K display. If you don't use LCD screen, please download the image without lcd

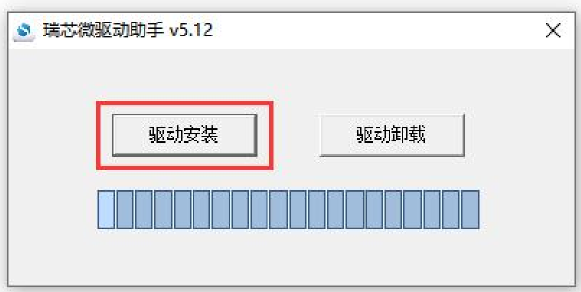
b) If you want to use an LCD screen, please choose the image with LCD

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

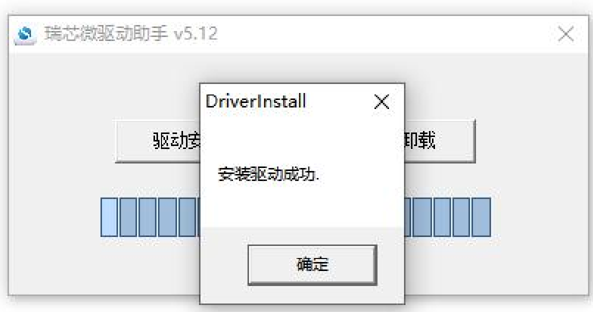
5) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

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

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

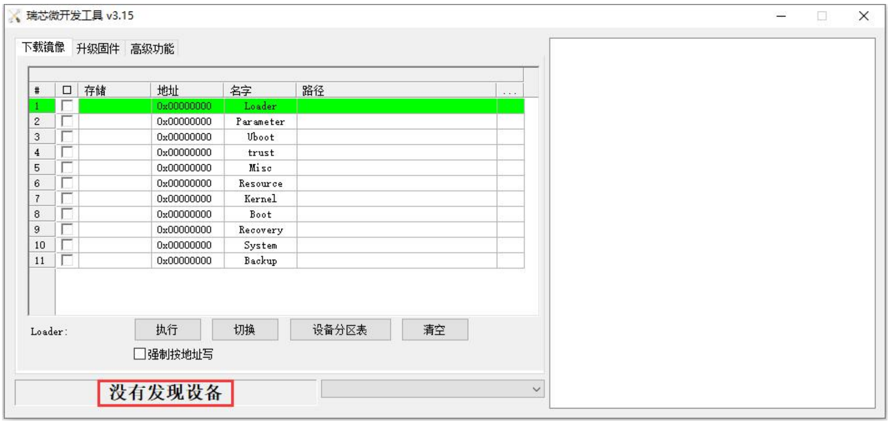


b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



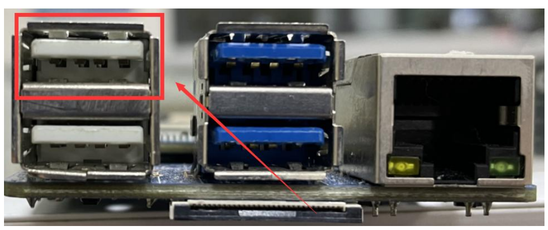
7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

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



9) Then start burning the Android image into eMMC

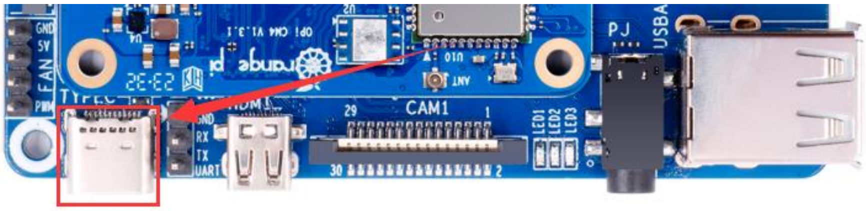
a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the development board's USB2.0 burning interface is as shown in the figure below.



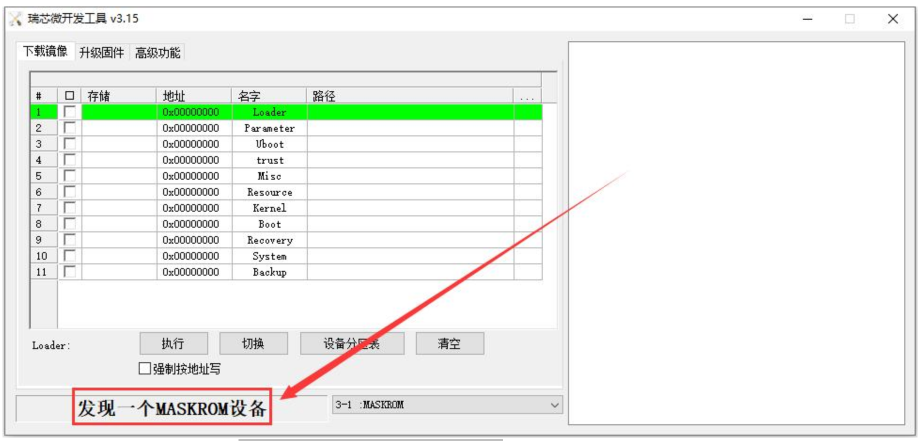
b. Then make sure that the development board is not inserted into the TF card and not connected to the power supply

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

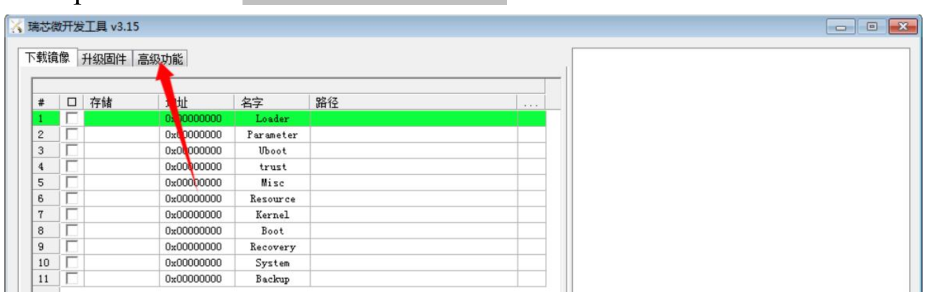
d. Then connect the development board to the power supply of the Type-C interface and power on



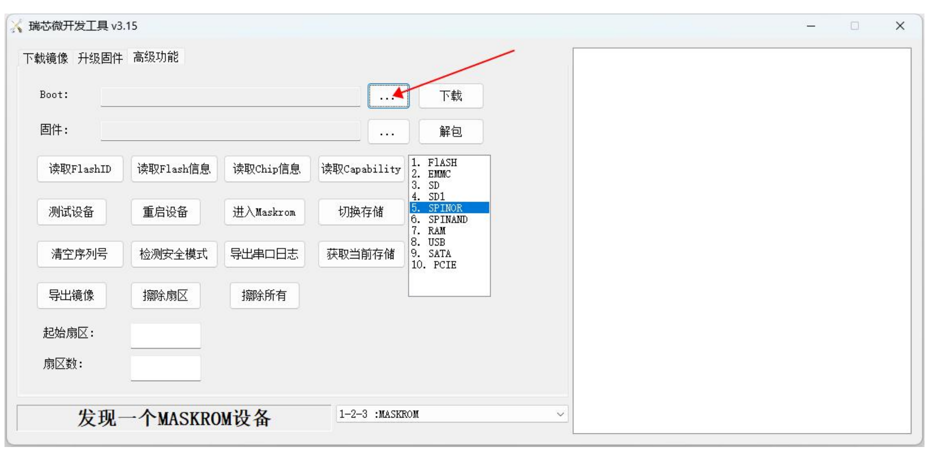
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



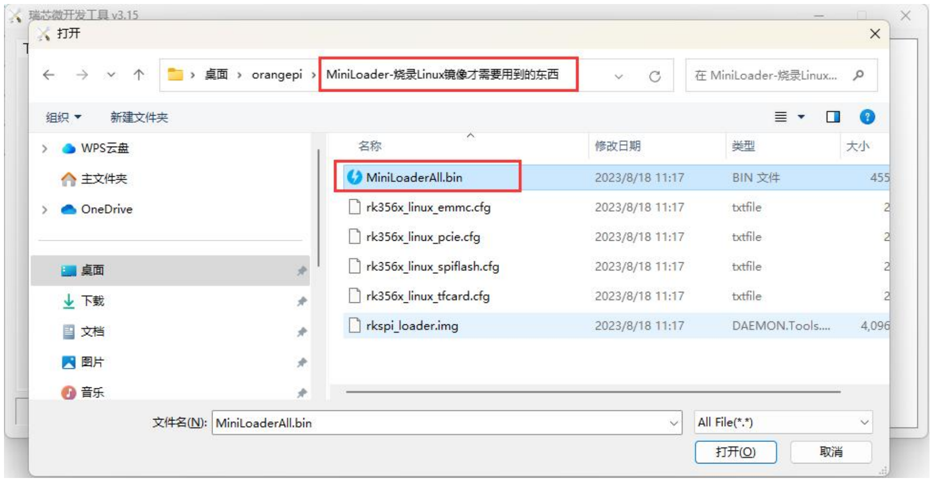
f. Then please select **advanced functions**



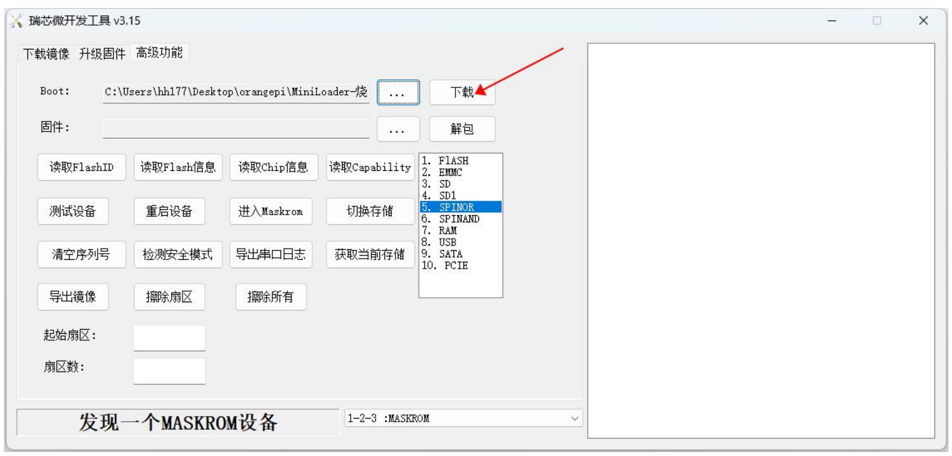
g. Then click the position shown in the figure below



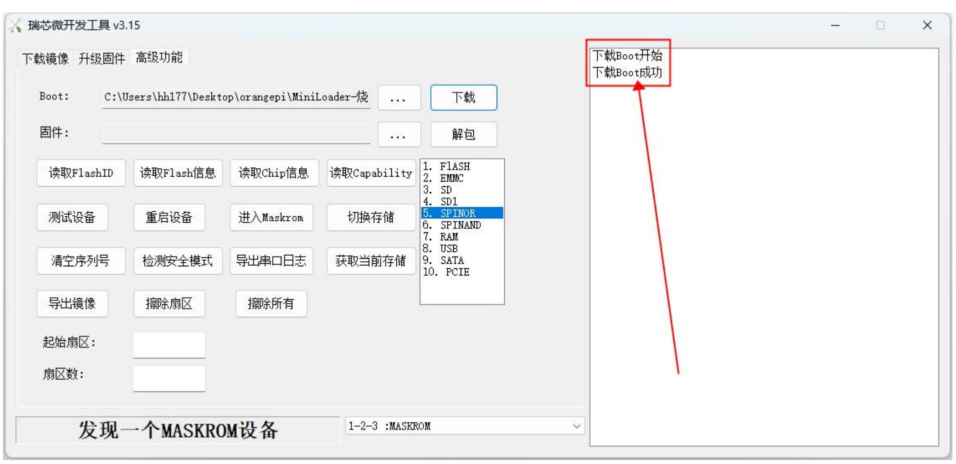
h. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click Open



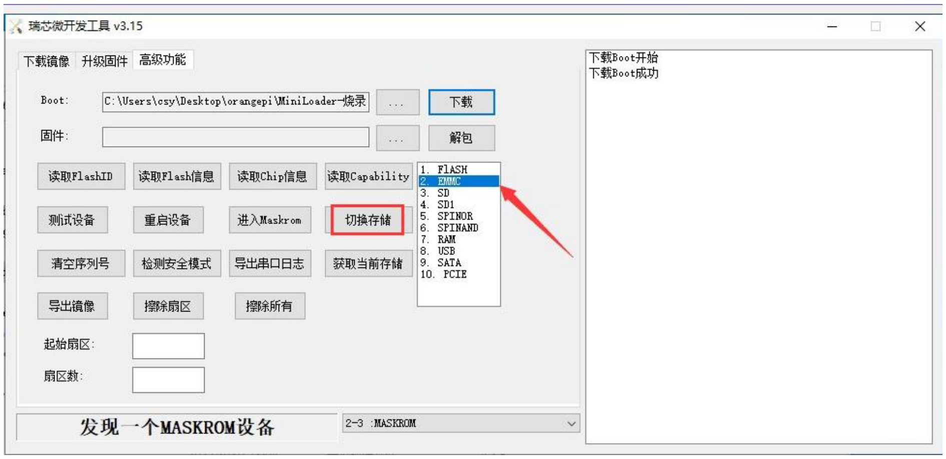
i. Then click **Download**



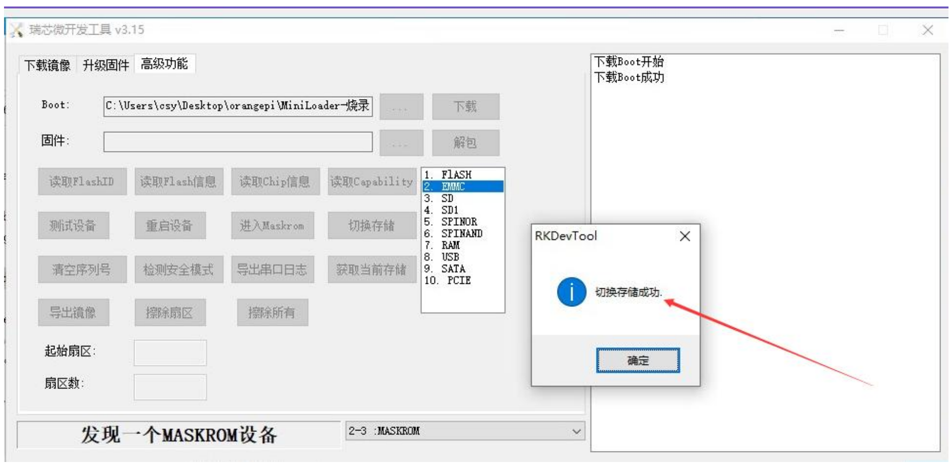
j. The display after downloading **MiniLoaderAll.bin** is shown in the figure below



k. Then select the storage device as **EMMC**, and then click **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 burning is completed, the Android system will start automatically.

2.8.2. How to burn the Android11 image into EMMC through the TF card

Note that all of the following operations are performed in Windows computers。

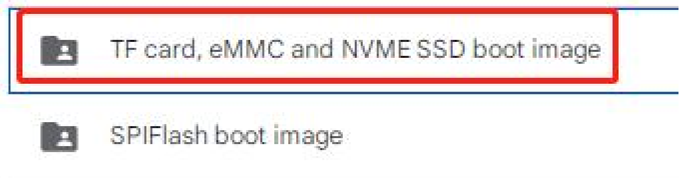
1) It is also necessary to prepare a 8GB or larger TF card. The transmission speed of the TF card must be **class10** or more. It is recommended to use TF cards from brands such as SanDisk.

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

3) Then download the SDDiskTool burn tool from **Orange PI's data download page**, **please make sure that the version of the SDDiskTool tool is the latest V1.72**

4) Then download the image of Android from **Orange Pi's data download page**

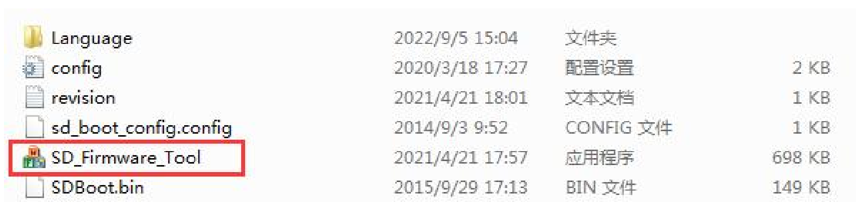
a. After turning on the download link of the Android image, you can see the two types of Android images below. Please select **the image in the TF card and EMMC startup image** folder for download



b. After entering the **TF card and EMMC startup image** folder, you can see the following two mirrors. The difference between them is：
   a) The first image is specifically used for HDMI display. It supports 4K display. If you do not use the LCD screen, download the image without LCD
   b) If you want to use the LCD screen, select the image with LCD

5) Then use the decompression software to decompress the downloaded Android image compressed package. In the files that are decompressed, the file ending with ".img" is the Android image file with a memory of more than 1GB

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



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

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

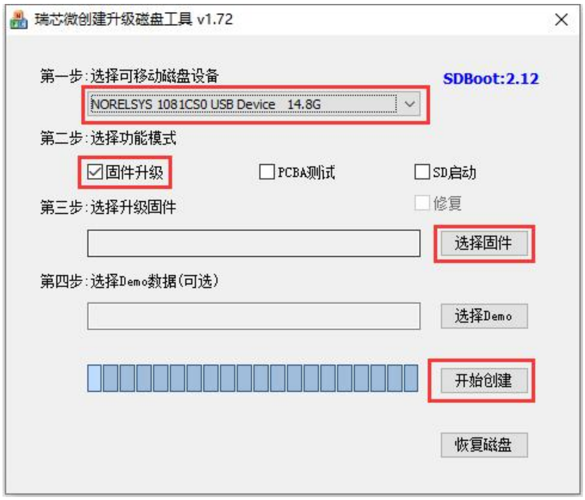
9) Then start writing the Android image to the TF card

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

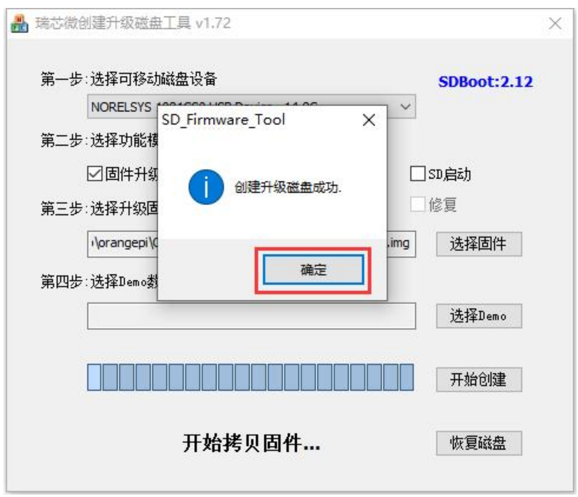
b. Then select **"Firmware Upgrade"** in **"Select Function Mode"**

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

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

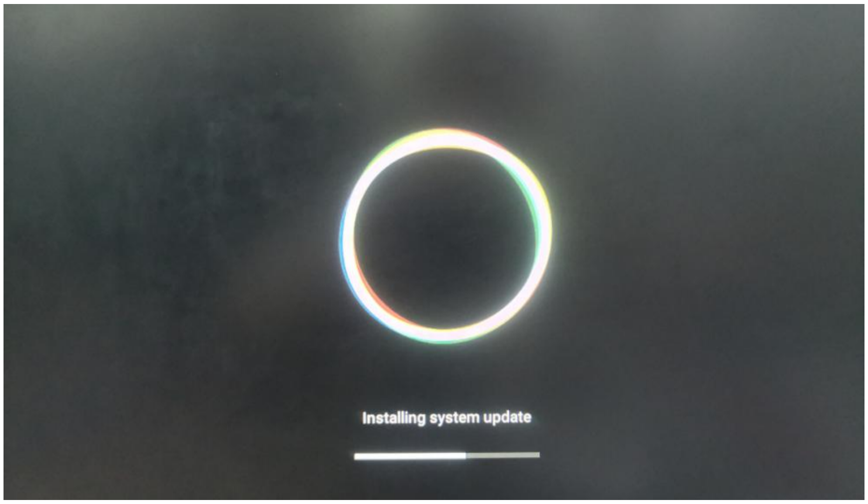


10) After the burning is completed, the display is as shown below, and then you can exit SDDiskTool

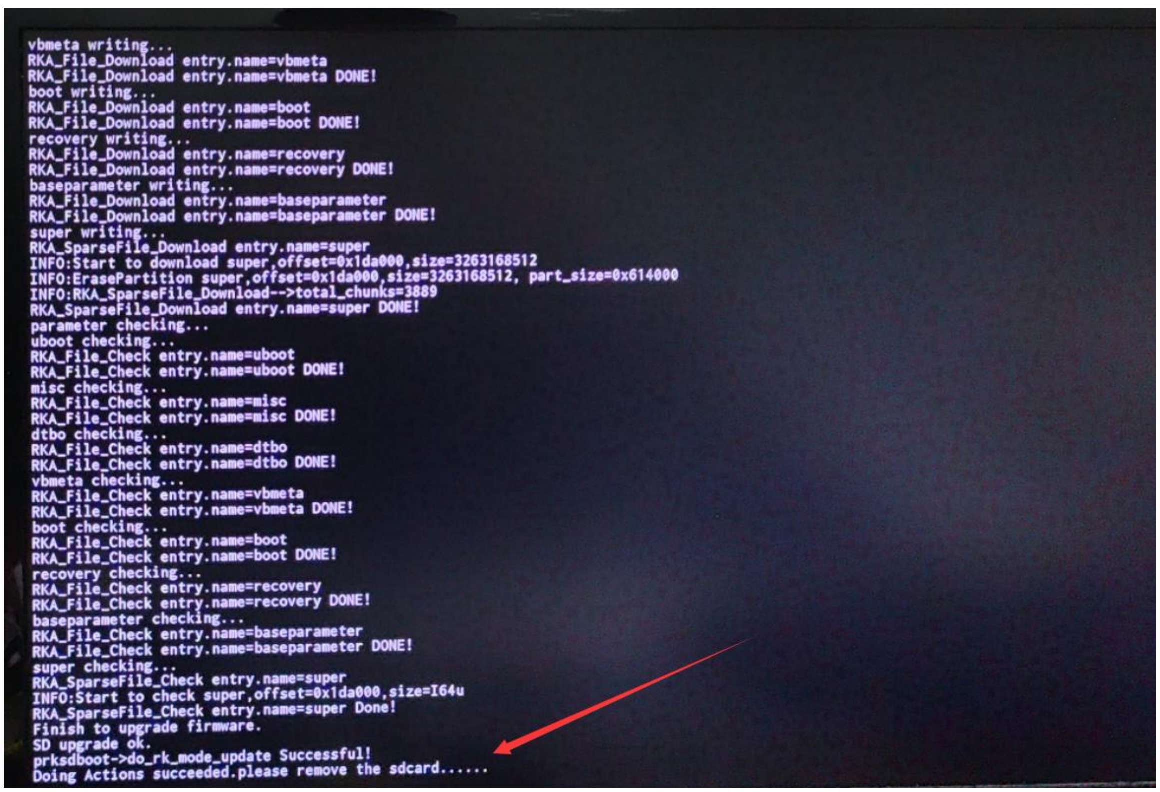


11) Then pull out the TF card from the computer and insert it into the development board. After the development board is powered on, it will automatically start burning the Android image in the TF card into the eMMC of the development board.

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



13) When the HDMI monitor displays the following information, it means that the burning of the Android image into the eMMC has been completed. At this time, the TF card can be pulled out, and then the Android system in the eMMC will start。



2. 9. How to burn Android image to SPIFlash+NVMe SSD

Before starting to burn the image, you must make sure that the development board has been pasted with the SPI Flash chip, because the development board is not pasted with the SPI Flash chip when it leaves the factory, so it needs to be purchased and soldered on by yourself. The SPI Flash chip model we recommend is XM25QU128CWIQT08Q

Since the startup priority of eMMC is higher than that of NVMe SSD, the system of eMMC needs to be cleared before burning the image. For the method of clearing eMMC, please refer to the instructions in the section "How to Clear eMMC Using RKDevTool"

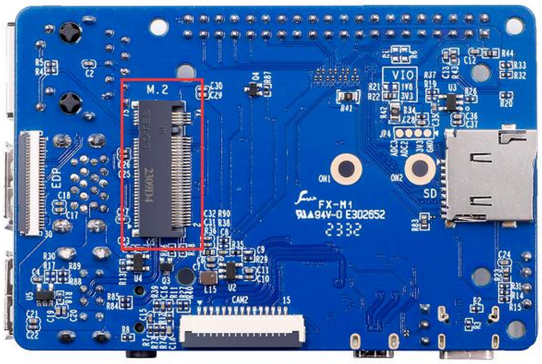
Note that all the following operations are performed on a Windows computer

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

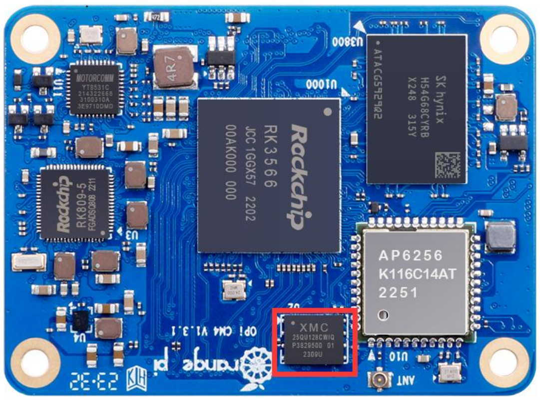
a. The M.2 2230 SSD is as follows

b. The M.2 2242 specification 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 make sure that the SPI Flash has been attached to the development board. The location of the SPI Flash on the development board is as shown in the picture below. No other settings are required before starting burning



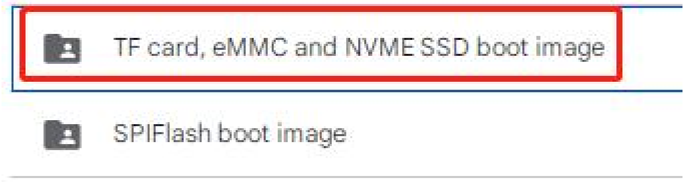
4) You also need to prepare a good quality USB2.0 male-to-male data cable

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

6) Then download the image of Android11

a. After opening the download link of the Android image, you can see the following two types of Android images. Please select the image in the

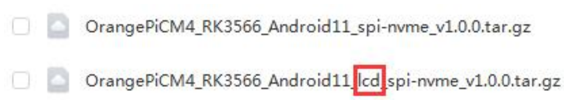
SPIFlash-NVME SSD boot image folder to download



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

a) The image without lcd is specially used for HDMI display and supports 4K display. If you don't use the LCD screen, please download the image without lcd

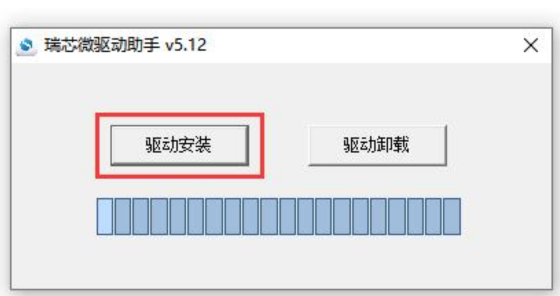
b) If you want to use an LCD screen, please choose the image with lcd



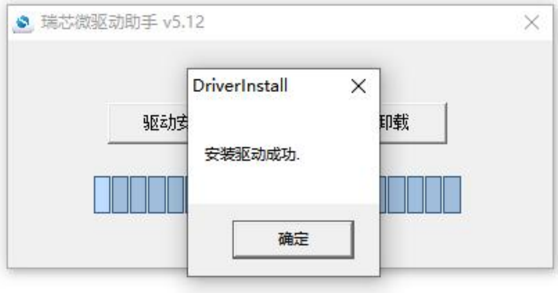
7) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

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

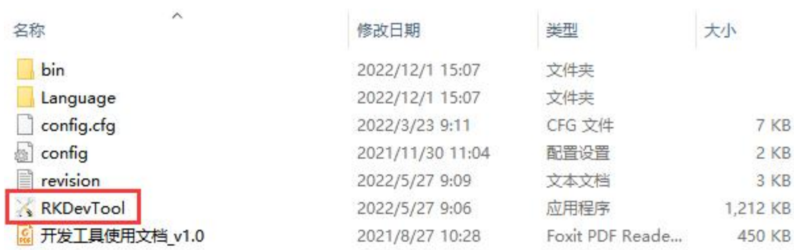
a. Click the "**Driver Installation**" button



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



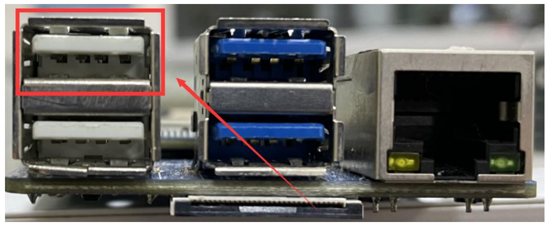
9) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it



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

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

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

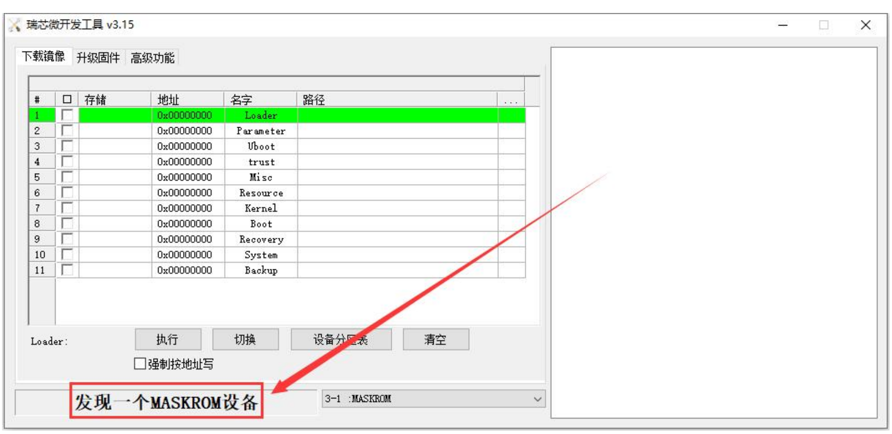


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

c. Then press and hold the MaskROM button on the development board. The location of the MaskROM button on the development board is as shown in the 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 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 devic**e "



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

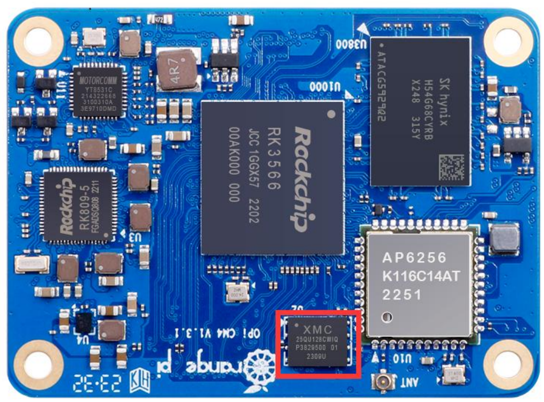
g. Then click the "**Firmware**" button to select the Android image that needs to be burned

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

2. 10. How to clear SPIFlash using RKDevTool

There is no SPI Flash chip on the development board when it leaves the factory, so it needs to be pasted by yourself. The model of the SPI Flash chip we recommend is XM25QU128CWIQT08Q

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

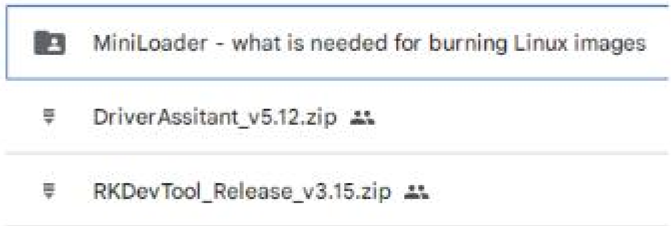


2) First of all, you need to prepare a good quality USB2.0 male-to-male data cable

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

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

b. Then download all the files below

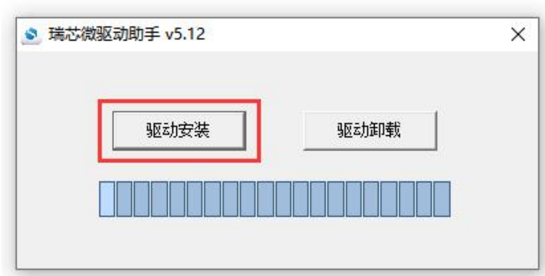


Note that the "MiniLoader-things needed to burn Linux images" folder will be referred to as the MiniLoader folder below

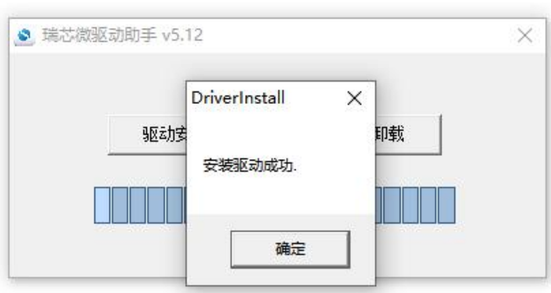
4) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it

5) The steps to install the Rockchip driver after opening **DriverInstall.exe** are as follows

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



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

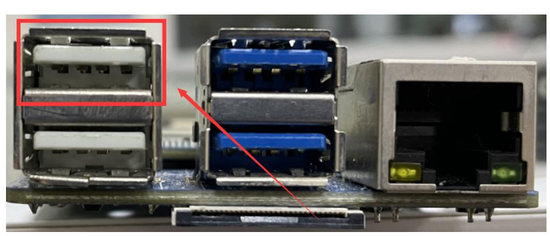


6) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

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

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

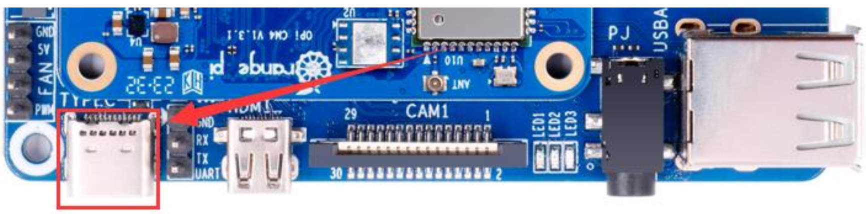
a. First, connect the development board to the Windows computer through a USB2.0 male-to-male data cable. The location of the USB2.0 programming port on the development board is as shown in the figure below.



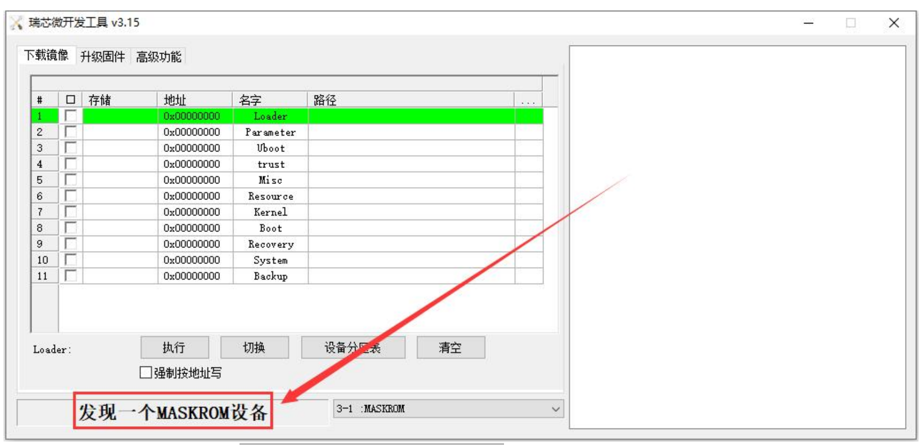
b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

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

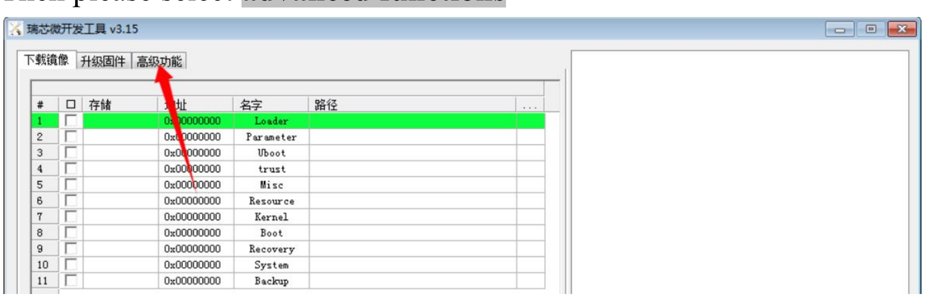
d. Then connect the development board to the power supply of the Type-C interface, power it on, and then release the MaskROM button.



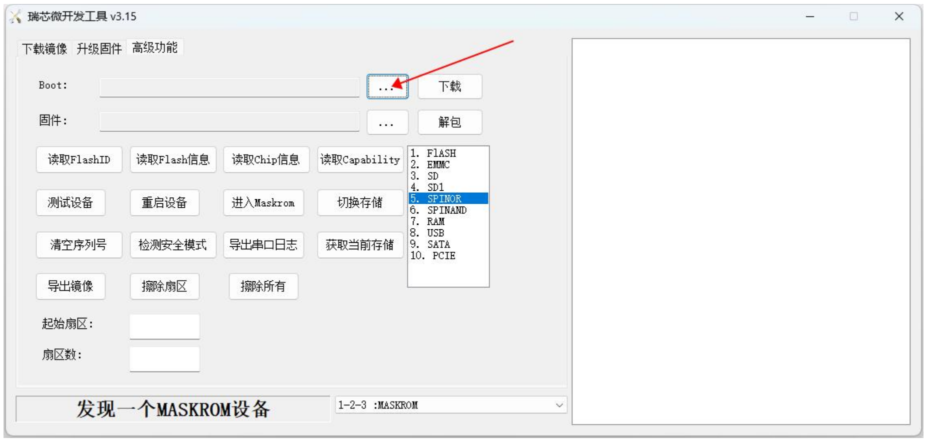
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



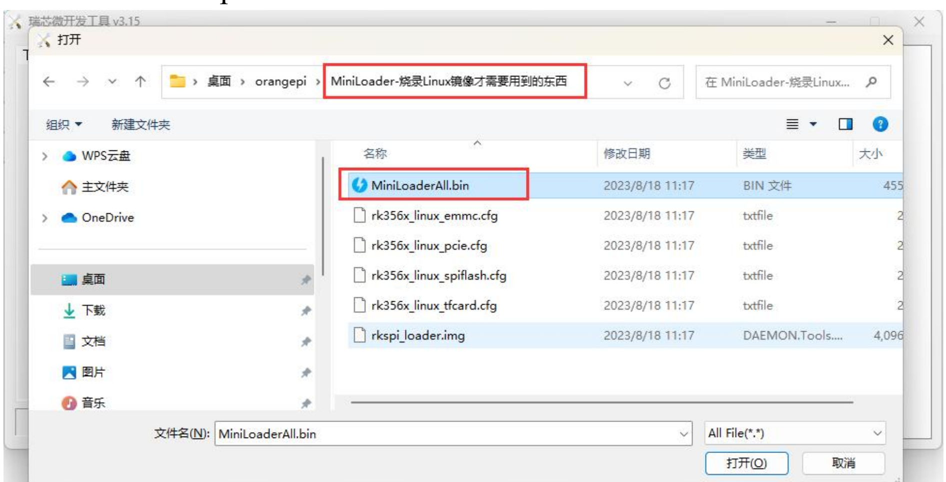
f. Then please select **advanced functions**



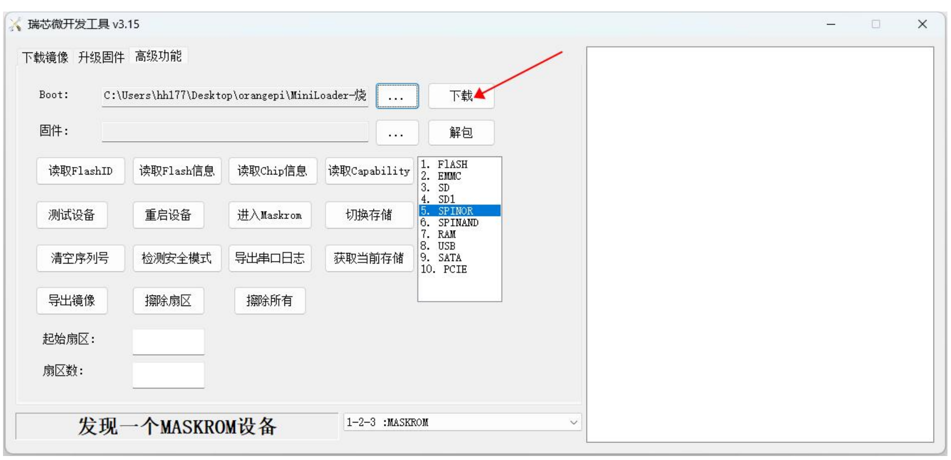
g. Then click the position shown in the figure below



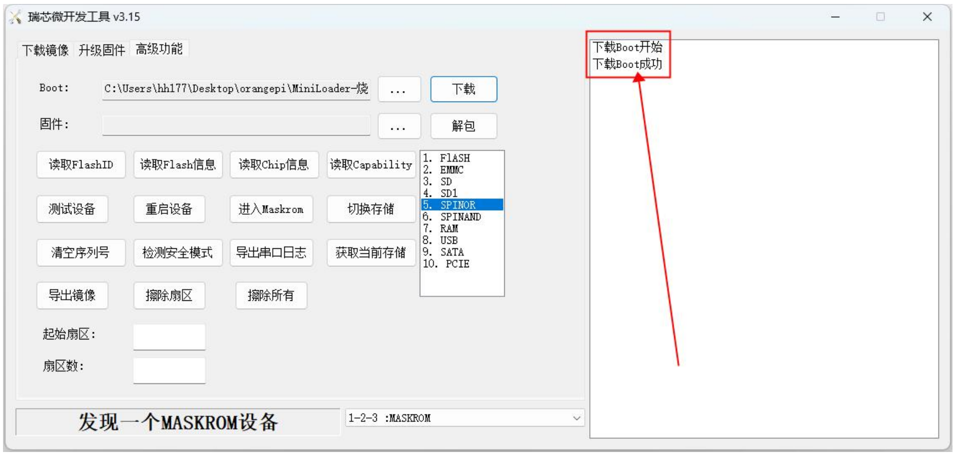
h. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click Open



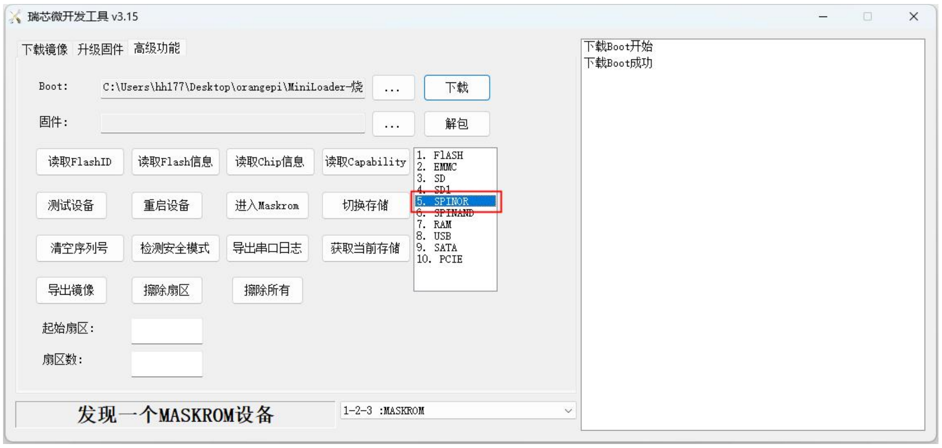
i. Then click **Download**



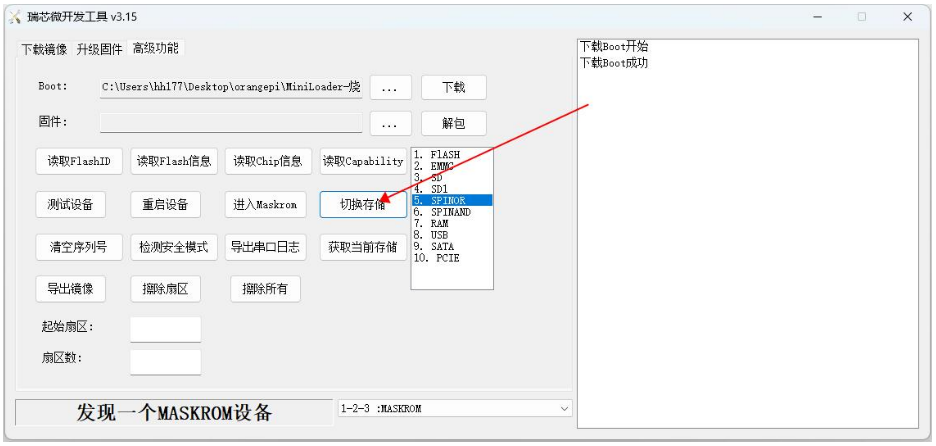
j. The display after downloading **MiniLoaderAll.bin** is shown in the figure below



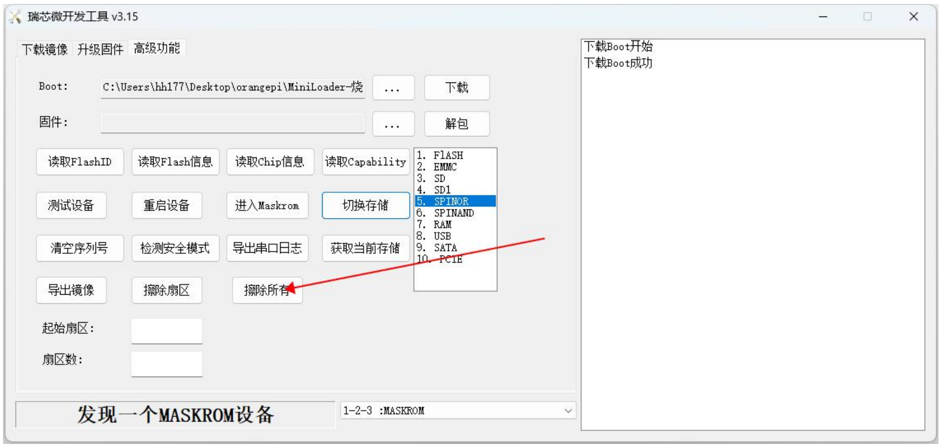
k. Then select the storage device as **SPINOR**



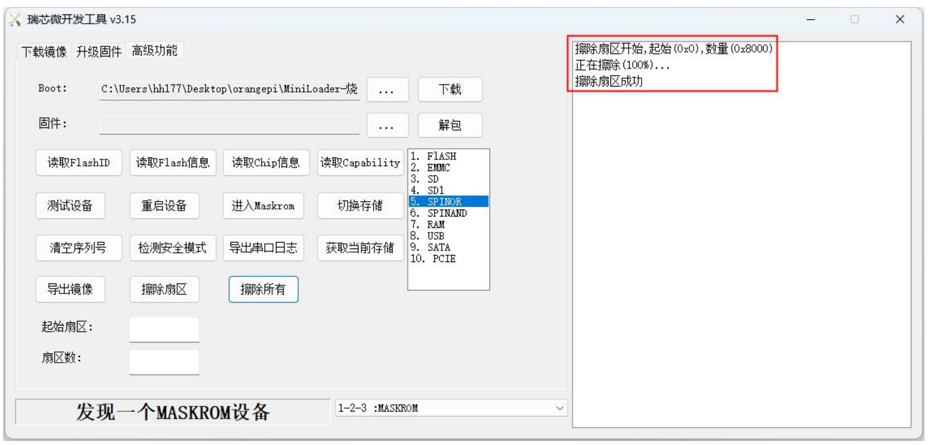
l. Then click **switch storage**



m. Then click **Erase All** and it will start erasing SPIFlash.



n. The display log after erasing SPIFlash is shown in the figure below



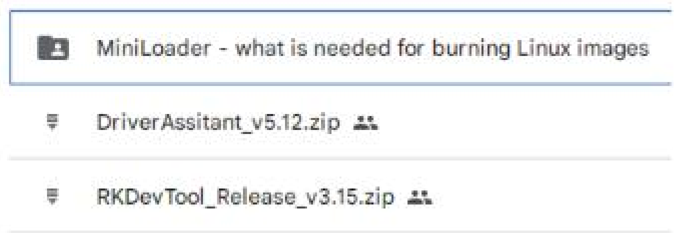
2.11. How to clear eMMC using RKDevTool

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

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

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

b. Then download all the files below

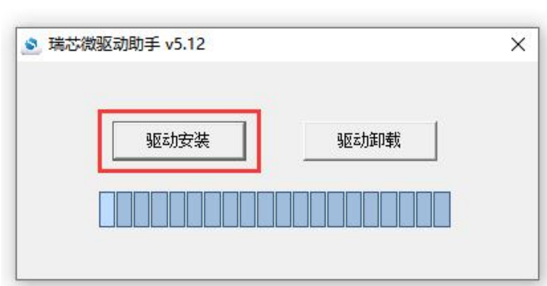


Note that the "MiniLoader-things needed to burn the Linux image" folder is hereinafter referred to as the MiniLoader folder。

3) 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

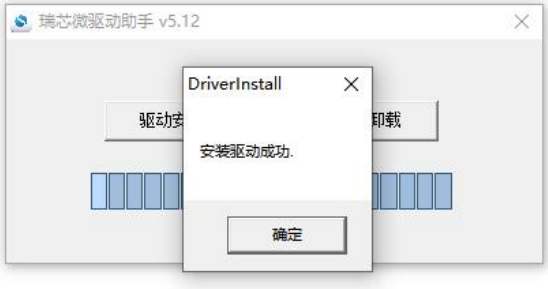
4) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

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



b. After waiting for a period of time, a window will pop up prompting "**Driver**

installation successful", then click the "**OK**" button.

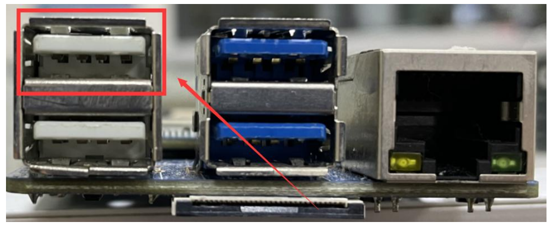


5) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

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

7) Then you can start to clear the content in eMMC

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

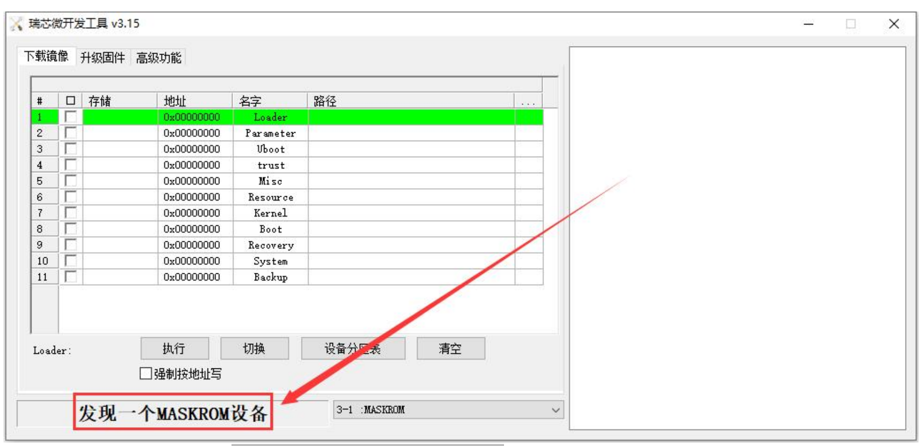


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

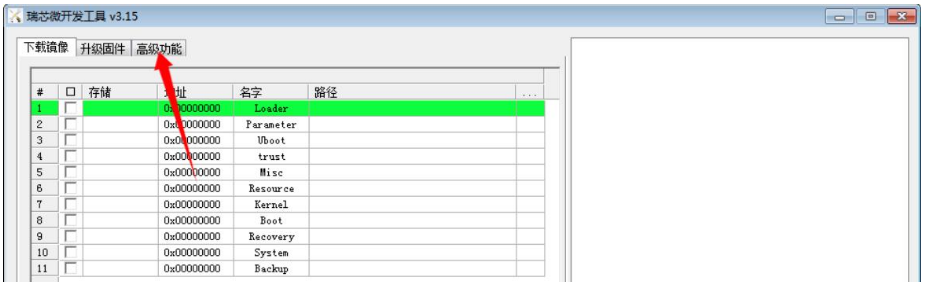
c. Then press and hold the MaskROM button on the development board. The location of the MaskROM button on the development board is as shown in the figure below：

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

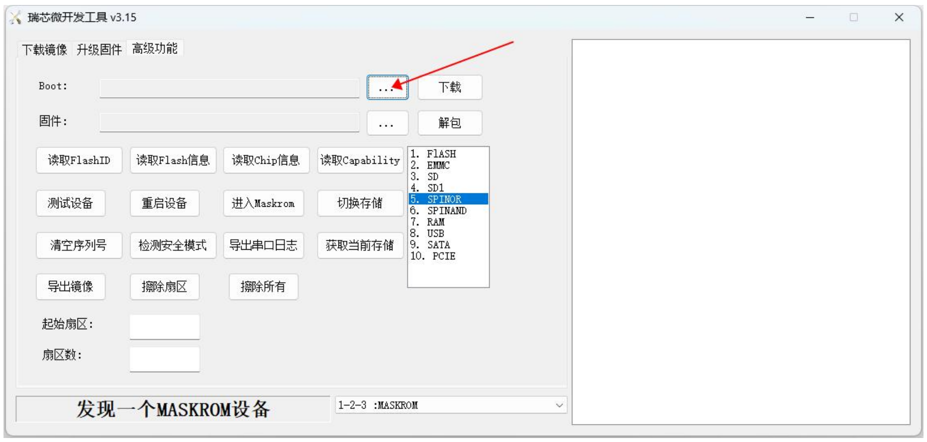
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



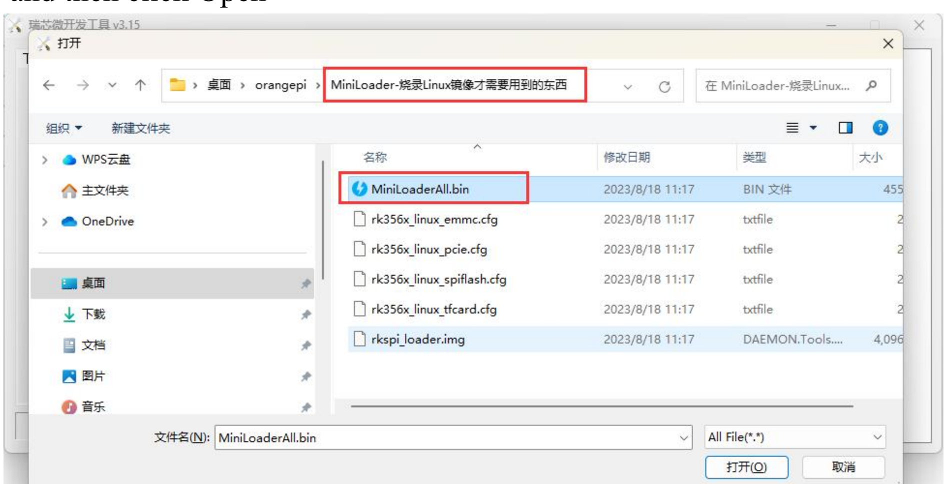
f. Then please select **advanced functions**



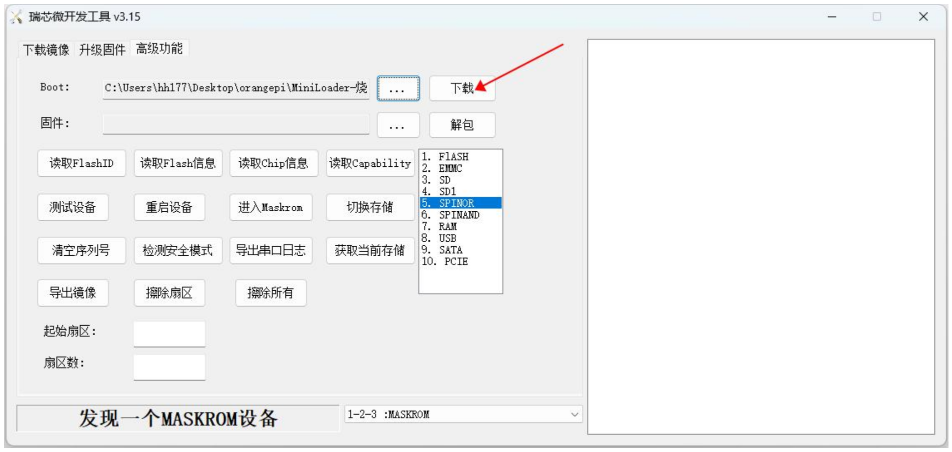
g. Then click the position shown in the figure below



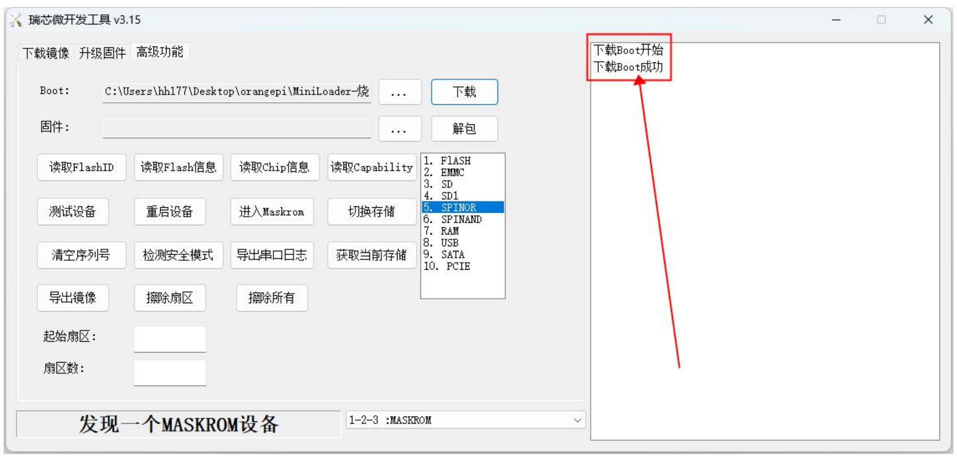
h. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click Open



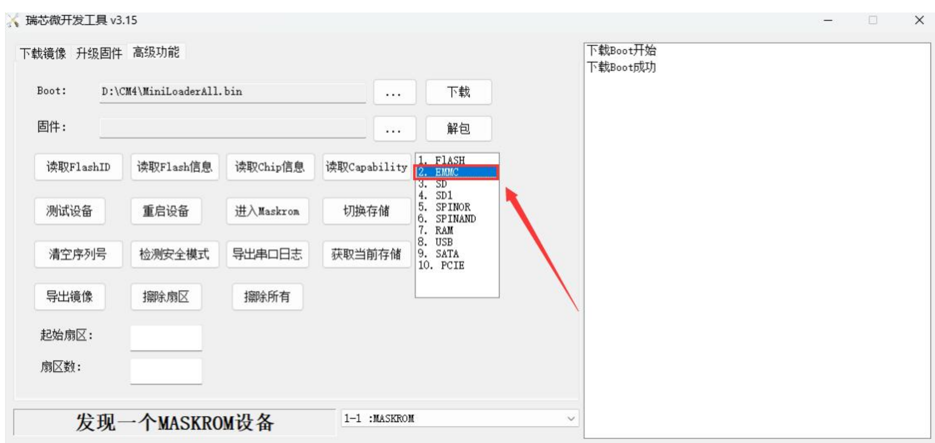
i. Then click **Download**



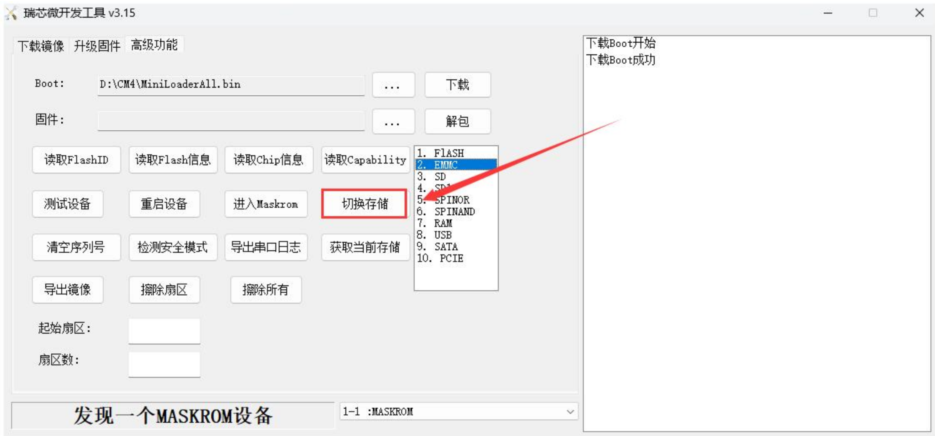
j. The display after downloading **MiniLoaderAll.bin** is shown in the figure below



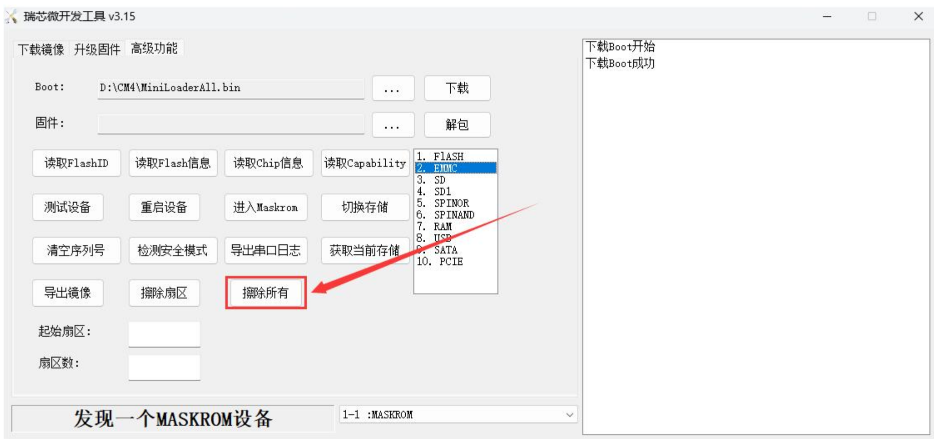
k. Then select the storage device as **eMMC**



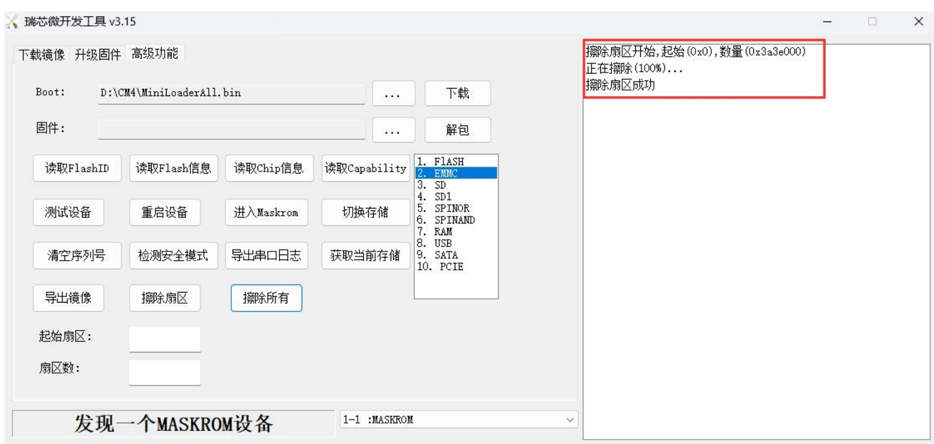
l. Then click **switch storage**



m. Then click **Erase All** to start erasing the eMMC.

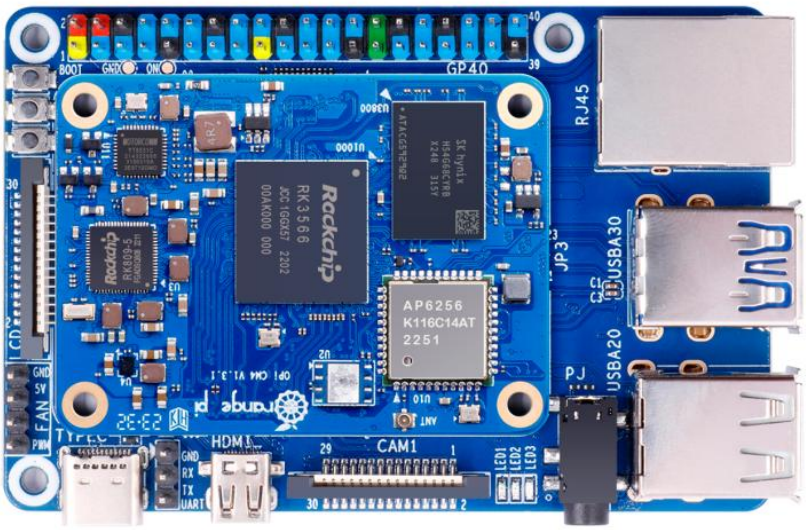


n. The display log after erasing eMMC is as shown below



2. 12. Boot the Orange Pi development board

1) First install the Orange Pi CM4 core board on the base board



2) Then insert the TF card with the burned image into the TF card slot of the Orange Pi development board.

3) The development board has a Micro HDMI interface, and the development board can be connected to a TV or HDMI display through a Micro HDMI to HDMI cable. If you have purchased an LCD screen, you can also use the LCD screen to display the system interface of the development board

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

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

6) Connect a **high-quality** power adapter with a 5V/3A USB Type-C interface。

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

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

Type-C power interface does not support PD negotiation.

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

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

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

2. 13. How to use the debugging serial port

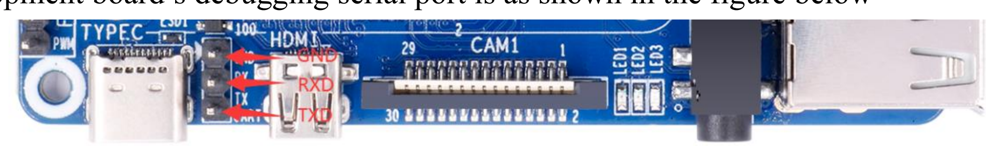
2. 13. 1. Connection instruction of debugging serial port

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

For better compatibility, it is recommended to use CH340 USB to TTL module, please do not use CP2102, PL2303 type USB to TTL module.

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

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



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

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

4) The schematic diagram of connecting the USB to TTL module to the computer and Orange Pi development board is as 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 casually. If there is no output in the test, then exchange the order of TX and RX, so that there is always a The order is correct

2. 13. 2. How to use the debugging serial port on Ubuntu platform

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

1) First insert the USB to TTL module into the USB interface of the Ubuntu computer. If the USB to TTL module is connected and recognized normally, you can see the corresponding device node name under **/dev** of the Ubuntu PC. Remember this node name and set the serial port later. software will be used.

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

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

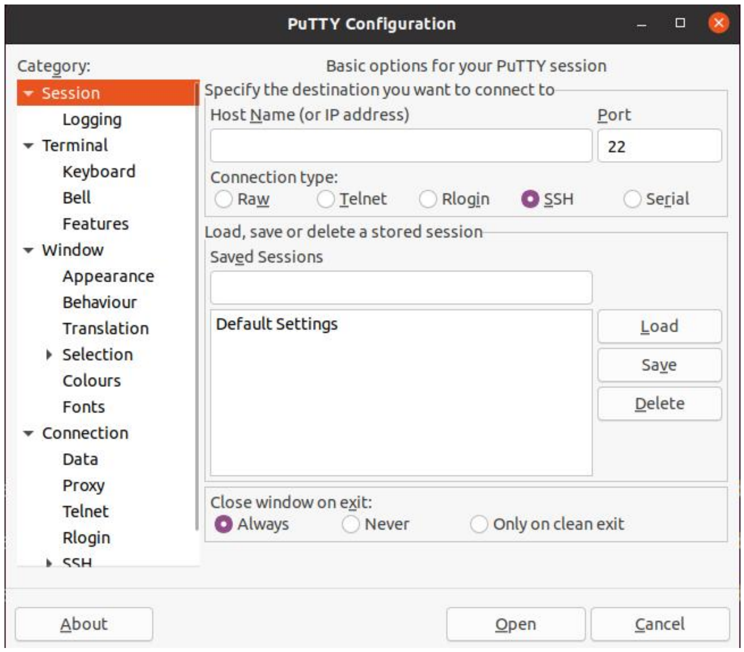
```
test@test:~$ sudo apt-get update
```

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

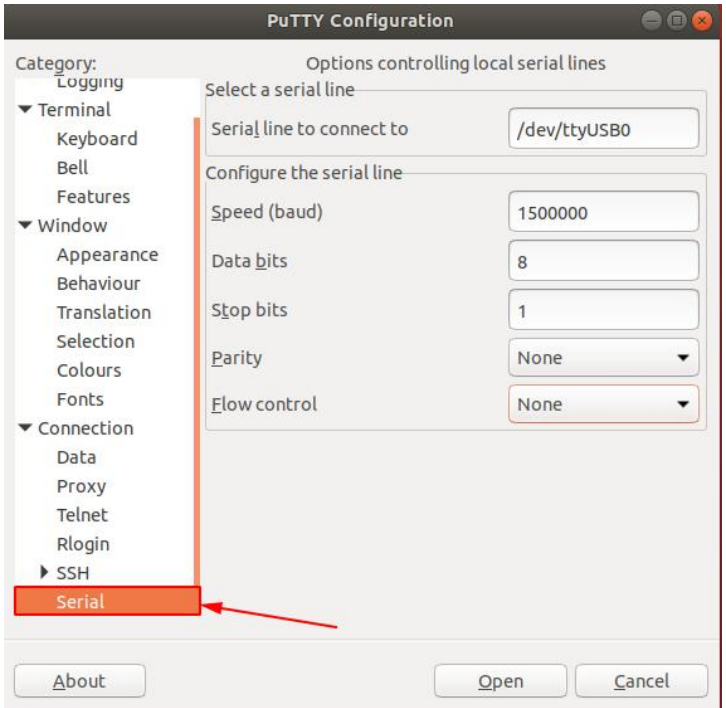
3) Then run putty, **remember to add sudo permissions**

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

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



5) First select the serial port setting interface

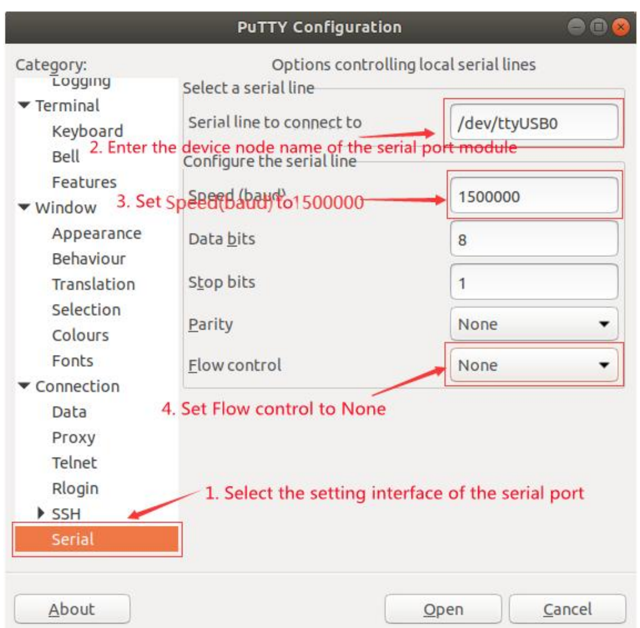


6) Then set the parameters of the serial port

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

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

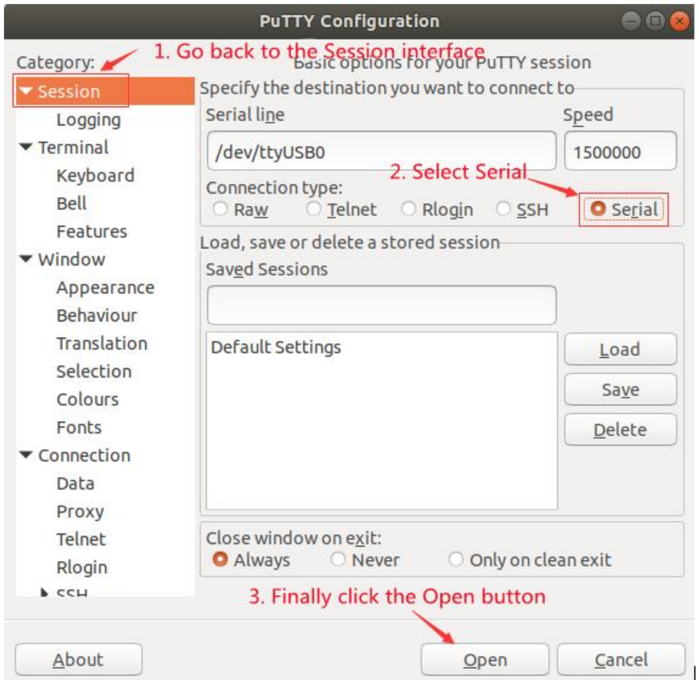
   c. Set Flow control to None



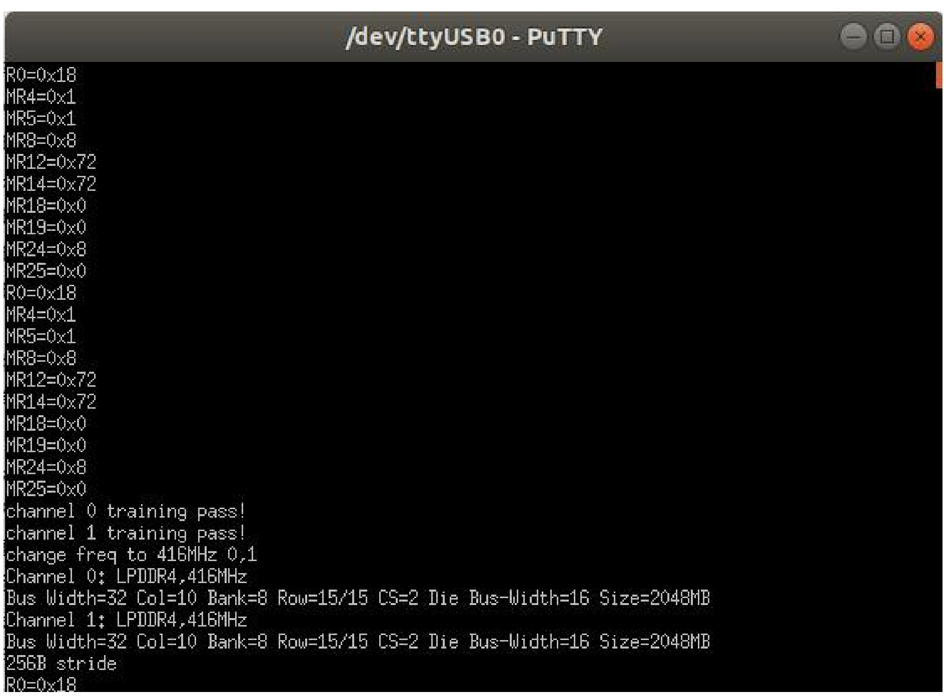
7) After setting up the serial port setting interface, return to the Session interface.

a. First select the Connection type as Serial

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



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



2. 13. 3. How to use the debugging serial port on Windows platform

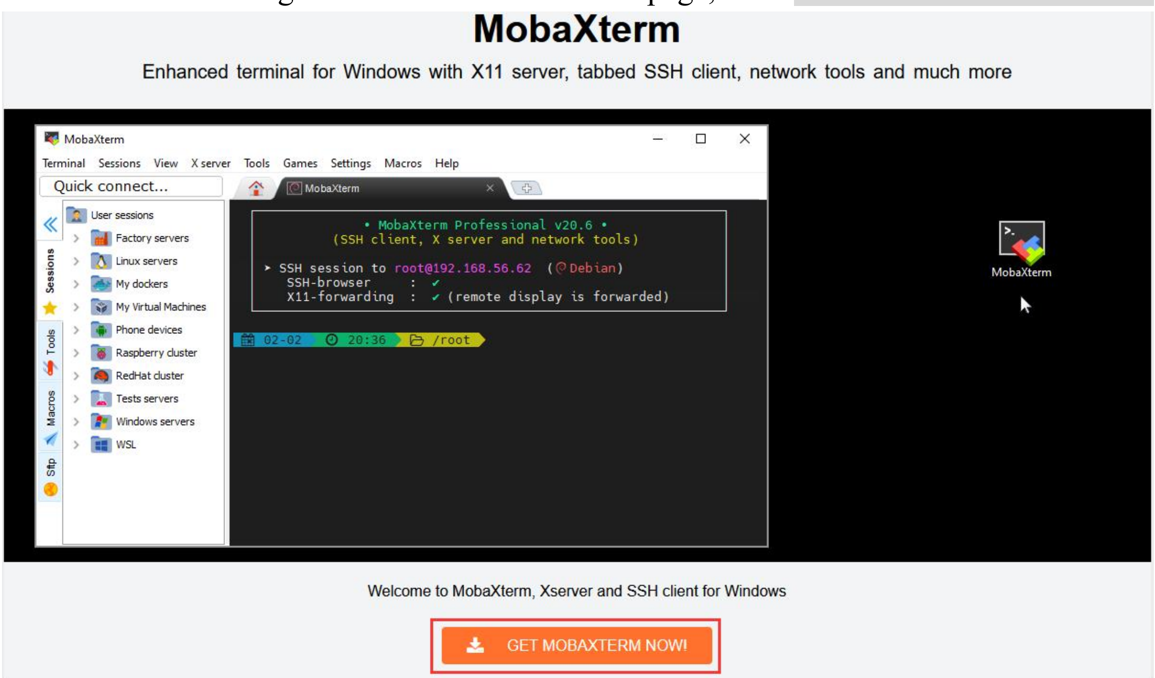
There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without buying a serial number.

1) Download MobaXterm

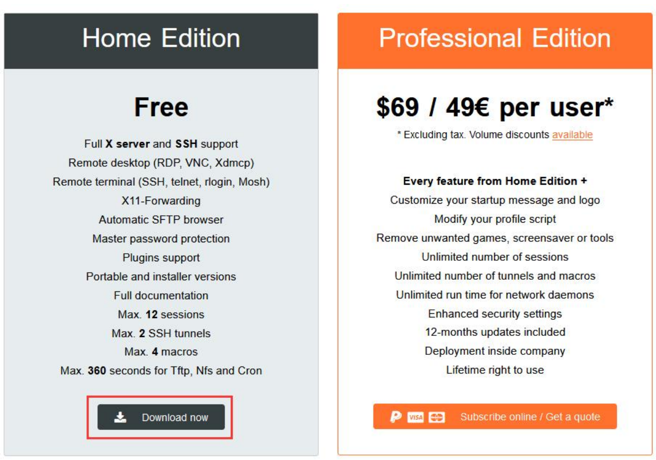
   a. Download MobaXterm website as follows

https://mobaxterm.mobatek.net

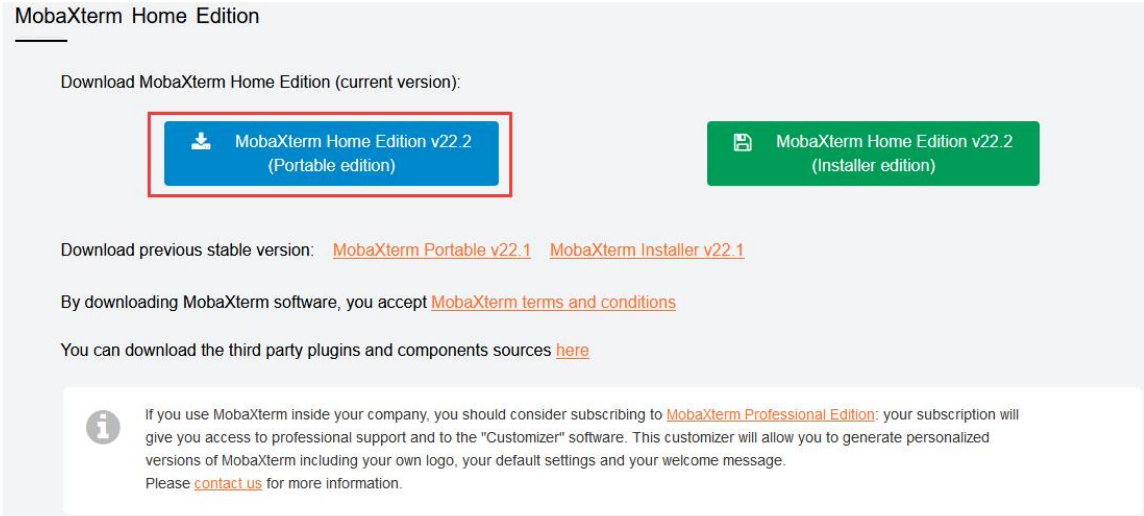
   b. After entering the MobaXterm download page, click **GET XOBATERM NOW!**



   c. Then choose to download the Home version



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



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

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

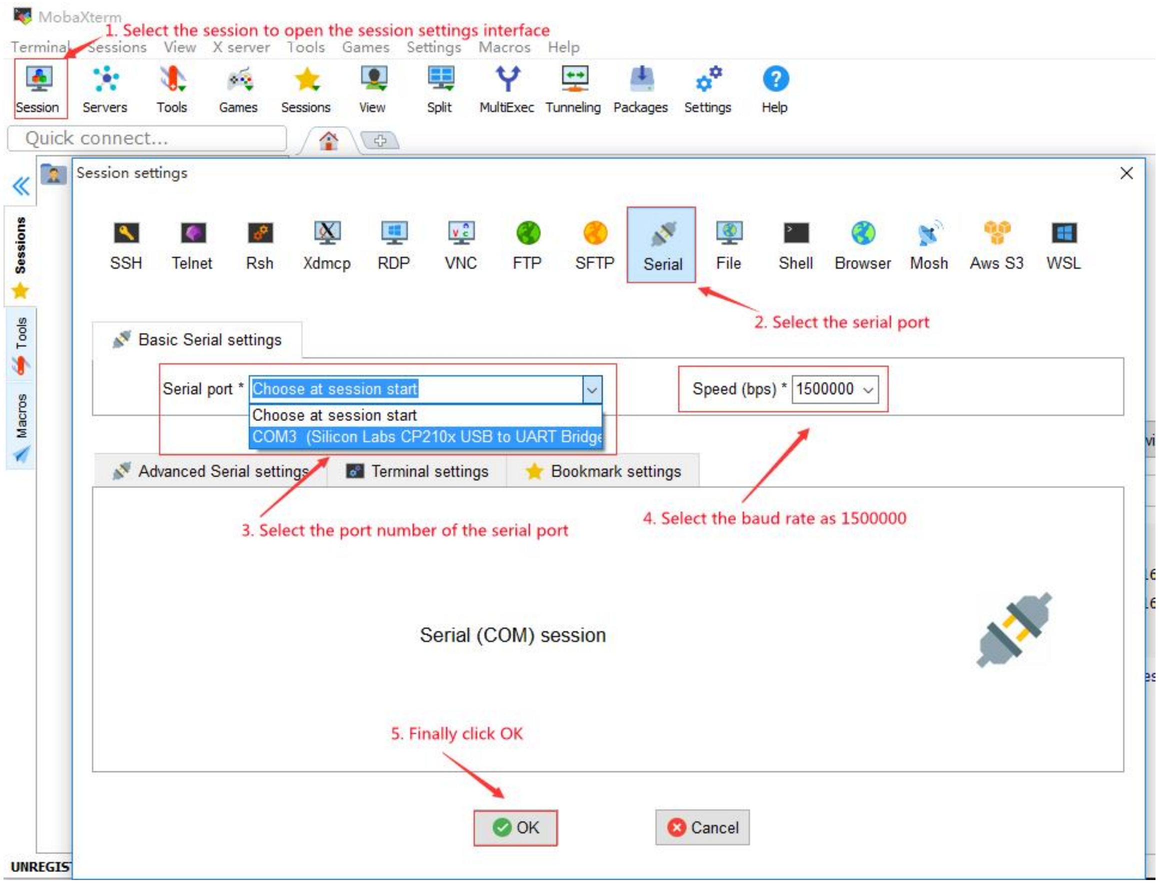
a. Open the session settings interface

b. Select the serial port type

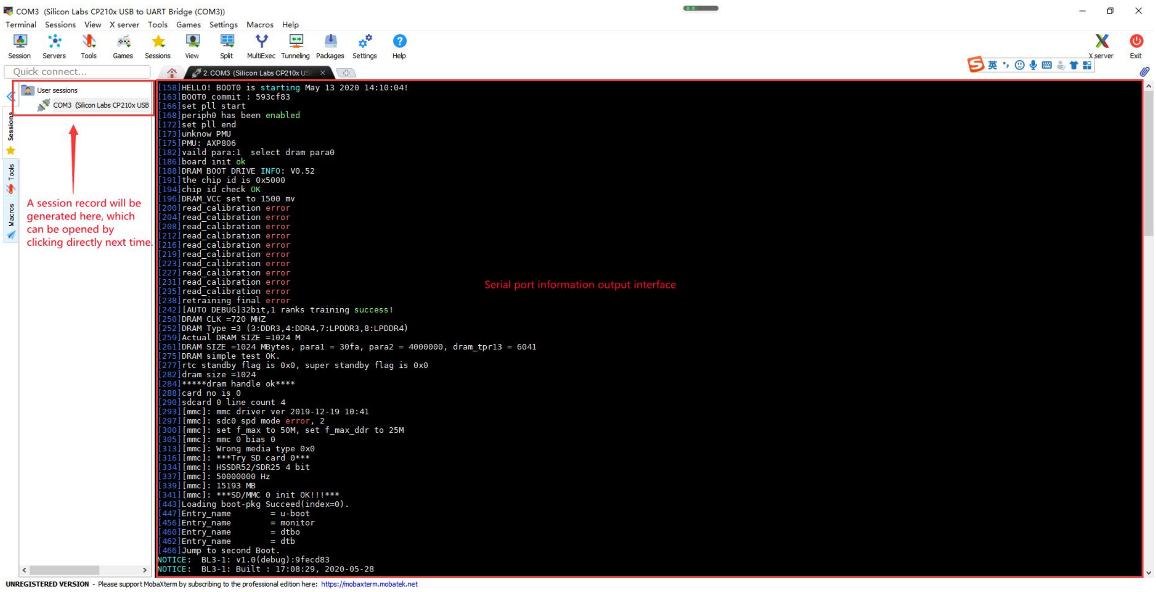
c. Select the port number of the serial port (select the corresponding port number according to the actual situation). If you cannot see the port number, please use **360 Driver Master** to scan and install the driver for the USB to TTL serial port chip.

d. Select the baud rate of the serial port to be **1500000**

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



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



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

The power supply method we recommend for the development board is to use the 5V/3A Type C interface power cord to plug into the Type-C power interface of the development board. If you need to use the 5V pin in the 40-pin interface to power the development board, please ensure that the power cord and power adapter used can meet the power supply requirements of the development board. If the use is unstable, please switch to Type-C power supply。

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

The power cord shown in the figure above can be bought on Taobao, please search for purchase by yourself。

2) Use the 5V pin in the 40pin interface to power the development board. The power cord connection is as follows

a. The USB A port of the power cord shown in the picture above needs to be plugged into the 5V/3A power adapter connector.（**Please do not plug it into the USB port of your computer for power supply**）

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

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

d. The position of the 40Pin interface 5V pin and GND pin in the development board is shown in the figure below，**Remember not to reverse**

3. Ubuntu/Debian Server and Xfce desktop system instructions

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

3. 1. Supported Linux image types and kernel versions

Linux image type	Kernel version	server version	desktop version
Debian 11 - Bullseye	Linux5.10	support	support
Debian12 - Bookworm	Linux5.10	support	support
Ubuntu 20.04 - Focal	Linux5.10	support	support
Ubuntu 22.04 - Jammy	Linux5.10	support	support

3. 2. Linux system adaptation situation

Function	Debian11	Debian12	Ubuntu20.04	Ubuntu22.04
USB2.0x3	OK	OK	OK	OK
USB3.0x1	OK	OK	OK	OK
M.2 NVMe SSD Boot	OK, you need to attach SPI Flash to it for normal use.			
WIFI	OK	OK	OK	OK
Bluetooth	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**
PWM（40pin）	**OK**	**OK**	**OK**	**OK**
PWM fan interface	**OK**	**OK**	**OK**	**OK**
3pin debugging serial port	**OK**	**OK**	**OK**	**OK**
EMMC	**OK**	**OK**	**OK**	**OK**
TF card startup	**OK**	**OK**	**OK**	**OK**
HDMI video	**OK**	**OK**	**OK**	**OK**
HDMI audio	**OK**	**OK**	**OK**	**OK**
OV5647 camera	**Kernel driver is OK, 3A is not adjusted**			
LCD	**OK**	**OK**	**OK**	**OK**
eDP display	**OK**	**OK**	**OK**	**OK**
Gigabit Ethernet port	**OK**	**OK**	**OK**	**OK**
Network port status light	**OK**	**OK**	**OK**	**OK**
Headphone playback	**OK**	**OK**	**OK**	**OK**
headphone recording	**OK**	**OK**	**OK**	**OK**
LED light	**OK**	**OK**	**OK**	**OK**
GPU	**OK**	**OK**	**OK**	**OK**
NPU	**OK**	**OK**	**OK**	**OK**
VPU	**OK**	**OK**	**OK**	**OK**
watchdog test	**OK**	**OK**	**OK**	**OK**
Chromium hard decryption video	**OK**	**OK**	**OK**	**OK**

3. 3. Linux command format description in this manual

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

```

```

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

```

```

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

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

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

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

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

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

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

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

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

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

```
console=serial
```

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

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

3. 4. Linux system login instructions

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

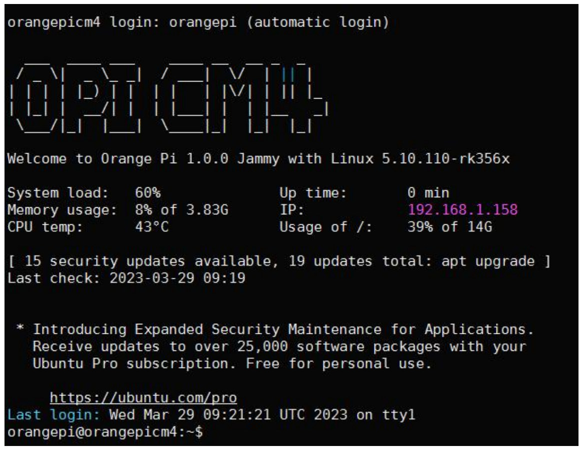
account	password
root	orangepi
orangepi	orangepi

Note that when entering a password, the specific content of the entered password will not be displayed on the screen. Please do not think that there is something wrong. Just press Enter after entering it。

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

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

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



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

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

3) Use the following command to disable automatic login to the terminal

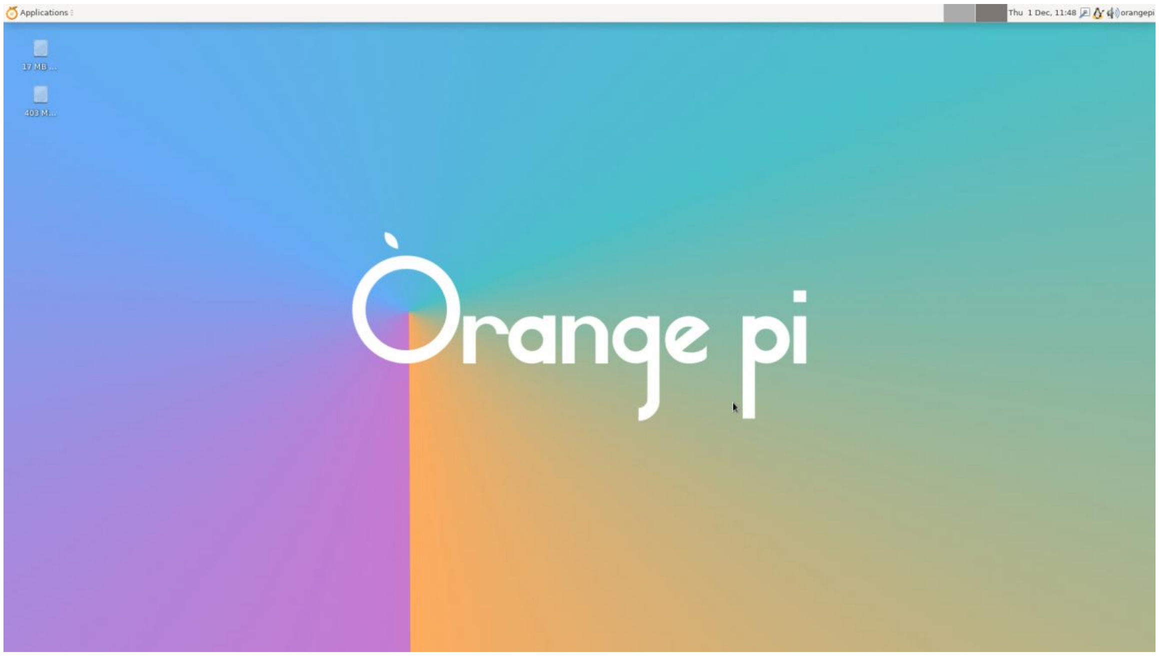
orangepi@orangepi:~$ **sudo auto_login_cli.sh -d**

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

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

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

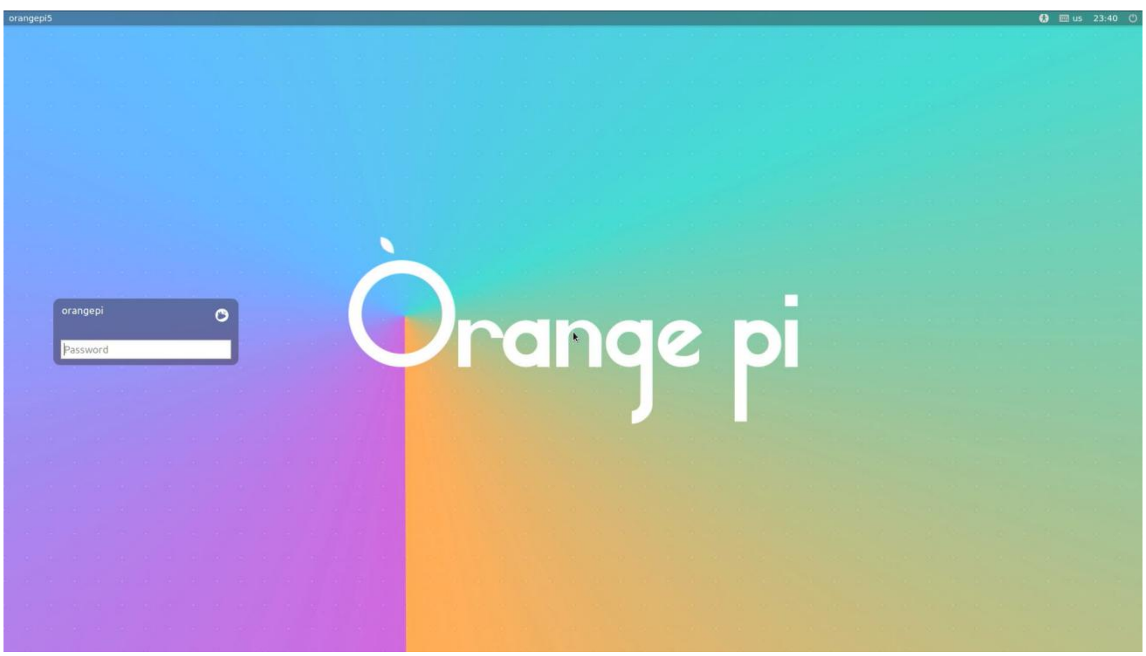
1) The desktop version system will automatically log in to the desktop after startup without entering a password.



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

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

3) Then restart the system and a login dialog box will appear. At this time, you need to enter a **password** to enter the system



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

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

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

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

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

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

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

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

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

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

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

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

orangepi@orangepi:~$ **sudo reboot**

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

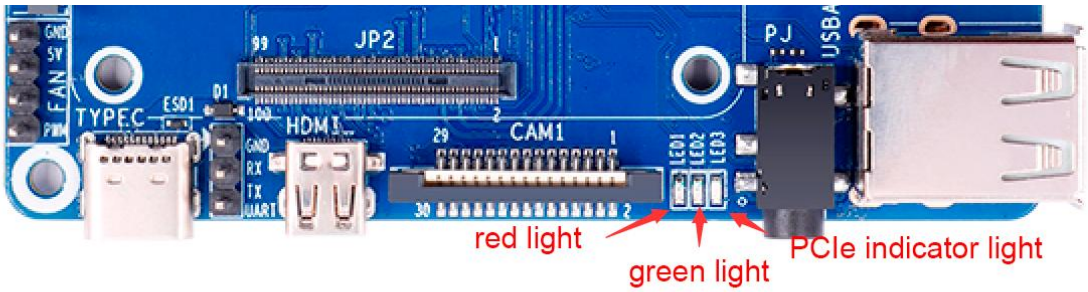
a. First enter the following command in the command line, **please remember to add sudo permission**

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

b. After the command is executed, the desktop will be displayed

3. 5. Onboard LED light test instructions

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



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

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

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

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

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

a. First enter the setting directory of the green light

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

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

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

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

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

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

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

3. 6. Network connection test

3. 6. 1. Ethernet port test

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

2) After the system starts, it will automatically assign an IP address to the Ethernet card through **DHCP**.，**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 eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 4a:fe:2b:3d:17:1c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.150/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
       valid_lft 43150sec preferred_lft 43150sec
    inet6 fe80::9a04:3703:faed:23be/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

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

```
orangepi@orangepi:~$ ifconfig
Command 'ifconfig' is available in the following places
  * /sbin/ifconfig
  * /usr/sbin/ifconfig
The command could not be located because '/sbin:/usr/sbin' is not included in the PATH environment variable.
This is most likely caused by the lack of administrative privileges associated with your user account.
ifconfig: command not found
```

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

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

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

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

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

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

C）Try another router (I have encountered many problems with routers, such as the router being unable to assign an IP address normally, or the IP address being assigned normally but not visible in the router);

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

In addition, it should be noted that the development board's DHCP automatic allocation of IP addresses does not require any settings.

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

```
orangepi@orangepi:~$ ping www.baidu.com -I eth0
PING www.a.shifen.com (14.215.177.38) from 192.168.1.12 eth0: 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=56 time=6.74 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=56 time=6.80 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=56 time=6.26 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=56 time=7.27 ms
^C
```

```
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 6.260/6.770/7.275/0.373 ms
```

3. 6. 2. WIFI connection test

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

3. 6. 2. 1. The server image connects to WIFI through commands

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

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

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

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

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

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

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

```
root@orangepi:~# nmcli dev wifi
IN-USE  BSSID              SSID                  MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
        28:6C:07:6E:87:2E  [redacted]_orangepi   Infra  9     260 Mbit/s  97      ▂▄▆█  WPA1 WPA2
        D8:D8:66:A5:BD:D1  [redacted]            Infra  10    270 Mbit/s  90      ▂▄▆█  WPA1 WPA2
        A0:40:A0:A1:72:20  [redacted]            Infra  4     405 Mbit/s  82      ▂▄▆█  WPA2
        28:6C:07:6E:87:2F  [redacted]orangepi_5G Infra  149   540 Mbit/s  80      ▂▄▆_  WPA1 WPA2
        CA:50:E9:89:E2:44  [redacted]            Infra  1     130 Mbit/s  79      ▂▄▆_  WPA1 WPA2
        A0:40:A0:A1:72:31  [redacted]            Infra  100   405 Mbit/s  67      ▂▄▆_  WPA2
        D4:EE:07:08:A9:E0  [redacted]            Infra  4     130 Mbit/s  55      ▂▄__  WPA1 WPA2
        88:C3:97:49:25:13  [redacted]            Infra  6     130 Mbit/s  52      ▂▄__  WPA1 WPA2
        00:BD:82:51:53:C2  [redacted]            Infra  12    130 Mbit/s  49      ▂▄__  WPA1 WPA2
        C0:61:18:FA:49:37  [redacted]            Infra  149   270 Mbit/s  47      ▂▄__  WPA1 WPA2
        04:79:70:8D:0C:B8  [redacted]            Infra  153   270 Mbit/s  47      ▂▄__  WPA2
        04:79:70:FD:0C:B8  [redacted]            Infra  153   270 Mbit/s  47      ▂▄__  WPA2
        9C:A6:15:DD:E6:0C  [redacted]            Infra  10    270 Mbit/s  45      ▂▄__  WPA1 WPA2
        B4:0F:3B:45:D1:F5  [redacted]            Infra  48    270 Mbit/s  45      ▂▄__  WPA1 WPA2
        E8:CC:18:4F:7B:44  [redacted]            Infra  157   135 Mbit/s  45      ▂▄__  WPA1 WPA2
        B0:95:8E:D8:2F:ED  [redacted]            Infra  11    405 Mbit/s  39      ▂▄__  WPA1 WPA2
        C0:61:18:FA:49:36  [redacted]            Infra  11    270 Mbit/s  24      ▂___  WPA1 WPA2
root@orangepi:~#
```

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

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

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

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

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

```
orangepi@orangepi:~$ ip addr show wlan0
11: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 23:8c:d6:ae:76:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
       valid_lft 259192sec preferred_lft 259192sec
    inet6 240e:3b7:3240:c3a0:c401:a445:5002:ccdd/64 scope global dynamic noprefixroute
       valid_lft 259192sec preferred_lft 172792sec
    inet6 fe80::42f1:6019:a80e:4c31/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

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

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

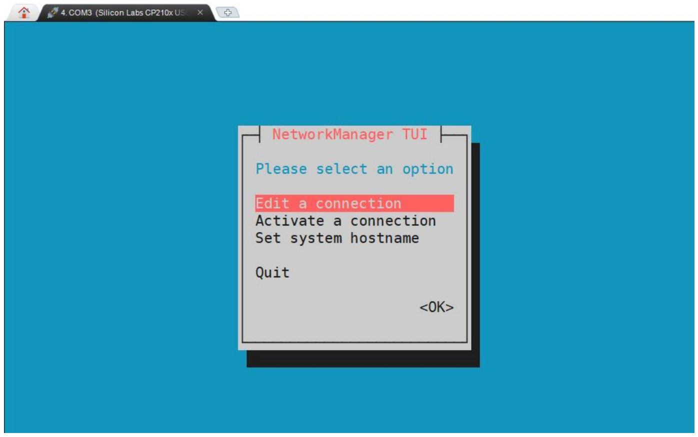
3. 6. 2. 2. Server version image connects to WIFI graphically

1) First log in to the Linux system, there are three ways:
   a. If the development board is connected with a network cable, you can **remotely log in to the Linux system through ssh**
   b. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the Linux system (please use MobaXterm for the serial port software, the graphical interface cannot be displayed using minicom)
   c. If the development board is connected to the HDMI display, you can log in to the Linux system through the terminal displayed on the HDMI

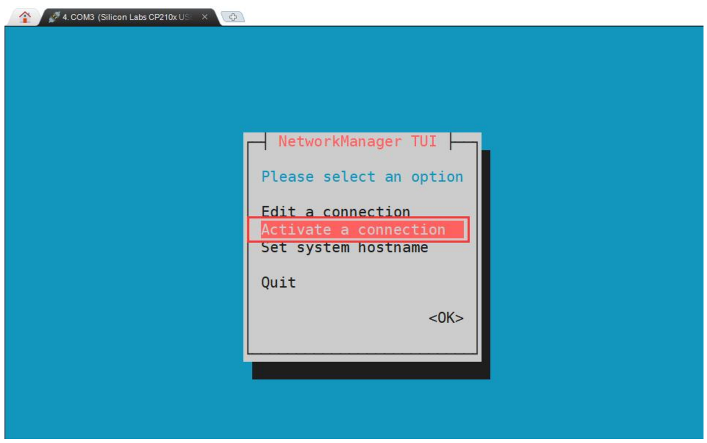
2) Then enter the nmtui command in the command line to open the wifi connection interface

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

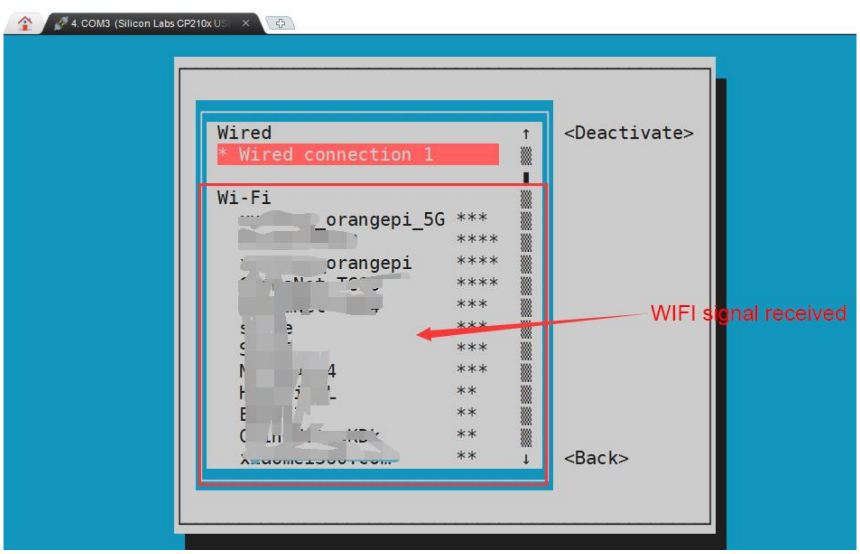
3) Enter the nmtui command to open the interface as shown below



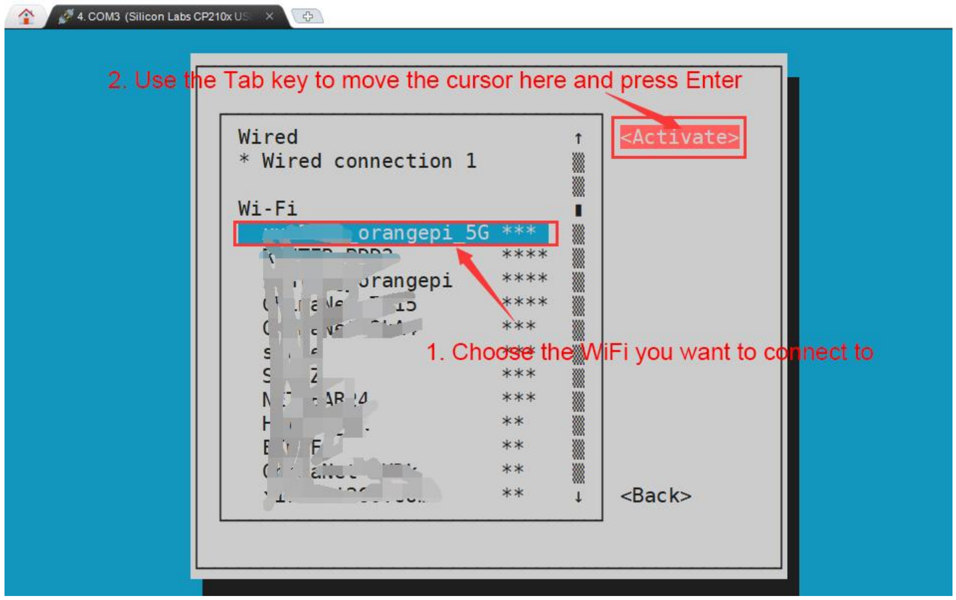
4) Select **Activate a connect** and press Enter



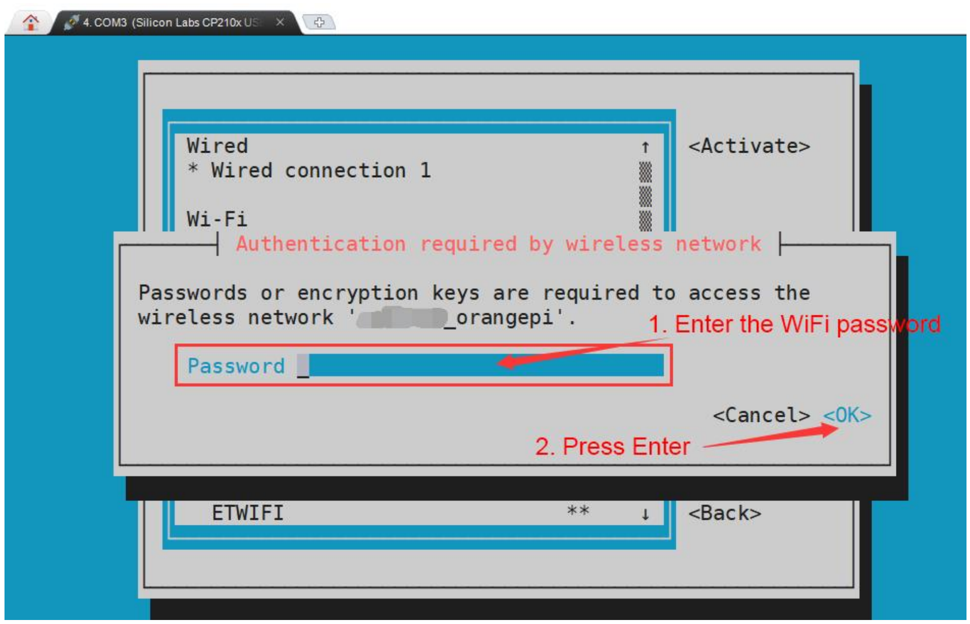
5) Then you can see all the searched WIFI hotspots



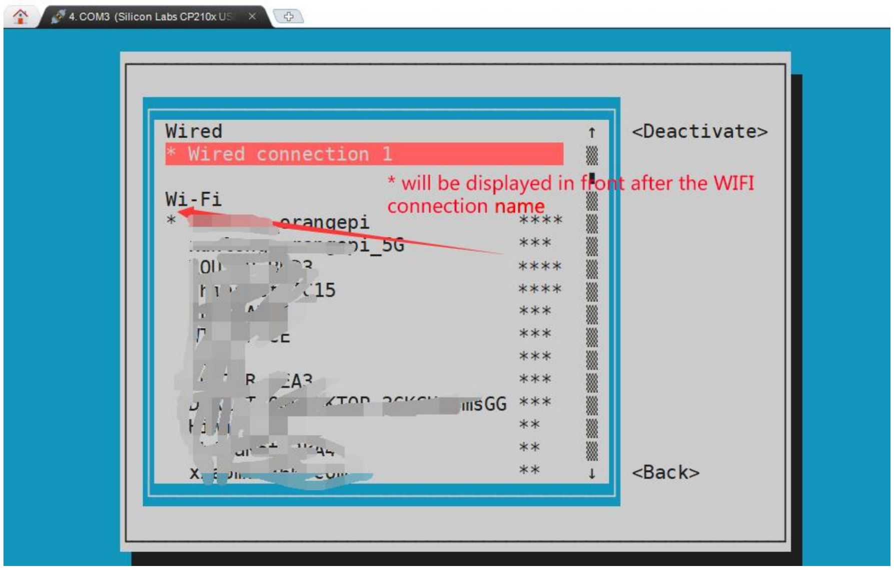
6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the cursor on **Activate** and press Enter



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



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



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

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

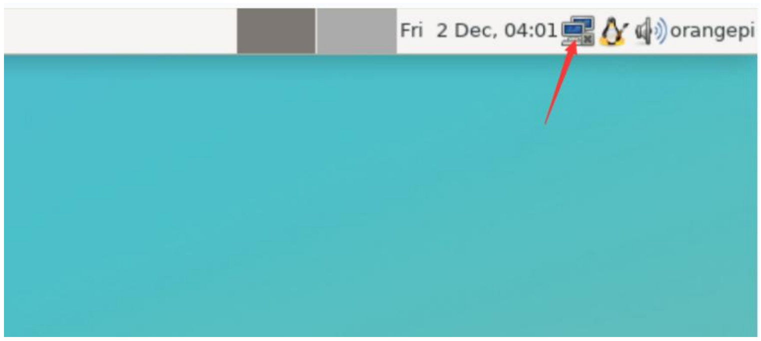
10) Use the **ping** command to test the connectivity of the wifi network. The **ping** command can be interrupted by pressing the **Ctrl+C** shortcut key

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

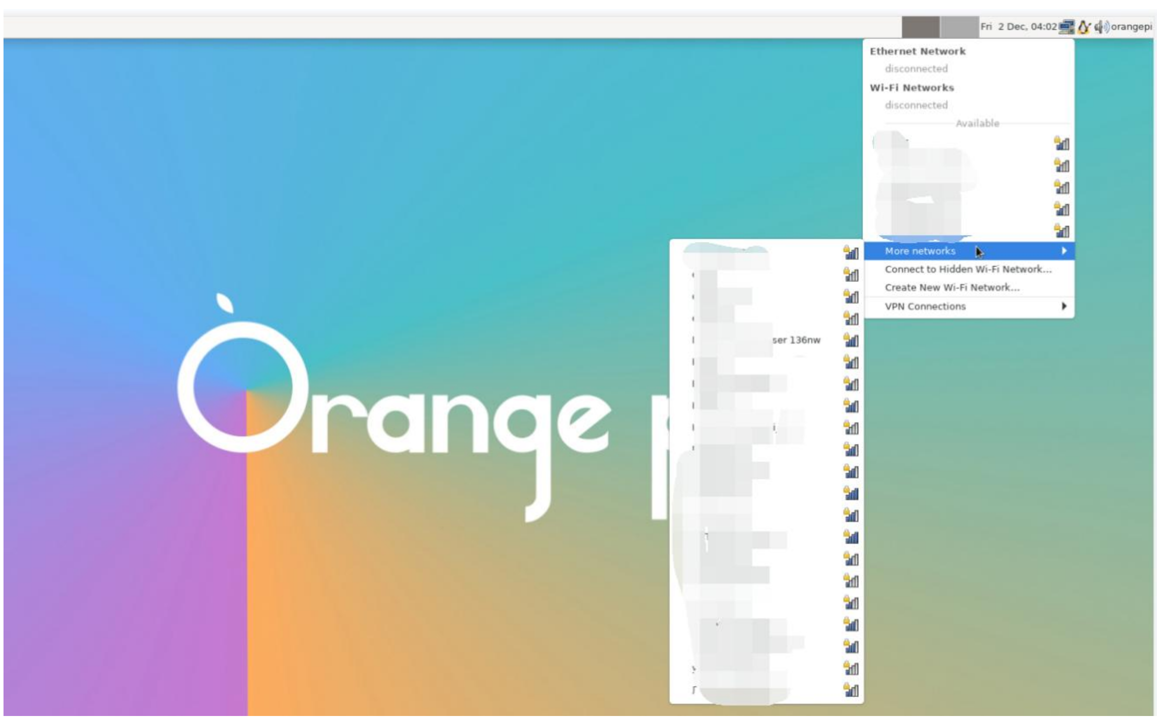
```
^C
--- www.orangepi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

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

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

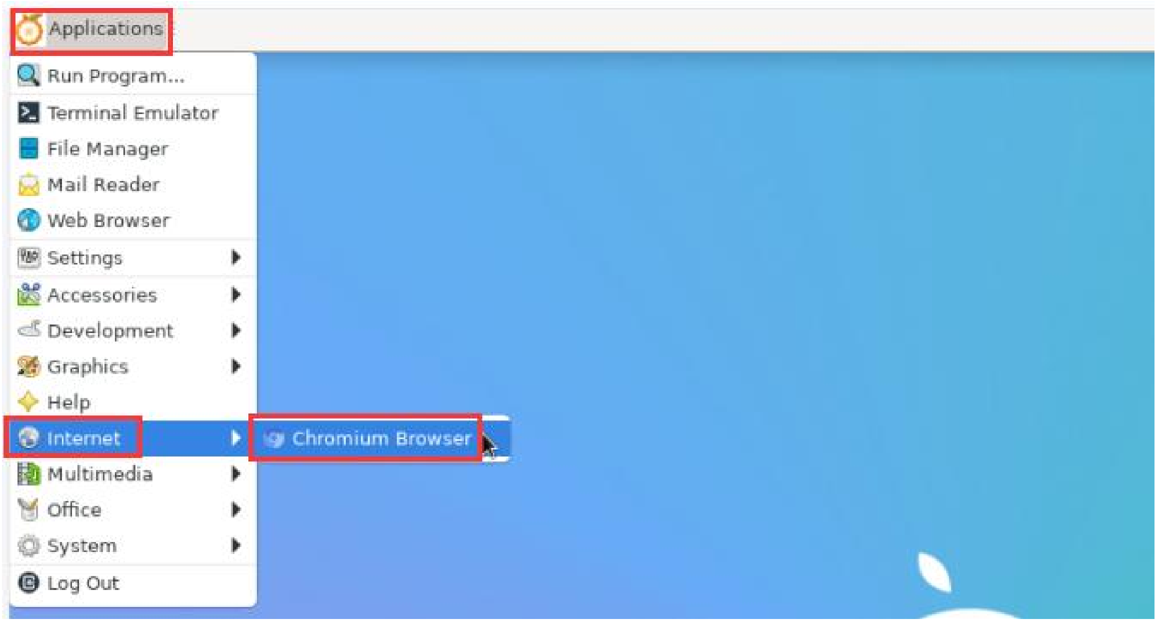


2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to.

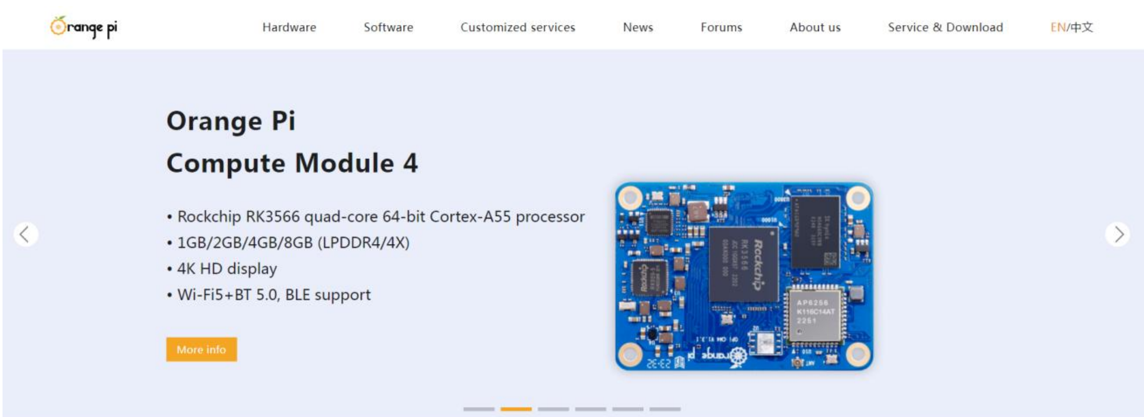


3) Then enter the password of the WIFI hotspot, and then click **Connect** to start connecting to WIFI

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



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



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

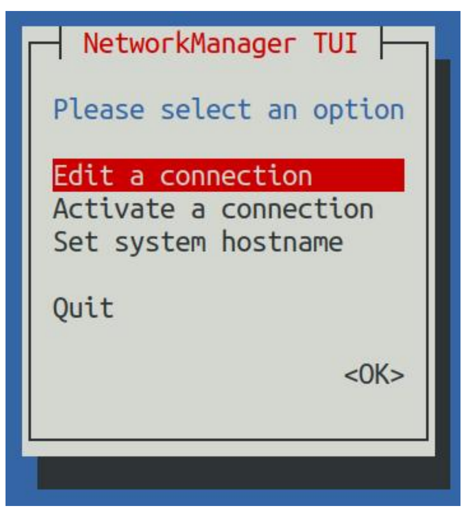
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file

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

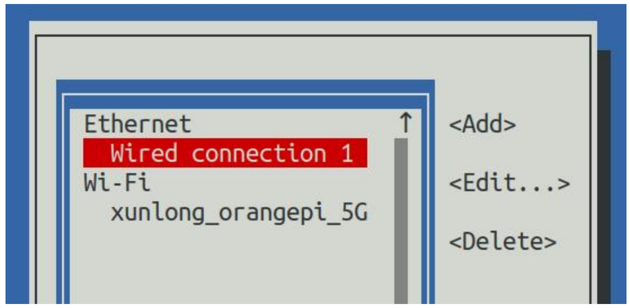
1) First run the nmtui command

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

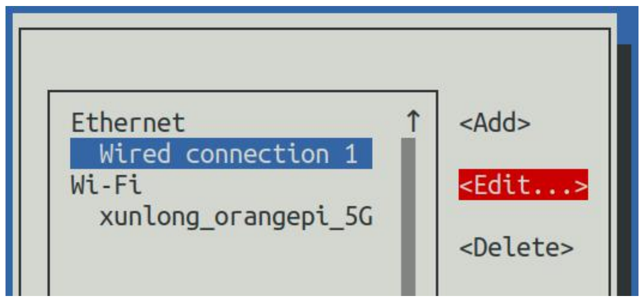
2) Then select **Edit a connection** and press Enter



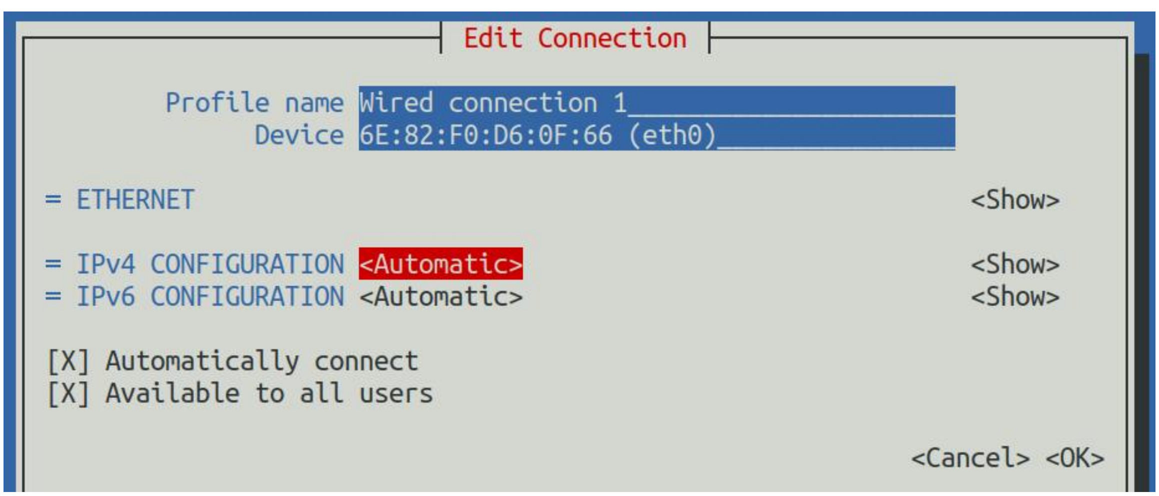
3) Then select the network interface for which a static IP address needs to be set. For example, to set the static IP address of the **Ethernet** interface, select **Wired connection 1**.



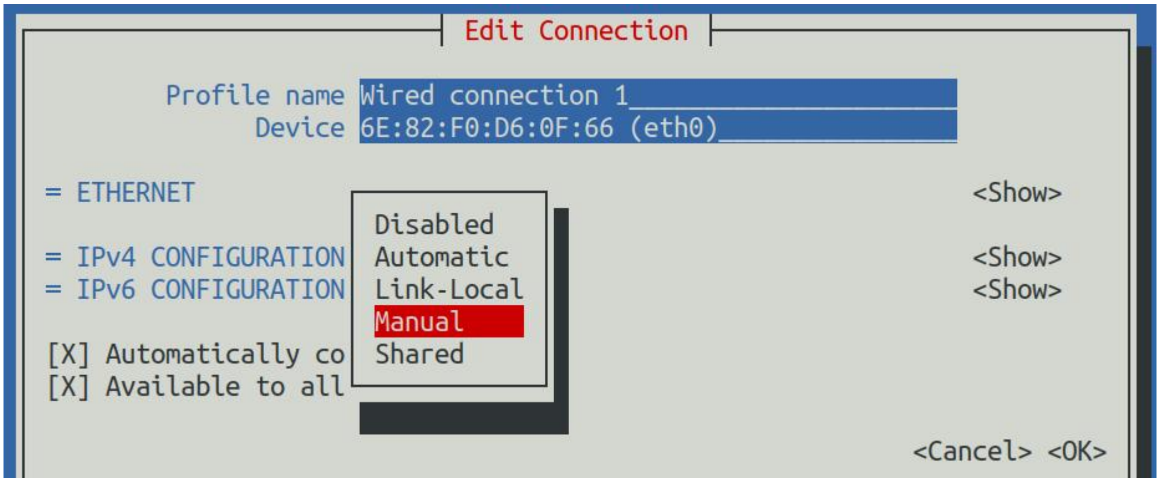
4) Then select **Edit** via the **Tab** key and press the Enter key



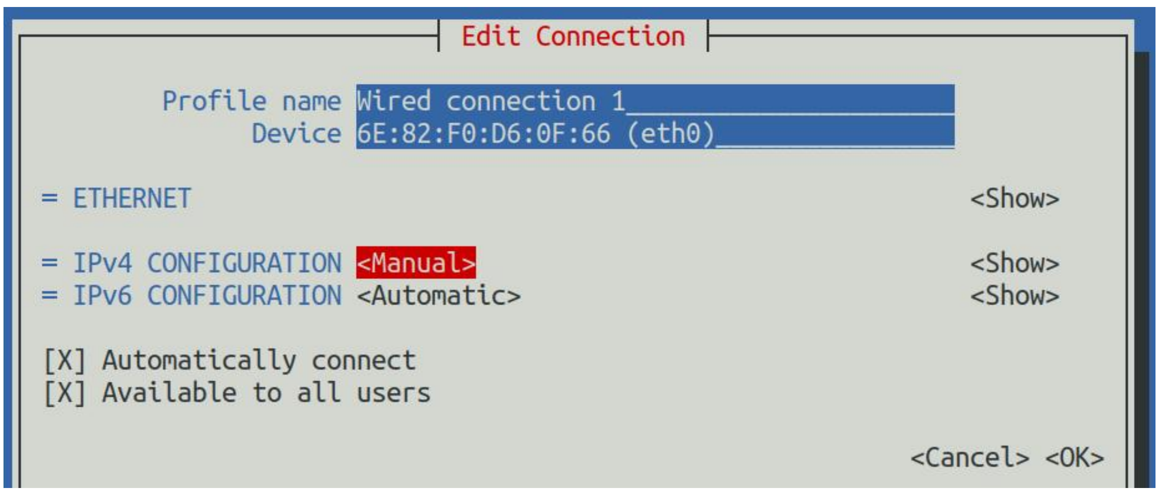
5) Then use the Tab key to move the cursor to the**<Automatic>** position shown in the figure below to configure IPv4



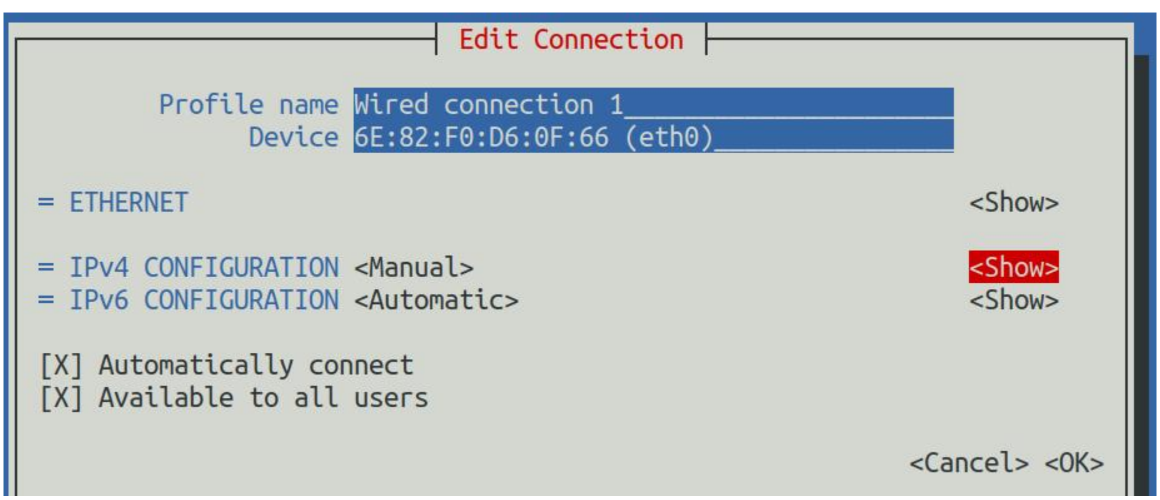
6) Then press Enter, select **Manual** through the up and down arrow keys, and then press Enter to confirm.



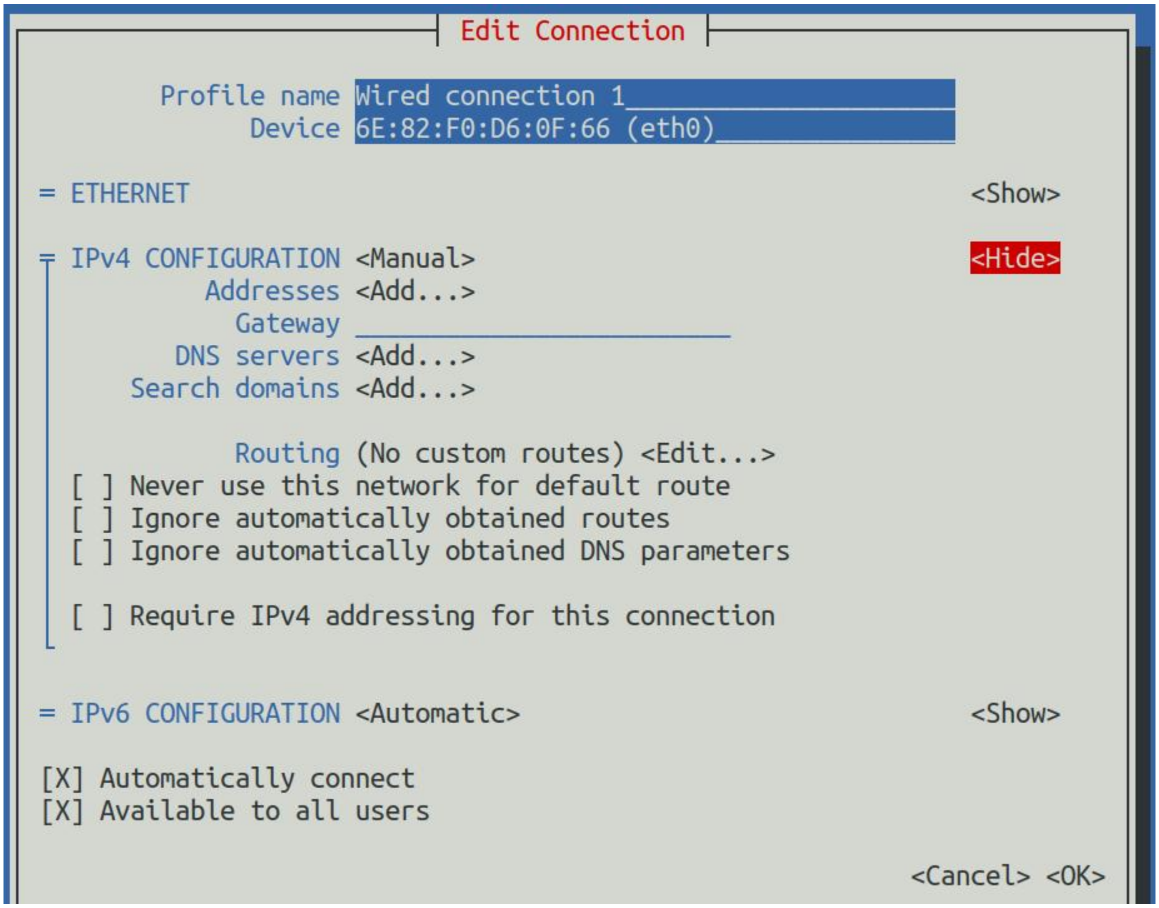
7) The display after selection is as shown below



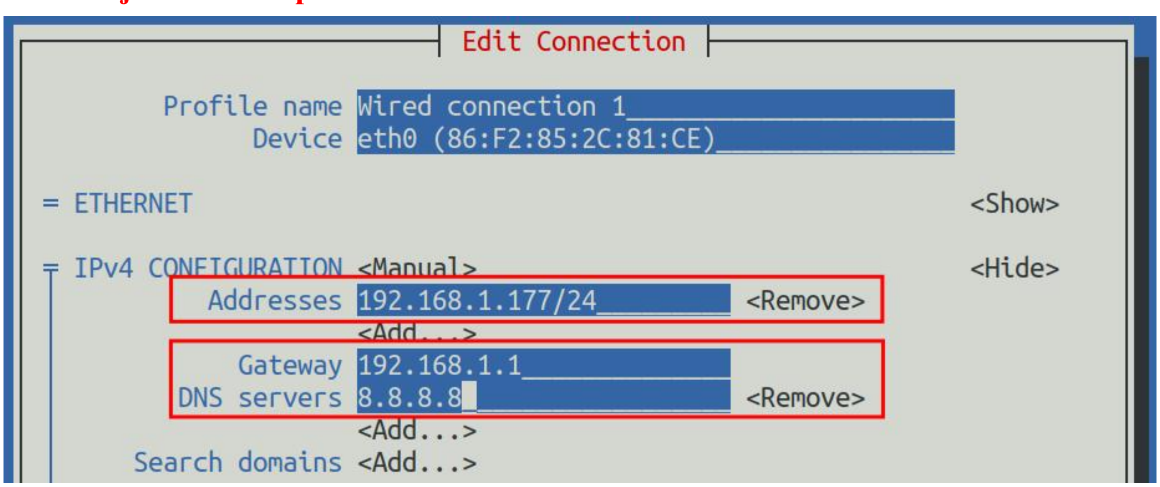
8) Then move the cursor to **<Show>** via the Tab key



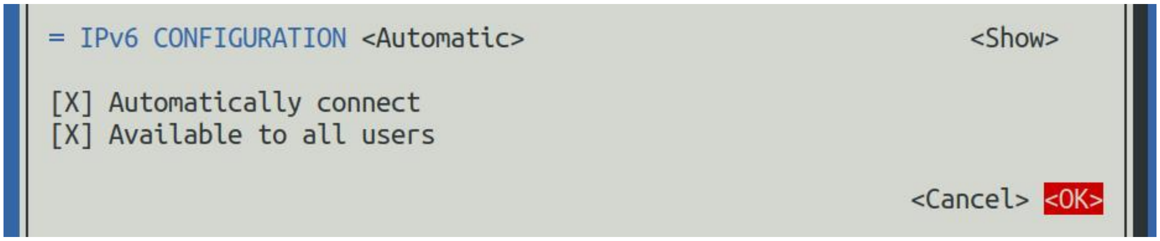
9) Then press Enter. After pressing Enter, the following setting interface will pop up.



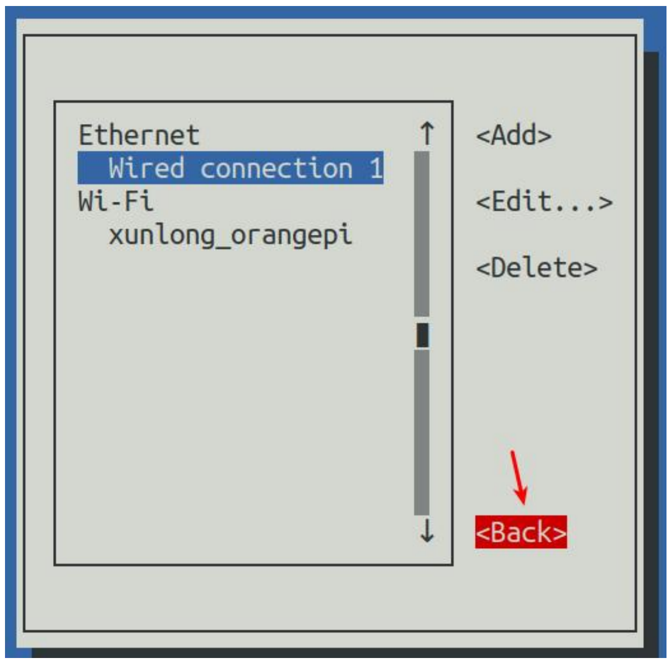
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address as shown in the figure below (there are many other setting options, please explore by yourself),**Please set it according to your specific needs. The value set in the picture below is just an example.**



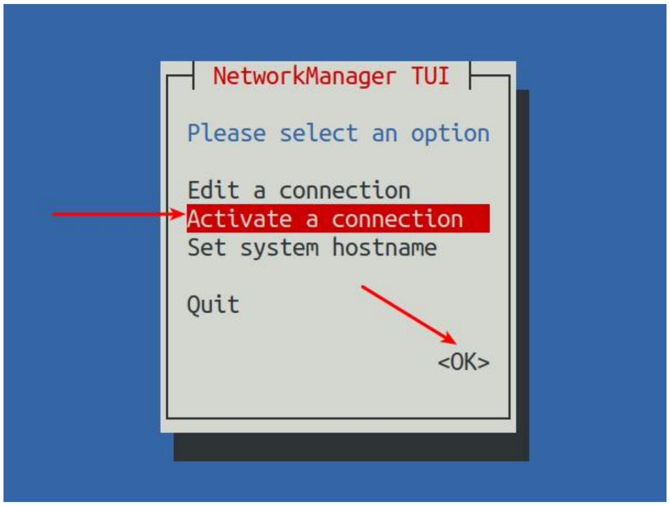
11) After setting, move the cursor to **<OK>** in the lower right corner, and then press Enter to confirm.



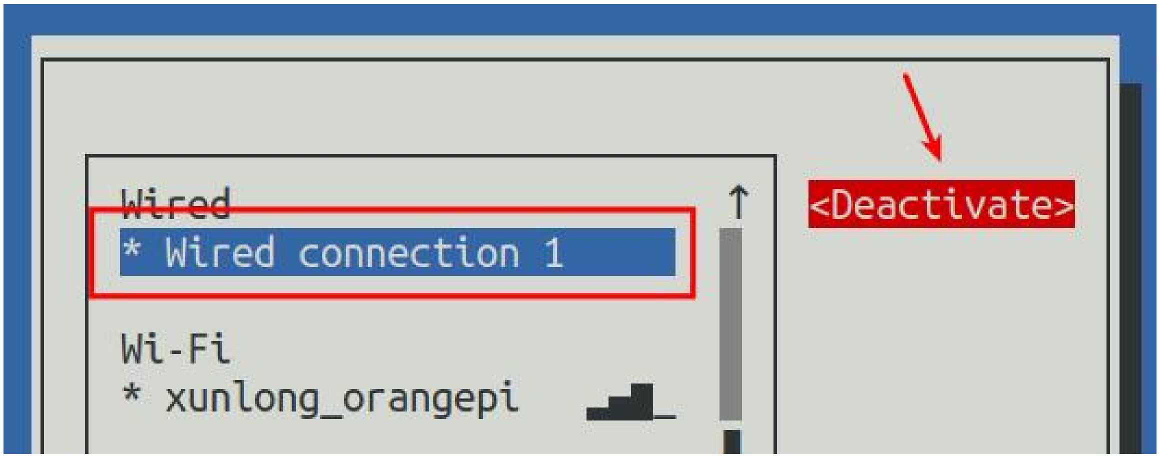
12) Then click **<Back>** to return to the previous level selection interface



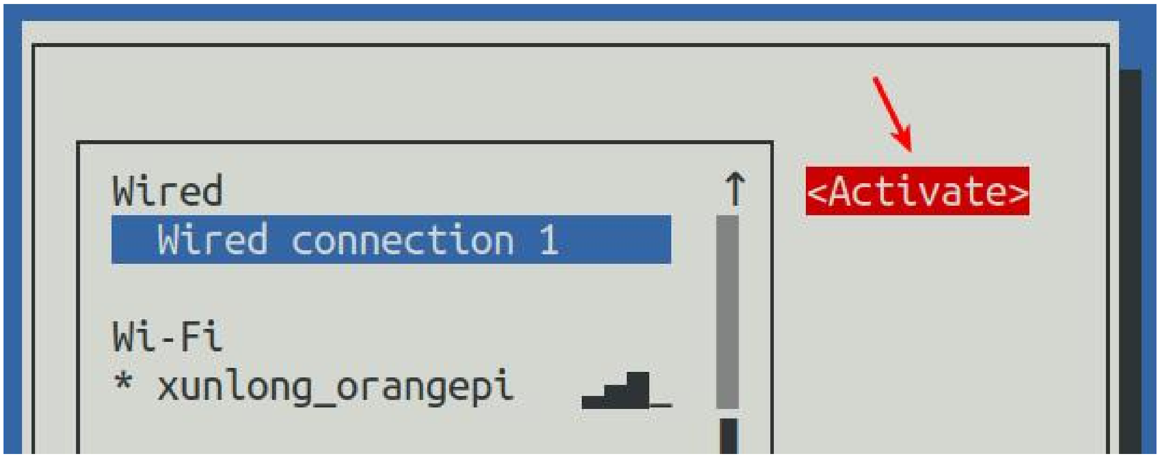
13) Then select **Activate a connection**, then move the cursor to **<OK>**, and finally click Enter



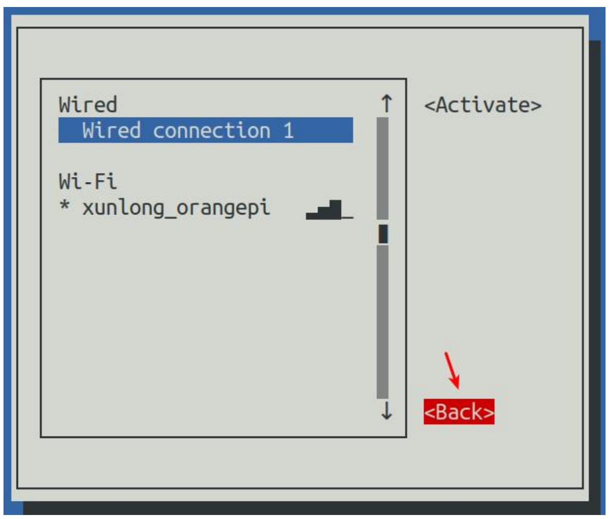
14) Then select the network interface that needs to be set, such as **Wired connection 1**, then move the cursor to **<Deactivate>**, and press Enter to disable **Wired connection 1**

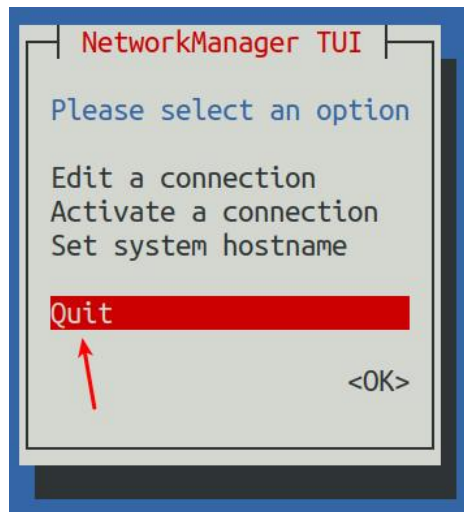


15) Then please do not move the cursor, and then press the Enter key to re-enable **Wired connection 1**, so that the static IP address set earlier will take effect



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





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

```
orangepi@orangepi:~$ ip addr show eth0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 5e:ac:14:a5:92:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.177/24 brd 192.168.1.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet6 241e:3b8:3240:c3a0:e269:8305:dc08:135e/64 scope global dynamic noprefixroute
       valid_lft 259149sec preferred_lft 172749sec
    inet6 fe80::957d:bbbe:4928:3604/64 scope link noprefixroute
```

```
        valid_lft forever preferred_lft forever
```

18) Then you can test the network connectivity to check whether the IP address is configured OK. The ping command can be interrupted by using the **Ctrl+C** shortcut key.

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

3. 6. 3. 2. Use the nmcli command to set a static IP address

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

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

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

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

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

3) Then enter the following command,

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

b. **ipv4.addresses** Behind is 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:~$ nmcli con mod "Wired connection 1"   \
ipv4.addresses "192.168.1.110"   \
ipv4.gateway "192.168.1.1"   \
ipv4.dns "8.8.8.8"   \
ipv4.method "manual"
```

4) Then restart the Linux system

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

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

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

3.6.4. Method to create WIFI hotspot through create_ap

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

https://github.com/oblique/create_ap

If you are using the latest image, the create_ap script has been pre-installed. You

can create a WIFI hotspot through the create_ap command. The basic command format of create_ap is as follows:

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

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

3.6.4.1. create_ap method to create WIFI hotspot in NAT mode

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

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

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

```
orangepi@orangepi:~$ sudo create_ap --no-virt -m nat wlan0 eth0 orangepi orangepi
Config dir: /tmp/create_ap.wlan0.conf.Ji9Coeqo
PID: 5526
Network Manager found, set wlan0 as unmanaged device... DONE
Sharing Internet using method: nat
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.Ji9Coeqo/hostapd_ctrl
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

3） At this time, take out your mobile phone and find the WIFI hotspot named **orangepi** created by the development board in the searched WIFI list. Then you can click **orangepi** to connect to the hotspot. The password is **orangepi** set above

4）The display after successful connection is as shown below

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

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

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

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

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

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

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

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

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

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

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

3.6.4.2. create_ap method to create WIFI hotspot in bridge mode

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

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

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

```
orangepi@orangepi:~$ sudo create_ap --no-virt -m bridge wlan0 eth0 orangepi orangepi
[sudo] password for orangepi:
Config dir: /tmp/create_ap.wlan0.conf.hXrfLdof
PID: 8372
Network Manager found, set wlan0 as unmanaged device... DONE
Sharing Internet using method: bridge
Create a bridge interface... br0 created.
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.hXrfLdof/hostapd_ctrl
wlan0: interface state UNINITIALIZED->ENABLED
```

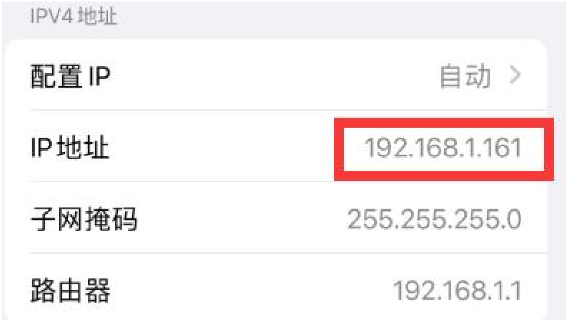
3) At this time, take out your mobile phone and find the WIFI hotspot named **orangepi** created by the development board in the searched WIFI list. Then you can click **orangepi** to connect to the hotspot. The password is **orangepi** set above.

4）The display after successful connection is as shown below

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

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

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



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

Note that in the following command, Debian12 needs to modify eth0 to end1.

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

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

Note that in the following command, Debian12 needs to modify eth0 to end1.

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

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

3. 7. SSH remote login development board

Linux systems enable ssh remote login by default and allow root users to log in

to the system. Before ssh login, you first need to ensure that the Ethernet or wifi network is connected, and then use the ip addr command or obtain the IP address of the development board by checking the router.

3. 7. 1. SSH remote login development board under Ubuntu

1) Obtain the IP address of the development board

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

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

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

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

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

If ssh fails to log in to the Linux system normally, first check whether the IP address of the development board can be pinged. If there is no problem with the ping, you can log in to the Linux system through the serial port or HDMI display, and then enter the following command on the development board and try again. Is it

possible to connect:

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

If it still doesn't work, please try again by resetting the system.

3. 7. 2. SSH remote login development board under Windows

1) First obtain the IP address of the development board

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

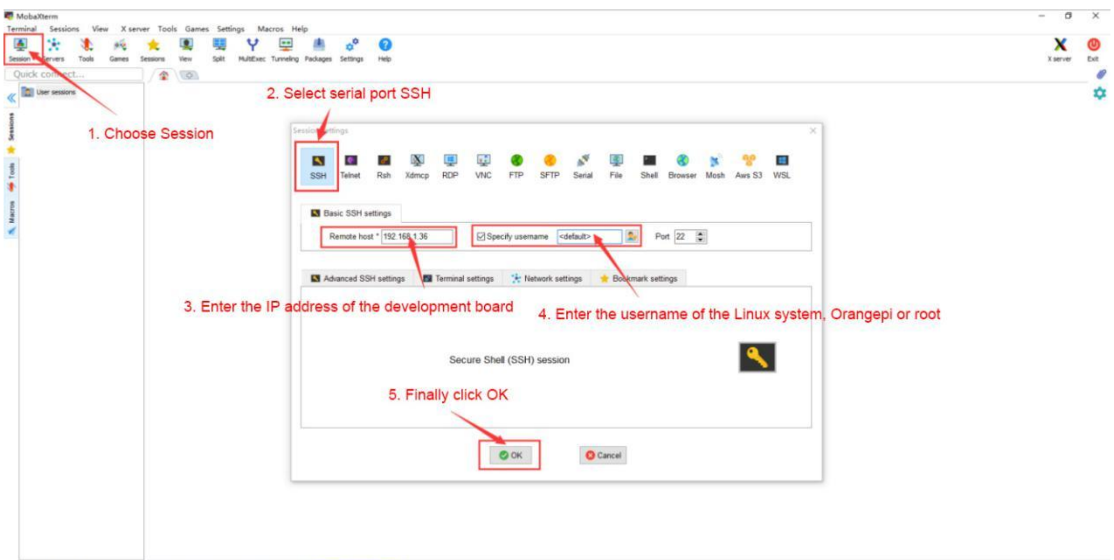
a. Open **Session**

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

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

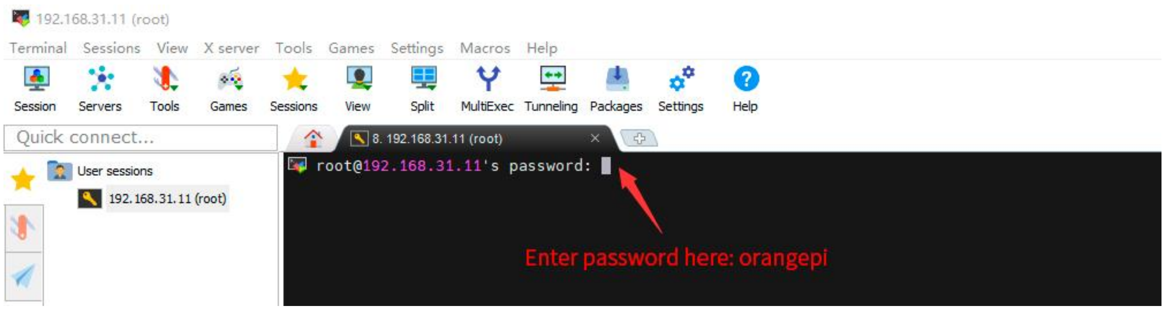
d. Then enter the username **root** or **orangepi** of the Linux system in **Specify username**

e. Finally click **OK**

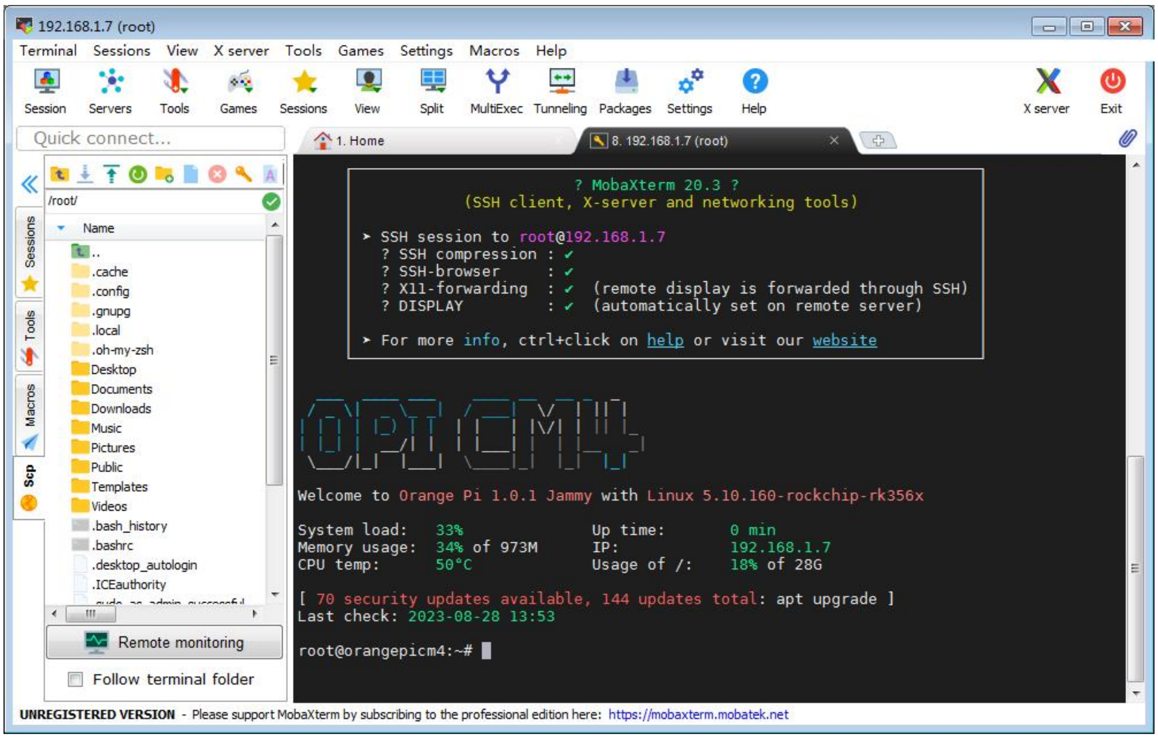


3) Then you will be prompted to enter a password. The default passwords for root and orangepi users are orangepi

Note that when entering a password, the specific content of the entered password will not be displayed on the screen. Please do not think that there is something wrong. Just press Enter after entering it.



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



3. 8. Method of uploading files to the development board Linux system

3. 8. 1. How to upload files from Ubuntu PC to development board Linux system

3. 8. 1. 1. How to upload files using scp command

1) Use the scp command to upload files from the Ubuntu PC to the Linux system of the

development board. The specific commands are as follows

a. **file_path：** Need to be replaced with the path of the file to be uploaded
b. **orangepi：** This is the username of the Linux system of the development board. It can also be replaced with another one, such as root.
c. **192.168.xx.xx:** It is the IP address of the development board, please modify it according to the actual situation
d. **/home/orangepi:** The path in the Linux system of the development board can also be modified to other paths

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

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

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

3) There are more usages of scp, please use the following command to view the man manual

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

3. 8. 1. 2. How to upload files using filezilla

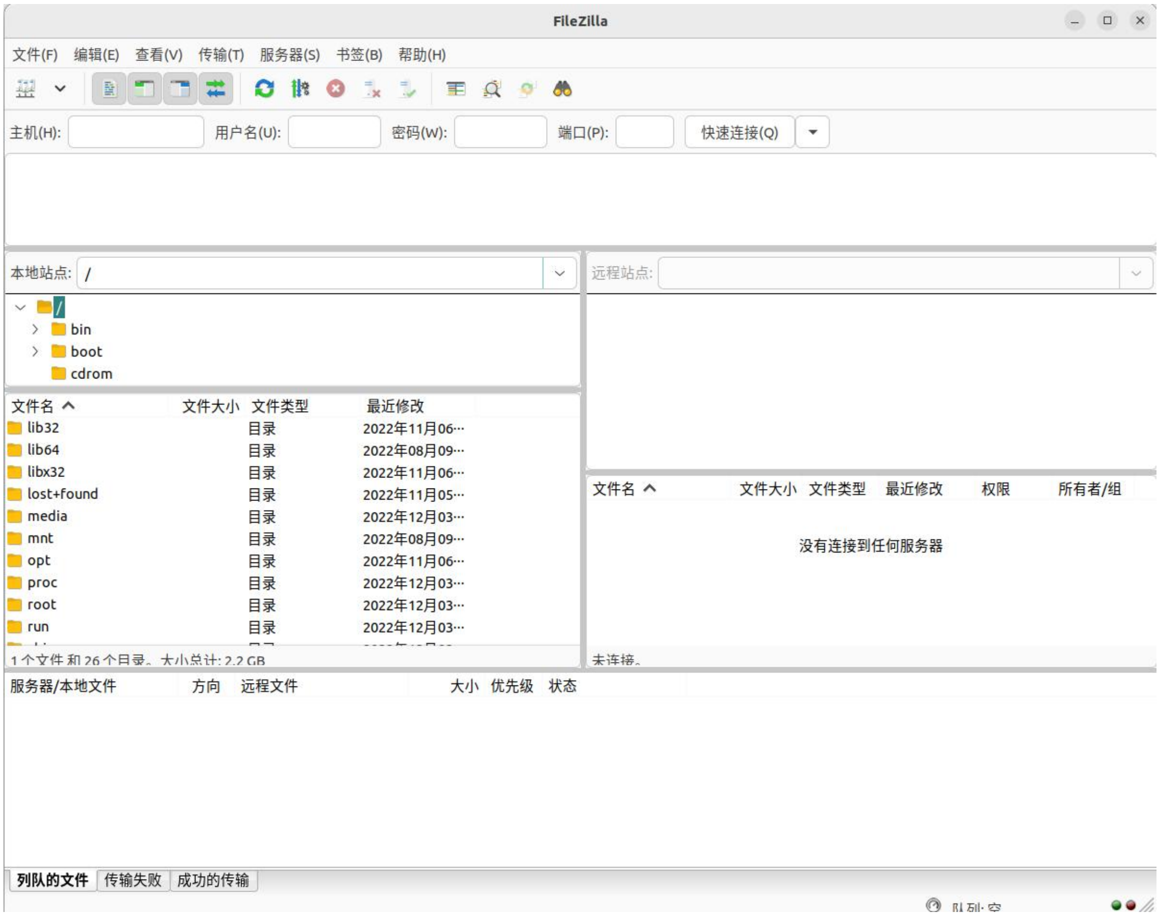
1) First install filezilla in Ubuntu PC

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

2) Then use the following command to open filezilla

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

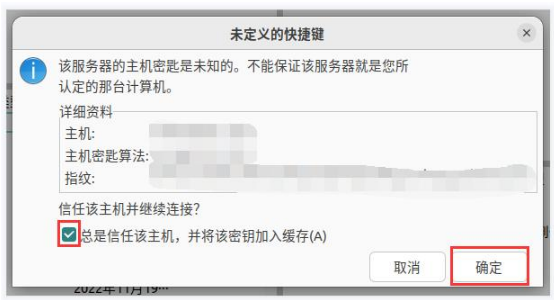
3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.



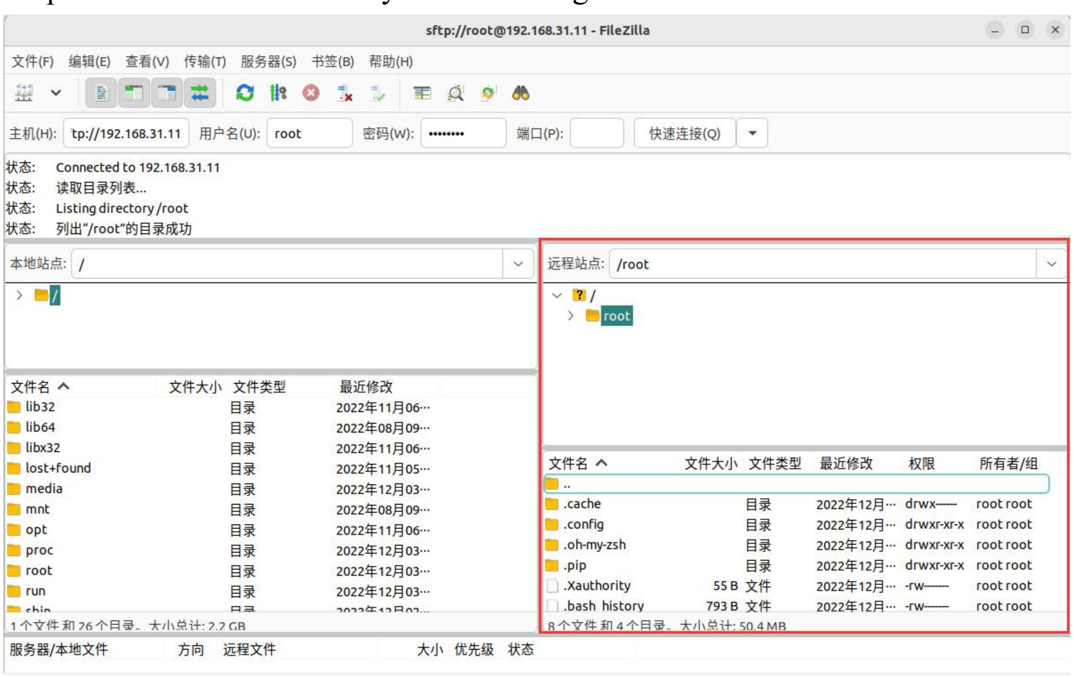
4) The method of connecting the development board is as shown in the figure below

5) Then choose to **save the password**, and then click **OK**

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

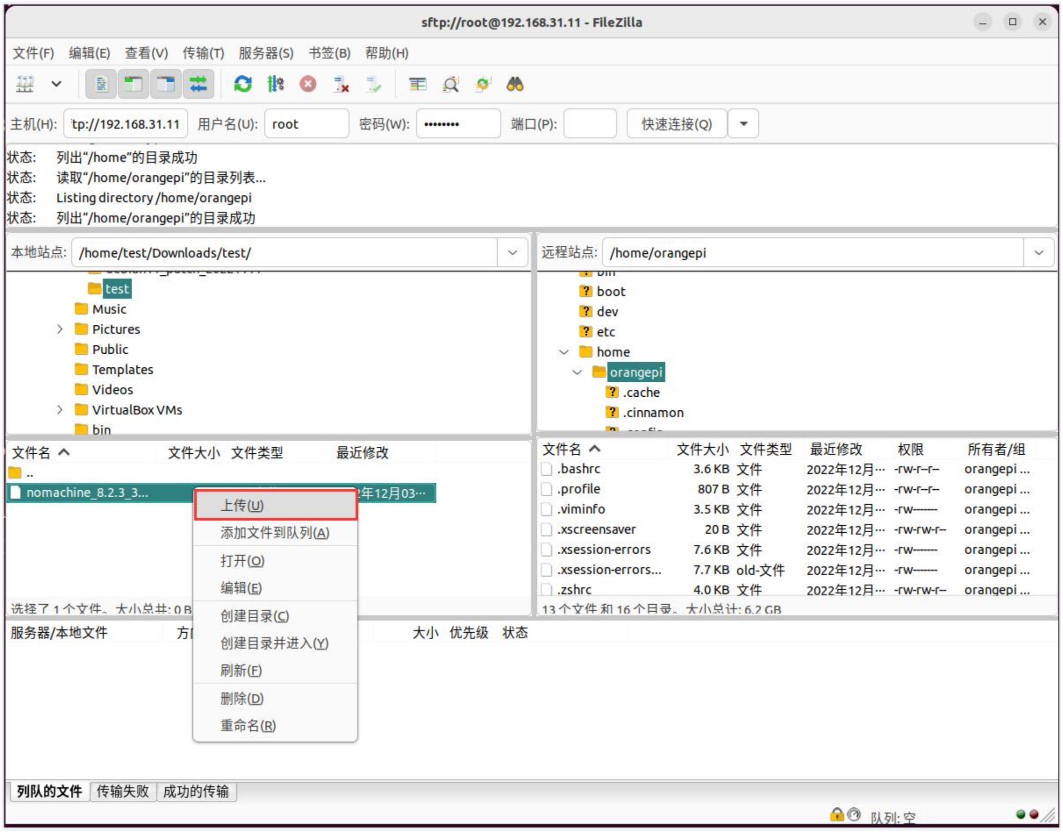


7) After the connection is successful, you can see the directory structure of the development board's Linux file system on the right side of the filezilla software.



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

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



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

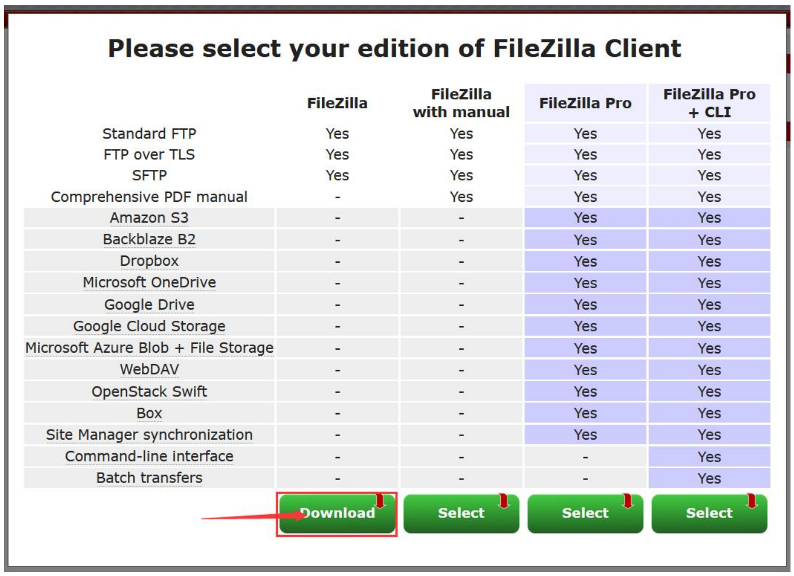
10) The method of uploading a folder is the same as the method of uploading a file, so I won't go into details here.

3. 8. 2. The method of uploading files to the Linux system of the development board in Windows PC

3. 8. 2. 1. How to upload files using filezilla

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

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



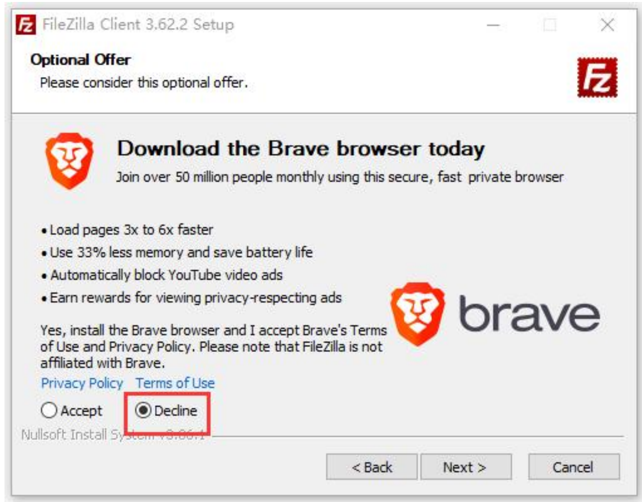
Please select your edition of FileZilla Client

	FileZilla	FileZilla with manual	FileZilla Pro	FileZilla Pro + CLI
Standard FTP	Yes	Yes	Yes	Yes
FTP over TLS	Yes	Yes	Yes	Yes
SFTP	Yes	Yes	Yes	Yes
Comprehensive PDF manual	-	Yes	Yes	Yes
Amazon S3	-	-	Yes	Yes
Backblaze B2	-	-	Yes	Yes
Dropbox	-	-	Yes	Yes
Microsoft OneDrive	-	-	Yes	Yes
Google Drive	-	-	Yes	Yes
Google Cloud Storage	-	-	Yes	Yes
Microsoft Azure Blob + File Storage	-	-	Yes	Yes
WebDAV	-	-	Yes	Yes
OpenStack Swift	-	-	Yes	Yes
Box	-	-	Yes	Yes
Site Manager synchronization	-	-	Yes	Yes
Command-line interface	-	-	-	Yes
Batch transfers	-	-	-	Yes
	Download	Select	Select	Select

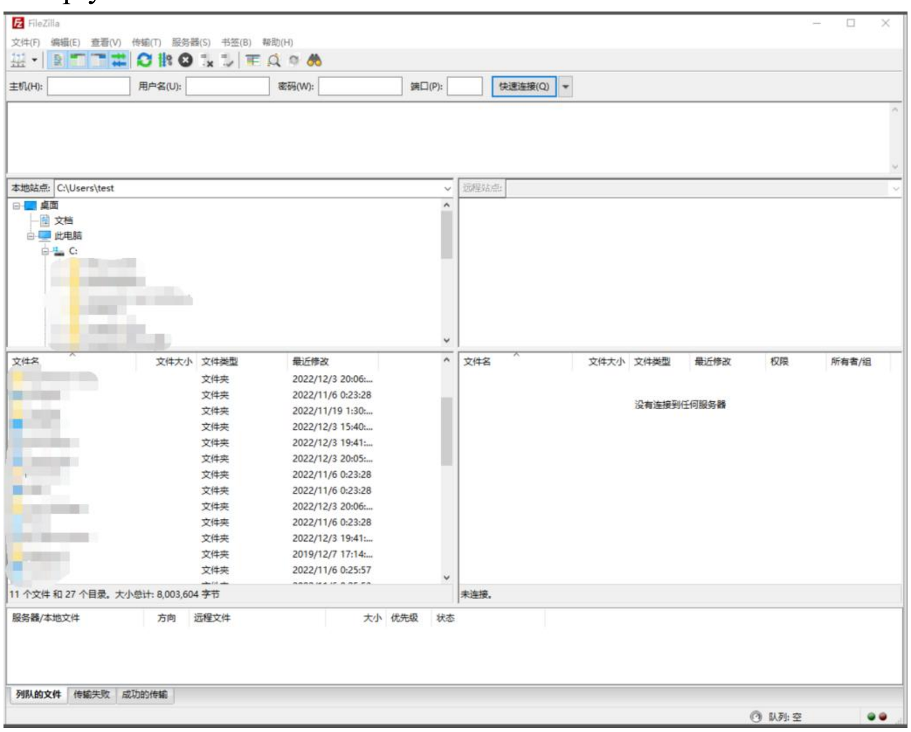
2) The downloaded installation package is as shown below, then double-click to install it directly

FileZilla_Server_1.5.1_win64-setup.exe

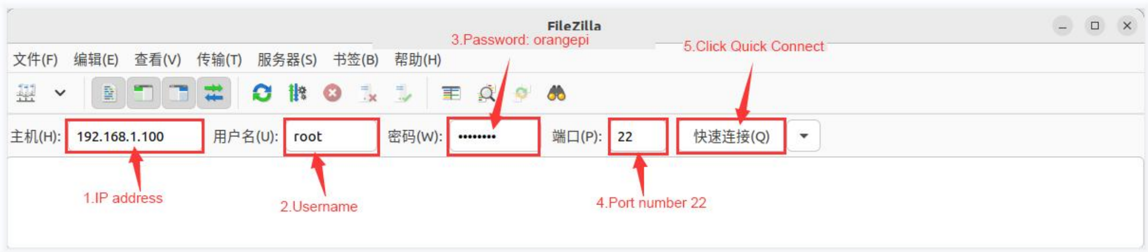
During the installation process, please select **Decline** on the following installation interface, and then select **<Next>**



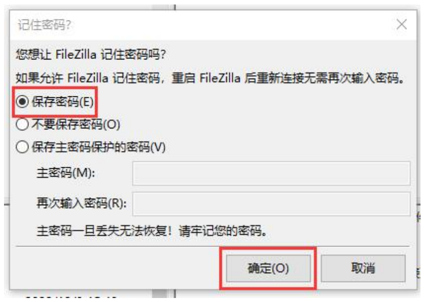
3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.



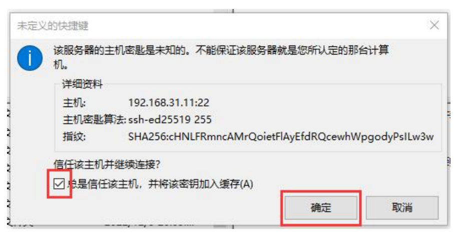
4) The method of connecting the development board is shown in the figure below：



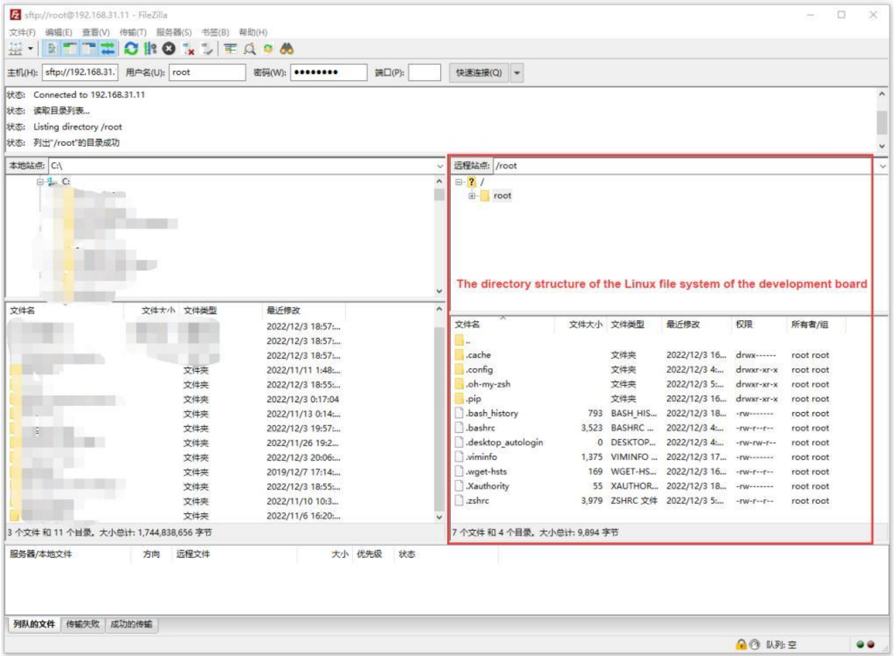
5) Then choose to **save the password,** and then click **OK**



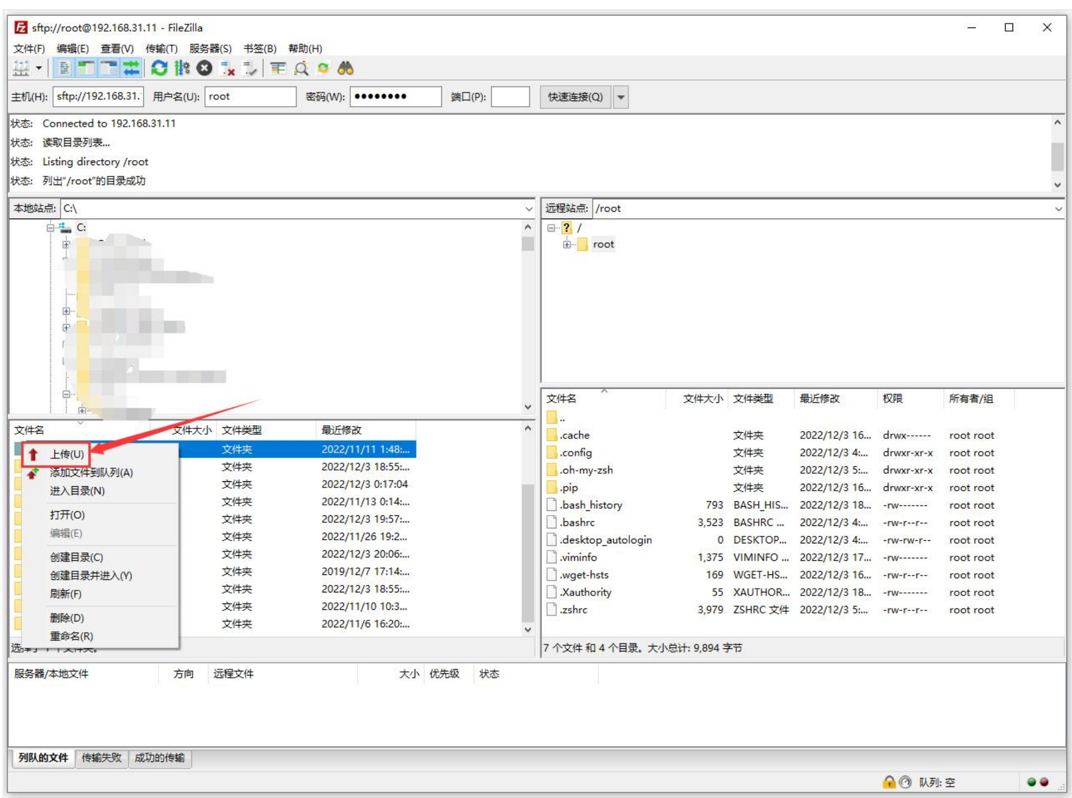
6) Then select **Always trust this host** and click **OK**



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



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



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

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

3. 9. HDMI test

3. 9. 1. HDMI display test

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

2) After starting the Linux system, if there is image output on the HDMI display, it means that the Micro HDMI interface is working normally.

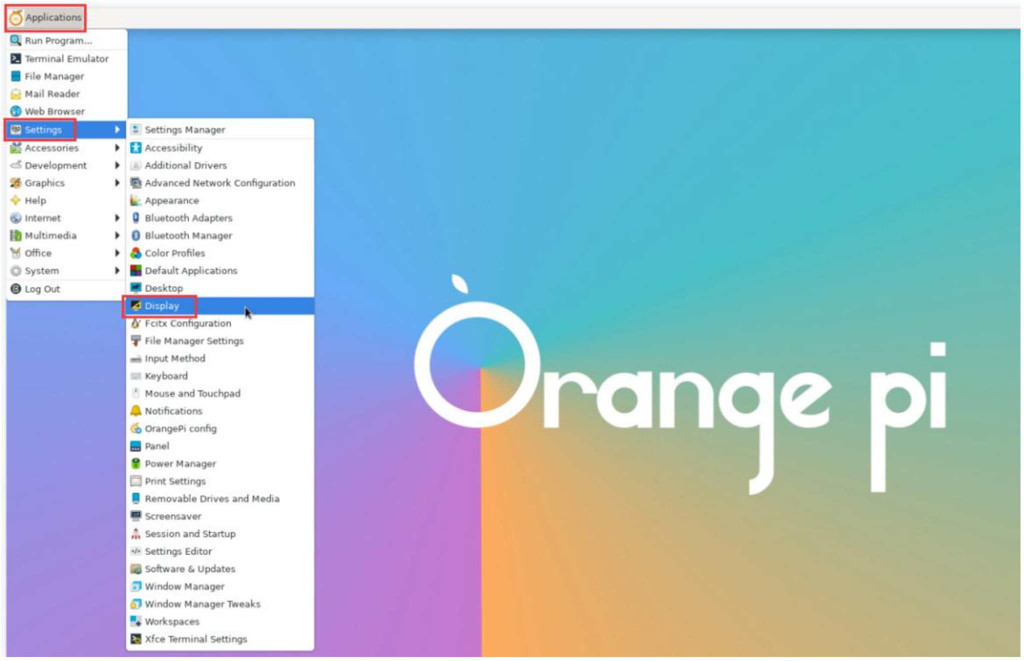
Note that although many laptops have HDMI interfaces, the HDMI interface of the laptop generally only has the output function and does not have the HDMI in function, which means that the HDMI output of other devices cannot be displayed on the laptop screen.

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

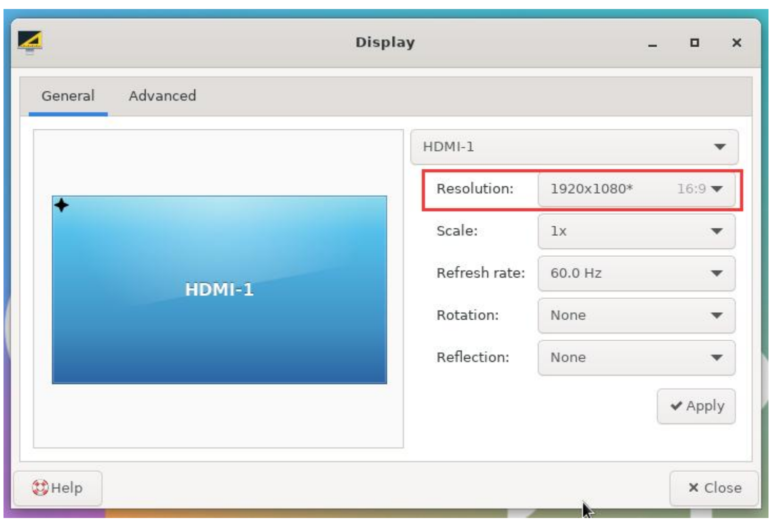
When HDMI does not display, please first check whether the Micro HDMI cable is plugged in tightly. After confirming that the wiring is OK, you can try a different screen to see if there is a display.

3. 9. 2. HDMI resolution setting method

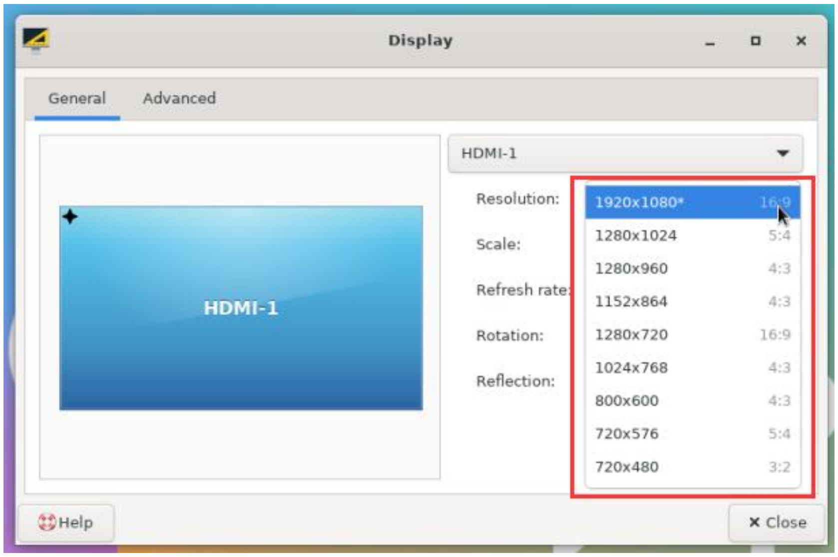
1) First open **Display** in **Settings**



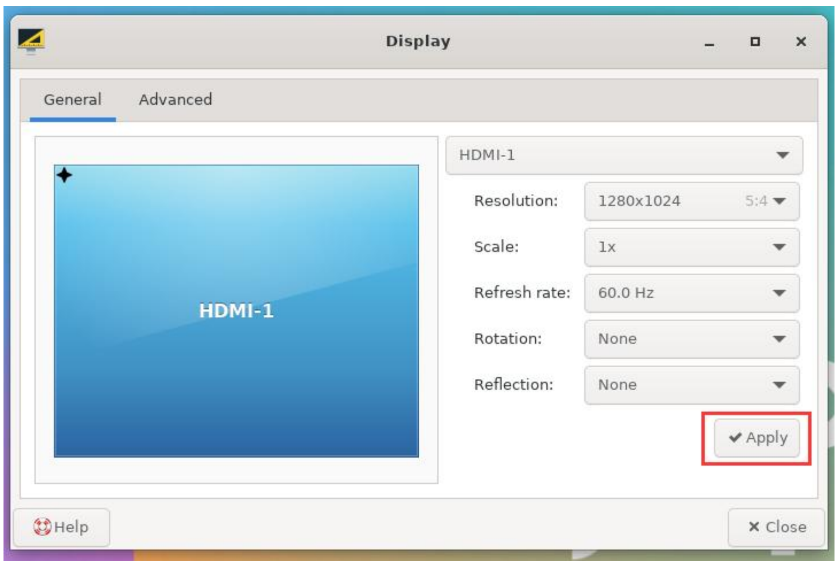
2) Then you can see the current resolution of the system



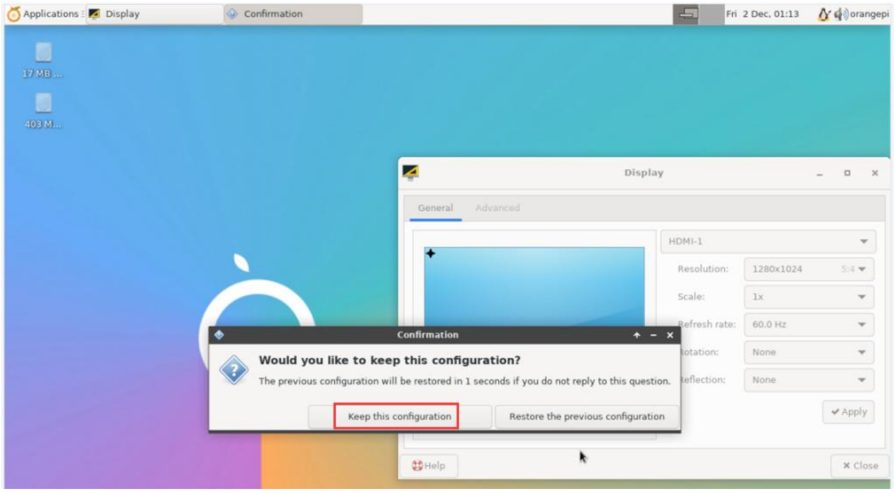
3) Click the drop-down box of Resolution to see all resolutions currently supported by the display



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



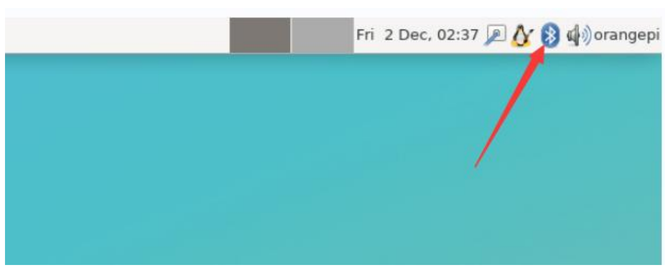
5) After the new resolution is set, select **Keep the configuration**



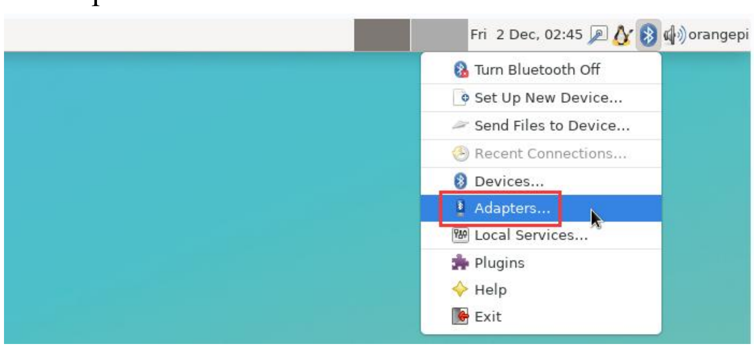
3. 10. How to use Bluetooth

3. 10. 1. Test method for desktop image

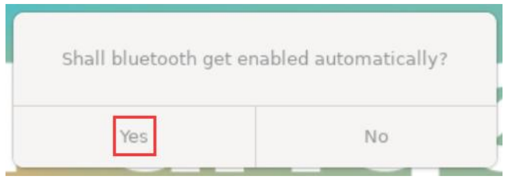
1) Click the Bluetooth icon in the upper right corner of the desktop



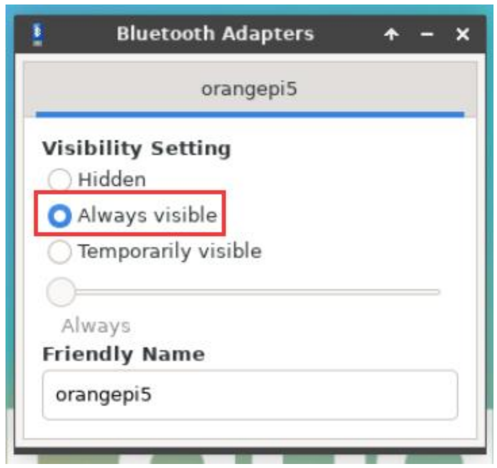
2) Then select the adapter



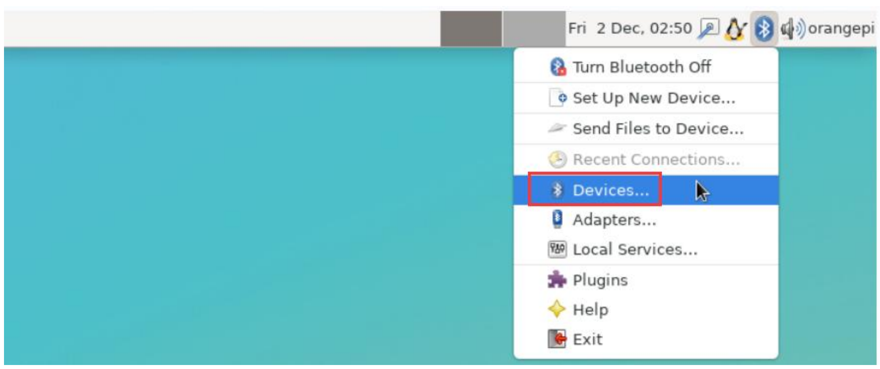
3) If prompted with the following interface, please select **Yes**.



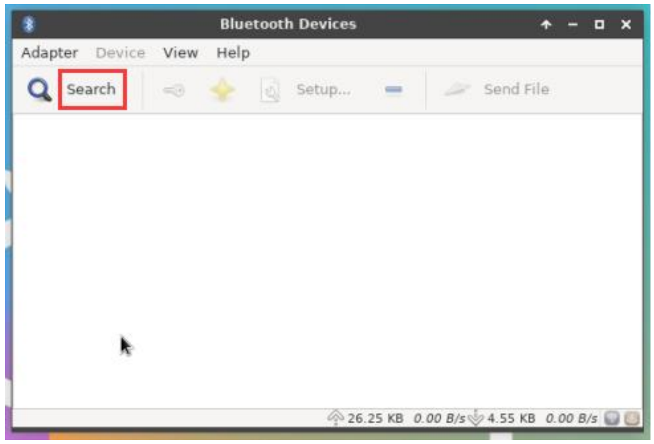
4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter setting interface, and then close it



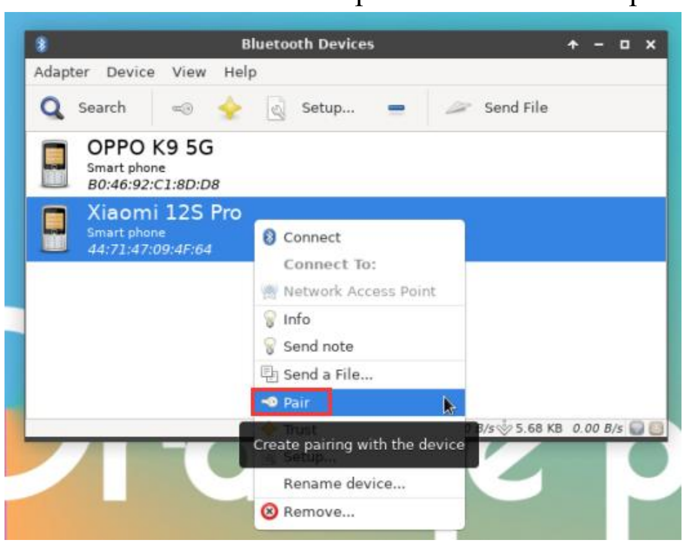
5) Then open the configuration interface of the Bluetooth device



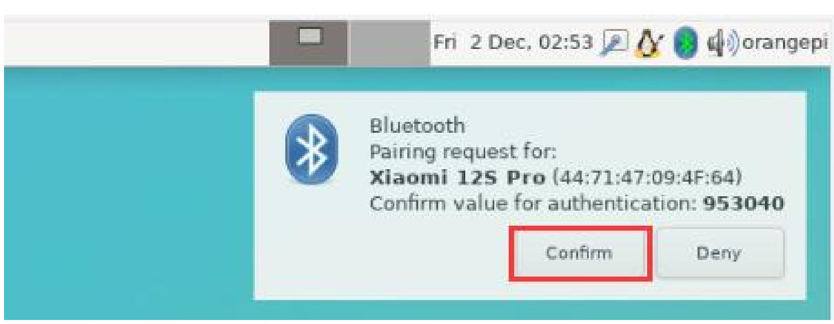
6) Click **Search** to start scanning for surrounding Bluetooth devices



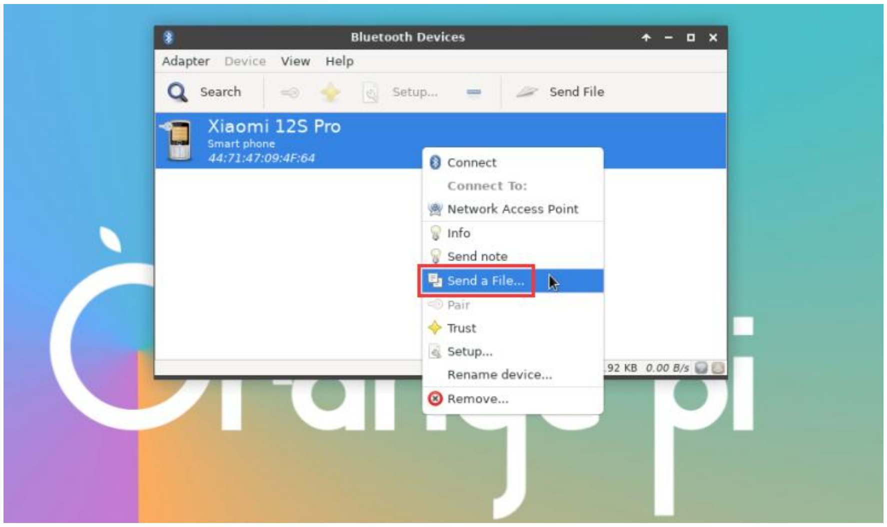
7) Then select the Bluetooth device you want to connect to, and then click the right mouse button to pop up the operation interface for this Bluetooth device, select **Pair** to start pairing, and the demonstration here is to pair with an Android phone



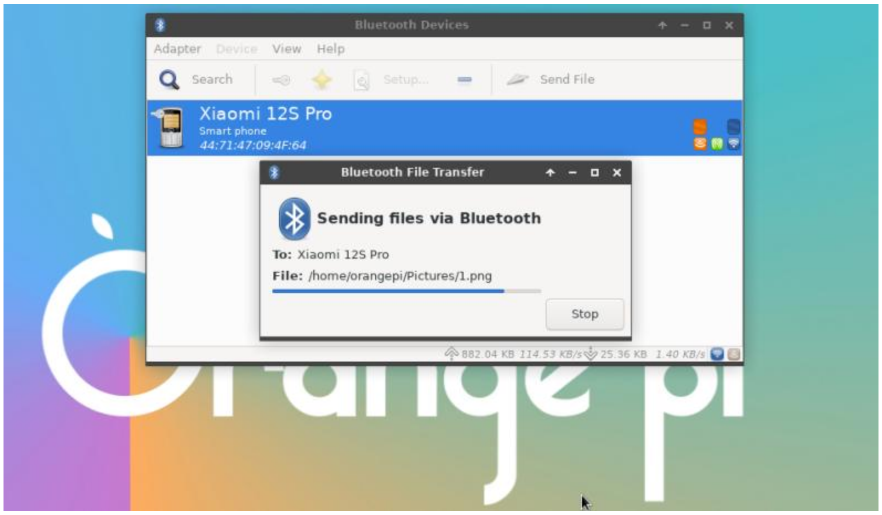
8) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop. Just select **Confirm** to confirm. At this time, you also need to confirm on the mobile phone.



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



10) The interface for sending pictures is as follows



3. 11. USB interface test

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

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

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

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

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

3. 11. 2. Connect USB storage device to test

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

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

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

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

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

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

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

3. 11. 3. USB wireless network card test

The usable USB wireless network cards that have been **tested** so far are as follows. Please test other types of USB wireless network cards by yourself. If they cannot be used, you need to transplant the corresponding USB wireless network card driver。

serial number	model	
1	RTL8723BU support 2.4G WIFI+BT4.0	

2	RTL8811 support 2.4G +5G WIFI	

3. 11. 3. 1. RTL8723BU test

1) First insert the RTL8723BU wireless network card module into the USB interface of the development board

2) Then the Linux system will automatically load the RTL8723BU bluetooth and WIFI-related kernel modules, through the lsmod command, you can see that the following kernel modules have been automatically loaded

```
orangepi@orangepi:~$ lsmod
Module                  Size  Used by
rfcomm                 57344  16
rtl8xxxu              106496  0
rtk_btusb              61440  0
```

3) Through the dmesg command, you can see the loading information of the RTL8723BU module

```
orangepi@orangepi:~$ dmesg
......
[   83.438901] usb 2-1: new high-speed USB device number 2 using ehci-platform
[   83.588375] usb 2-1: New USB device found, idVendor=0bda, idProduct=b720,
bcdDevice= 2.00
[   83.588403] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   83.588422] usb 2-1: Product: 802.11n WLAN Adapter
[   83.588443] usb 2-1: Manufacturer: Realtek
[   83.588460] usb 2-1: SerialNumber: 00e04c000001
[   83.601974] Bluetooth: hci0: RTL: examining hci_ver=06 hci_rev=000b lmp_ver=06
lmp_subver=8723
[   83.603894] Bluetooth: hci0: RTL: rom_version status=0 version=1
[   83.603920] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_fw.bin
[   83.610108] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_config.bin
```

```
[   83.611274] Bluetooth: hci0: RTL: cfg_sz 68, total sz 22564
[   83.658494] rtk_btusb: Realtek Bluetooth USB driver ver 3.1.6d45ddf.20220519-142432
[   83.658651] usbcore: registered new interface driver rtk_btusb
[   83.667124] usb 2-1: This Realtek USB WiFi dongle (0x0bda:0xb720) is untested!
[   83.667137] usb 2-1: Please report results to Jes.Sorensen@gmail.com
[   83.890140] usb 2-1: Vendor: Realtek
[   83.890153] usb 2-1: Product: 802.11n WLAN Adapter
[   83.890159] usb 2-1: rtl8723bu_parse_efuse: dumping efuse (0x200 bytes):
......
[   83.890412] usb 2-1: RTL8723BU rev E (SMIC) 1T1R, TX queues 3, WiFi=1, BT=1, GPS=0, HI PA=0
[   83.890417] usb 2-1: RTL8723BU MAC: 00:13:ef:f4:58:ae
[   83.890421] usb 2-1: rtl8xxxu: Loading firmware rtlwifi/rtl8723bu_nic.bin
[   83.895289] usb 2-1: Firmware revision 35.0 (signature 0x5301)
[   84.050893] Bluetooth: hci0: RTL: fw version 0x0e2f9f73
[   84.266905] Bluetooth: RFCOMM TTY layer initialized
[   84.266949] Bluetooth: RFCOMM socket layer initialized
[   84.266999] Bluetooth: RFCOMM ver 1.11
[   84.884270] usbcore: registered new interface driver rtl8xxxu
[   84.912046] rtl8xxxu 2-1:1.2 wlx0013eff458ae: renamed from wlan0
```

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

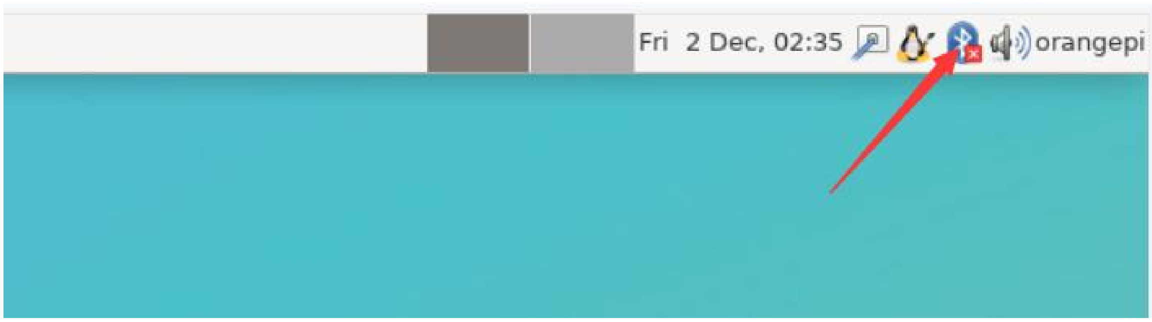
```
orangepi@orangepi:~$ sudo ifconfig wlx0013eff458ae
wlx0013eff458ae: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:13:ef:f4:58:ae  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0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

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

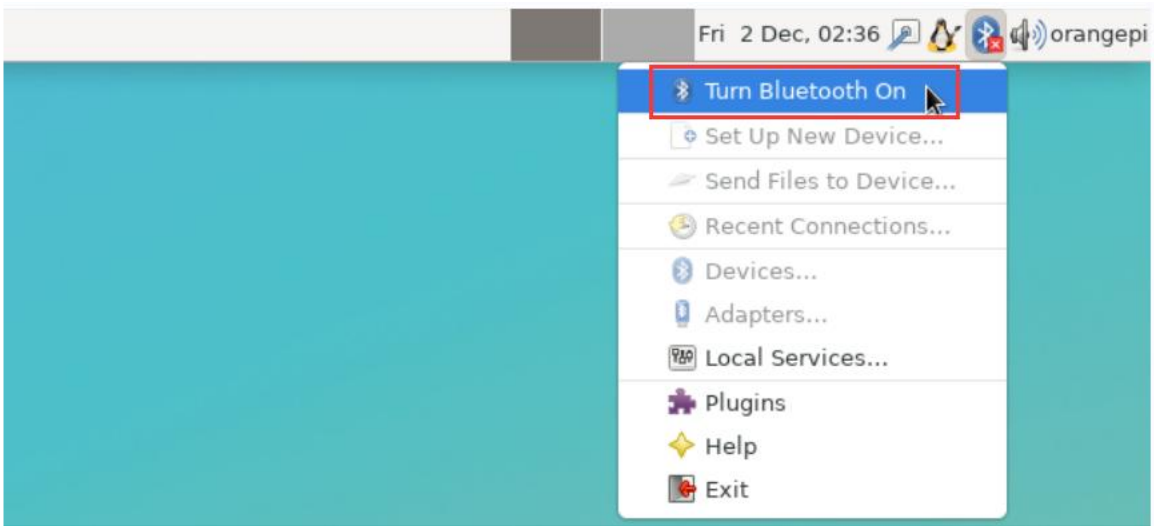
```
orangepi@orangepi:~$ sudo apt update && sudo apt install bluez
orangepi@orangepi:~$ hciconfig
```

```
hci0:    Type: Primary   Bus: USB
         BD Address: 00:13:EF:F4:58:AE   ACL MTU: 820:8   SCO MTU: 255:16
         DOWN
         RX bytes:1252 acl:0 sco:0 events:125 errors:0
         TX bytes:23307 acl:0 sco:0 commands:125 errors:0
```

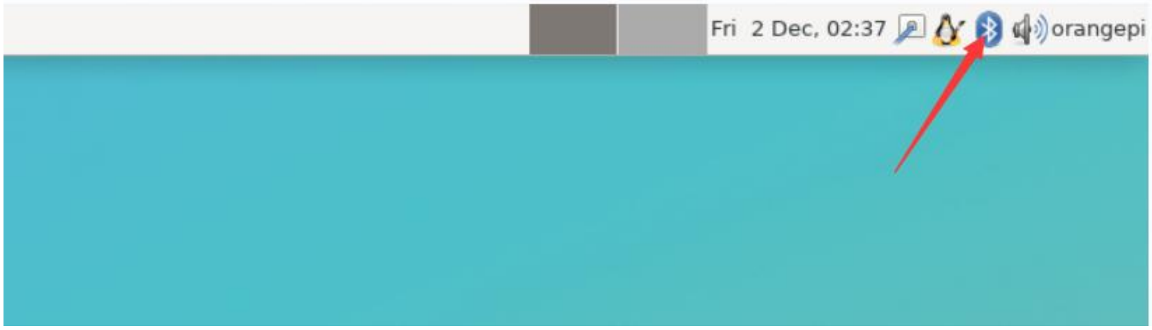
6) You can also see the Bluetooth icon on the desktop. Bluetooth is not turned on at this time, so a red **x** will be displayed



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



8) The display after turning on Bluetooth is as follows



9) For the Bluetooth test method, please refer to the section on **Bluetooth usage** and will not be repeated here.

3. 11. 3. 2. RTL8811 test

1) First insert the RTL8811 wireless network card module into the USB interface of the development board

2) Then the Linux system will automatically load the kernel module related to RTL8811 WIFI. Through the lsmod command, you can see that the following kernel module has been automatically loaded.

```
orangepi@orangepi:~$ lsmod
Module                  Size  Used by
8821cu               1839104  0
```

3) You can see the loading information of the RTL8811 module through the dmesg command

```
orangepi@orangepi:~$ dmesg
[   118.618194] usb 2-1: new high-speed USB device number 2 using ehci-platform
[   118.767152] usb 2-1: New USB device found, idVendor=0bda, idProduct=c811, bcdDevice= 2.00
[   118.767181] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   118.767199] usb 2-1: Product: 802.11ac NIC
[   118.767219] usb 2-1: Manufacturer: Realtek
[   118.767235] usb 2-1: SerialNumber: 123456
[   119.500530] usbcore: registered new interface driver rtl8821cu
[   119.525498] rtl8821cu 2-1:1.0 wlx1cbfced9d260: renamed from wlan0
```

4) Then, you can see the WIFI device node through the **sudo ifconfig** command. For the WIFI connection and test method, please refer to the **WIFI connection test** section, which will not be repeated 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. 11. 4. USB camera test

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

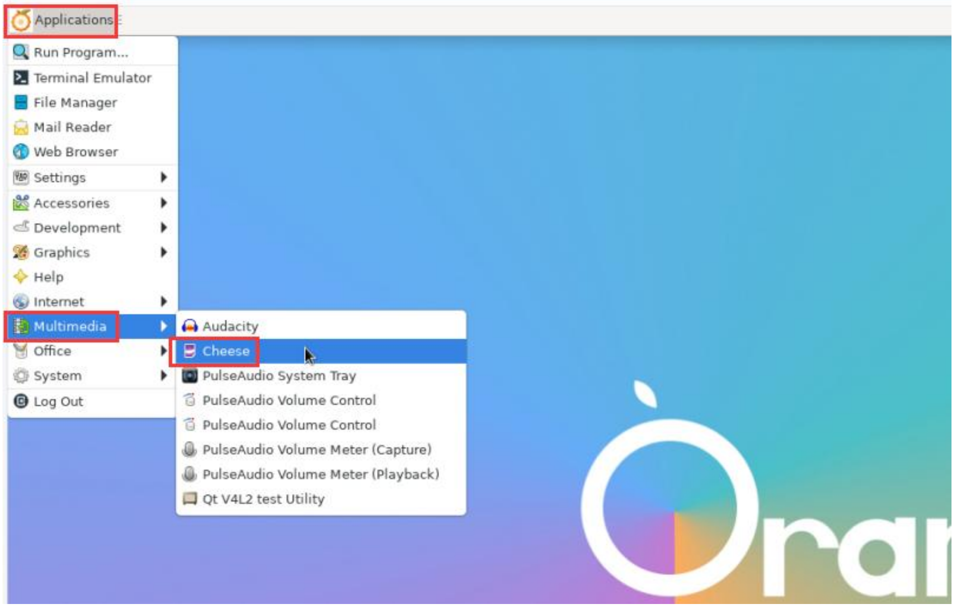
2) Through the v4l2-ctl command, you can see that the device node information of the USB camera is /dev/video0

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

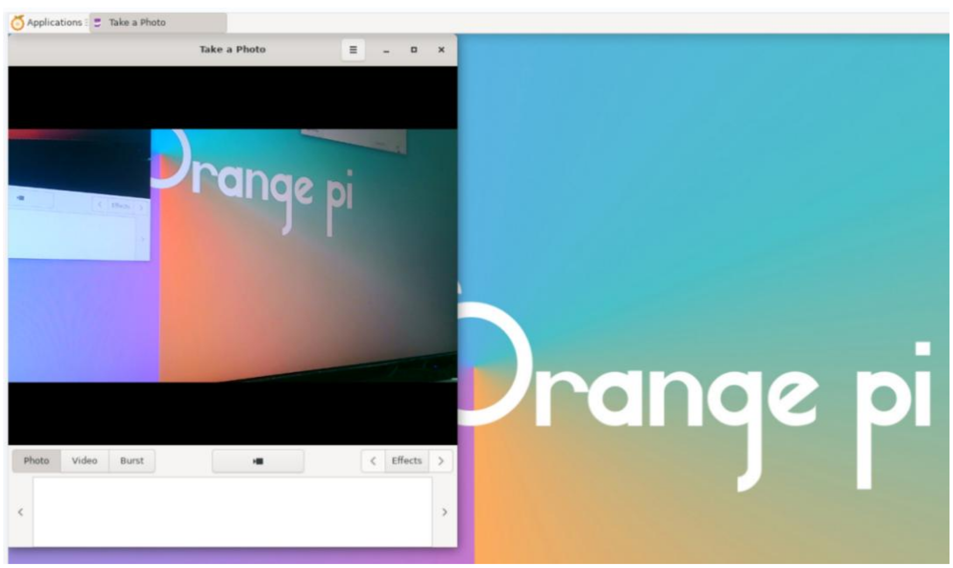
Note that l in v4l2 is a lowercase letter l, not the number 1

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

3) In the desktop system, you can use Cheese to directly open the USB camera. The method of opening Cheese is as shown in the figure below:



The interface after Cheese turns on the USB camera is as shown below:



4) How to use fswebcam to test USB camera

a. Install fswebcam

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

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

a) -d option is used to specify the device node of the USB camera
b) --no-banner Used to remove watermarks from photos
c) -r option to specify the resolution of the photo
d) -S Option to set the number of previous frames to skip
e) ./image.jpg Used to set the name and path of the generated photo

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

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

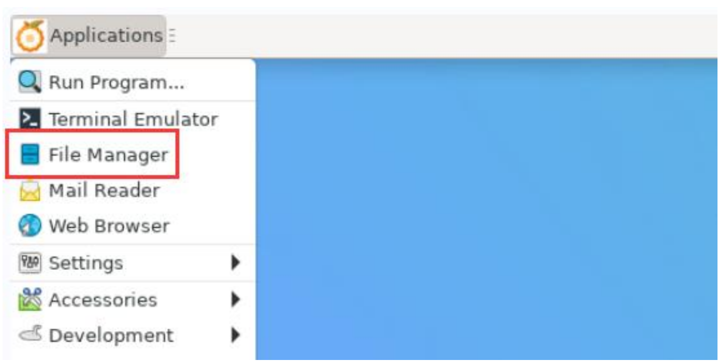
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test  （Modify the IP address and path according to the actual situation）
```

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

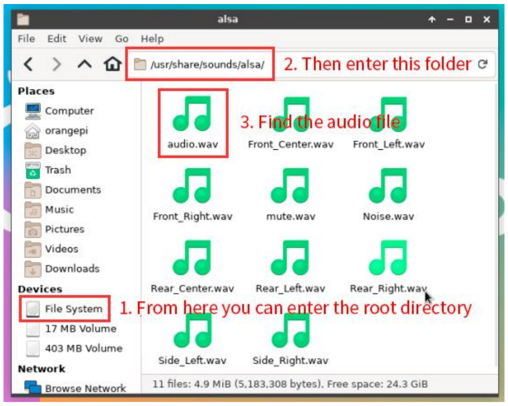
3. 12. Audio test

3. 12. 1. Testing audio methods on desktop systems

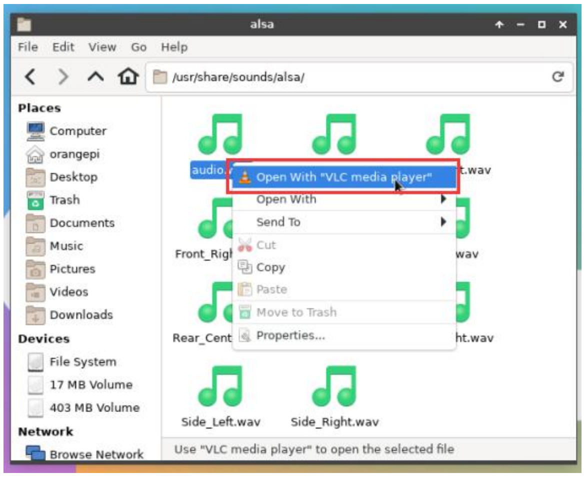
1) First open the file manager



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)

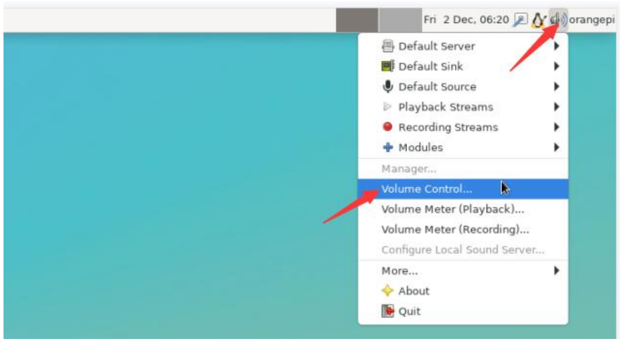


3) Then select the audio.wav file, right-click and select open with vlc to start playing.

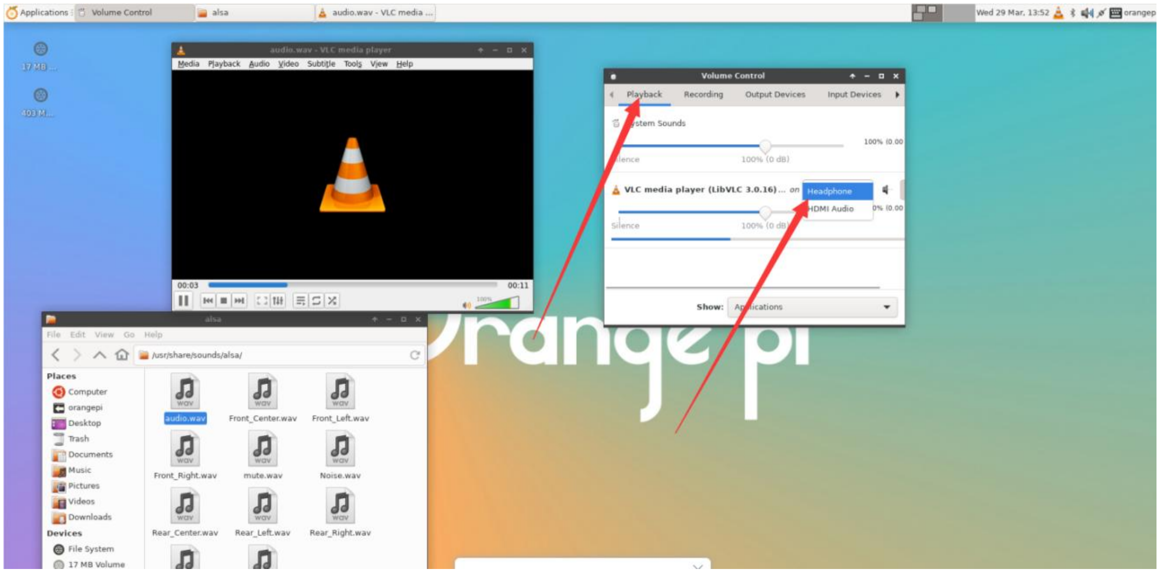


4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First open the volume control interface



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



3. 12. 2. How to use commands to play audio

3. 12. 2. 1. Headphone interface audio playback test

1) First insert the headphones into the headphone jack of the development board

2) Then you can use the **aplay -l** command to check the sound card devices supported by the Linux system. From the output below, we can see that **card 0** is the sound card device of rk809, which is the sound card device of the headset.

```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchiprk809 [rockchip-rk809], device 0: dailink-multicodecs rk817-hifi-0 [dailink-multicodecs rk817-hifi-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
card 1: rockchiphdmi [rockchip,hdmi], device 0: fe400000.i2s-i2s-hifi i2s-hifi-0 [fe400000.i2s-i2s-hifi i2s-hifi-0]
  Subdevices: 0/1
```

```
  Subdevice #0: subdevice #0
  Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio file that comes with the system. If the headset can hear the sound, it means that the hardware can be used normally.

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

3. 12. 2. 2. HDMI audio playback test

1) First use a Micro 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 serial number of the HDMI sound card. From the output below, you can know that the HDMI sound card is **card 1**

```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchiprk809 [rockchip-rk809], device 0: dailink-multicodecs rk817-hifi-0 [dailink-multicodecs rk817-hifi-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
card 1: rockchiphdmi [rockchip,hdmi], device 0: fe400000.i2s-i2s-hifi i2s-hifi-0 [fe400000.i2s-i2s-hifi i2s-hifi-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
  Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio file that comes with the system. If the sound can be heard on the HDMI display or TV, it means that the hardware can be used normally.

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

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

1) The Orange Pi CM4 development board does not have an onboard MIC, and audio can only be recorded through headphones with a MIC function. After inserting the headset with MIC function into the development board, run the following command to

record an audio period through the headset.

```
orangepi@orangepi:~$ amixer -c 0 cset name='Capture MIC Path' 'Main Mic'
orangepi@orangepi:~$ arecord -D hw:0,0 -d 5 -f cd -t wav /tmp/test.wav
```

3. 13. Temperature sensor

1) The command to view the system temperature sensor is:

```
orangepi@orangepi:~$ sensors
soc_thermal-virtual-0
Adapter: Virtual device
temp1:          +41.9°C  (crit = +115.0°C)

gpu_thermal-virtual-0
Adapter: Virtual device
temp1:          +43.8°C
```

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

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

3. 14. 40 Pin interface pin description

1) Please refer to the picture below for the order of the 40-pin interface pins of the Orange Pi CM4 development board

2) The functions of the 40-pin interface pins of the Orange Pi CM4 development board are as shown in the table below

a. The following is the complete pin diagram of 40pin

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
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	14		GND		
	UART7_TX_M2	GPIO4_A2	130	15	16	131	GPIO4_A3	UART7_RX_M2	
		3.3V		17	18	129	GPIO4_A1		
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19	20		GND		
	SPI3_MISO_M0	GPIO4_B0	136	21	22	132	GPIO4_A4	UART9_TX_M2	
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23	24	134	GPIO4_A6	SPI3_CS0_M0	
		GND		25	26	135	GPIO4_A7	SPI3_CS1_M0	
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27	28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
	UART9_RX_M2	GPIO4_A5	133	29	30		GND		
		GPIO3_D4	124	31	32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
		GPIO3_D7	127	33	34		GND		
		GPIO3_D0	120	35	36	125	GPIO3_D5		
		GPIO3_D3	123	37	38	122	GPIO3_D2		
		GND		39	40	121	GPIO3_D1		

b. The table below is the picture of the left half of the complete table above, so you can see it clearly

复用功能	复用功能	GPIO	GPIO序号	引脚序号
		3.3V		1
	I2C2_SDA_M1	GPIO4_B4	140	3
	I2C2_SCL_M1	GPIO4_B5	141	5
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7
		GND		9
		GPIO3_C6	118	11
		GPIO4_A0	128	13
	UART7_TX_M2	GPIO4_A2	130	15
		3.3V		17
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19
	SPI3_MISO_M0	GPIO4_B0	136	21
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23
		GND		25
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27
	UART9_RX_M2	GPIO4_A5	133	29
		GPIO3_D4	124	31
		GPIO3_D7	127	33
		GPIO3_D0	120	35
		GPIO3_D3	123	37
		GND		39

c. The table below is the picture of the right half of the complete table above, so you can see it clearly

引脚序号	GPIO序号	GPIO	复用功能	复用功能
2		5V		
4		5V		
6		GND		
8	25	GPIO0_D1	UART2_TX_M0	
10	24	GPIO0_D0	UART2_RX_M0	
12	119	GPIO3_C7		
14		GND		
16	131	GPIO4_A3	UART7_RX_M2	
18	129	GPIO4_A1		
20		GND		
22	132	GPIO4_A4	UART9_TX_M2	
24	134	GPIO4_A6	SPI3_CS0_M0	
26	135	GPIO4_A7	SPI3_CS1_M0	
28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
30		GND		
32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
34		GND		
36	125	GPIO3_D5		
38	122	GPIO3_D2		
40	121	GPIO3_D1		

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

3. 15. How to install wiringOP

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

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

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

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

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |     IN | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
 |  118 |   5 | GPIO3_C6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO3_C7 | 6   | 119  |
 |  128 |   7 | GPIO4_A0 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |  130 |   8 |    TXD.7 |     IN | 0 | 15 || 16 | 0 | IN     | RXD.7    | 9   | 131  |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO4_A1 | 10  | 129  |
 |  138 |  11 | SPI3_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |  136 |  12 | SPI3_RXD |     IN | 0 | 21 || 22 | 0 | IN     | TXD.9    | 13  | 132  |
 |  139 |  14 | SPI3_CLK |     IN | 0 | 23 || 24 | 0 | IN     | SPI3_CS1 | 15  | 134  |
 |      |     |      GND |        |   | 25 || 26 | 0 | IN     | GPIO4_A7 | 16  | 135  |
 |   32 |  17 |    SDA.3 |   ALT1 | 1 | 27 || 28 | 1 | ALT1   | SCL.3    | 18  | 33   |
 |  133 |  19 |    RXD.9 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |  124 |  20 | GPIO3_D4 |     IN | 0 | 31 || 32 | 0 | IN     | PWM11    | 21  | 144  |
 |  127 |  22 | GPIO3_D7 |     IN | 0 | 33 || 34 |   |        | GND      |     |      |
 |  120 |  23 | GPIO3_D0 |     IN | 0 | 35 || 36 | 0 | IN     | GPIO3_D5 | 24  | 125  |
 |  123 |  25 | GPIO3_D3 |     IN | 0 | 37 || 38 | 0 | IN     | GPIO3_D2 | 26  | 122  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO3_D1 | 27  | 121  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
```

WiringOP currently mainly adapts to the functions of setting GPIO port input and output, setting GPIO port output high and low levels, and setting pull-up and pull-down resistors. Functions such as hardware PWM cannot be used.

1) Download the code of wiringOP

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

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

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

Note that Orange Pi CM4 needs to download the code of wiringOP next branch. Please don't miss the -b next parameter.

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

2) Compile and install wiringOP

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

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

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

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |     IN | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
 |  118 |   5 | GPIO3_C6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO3_C7 | 6   | 119  |
 |  128 |   7 | GPIO4_A0 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |  130 |   8 |    TXD.7 |     IN | 0 | 15 || 16 | 0 | IN     | RXD.7    | 9   | 131  |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO4_A1 | 10  | 129  |
 |  138 |  11 | SPI3_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |  136 |  12 | SPI3_RXD |     IN | 0 | 21 || 22 | 0 | IN     | TXD.9    | 13  | 132  |
 |  139 |  14 | SPI3_CLK |     IN | 0 | 23 || 24 | 0 | IN     | SPI3_CS1 | 15  | 134  |
 |      |     |      GND |        |   | 25 || 26 | 0 | IN     | GPIO4_A7 | 16  | 135  |
 |   32 |  17 |    SDA.3 |   ALT1 | 1 | 27 || 28 | 1 | ALT1   | SCL.3    | 18  | 33   |
 |  133 |  19 |    RXD.9 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |  124 |  20 | GPIO3_D4 |     IN | 0 | 31 || 32 | 0 | IN     | PWM11    | 21  | 144  |
 |  127 |  22 | GPIO3_D7 |     IN | 0 | 33 || 34 |   |        | GND      |     |      |
 |  120 |  23 | GPIO3_D0 |     IN | 0 | 35 || 36 | 0 | IN     | GPIO3_D5 | 24  | 125  |
 |  123 |  25 | GPIO3_D3 |     IN | 0 | 37 || 38 | 0 | IN     | GPIO3_D2 | 26  | 122  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO3_D1 | 27  | 121  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
```

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

Note, if you need to set overlays to open multiple configurations at the same time, please use spaces to separate them and write them on one line as follows.

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

overlays=spi3-m0-cs0-spidev i2c2-m1 i2c3-m0 uart7-m2 uart9-m2 pwm11-m1

3. 16. 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 40pin to switch between high and low levels continuously.

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

The way to run the blink_all_gpio program is as follows:

orangepi@orangepicm4:~$ **sudo blink_all_gpio** # Remember to add sudo permission
[sudo] password for orangepi: # A password is required here

1) There are a total of 28 GPIO ports in the 40pins of the development board that can be used. The following uses pin 7—the corresponding GPIO is GPIO4_C3——the corresponding wPi serial number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |     IN | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
```

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

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

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

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

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

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |    OUT | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
```

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

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

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

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |    OUT | 1 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
```

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

3. 16. 2. How to set the pull-up and pull-down resistance of 40pin GPIO port

Note that the 4 GPIO pins below the Orange Pi CM4 have an external 3.3V pull-up, so setting the pull-down is invalid, and the other pins can normally set the pull-up and pull-down resistor function

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |     IN | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
 |  118 |   5 | GPIO3_C6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO3_C7 | 6   | 119  |
 |  128 |   7 | GPIO4_A0 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |  130 |   8 |    TXD.7 |     IN | 0 | 15 || 16 | 0 | IN     | RXD.7    | 9   | 131  |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO4_A1 | 10  | 129  |
 |  138 |  11 | SPI3_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |  136 |  12 | SPI3_RXD |     IN | 0 | 21 || 22 | 0 | IN     | TXD.9    | 13  | 132  |
 |  139 |  14 | SPI3_CLK |     IN | 0 | 23 || 24 | 0 | IN     | SPI3_CS1 | 15  | 134  |
 |      |     |      GND |        |   | 25 || 26 | 0 | IN     | GPIO4_A7 | 16  | 135  |
 |   32 |  17 |    SDA.3 |   ALT1 | 1 | 27 || 28 | 1 | ALT1   | SCL.3    | 18  | 33   |
 |  133 |  19 |    RXD.9 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |  124 |  20 | GPIO3_D4 |     IN | 0 | 31 || 32 | 0 | IN     | PWM11    | 21  | 144  |
 |  127 |  22 | GPIO3_D7 |     IN | 0 | 33 || 34 |   |        | GND      |     |      |
 |  120 |  23 | GPIO3_D0 |     IN | 0 | 35 || 36 | 0 | IN     | GPIO3_D5 | 24  | 125  |
 |  123 |  25 | GPIO3_D3 |     IN | 0 | 37 || 38 | 0 | IN     | GPIO3_D2 | 26  | 122  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO3_D1 | 27  | 121  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
```

1) The following takes pin 11—the corresponding GPIO is GPIO3_C6—the corresponding wPi serial number is 5—as an example to demonstrate how to set the

pull-up and pull-down resistance of the GPIO port

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |     IN | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
 |  118 |   5 | GPIO3_C6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO3_C7 | 6   | 119  |
 |  128 |   7 | GPIO4_A0 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |  130 |   8 |    TXD.7 |     IN | 0 | 15 || 16 | 0 | IN     | RXD.7    | 9   | 131  |
```

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

root@orangepi:~/wiringOP# **gpio mode 5 in**

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

root@orangepi:~/wiringOP# **gpio mode 5 up**

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

root@orangepi:~/wiringOP# **gpio read 5**
1

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

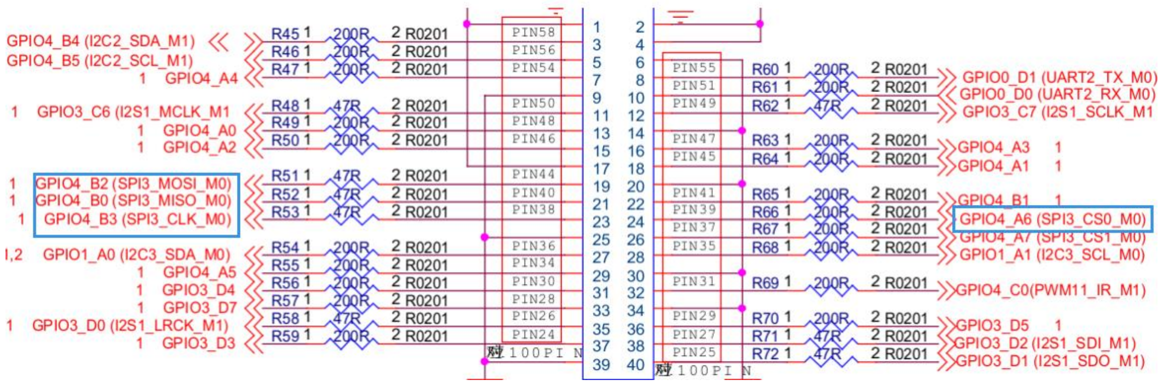
root@orangepi:~/wiringOP# **gpio mode 5 down**

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

root@orangepi:~/wiringOP# **gpio read 5**
0

3.16.3. 40pin SPI test

1) According to the schematic diagram of the 40pin interface, the spi available for Orange Pi CM4 is spi3



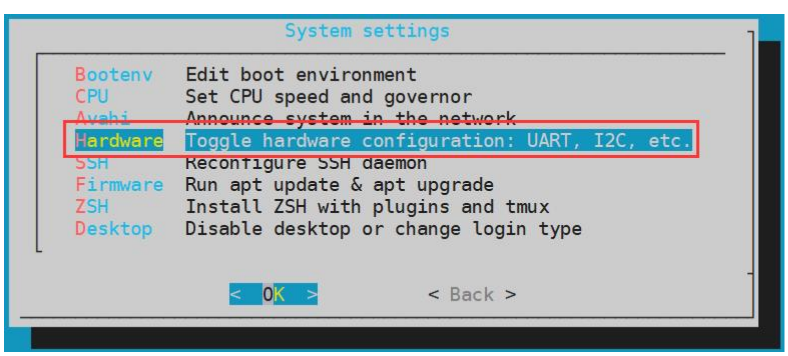
2) In the linux system, the SPI in the 40 pin is closed by default, and it needs to be opened manually to use it. The detailed steps are as follows:

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

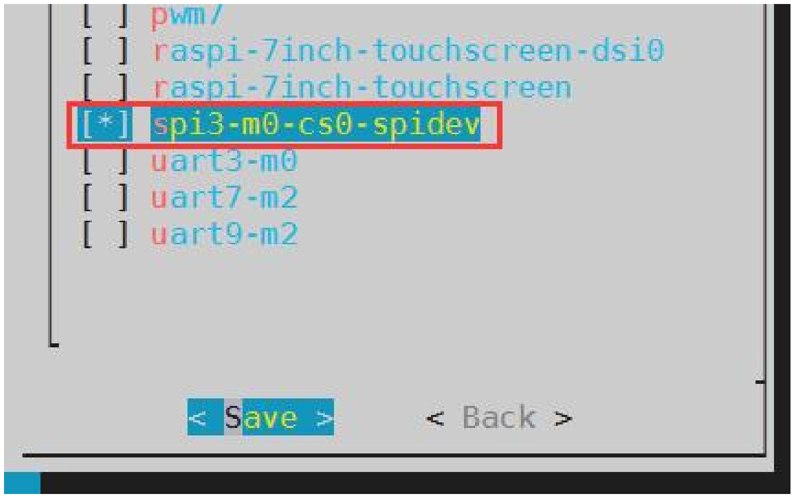
```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**

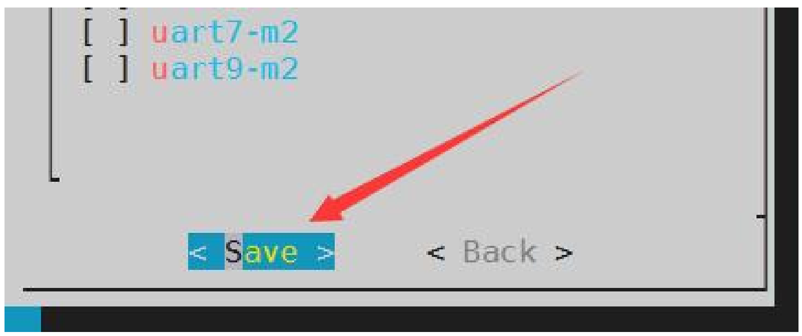
c. Then select **Hardware**



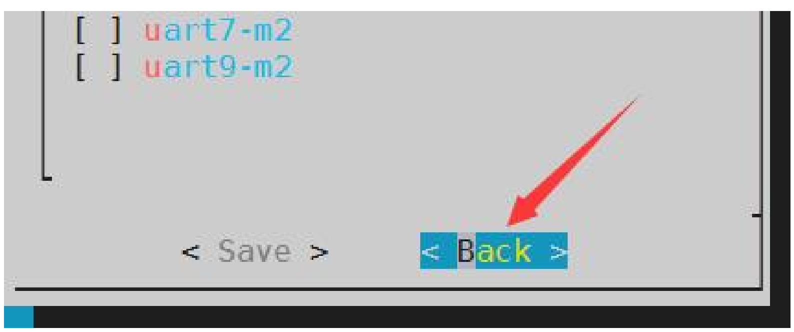
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the SPI configuration you want to open



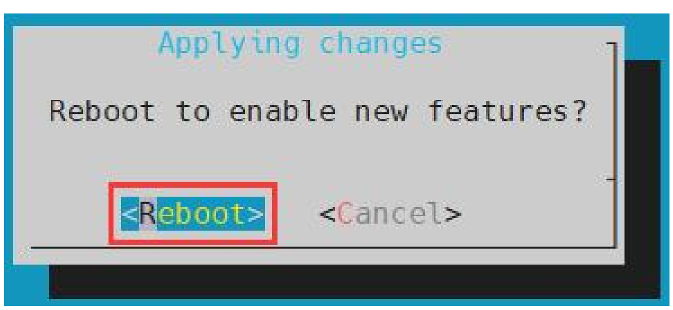
e. Then select **<Save>**Save



f. Then select**<Back>**



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



3) After restarting, enter the system to check whether there is a **spidev3.0** device node in the Linux system. If it exists, it means that SPI3 has been set up and can be used directly

```
orangepi@orangepi:~$ ls /dev/spidev3.0
/dev/spidev3.0
```

4) Do not short-circuit the mosi and miso pins of SPI3 first, and the output result of running spidev_test is as follows, you can see that the data of TX and RX are inconsistent

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev3.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D   | ......@....▒..................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF   | ................................
```

5) Then short-circuit the two pins of mosi (pin 19 in the 40pin interface) and miso (pin 21 in the 40pin interface) of SPI3 and run the output of spidev_test as follows, you can see the sending and receiving same data

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev3.0
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. 16. 4. 40pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi CM4 is i2c2, i2c3 and i2c4, a total of three sets of i2c buses

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
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	14		GND		
	UART7_TX_M2	GPIO4_A2	130	15	16	131	GPIO4_A3	UART7_RX_M2	
		3.3V		17	18	129	GPIO4_A1		
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19	20		GND		
	SPI3_MISO_M0	GPIO4_B0	136	21	22	132	GPIO4_A4	UART9_TX_M2	
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23	24	134	GPIO4_A6	SPI3_CS0_M0	
		GND		25	26	135	GPIO4_A7	SPI3_CS1_M0	
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27	28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
	UART9_RX_M2	GPIO4_A5	133	29	30		GND		
		GPIO3_D4	124	31	32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
		GPIO3_D7	127	33	34		GND		
		GPIO3_D0	120	35	36	125	GPIO3_D5		
		GPIO3_D3	123	37	38	122	GPIO3_D2		
		GND		39	40	121	GPIO3_D1		

As can be seen from the above table, i2c4_m0 and spi3_m0 are multiplexed pins, and both cannot be opened at the same time. i2c3_m0 and uart3_m0 are also multiplexed pins, and both cannot be opened at the same time

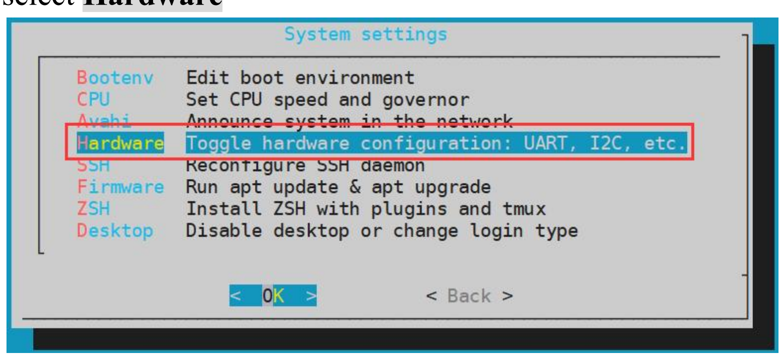
2) In the linux system, the I2C bus in the 40 pin is closed by default, and it needs to be opened manually to use it. The detailed steps are as follows:

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

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

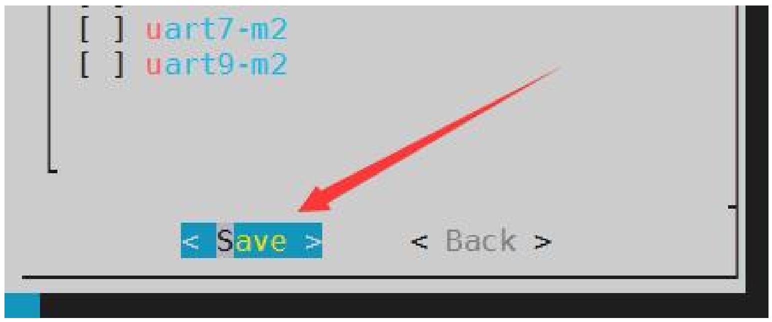
b. Then select **System**

c. Then select **Hardware**

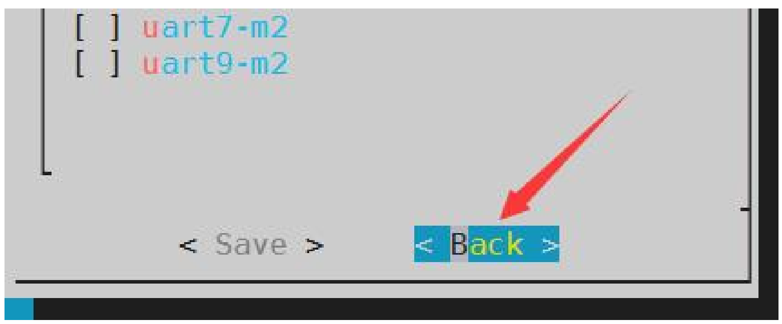


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the I2C configuration you want to open

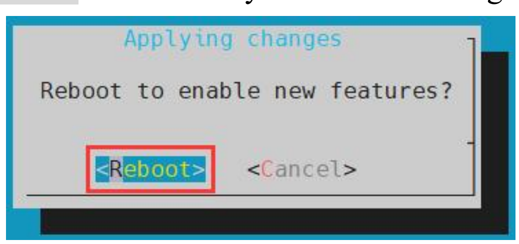
e. Then select **<Save>** Save



f. Then select **<Back>**



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



3) After starting the Linux system, first confirm that there is an i2c device node under /dev

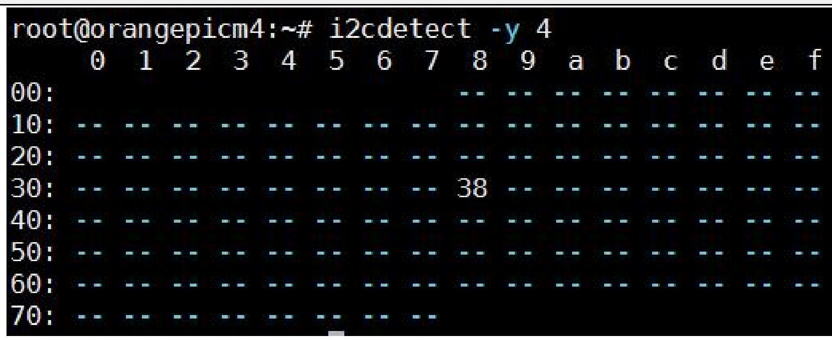
```
orangepi@orangepi:~# ls /dev/i2c-*
/dev/i2c-0   /dev/i2c-2   /dev/i2c-3   /dev/i2c-4   /dev/i2c-6
```

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

	i2c2-m1	i2c3-m0	i2c4-m0
Sda Pin	Corresponding to No. 3 pin	Corresponding to No. 27 pin	Corresponding to No. 19 pin
Sck Pin	Corresponding to No. 5 pin	Corresponding to No. 28 pin	Corresponding to No. 23 pin
Vcc Pin	Corresponding to No. 1 pin	Corresponding to No. 1 pin	Corresponding to No. 1 pin
Gnd Pin	Corresponding to No. 6 pin	Corresponding to No. 6 pin	Corresponding to No. 6 pin

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 i2cdetect -y 2        #i2c2 command
orangepi@orangepi:~$ sudo i2cdetect -y 3        #i2c3 command
orangepi@orangepi:~$ sudo i2cdetect -y 4        #i2c4 command
```



3. 16. 5. 40pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi CM4 are uart2, uart3, uart7 and uart9, a total of four groups of uart buses, of which uart2 is the system's debugging serial port by default.

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
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	14		GND		
	UART7_TX_M2	GPIO4_A2	130	15	16	131	GPIO4_A3	UART7_RX_M2	
		3.3V		17	18	129	GPIO4_A1		
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19	20		GND		
	SPI3_MISO_M0	GPIO4_B0	136	21	22	132	GPIO4_A4	UART9_TX_M2	
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23	24	134	GPIO4_A6	SPI3_CS0_M0	
		GND		25	26	135	GPIO4_A7	SPI3_CS1_M0	
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27	28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
	UART9_RX_M2	GPIO4_A5	133	29	30		GND		
		GPIO3_D4	124	31	32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
		GPIO3_D7	127	33	34		GND		
		GPIO3_D0	120	35	36	125	GPIO3_D5		
		GPIO3_D3	123	37	38	122	GPIO3_D2		
		GND		39	40	121	GPIO3_D1		

As can be seen from the above table, i2c3_m0 and uart3_m0 are pin-multiplexed, and both cannot be turned on at the same time.

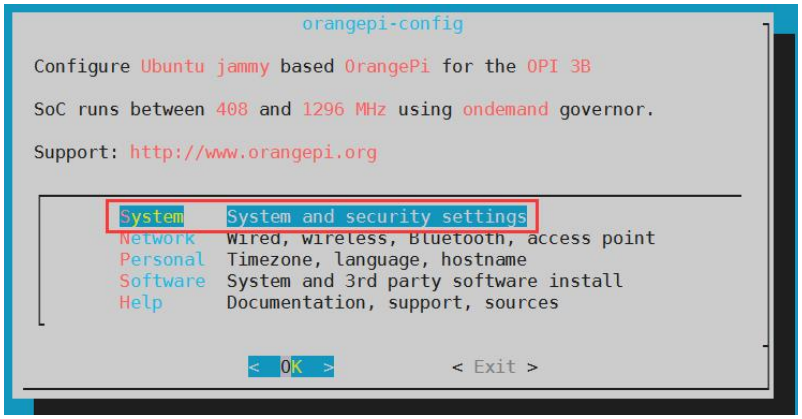
Please note that uart2_m0 is used as the debugging serial port of the system by default. If the configuration of uart2 is turned on, the debugging serial port function will not be available.

2) In the linux system, the UART in the 40 pins is closed by default, and it needs to be opened manually to use. The detailed steps are as follows:

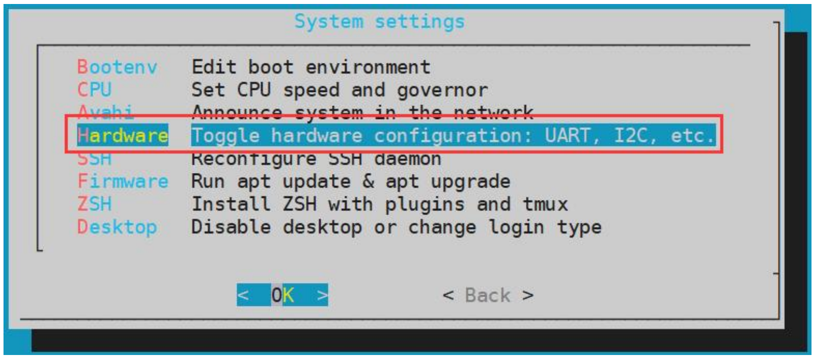
a. First run **orangepi-config**, common users remember to add **sudo** permission

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

b. Then select **System**

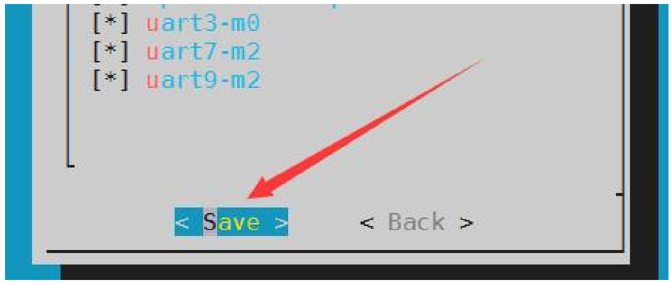


c. Then select **Hardware**

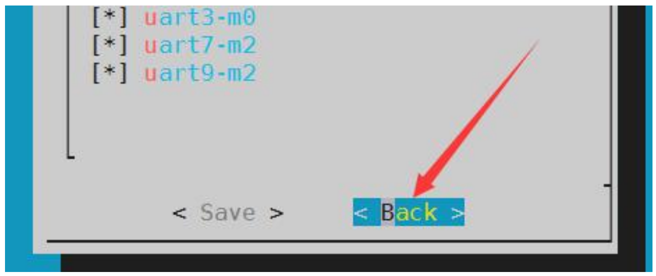


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the UART configuration you want to open

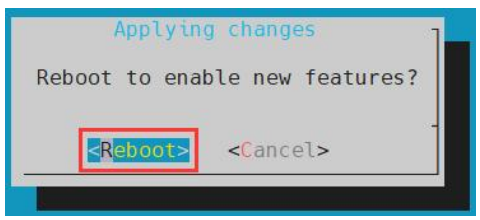
e. Then select **<Save>** Save



f. Then select **<Back>**



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



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/ttyS7   /dev/ttyS9
```

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

	uart3	uart7	uart9
tx pin	Corresponding to pin 28	Corresponding to pin 16	Corresponding to pin 29
rx pin	Corresponding to pin 27	Corresponding to pin 15	Corresponding to pin 22

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

a. Test UART3

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS3
[sudo] password for orangepi:    # enter password here

Out:     0:    ->     0
Out:     1:    ->     1
Out:     2:    ->     2
Out:     3:    ->     3
Out:     4:    ->     4
Out:     5:    ->     5^C
```

b. Test UART7

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS7
[sudo] password for orangepi:    #enter password here

Out:     0:    ->     0
Out:     1:    ->     1
Out:     2:    ->     2
Out:     3:    ->     3
Out:     4:    ->     4
Out:     5:    ->     5^C
```

c. Test UART9

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS9
[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. 16. 6. PWM test method

1) As can be seen from the table below, Orange Pi CM4 has two channels of pwm available pwm11 and pwm15

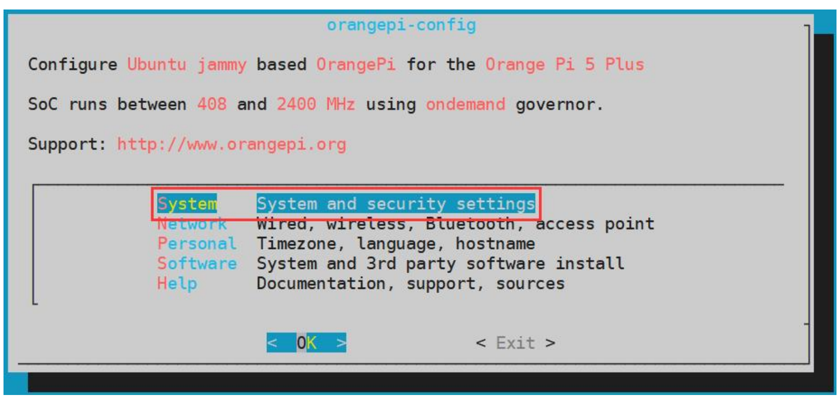
复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	14		GND		
	UART7_TX_M2	GPIO4_A2	130	15	16	131	GPIO4_A3	UART7_RX_M2	
		3.3V		17	18	129	GPIO4_A1		
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19	20		GND		
	SPI3_MISO_M0	GPIO4_B0	136	21	22	132	GPIO4_A4	UART9_TX_M2	
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23	24	134	GPIO4_A6	SPI3_CS0_M0	
		GND		25	26	135	GPIO4_A7	SPI3_CS1_M0	
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27	28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
	UART9_RX_M2	GPIO4_A5	133	29	30		GND		
		GPIO3_D4	124	31	32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
		GPIO3_D7	127	33	34		GND		
		GPIO3_D0	120	35	36	125	GPIO3_D5		
		GPIO3_D3	123	37	38	122	GPIO3_D2		
		GND		39	40	121	GPIO3_D1		

2) In the linux system, the PWM in the 40 pin is turned off by default, and it needs to be turned on manually before it can be used. The detailed steps are as follows:

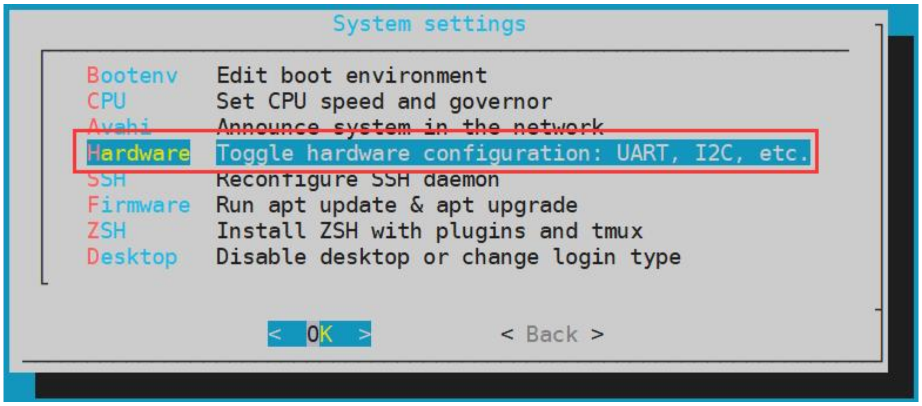
a. First run **orangepi-config**, common users remember to add **sudo** permission

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

b. Then select **System**

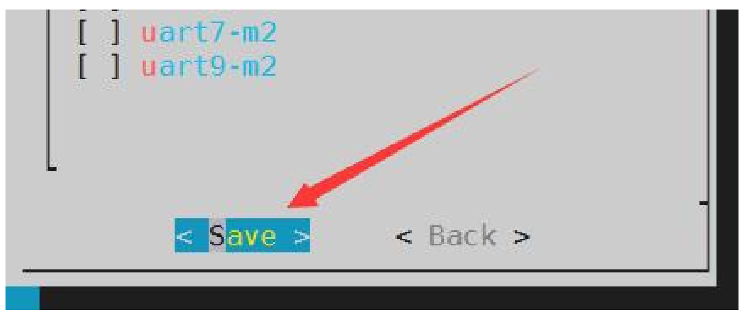


c. Then select **Hardware**

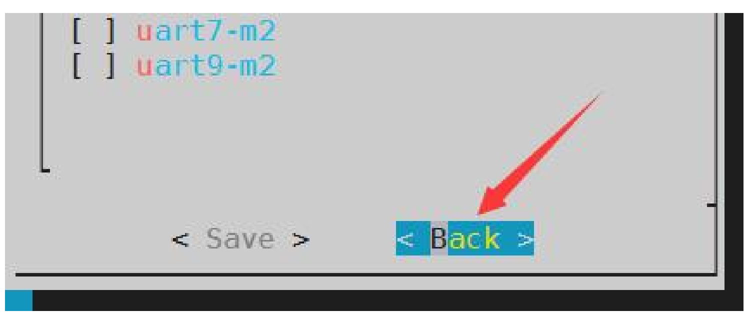


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the PWM configuration you want to open

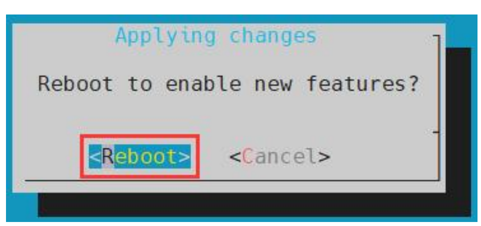
e. Then select **<Save>** Save



f. Then select **<Back>**



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



3) After opening a pwm, there will be an extra pwmchipX in /sys/class/pwm/ (X is a specific number), for example, after opening pwm11, check the pwmchipX under /sys/class/pwm/ one becomes two

```
orangepi@orangepi:~$ ls /sys/class/pwm/
pwmchip0    pwmchip1
```

4) Which of the above pwmchips corresponds to pwm11? Let's first check the output of the **ls /sys/class/pwm/ -l** command, as shown below:

```
orangepi@orangepicm4:~$ ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Jan  1  1970 pwmchip0 -> ../../devices/platform/fe6e0030.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Jan  1  1970 pwmchip1 -> ../../devices/platform/fe6f0030.pwm/pwm/pwmchip1
```

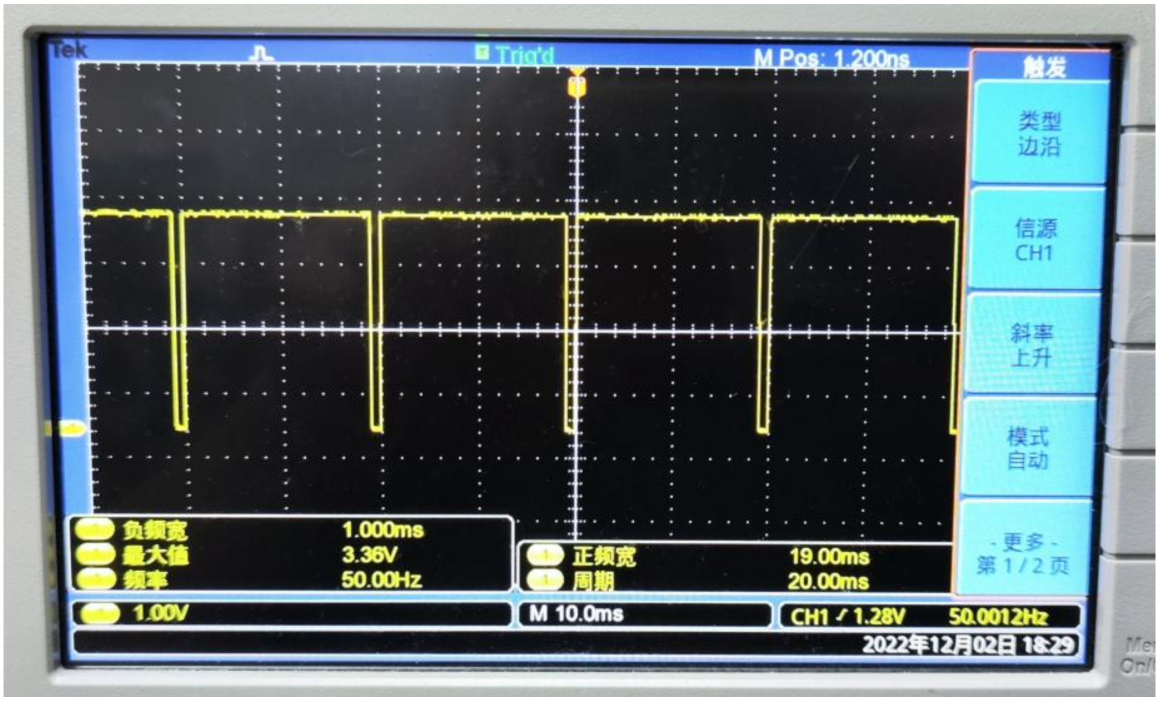
5) Then it can be known from the table below that the base address of the pwm11 register is fe6f0030, and then look at the output of the **ls /sys/class/pwm/ -l** command, you can see that pwmchip1 is linked to fe6f0030.pwm, so pwm11 corresponds to pwmchip as pwmchip1

引脚序号	GPIO序号	GPIO	复用功能	复用功能
2		5V		
4		5V		
6		GND		
8	25	GPIO0_D1	UART2_TX_M0	
10	24	GPIO0_D0	UART2_RX_M0	
12	119	GPIO3_C7		
14		GND		
16	131	GPIO4_A3	UART7_RX_M2	
18	129	GPIO4_A1		
20		GND		
22	132	GPIO4_A4	UART9_TX_M2	
24	134	GPIO4_A6	SPI3_CS0_M0	
26	135	GPIO4_A7	SPI3_CS1_M0	
28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
30		GND		
32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
34		GND		
36	125	GPIO3_D5		
38	122	GPIO3_D2		
40	121	GPIO3_D1		

6) Then use the following command to make pwm11 output a 50Hz square wave (please switch to the root user first, and then execute the following command)

```
root@orangepi:~# echo 0 > /sys/class/pwm/pwmchip1/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip1/pwm0/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip1/pwm0/duty_cycle
```

```
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip1/pwm0/enable
```



7) The test method of pwm11 demonstrated above is similar to other pwm test methods.

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

wiringOP-Python is the Python language version of wiringOP, which is used to operate the hardware resources of the development board, such as GPIO, I2C, SPI and UART, in the Python program.

In addition, please note that all the following commands are operated under the root user.

3. 17. 1. 3.17.1. How to install wiringOP-Python

1) First install the dependent packages

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python

Note that the following git clone--recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on

wiringOP. Please make sure that the download process does not report errors due to network problems.

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

```
root@orangepi:~# git clone --recursive https://github.com/orangepi-xunlong/wiringOP-Python -b next
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# git submodule update --init --remote
```

3) Then use the following command to compile wiringOP-Python and install it into the Linux system of the development board

```
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# python3 generate-bindings.py > bindings.i
root@orangepi:~/wiringOP-Python# sudo python3 setup.py install
```

4) Then enter the following command, if there is help information output, it means that wiringOP-Python is installed successfully, press the q key to exit the help information interface

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi;    help(wiringpi)"
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.
```

5) The steps to test whether wiringOP-Python is successfully installed under the python command line are as follows：

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

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi;
```

c. Finally, enter the following command to view the help information of wiringOP-Python, press the q key to exit the help information interface

```
>>> help(wiringpi)
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

    class GPIO(builtins.object)
     |  GPIO(pinmode=0)
     |

>>>
```

3. 17. 2. 40pin GPIO port test

wiringOP-Python is the same as wiringOP, you can also determine which GPIO pin to operate by specifying the wPi number, because there is no command to check the wPi number in wiringOP-Python, so you can only check the board wPi number and physical Correspondence between pins.

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |     IN | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
 |  118 |   5 | GPIO3_C6 |     IN | 0 | 11 || 12 | 0 | IN     | GPIO3_C7 | 6   | 119  |
 |  128 |   7 | GPIO4_A0 |     IN | 0 | 13 || 14 |   |        | GND      |     |      |
 |  130 |   8 |    TXD.7 |     IN | 0 | 15 || 16 | 0 | IN     | RXD.7    | 9   | 131  |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO4_A1 | 10  | 129  |
 |  138 |  11 | SPI3_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |  136 |  12 | SPI3_RXD |     IN | 0 | 21 || 22 | 0 | IN     | TXD.9    | 13  | 132  |
 |  139 |  14 | SPI3_CLK |     IN | 0 | 23 || 24 | 0 | IN     | SPI3_CS1 | 15  | 134  |
 |      |     |      GND |        |   | 25 || 26 | 0 | IN     | GPIO4_A7 | 16  | 135  |
 |   32 |  17 |    SDA.3 |   ALT1 | 1 | 27 || 28 | 1 | ALT1   | SCL.3    | 18  | 33   |
 |  133 |  19 |    RXD.9 |     IN | 0 | 29 || 30 |   |        | GND      |     |      |
 |  124 |  20 | GPIO3_D4 |     IN | 0 | 31 || 32 | 0 | IN     | PWM11    | 21  | 144  |
 |  127 |  22 | GPIO3_D7 |     IN | 0 | 33 || 34 |   |        | GND      |     |      |
 |  120 |  23 | GPIO3_D0 |     IN | 0 | 35 || 36 | 0 | IN     | GPIO3_D5 | 24  | 125  |
 |  123 |  25 | GPIO3_D3 |     IN | 0 | 37 || 38 | 0 | IN     | GPIO3_D2 | 26  | 122  |
 |      |     |      GND |        |   | 39 || 40 | 0 | IN     | GPIO3_D1 | 27  | 121  |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
```

1) The following takes pin No. 7—the corresponding GPIO is GPIO4_C3—the corresponding wPi number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepicm4:~# gpio readall
 +------+-----+----------+--------+---+  PI CM4  +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  140 |   0 |    SDA.2 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |  141 |   1 |    SCL.2 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |  147 |   2 |    PWM15 |     IN | 0 |  7 || 8  | 1 | ALT1   | RXD.2    | 3   | 25   |
 |      |     |      GND |        |   |  9 || 10 | 1 | ALT1   | TXD.2    | 4   | 24   |
```

2) The steps to test directly with the command are as follows

a. First set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.pinMode(2, GPIO.OUTPUT) ;  "
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.LOW)"
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.HIGH)"
```

3) The steps to test in the python3 command line are as follows:

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

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

c. Then set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
>>> wiringpi.wiringPiSetup()
0
>>> wiringpi.pinMode(2, GPIO.OUTPUT)
```

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

```
>>> wiringpi.digitalWrite(2, GPIO.LOW)
```

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

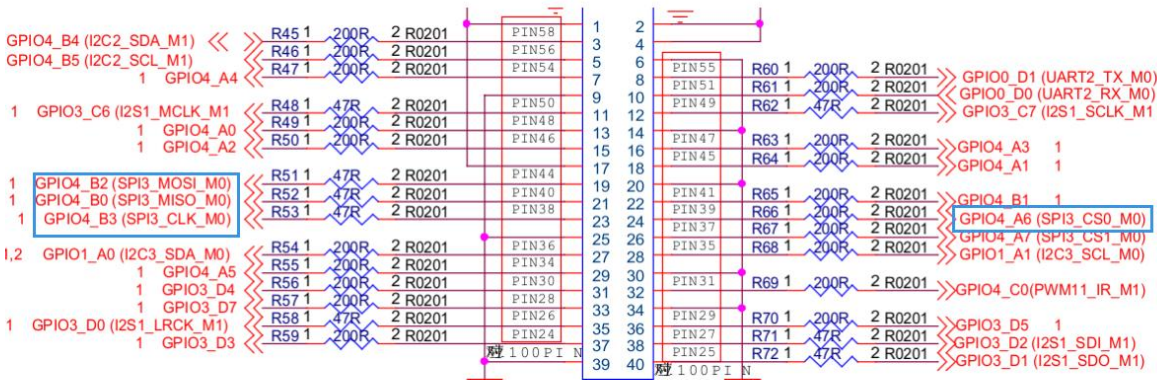
```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

4) The method of setting GPIO high and low levels in the python code by wiringOP-Python can refer to the **blink.py** test program in the examples below. The **blink.py** test program will set the voltage of all GPIO ports in the 40 pins of the development board to change continuously.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# ls blink.py
blink.py
root@orangepi:~/wiringOP-Python/examples# python3 blink.py
```

3. 17. 3. 40pin SPI test

1) According to the schematic diagram of the 40pin interface, the spi available for Orange Pi CM4 is spi3

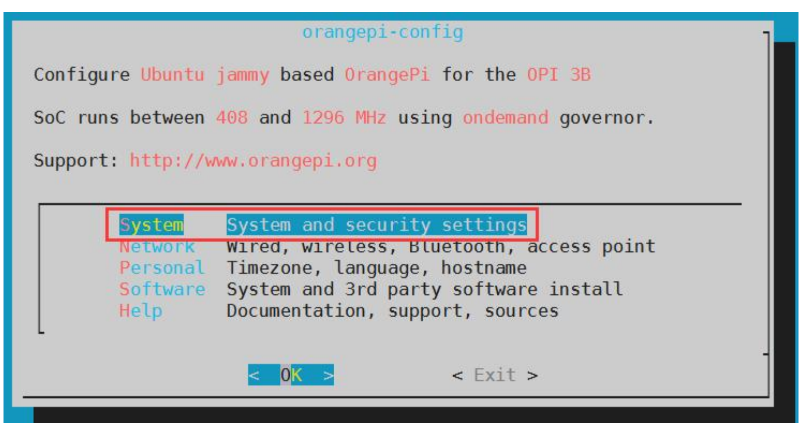


2) In the linux system, the SPI in the 40 pin is closed by default, and it needs to be opened manually to use it. The detailed steps are as follows:

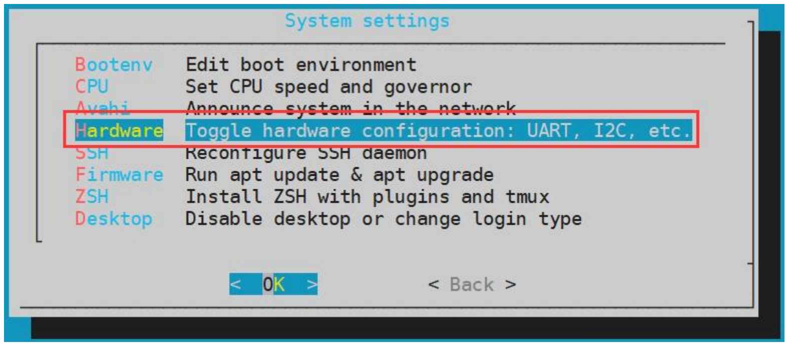
a. First run **orangepi-config**, common users remember to add **sudo** permission

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

b. Then select **System**

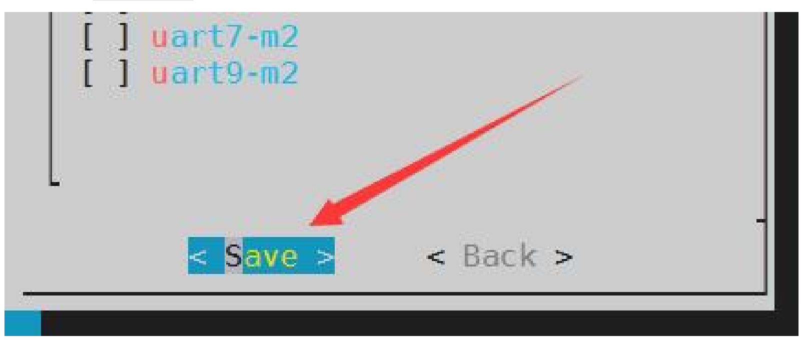


c. Then select **Hardware**

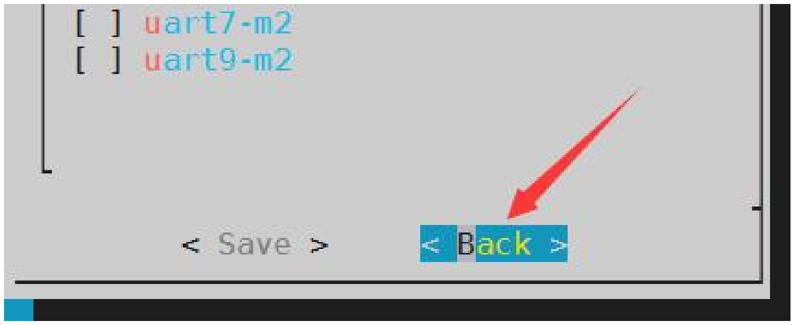


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the SPI configuration you want to open

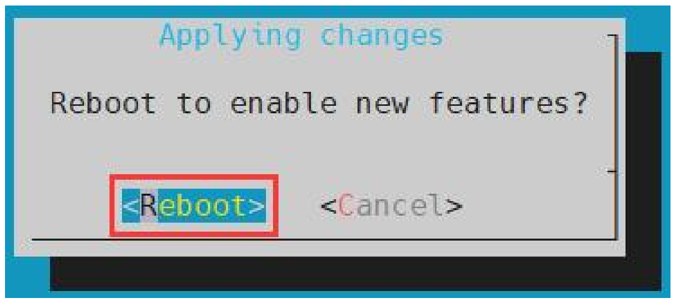
e. Then select **<Save>**Save



f. Then select **<Back>**



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



3) First check whether there is a **spidev3.0** device node in the Linux system. If it exists, it means that SPI3 has been set up and can be used directly

```
orangepi@orangepi:~$ ls /dev/spidev3.0
/dev/spidev3.0
```

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

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

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

5) Do not short-circuit the mosi and miso pins of SPI3, the output of running spidev_test.py is as follows, you can see that the data of TX and RX are inconsistent

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 3 --port 0
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev3.0
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    |................|
```

6) Then use the Dupont wire to short-circuit the two pins of txd (pin 19 in the 40pin interface) and rxd (pin 21 in the 40pin interface) of SPI3 and then run the output of spidev_test.py as follows, you can see The data sent and received are the same, indicating that the SPI3 loopback test is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 3 --port 0
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev3.0
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. 17. 4. 40pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi CM4 is i2c2, i2c3 and i2c4, a total of three sets of i2c buses

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
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	14		GND		
	UART7_TX_M2	GPIO4_A2	130	15	16	131	GPIO4_A3	UART7_RX_M2	
		3.3V		17	18	129	GPIO4_A1		
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19	20		GND		
	SPI3_MISO_M0	GPIO4_B0	136	21	22	132	GPIO4_A4	UART9_TX_M2	
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23	24	134	GPIO4_A6	SPI3_CS0_M0	
		GND		25	26	135	GPIO4_A7	SPI3_CS1_M0	
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27	28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
	UART9_RX_M2	GPIO4_A5	133	29	30		GND		
		GPIO3_D4	124	31	32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
		GPIO3_D7	127	33	34		GND		
		GPIO3_D0	120	35	36	125	GPIO3_D5		
		GPIO3_D3	123	37	38	122	GPIO3_D2		
		GND		39	40	121	GPIO3_D1		

As can be seen from the above table, i2c4_m0 and spi3_m0 are multiplexed pins, and both cannot be opened at the same time. i2c3_m0 and uart3_m0 are also multiplexed pins, and both cannot be opened at the same time

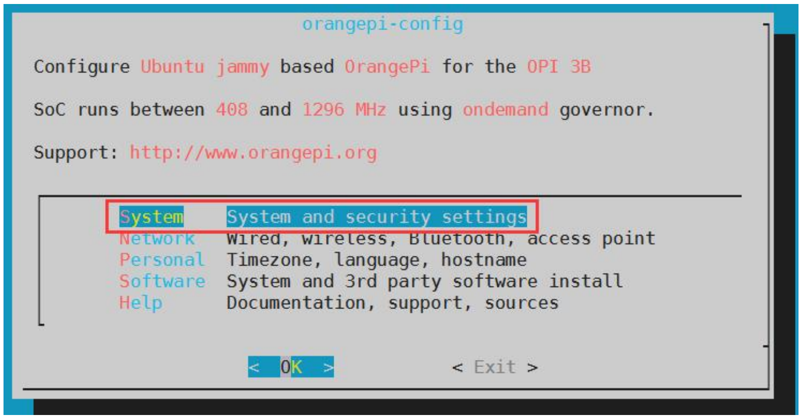
2) In the linux system, the I2C bus in the 40 pin is closed by default, and it needs to be

opened manually to use it. The detailed steps are as follows:

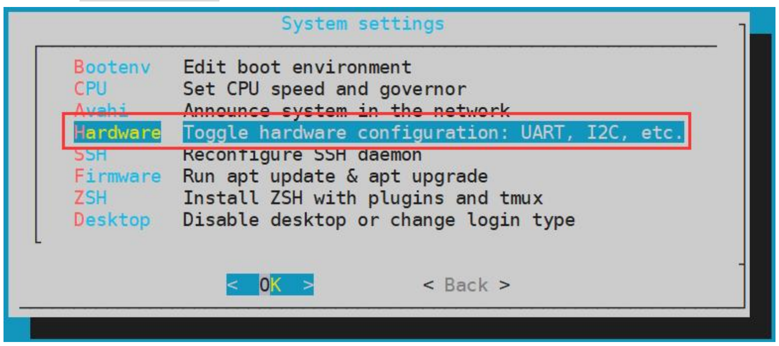
a. First run **orangepi-config**, common users remember to add **sudo** permission

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

b. Then select **System**

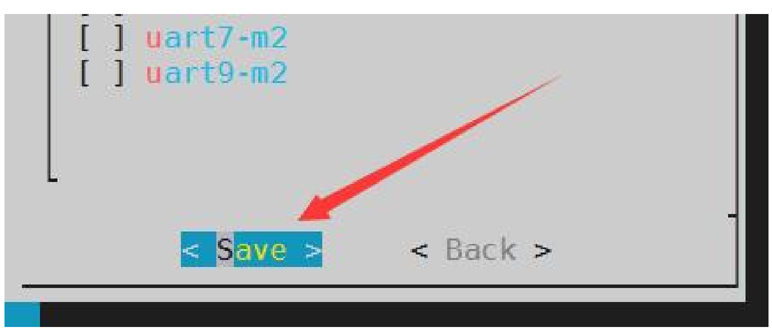


c. Then select **Hardware**

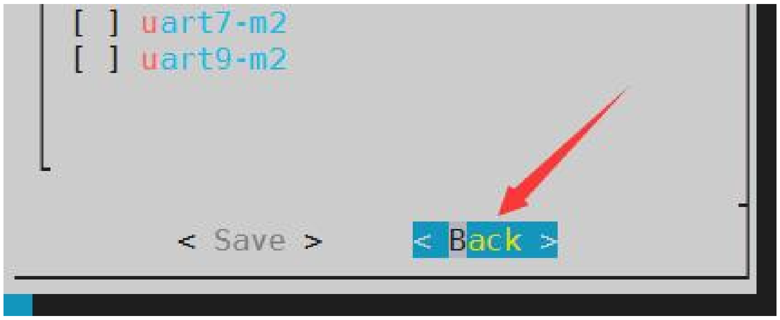


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the I2C configuration you want to open

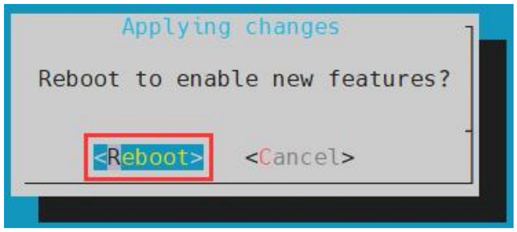
e. Then select **<Save>** Save



f. Then select **<Back>**



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

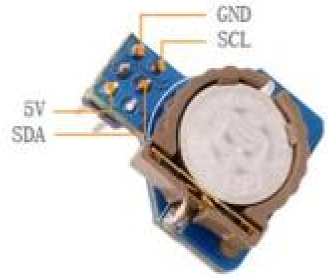


3) After starting the Linux system, first confirm that there is an i2c device node under /dev

```
orangepi@orangepi:~# ls /dev/i2c-*
/dev/i2c-0   /dev/i2c-2   /dev/i2c-3   /dev/i2c-4   /dev/i2c-6
```

4) Then connect an i2c device to the i2c pin of the 40pin connector, here we take the ds1307 RTC module as an example

	i2c2-m1	i2c3-m0	i2c4-m0
Sda pin	Corresponding to pin 3	Corresponding to pin 27	Corresponding to pin 19
Sck pin	Corresponding to pin 5	Corresponding to pin 28	Corresponding to pin 28
Vcc pin	Corresponding to pin 1	Corresponding to pin 1	Corresponding to pin 1
Gnd pin	Corresponding to pin 6	Corresponding to pin 6	Corresponding to pin 6



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 i2cdetect -y 2        #i2c2 command
orangepi@orangepi:~$ sudo i2cdetect -y 3        #i2c3 command
orangepi@orangepi:~$ sudo i2cdetect -y 4        #i2c4 command
```

```
root@orangepicm4:~# i2cdetect -y 4
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- 38 -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

6) 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-4"
Thu 2023-01-05    14:57:55
Thu 2023-01-05    14:57:56
Thu 2023-01-05    14:57:57
^C
exit
```

3. 17. 5. 40pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi CM4 are uart2, uart3, uart7 and uart9, a total of four groups of uart buses, of which uart2 is the system’s debugging serial port by default.

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
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	14		GND		
	UART7_TX_M2	GPIO4_A2	130	15	16	131	GPIO4_A3	UART7_RX_M2	
		3.3V		17	18	129	GPIO4_A1		
I2C4_SDA_M0	SPI3_MOSI_M0	GPIO4_B2	138	19	20		GND		
	SPI3_MISO_M0	GPIO4_B0	136	21	22	132	GPIO4_A4	UART9_TX_M2	
I2C4_SCL_M0	SPI3_CLK_M0	GPIO4_B3	139	23	24	134	GPIO4_A6	SPI3_CS0_M0	
		GND		25	26	135	GPIO4_A7	SPI3_CS1_M0	
UART3_RX_M0	I2C3_SDA_M0	GPIO1_A0	32	27	28	33	GPIO1_A1	I2C3_SCL_M0	UART3_TX_M0
	UART9_RX_M2	GPIO4_A5	133	29	30		GND		
		GPIO3_D4	124	31	32	144	GPIO4_C0	PWM11_IR_M1(fe6f0030)	
		GPIO3_D7	127	33	34		GND		
		GPIO3_D0	120	35	36	125	GPIO3_D5		
		GPIO3_D3	123	37	38	122	GPIO3_D2		
		GND		39	40	121	GPIO3_D1		

As can be seen from the above table, i2c3_m0 and uart3_m0 are pin-multiplexed, and both cannot be turned on at the same time.

Please note that uart2_m0 is used as the debugging serial port of the system by default. If the configuration of uart2 is turned on, the debugging serial port function will not be available.

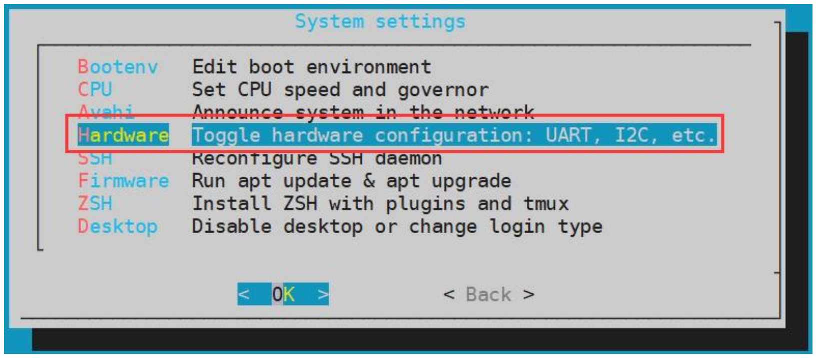
2) In the linux system, the UART in the 40 pins is closed by default, and it needs to be opened manually to use. The detailed steps are as follows:

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

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

b. Then select **System**

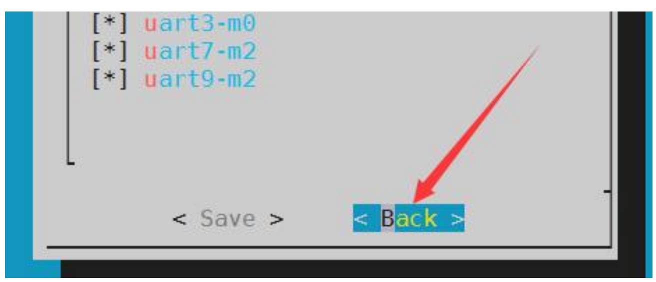
c. Then select **Hardware**



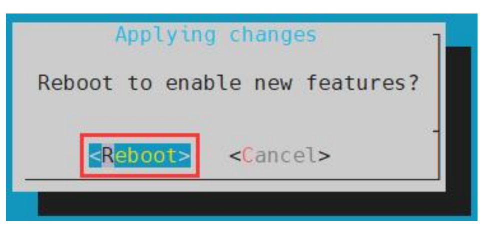
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the UART configuration you want to open

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

f. Then select **<Back>**



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



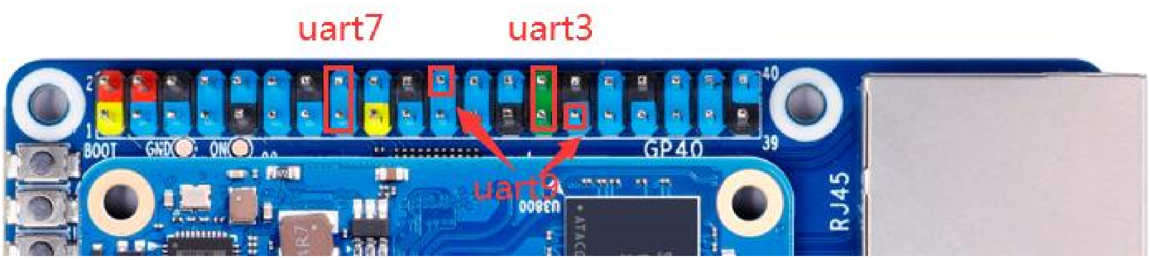
3) After entering the Linux system, first confirm whether there is a device node

corresponding to uart under /dev

```
orangepi@orangepi:~# ls /dev/ttyS*
/dev/ttyS1    /dev/ttyS3    /dev/ttyS7    /dev/ttyS9
```

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

	uart3	uart7	uart9
tx pin	Corresponding to pin 28	Corresponding to pin 16	Corresponding to pin 29
rx pin	Corresponding to pin 27	Corresponding to pin 15	Corresponding to pin 22



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

a. Test UART3

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS3"

Out:    0: ->    0
Out:    1: ->    1
Out:    2: ->    2
Out:    3: ->    3
Out:    4:^C
exit
```

b. Test UART7

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS7"

Out:    0: ->    0
```

```
Out:    1: ->   1
Out:    2: ->   2
Out:    3: ->   3
Out:    4:^C
exit
```

c. Test UART9

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS9"

Out:    0: ->   0
Out:    1: ->   1
Out:    2: ->   2
Out:    3: ->   3
Out:    4:^C
exit
```

3. 18. Hardware watchdog test

The watchdog_test program is pre-installed in the Linux system released by Orange Pi, which can be tested directly.

The method to run the watchdog_test program is as follows:

a. The second parameter 10 indicates the counting time of the watchdog. If the dog is not fed within this time, the system will restart

b. We can feed the dog by pressing any key on the keyboard (except ESC). After feeding the dog, the program will print a line of keep alive to indicate that the dog is fed successfully

```
orangepi@orangepi:~$ sudo watchdog_test 10
open success
options is 33152,identity is sunxi-wdt
put_usr return,if 0,success:0
The old reset time is: 16
return ENOTTY,if -1,success:0
return ENOTTY,if -1,success:0
```

```
put_user return,if 0,success:0
put_usr return,if 0,success:0
keep alive
keep alive
keep alive
```

3. 19. Check the serial number of RK3566 chip

The command to view the serial number of the RK3566 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          : 8fa18eaf489041f0
```

3. 20. How to install Docker

1) The linux image provided by Orange Pi has pre-installed Docker, but the Docker service is not enabled by default

2) Use the **enable_docker.sh**script to enable the docker service, and then you can start using the docker command, and the docker service will be automatically started when the system is started next time

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

3) Then you can use the following command to test docker, if you can run hello-world, it means that docker can be used normally

```
orangepi@orangepi:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
256ab8fe8778: Pull complete
Digest:
sha256:7f0a9f93b4aa3022c3a4c147a449ef11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

```
.....
```

3. 21. How to download and install arm64 version balenaEtcher

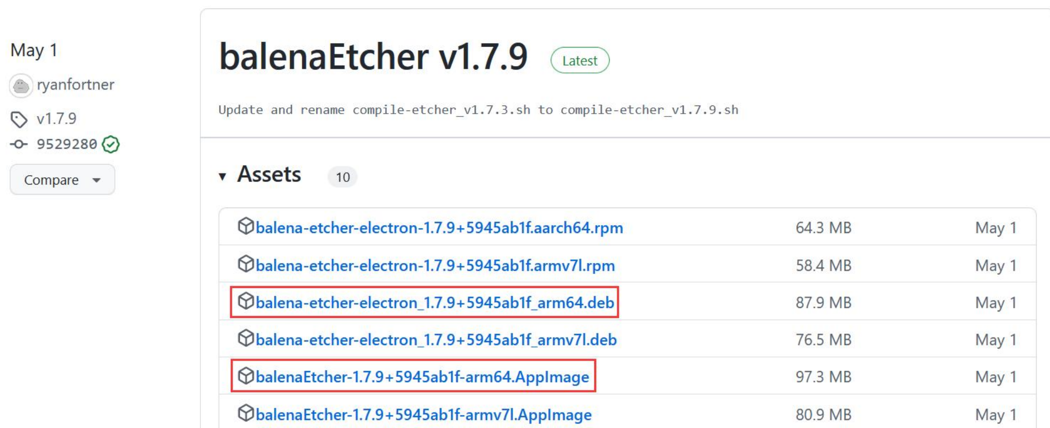
1) The download address of balenaEtcher arm64 version is:

a. The download address of the deb installation package is as follows, which needs to be installed before it can be used

https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balena-etcher-electron_1.7.9+5945ab1f_arm64.deb

b. The download address of the AppImage version that does not need to be installed is as follows:

https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balenaEtcher-1.7.9+5945ab1f-arm64.AppImage

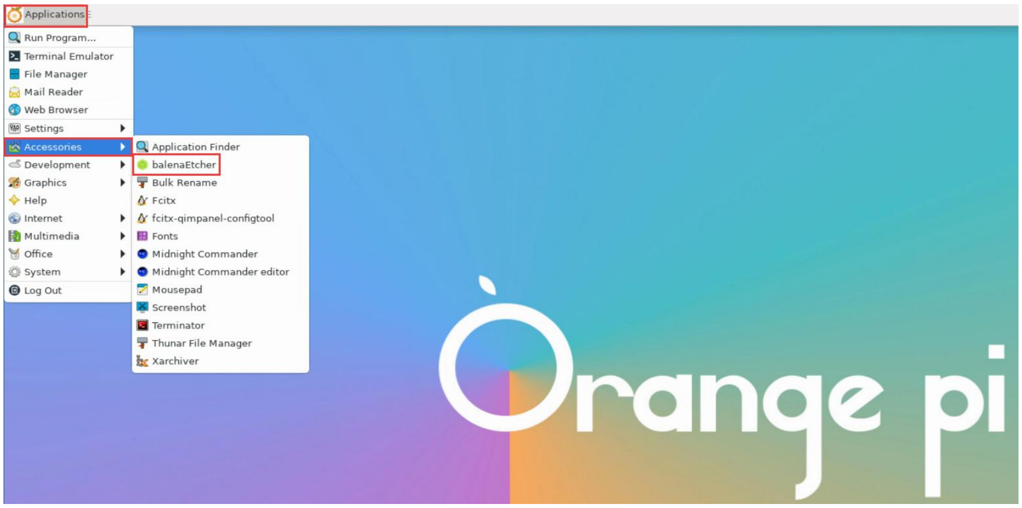


2) How to install and use the deb version of balenaEtcher:

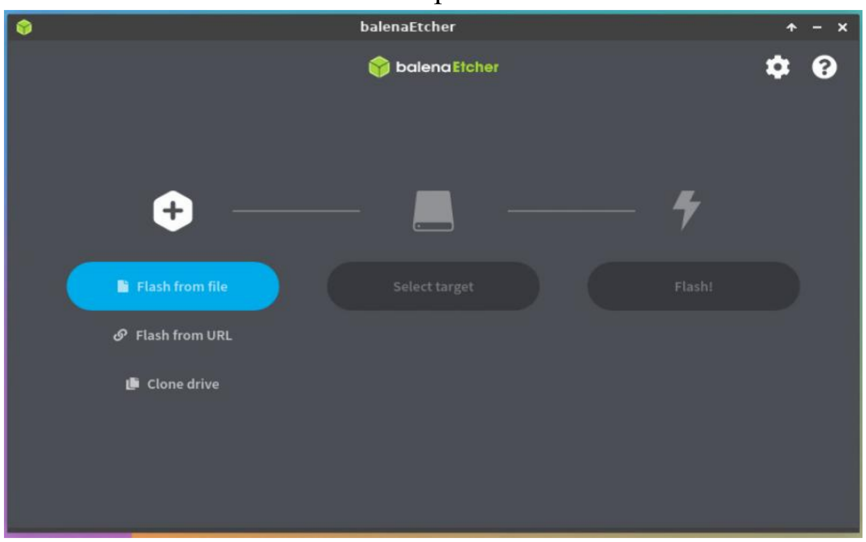
a. The deb version of balenaEtcher installation command is as follows:

```
orangepi@orangepi:~$ sudo apt install -y \
--fix-broken ./balena-etcher-electron_1.7.9+5945ab1f_arm64.deb
```

b. After the deb version of balenaEtcher is installed, it can be opened in the Application



c. The interface after balenaEtcher is opened is as follows:



3) How to use AppImage version of balenaEtcher:

a. First add permissions to balenaEtcher

```
orangepi@orangepi:~/Desktop$ chmod +x balenaEtcher-1.7.9+5945ab1f-arm64.AppImage
```

b. Then select the AppImage version balenaEtcher, then click the right mouse button, and then click Execute to open balenaEtcher

3. 22. How to install BaoTa Linux panel

BaoTa Linux panel is a server management software that improves the efficiency of operation and maintenance. It supports one-click configuration of more than 100 server management functions such as LAMP/LNMP/cluster/monitoring/website/FTP/database/JAVA (excerpted from BaoTa Linux official website)

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

```
orangepi@orangepi:~$ sudo sed -i 's/nosuid/&,size=2G/' /etc/fstab
orangepi@orangepi:~$ sudo reboot
```

2) After restarting, you can see that the size of **/tmp** space has changed to 2G

```
orangepi@orangepi:~$ df -h | grep "/tmp"
tmpfs                2.0G     12K   2.0G     1% /tmp
```

3) Then enter the following command in the Linux system to start the installation of the BaoTa

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

4) Then the installer will remind whether to install **Bt-Panel** to the **/www** folder, and then enter y

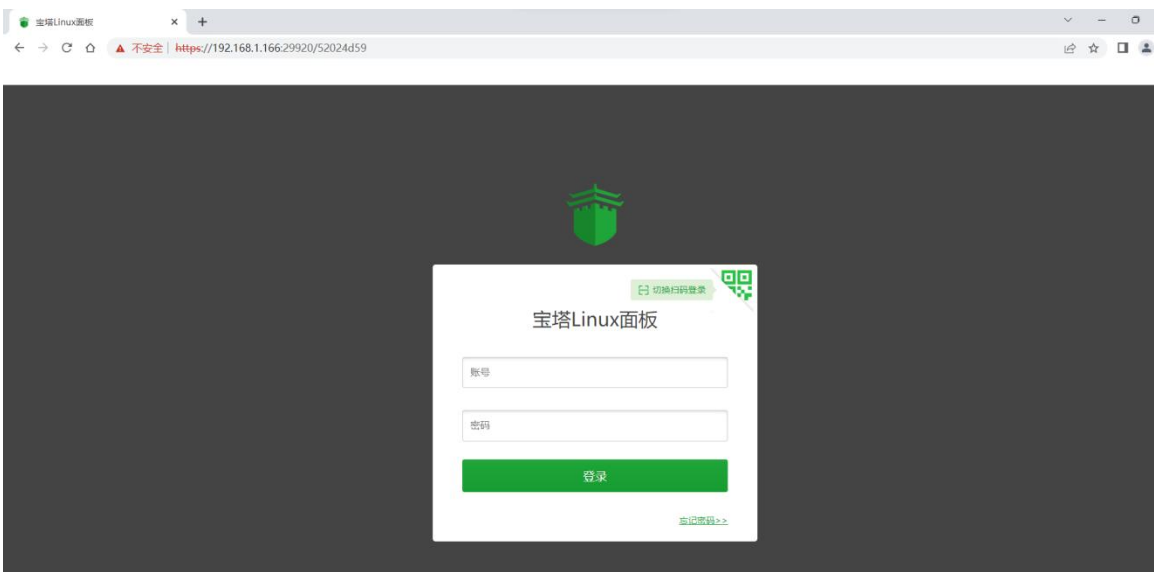
```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| The WebPanel URL will be http://SERVER_IP:8888 when installed.
+----------------------------------------------------------------------

Do you want to install Bt-Panel to the /www directory now?(y/n): y
```

5) The next thing to do is to wait patiently. When you see the following print information

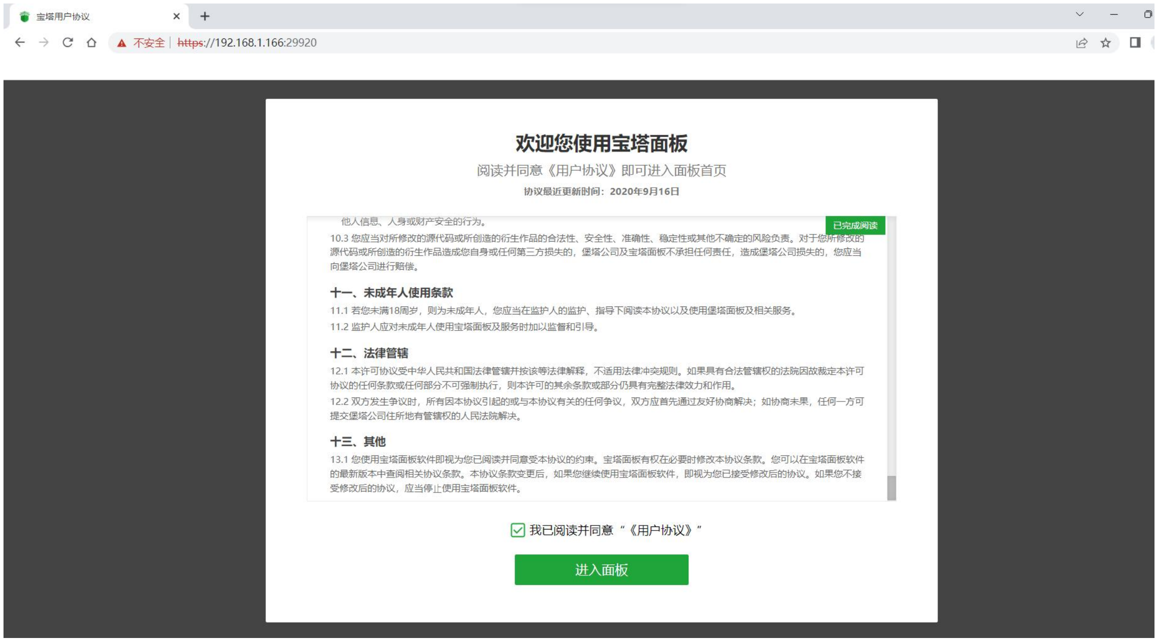
output from the terminal, it means that the BaoTa has been installed. The entire installation process takes about 34 minutes, and there may be some differences depending on the network speed

```
==================================================================
Congratulations! Installed successfully!
==================================================================
外网面板地址: https://183.15.204.194:29920/52024d59
内网面板地址: https://192.168.1.166:29920/52024d59
username: 4qhagfrc
password: 27b2d026
If you cannot access the panel,
release the following panel port [29920] in the security group
若无法访问面板，请检查防火墙/安全组是否有放行面板[29920]端口
因已开启面板自签证书，访问面板会提示不匹配证书，请参考以下链接配置证书
https://www.bt.cn/bbs/thread-105443-1-1.html
==================================================================
Time consumed: 34 Minute!
orangepi@orangepi:~$
```

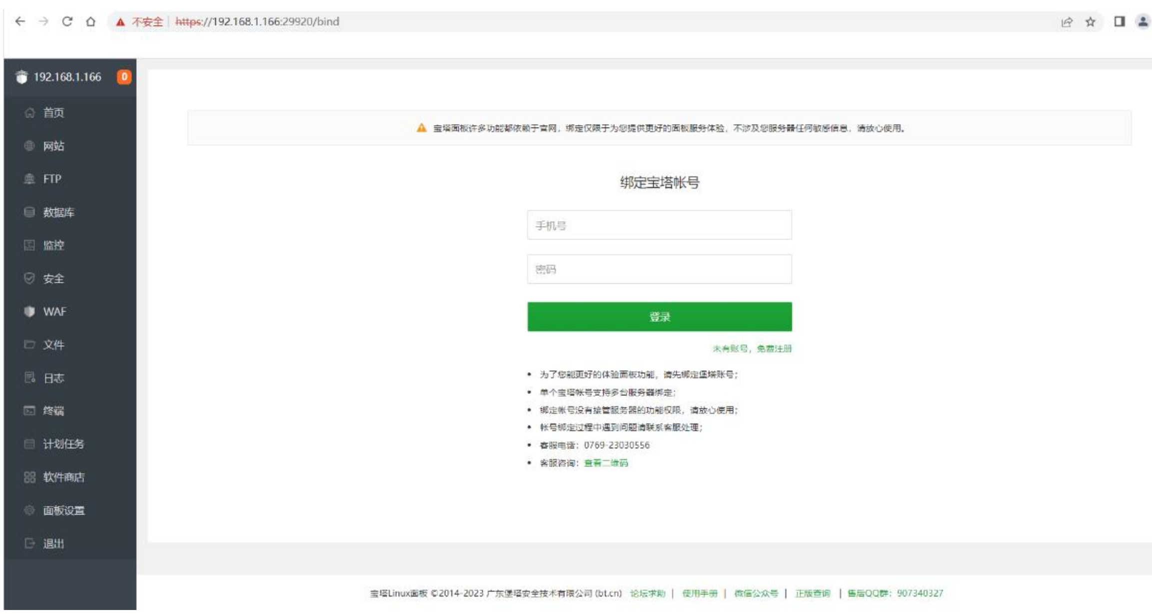
6) At this time, enter the panel address shown above in the browser to open the login interface of the BaoTa Linux panel, and then enter the **username** and **password** shown in the above figure at the corresponding position to log in.



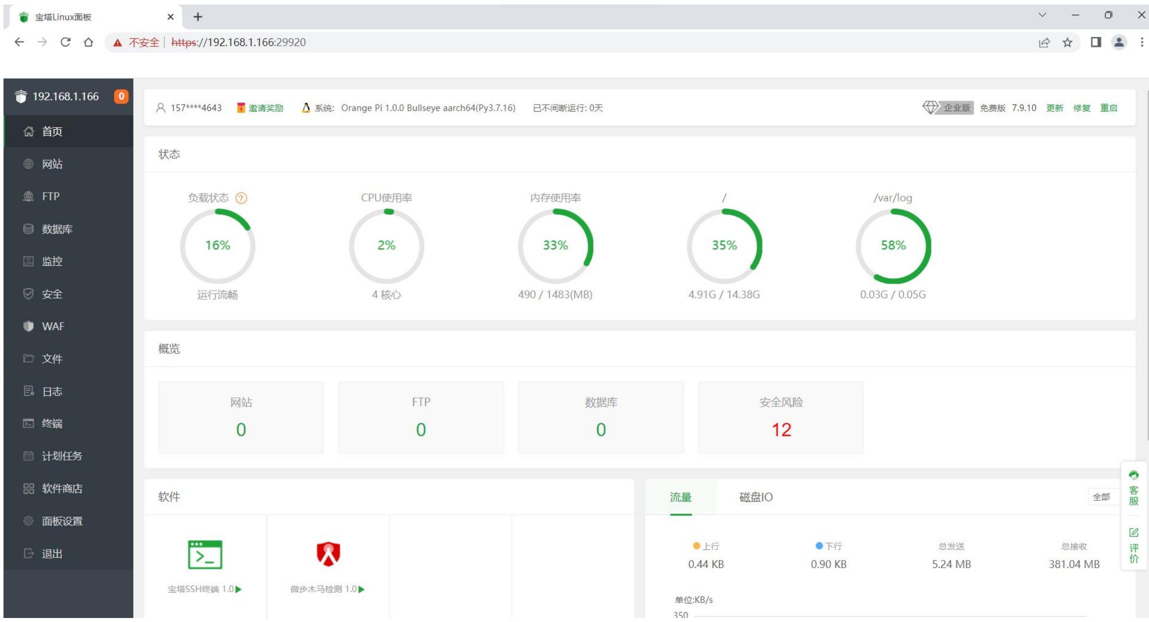
7) After successfully logging into the pagoda, the following welcome interface will pop up. First, please read the user notice in the middle and drag it to the bottom, then you can select "I have agreed and read the "User Agreement"", and then click "Enter the panel" You can enter the BaoTa



8) If you do not have an account, you can go to the official website of the BaoTa (http://www.aapanel.com/)to register



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



10) For more functions of the pagoda, you can refer to the following information to explore by yourself

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

3. 23. Setting Chinese environment and installing Chinese input method

Note, before installing the Chinese input method, please make sure that the Linux system used by the development board is a desktop system.

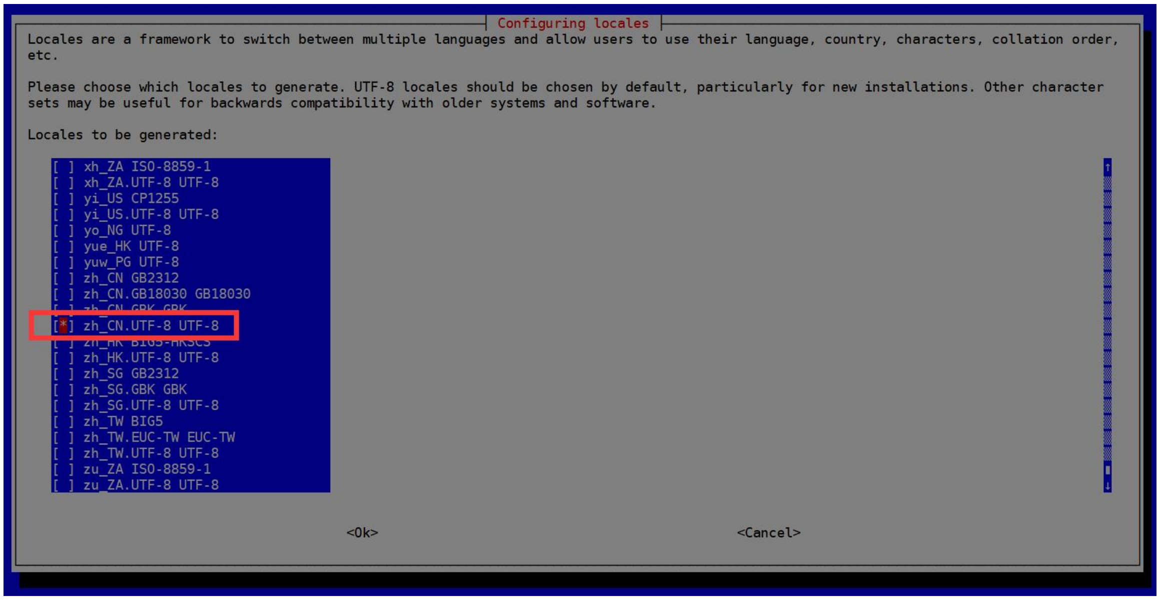
3. 23. 1. How to install Debian system

1) First set the default **locale**to Chinese

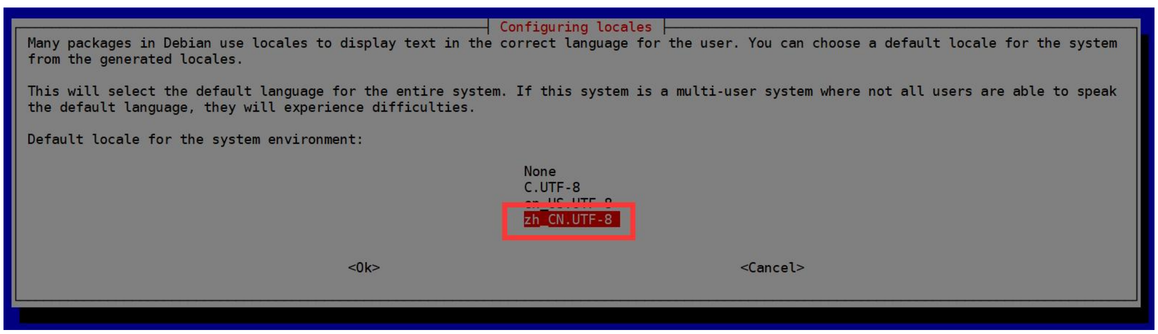
a. Enter the following command to start the configuration **locale**

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
```

b. Then select **zh_CN.UTF-8 UTF-8** in the pop-up interface (move up and down through the up and down direction keys on the keyboard, select through the space bar, and finally move the cursor to**<OK>**through the Tab key, and then return to car)



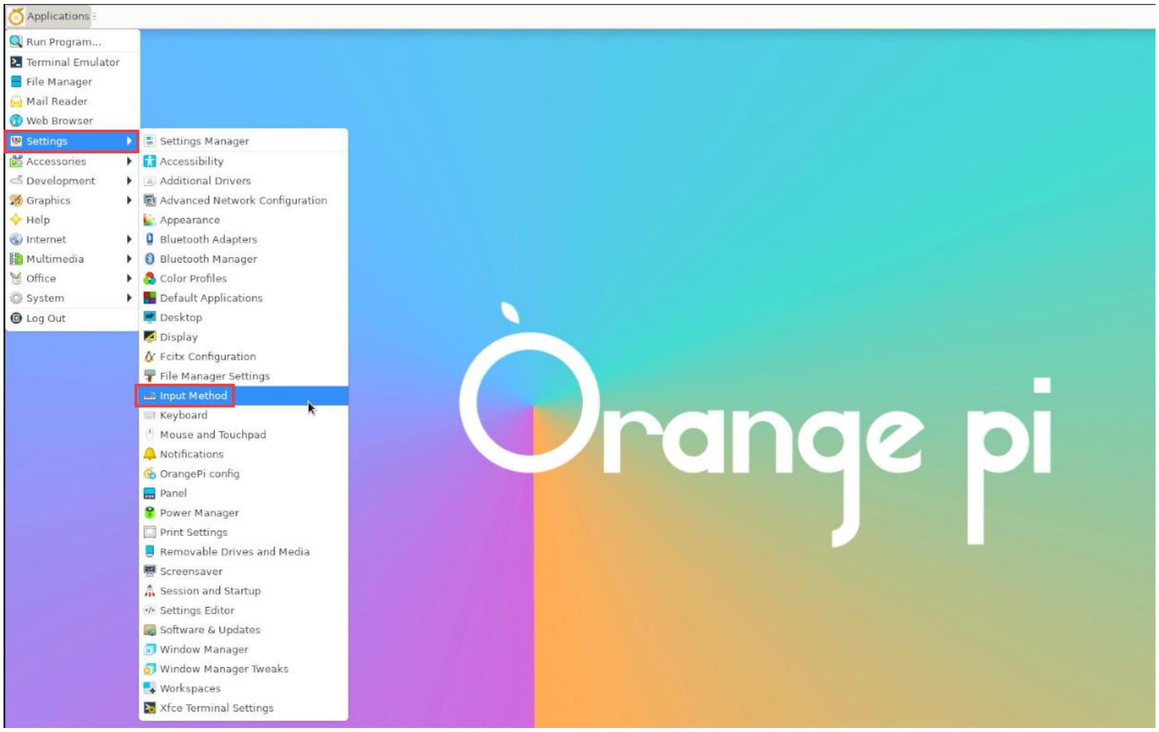
c. Then set the default **locale** as **zh_CN.UTF-8**



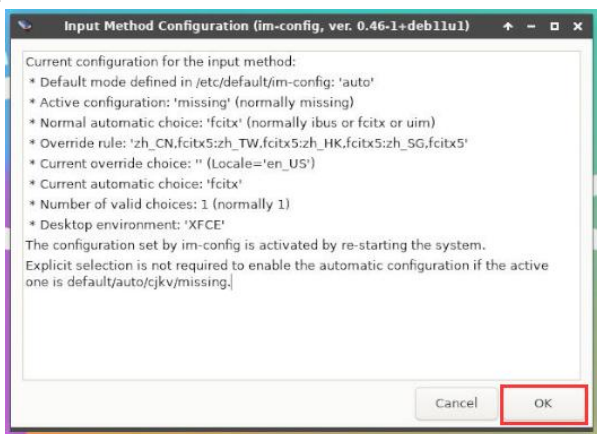
d. After exiting the interface, the **locale** setting will start, and the output displayed on the command line is as follows

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
  en_US.UTF-8... done
  zh_CN.UTF-8... done
Generation complete.
```

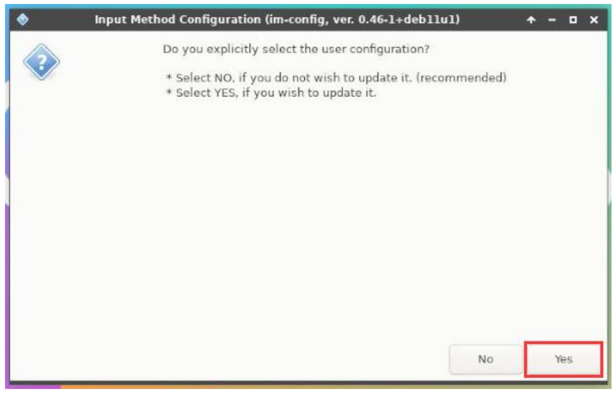
2) Then open **Input Method**



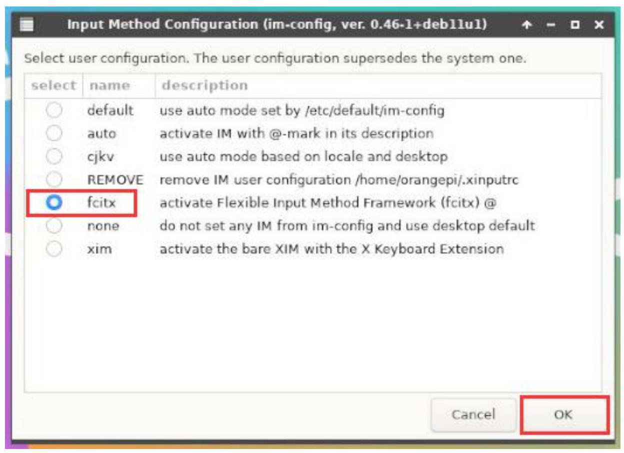
3) Then Select **OK**



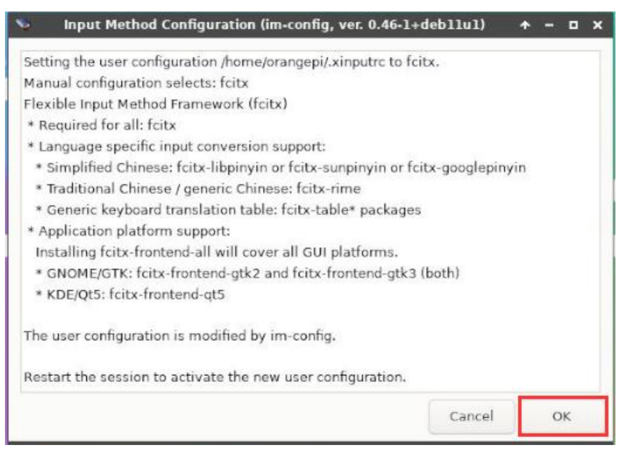
4) Then Select **Yes**



5) Then Select **fcitx**

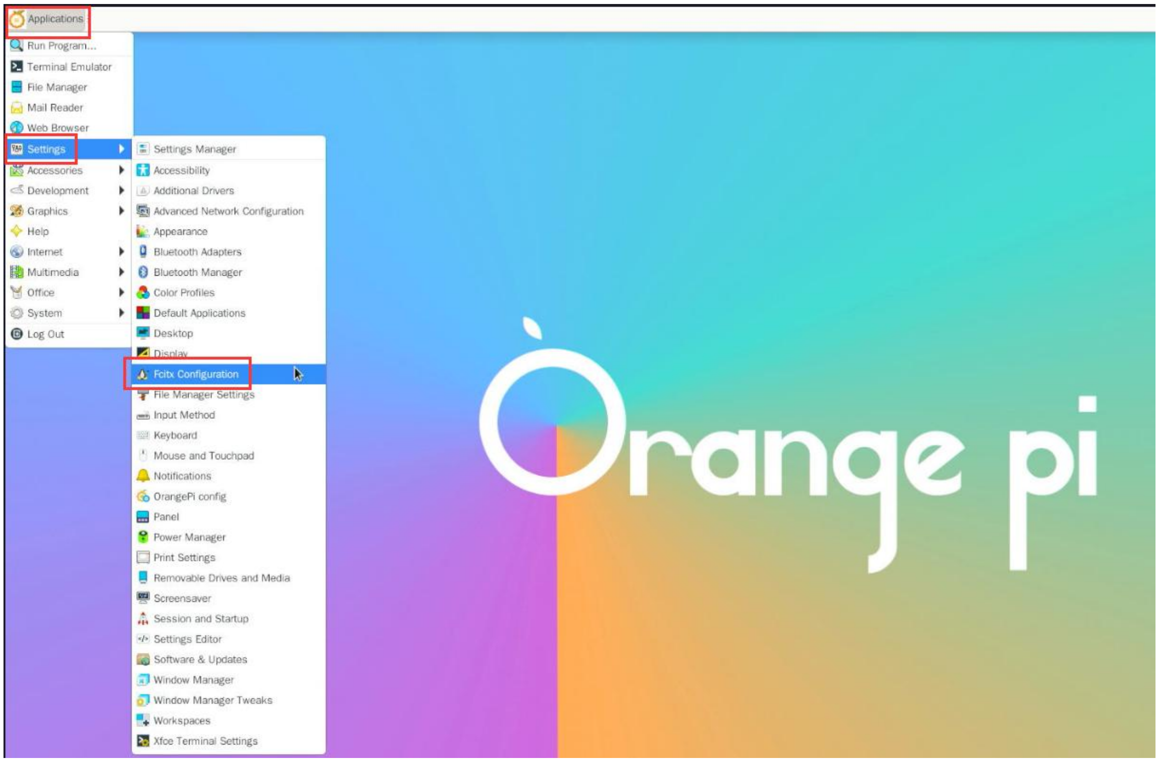


6) Then Select **OK**

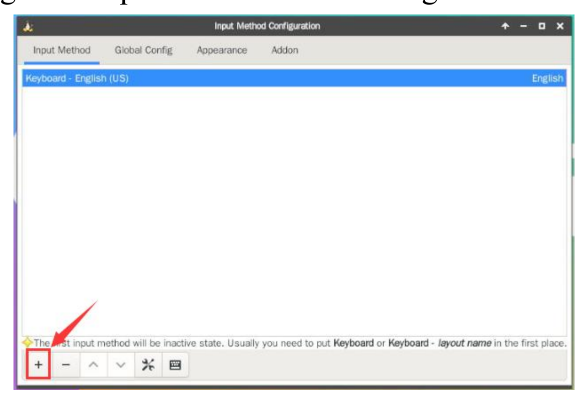


7) **Then restart the Linux system to make the configuration take effect**

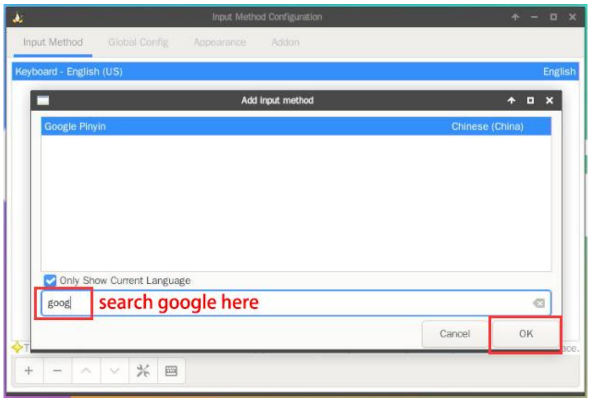
8) Then open **Fcitx configuration**



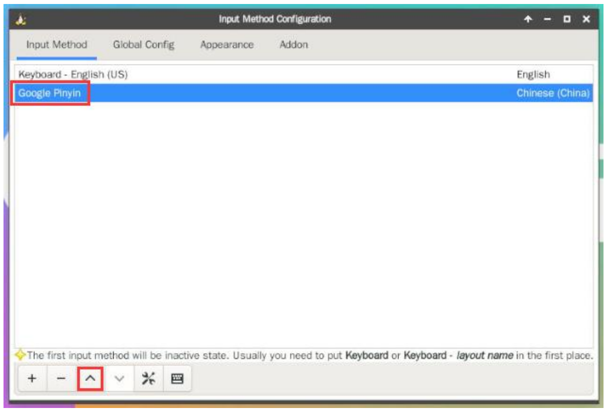
9) Then click the + sign in the position shown in the figure below

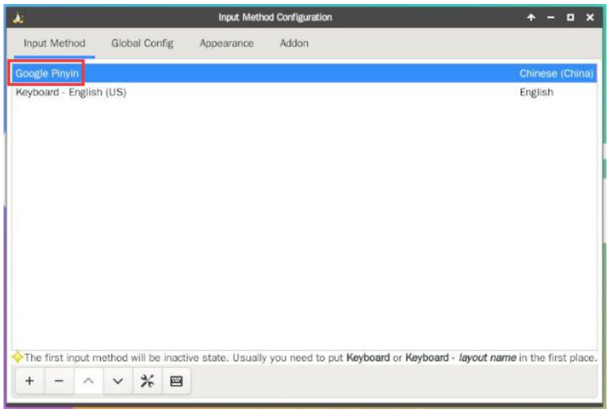


10) Then search **Google Pinyin** and click **OK**



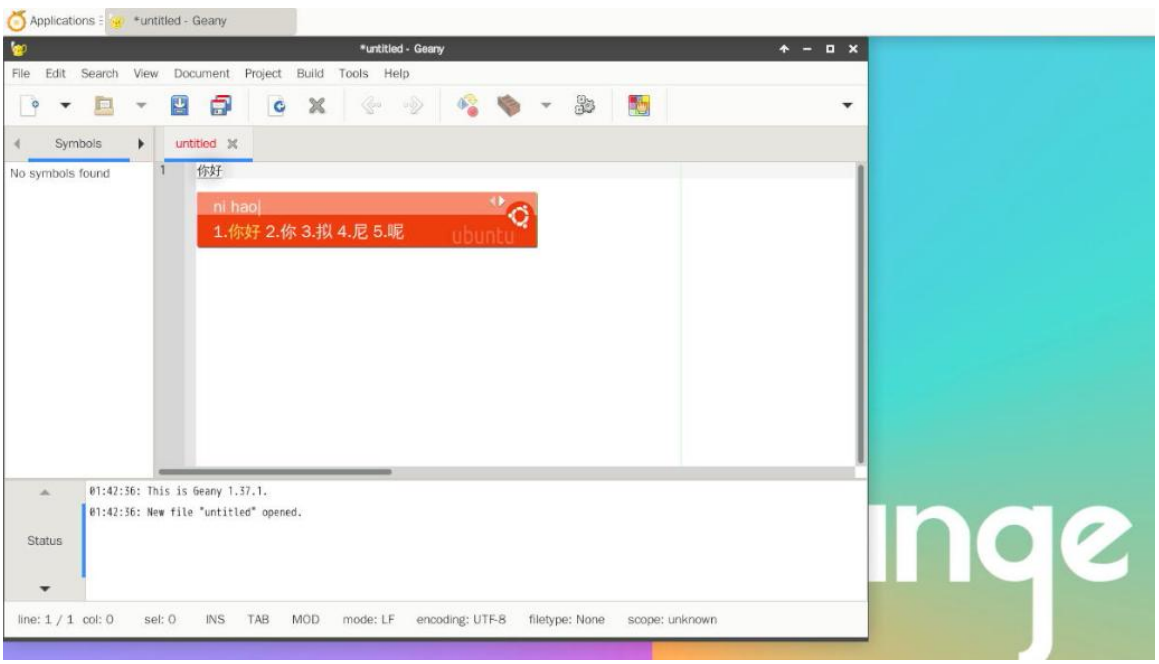
11) Then bring**Google Pinyin** to the front





12) Then open the **Geany** editor to test the Chinese input method

13) The Chinese input method test is as follows



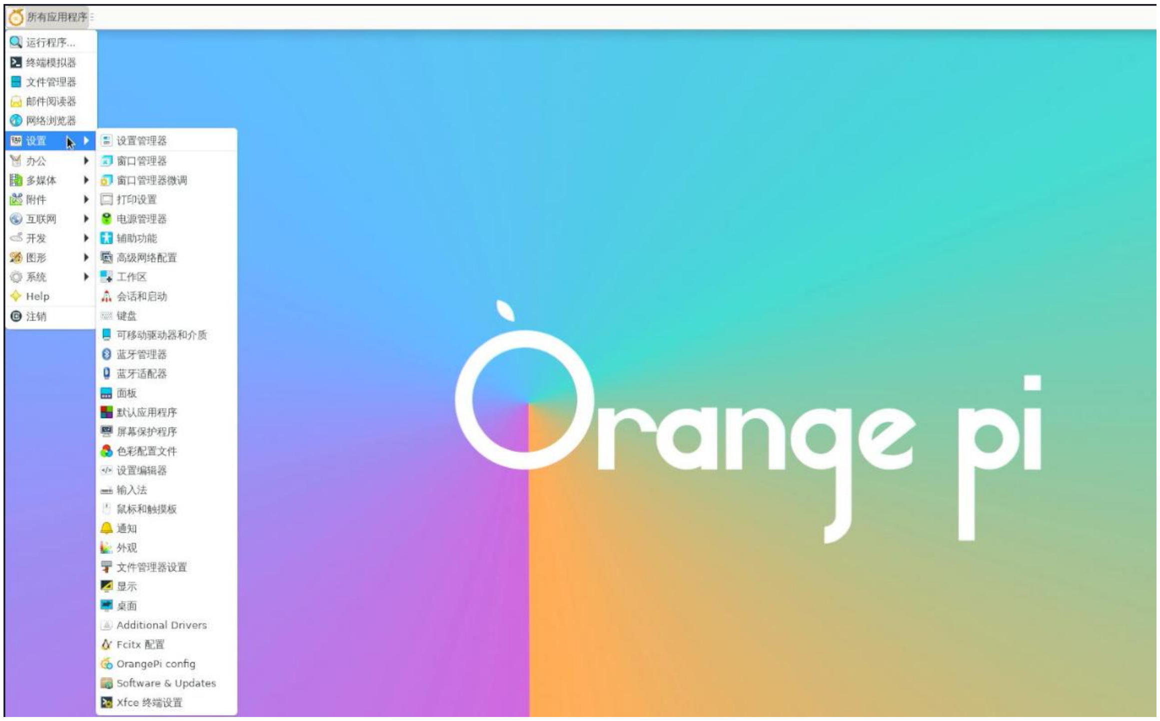
14) The Chinese and English input methods can be switched through the **Ctrl+Space** shortcut key

15) If the entire system needs to be displayed in Chinese, you can set the variables in **/etc/default/locale** to **zh_CN.UTF-8**

```
orangepi@orangepi:~$ sudo vim /etc/default/locale
#  File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
```

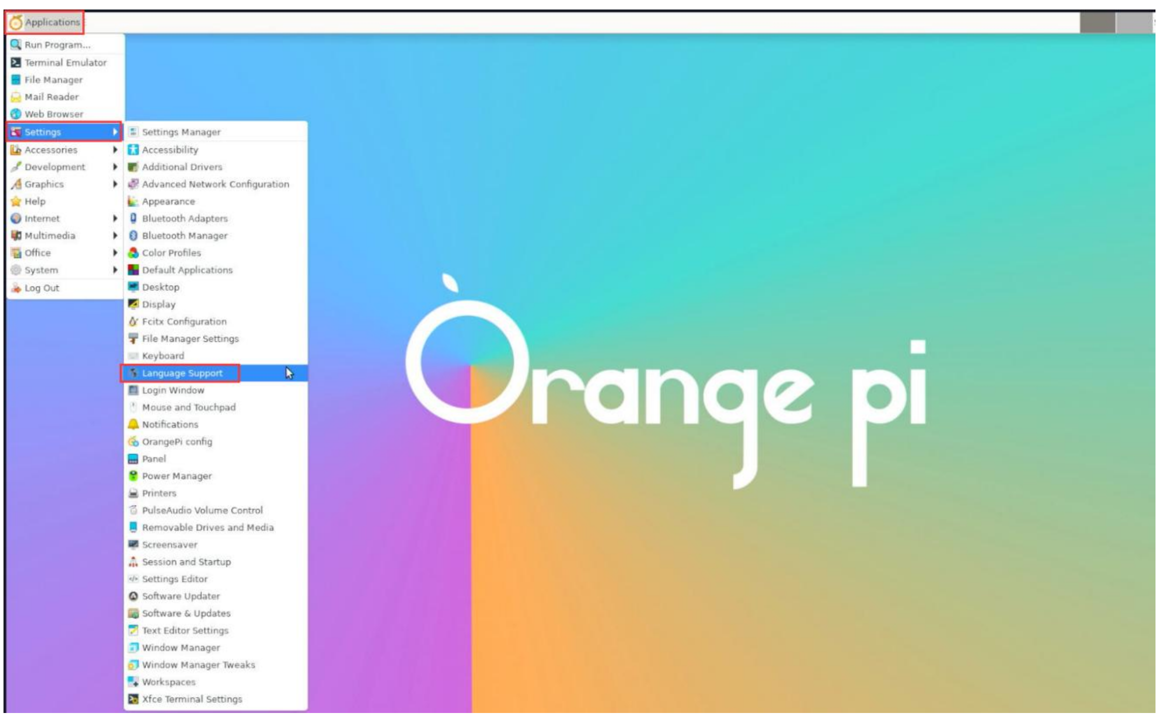
```
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** and you can see that the system is displayed in Chinese

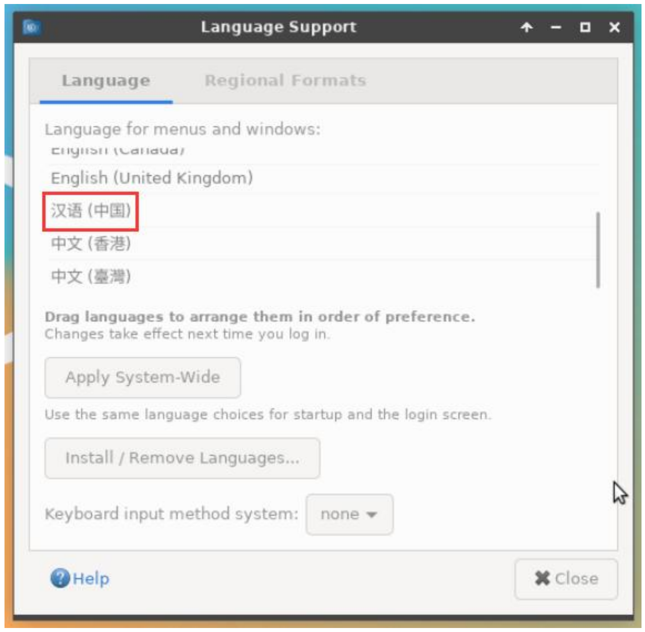


3. 23. 2. How to install Ubuntu 20.04 system

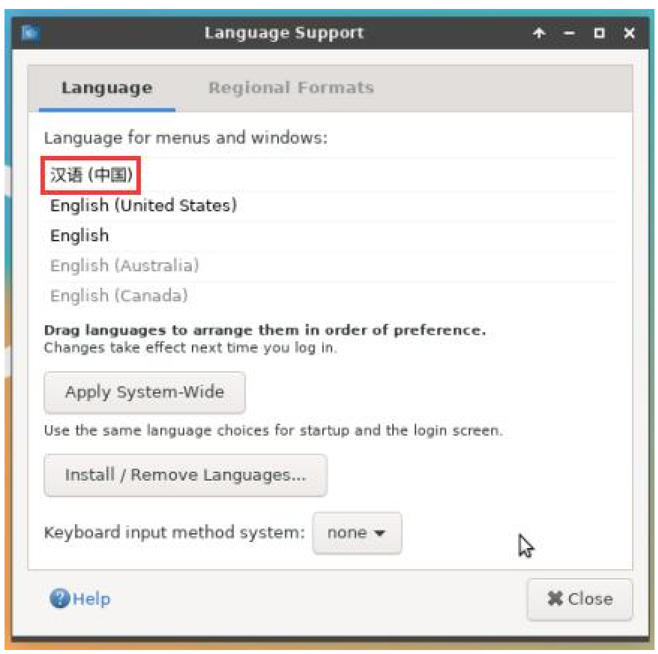
1) Open first **Language Support**



2) Then find the **Chinese (China)** option

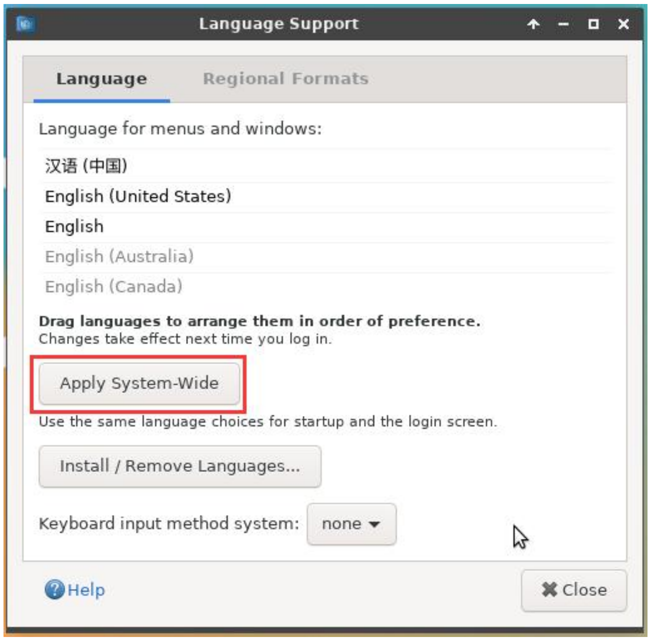


3) Then please use the left mouse button to select **Chinese (China)** and hold it down, then drag it up to the initial position, and the display after dragging is as shown in the figure below:

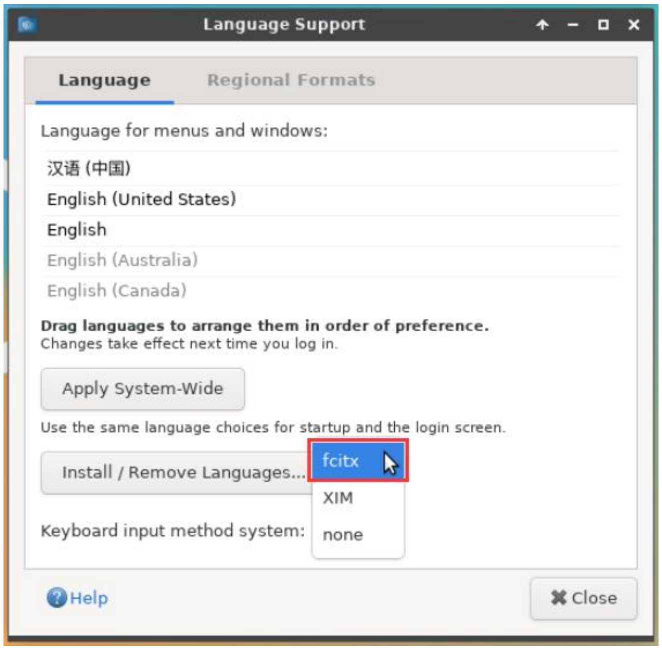


Note that this step is not easy to drag, please be patient and try a few more times.

4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



5) Then set **Keyboard input method system** as **fcitx**

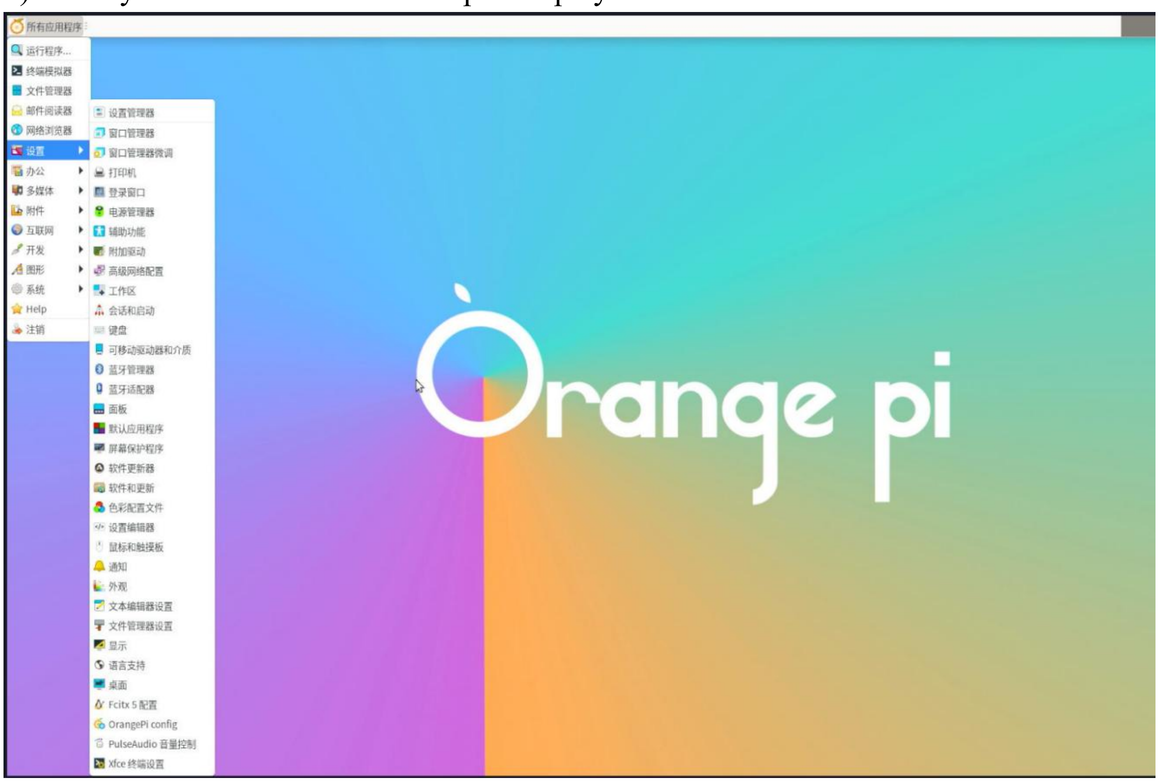


6) **Then restart the Linux system to make the configuration take effect**

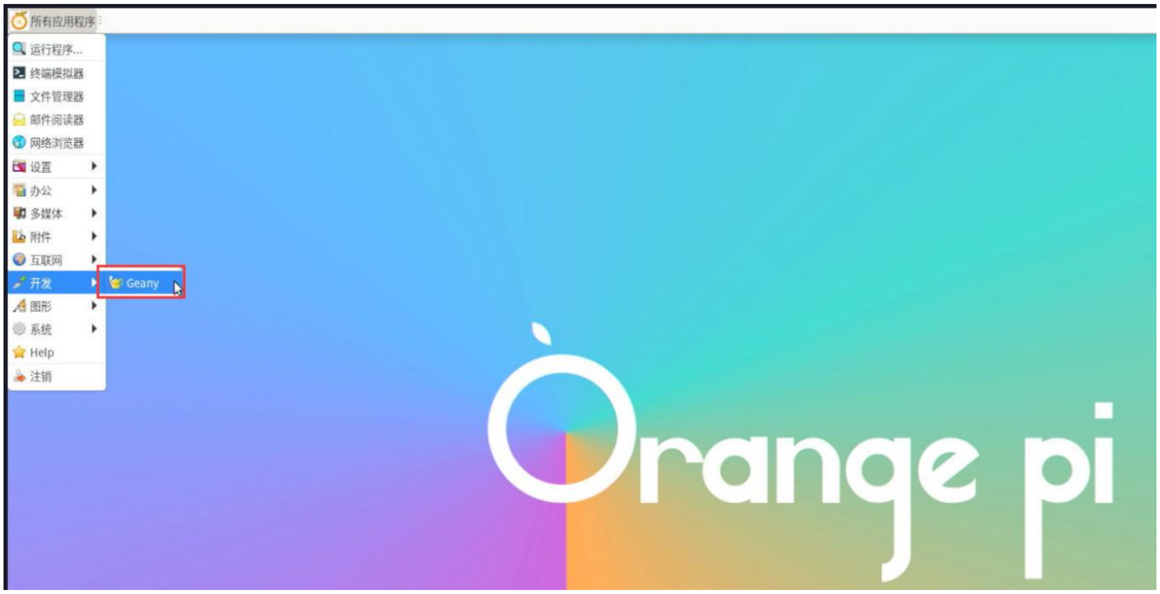
7) After re-entering the system, please choose **not to ask me again** in the following

interface, and then please decide whether the standard folder should also be updated to Chinese according to your preferences

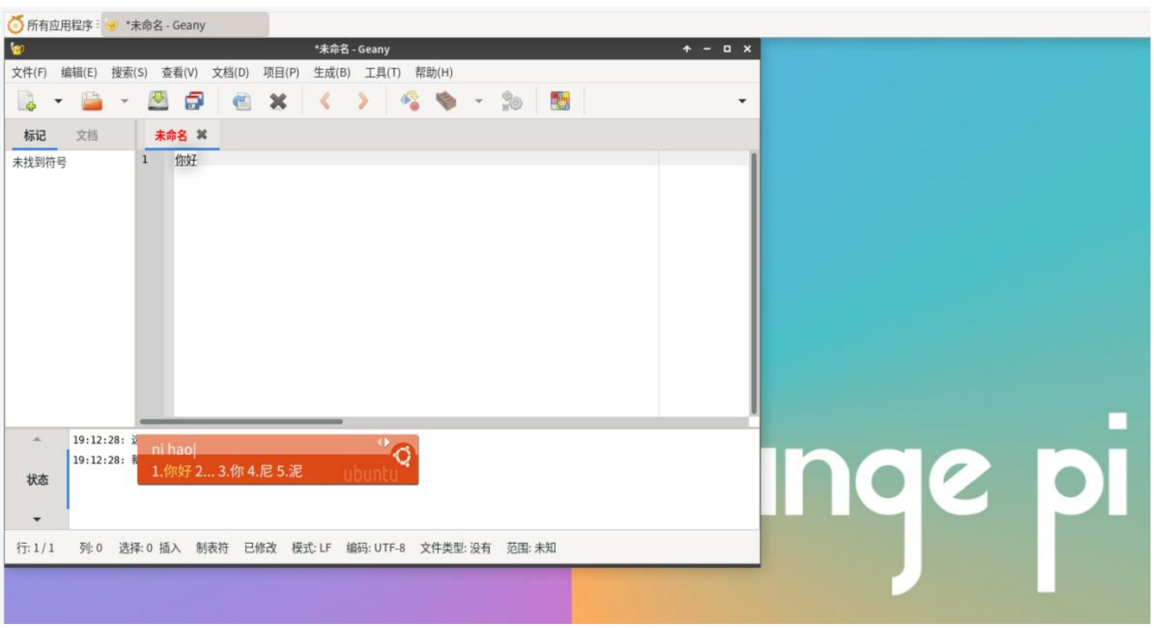
8) Then you can see that the desktop is displayed in Chinese



9) Then we can open **Geany** to test the Chinese input method, the opening method is as shown in the figure below

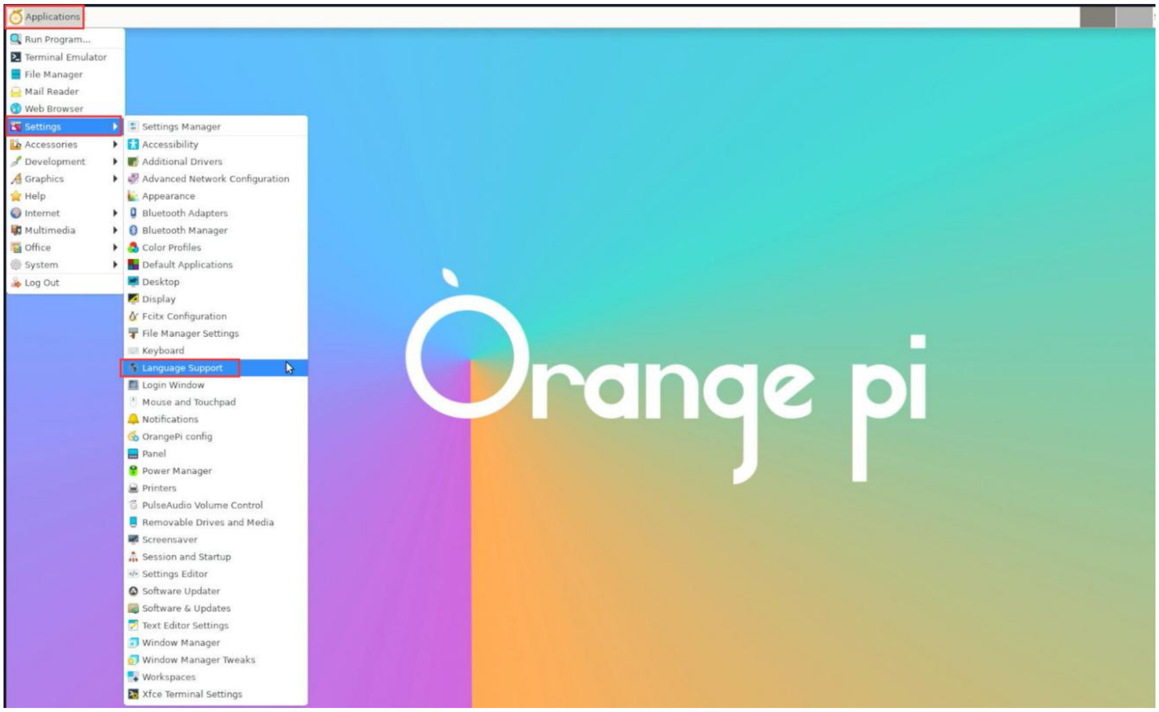


10) After opening**Geany**, the default is English input method, we can switch to Chinese input method through **Ctrl+Space**shortcut key, and then we can input Chinese

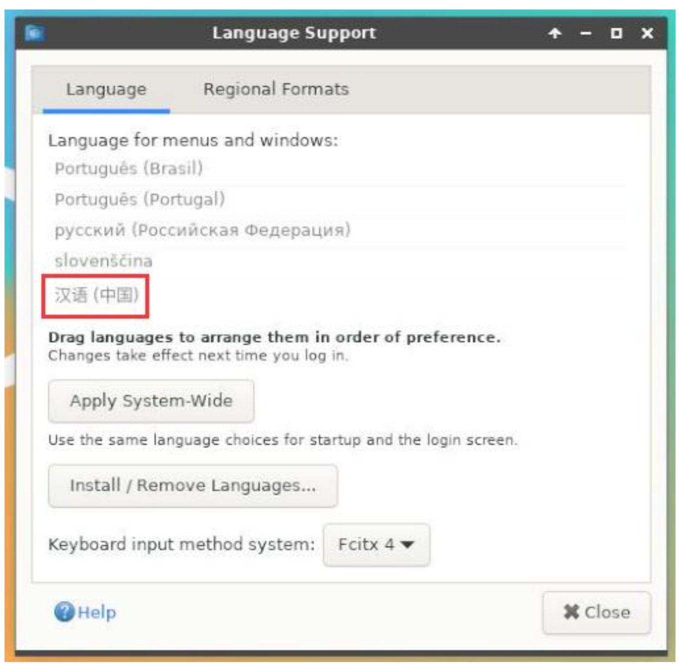


3. 23. 3. How to install Ubuntu 22.04 system

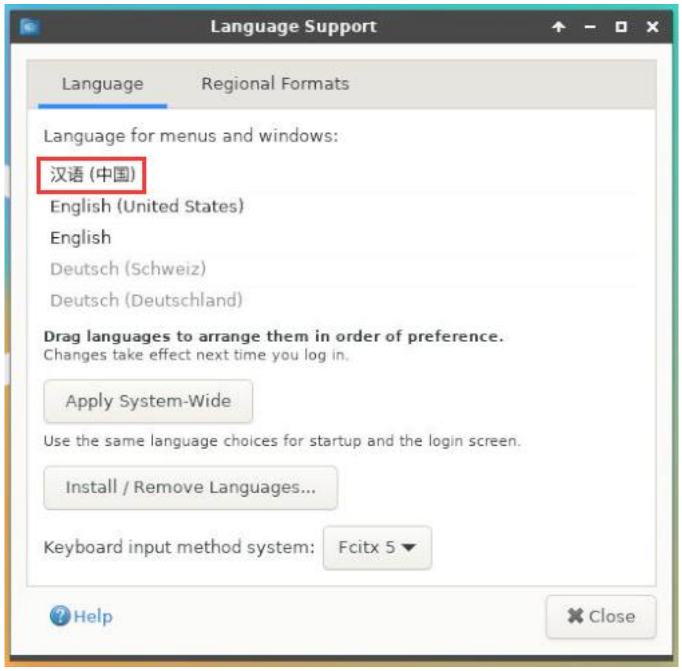
1) Open first **Language Support**



2) Then find the **Chinese (China)** option

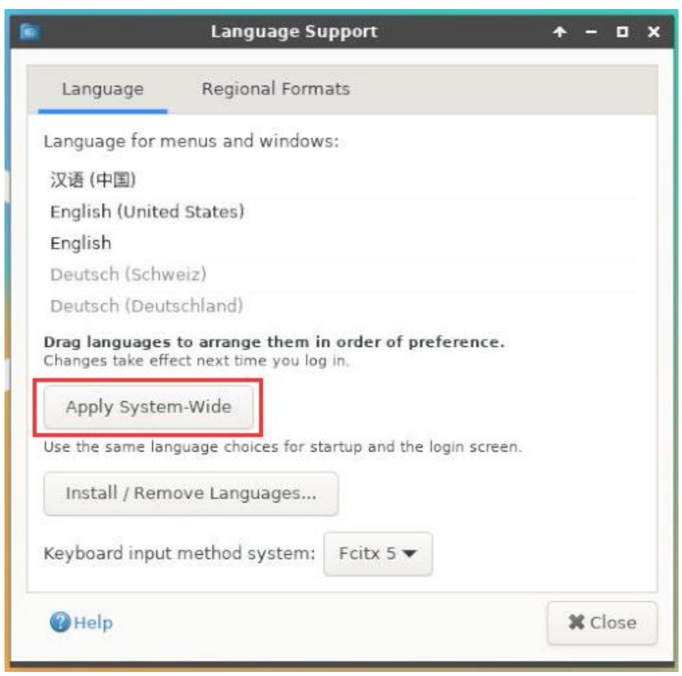


3) Then please use the left mouse button to select **Chinese (China)** and hold it down, then drag it up to the initial position, and the display after dragging is as shown in the figure below:



Note that this step is not easy to drag, please be patient and try a few more times.

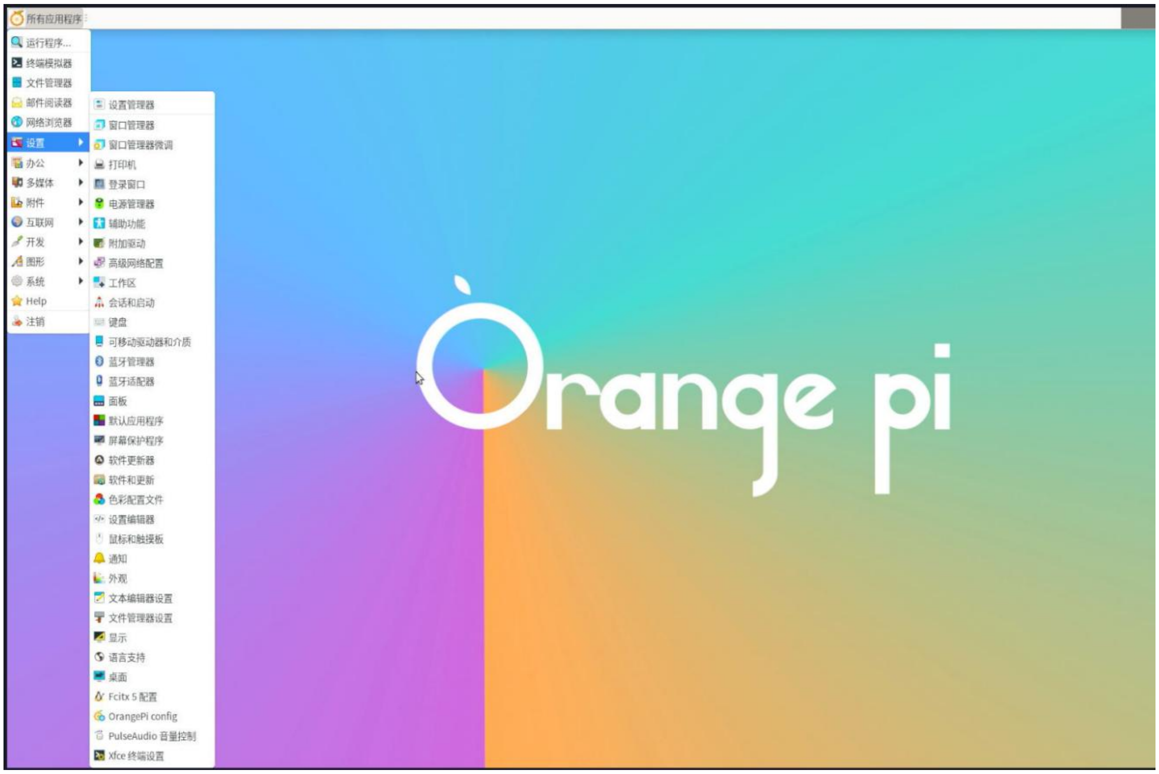
4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



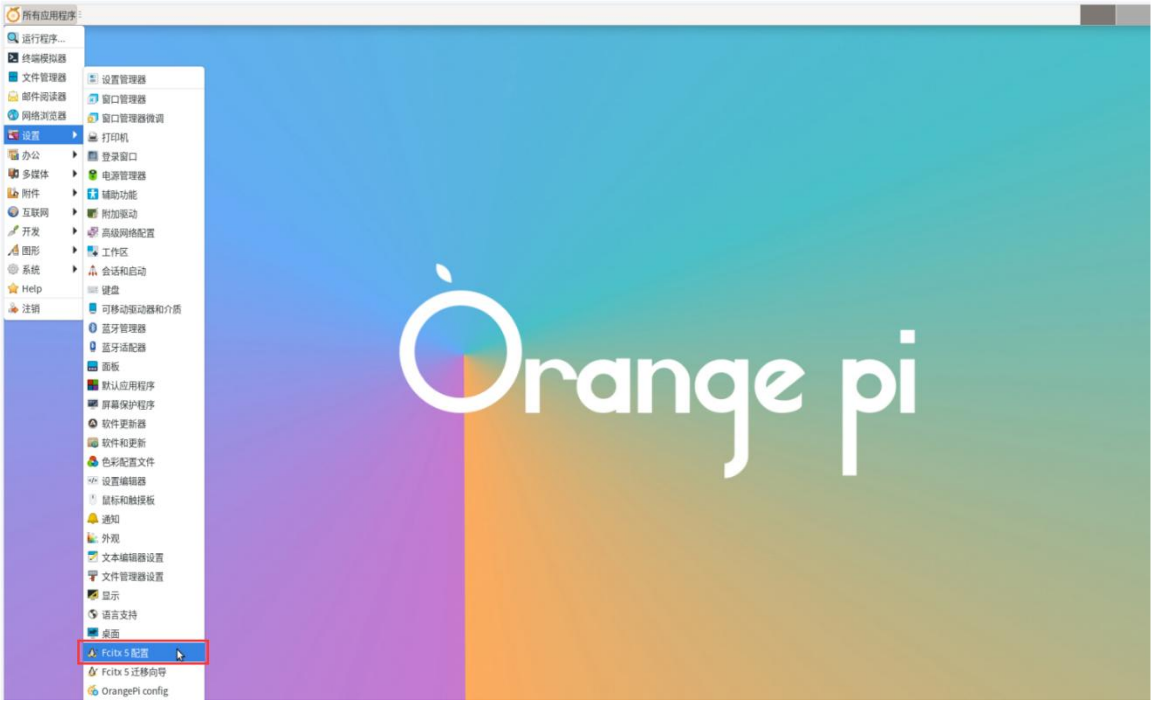
5) **Then restart the Linux system to make the configuration take effect**

6) After re-entering the system, please choose **not to ask me again** in the following interface, and then please decide whether the standard folder should also be updated to Chinese according to your preferences

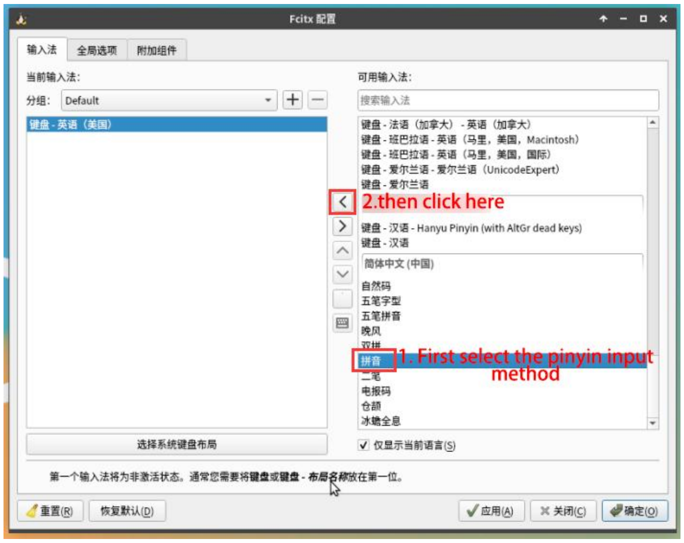
7) Then you can see that the desktop is displayed in Chinese



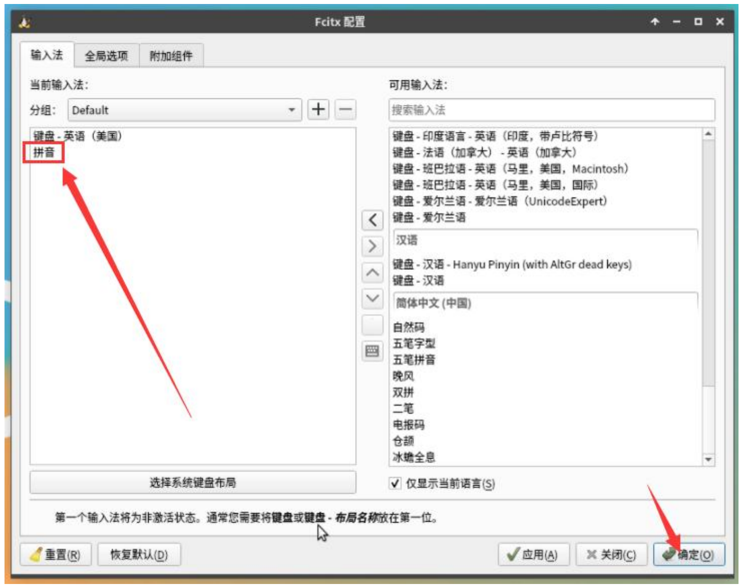
8) Then open the Fcitx5 configuration program



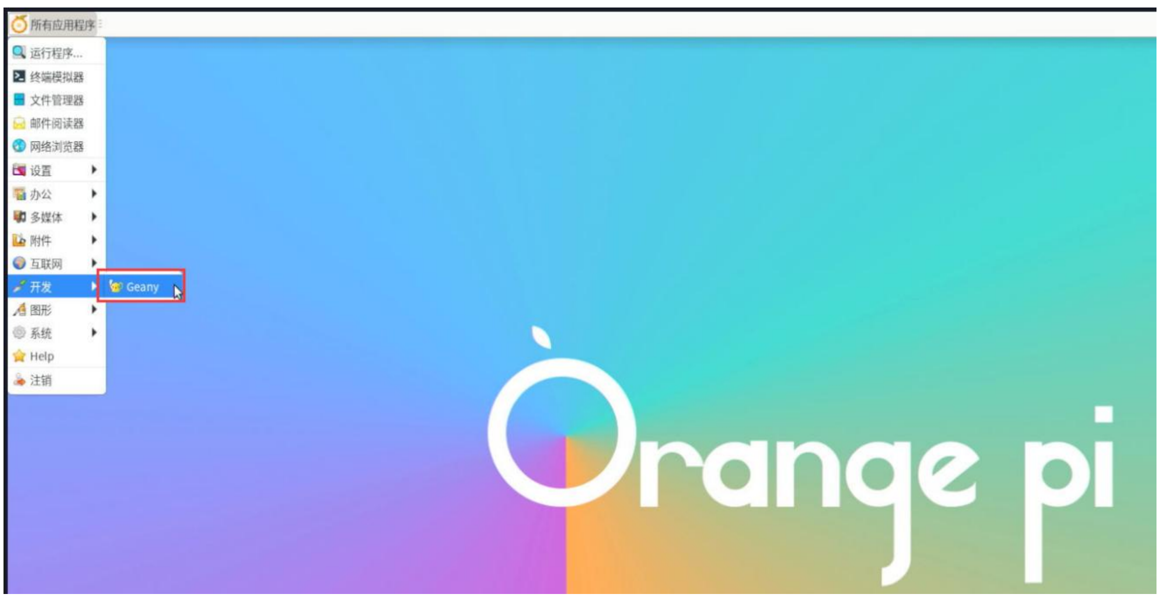
9) Then choose to use Pinyin input method



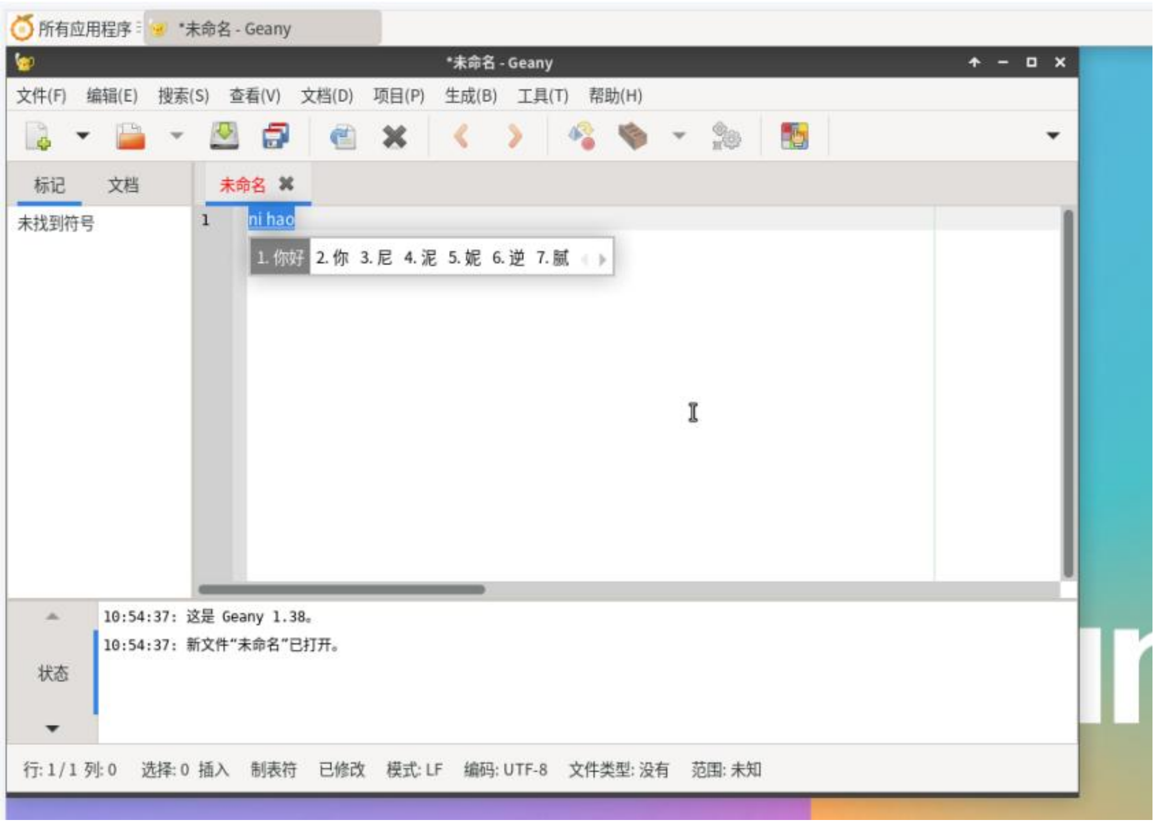
10) The interface after selection is as shown below, and then click OK



11) Then we can open **Geany**to test the Chinese input method, the opening method is as shown in the figure below



12) After opening **Geany**, the default is English input method, we can switch to Chinese input method through **Ctrl+Space** shortcut key, and then we can input Chinese



3. 24. How to remotely log in to the desktop of the Linux system

3. 24. 1. 3.24.1. Remote login using NoMachine

Please make sure that the Ubuntu or Debian system installed on the development board is a desktop version. In addition, NoMachine also provides detailed usage documents. It is strongly recommended to read this document to familiarize yourself with the use of NoMachine. The document link is as follows: Please make sure that the Ubuntu or Debian system installed on the development board is a desktop version. In addition, NoMachine also provides detailed usage documents. It is strongly recommended to read this document to familiarize yourself with the use of NoMachine. The document link is as follows:

https://knowledgebase.nomachine.com/DT10R00166

NoMachine supports Windows, Mac, Linux, iOS and Android platforms, so we can remotely log in and control the Orange Pi development board through NoMachine on a variety of devices. The following demonstrates how to remotely log

in to the Linux system desktop of the Orange Pi development board through NoMachine in Windows. For installation methods on other platforms, please refer to the official documentation of NoMachine.

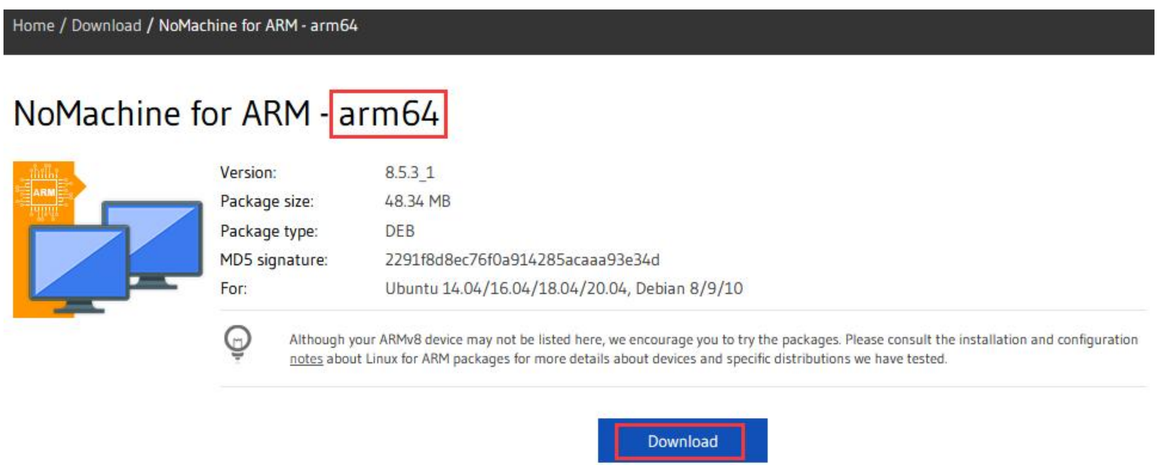
Before operation, please make sure that the Windows computer and the development board are in the same LAN, and you can log in to the Ubuntu or Debian system of the development board through ssh.

1) First download the installation package of the NoMachine software Linux arm64 deb version, and then install it in the Linux system of the development board

a. Since RK3566 is an SOC with ARMv8 architecture, the system we use is Ubuntu or Debian, so here we need to download the **NoMachine for ARM ARMv8 DEB** installation package. The download link is as follows:

Note that this download link may change, please look for the deb package of the Armv8/Arm64 version.

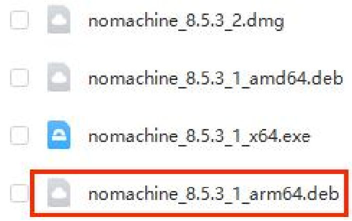
https://downloads.nomachine.com/download/?id=116&distro=ARM



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

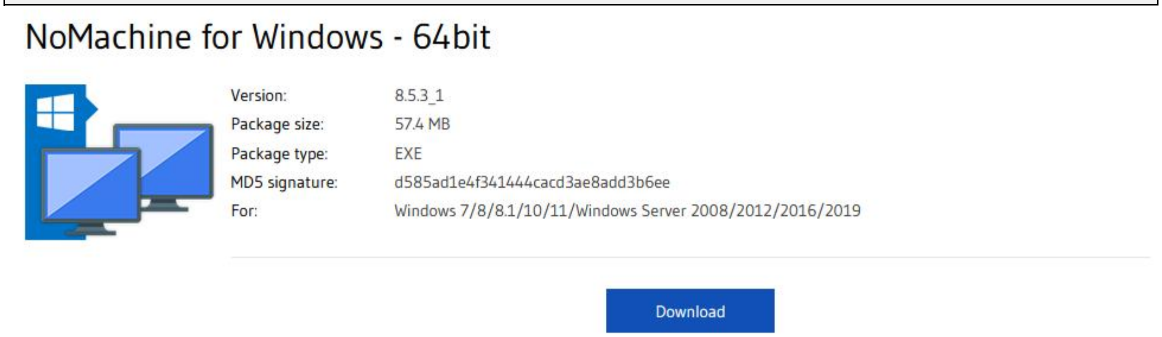
d. Then use the following command to install NoMachine in the Linux system of the development board **NoMachine**

```
orangepi@orangepi:~$ sudo dpkg -i nomachine_x.x.x_x_arm64_arm64.deb
```

2) Then download the installation package of the Windows version of the NoMachine software, the download address is as follows

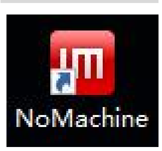
Note that this download link may change.

https://downloads.nomachine.com/download/?id=8



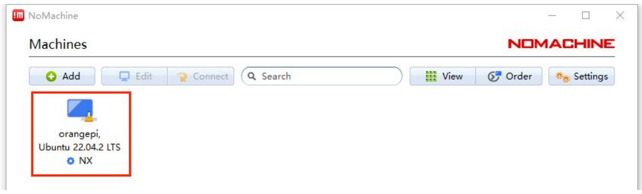
3) Then install NoMachine in Windows, **please restart the computer after installation**

4) Then open NoMachine in Window **NoMachine**

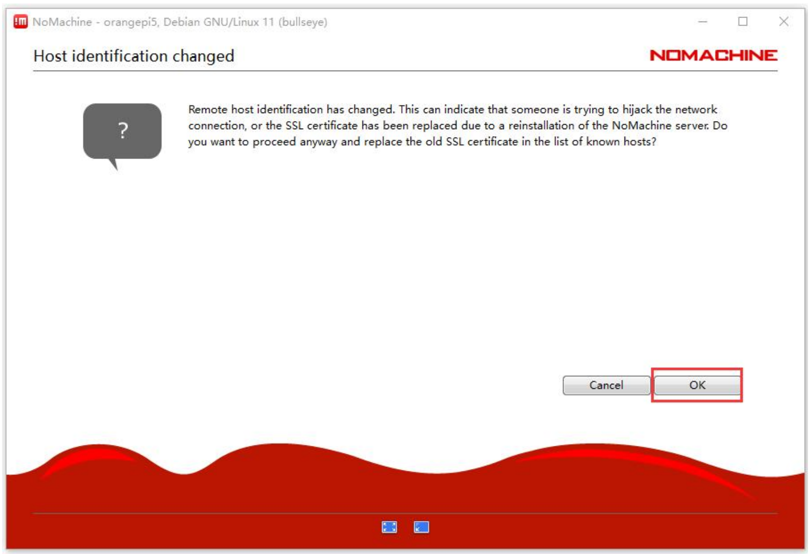


5) After NoMachine starts, it will automatically scan other devices installed with

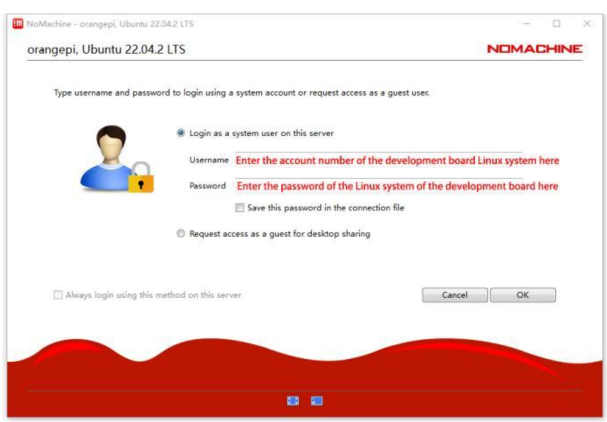
NoMachine in the LAN. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices, and then click the position shown in the red box in the figure below You can start to log in to the Linux system desktop of the development board



6) Then Click **OK**

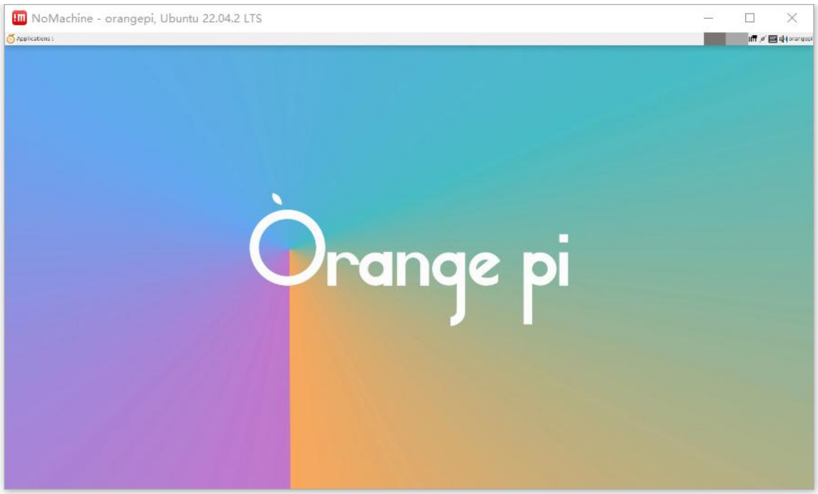


7) Then enter the user name and password of the Linux system of the development board in the corresponding position in the figure below, and then click **OK** to start logging in



8) Then click OK in the next interface

9) Finally, you can see the desktop of the development board Linux system



3. 24. 2. Use VNC to log in remotely

Before operation, please make sure that the Windows computer and the development board are in the same LAN, and you can log in to the Ubuntu or Debian system of the development board through ssh.

Ubuntu20.04 has many problems testing VNC, please do not use this method.

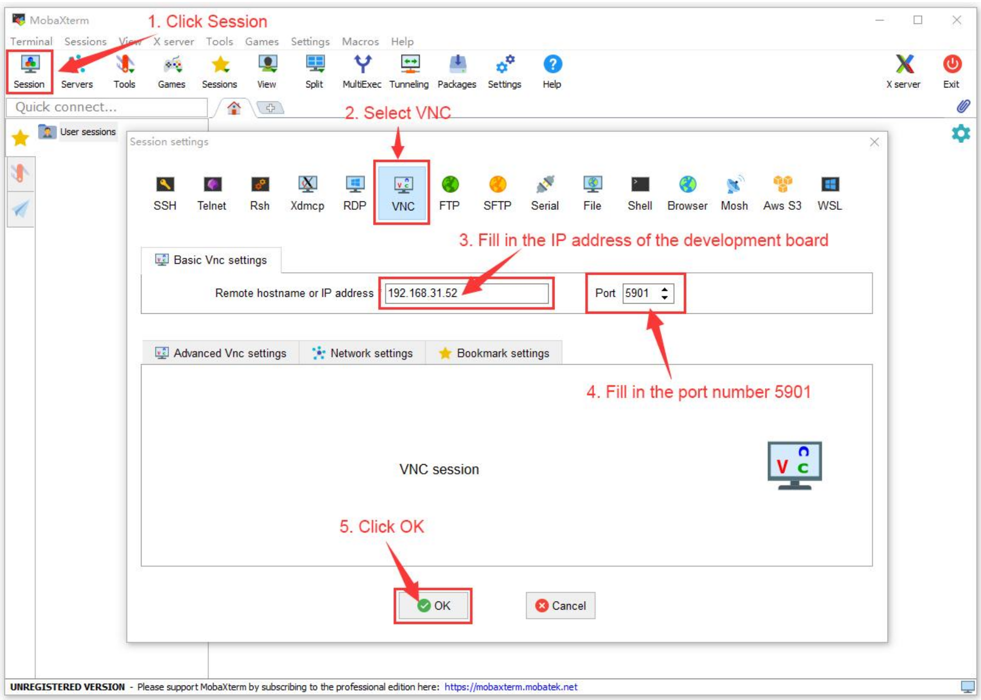
1) First run the **set_vnc.sh** script to set up vnc, **remember to add sudo permission**

```
orangepi@orangepi:~$ sudo set_vnc.sh
You will require a password to access your desktops.

Password:          # Set the vnc password here, 8 characters
Verify:            # Set the vnc password here, 8 characters
Would you like to enter a view-only password (y/n)? n
xauth:    file /root/.Xauthority does not exist

New 'X' desktop is orangepicm4:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepicm4:1.log

Killing Xtightvnc process ID 3047
```

```
New 'X' desktop is orangepicm4:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepicm4:1.log
```

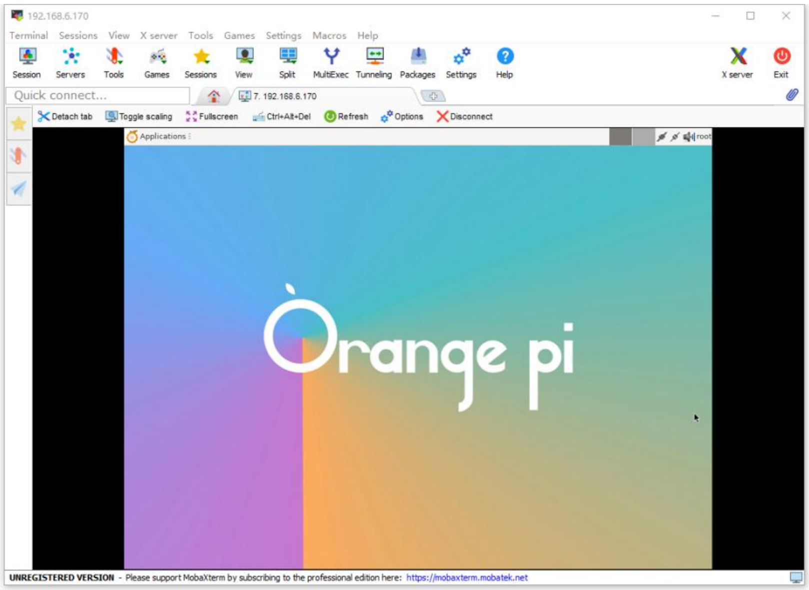
2) The steps to use MobaXterm software to connect to the Linux system desktop of the development board are as follows:

a. First click on Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm



b. Then enter the VNC password set earlier

c. After successful login, the interface is displayed as shown in the figure below, and then you can remotely operate the desktop of the Linux system on the development board



3. 25. Some programming language tests supported by Linux system

3. 25. 1. Debian Bullseye System

1) Debian Bullseye has a gcc compilation tool chain installed by default, which can directly compile C language programs in the Linux system of the development board

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 10.2.1-6) 10.2.1 20210110
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

```

```
        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bullseye has Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py**is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bullseye does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk, the latest version in Debian Bullseye is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

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

c. Write the Java version **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. 25. 2. Debian Bookworm System

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

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 12.2.0-14) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bookworm has Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bookworm does not install Java compilation tools and 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 version of Java

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

c. Write the Java version **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. 25. 3. Ubuntu Focal system

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

a. The version of gcc is as follows

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

```
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Ubuntu Focal has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.8.10 (default, Nov 14 2022, 12:59:47)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Ubuntu Focal does not install Java compilation tools and operating environment by

default

a. You can use the following command to install openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
openjdk 17.0.2 2022-01-18
OpenJDK Runtime Environment (build 17.0.2+8-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 17.0.2+8-Ubuntu-120.04, mixed mode, sharing)
```

c. Write the Java version **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. 25. 4. Ubuntu Jammy system

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

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.2.0-19ubuntu1) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>
```

```
int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Ubuntu Jammy has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.4 (main, Apr  2 2022, 09:04:19) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Ubuntu Jammy does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk-18

```
orangepi@orangepi:~$ sudo apt install -y openjdk-18-jdk
```

b. After installation, you can check the version of Java

```
orangepi@orangepi:~$ java --version
openjdk 18-ea 2022-03-22
OpenJDK Runtime Environment (build 18-ea+36-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 18-ea+36-Ubuntu-1, mixed mode, sharing)
```

c. Write the Java version **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. 26. How to install QT

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

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

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

a. The qt version that comes with Ubuntu 20.04 is **5.12.8**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/aarch64-linux-gnu
```

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

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

c. The QT version that comes with Debian11 is **5.15.2**

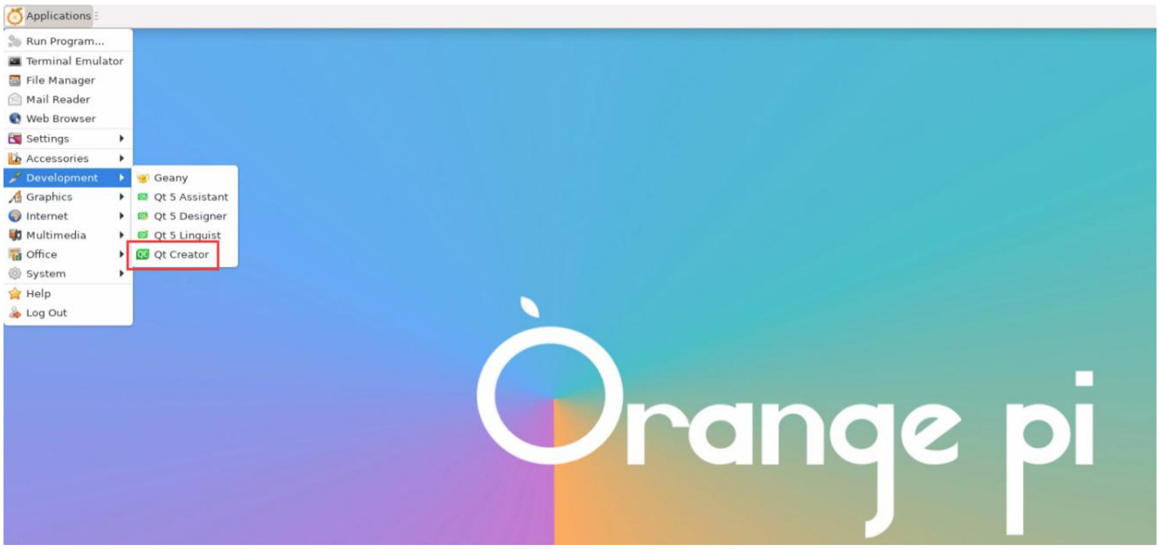
```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

d. The QT version that comes with Debian12 **is 5.15.8**

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

```
......
QMake version 3.1
Using Qt version 5.15.8 in /usr/lib/aarch64-linux-gnu
```

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



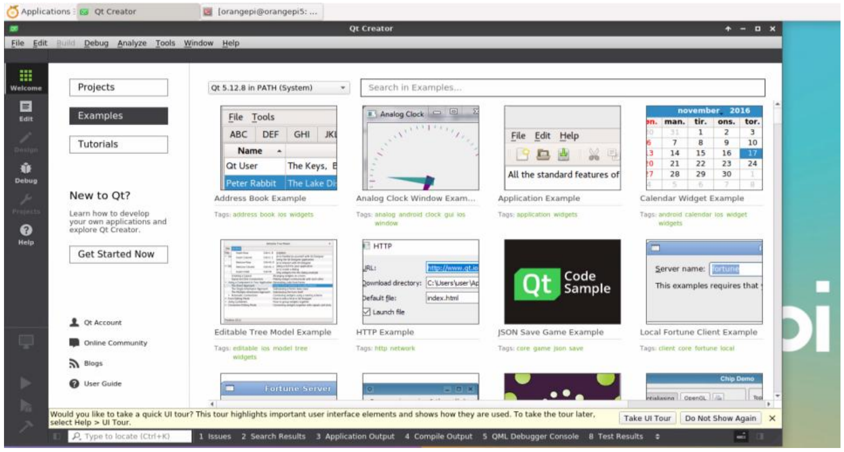
You can also use the following command to open QT Creator

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

During the startup process of QT and QT application, if the following error is prompted, please ignore it directly, this error will not affect the operation of the application.

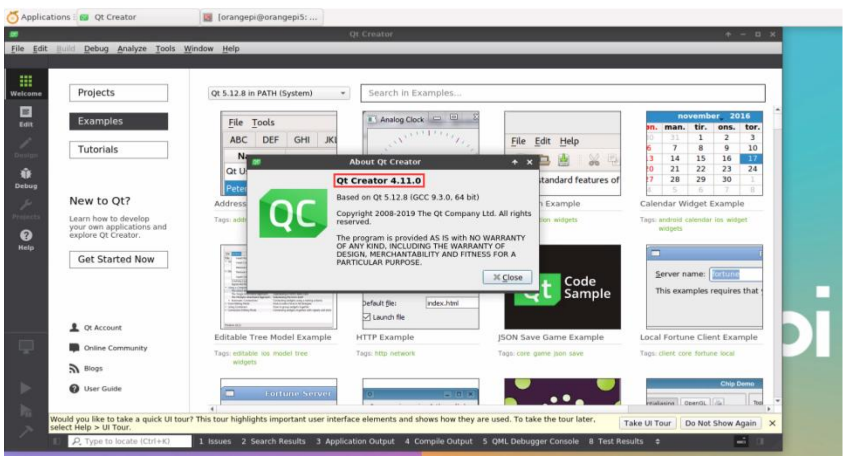
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip

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

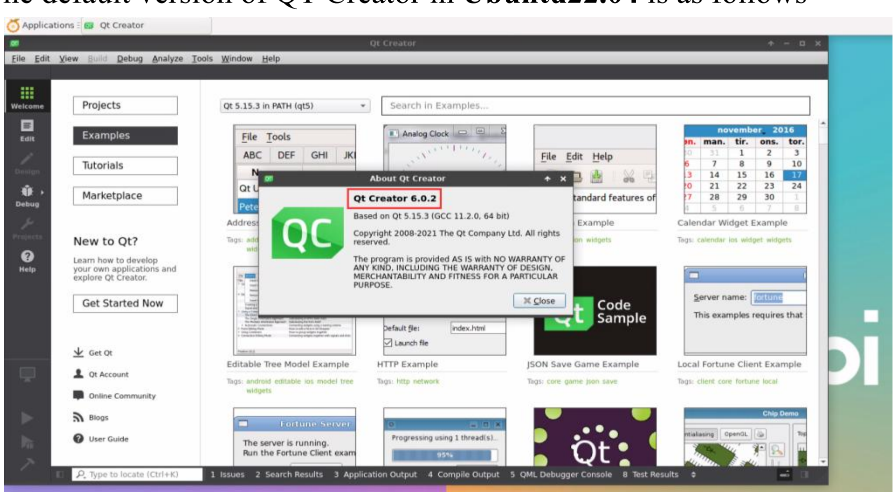


5) The version of QT Creator is as follows

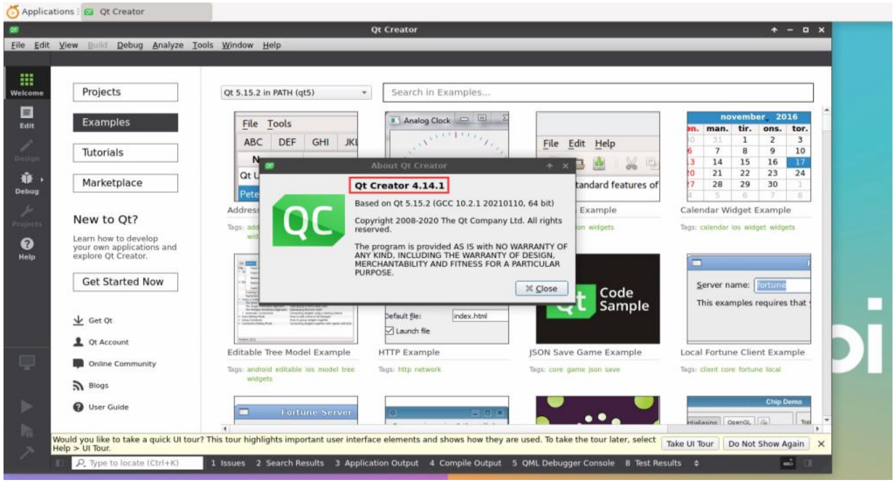
a. The default version of QT Creator in **Ubuntu20.04** is as follows



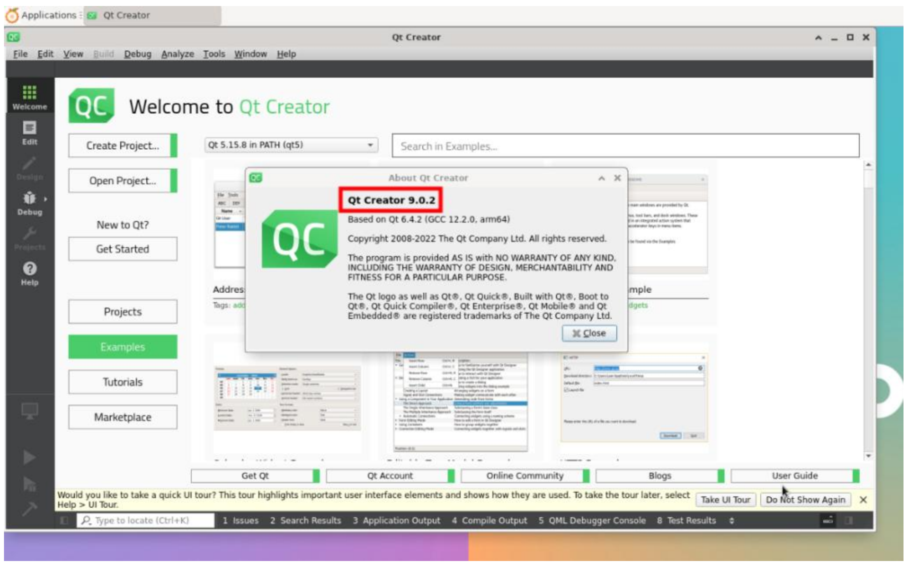
b. The default version of QT Creator in **Ubuntu22.04** is as follows



c. The default version of QT Creator in **Debian11** is as follows

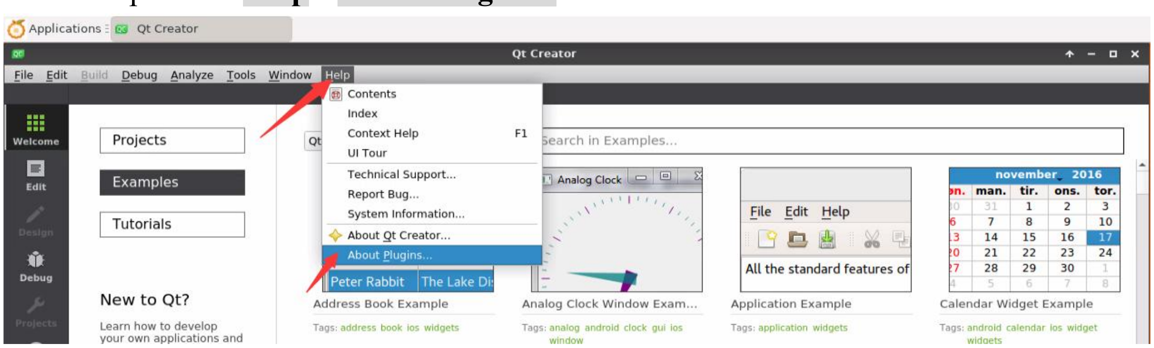


d. The default version of QT Creator in **Debian12** is as follows

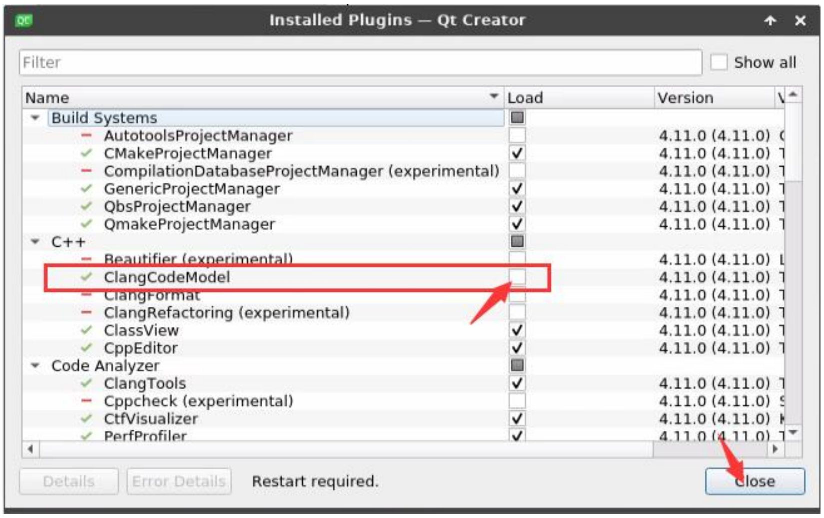


6) Then set the QT

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



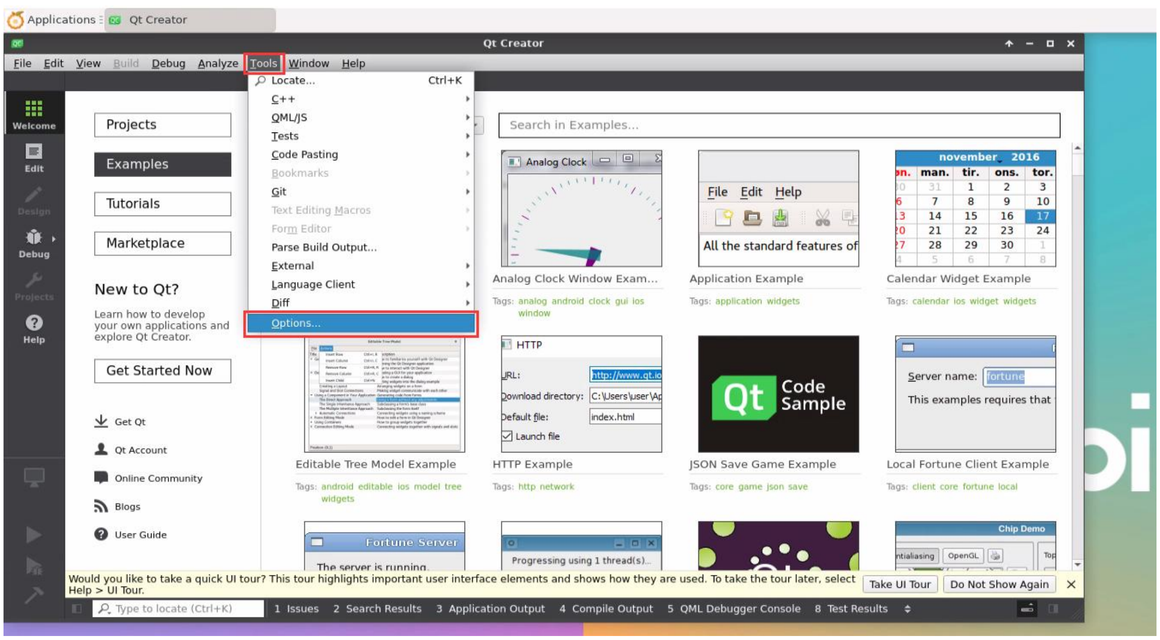
b. Then remove the tick of **ClangCodeModel**

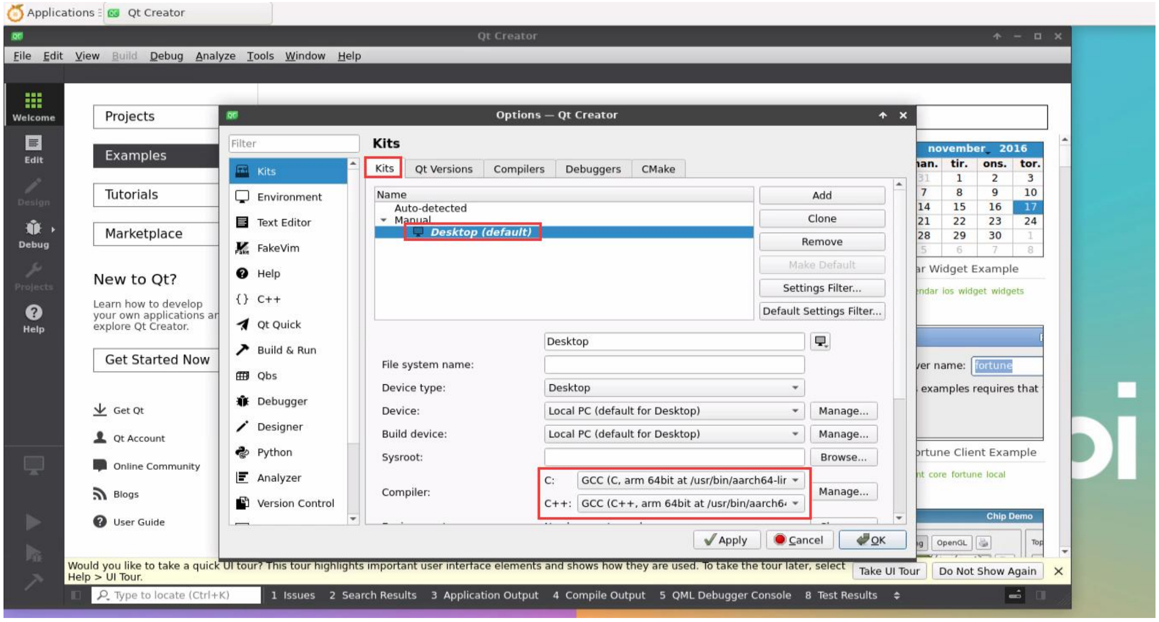


c. **After setting, you need to restart QT Creator**

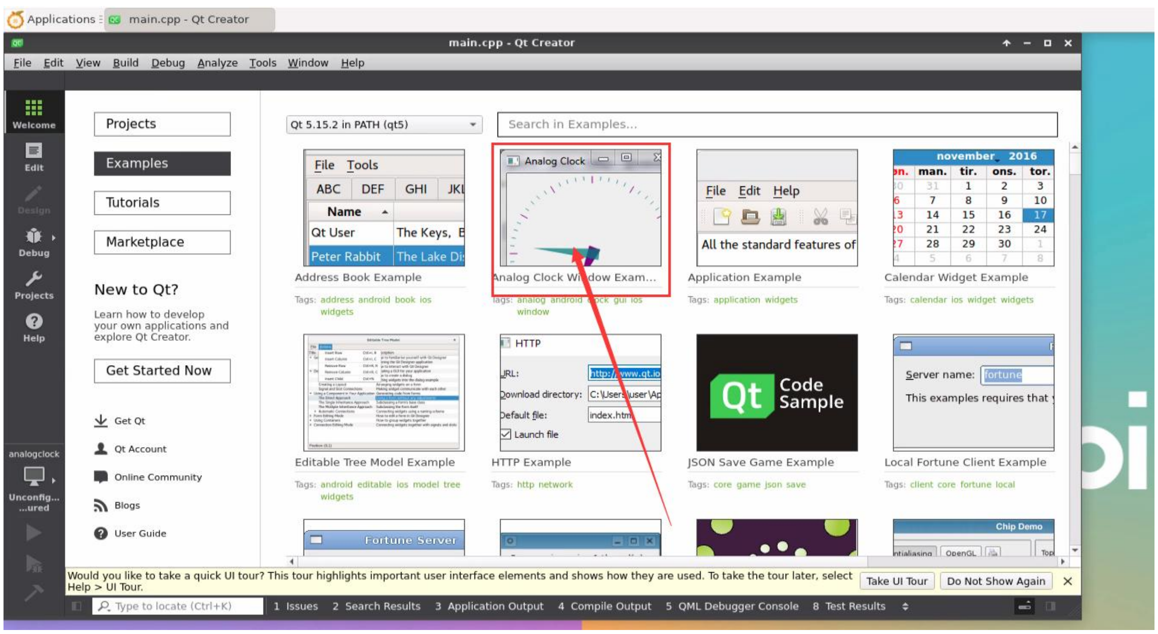
d. Then make sure the GCC compiler used by QT Creator, if the default is Clang, please modify it to GCC

Debian12 please skip this step.

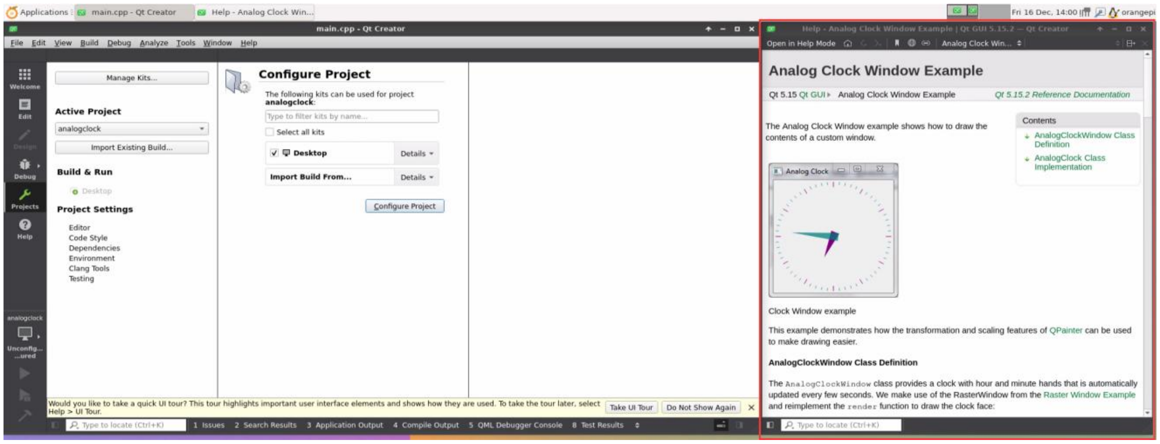




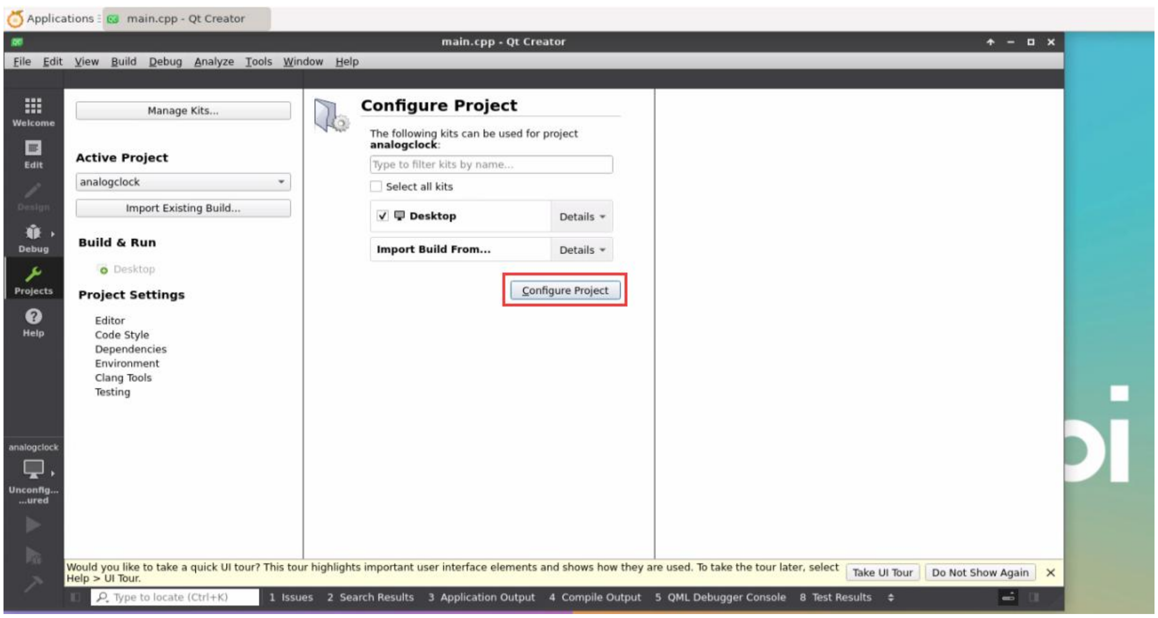
7) Then you can open a sample code



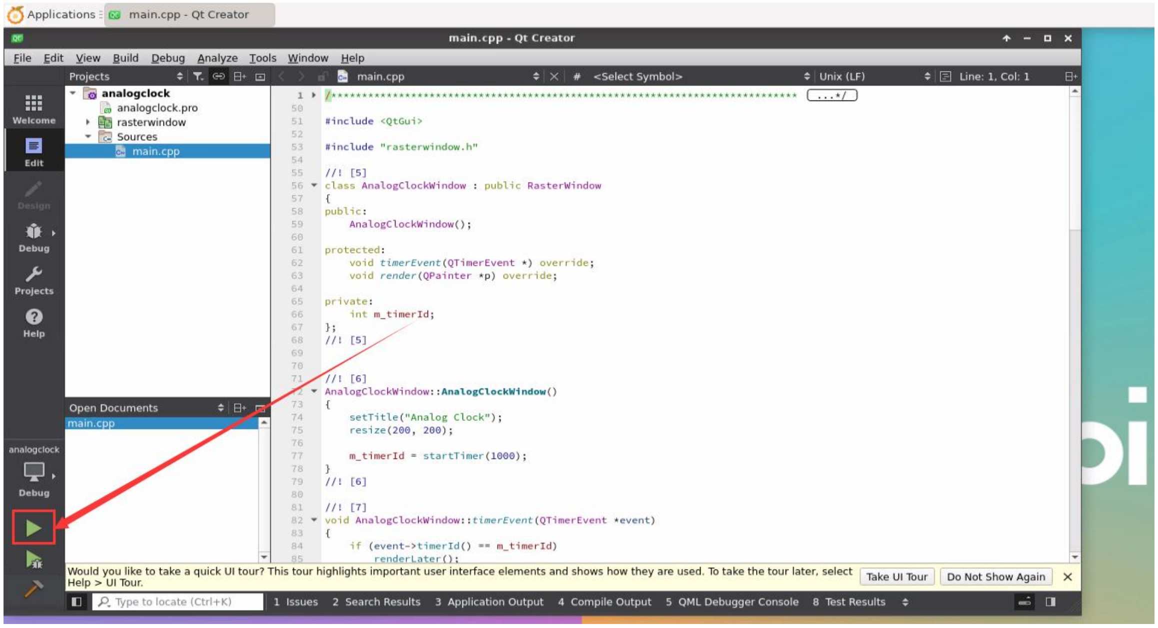
8) After clicking the sample code, the corresponding instruction document will be opened automatically, you can read the instruction carefully



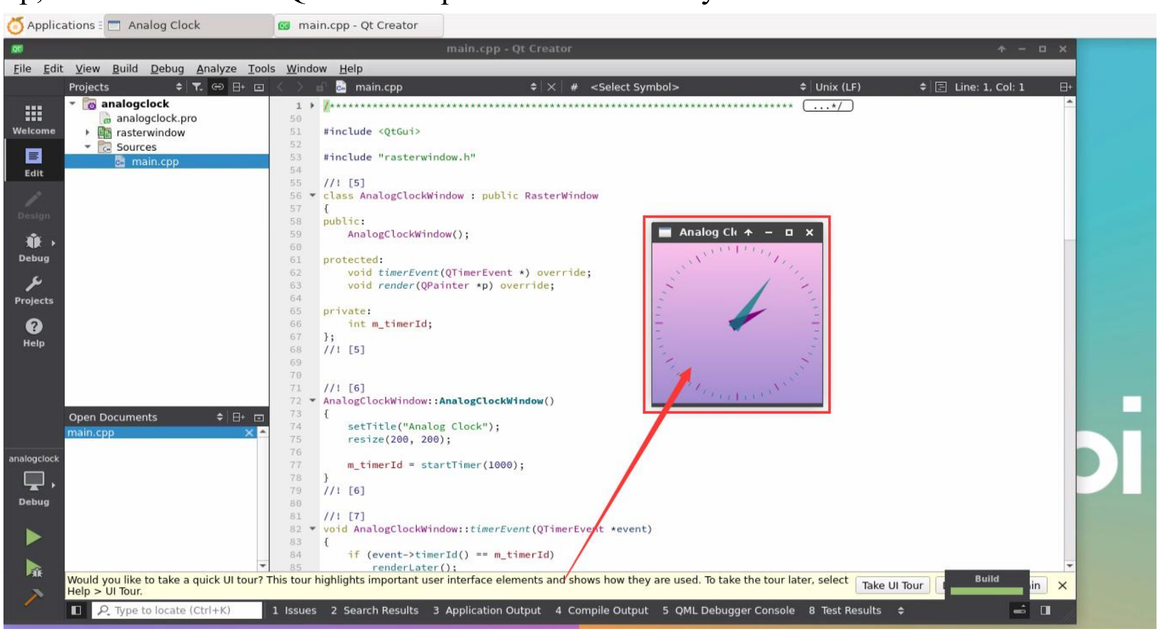
9) Then click next **Configure Project**



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



11) After waiting for a period of time, the interface shown in the figure below will pop up, which means that QT can compile and run normally



12) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu
https://download.qt.io/archive/qtcreator
https://download.qt.io/archive/qt

3. 27. How to install ROS

3. 27. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The current active version of ROS 1 is as follows, and the recommended version is **Noetic Ninjemys**

Active ROS 1 distributions

Recommended

Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

http://docs.ros.org
https://wiki.ros.org/Distributions

2) The official installation document link of ROS 1 **Noetic Ninjemys** is as follows:

http://wiki.ros.org/noetic/Installation/Ubuntu

3) In the official installation document of ROS **Noetic Ninjemys**, Ubuntu recommends using Ubuntu20.04, so please make sure that the system used by the development board is **Ubuntu20.04 desktop system**

http://wiki.ros.org/noetic/Installation

4) Then use the following script to install ros1

```
orangepi@orangepicm4:~$ install_ros.sh ros1
```

5) Before using the ROS tool, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code

Note that running the following command needs to ensure that the development board can access github normally, otherwise an error will be reported due to network problems.

The install_ros.sh script will try to modify /etc/hosts and automatically run the following commands. However, this method cannot guarantee normal access to github every time. If the following error is prompted after installing ros1 in install_ros.sh, please find other ways to allow the Linux system of the development board to access github normally, and then manually run the following Order.

https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
ERROR: error loading sources list:
The read operation timed out

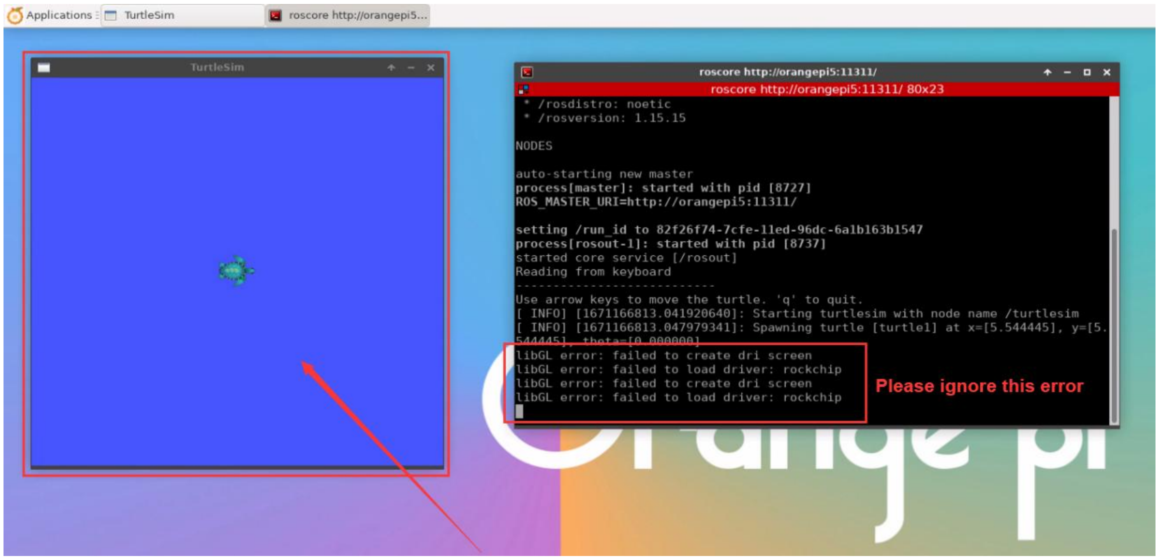
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run
```

```
        rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

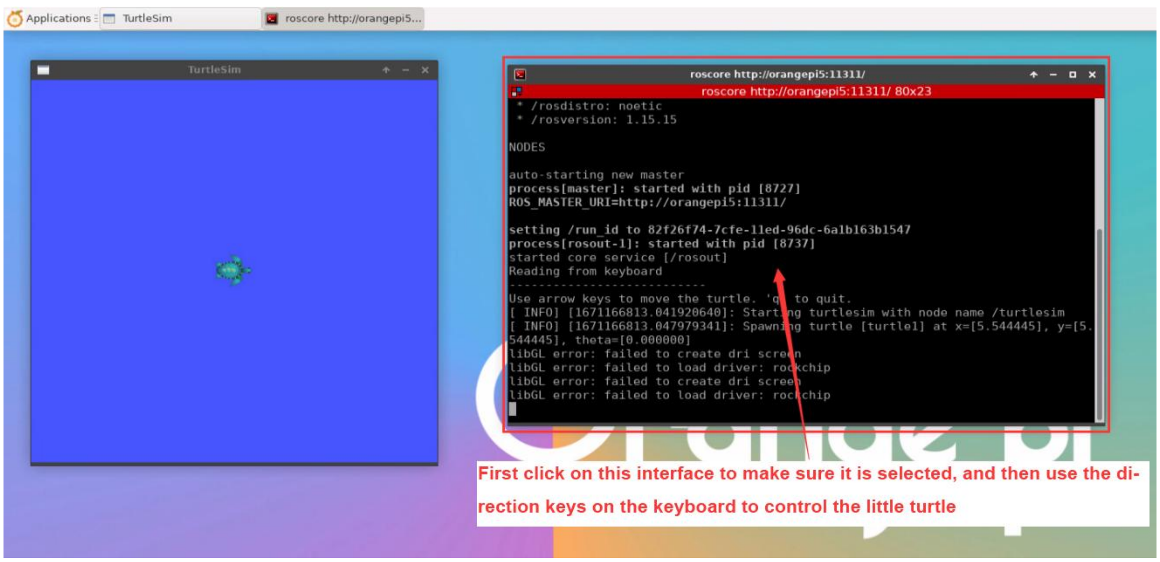
6) Then open a command line terminal window on the desktop, and then use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally

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

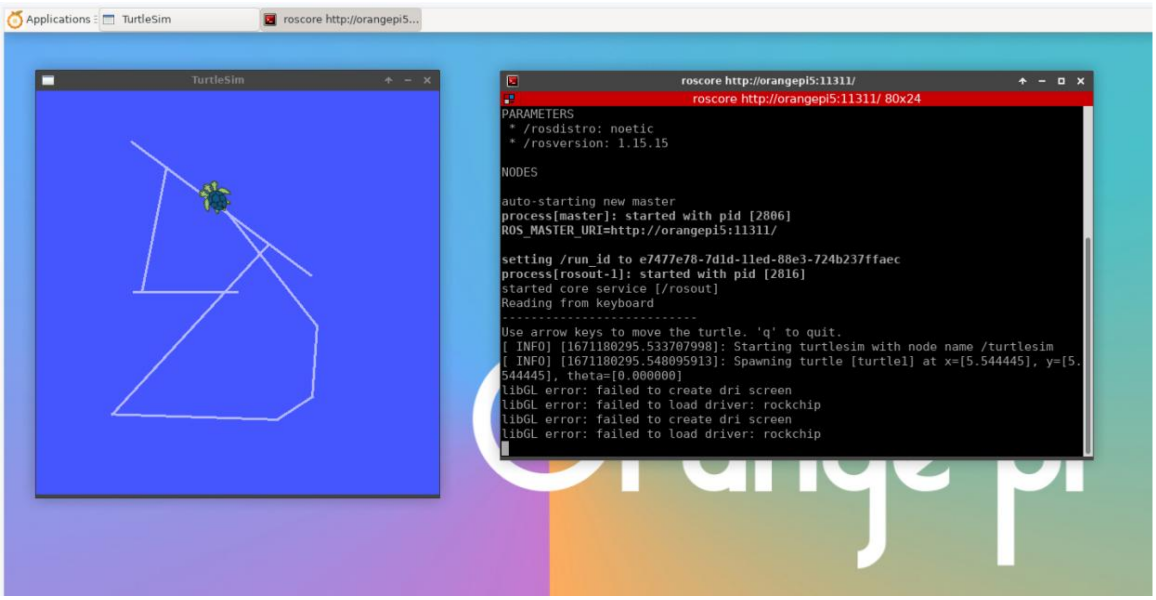
7) After running the **test_ros.sh** script, a little turtle as shown in the figure below will pop up



8) Then please keep the terminal window just opened at the top

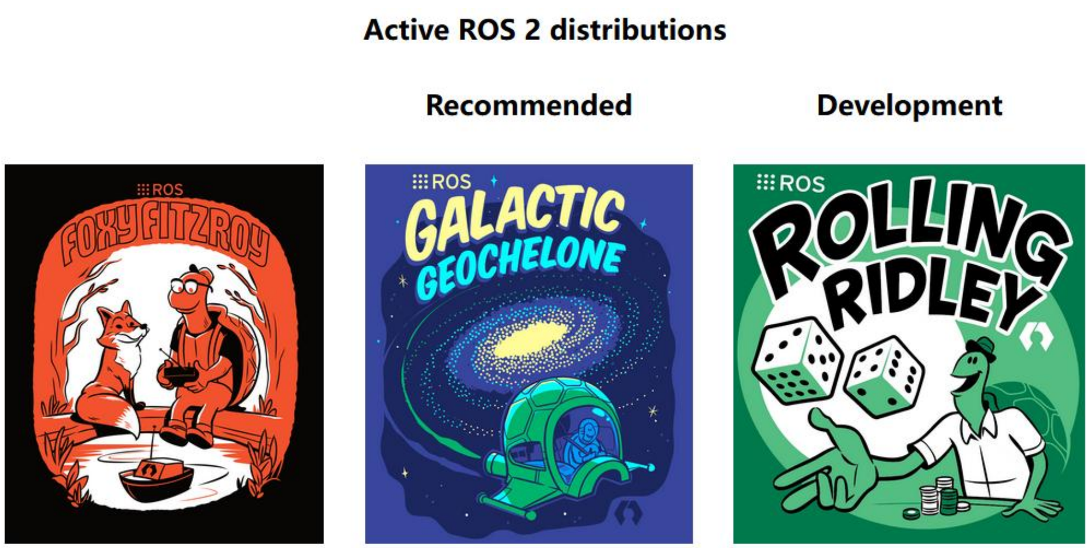


9) At this time, press the direction keys on the keyboard to control the little turtle to move up, down, left, and right



3. 27. 2. How to install ROS 2 Galactic on Ubuntu 20.04

1) The current active version of ROS 2 is as follows, and the recommended version is **Galactic Geochelone**



Distro	Release date	Logo	EOL date
Humble Hawksbill	May 23rd, 2022		May 2027
Galactic Geochelone	May 23rd, 2021		November 2022
Foxy Fitzroy	June 5th, 2020		May 2023

http://docs.ros.org
http://docs.ros.org/en/galactic/Releases.html

2) The link to the official ROS 2 **Galactic Geochelone** installation documentation is as follows:

docs.ros.org/en/galactic/Installation.html
http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html

3) In the official ROS 2 **Galactic Geochelone** installation document, Ubuntu Linux recommends using Ubuntu 20.04, so please make sure that the system used by the development board is the **Ubuntu 20.04 desktop system**. There are several ways to install ROS 2. The following demonstrates how to install it through **Debian packages** ROS 2 **Galactic Geochelone**

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

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

5) The **install_ros.sh** script will automatically run the **ros2 -h**command after installing ros2. If you can see the following print, it means that the ros2 installation is complete

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

ros2 is an extensible command-line tool for ROS 2.
```

```
optional arguments:
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

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

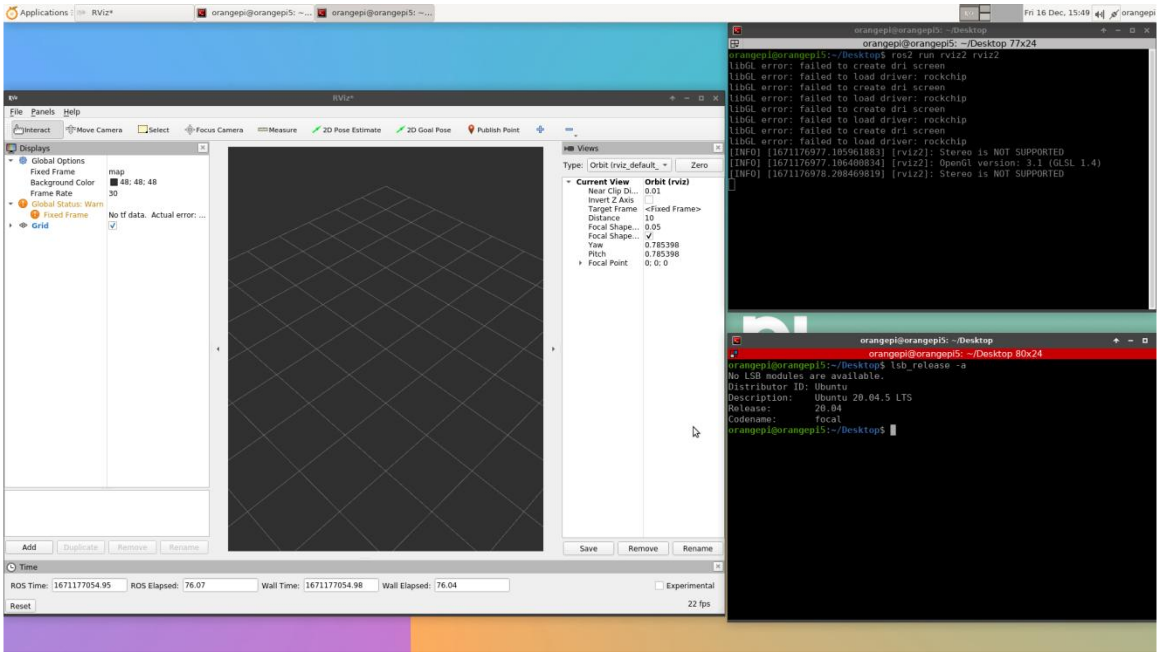
6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepicm4:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

orangepi@orangepi:~$ **source /opt/ros/galactic/setup.bash**

orangepi@orangepi:~$ **ros2 run rviz2 rviz2**



8) For how to use ROS, please refer to the documentation of ROS 2

http://docs.ros.org/en/galactic/Tutorials.html

3. 27. 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) The **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete

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

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action     Various action related sub-commands
  bag         Various rosbag related sub-commands
```

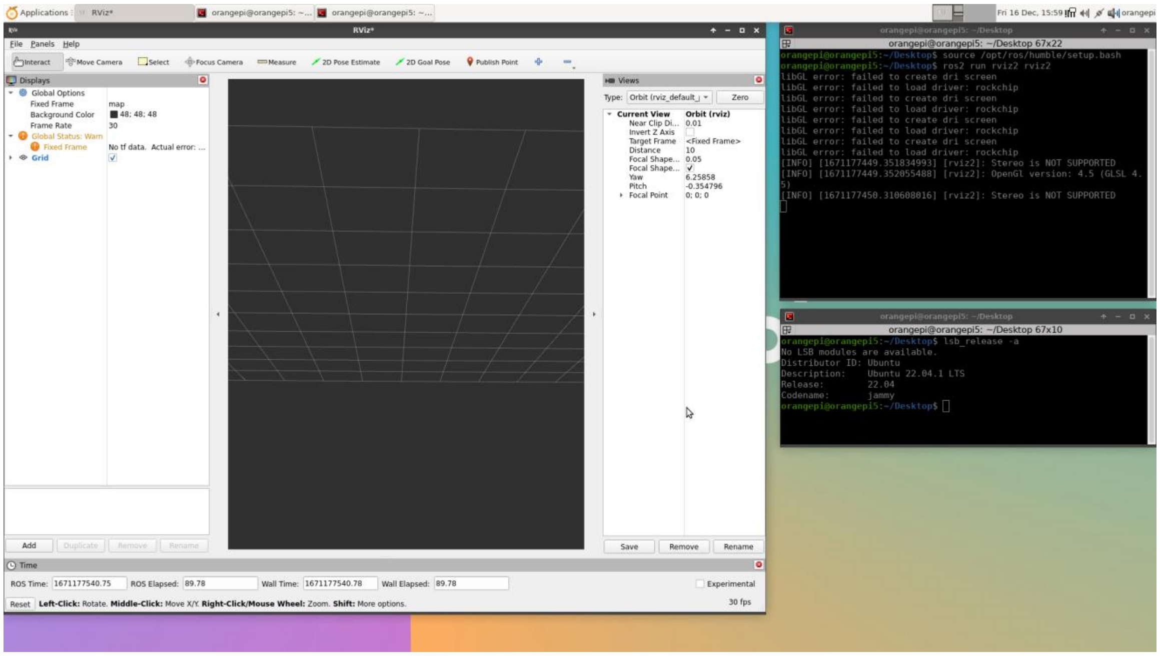
```
component    Various component related sub-commands
daemon       Various daemon related sub-commands
doctor       Check ROS setup and other potential issues
interface    Show information about ROS interfaces
launch       Run a launch file
lifecycle    Various lifecycle related sub-commands
multicast    Various multicast related sub-commands
node         Various node related sub-commands
param        Various param related sub-commands
pkg          Various package related sub-commands
run          Run a package specific executable
security     Various security related sub-commands
service      Various service related sub-commands
topic        Various topic related sub-commands
wtf          Use `wtf` as alias to `doctor`

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

3) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepicm4:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/humble/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



5) Reference documents

http://docs.ros.org/en/humble/index.html

http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

3. 28. How to install kernel header files

1) The Linux image released by OPi comes with the deb package of the kernel header file by default, and the storage location is **/opt/**

```
orangepi@orangepi:~$ ls /opt/linux-headers*
/opt/linux-headers-legacy-rockchip-rk356x_x.x.x_arm64.deb
```

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

The name of the kernel header file deb package needs to be replaced with the actual name, please do not copy it.

```
orangepi@orangepi:~$ sudo dpkg -i /opt/linux-headers-legacy-rockchip-rk356x_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-rk356x
```

4) Then you can write a hello kernel module to test the kernel header file

   a. First write the code of 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");
```

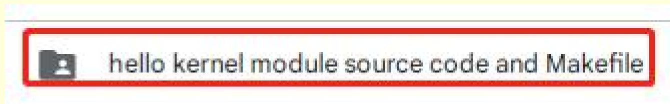
   b. Then write the Makefile for compiling the hello kernel module, as follows:

```
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 a problem with compiling the code you copied here, please go to the official tool to download the source code and upload it to the Linux system of the development board for testing.




```
orangepi@orangepi:~$ make
make -C /lib/modules/5.10.160-rockchip-35xx/build M=/home/orangepi modules
make[1]: Entering directory '/usr/src/linux-headers-5.10.160-rockchip-rk35xx'
  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-rk35xx'
```

d. After compiling, the **hello.ko** kernel module will be generated

```
orangepi@orangepi:~$ ls *.ko
hello.ko
```

e. Use the **insmod**command to insert the **hello.ko** kernel module into the kernel

```
orangepi@orangepi:~$ sudo insmod hello.ko
```

f. Then use the **demsg** command to view the output of the **hello.ko** kernel module. If you can see the output below, 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
```

3.29. Use of Raspberry Pi 5 Inch screen

3.29.1. How to assemble the Raspberry Pi 5 Inch screen

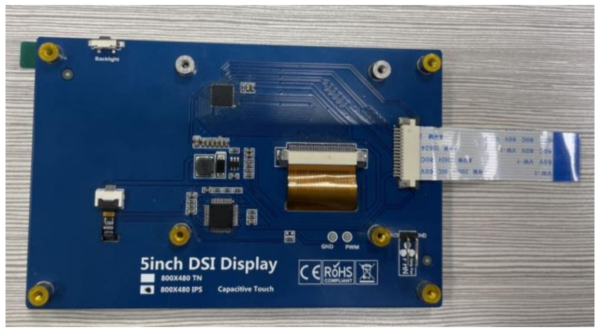
1) First prepare the required accessories

   a. Raspberry Pi 5-inch MIPI LCD display + Touch screen

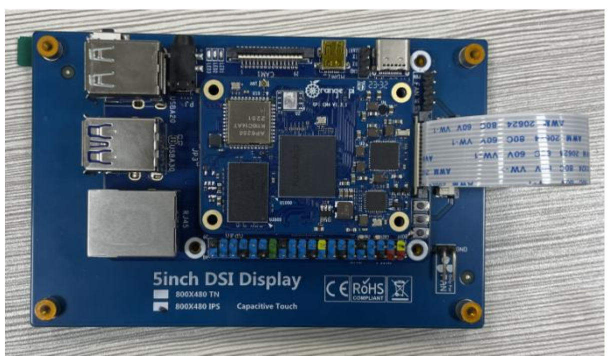
Raspberry Pi 5-inch MIPI LCD display + Touch screenis shown in the figure below

   b. 15pin MIPI cable

2) Then connect the 15pin MIPI cable to the Raspberry Pi 5-inch screen as shown in the figure below (note the orientation of the insulating surface)



3) Finally connect to the LCD interface of the Orange Pi CM4 development board



3. 29. 2. How to open Raspberry Pi 5-inch screen configuration

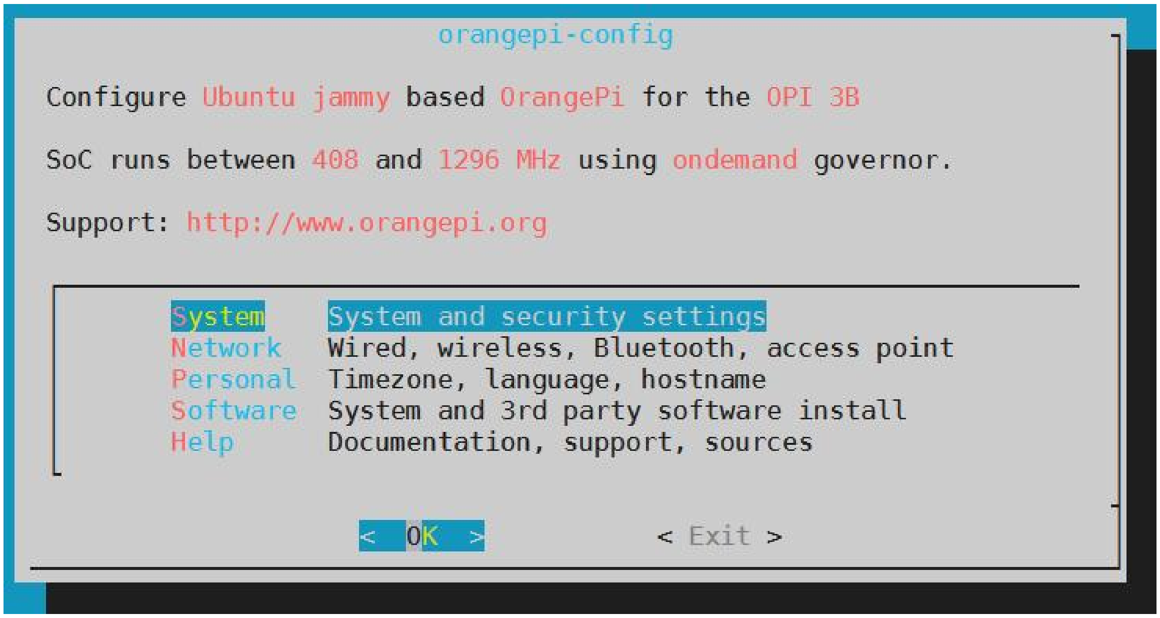
1) By default, the Linux image does not enable the configuration of the Raspberry Pi 5-inch screen. If you need to use the Raspberry Pi 5-inch screen, you need to manually open it.

2) The steps to open the mipi lcd configuration are as follows:

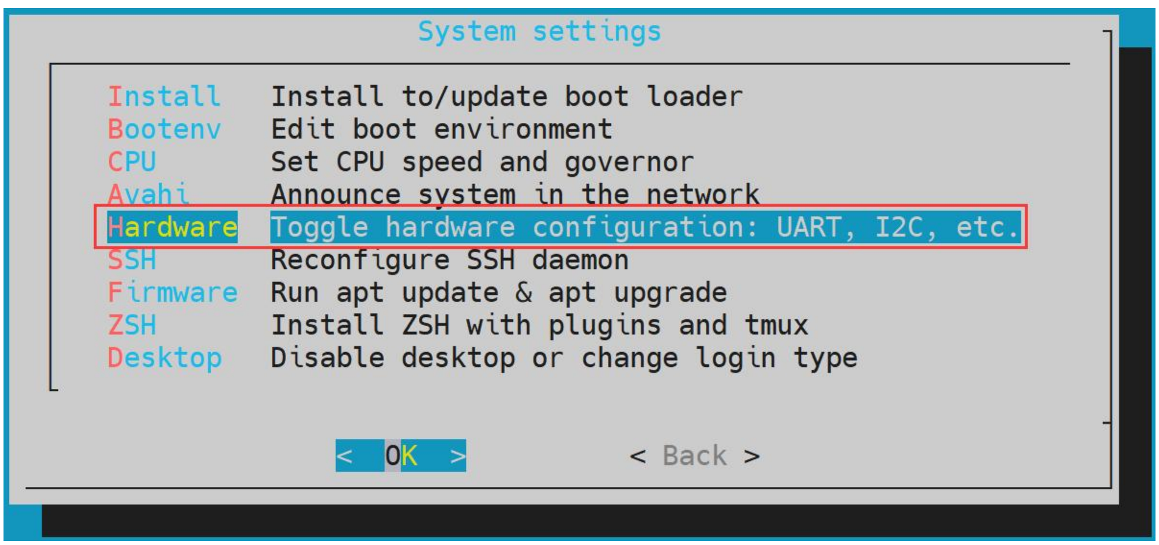
a. First run **orangepi-config**, common users remember to add **sudo** permission

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

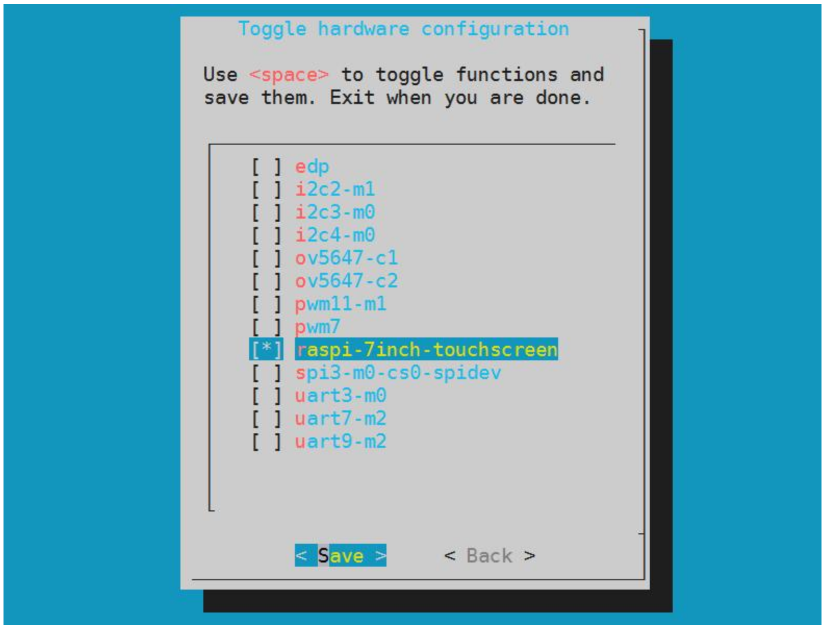
b. Then select **System**



c. Then select **Hardware**

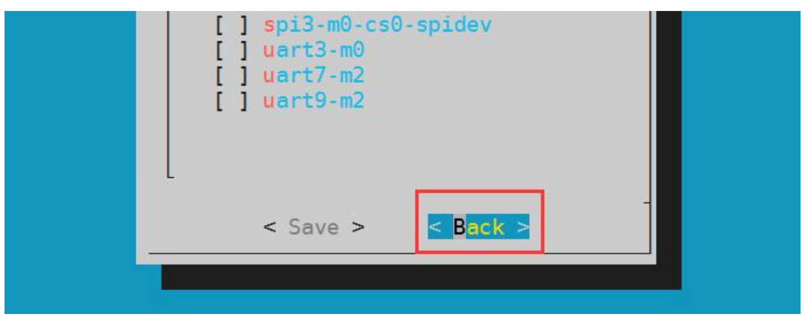


d. Then use the arrow keys on the keyboard to navigate to raspi-7inch-touchscreen, and then use the space to select

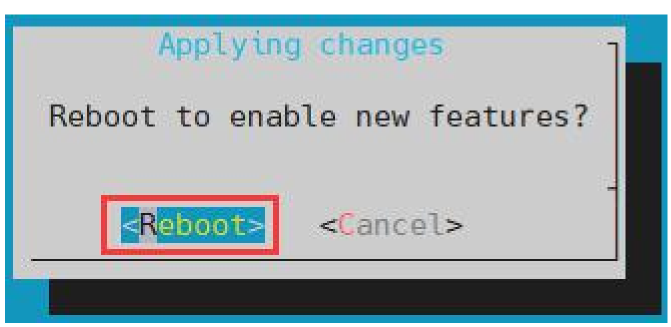


e. Then select **<Save>**Save

f. Then select **<Back>**



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



The above settings will eventually add the configuration of overlays=raspi-7inch-touchscreen to /boot/orangepiEnv.txt. After setting, you can check it first. If this configuration does not exist, then there is a problem with the settings.

If you find it troublesome to use orangepi-config, you can also use the vim editor to open /boot/orangepiEnv.txt, and then add the configuration of overlays=raspi-7inch-touchscreen is also possible.

```
orangepi@orangepi:~$ cat /boot/orangepiEnv.txt | grep "raspi"
overlays=raspi-7inch-touchscreen          # sample configuration
```

3) After startup, you can see the display of the LCD screen as follows:

3. 29. 3. The method of server version image rotation display direction

1) Add **extraargs=fbcon=rotate:the direction to rotate** in **/boot/orangepiEnv.txt** This line configuration can set the direction displayed by the server version of the Linux system, where the number after **fbcon=rotate:** can be set as:

a. 0: normal screen (default is landscape)

b. 1: Turn clockwise 90 degrees

c. 2: Flip 180 degrees

d. 3: Turn clockwise 270 degrees

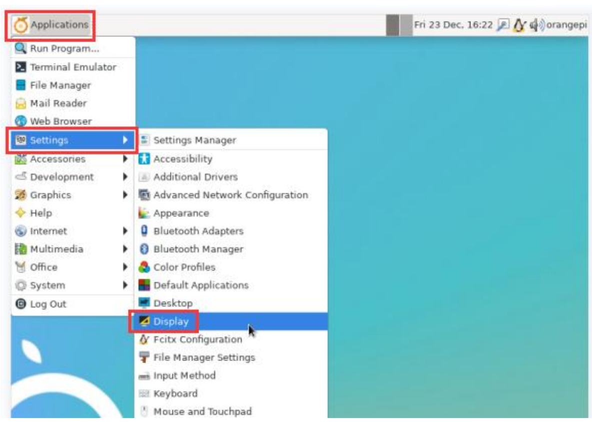
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=lcd1
extraargs=cma=64M fbcon=rotate:3
```

Note that if there is the line extraargs=cma=64M in /boot/orangepiEnv.txt by default, the configuration fbcon=rotate:3 can be added after extraargs=cma=64M (separated by spaces).

2) Then **restart** the Linux system and you can see that the direction displayed on the LCD screen has been rotated

3. 29. 4. Method of rotating display and touch direction of desktop version image

1) First open **Display** Settings in Linux



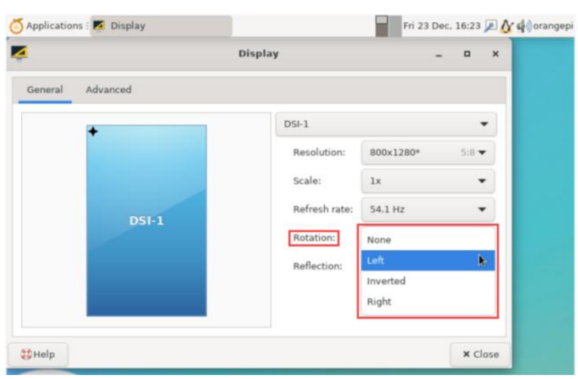
2) Then select the direction you want to rotate in **Rotation**

a. **None**: no rotation

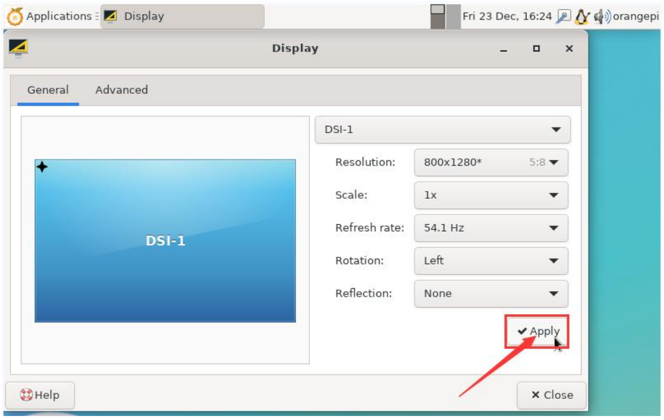
b. **Left**: rotate left 90 degrees

c. **Inverted**: Flip up and down, which is equivalent to rotating 180 degrees

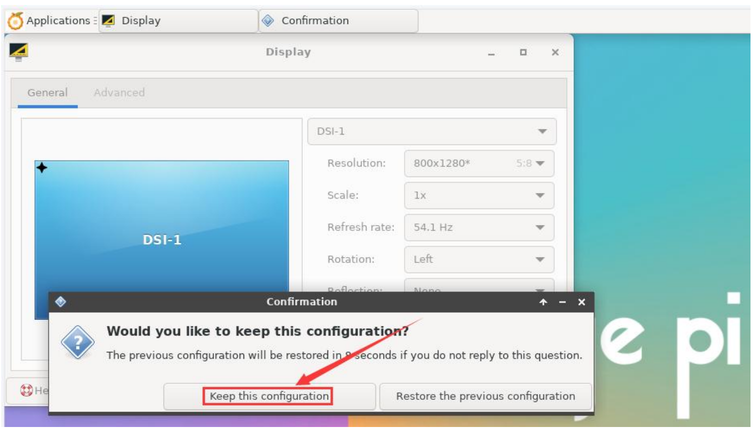
d. **Right**: rotate right 90 degrees



3) Then click **Apply**



4) Then select **Keep this configuration**



5) At this point, the screen display has been rotated, and then close the **Display** program

6) The above steps will only select the display direction, and will not rotate the direction of the touch. Use the **set_lcd_rotate.sh** script to rotate the direction of the touch. After the script is set, it will automatically restart, and then you can test whether the touch has been used normally.

a. **None**: no rotation

```
orangepi@orangepi:~$ set_lcd_rotate.sh none
```

b. **Left**: rotate left 90 degrees

```
orangepi@orangepi:~$ set_lcd_rotate.sh left
```

c. **Inverted**: Flip up and down, which is 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
```

The set_lcd_rotate.sh script mainly does four things:

1. Rotate the direction displayed by the framebuffer
2. Rotate the direction of the touch
3. Turn off the boot logo
4. Restart the system

Rotating the touch direction is achieved by adding the line 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" is configured differently for different directions.

7) Touch rotation reference

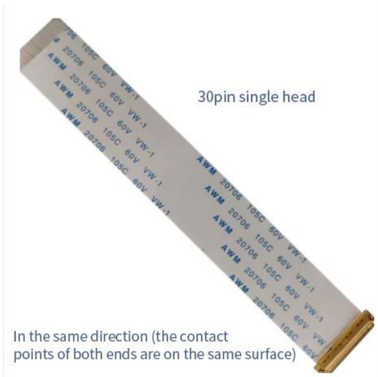
https://wiki.ubuntu.com/X/InputCoordinateTransformation

3. 30. How to use the eDP screen

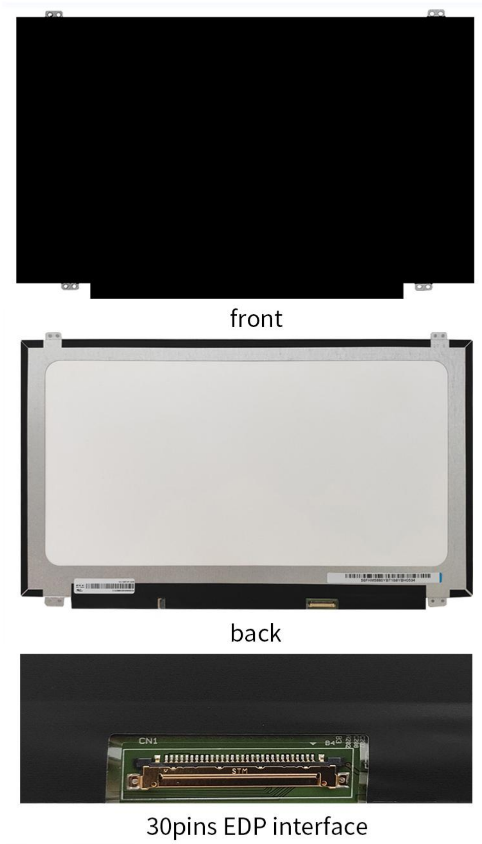
3. 30. 1. Assembly method of eDP screen

1) Currently only one eDP screen is compatible, including the following accessories:

a. 0.5 pitch 30pin single-head cable in the same direction



b. eDP display



2) Connect the 30pin eDP interface of the screen and the eDP interface of the development board with a 30pin single-head cable in the same direction

3. 30. 2. How to open the eDP screen configuration

Note that the method described below is only applicable to the adapted eDP screen. If the customer uses an unadapted screen, it cannot be turned on according to the following method.

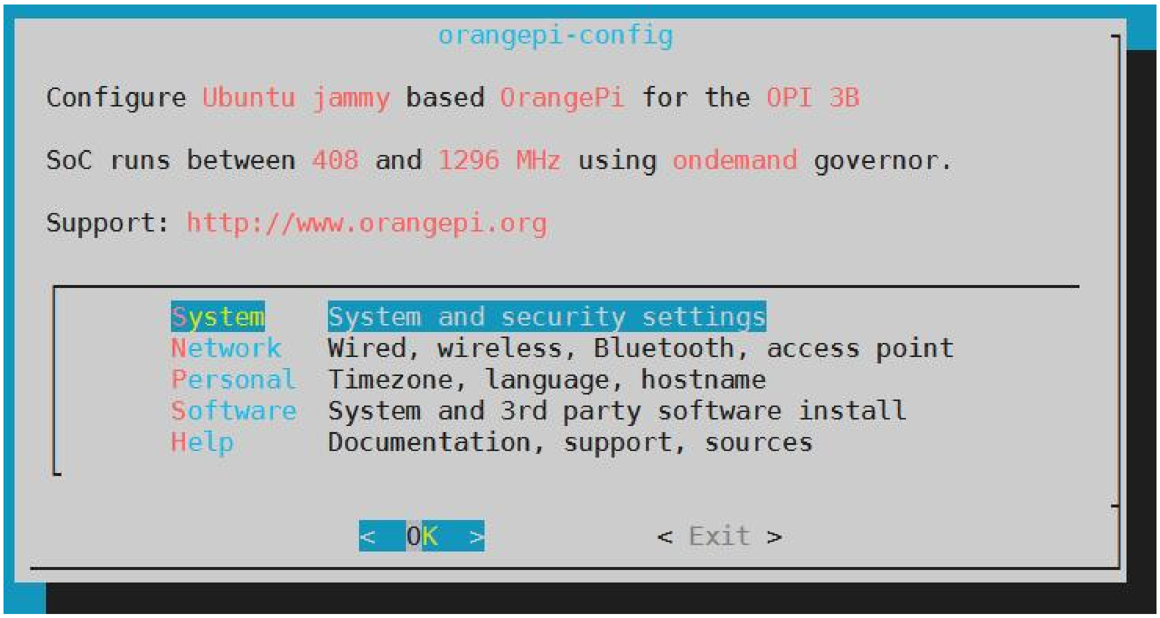
1) By default, the Linux image does not have the configuration to open the eDP screen. If you need to use the eDP screen, you need to open it manually.

2) The steps to open the eDP screen configuration are as follows:

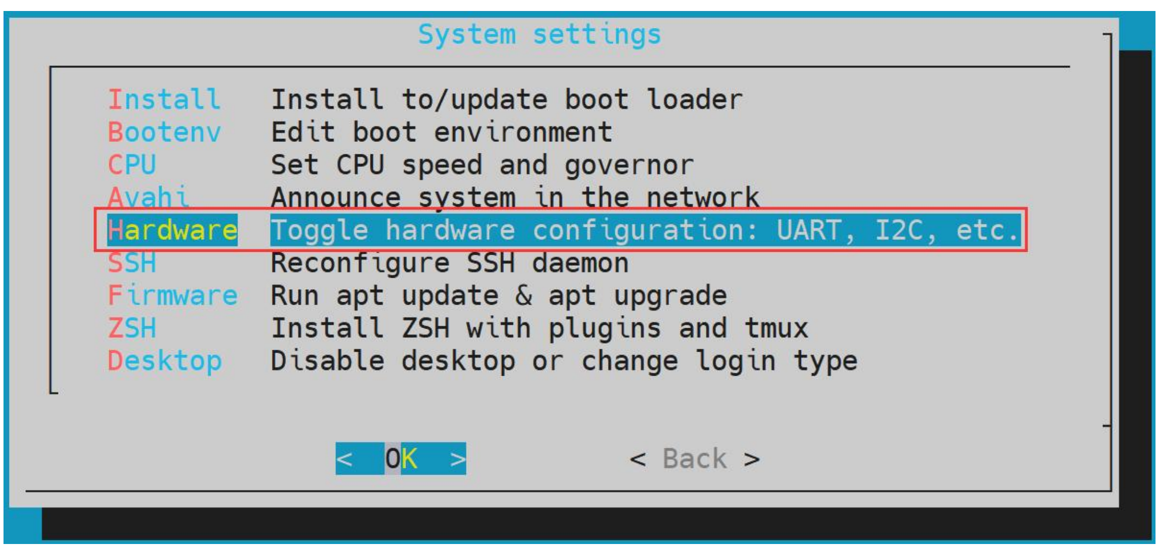
a. First run **orangepi-config**, ordinary users remember to add **sudo** permission

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

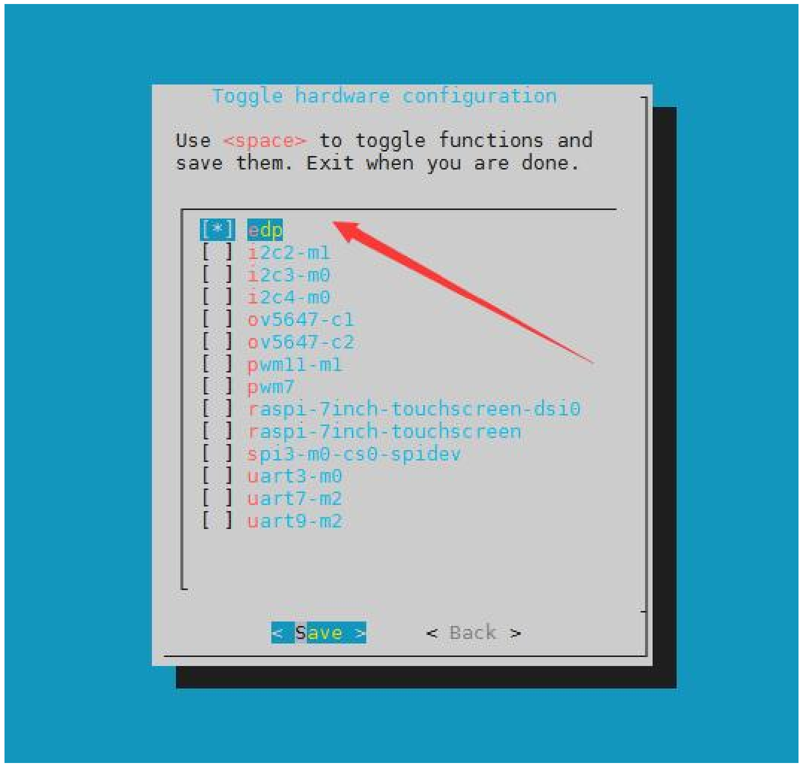
b. Then select **System**



c. Then select **Hardware**

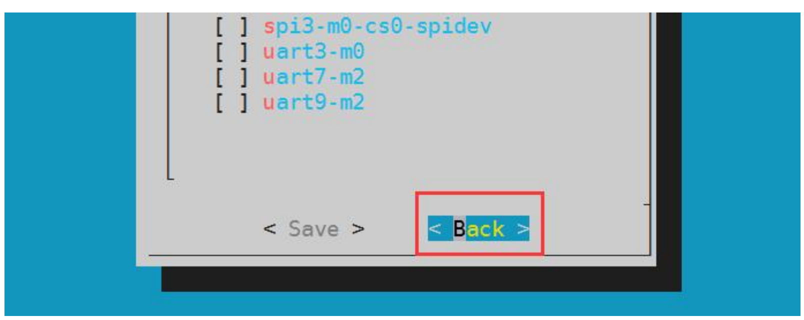


d. Then use the arrow keys on the keyboard to navigate to edp, and then use the **space** to select

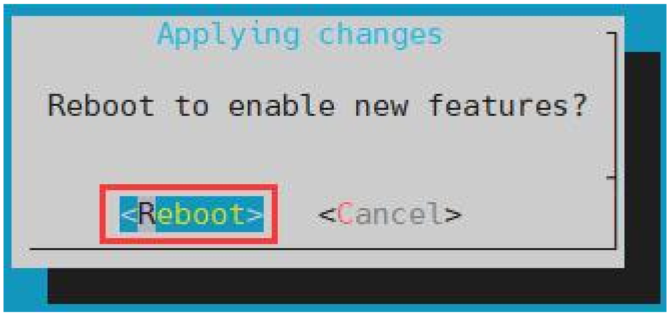


e. Then select **<Save>** to save

f. Then select **<Back>**



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



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

If you find it troublesome to use orangepi-config, you can also use the vim editor to open /boot/orangepiEnv.txt, and then add the configuration of overlays=edp.

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

3) After startup, you can see the display of the eDP screen as follows:

3. 31. Instructions for using the switch logo

1) By default, the switch logo will only be displayed in the desktop version of the system

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the switch logo

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=false
```

3) Set the **bootlogo** variable to **true** in **/boot/orangepiEnv.txt** to enable the switch logo

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
```

4) The location of the boot logo image in the Linux system is

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

5) After replacing the boot logo picture, you need to run the following command to take effect

```
orangepi@orangepi:~$ sudo update-initramfs -u
```

3. 32. How to use the ZFS file system

3. 32. 1. How to install ZFS

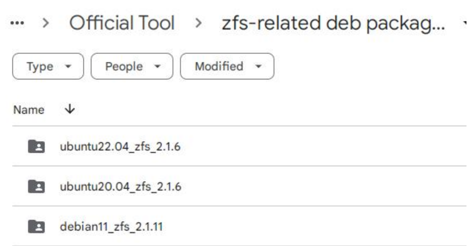
Before installing zfs, please make sure that the Linux image used is the latest version. In addition, if zfs is already installed in the system, it needs to be installed again.

Before installing zfs, you need to install the kernel header file first. For the method of installing the kernel header file, please refer to the instructions in the **section on the method of installing the kernel header file**.

In Ubuntu20.04, Ubuntu22.04 and Debian11 systems, zfs cannot be installed directly through apt, because the default apt source zfs version is lower than 2.1.6, and there is a problem of incompatibility with rk Linux5.10 kernel. This problem is fixed in zfs version

2.1.6 and later.

To solve this problem, we provide a zfs deb package that can be installed normally, which can be downloaded from the official tool of the development board. Open the **official tool**, and enter the **zfs-related deb package folders 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 zfs deb packages of the corresponding version, please upload them to the Linux system of the development board. For the upload method, please refer to **the description in the section of the method of uploading files to the Linux system of the development board.**

After the upload is complete, use the **cd** command in the command line of the development board Linux system to enter the directory of the deb package, and then use the following command to install the deb package of zfs.

```
orangepi@orangepi:~$ sudo apt install ./*.deb
```

After the installation is complete, use the following command to see the zfs-related kernel modules:

```
orangepi@orangepi:~$ ls /lib/modules/5.10.160-rockchip-rk356x/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 to see that the zfs kernel module will be automatically loaded:

```
orangepi@orangepi:~$ lsmod | grep "zfs"
zfs                     2801664   0
zunicode                 327680   1 zfs
zzstd                    471040   1 zfs
zlua                     139264   1 zfs
zcommon                   69632   1 zfs
```

znvpair	61440	2 zfs,zcommon
zavl	16384	1 zfs
icp	221184	1 zfs
spl	77824	6 zfs,icp,zzstd,znvpair,zcommon,zavl

In Debian12, the default version of zfs is 2.1.11, so we can install zfs directly through the following command. Again, 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. 32. 2. Methods of creating ZFS pools

ZFS is based on storage pools, we can add multiple physical storage devices to the pool, and then allocate storage space from this pool.

The following content is demonstrated based on the development board 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 U disk. The output is as follows:

```
orangepi@orangepi:~$ lsblk
NAME         MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda            8:0    1  28.8G  0 disk
├─sda1         8:1    1  28.8G  0 part
└─sda9         8:9    1    8M   0 part
mtdblock0     31:0    0    16M  0 disk
mmcblk0      179:0    0  29.7G  0 disk
├─mmcblk0p1  179:1    0     1G  0 part /boot
└─mmcblk0p2  179:2    0  28.4G  0 part /var/log.hdd
                                         /
zram0        254:0    0   7.7G  0 disk [SWAP]
zram1        254:1    0   200M  0 disk /var/log
nvme0n1      259:0    0 476.9G  0 disk
├─nvme0n1p1  259:3    0 476.9G  0 part
└─nvme0n1p9  259:4    0     8M  0 part
orangepi@orangepi:~$
```

2) Then enter the following command to create a ZFS pool, including two storage devices, NVMe SSD and U disk

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 U disk

```
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. 32. 3. Test the data deduplication function of ZFS

1) The data deduplication function of ZFS 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 random file with a size of 1G

```
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 random files 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 are currently 1002G of data in the pool, but in fact the size of the ZFS pool is only **504GB** (the total capacity of SSD+U disk), which cannot hold such 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 all duplicates, indicating that the data deduplication function is effective.

```
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  1.01G   503G        -         -     0%     0%  6.00x    ONLINE  -
```

3. 32. 4. Test the data compression function of ZFS

1) Because the stored data is different, the disk space saved by compression will also be different, so we choose to compress relatively large plain text files for compression testing, and execute the following commands to pack the **/var/log/** and **/etc/** directories into a tarball

```
orangepi@orangepi:~$ cd /pool1/
root@orangepi:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

2) Then the file size that can be seen through the **ls -lh** command and the space occupied in the ZFS pool are both **27M**

```
orangepi@orangepi:/pool1$ ls -lh
total 27M
-rw-r--r-- 1 root root 27M Jun  1 14:46 text.tar
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  26.7M   504G        -         -     0%     0%  1.00x    ONLINE  -
orangepi@orangepi:/pool1$
```

3) Then we enable compression in the ZFS pool pool1

```
root@orangepi:/pool1$ sudo zfs set compression=lz4 pool1
```

4) Then execute the following command again to package the **/var/log/** and **/etc/**

directories into a tar package

```
root@orangepi:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

5) At this time, you can see that the size of the **text.tar** file is still 27M, but it only occupies 9.47M in the ZFS pool, indicating that the file is compressed

```
orangepi@orangepi:/pool1$ ls -lh
total 9.2M
-rw-r--r-- 1 root root 27M Jun  1 14:54 text.tar
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  9.47M   504G        -         -     0%     0%  1.00x    ONLINE  -
```

3. 33. How to shut down and restart the development board

1) During the running of the Linux system, if the Type-C power supply is directly unplugged, some data may be lost or damaged in the file system, so please use the **poweroff** command to shut down the Linux system of the development board before powering off. Then unplug the power again.

```
orangepi@orangepi:~$ sudo poweroff
```

2) The command to restart the Linux system is

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

4. Linux SDK——orangepi-build instructions

4.1. Compilation system requirements

We can cross-compile the Linux image of the development board on the x64 computer, or compile the Linux image of the development board on 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 cooling (especially when the SSD starts). If the heat dissipation is not done well, it is prone to the error of file system runaway.

4.1.1. Compile with the Ubuntu22.04 system of the development board

1) The Linux SDK, namely **orangepi-build**, supports running on the **Ubuntu 22.04** of the development board (other systems have not been tested), so before downloading orangepi-build, please first ensure that the Ubuntu version installed on the development board is Ubuntu 22.04. The command to check the Ubuntu version installed on the development board is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please replace the system before performing the following operations.

```
test@test:~$ lsb_release -a
```

```
No LSB modules are available.
Distributor ID:     Ubuntu
Description:        Ubuntu 22.04 LTS
Release:            22.04
Codename:           jammy
```

2) If the computer is installed with Windows system and there is no computer with Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMwar** to install an Ubuntu 22.04 virtual machine in the Windows system. But please be careful not to compile orangepi-build on the WSL virtual machine, because orangepi-build has not been tested in the WSL virtual machine, so it cannot be guaranteed that orangepi-build can be used normally in WSL.

3) The download address of the installation image of Ubuntu 22.04 **amd64** version is:

https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso
or
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.1-desktop-amd64.iso

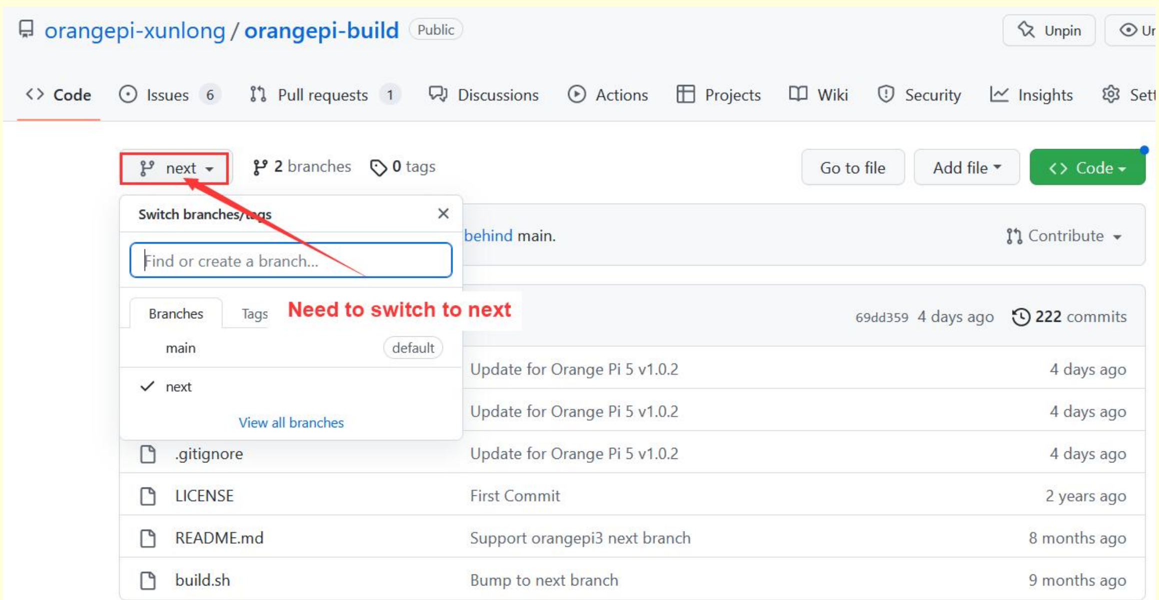
4. 2. Get the source code of Linux sdk

4. 2. 1. Download orangepi-build from github

1) The Linux sdk actually refers to the code of orangepi-build. orangepi-build is modified based on the armbian build system. Using orangepi-build, multiple versions of Linux images can be compiled. First download the code of orangepi-build, 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 3B development board needs to download the source code of the next branch of orangepi-build. The above git clone command needs to specify the branch of the orangepi-build source code as next.



Downloading the orangepi-build code through the git clone command does not require entering the user name and password of the github account (the same is true for downloading other codes in this manual), if the Ubuntu PC prompts the user to enter the github account after entering the git clone command The name and password are usually entered incorrectly in the address of the orangepi-build warehouse behind the git clone. Please check the spelling of the command carefully, instead of thinking that we forgot to provide the username and password of the github account.

2) The u-boot and Linux kernel versions currently used by the development board are as follows

branch	u-boot version	Linux Kernel version
legacy	u-boot 2017.09	Linux5.10

The branch mentioned here is not the same thing as the branch of the orangepi-build source code, please do not confuse it. 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) orangepi-build will contain the following files and folders after downloading

a. **build.sh**: Compile the startup script

b. **external**: Contains the configuration files needed to compile the image, specific scripts, and the source code of some programs, etc.

c. **LICENSE**: GPL 2 license file

d. README.md: orangepi-build documentation

e. **scripts**: General script for compiling Linux images

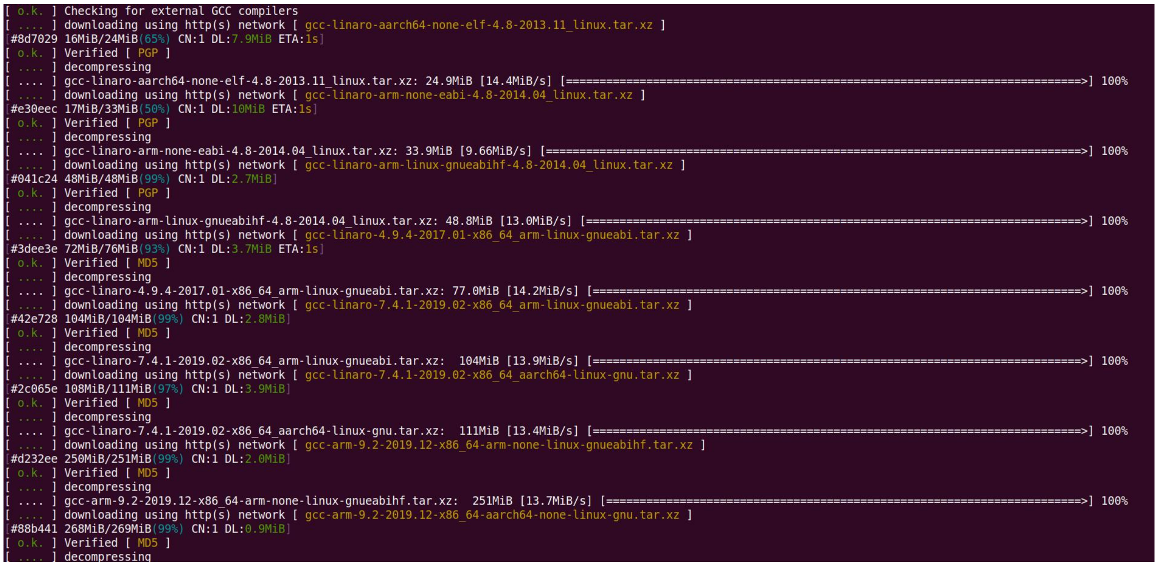
```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

If you downloaded the code of orangepi-build from github, after downloading, you may find that orangepi-build does not contain the source code of u-boot and Linux kernel, nor does u-boot and Linux kernel need to use cross-compilation tools Chain, this is normal, because these things are stored in other separate github warehouses or some servers (the addresses will be detailed below). orangepi-build will specify the address of u-boot, Linux kernel and cross-compilation toolchain in the script and configuration file. When running orangepi-build, when it finds that there are no such things locally, it will automatically go to the corresponding place to download them.

4.2.2. Download the cross-compilation toolchain

The cross-compilation toolchain will only be downloaded when the orangepi-build compilation image is used on an x64 computer. Compiling the Linux image of the development board in the Ubuntu22.04 of the development board will not download the cross-compilation toolchain. At this time, orangepi-build/toolchains will be an empty folder.

1) When orangepi-build runs for the first time, it will automatically download the cross-compilation toolchain and put it in the **toolchains** folder. Every time after running the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists , if it does not exist, the download will be restarted, if it exists, it will be used directly, and the download will not be repeated.



2) The image URL of the cross-compilation toolchain in China is the open source software image site of Tsinghua University

https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/

3) After **toolchains** is downloaded, it will contain multiple versions of cross-compilation toolchains, and the development board will only use two of them

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
gcc-arm-11.2-2022.02-x86_64-arm-none-linux-gnueabihf
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

4) The cross-compilation toolchain 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
```

4. 2. 3. orangepi-build complete directory structure description

1) The orangepi-build repository does not contain the source code of the Linux kernel, u-boot, and cross-compilation toolchain after downloading. The source code of the Linux kernel and u-boot is stored in an independent git repository

a. The git 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 warehouse where the b.u-boot source code is stored is as follows:

```
https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2017.09-rk3588
```

2) When orangepi-build runs for the first time, it will download the cross-compilation toolchain, u-boot and Linux kernel source code. After successfully compiling a Linux image, the files and folders that can be seen in orangepi-build are:

a. **build.sh**: compile startup script

b. **external**: Contains the configuration files needed to compile the image, scripts with specific functions, and the source code of some programs. The rootfs compressed package cached during the image compilation process is also stored in external

c. **kernel**: stores the source code of the Linux kernel, and the folder named orange-pi-5.10-rk35xx stores the kernel source code of the legacy branch of the RK3588/RK3588S/RK3566 series development boards. Please do not manually name the folder name of the kernel source code Modify, if modified, the kernel source code will be re-downloaded when the compilation system is running

d. **LICENSE**: GPL 2 license file

e. **README.md**: orangepi-build documentation

f. **output**: Store compiled deb packages such as u-boot and Linux, compilation logs, and compiled images and other files

g. **scripts**: general scripts for compiling Linux images

h. **toolchains**: store cross-compilation toolchain

i. **u-boot**: stores the source code of u-boot, the folder named **v2017.09-rk3588** stores the u-boot source code of the legacy branch of the RK3588/RK3588S/RK3566 series development boards, the name of the folder of the u-boot source code Please do not modify it manually, if it is modified, the u-boot source code will be re-downloaded when the compiling system is running

j. **userpatches**: Store configuration files needed to compile scripts

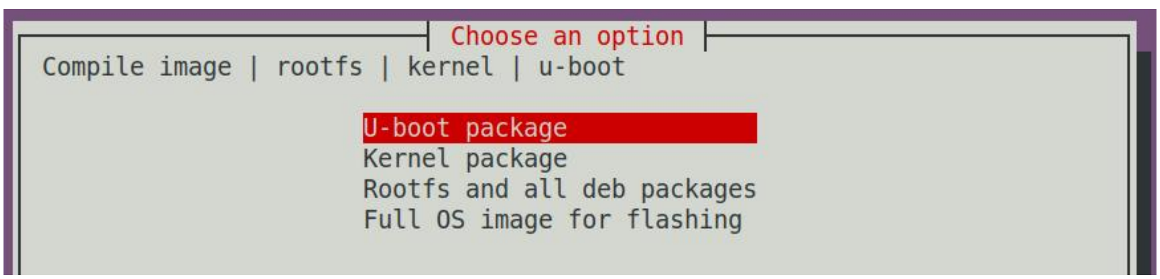
```
test@test:~/orangepi-build$ ls
build.sh   external   kernel   LICENSE   output   README.md   scripts   toolchains
u-boot   userpatches
```

4. 3. Compile u-boot

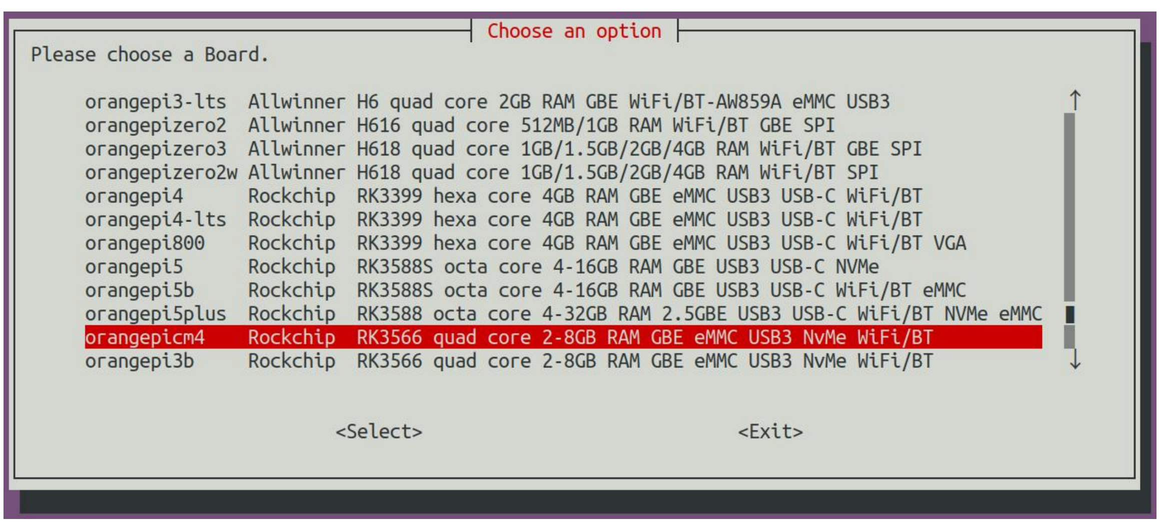
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package**，then enter



3) Then select the model of the development board



4) Then it will start to compile u-boot, and some information prompted during compilation is explained as follows

a. u-boot source code version

```
[ o.k. ] Compiling u-boot [ v2017.09 ]
```

b. The version of the cross-compilation toolchain

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 7.4.1 ]
```

c. Path to the generated u-boot deb package

```
[ o.k. ] Target directory [ orangepi-build/output/debs/u-boot ]
```

d. The package name of the generated u-boot deb package

```
[ o.k. ] File name [ linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb ]
```

e. Compilation time

```
[ o.k. ] Runtime [ 1 min ]
```

f. Repeat the command to compile u-boot, use the following command to start compiling u-boot directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi3b BRANCH=legacy
BUILD_OPT=u-boot KERNEL_CONFIGURE=no ]
```

5) View the u-boot deb package generated by compilation

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb
```

6) The files contained in the generated u-boot deb package are as follows

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ $ dpkg -x \
linux-u-boot-legacy-orangepi3b_1.0.0_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-orangepi3b_1.0.0_arm64.deb    usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-legacy-orangepi3b_1.0.0_arm64
    │   ├── idbloader.img
    │   ├── rkspi_loader.img
    │   └── u-boot.itb
    └── u-boot
```

```
├── LICENSE
├── orangepi-3b-rk3566_defconfig
└── platform_install.sh

3 directories, 6 files
```

7) When the orangepi-bulid compilation system compiles the u-boot source code, it will first synchronize the u-boot source code with the u-boot source code of the github server, so if you want to modify the u-boot source code, you first need to turn off the download and update function of the source code **(This function needs to be fully compiled once u-boot, otherwise it will prompt that the source code of u-boot cannot be found. If the source code package downloaded from Baidu cloud disk, there is no such problem, because the source code of u-boot is all cached)**, otherwise the changes made will be reverted, the method is as follows:

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

8) When debugging u-boot code, you can use the following method to update u-boot in the Linux image for testing

a. Upload the compiled u-boot deb package to the Linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the deb package of u-boot installed

```
root@orangepi:~# apt purge -y linux-u-boot-orangepi3b-legacy
```

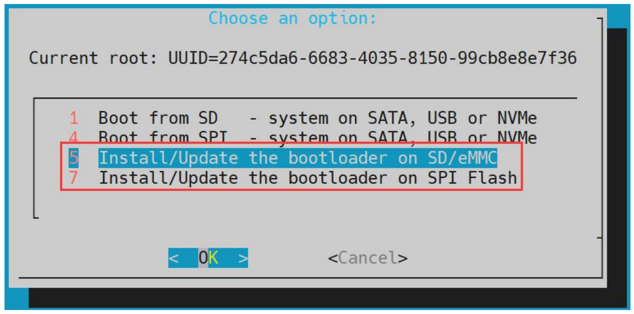
c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb
```

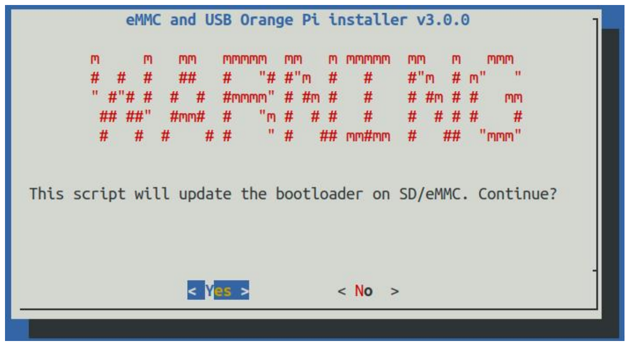
d. Then run the nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

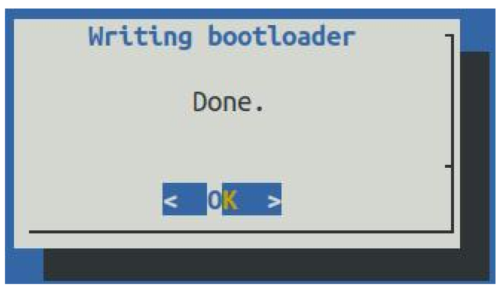
e. Then select **5 Install/Update the bootloader on SD/eMMC** to update the u-boot in the TF card or **7 Install/Update the bootloader on SPI Flash** to update the u-boot in the SPI Flash



f. After pressing the Enter key, a Warning will pop up first



g. Press the Enter key again to start updating u-boot, and the following information will be displayed after the update is completed



h. Then you can restart the development board to test whether the modification of u-boot takes effect

9) Other useful information

a. In the u-boot 2017.09 source code, the defconfig configuration file used by the development board is

orangepi-build/u-boot/v2017.09-rk3588/configs/orangepi-3b-rk3566_defconfig

b. In the u-boot 2017.09 source code, the dts file used by the development board is

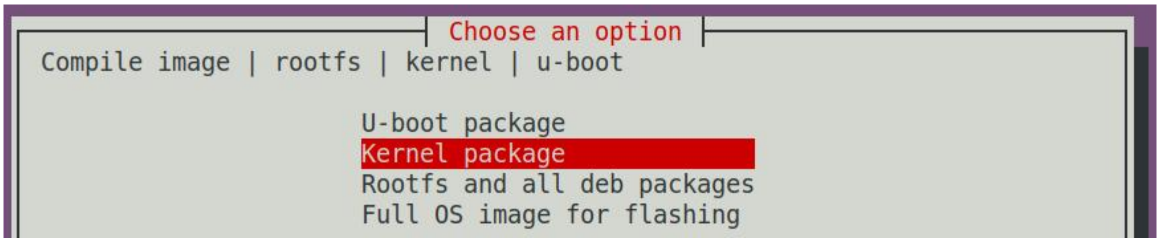
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3566-orangepi-3b.dts

4. 4. Compile the Linux kernel

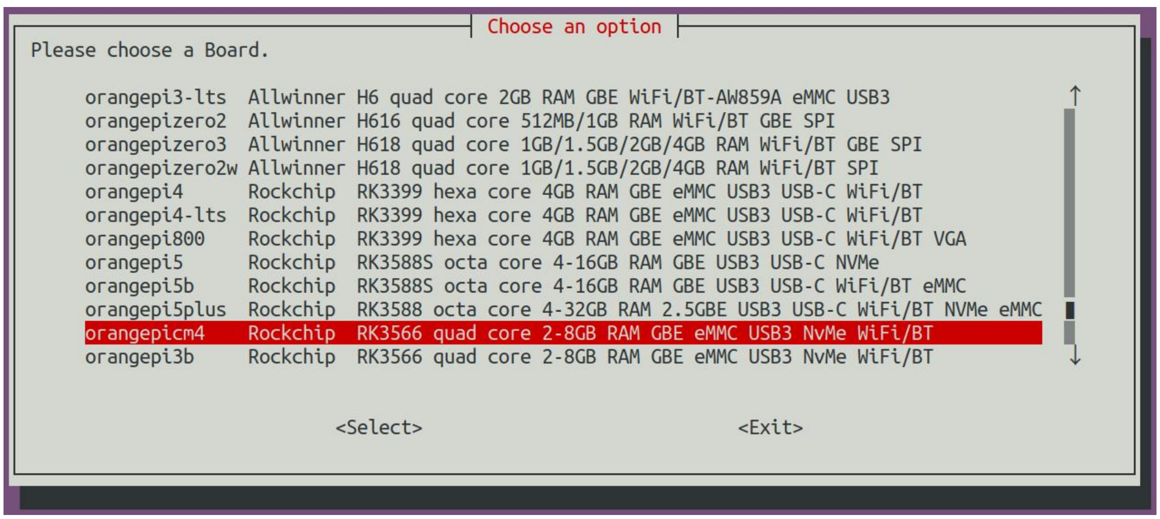
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

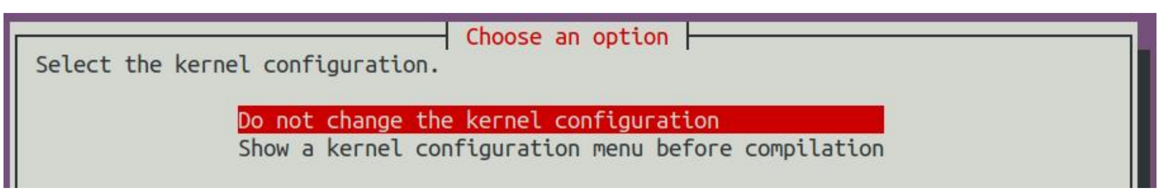
2) Select **Kernel package**, then enter



3) Then select the model of the development board

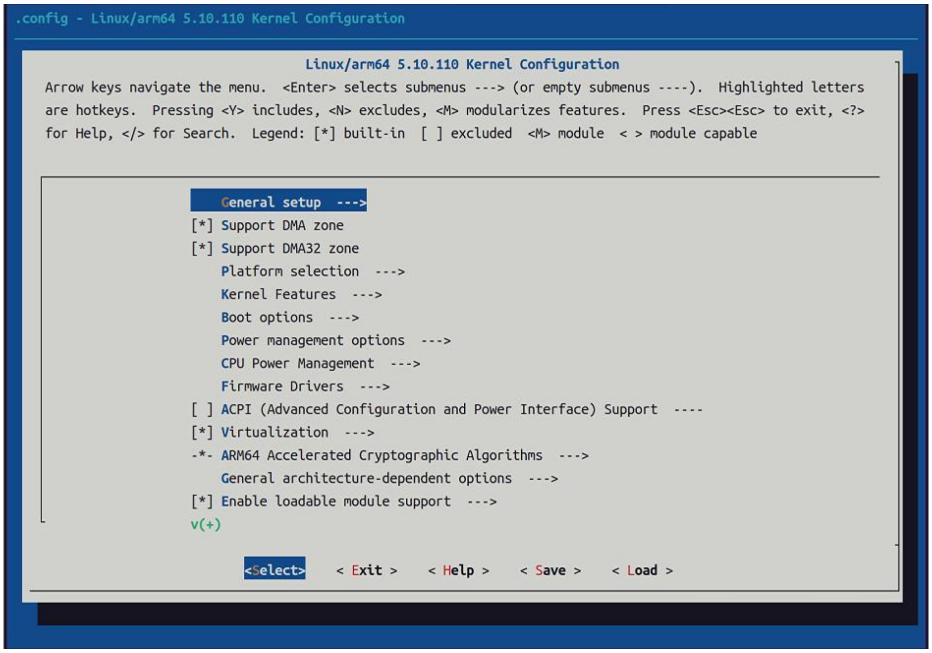


4) Then it will prompt whether to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If you need to modify the kernel configuration, select the second one.



5) If you choose to display the kernel configuration menu (the second option) in step 4, the kernel configuration interface opened by **make menuconfig** will pop up. At this time,

you can directly modify the kernel configuration, save and exit after modification. Yes, after exiting, the kernel source code will be compiled



a. If you do not need to modify the configuration options of the kernel, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily block the pop-up kernel configuration interface

```
test@test:~/orangepi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

b. You can also set **KERNEL_CONFIGURE=no** in the **orangepi-build/userpatches/config-default.conf** configuration file, which can permanently disable this function

c. If the following error is displayed when compiling the kernel, it is because the terminal interface of the Ubuntu PC is too small to display the **make menuconfig** interface. Please maximize the terminal of the Ubuntu PC and run the build.sh script again

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

6) Part of the information prompted when compiling the kernel source code is as follows

a. The version of the Linux kernel source code

[o.k.] Compiling current kernel [**5.10.160**]

b. The version of the cross-compilation toolchain used

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 11.2.1**]

c. The configuration file used by the kernel by default and the path where it is stored

[o.k.] Using kernel config file [**config/kernel/linux-rockchip-rk356x-legacy.config**]

d. The path of the deb package related to the kernel generated by compiling

[o.k.] Target directory [**orangepi-build/output/debs/**]

e. The package name of the compiled kernel image deb package

[o.k.] File name [**linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb**]

f. The time used for compilation

[o.k.] Runtime [**5 min**]

g. Finally, the compilation command to repeatedly compile the kernel selected last time will be displayed. Use the following command to start compiling the kernel source code directly without selecting through the graphical interface

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepi3b BRANCH=legacy BUILD_OPT=kernel KERNEL_CONFIGURE=no**]

7) View the deb package related to the kernel generated by compilation

a. **linux-dtb-legacy-rockchip-rk356x_1.0.0_arm64.deb** Contains dtb files used by the kernel

b. **linux-headers-legacy-rockchip-rk356x_1.0.0_arm64.deb** Include kernel header files

c. **linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-legacy-rockchip-rk356x_1.0.0_arm64.deb
output/debs/linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb
output/debs/linux-headers-legacy-rockchip-rk356x_1.0.0_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-rk356x_1.0.0_arm64.deb test/
test@test:~/orangepi_build/output/debs$ cd test
test@test:~/orangepi_build/output/debs/test$ dpkg -x \
linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb .
test@test:~/orangepi_build/output/debs/test$ ls
boot   etc   lib   linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb   usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.160-rockchip-rk356x
│   ├── System.map-5.10.160-rockchip-rk356x
│   └── vmlinuz-5.10.160-rockchip-rk356x
├── etc
│   └── kernel
├── lib
│   └── modules
├── linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb
└── usr
    ├── lib
    └── share
```

9) The orangepi-bulid compilation system will first synchronize the Linux kernel source code with the Linux kernel source code of the github server when compiling the Linux

kernel source code, so if you want to modify the Linux kernel source code, you first need to turn off the update function of the source code **(You need to fully compile the Linux kernel source code before turning off this function. Otherwise, you will be prompted that the source code of the Linux kernel cannot be found. If you download the source code package from Baidu cloud disk, there is no such problem, because the source code of Linux has been cached.)**, otherwise the The changes made will be reverted as follows:

Set the IGNORE_UPDATES variable in **userpatches/config-default.conf** to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If the kernel has been modified, the following method can be used to update the kernel and kernel modules of the development board Linux system

a. Upload the deb package of the compiled Linux kernel to the Linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the deb package of the installed Linux kernel

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk356x
```

c. Install the deb package of the new Linux kernel just uploaded

```
root@orangepi:~# dpkg -i linux-image-legacy-rockchip-rk356x_1.0.0_arm64.deb
```

d. Then restart the development board, and then check whether the kernel-related modifications have taken effect

```
root@orangepi:~# reboot
```

11) Other useful information

a. The storage location of the kernel configuration file is as follows, please do not go to the kernel source code to find the kernel configuration file used by the development board

```
orangepi-build/external/config/kernel/linux-rockchip-rk356x-legacy.config
```

b. The location of the dts file used by the development board is

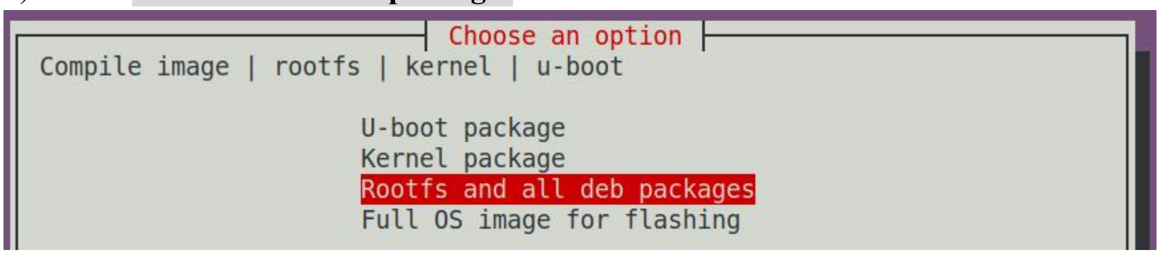
```
orangepi-build/kernel/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3566-orangepi-cm4.dts
```

4. 5. Compile rootfs

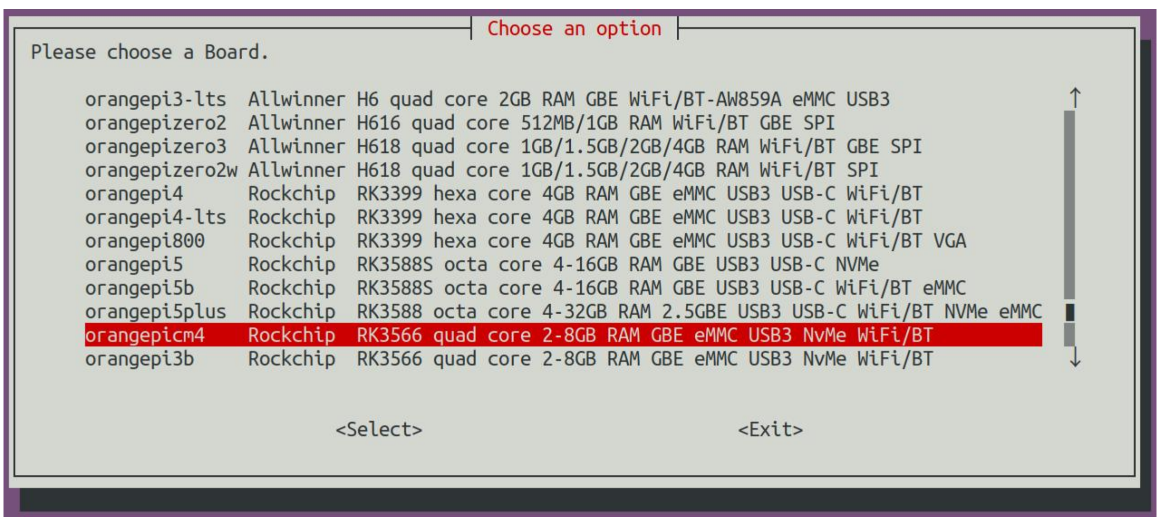
1) Run the build.sh script, remember to add sudo permission

test@test:~/orangepi-build$ **sudo ./build.sh**

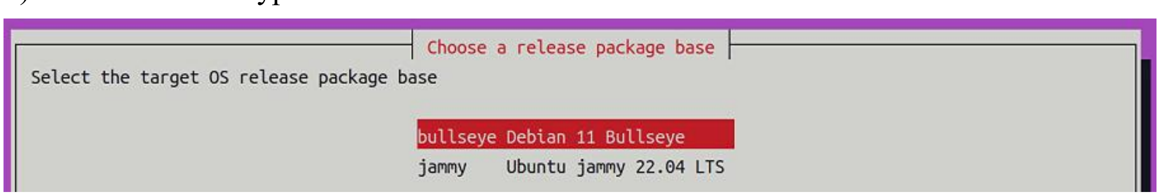
2) Select **Rootfs and all deb packages**, then enter



3) Then select the model of the development board



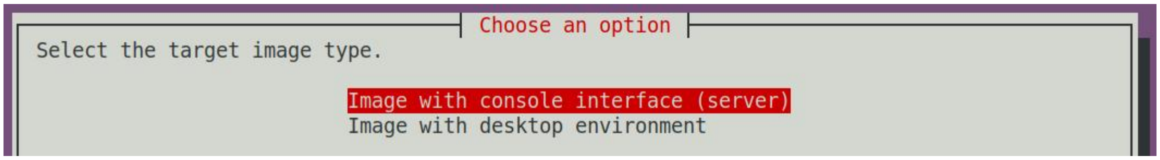
4) Then select the type of rootfs



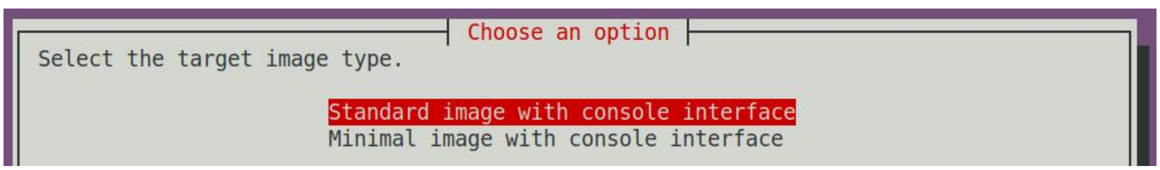
5) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small

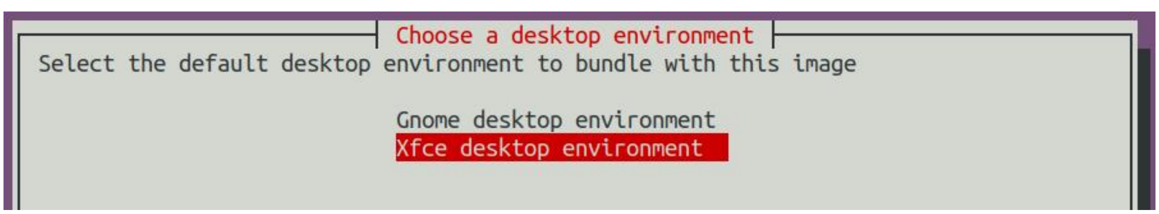
b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

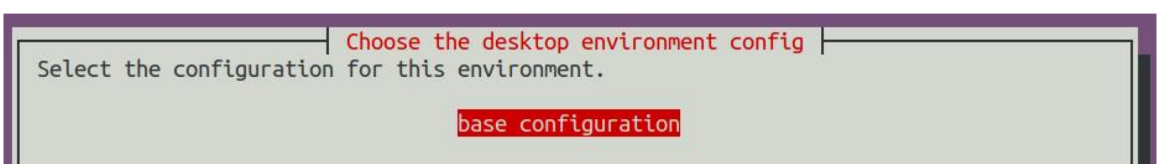


6) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**

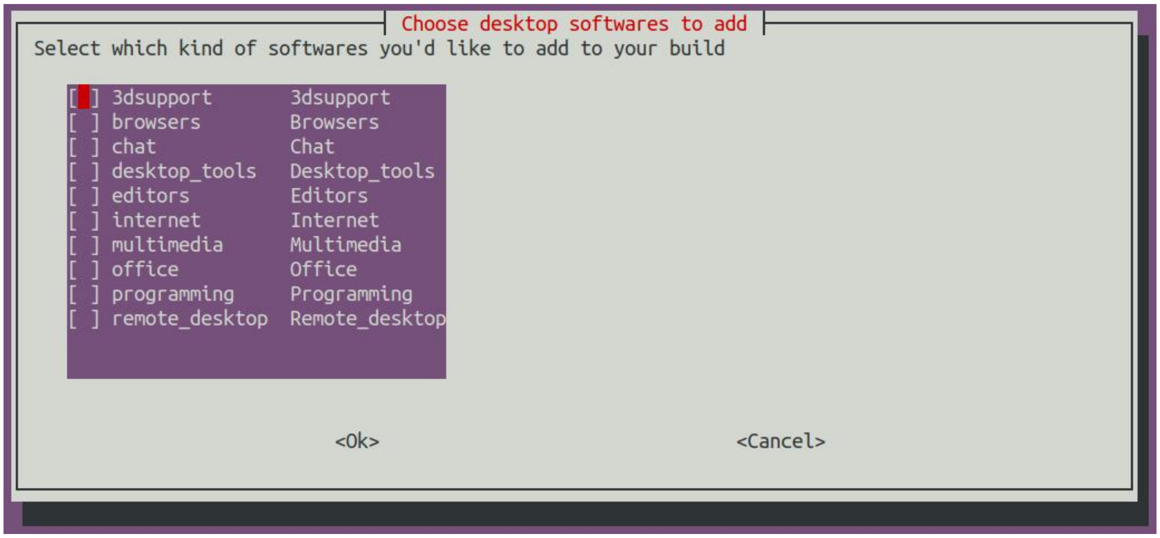


7) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, and Debian Bullseye mainly maintains XFCE and KDE desktops





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



8) Then it will start to compile rootfs, and some of the information prompted during compilation are as follows

a. The type of rootfs

[o.k.] local not found [Creating new rootfs cache for **jammy**]

b. The storage path of the compiled rootfs compressed package

[o.k.] Target directory [**external/cache/rootfs**]

c. The name of the rootfs compressed package generated by compilation

[o.k.] File name [**jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**]

d. The time used for compilation

[o.k.] Runtime [**13 min**]

9) View the rootfs compressed package generated by compilation

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is the rootfs compressed package, the meaning of each field of the name is

a) **jammy** indicates the type of Linux distribution of rootfs

b) **xfce** means rootfs is the type of desktop version, if it is **cli**, it means the type of server version

c) **arm64** represents the architecture type of rootfs

d) **f930ff6ebbac1a72108a2e100762b18f** is the MD5 hash value generated by the package names of all software packages installed by rootfs. As long as the list of software packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to generate Determine whether rootfs needs to be recompiled

b. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.current
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list
```

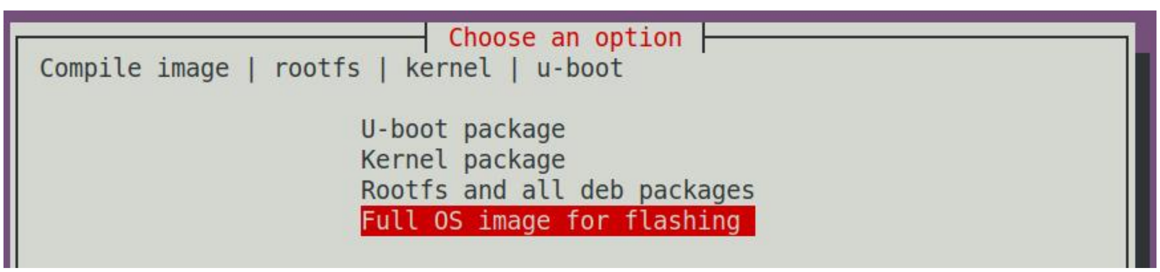
10) If the required rootfs already exists under **external/cache/rootfs**, then compiling rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has If there is rootfs available in the cache, use it directly, which can save a lot of download and compilation time.

4. 6. Compile Linux image

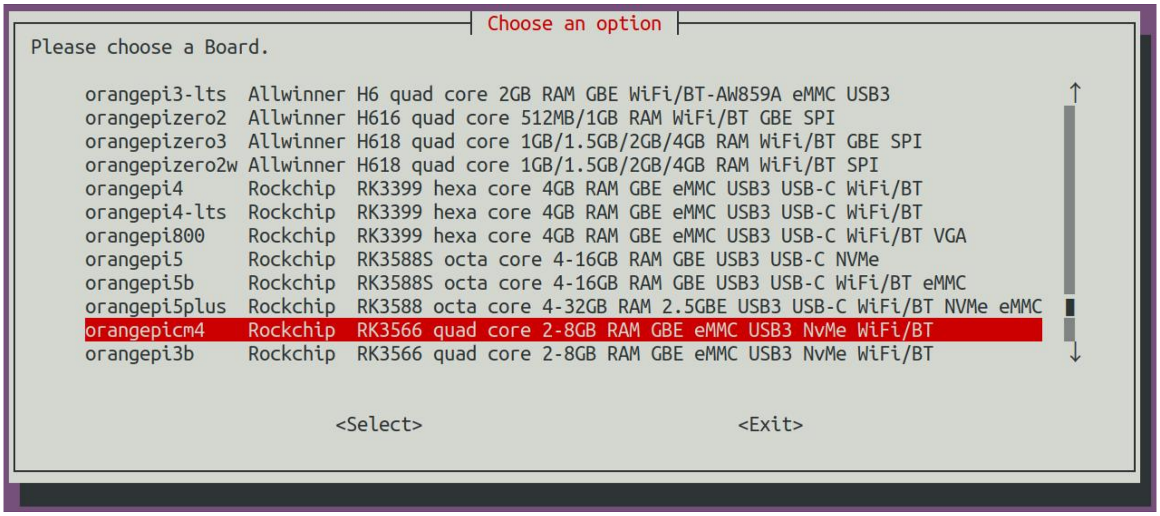
1) Run the build.sh script, remember to add sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

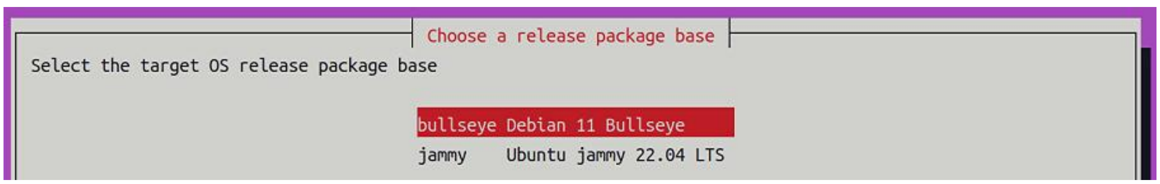
2) Select **Full OS image for flashing**，then enter



3) Then select the model of the development board



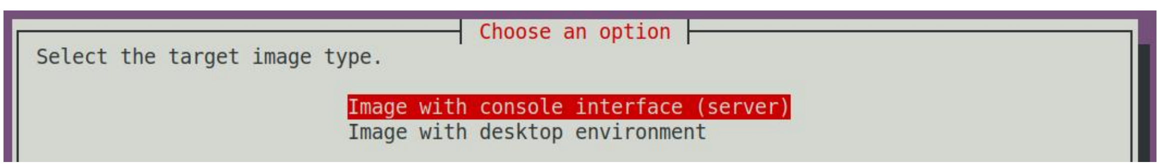
4) Then select the type of rootfs



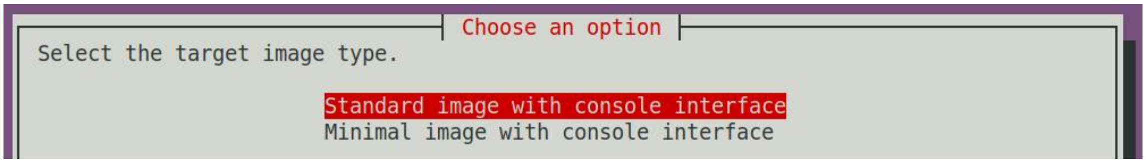
5) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small

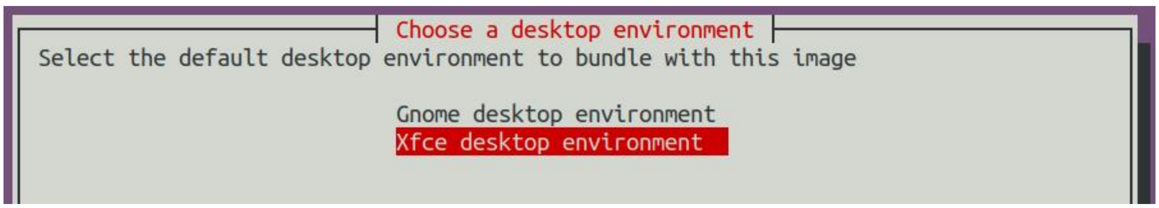
b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

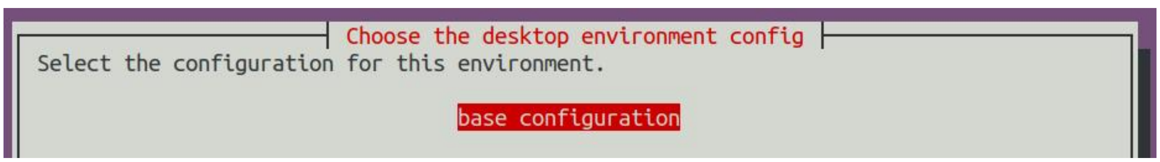


6) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**

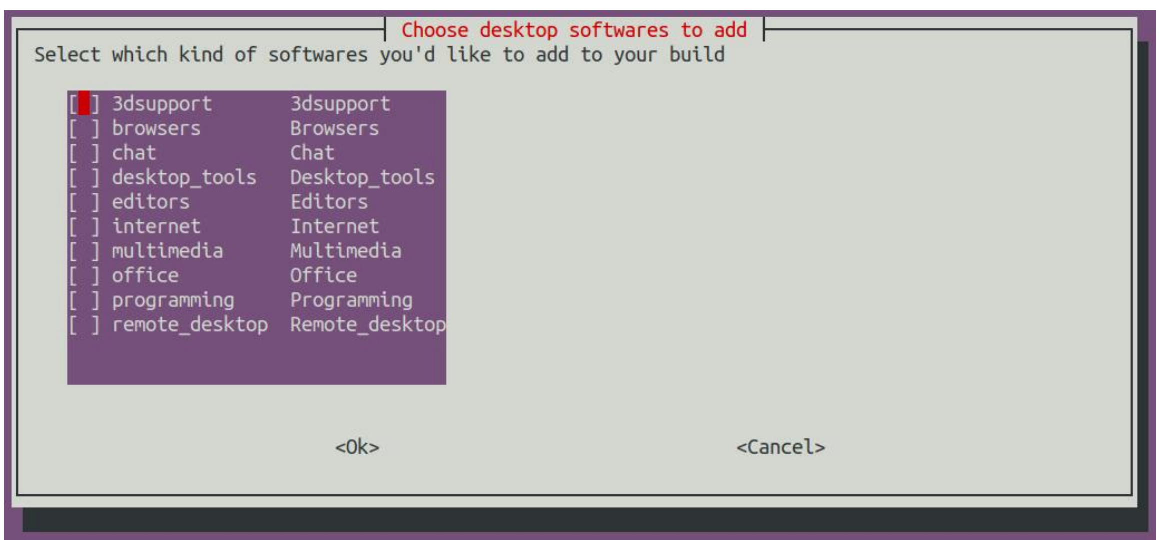


7) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, and Debian Bullseye mainly maintains XFCE and KDE desktops





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



8) Then it will start to compile the Linux image. The general process of compilation is as follows

a. Initialize the compilation environment of Ubuntu PC and install the software packages required for the compilation process

b. Download the source code of u-boot and Linux kernel (if cached, only update the code)

c. Compile u-boot source code and generate u-boot deb package

d. Compile the Linux source code and generate Linux-related deb packages

e. Make the deb package of Linux firmware

f. Make the deb package of the orangepi-config tool

g. Create a deb package supported by the board

h. If you are compiling the desktop image, you will also create desktop-related deb packages

i. Check whether the rootfs has been cached, if not, recreate the rootfs, if it has been cached, directly decompress and use

j. Install the previously generated deb package into rootfs

k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.

l. Then make an image file and format the partition, the default type is ext4

m. Then copy the configured rootfs to the mirrored partition

n. Then update initramfs

o. Finally, write the bin file of u-boot into the image through the dd command

9) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160/Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img ]
```

b. Compilation time

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the command to compile the image, and use the following command to start compiling the image directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh    BOARD=orangepi3b BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

5. Instructions for using the Orange Pi OS Arch system

5. 1. Orange Pi OS Arch system function adaptation

Function	OPi OS Arch
USB2.0x3	OK
USB3.0x1	OK
SPIFlash+M.2 NVMe SSD Boot	OK, It needs to be pasted with SPI Flash to work normally
WIFI	OK
Bluetooth	OK
GPIO（40pin）	OK
UART（40pin）	OK
SPI（40pin）	OK
I2C（40pin）	OK
PWM（40pin）	OK
3pin debugging serial port	OK
eMMC start	OK
TF card start	OK
HDMI video	OK
HDMI audio	OK
Raspberry Pi 5 inch screen display	OK
Raspberry Pi 5-inch screen touch function	OK
eDP display	OK
OV5647 camera	The kernel driver is OK, 3A is not

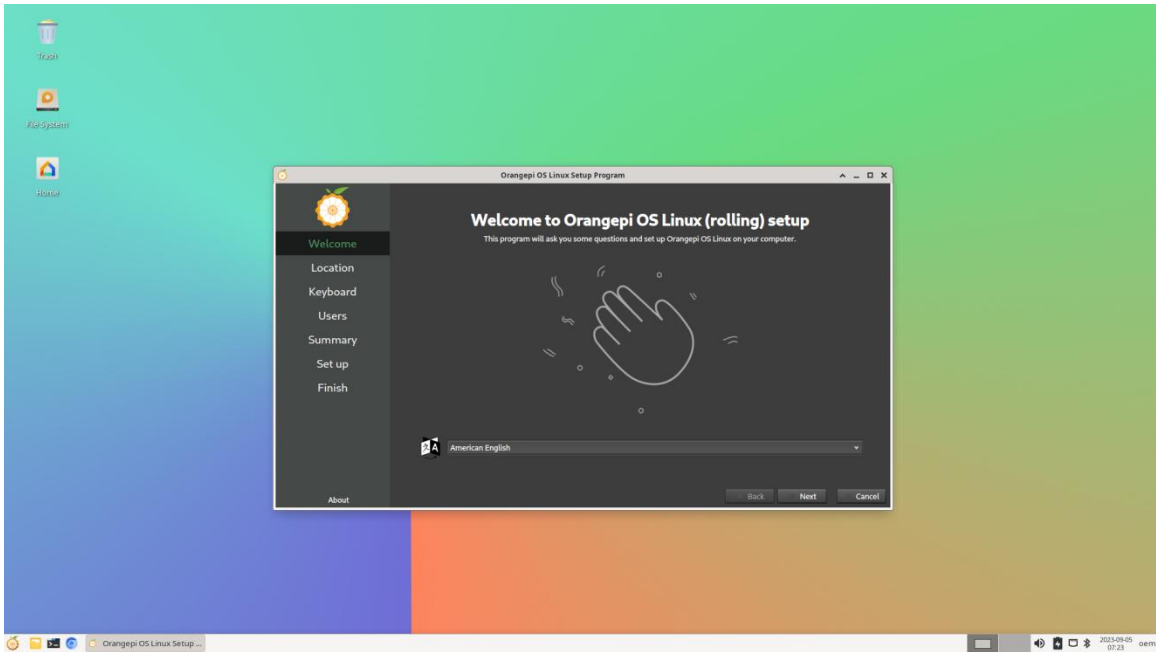
	adjusted
Gigabit Ethernet port	OK
Network port status light	OK
headphone playback	OK
headphone recording	OK
LED lights	OK
GPU	NO
NPU	NO
VPU	NO

5. 2. Orange Pi OS Arch System User Guide Instructions

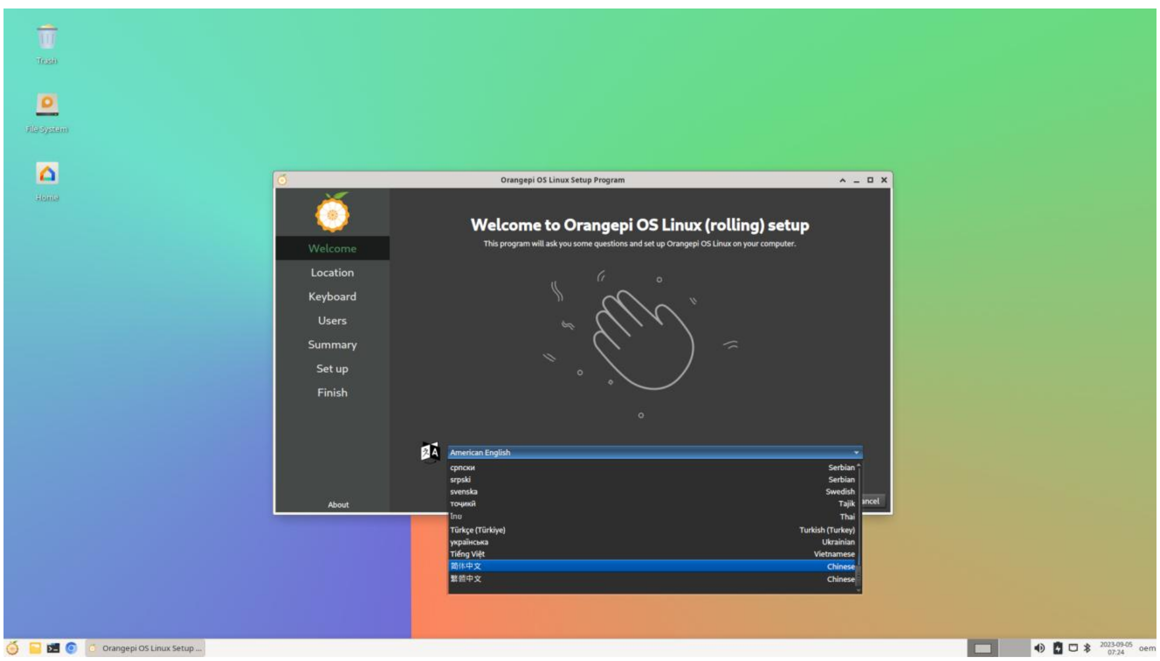
First of all, please note that the OPi OS Arch system does not have a default orangepi user and password, so after burning and starting the system, it is impossible to log in remotely through the serial port and ssh directly (not even the root user). This is different from Ubuntu and Debian systems.

When the OPi OS Arch system starts for the first time, it needs to be connected to an HDMI display, and then initialize the system settings through the user wizard (including creating a new user name and setting a password). The setup steps of the User Wizard are as follows:

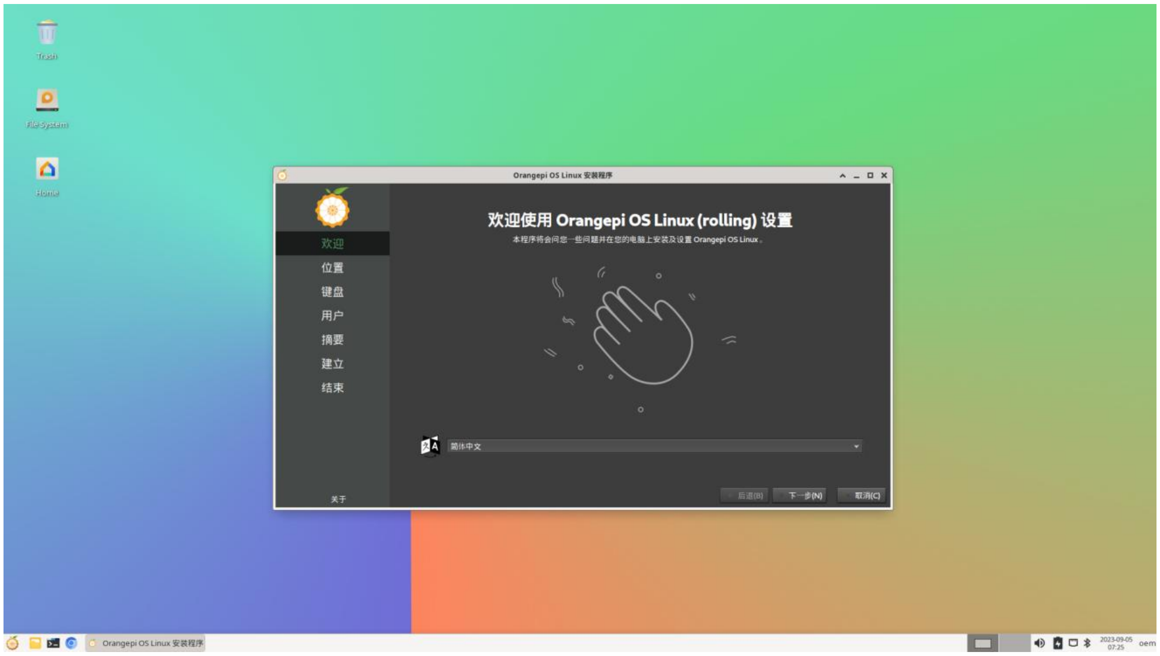
a) After burning the system, you will see the user wizard program shown in the figure below after starting the system for the first time and entering the desktop



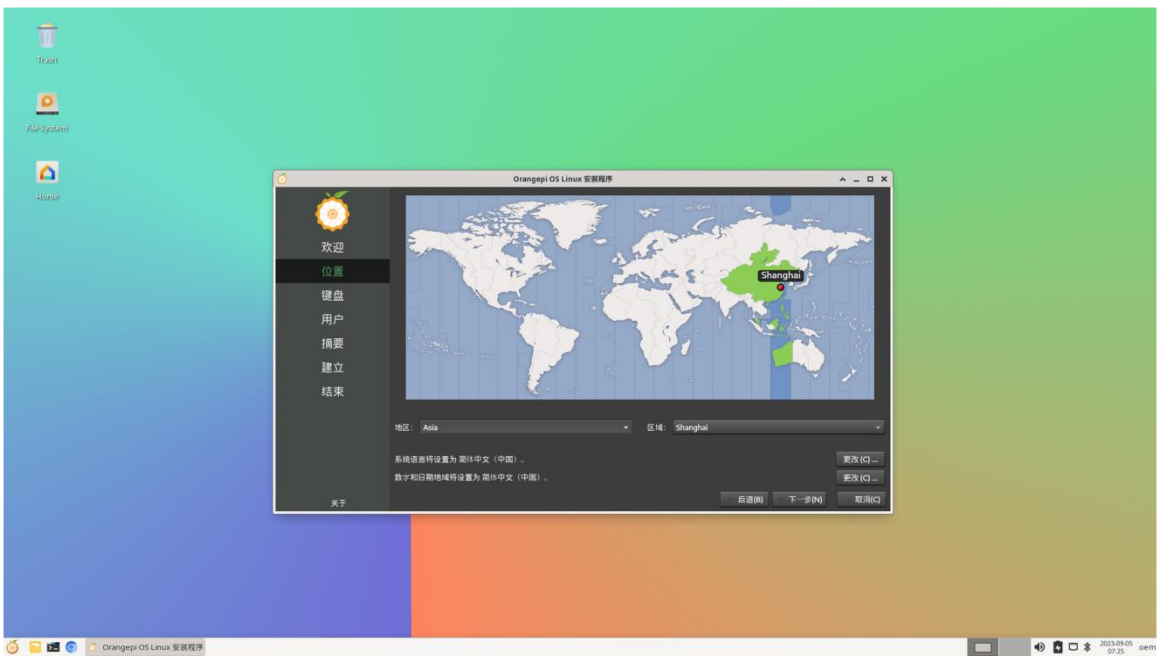
b) First you need to choose the desired language



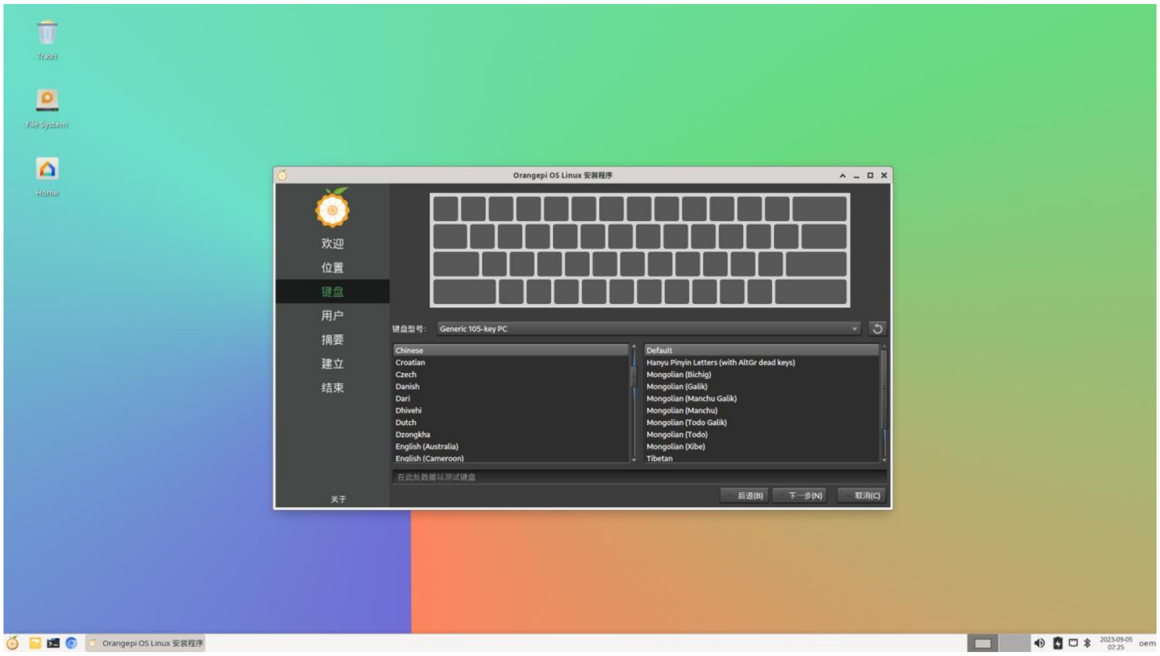
c) After selecting the language, the user guide will immediately switch to the corresponding language interface, such as the Chinese display as shown below



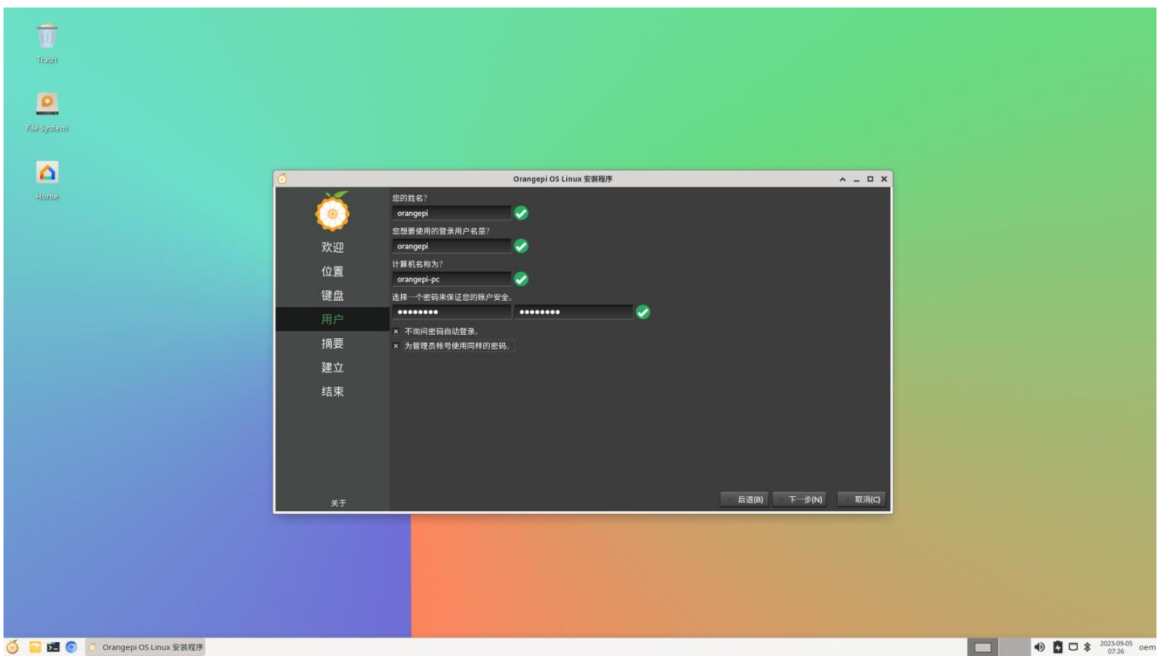
d) Then select the area



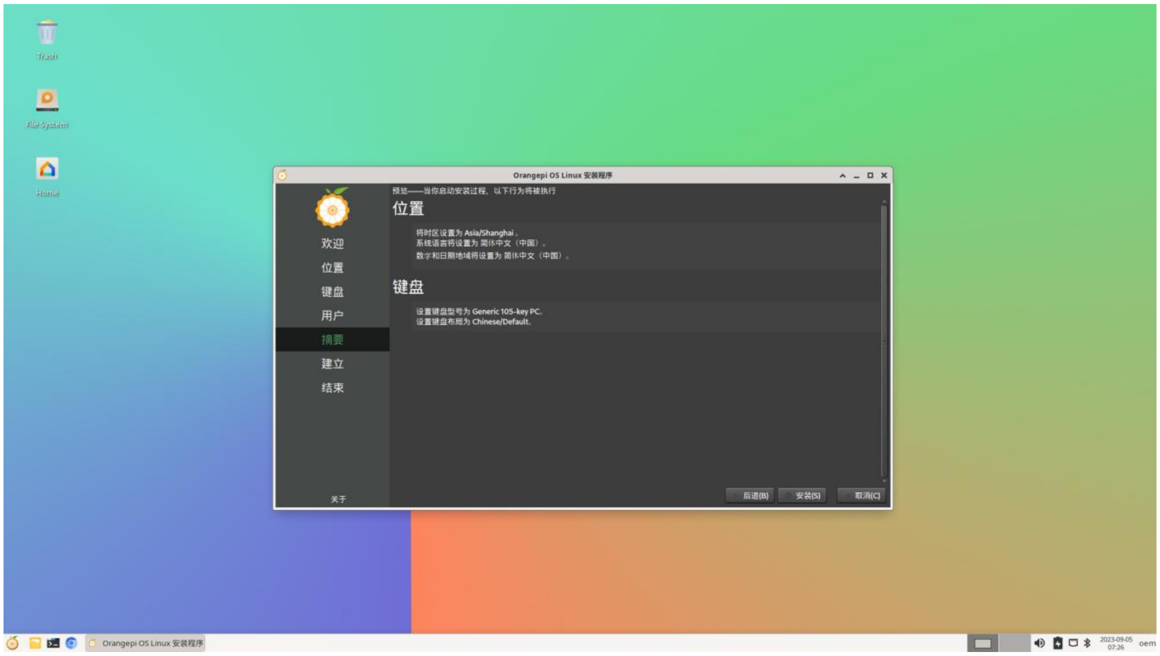
e) Then select the keyboard model



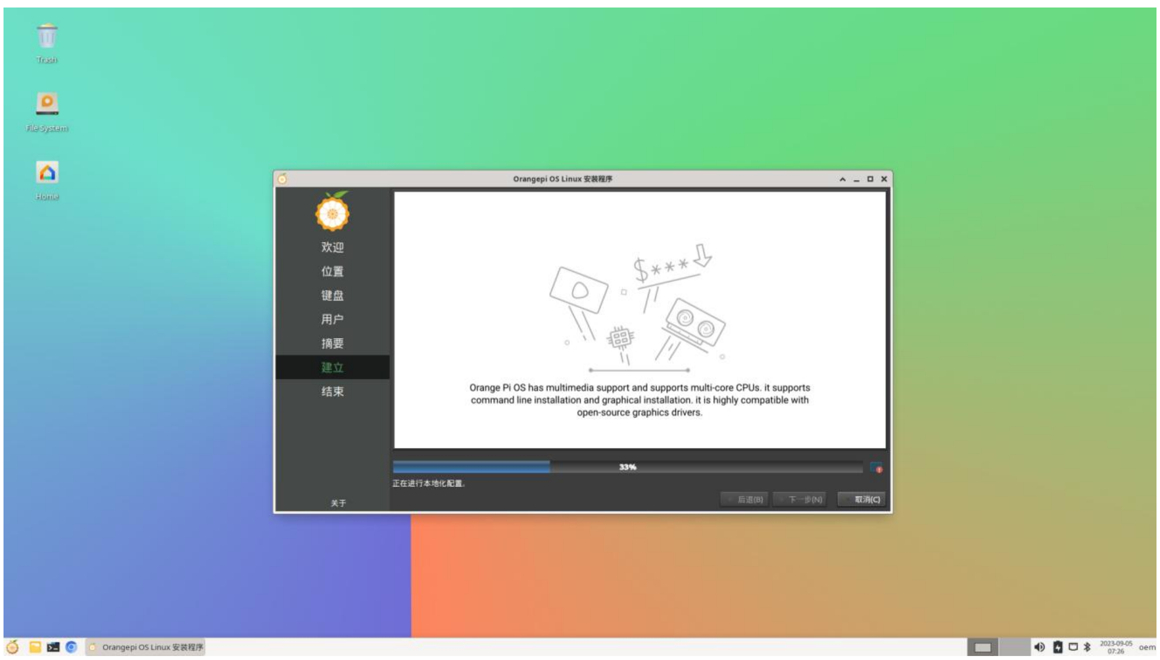
f) Then create a new user name and set a password



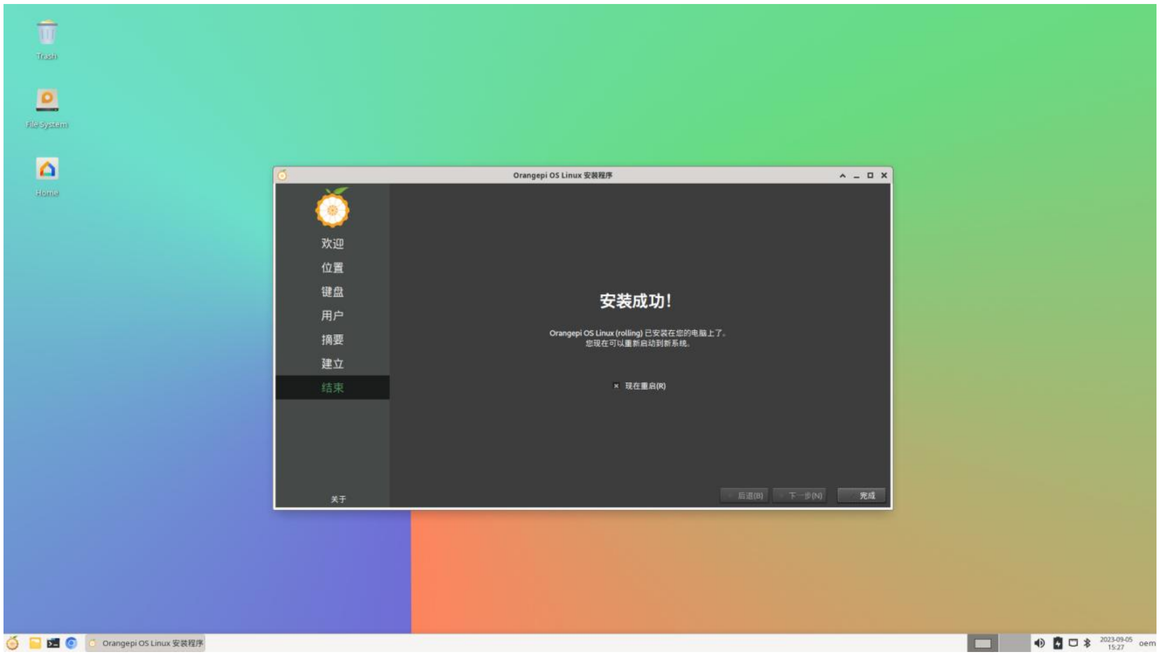
g) Then make sure that there is no problem with the selection, and then click the **install** button



h) Then wait for the installation to complete



i) After the installation is complete, you need to click the **Finish** button to restart the system



j) After restarting, the Orange Pi Hello program will be started automatically. At this time, you need to remove the check status in the lower right corner **when starting up**. Otherwise, you need to manually close the Orange Pi Hello program every time you start it.

At this point, you can use the newly created user name and password to log in to the OPi OS system through the serial port or ssh.

5. 3. How to set DT overlays

LCD MIPI screen, eDP screen, and multiplexing functions such as I2C/SPI/UART/PWM in 40pin are disabled by default in the dts of the kernel, and the corresponding DT overlays need to be manually enabled to use.

The method of opening DT overlays in OPi OS Arch system is as follows:

1) First open the **/boot/extlinux/extlinux.conf** configuration file

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

2) Then open the corresponding configuration by adding **FDTOVERLAYS /dtbs/rockchip/overlay/xxx.dtbo** in **/boot/extlinux/extlinux.conf**

Note that xxx.dtbo in FDTOVERLAYS /dtbs/rockchip/overlay/xxx.dtbo needs to be replaced with the specific dtbo configuration, please do not copy it.

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/xxx.dtbo       #Configuration that needs to be added
```

3) The storage path of xxx.dtbo in the OPi OS Arch image is as follows, please note that not all dtbos under this path can be used.

```
/boot/dtbs/rockchip/overlay/
```

4) The DT overlays configuration that can be used by the development board is as follows

Features on the development board	Corresponding DT overlays configuration
EDP screen	rk356x-edp.dtbo
Raspberry Pi 5 inch screen	rk356x-raspi-7inch-touchscreen.dtbo
40pin expansion interface - I2C2	rk356x-i2c2-m1.dtbo

40pin expansion interface - I2C3	**rk356x-i2c3-m0.dtbo**
40pin expansion interface - I2C4	**rk356x-i2c4-m0.dtbo**
40pin extension interface - PWM11	**rk356x-pwm11-m1.dtbo**
40pin extension interface - PWM15	**rk356x-pwm15-m1.dtbo**
40pin expansion interface - UART3	**rk356x-uart3-m0.dtbo**
40pin expansion interface - UART7	**rk356x-uart7-m2.dtbo**
40pin expansion interface - UART9	**rk356x-uart9-m2.dtbo**
40pin expansion interface - SPI3	**rk356x-spi3-m0-cs0-spidev.dtbo**

5) If you need to open multiple configurations at the same time, just add the paths of multiple configurations directly behind **FDTOVERLAYS**. For example, the configurations to open i2c2 and pwm11 at the same time are as follows

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk356x-i2c2-m1.dtbo   /dtbs/rockchip/overlay/rk356x-pwm11-m1.dtbo
```

6) After setting, you need to restart the system to make the configuration take effect

```
[orangepi@orangepi-pc ~]$ sudo reboot
```

5. 4. Use of Raspberry Pi 5-inch screen

5. 4. 1. How to assemble the Raspberry Pi 5-inch screen

Please refer to **the assembly method of the Raspberry Pi 5-inch screen** (**click the text in the blue part to jump to the corresponding position**).

5. 4. 2. How to open Raspberry Pi 5-inch screen configuration

By default, OPi OS Arch mirroring does not enable the configuration of the Raspberry Pi 5-inch screen. If you need to use the Raspberry Pi 5-inch screen, you need to manually open it. The method to open the configuration is as follows:

a. First add the following configuration in **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk356x-raspi-7inch-touchscreen.dtbo            #Configuration that needs to be added
```

b. Then restart the system

```
[orangepi@orangepi-pc ~]$ sudo reboot
```

After restarting, you can see the display on the LCD screen as follows:

5. 5. How to use the eDP screen

5. 5. 1. Assembly method of eDP screen

Please refer to **how to use the eDP screen** (**click the text in the blue part to jump**

to the corresponding position).

5. 5. 2. How to open eDP screen configuration

The OPi OS Arch image does not enable the eDP screen configuration by default. If you want to use the eDP screen, you need to manually open it. The method to open the configuration is as follows:

a. First add the following configuration in **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL OPIOS ARM
LINUX /Image
FDT /dtbs/rockchip/rk3566-orangepi-3b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk356x-edp.dtbo          #Configuration that needs to be added
```

b. Then restart the system

```
[orangepi@orangepi-pc ~]$ sudo reboot
```

After restarting, you can see that the display of the eDP screen is as follows:

5. 6. How to install the software

Use the pacman package management tool to install software that is not in OPi OS. For example, the command to install the vim editor is as follows. If you want to install other software, you only need to replace vim with the package name of the software you want to install.

```
[orangepi@orangepi-pc ~]$ sudo pacman -Syy vim
```

6. Android 11 operating system instructions

6. 1. Supported Android versions

Android version	Kernel version
Android 11	Linux4.19

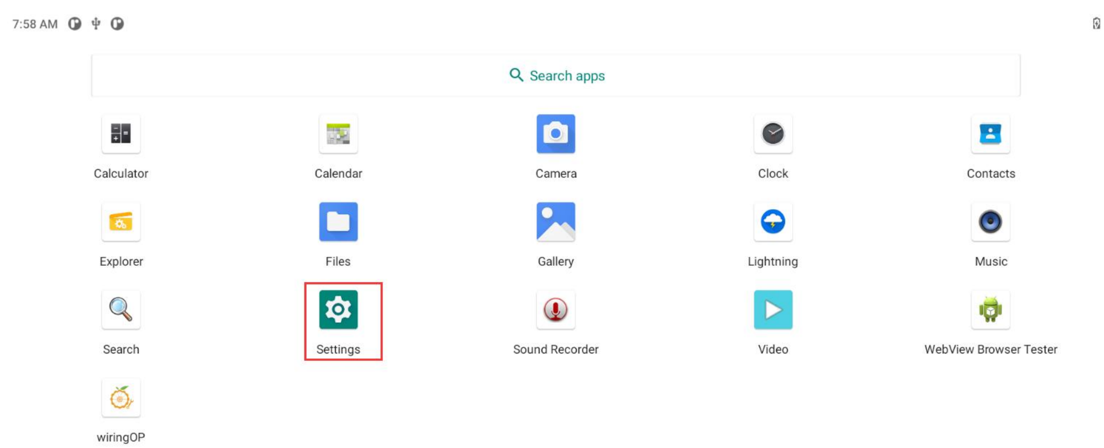
6. 2. Android Function Adaptation

Functions	Android 11
USB2.0x3	OK
USB3.0x1	OK
M.2 NVMe SSD boot	OK
WIFI	OK
Bluetooth	OK
GPIO（40pin）	OK
UART（40pin）	OK
SPI（40pin）	OK
I2C（40pin）	OK
PWM（40pin）	OK
PWM fan interface	OK
3pin Debugging serial port	OK
EMMC	OK
TF card boot	OK
HDMI video	OK
HDMI Audio	OK
LCD	OK
eDP display	OK
OV5647 Camera	The kernel driver is OK, 3A is not

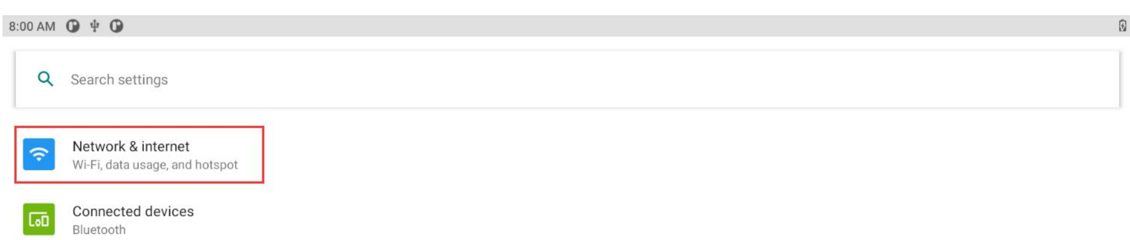
	adjusted
Gigabit network port	**OK**
Network port status indicator	**OK**
Headphone playback	**OK**
Headphone recording	**OK**
LED Light	**OK**
GPU	**OK**
NPU	**OK**
VPU	**OK**
RTC	**OK**

6. 3. WIFI connection test method

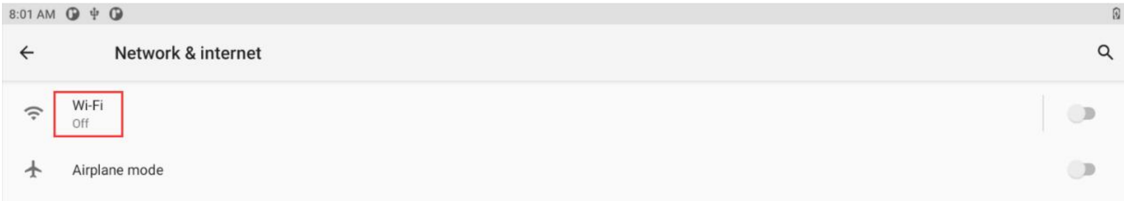
1) First click enter **Setting**



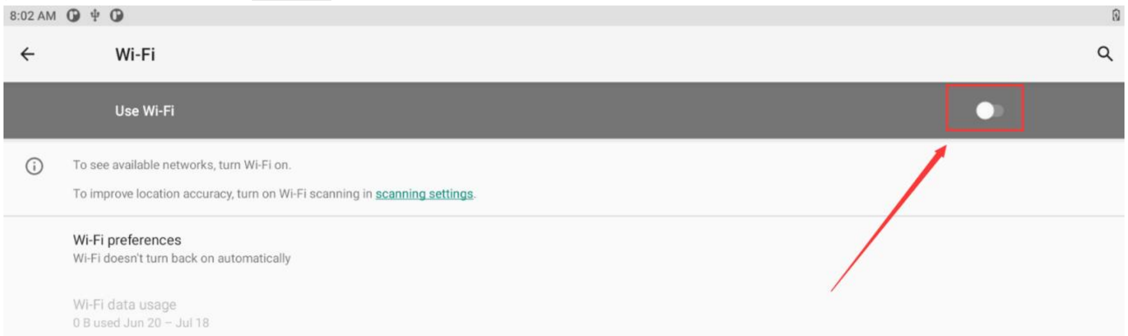
2) Then select **Network & internet**



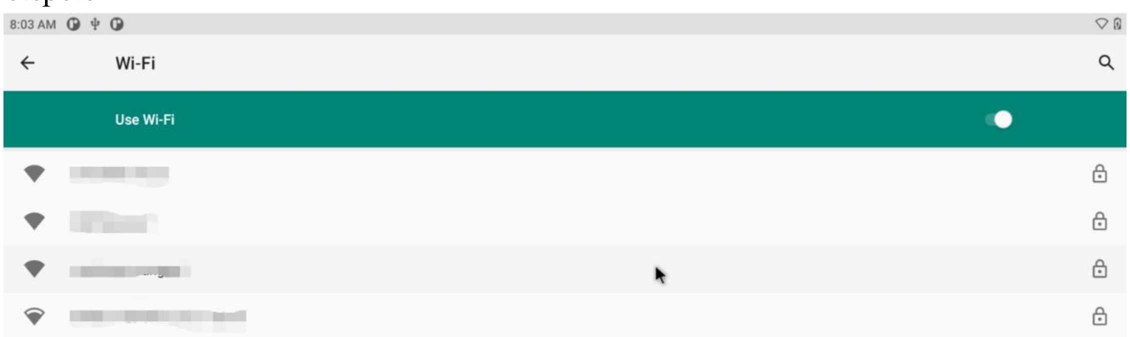
3) Then select **Wi-Fi**



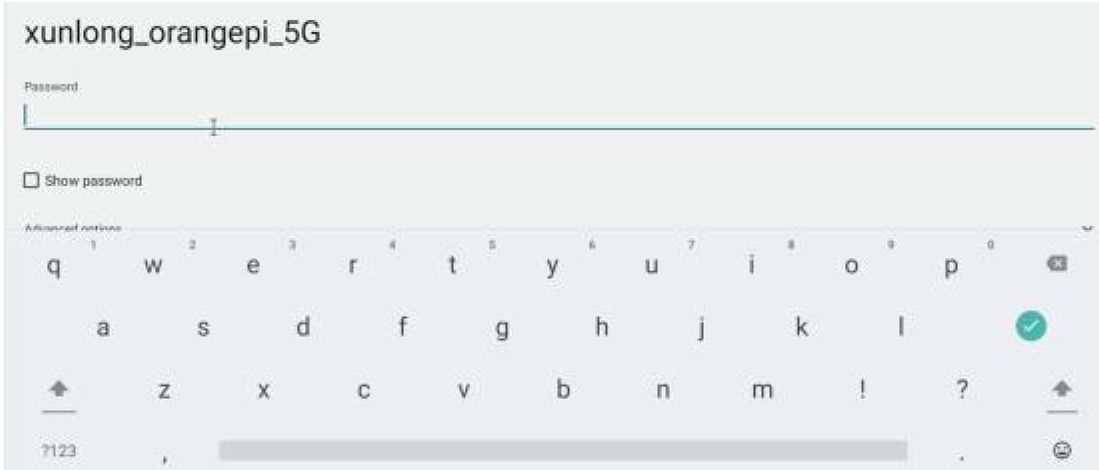
4) Then turn on the **Wi-Fi** switch



5) After turning on **Wi-Fi**, if everything is normal, you can scan for nearby Wi-Fi hotspots

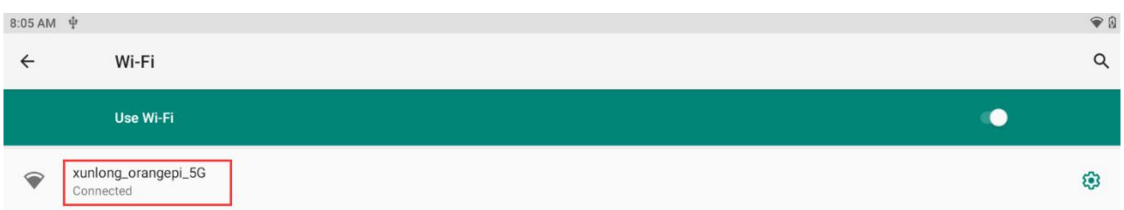


6) Then select the Wi-Fi you want to connect to, and the password input interface shown in the figure below will pop up



7) Then use the keyboard to enter the password corresponding to Wi-Fi, and then use the mouse to click the Enter button in the virtual keyboard to start connecting to Wi-Fi

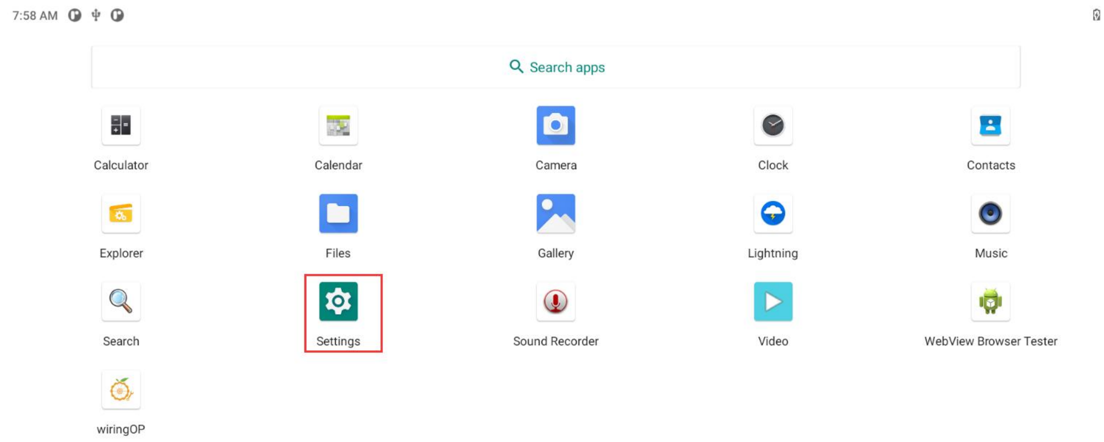
8) After the Wi-Fi connection is successful, the display is as shown in the figure below:



6. 4. How to use Wi-Fi hotspot

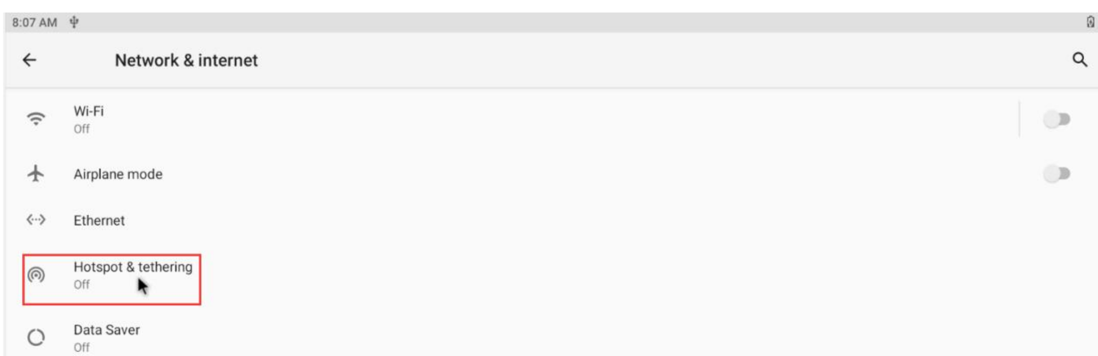
1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally

2) Then select **Settings**

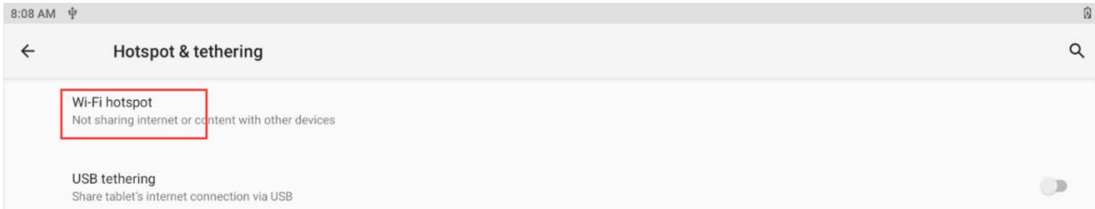


3) Then select **Network & internet**

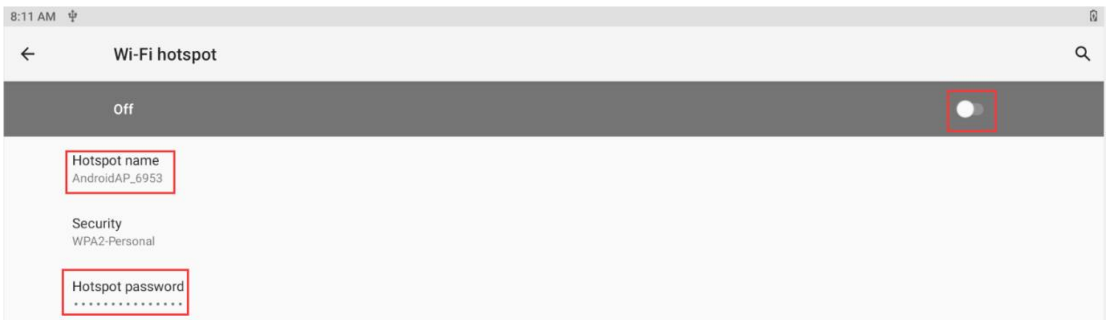
4) Then select **Hotspot & tethering**



5) Then select **Wi-Fi hotspot**



6) Then turn on the **Wi-Fi hotspot**, you can also see the name and password of the generated hotspot in the figure below, remember them, and use them when connecting to the hotspot (If you need to modify the name and password of the hotspot, you need to turn off the **Wi-Fi hotspot** first, and then you can modify it)



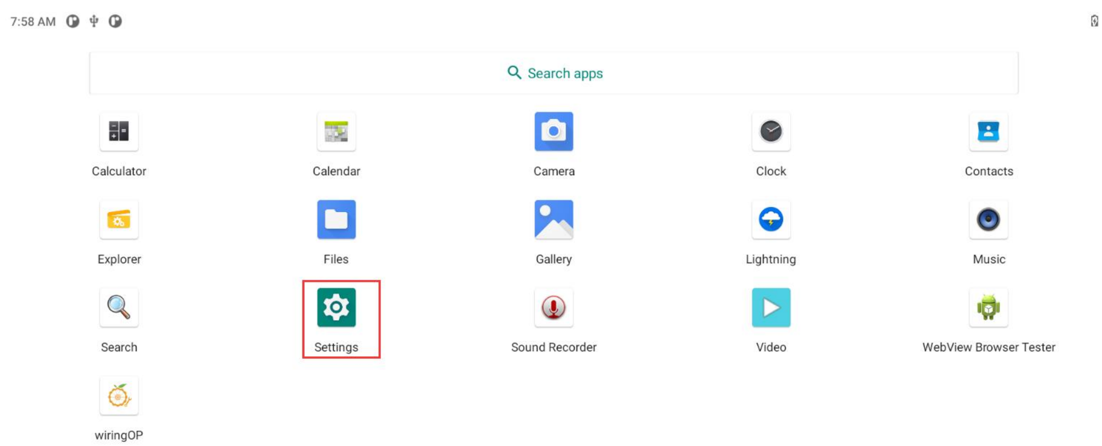
7) At this time, you can take out your mobile phone. If everything is normal, you can find the WIFI hotspot with the same name (**here AndroidAP_6953**) displayed under the **Hotspot name** in the above picture in the WI-FI list searched by the mobile phone. Then you can click **AndroidAP_6953** to connect to the hotspot, and the password can be seen under the **Hotspot password** in the above picture

8) After the connection is successful, it will be displayed as shown in the figure below (the interface of different mobile phones will be different, the specific interface is subject

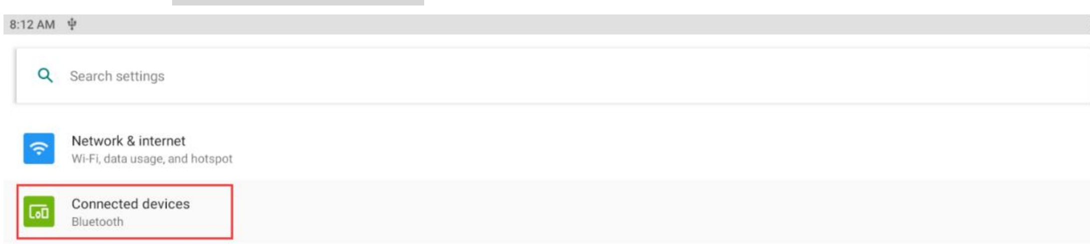
to the display of your mobile phone). At this point, you can open a webpage on your mobile phone to see if you can access the Internet. If you can open the webpage normally, it means that the **WI-FI Hotspot** of the development board can be used normally.

6. 5. Bluetooth test method

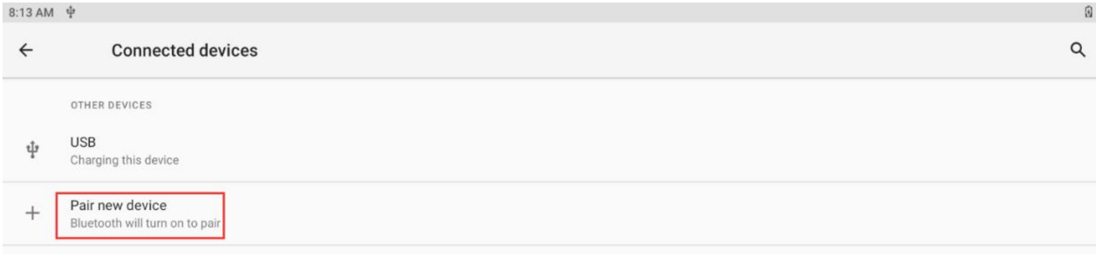
1) First click enter **Setting**



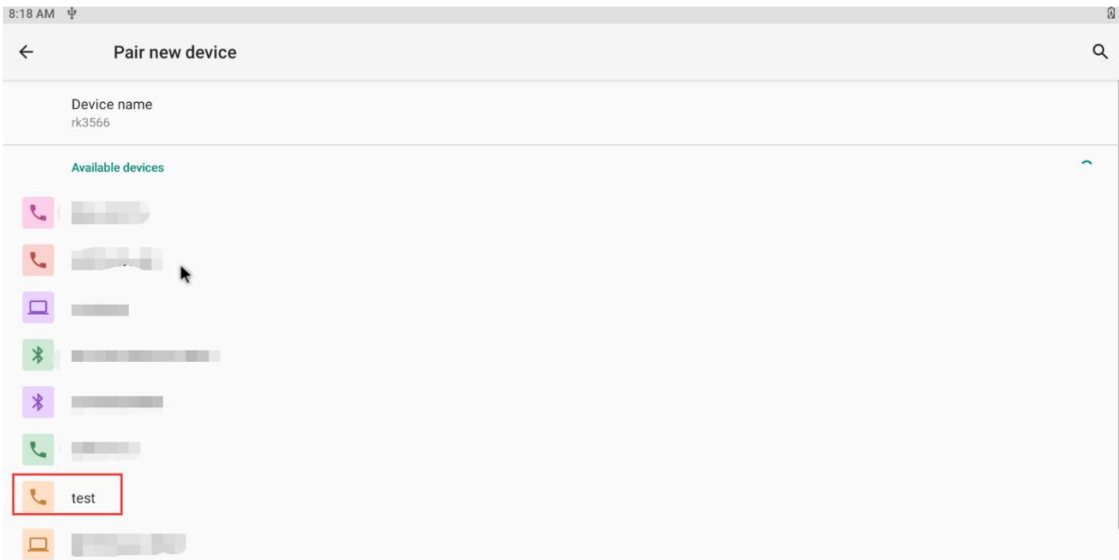
2) Then select **Connected devices**



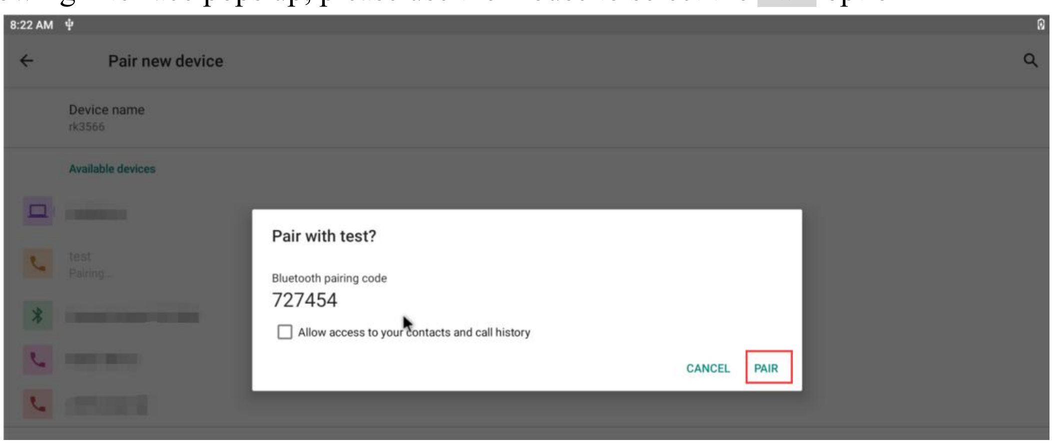
3) Then click **Pair new device** to turn on Bluetooth and start scanning the surrounding Bluetooth devices



4) The searched Bluetooth devices will be displayed under **Available devices**



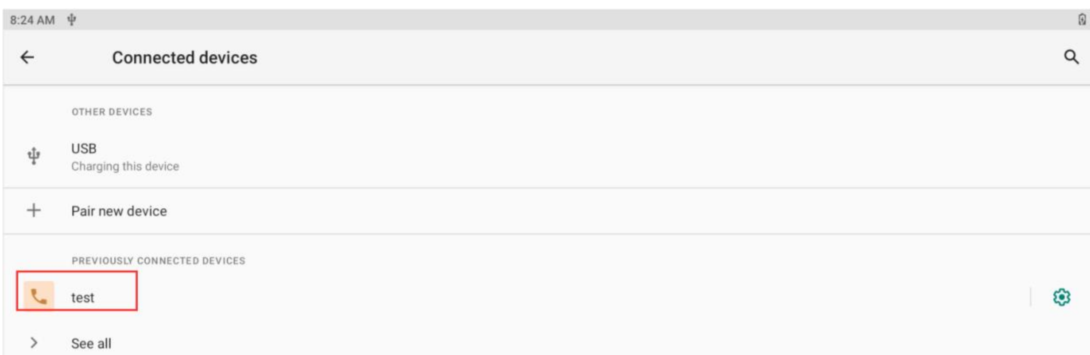
5) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option



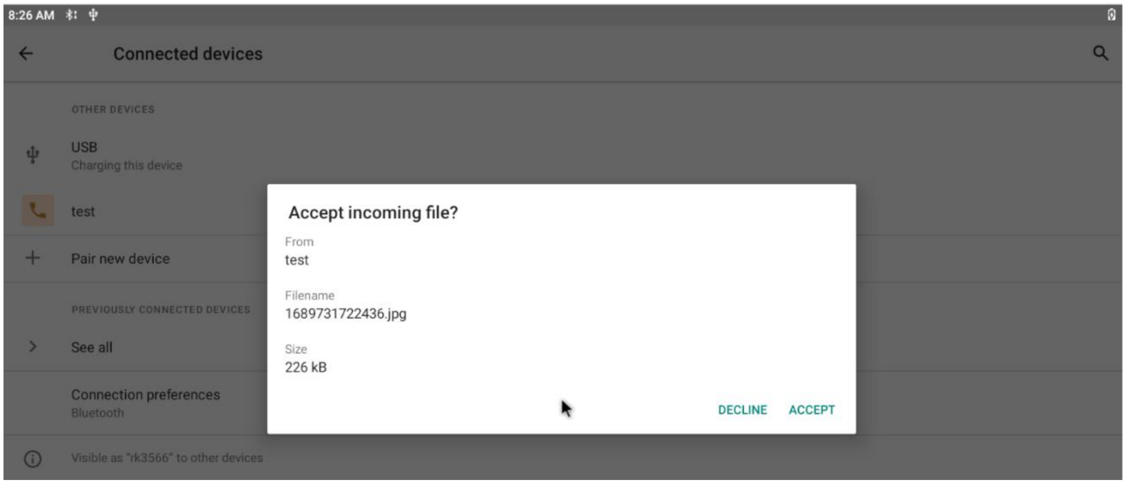
6) The test here is the configuration process of the development board and the Bluetooth of the Android mobile phone. At this time, the following confirmation interface will pop up on the mobile phone. After clicking the pairing button on the mobile phone, the

pairing process will start

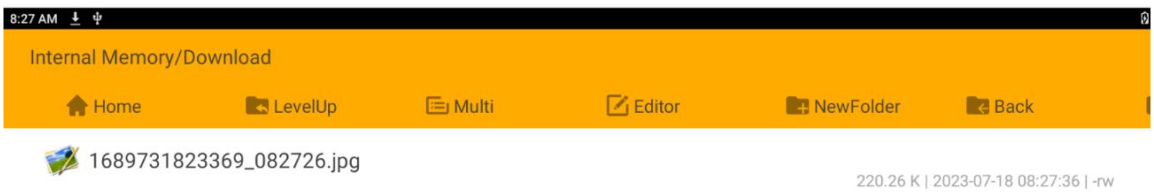
7) After the pairing is completed, you can see the paired Bluetooth device as shown in the figure below



8) At this time, you can use the Bluetooth of your mobile phone to send a picture to the development board. After sending, you can see the following confirmation interface in the Android system of the development board, and then click **Accept** to start receiving the picture sent by the mobile phone.



9) You can open the **Download** directory in the file manager to view the pictures received by the Android system Bluetooth of the development board



6. 6. How to use Raspberry Pi 5-inch screen

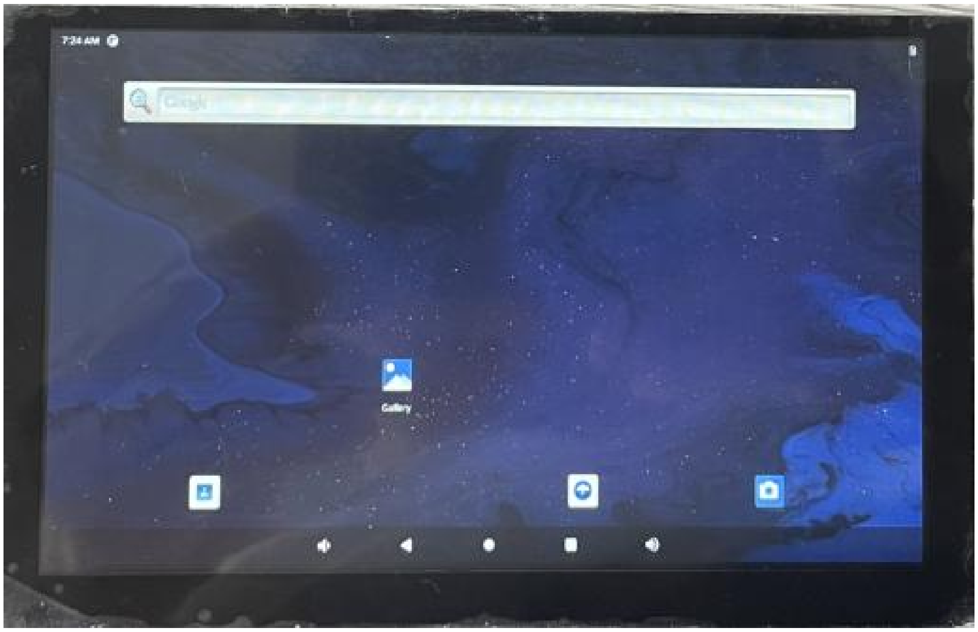
Please make sure that the image used is the following two versions of the image:
OrangePi3B_RK3566_Android11_lcd_v1.x.x.img
OrangePi3B_RK3566_Android11_spi-nvme_lcd_v1.x.x.img

1) The screen needs to be assembled first, please refer to **the assembly method of the Raspberry Pi 5-inch screen**

2) 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 in the figure below

Both the display and touch of the Raspberry Pi 5-inch screen can be used. If you have problems with the screen test, please make sure that the screen you purchased is exactly the same as the screen that the Orange Pi is compatible with.

The Orange Pi compatible screen is described in the assembly method of the Raspberry Pi 5-inch screen.

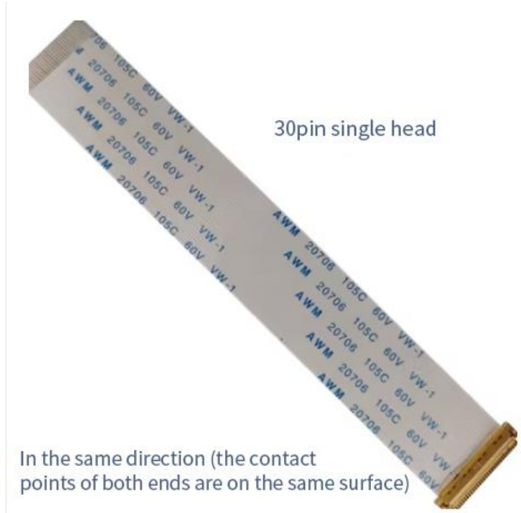


6. 7. How to use the eDP screen

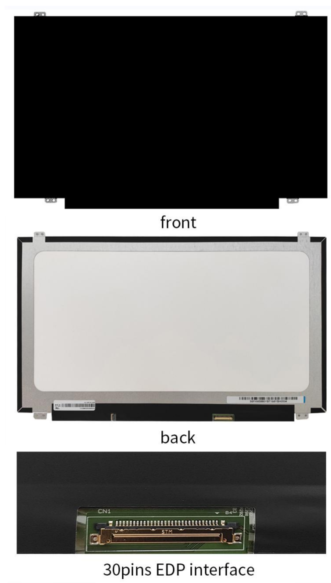
Please make sure that the image used is the following two versions of the image:
OrangePi3B_RK3566_Android11_lcd_v1.x.x.img
OrangePi3B_RK3566_Android11_spi-nvme_lcd_v1.x.x.img

The eDP screen has no touch function.

1) Currently only one eDP screen is compatible, including the following accessories:
   a. 0.5 pitch 30pin single-head cable in the same direction



b. 15.6-inch eDP display with a resolution of 1920x1080



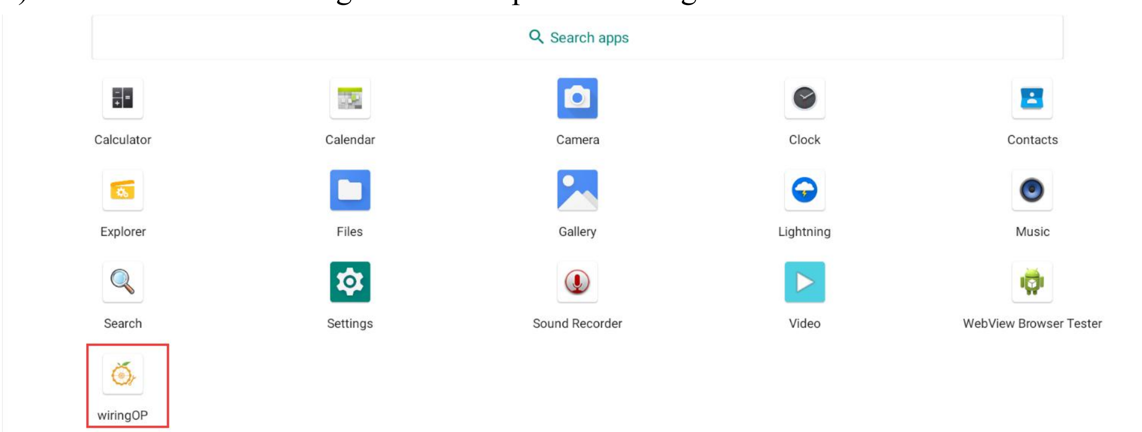
2) Connect the FPC end of the 30pin single-head codirectional cable to the eDP interface of the development board, and connect the other end to the eDP interface of the screen

3) Then 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 in the figure below

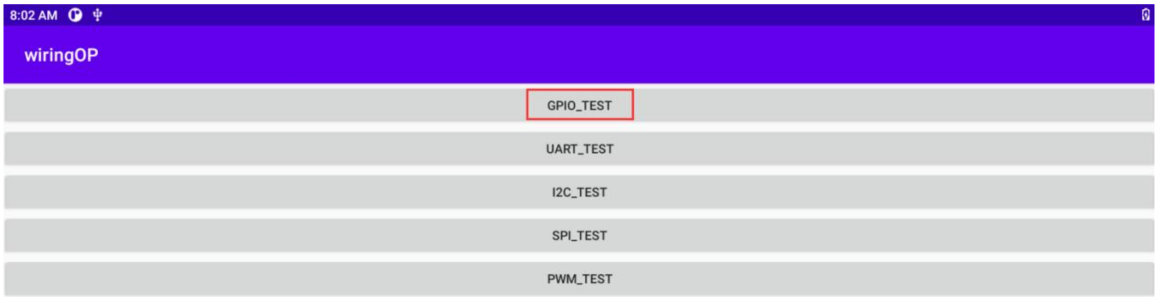
6. 8. 40pin interface GPIO, UART, SPI and PWM test

6. 8. 1. 40pin GPIO port test

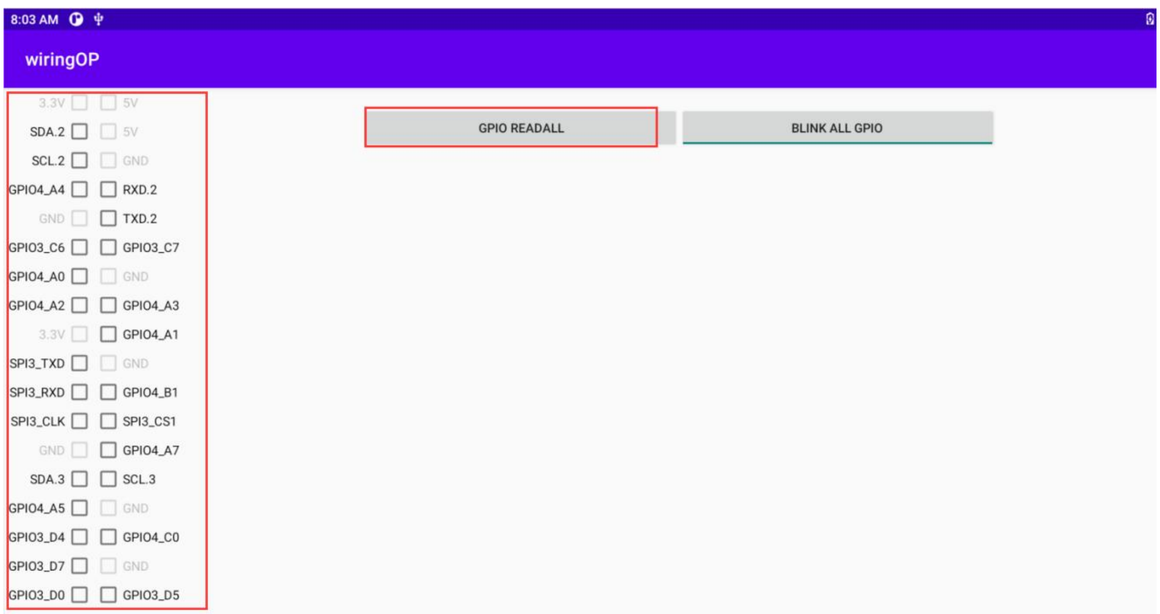
1) First click on the wiringOP icon to open the wiringOP APP



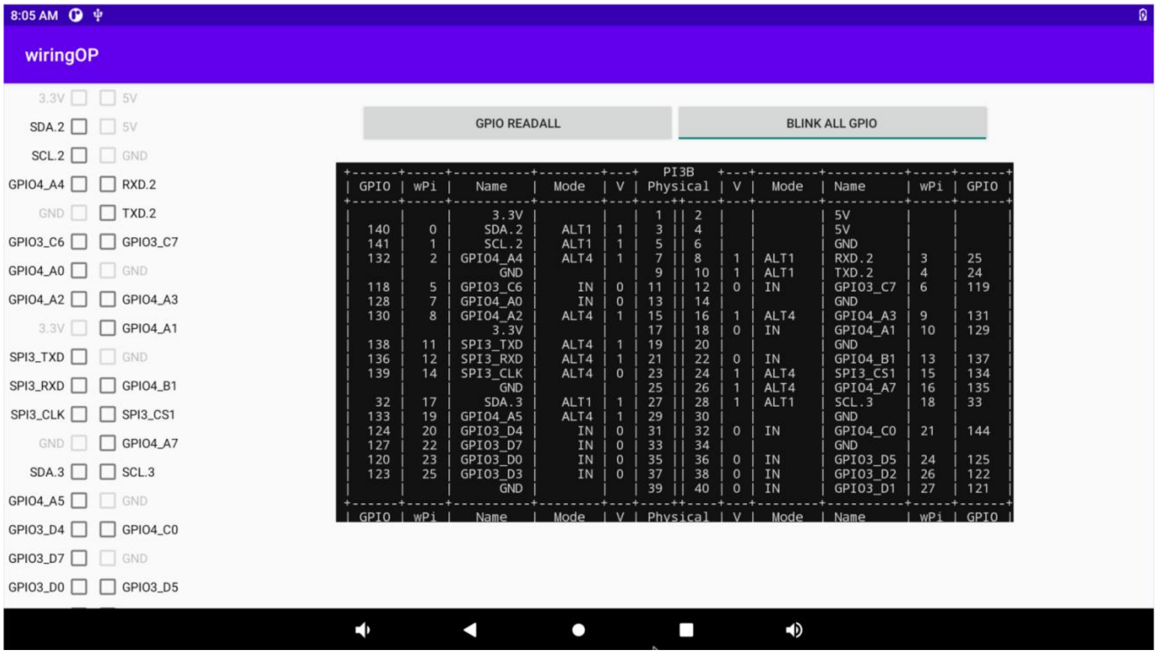
2) The main interface of wiringOP APP is displayed as shown in the figure below, and then click the **GPIO_TEST** button to open the GPIO test interface



3) The GPIO test interface is shown in the figure below. The two rows of **CheckBox** buttons on the left are in one-to-one correspondence with the 40pin pins. When the **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT** mode, and the pin level will be set to high level; when the checkbox is unchecked, the GPIO pin level will be set to low level; When the **GPIO READALL** button is pressed, information such as wPi number, GPIO mode, and pin level can be obtained; when the **BLINK ALL GPIO** button is clicked, the program will control the 28 GPIO ports to continuously switch between high and low levels

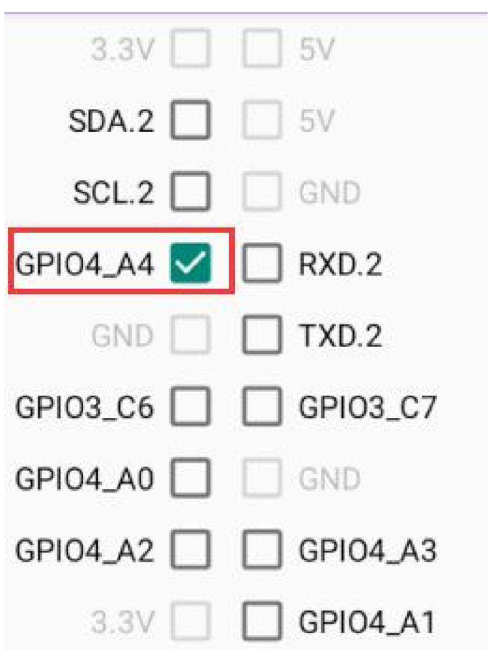


4) Then click the **GPIO READALL** button, the output information is as shown in the figure below:

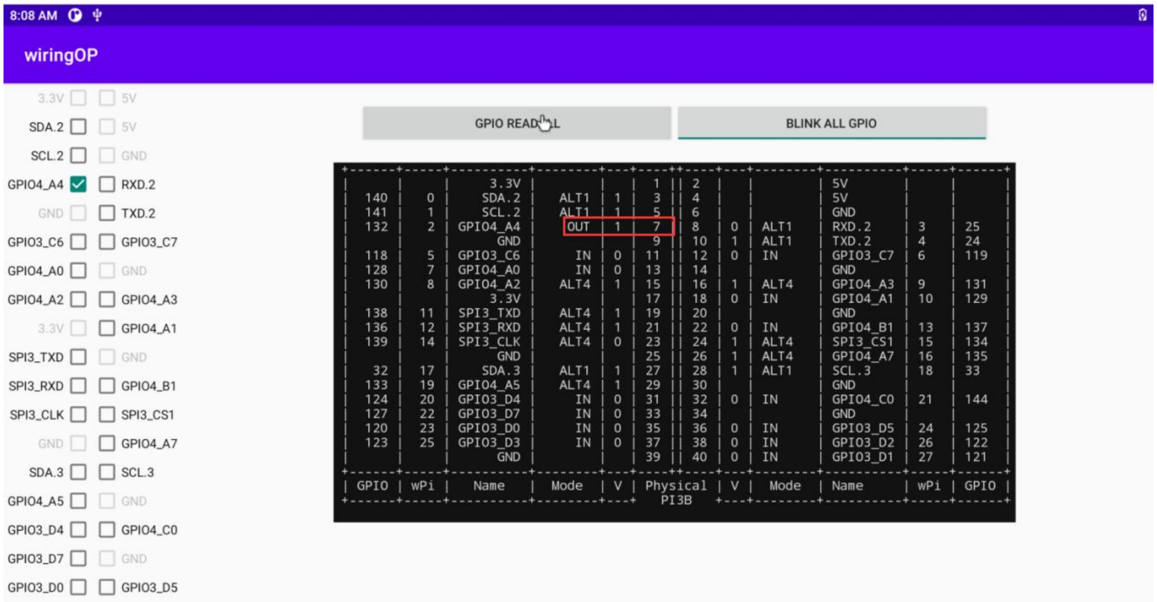


GPIO	wPi	Name	Mode	V	Physical		V	Mode	Name	wPi	GPIO
		3.3V			1	2			5V		
140	0	SDA.2	ALT1	1	3	4			5V		
141	1	SCL.2	ALT1	1	5	6			GND		
132	2	GPIO4_A4	ALT4	1	7	8	1	ALT1	RXD.2	3	25
		GND			9	10	1	ALT1	TXD.2	4	24
118	5	GPIO3_C6	IN	0	11	12	0	IN	GPIO3_C7	6	119
128	7	GPIO4_A0	IN	0	13	14			GND		
130	8	GPIO4_A2	ALT4	1	15	16	1	ALT4	GPIO4_A3	9	131
		3.3V			17	18	0	IN	GPIO4_A1	10	129
138	11	SPI3_TXD	ALT4	1	19	20			GND		
136	12	SPI3_RXD	ALT4	1	21	22	0	IN	GPIO4_B1	13	137
139	14	SPI3_CLK	ALT4	0	23	24	1	ALT4	SPI3_CS1	15	134
		GND			25	26	1	ALT4	GPIO4_A7	16	135
32	17	SDA.3	ALT1	1	27	28	1	ALT1	SCL.3	18	33
133	19	GPIO4_A5	ALT4	1	29	30			GND		
124	20	GPIO3_D4	IN	0	31	32	0	IN	GPIO4_C0	21	144
127	22	GPIO3_D7	IN	0	33	34			GND		
120	23	GPIO3_D0	IN	0	35	36	0	IN	GPIO3_D5	24	125
123	25	GPIO3_D3	IN	0	37	38	0	IN	GPIO3_D2	26	122
		GND			39	40	0	IN	GPIO3_D1	27	121
GPIO	wPi	Name	Mode	V	Physical		V	Mode	Name	wPi	GPIO

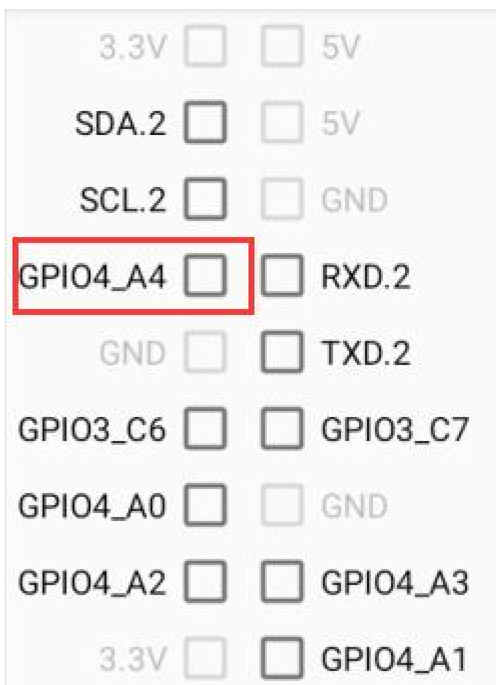
5) There are a total of 28 GPIO ports in the 40pins of the development board that can be used. The following uses pin 7 — the corresponding GPIO is GPIO4_A4 — the corresponding wPi serial number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port. First click the **CheckBox** button corresponding to pin 7. When the button is selected, pin 7 will be set to high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means setting high level success



6) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is **OUT**, and the pin level is high

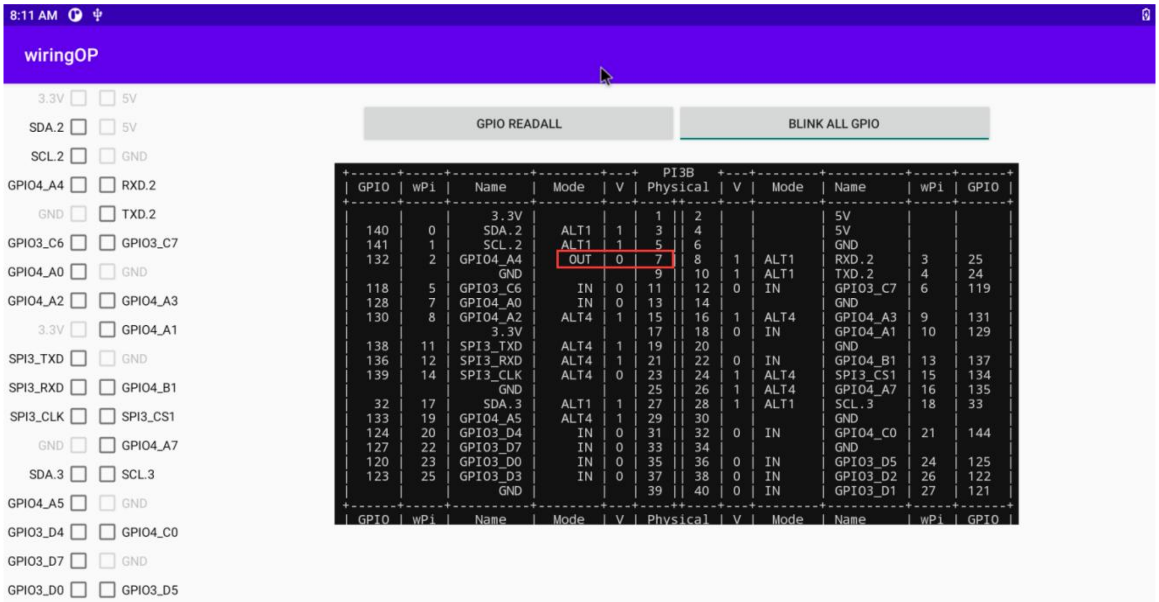


7) Click the **CheckBox** button in the figure below again to cancel the check status. Pin 7 will be set to low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level is set successfully.



8) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is

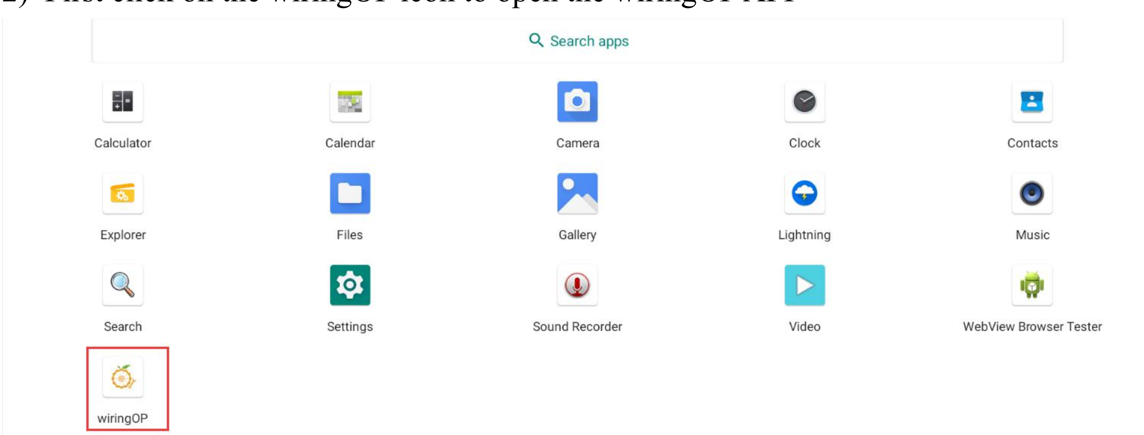
OUT, and the pin level is low



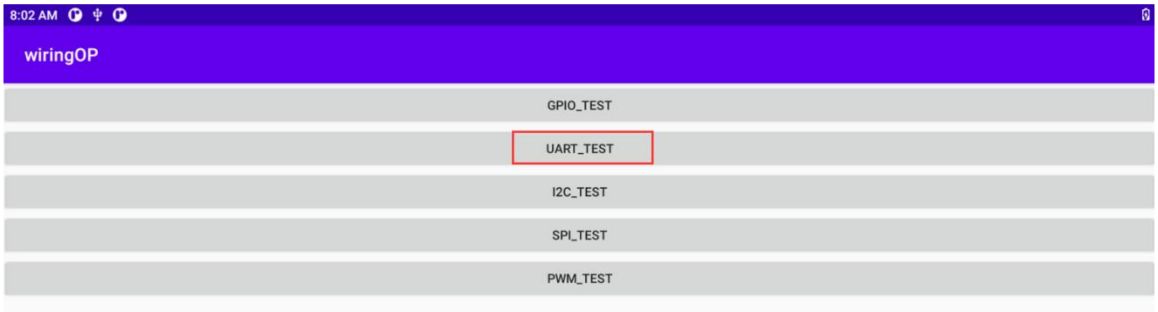
6. 8. 2. 40pin UART test

1) UART7 and UART9 are enabled by default in Android. The position of the 40pin is shown in the figure below, and the corresponding device nodes are **/dev/ttyS7** and **/dev/ttyS9** respectively

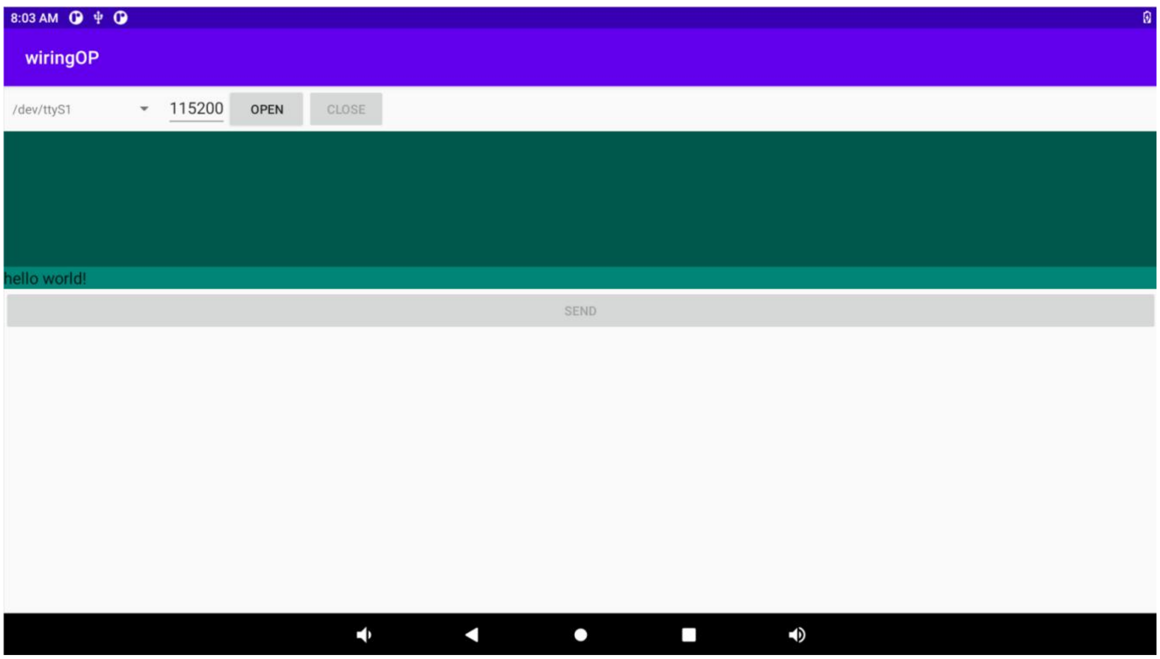
2) First click on the wiringOP icon to open the wiringOP APP



3) The main interface of wiringOP APP is displayed as shown in the figure below, and then click the **UART_TEST** button to open the UART test interface



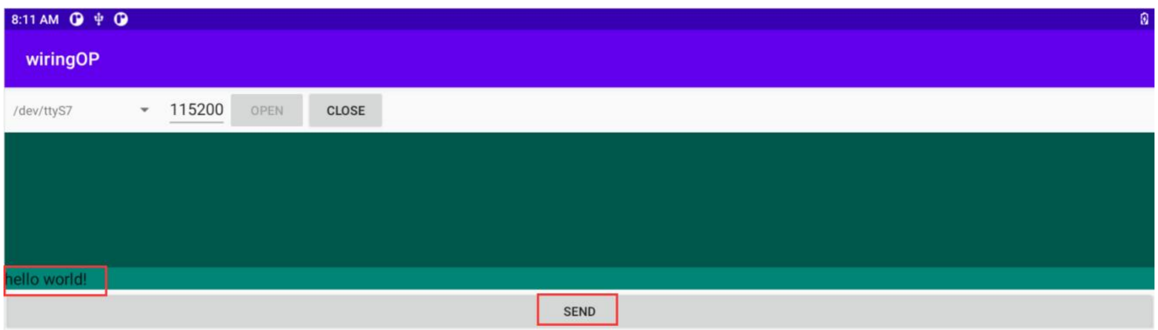
4) The serial port test interface of the APP is shown in the figure below



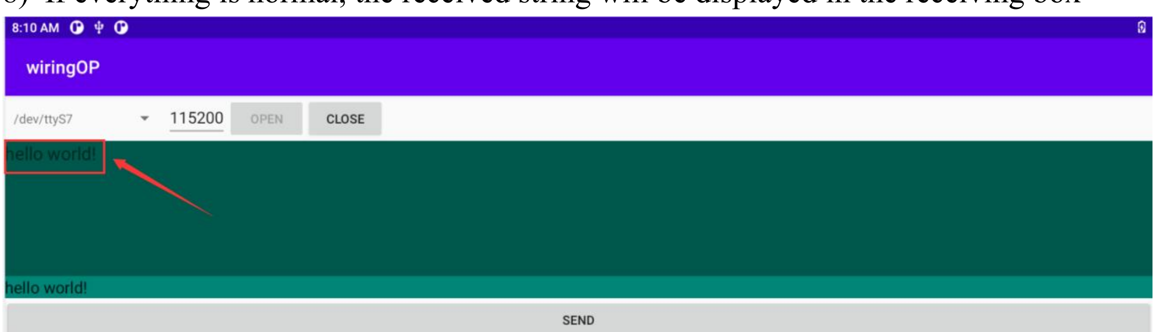
5) Take the test of **UART7** as an example below, select the **/dev/ttyS7** node in the selection box, enter the baud rate you want to set in the edit box, and then click the **OPEN** button to open the /dev/ttyS7 node. After the opening is successful, the **OPEN** button becomes unselectable, and the **CLOSE** button and **SEND** button become selectable

6) Then use Dupont wire to short the RXD and TXD pins of uart7

7) Then you can enter a character in the send edit box below, and click the **SEND** button to start sending

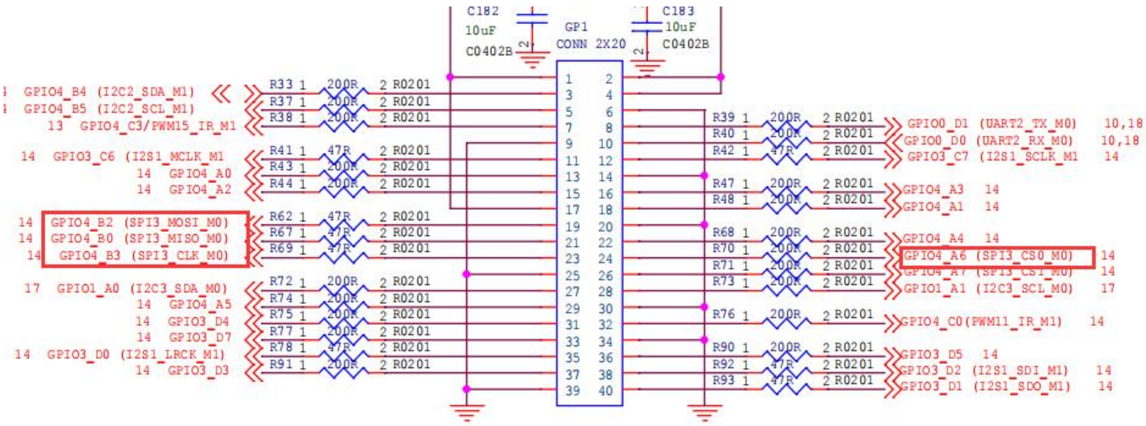


8) If everything is normal, the received string will be displayed in the receiving box

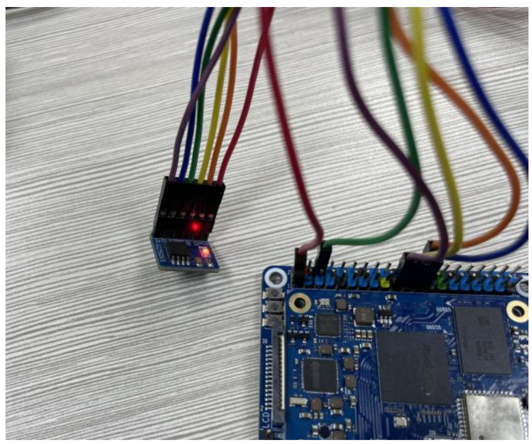


6. 8. 3. 40pin SPI test

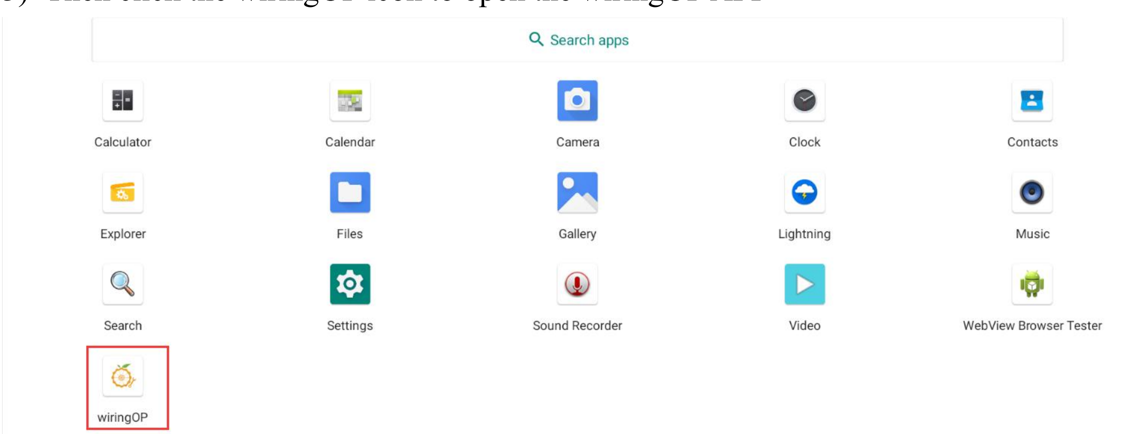
1) According to the schematic diagram of the 40pin interface, the spi available for Orange Pi 3B is spi3



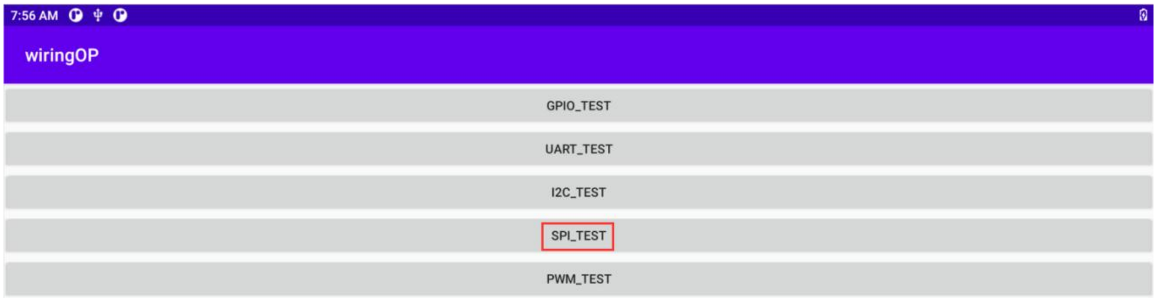
2) Here, the SPI interface is tested through the w25q64 module. First, the w25q64 device is connected to the SPI3 interface



3) Then click the wiringOP icon to open the wiringOP APP

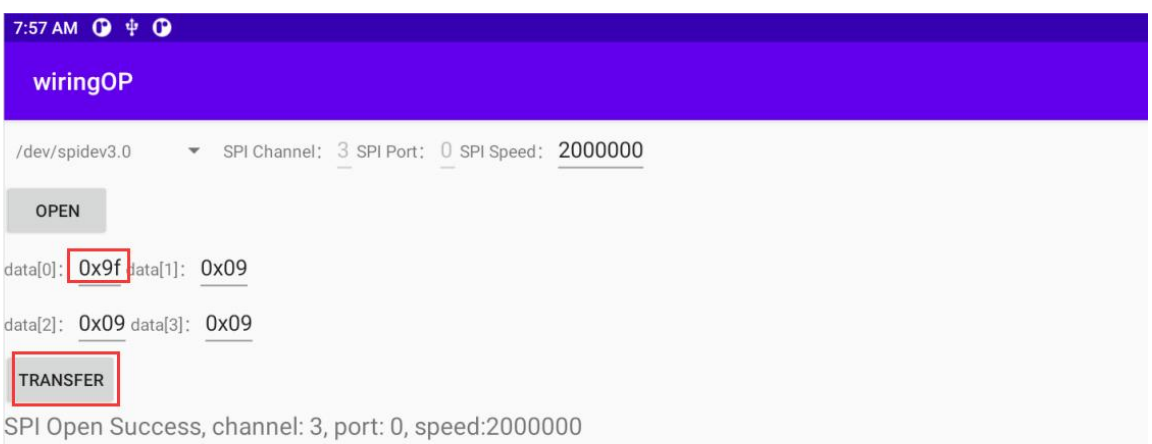


4) The main interface of wiringOP APP is displayed as shown in the figure below, click the SPI_TEST button to open the SPI test interface

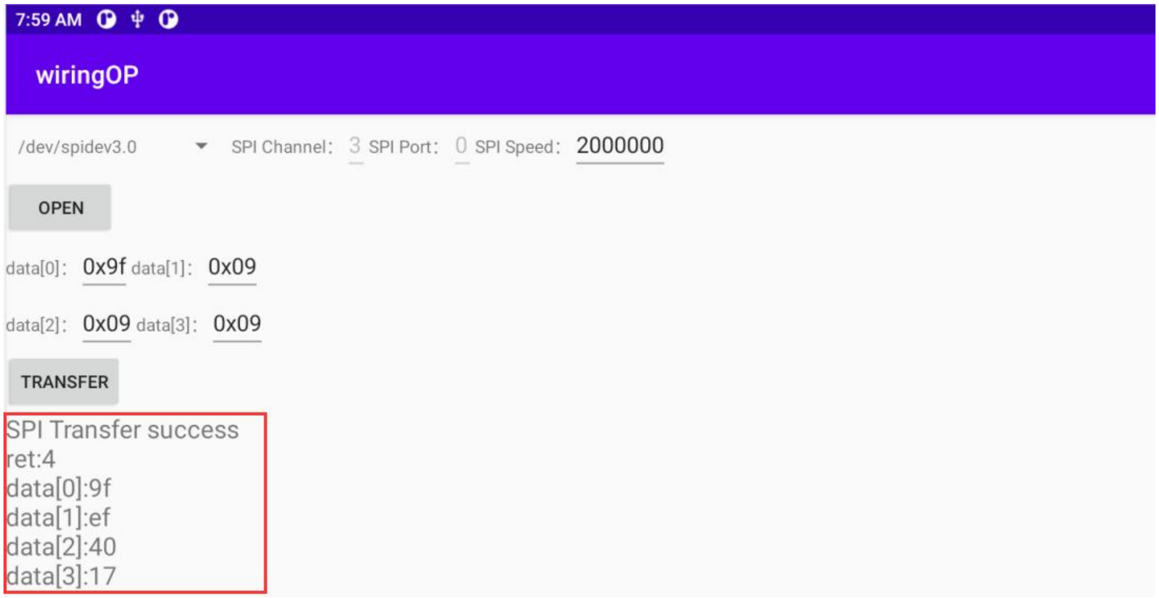


5) Then click the **OPEN** button to initialize the SPI

6) Then fill in the bytes that need to be sent, such as reading the ID information of w25q64, fill in the address 0x9f in data[0], and then click the **TRANSFER** button



7) Finally, the APP will display the read ID information



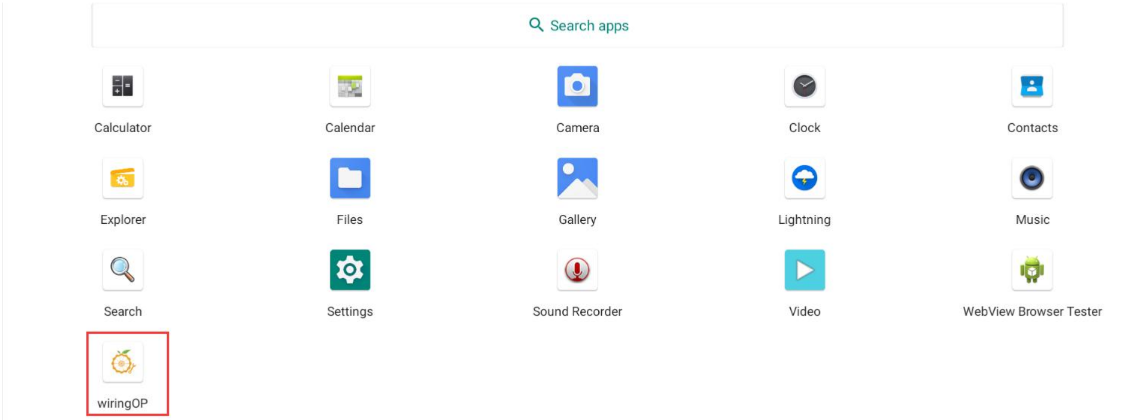
8) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, corresponding to the value read above (h stands for hexadecimal)

MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

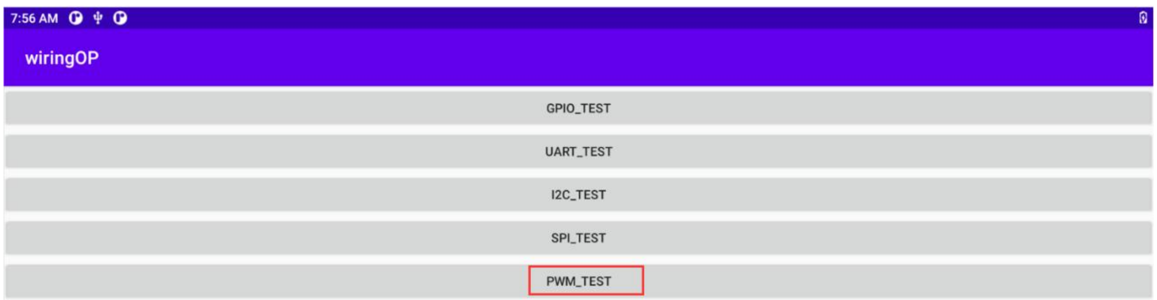
6. 8. 4. 40pin PWM test

1) Android enables **PWM11** by default, and the corresponding pin is located at 40pin as shown in the figure below

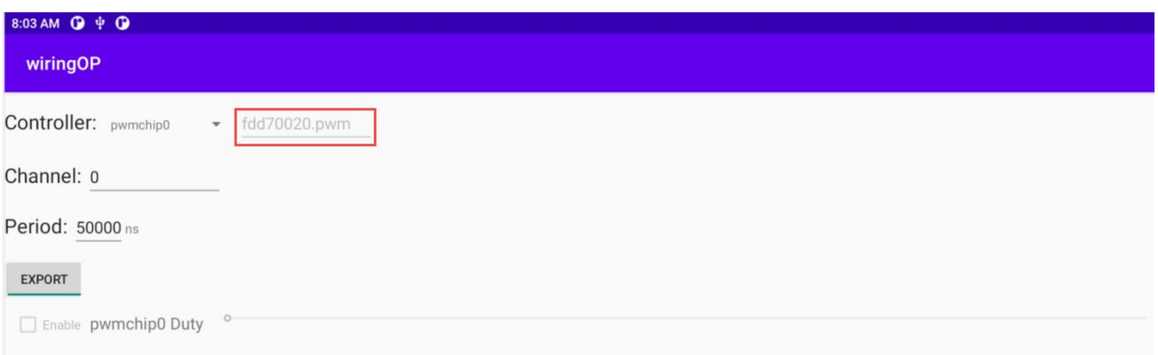
2) First click on the wiringOP icon to open the wiringOP APP



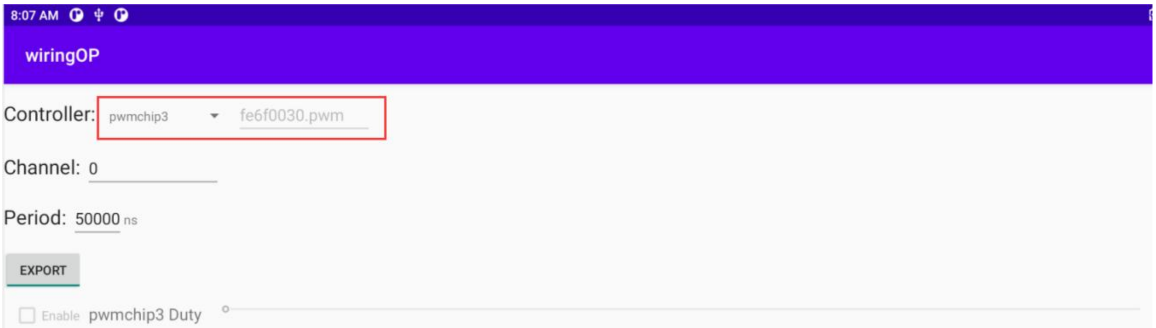
3) Then click the **PWM_TEST** button on the main interface of wiringOP to enter the PWM test interface



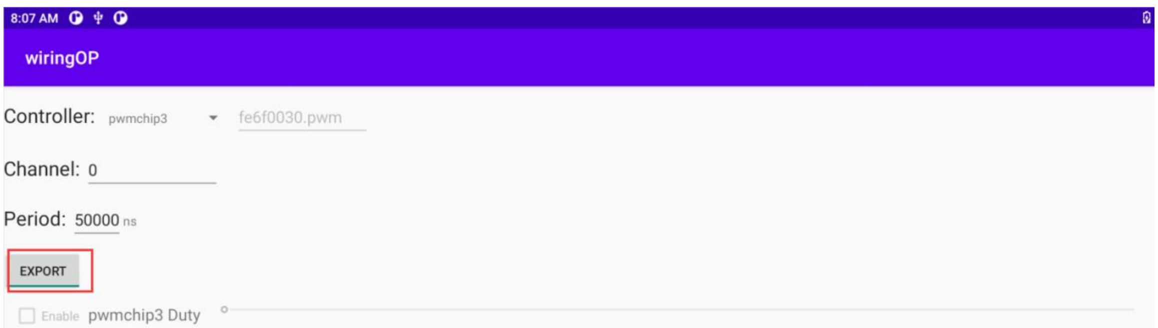
4) The base address corresponding to PWM11 is **fe6f0030**, here pwmchip0 shows **fdd70020.pwm** on the right, then you need to click the drop-down option to select other pwmchips until **fe6f0030.pwm** is displayed on the right



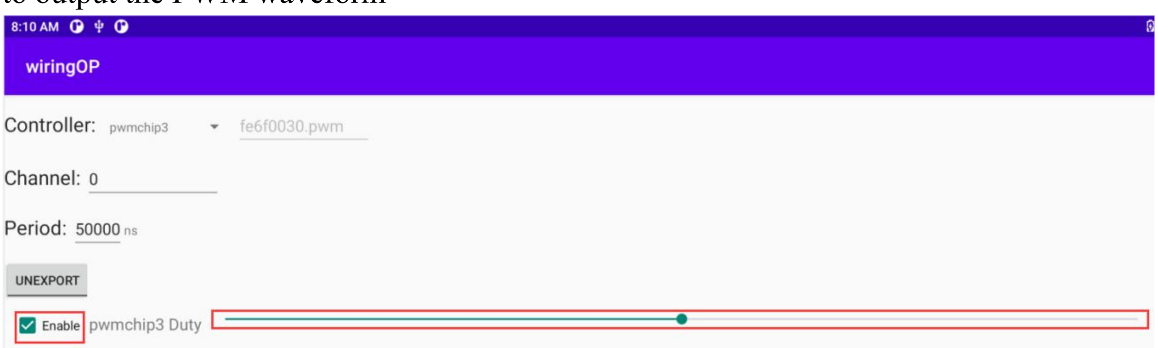
5) When the drop-down option selects **pwmchip3**, the corresponding base address of PWM11 is **fe6f0030** on the right



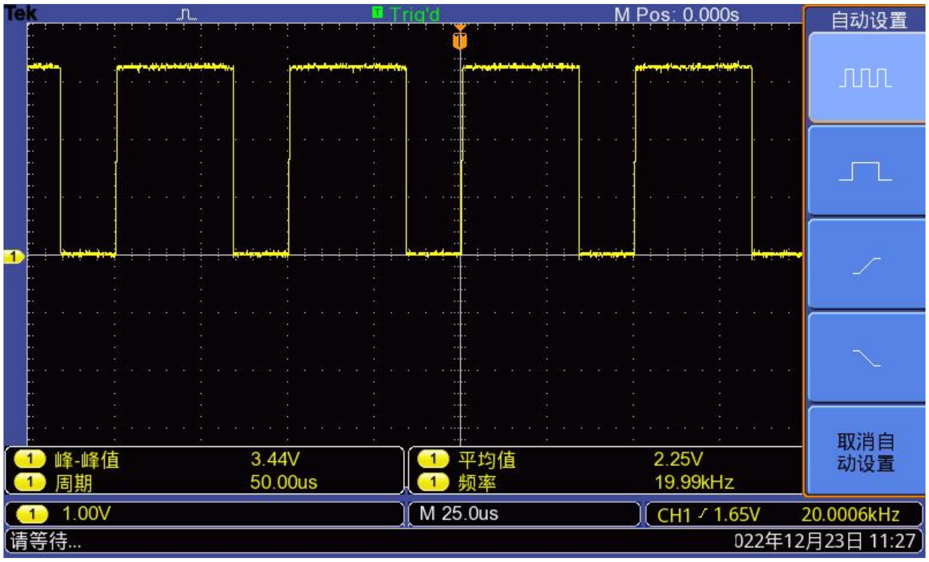
6) Then confirm the PWM channel, the default is channel 0, and confirm the PWM cycle, the default configuration is **50000ns**, converted to PWM frequency is **20KHz**, you can modify it yourself, click the **EXPORT** button to export **PWM11**



7) Then drag the drag bar below to change the PWM duty cycle, and then check Enable to output the PWM waveform



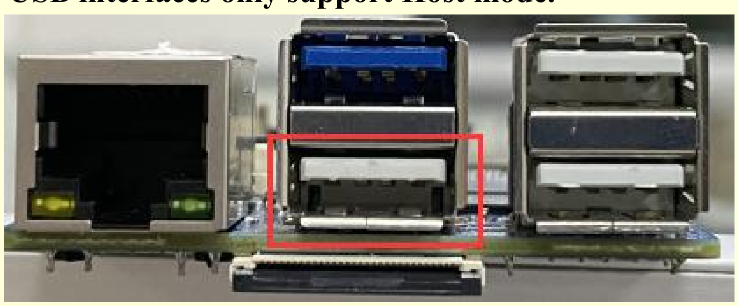
8) Then use an oscilloscope to measure the No. 32 pin in the 40pin of the development board, and you can see the following waveform



6. 9. How to use ADB

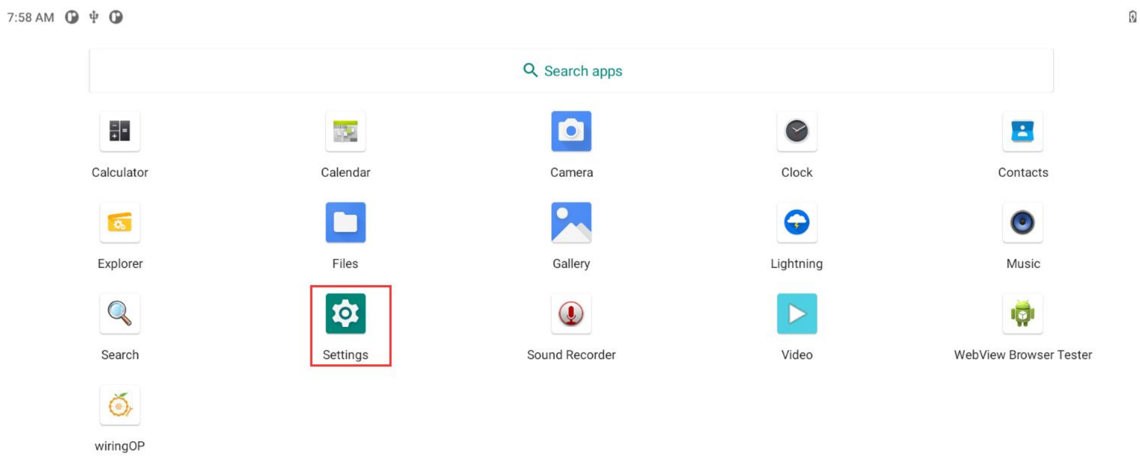
6. 9. 1. The method of USB OTG mode switching

The development board has 4 USB interfaces, among which the USB interface marked in red box in the figure below can support both Host mode and Device mode, and the other 3 USB interfaces only support Host mode.

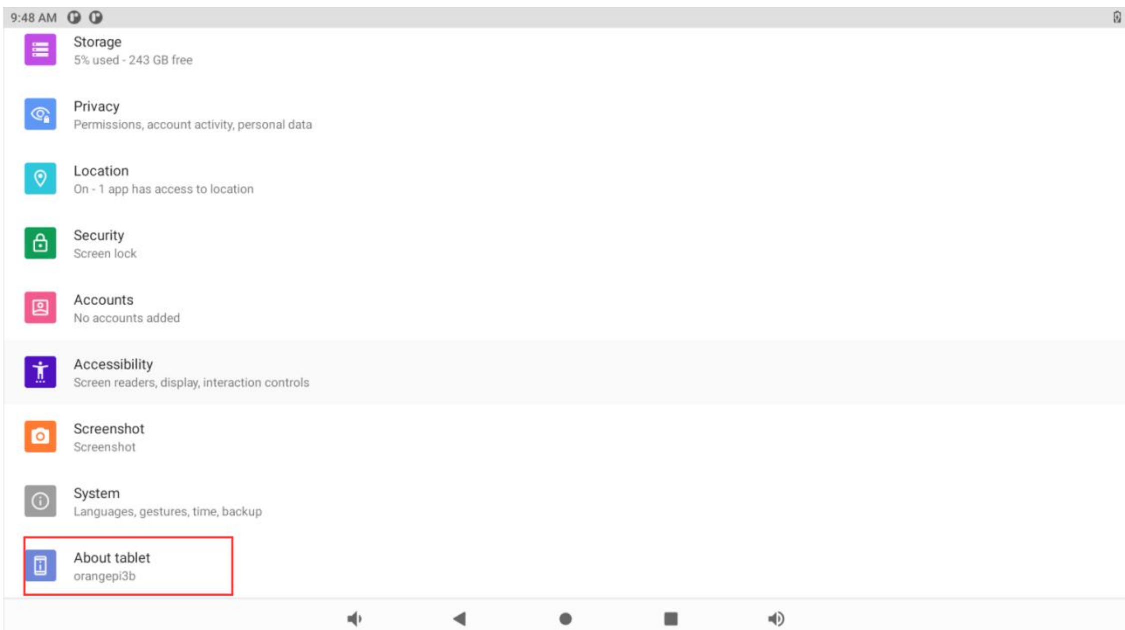


The USB OTG interface defaults to Host mode, which can be used to connect USB devices such as mouse and keyboard. If you want to use ADB, you need to manually switch to Device mode.

1) First open Settings



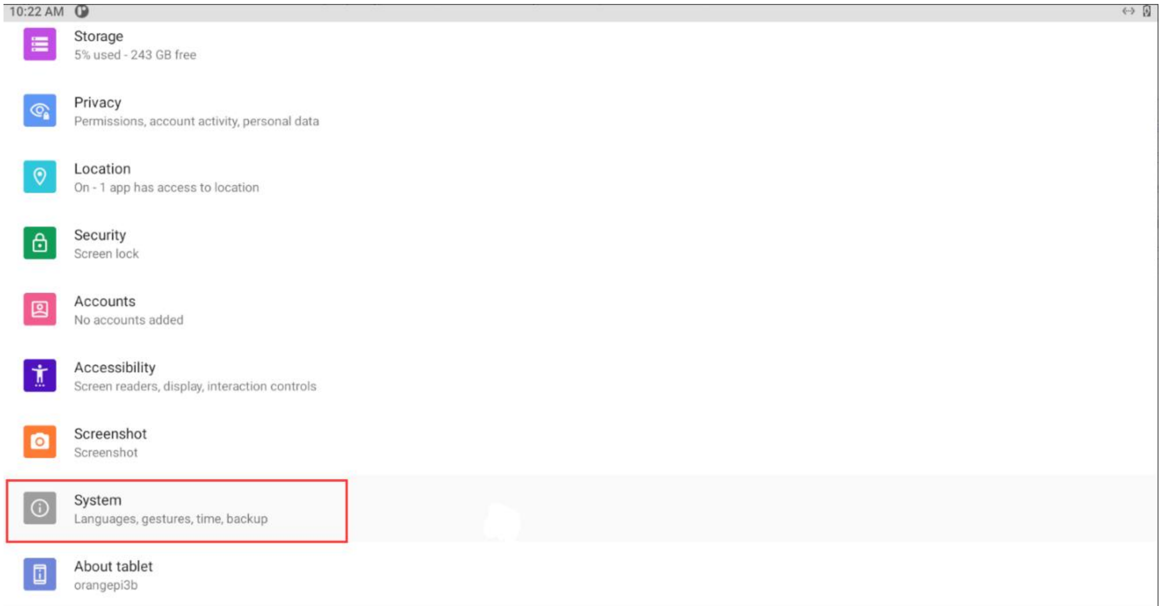
2) Then select **About tablet**



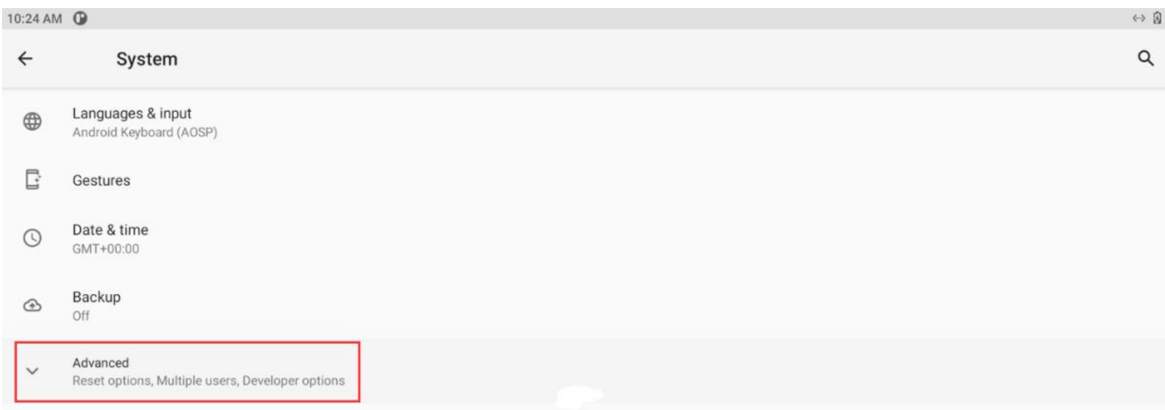
3) Then click the **Build number** menu bar several times with the mouse until the prompt **You are now a developer!** appears

4) Then click to return to the previous menu

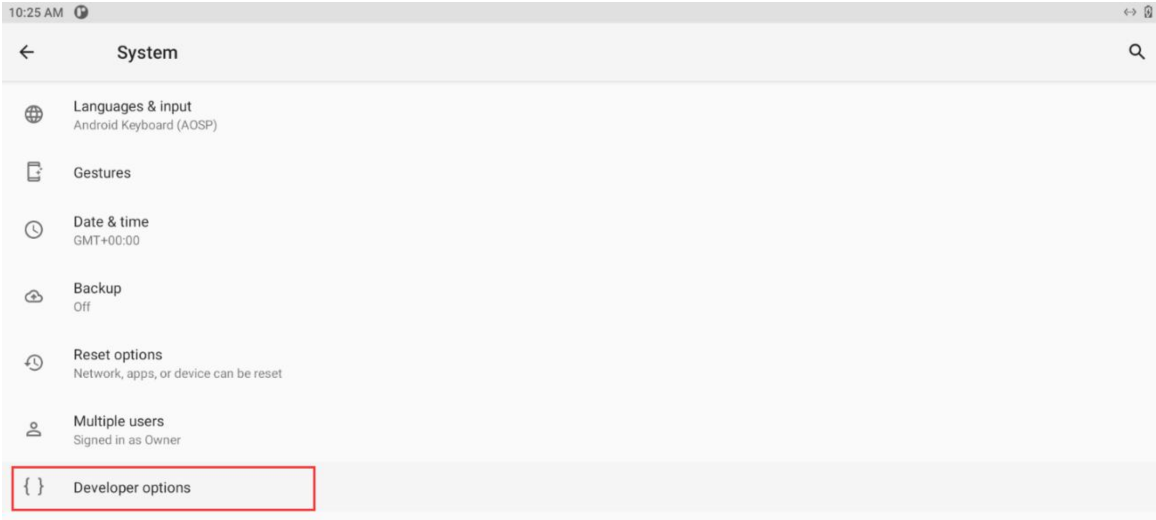
5) Then select **System**



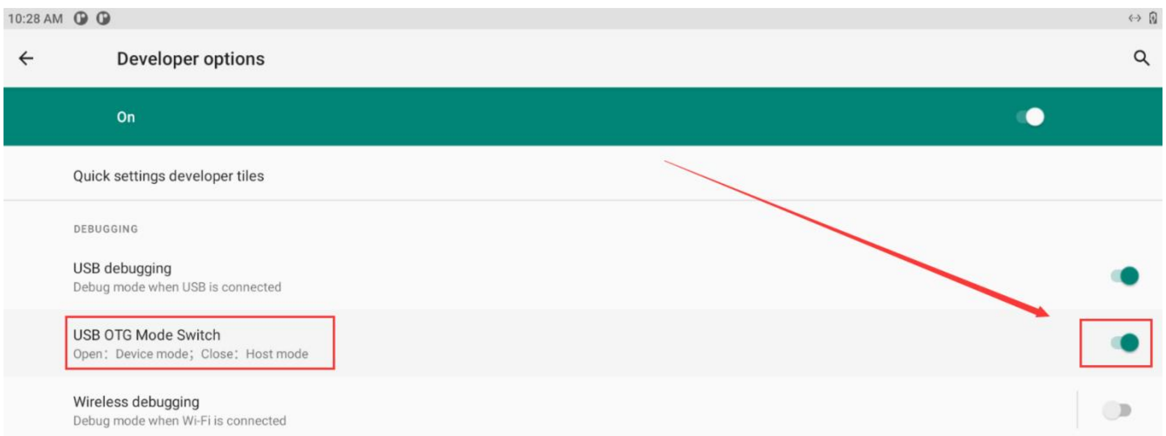
6) Then select **Advanced**



7) Then select **Developer options** in the expanded column



8) Finally find the **USB OTG Mode Switch** switch, **turn on the switch to switch to Device mode, turn off the switch to switch to Host mode**

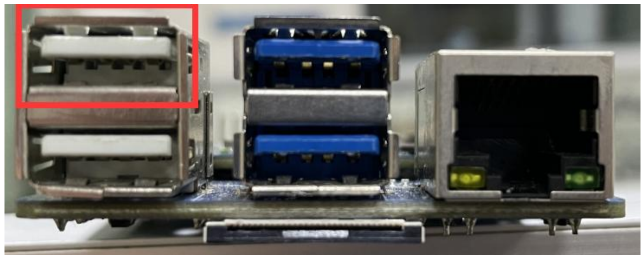


6. 9. 2. Use the data cable to connect to adb debugging

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

2) Then refer to **the method of USB OTG mode switching** to switch USB OTG to Device mode

3) Then connect the development board to the Ubuntu PC through the USB2.0 male-to-male data cable. The position of the USB OTG interface on the development board is shown in the figure below:



4) Then install the adb tool on the Ubuntu PC

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

5) You can view the identified ADB devices through the following command

```
test@test:~$ adb devices
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

6) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell
console:/ $
```

7) Execute the following 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/
```

6. 9. 3. Use network connection adb debugging

Using the network adb does not require a data cable to connect the computer and the development board, but to communicate through the network, so first make sure that the wired or wireless network of the development board is connected, and

then obtain the IP address of the development board, which will be used later.

1) Make sure that the **service.adb.tcp.port** of the Android system is set to port number 5555

```
console:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the port number of network adb

```
console:/ # setprop service.adb.tcp.port 5555
console:/ # stop adbd
console:/ # start adbd
```

3) Install adb tool on Ubuntu PC

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

4) Then connect to the network adb on the Ubuntu PC

```
test@test:~$ adb connect 192.168.1.xxx     (IP 地址需要修改为开发板的 IP 地址)
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xxx:5555

test@test:~$ adb devices
List of devices attached
192.168.1.xxx:5555          device
```

5) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell
console:/ #
```

7. How to compile Android11 source code

7. 1. Download the source code of Android 11

1) First download the Android 11 source code sub-volume compressed package from the Google network disk

a. Google Drive

2) After downloading the sub-volume compression package of the Android 11 source code, please check whether the MD5 checksum is correct, if not, please download the source code again

```
test@test:~$ md5sum -c RK356X_Android11.tar.gz.md5sum
RK356X_Android11.tar.gz00: OK
RK356X_Android11.tar.gz01: OK
RK356X_Android11.tar.gz02: OK
RK356X_Android11.tar.gz03: OK
RK356X_Android11.tar.gz04: OK
RK356X_Android11.tar.gz05: OK
RK356X_Android11.tar.gz06: OK
```

3) Then you need to merge multiple compressed files for decompression

```
test@test:~$ cat RK356X_Android11.tar.gz0* | tar -xvzf -
```

7. 2. Compile the source code of Android 11

1) First install the software packages required to compile the Android11 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 libncurses5 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip liblz4-tool
```

2) There is a build.sh compilation script in the source code, and the compilation parameters are as follows

a. **-U**：Compile uboot
b. **-K**：Compile kernel
c. **-A**：compile android
d. **-u**：Package and generate update.img and update_spi_nvme.img
e. **-o**：Compile OTA package
f. **-d**：Specify kernel dts

3) Compile uboot, kernel, android and package them into update.img

a. The command to compile and support HDMI 4K display mirroring (LCD is turned off by default) is as follows:

```
test@test:~$ cd RK356X_Android11
test@test:~/ RK356X_Android11$ export BOARD=orangepi3b
test@test:~/ RK356X_Android11$ source build/envsetup.sh
test@test:~/ RK356X_Android11$ lunch rk3566_r-userdebug
test@test:~/ RK356X_Android11$ ./build.sh -AUKu
```

b. The command to compile and support LCD display mirroring (HDMI is disabled by default) is as follows:

```
test@test:~$ cd RK356X_Android11
test@test:~/ RK356X_Android11$ export BOARD=orangepi3b
test@test:~/ RK356X_Android11$ export DUAL_LCD=true
test@test:~/ RK356X_Android11$ source build/envsetup.sh
test@test:~/ RK356X_Android11$ lunch rk3566_r-userdebug
test@test:~/ RK356X_Android11$ ./build.sh -AUKu
```

4) After the compilation is complete, the following information will be printed

```
********rkImageMaker ver 2.1********
Generating new image, please wait...
storage is spinor
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
********rkImageMaker ver 2.1********
Merging storage firmware, please wait...
storage count = 2
adding spinor_update.img...ok
adding pcie_update.img...ok
Merging firmware success.
Making update_spi_nvme.img OK.
Make update image ok!
/wspace3/RK3566/RK356X_Android11
```

5) The final image file will be placed in the **rockdev/Image-rk3566_r/** directory. Among them, **update.img** is the boot image that supports TF card and eMMC, and **update_spi_nvme.img** is the boot image of NVME SSD

```
test@test:~/RK356X_Android11$ cd rockdev/Image-rk3566_r
test@test:~/RK356X_Android11/rockdev/Image-rk3566_r $ ls update*
update.img update_spi_nvme.img
```

8. Appendix

8. 1. User Manual Update History

Version	Date	Update Notes
v1.0	2023-08-29	initial version
v1.1	2023-09-05	Instructions for using the Orange Pi OS Arch system
v1.2	2023-09-21	Linux: How to create a WIFI hotspot through create_ap

8. 2. Image Update History

Date	Update Notes
2023-08-29	Orangepicm4_1.0.0_ubuntu_focal_server_linux5.10.160.7z Orangepicm4_1.0.0_ubuntu_jammy_server_linux5.10.160.7z Orangepicm4_1.0.0_debian_bullseye_server_linux5.10.160.7z Orangepicm4_1.0.0_debian_bookworm_server_linux5.10.160.7z Orangepicm4_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepicm4_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepicm4_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepicm4_1.0.0_debian_bookworm_desktop_xfce_linux5.10.160.7z OrangePiCM4_RK3566_Android11_v1.0.0.tar.gz OrangePiCM4_RK3566_Android11_lcd_v1.0.0.tar.gz OrangePiCM4_RK3566_Android11_spi-nvme_v1.0.0.tar.gz OrangePiCM4_RK3566_Android11_lcd_spi-nvme_v1.0.0.tar.gz * initial version
2023-09-05	Opios-arch-aarch64-xfce-opicm4-23.09-linux5.10.160.img.xz * initial version
2023-09-21	Orangepicm4_1.0.2_ubuntu_focal_server_linux5.10.160.7z Orangepicm4_1.0.2_ubuntu_jammy_server_linux5.10.160.7z Orangepicm4_1.0.2_debian_bullseye_server_linux5.10.160.7z

	Orangepicm4_1.0.2_debian_bookworm_server_linux5.10.160.7z Orangepicm4_1.0.2_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepicm4_1.0.2_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepicm4_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepicm4_1.0.2_debian_bookworm_desktop_xfce_linux5.10.160.7z * Solve the problem of CPU frequency being limited to 1.2GHz * Add rk356x-uart2-m0.dtbo