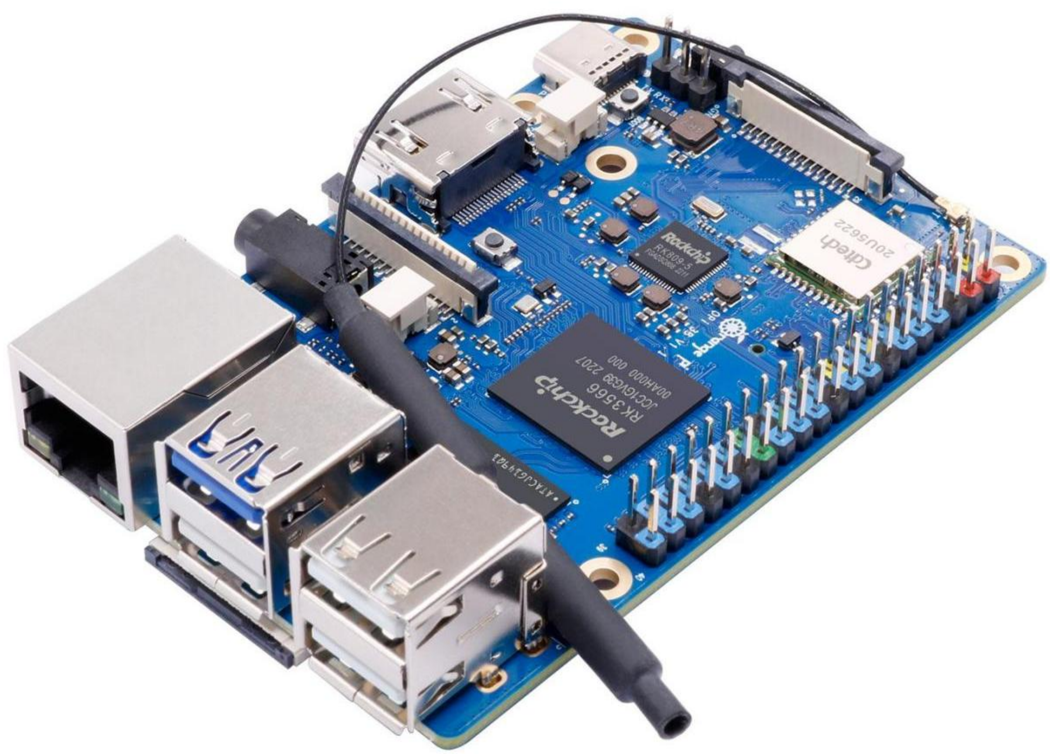
Orange Pi 3B
User Manual



Catalog

1. Basic features of Orange Pi 3B

1. 1. What is Orange Pi 3B

Orange Pi 3B adopts Rockchip RK3566 quad-core 64-bit Cortex-A55 processor, adopts 22nm process, the main frequency can reach up to 1.8GHz, integrates ARM Mali-G52 GPU, embedded high-performance 2D image acceleration module, built-in 1 The AI accelerator NPU of Tops computing power can choose 2GB, 4GB or 8GB memory, and has up to 4K display processing capability.

Orange Pi 3B brings out quite a lot of interfaces, including 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 3B supports Android11, Ubuntu22.04, Ubuntu20.04, Debian11, Debian12, OpenHarmony 4.0 Beta1, Orange Pi OS (Arch), Orange Pi OS (OH) based on OpenHarmony and other operating systems.

1. 2. Purpose of Orange Pi 3B

We can use it to achieve:

- A Linux desktop computer
- A Linux web 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 realize the delivery from idea to prototype to mass production. It is a maker, dreamer, hobby The ideal creative platform for readers.

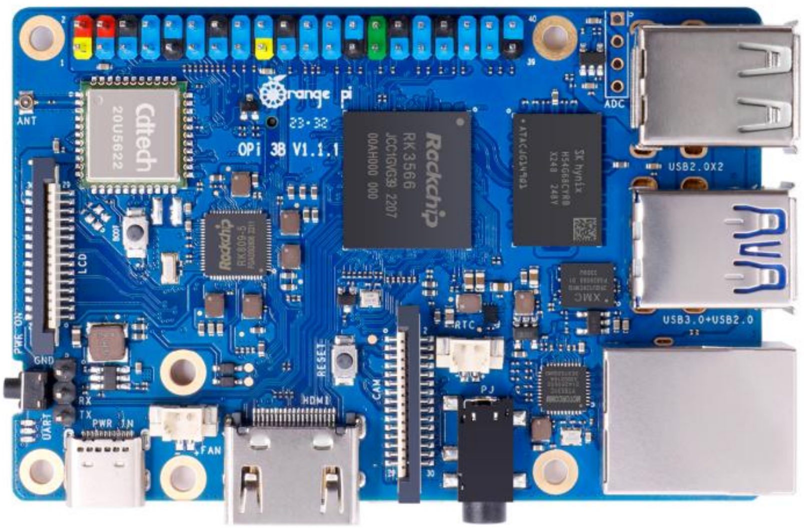
1.3. Hardware Features of Orange Pi 3B

Introduction to hardware features	
Master chip	Rockchip RK3566
CPU	Quad-core 64-bit Cortex-A55 processor, 22nm advanced technology, main frequency 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@100fps H.265 video encoding • **1080P@60fpsH.264** video encoding
PMU	Rockchip RK809-5
Memory	2GB/4GB/8GB （LPDDR4/4x）
storage	• Support eMMC module: 16GB/32GB/64GB/128GB/256GB • SPI Flash: 16MB/32MB • M.2 M-KEY slot: SATA3 or PCIe2.0 NVME SSD • TF card slot
Wi-Fi+Bluetooth	Wi-Fi 5+BT 5.0， BLE（20U5622）
ethernet transceiver	10/100/1000Mbps Ethernet (onboard PHY chip: YT8531C)
show	• 1x HDMI TX 2.0, maximum support 4K@60FPS • 1xMIPI DSI 2 Lane • eDP1.3

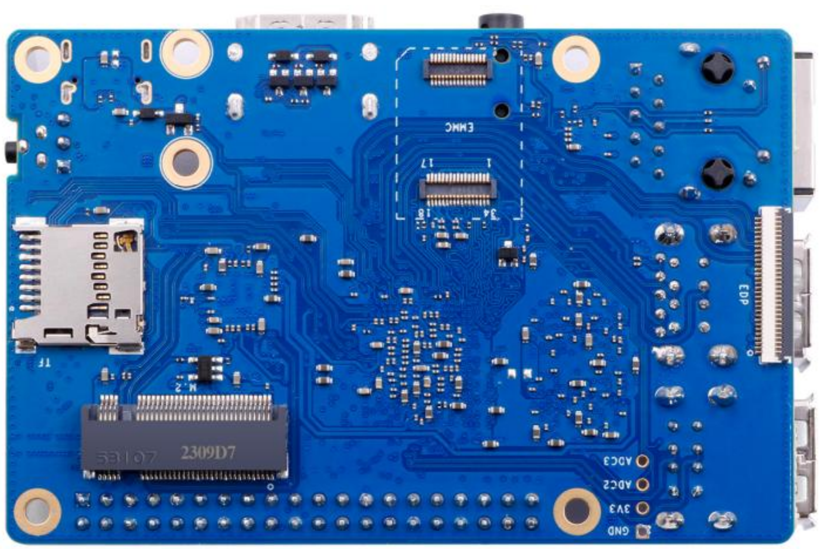
Camera	1xMIPI CSI 2 Lane camera interface
USB	• 1xUSB 2.0 supports Device or HOST mode • 1xUSB 3.0 HOST • 2xUSB 2.0 HOST
Audio	3.5mm headphone jack audio input/output
Button	1x MaskROM key, 1xRESET key, 1 xPOWER key
FAN	2Pin 1.25mm 5V fan interface
RTC	2Pin 1.25mm backup battery interface
40Pin	40Pin function expansion interface, supports the following interface types: GPIO, UART, I2C, SPI, PWM
Power Supply	Type-C 5V3A
Supported OS	Android11, Ubuntu22.04, Ubuntu20.04, Debian11, Debian12, OpenHarmony 4.0 Beta1, Orange Pi OS (Arch), Orange Pi OS (OH) based on OpenHarmony and other operating systems.
Introduction of Appearance Specifications	
PCB size	85mm x 56mm x 17mm
weight	49g
range Pi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd.	

1.4. Top view and bottom view of Orange Pi 3B

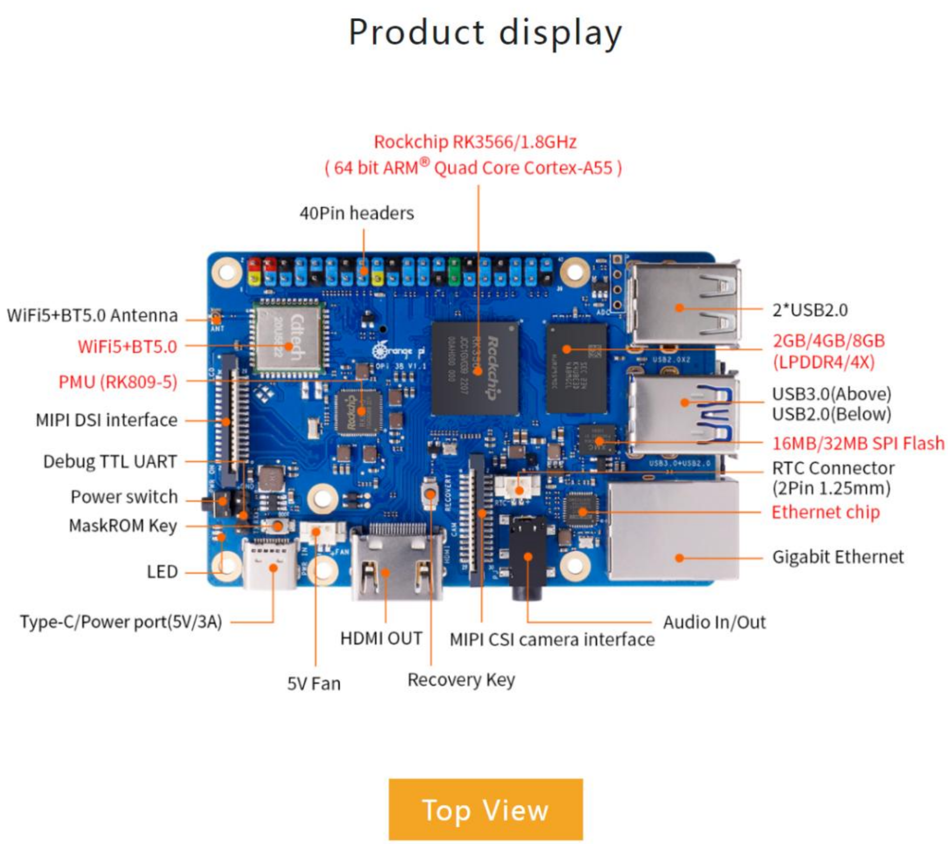
Top view:

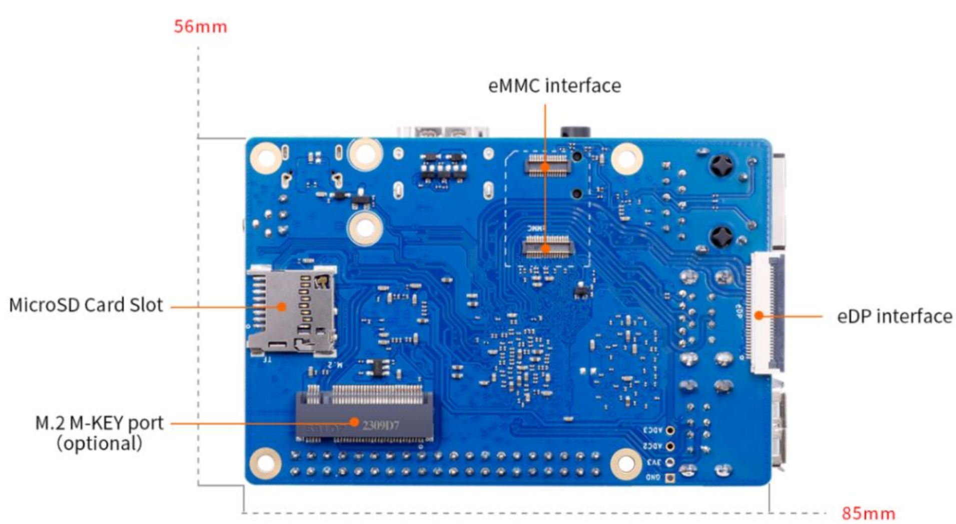


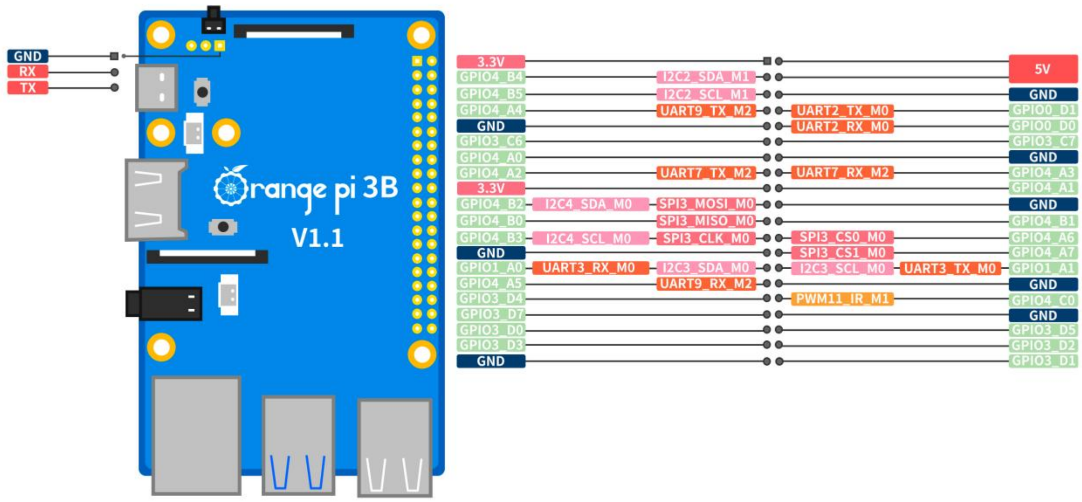
Bottom view:



1.5. Interface details of Orange Pi 3B







The diameter of the four positioning holes is 2.7mm, and the diameter of the M.2 PICE device fixing hole is 2.9mm.

2. Introduction to the use of the development board

2. 1. Prepare the required accessories

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

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

3) Display with HDMI interface

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

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

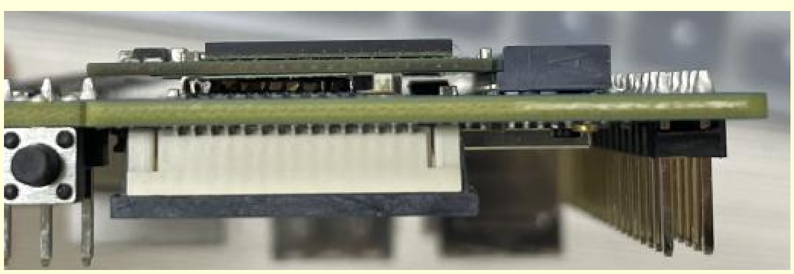
5) Power adapter, Orange Pi 3B is recommended to use 5V/3A Type-C power supply for power supply

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

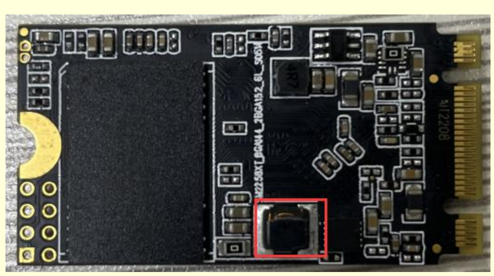
7) USB camera

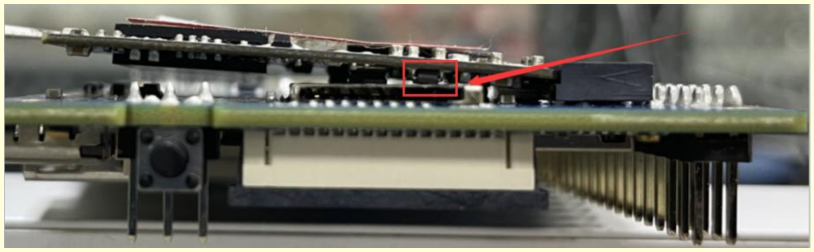
8) PCIe NVMe SSDs. The development board supports M.2 2230 and M.2 2242 SSDs, and M.2 2280 can also be supported, but it cannot be fixed with screws.

Because the PCIe seat of the development board is relatively low, and the TF card slot next to the seat also has a part protruding. When buying an SSD, you need to choose the one with a smooth back and no components as shown in the figure below, so that it will not withstand the card when installed on the board. groove.



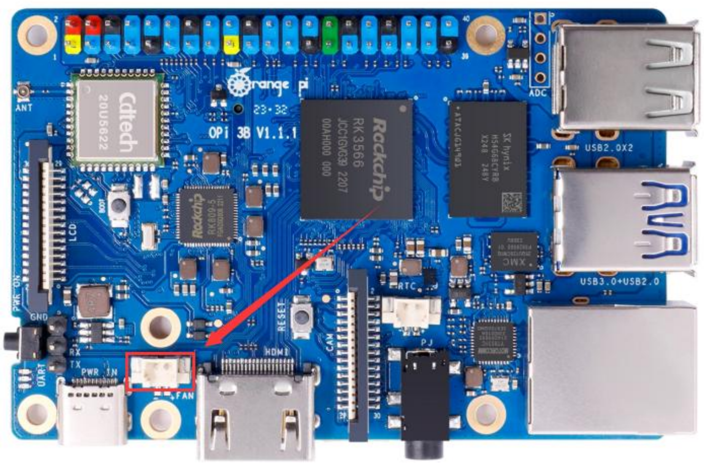
SSDs with components on the back that are not smooth may hold against the TF card slot during installation and cause the other end to tilt up, so they cannot be fixed with screws. For the SSD in the figure below, there is just a component protrusion that supports the card slot.





9) 5V cooling fan. As shown in the figure below, the development board has an interface for connecting the cooling fan, and the interface specification is **2pin 1.25mm pitch**

The fan on the development board can adjust the speed and switch through PWM.



10) 100M or 1000M network cable, used to connect the development board to the Internet

11) USB2.0 male-to-male data cable, used to burn images to eMMC, NVMe SSD and other functions

12) **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

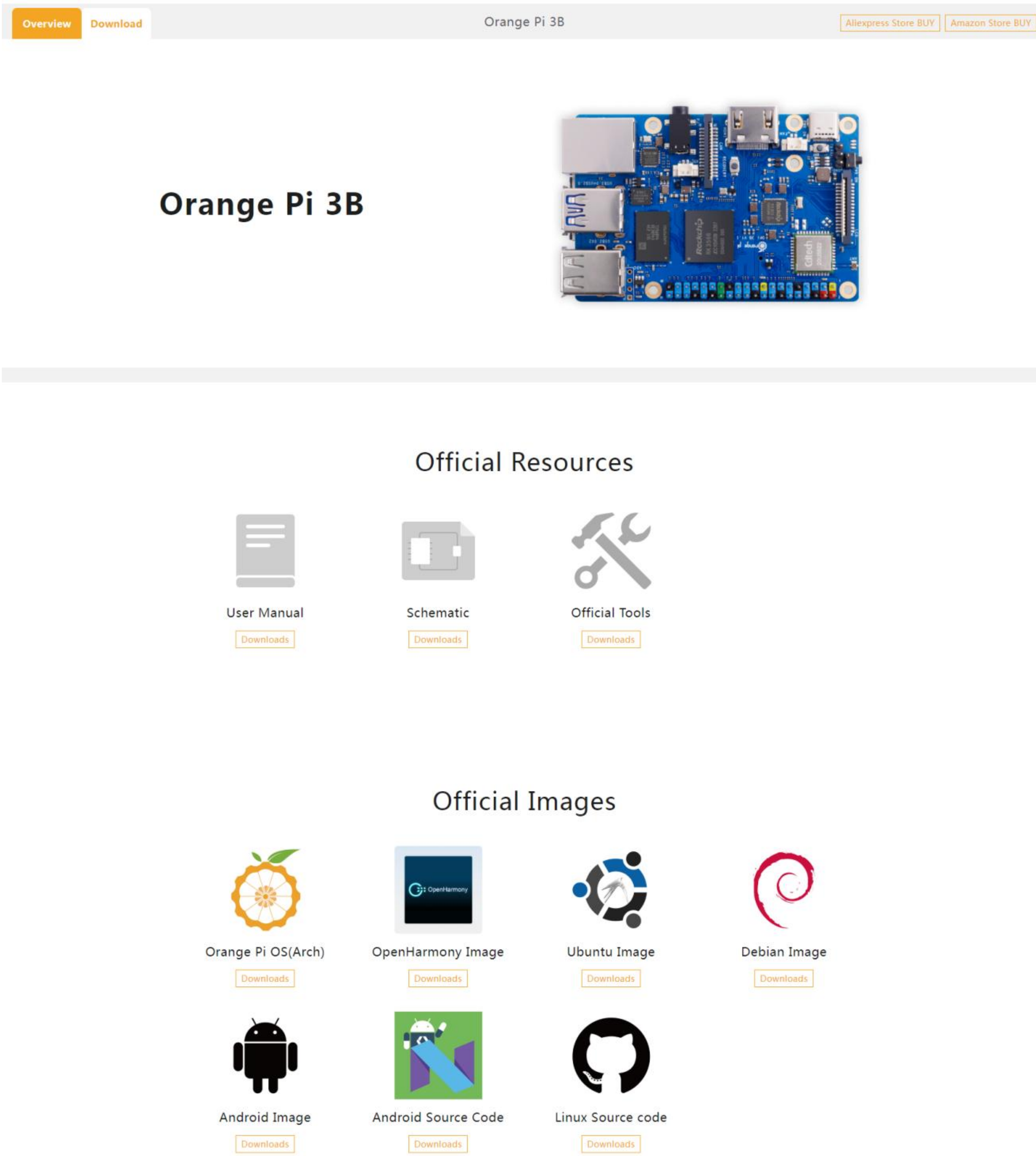
13) A personal computer with Ubuntu and Windows operating systems installed

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 of materials is:

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-3B.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
i. **Orange Pi OS OH image:** saved on Google Cloud Disk
j. **OpenWRT image:** saved on Google Cloud Disk

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

Note that the Linux image mentioned here specifically refers to the images of Linux distributions such as Debian, Ubuntu, OpenWRT or OPi OS Arch **downloaded from the Orange Pi 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 more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brands

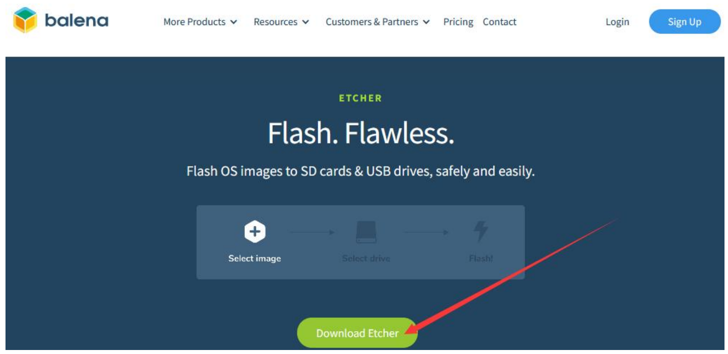
2) Then use the card reader to insert the TF card into the computer

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

4) Then download the burning software of Linux image—**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


DOWNLOAD

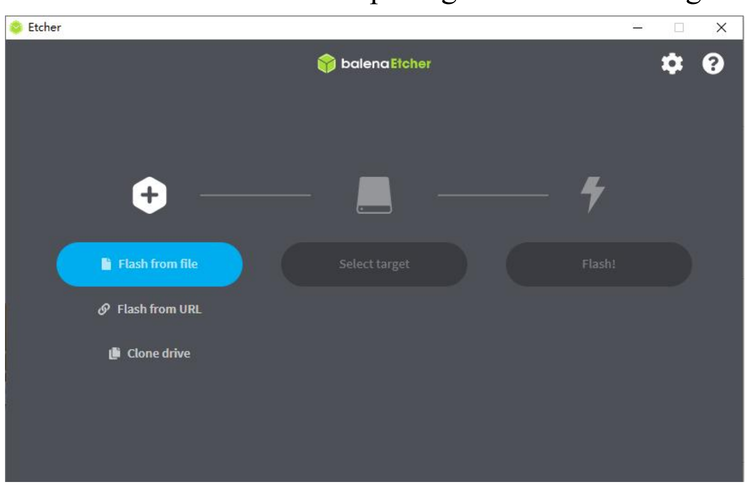
Download Etcher

ASSET	OS	ARCH	
ETCHER FOR WINDOWS (X86|X64) (INSTALLER)	WINDOWS	X86|X64	Download
ETCHER FOR WINDOWS (X86|X64) (PORTABLE)	WINDOWS	X86|X64	Download
ETCHER FOR WINDOWS (LEGACY 32 BIT) (X86|X64) (PORTABLE)	WINDOWS	X86|X64	Download
ETCHER FOR MACOS	MACOS	X64	Download
ETCHER FOR LINUX X64 (64-BIT) (APPIMAGE)	LINUX	X64	Download
ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)	LINUX	X86	Download

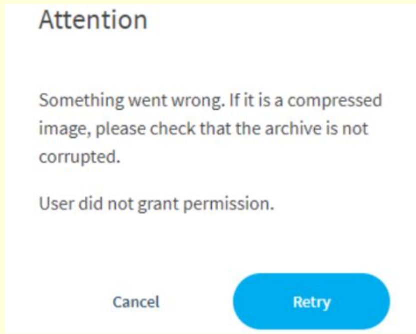
Looking for Debian (.deb) packages or Red Hat (.rpm) packages?

OSS hosting by cloudsmith

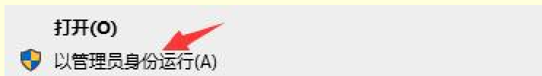
7) If the downloaded version of balenaEtcher needs to be installed, please install it before using it. If you downloaded the Portable version of balenaEtcher, just double-click to open it. The balenaEtcher interface after opening is shown in the figure below:



When opening balenaEtcher, if the following error is prompted:



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

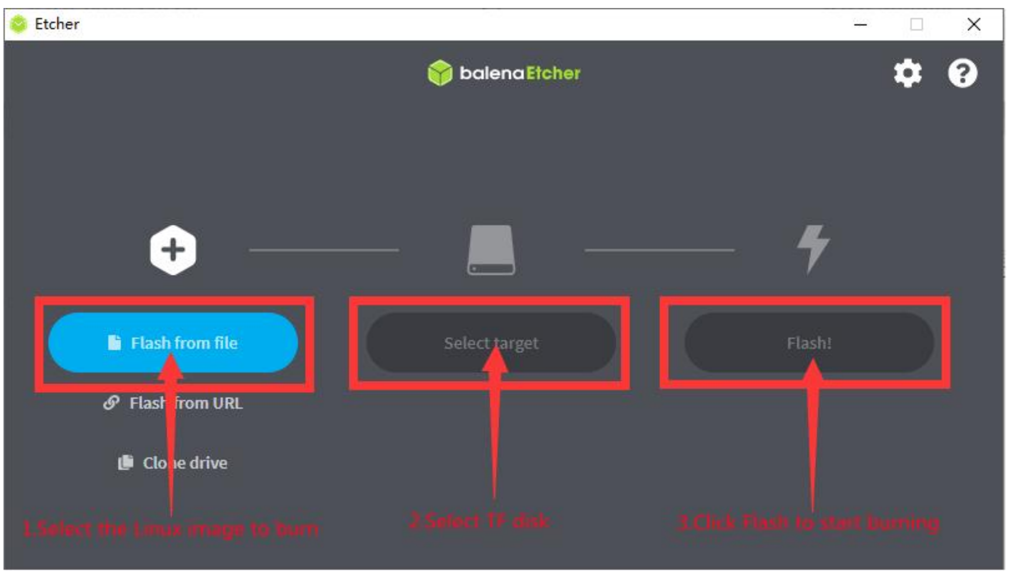


8) The specific steps to use balenaEtcher to burn the Linux image are as follows

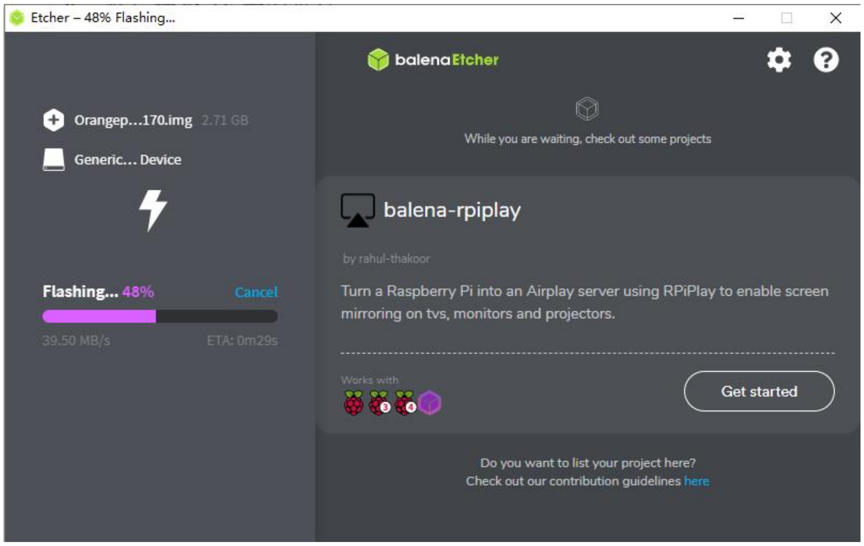
   a. First select the path of the Linux image file to be burned

   b. Then select the drive letter of the TF card

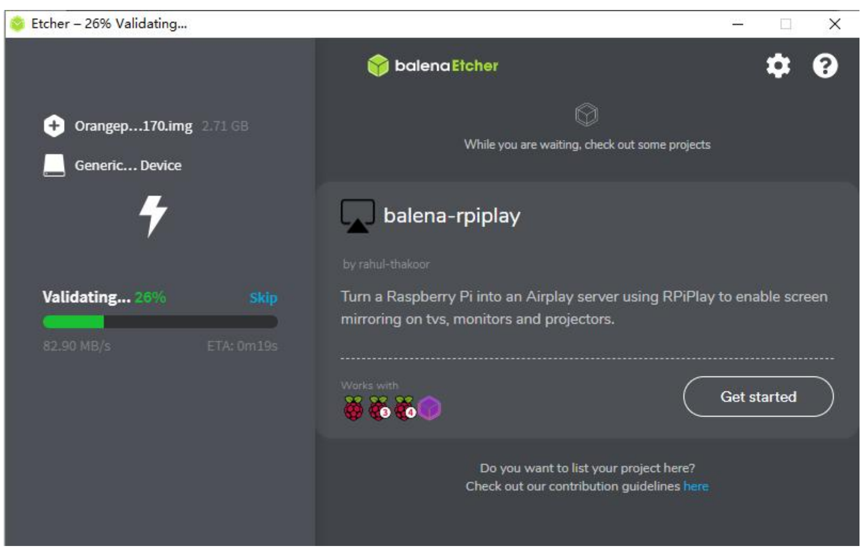
   c. Finally click Flash to start burning the Linux image to the TF card



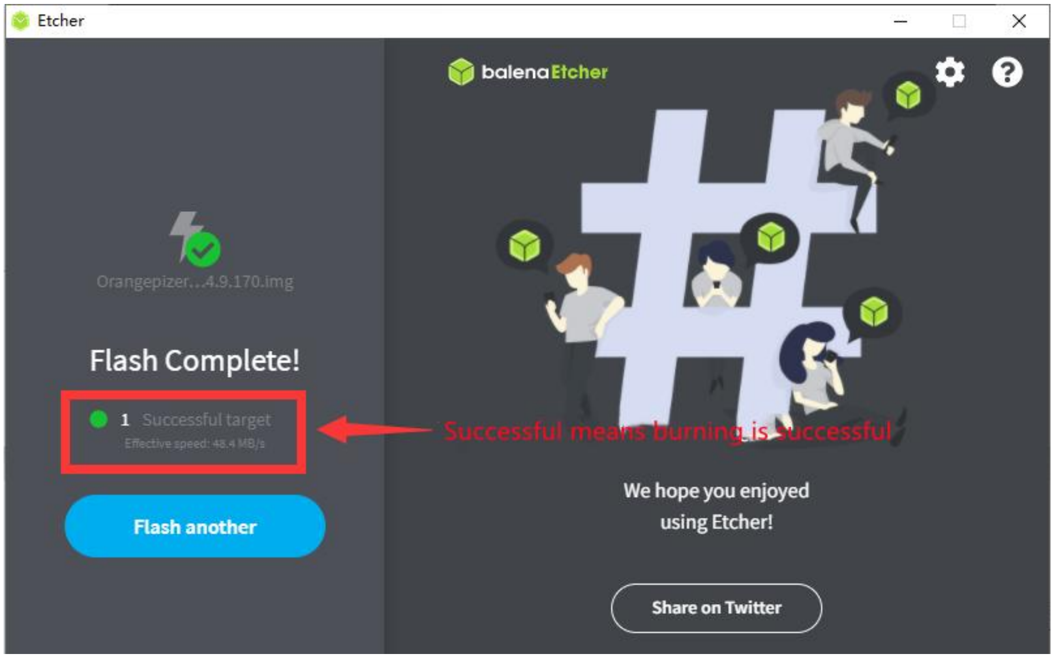
9) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card



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



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

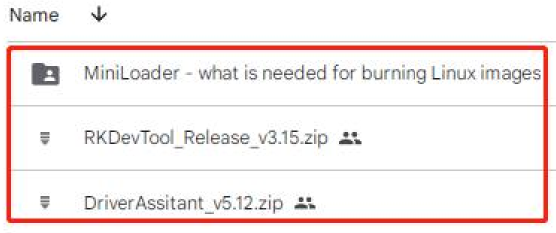
1) First, you need to prepare a good quality USB2.0 male-to-male data cable

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

3) Then download the Rockchip driver **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 data download page of Orange Pi, first select the **official tool**, and then enter the following folder

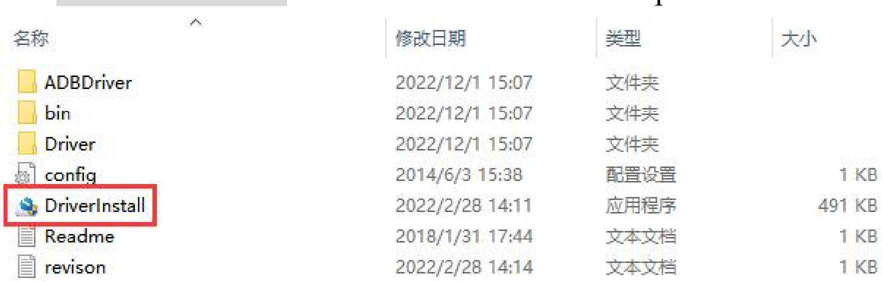
b. Then download all the files below



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

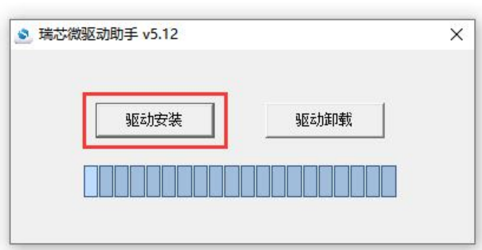
4) Then download the Linux operating system image file compression package that you want to burn from the **Orange Pi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system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

5) 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

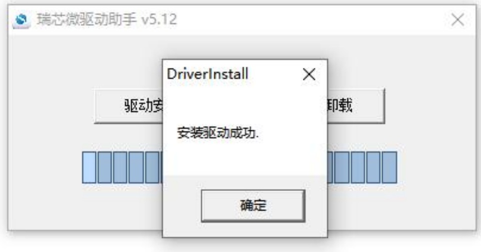


6) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

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



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

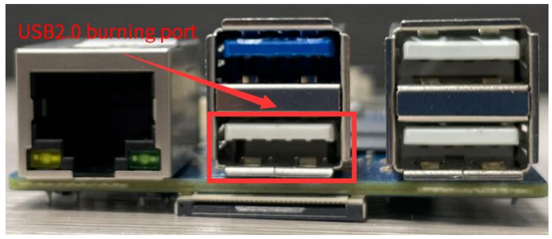


7) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

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**"

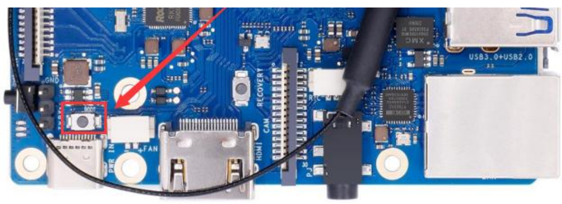
9) Then start burning the Linux image to the TF card

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

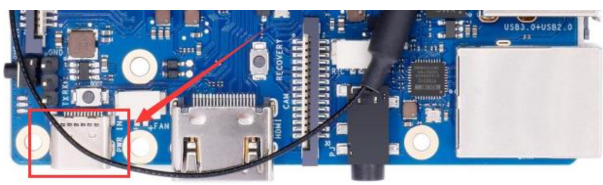


b. Make sure the TF card slot is not inserted into the TF card

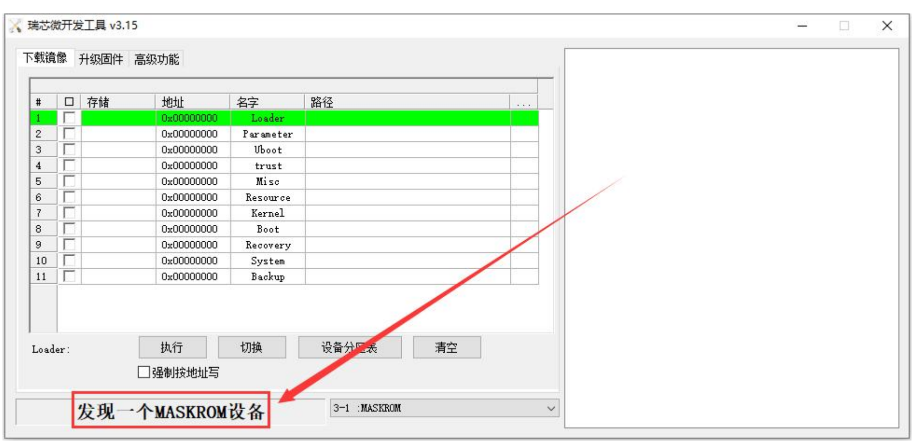
c. Then press and hold the MaskROM button on the development board, the position of the MaskROM button on the development board is shown in the figure below:



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

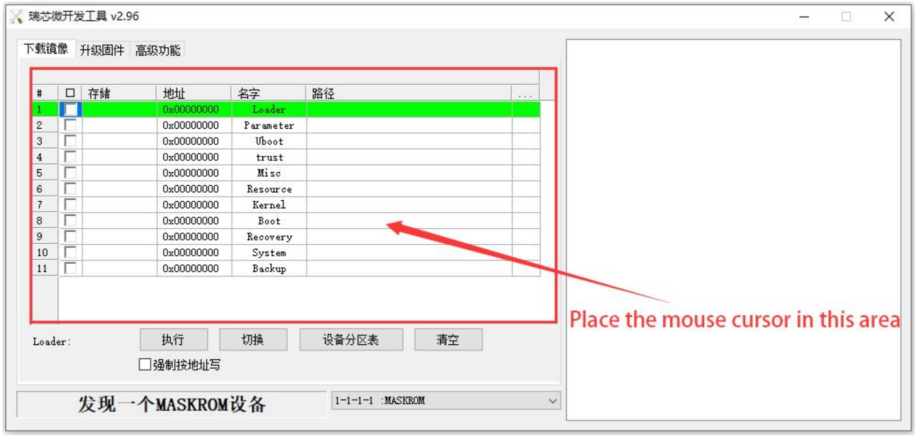


e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"

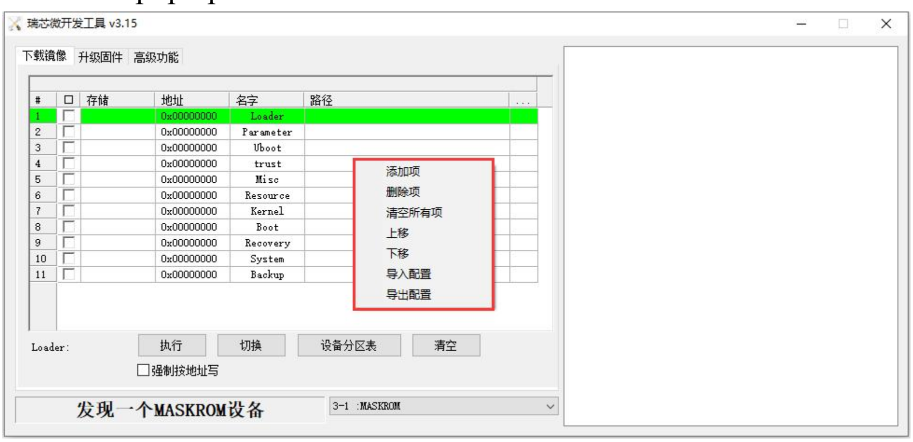


f. At this time, please insert the TF card into the TF card slot

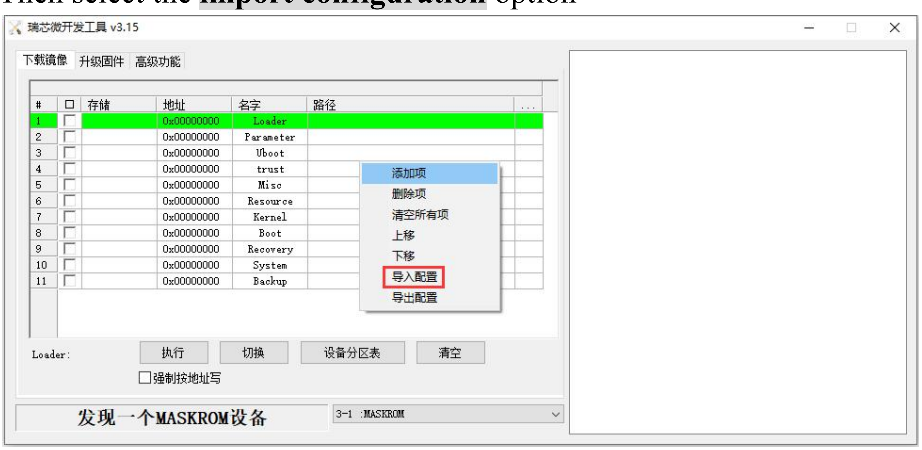
g. Then place the mouse cursor in the area below



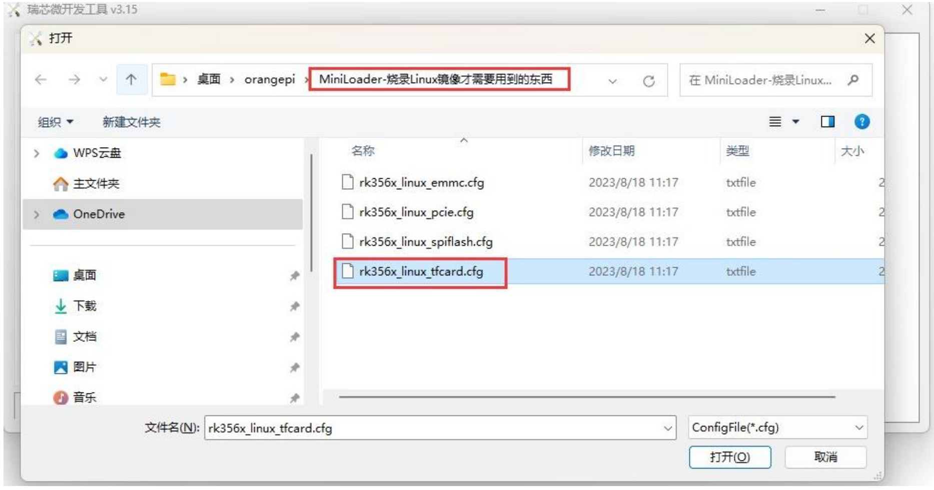
h. Then click the right mouse button and the selection interface shown in the figure below will pop up



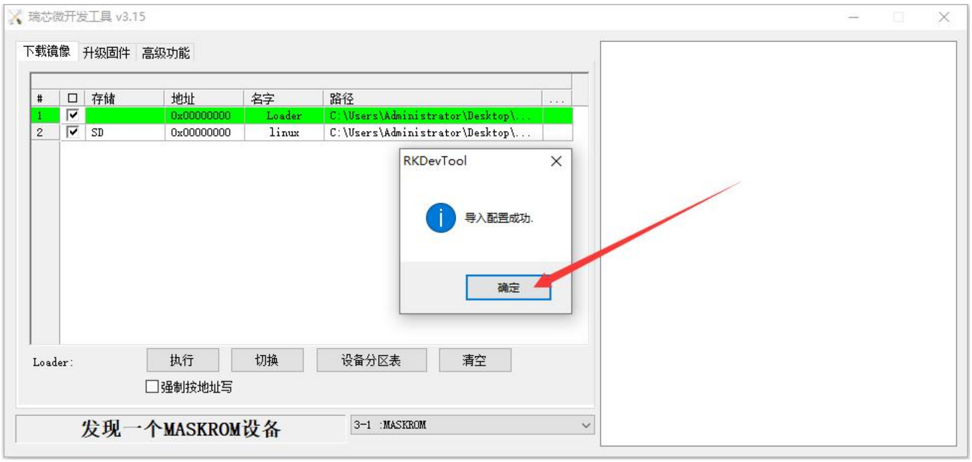
i. Then select the **import configuration** option



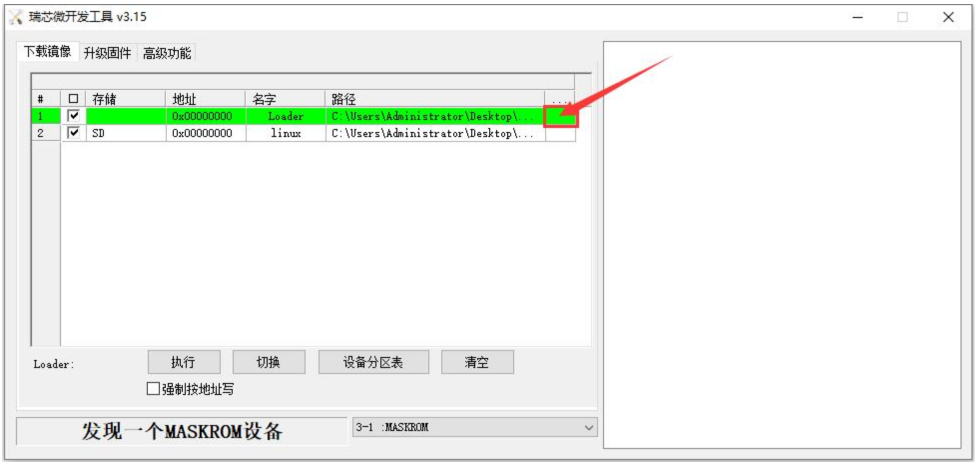
j. Then select the **rk356x_linux_tfcard.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and click **Open**



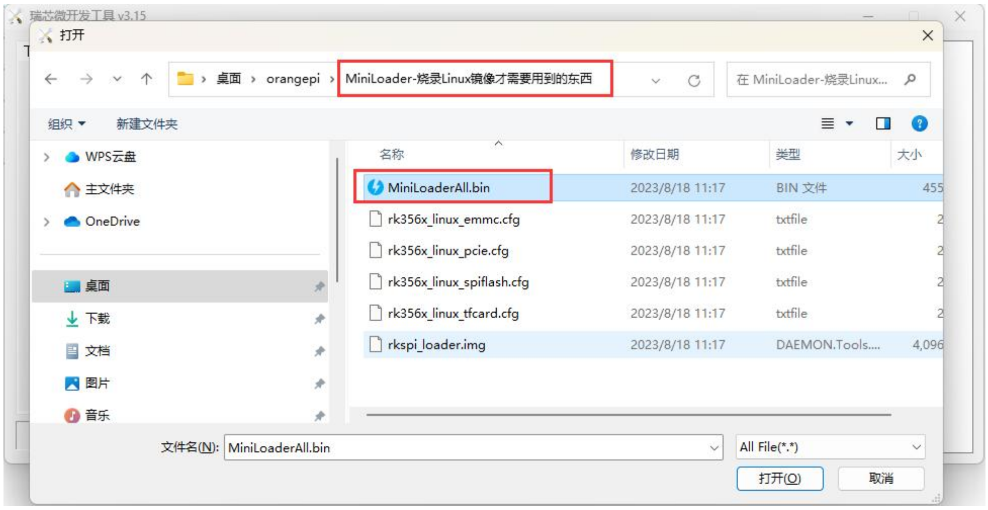
k. Then click **OK**



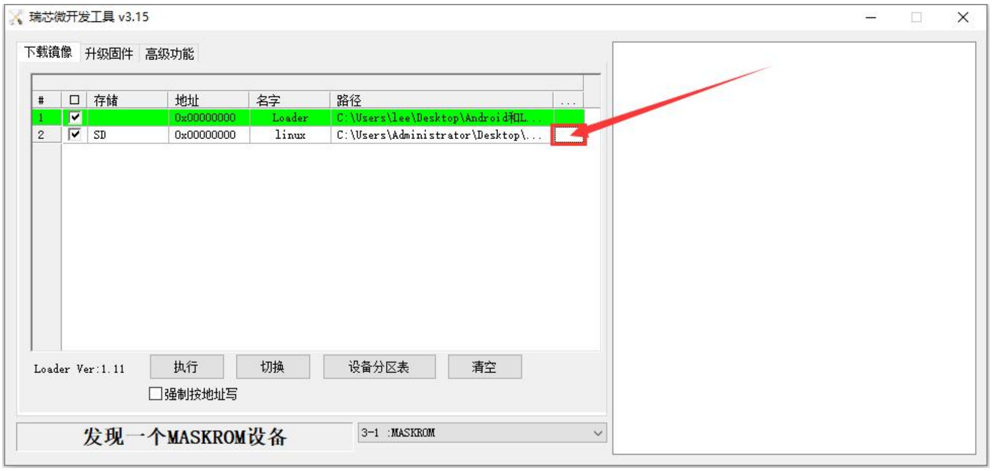
l. Then click the position shown in the figure below



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

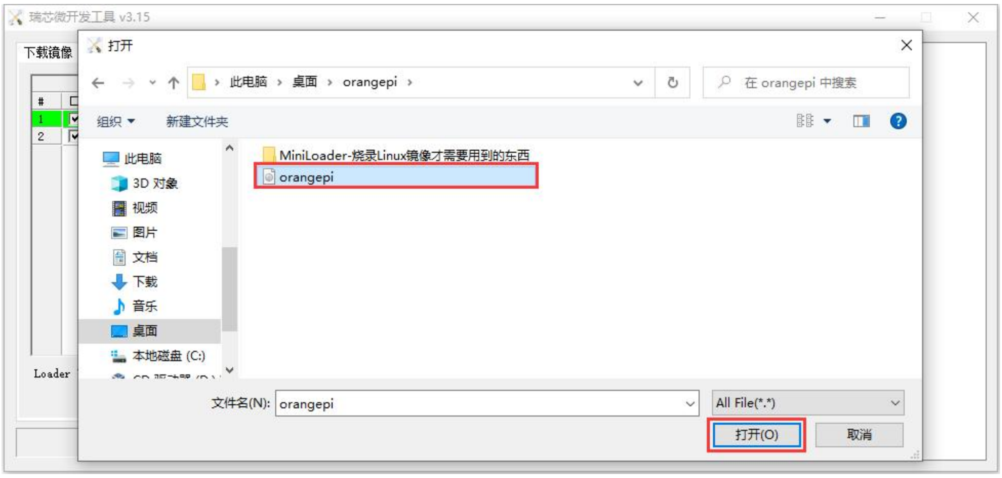


n. Then click the position shown in the figure below

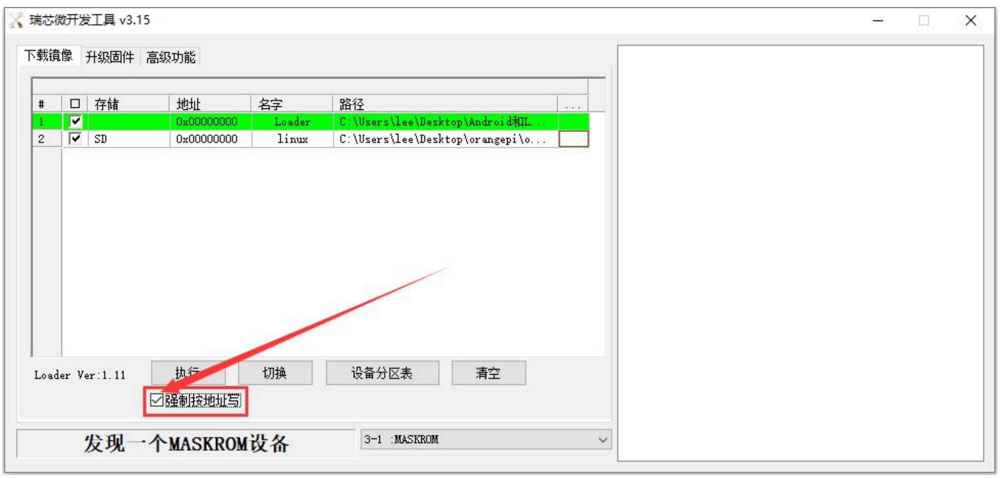


o. Then select the path of the Linux image you want to burn, and then click **Open**

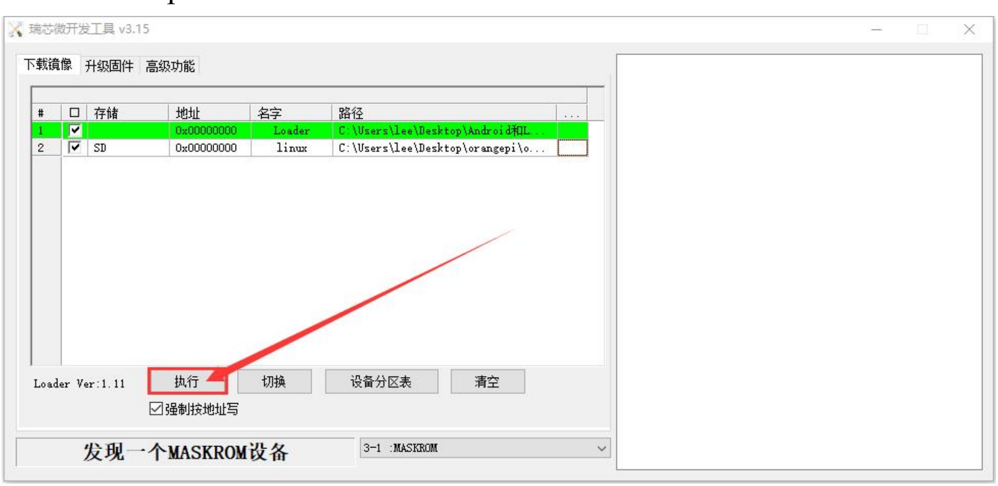
Before burning the image, it is recommended to rename the Linux image to be burned to orangepi.img or other shorter names, so that you can see the percentage value of the burning progress when burning the image.



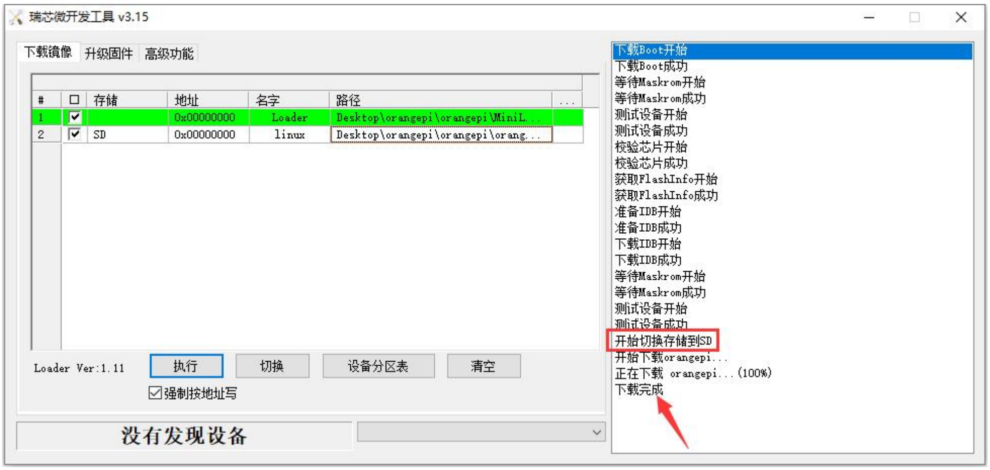
p. Then please check the **mandatory write by address** option



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



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



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

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

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

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

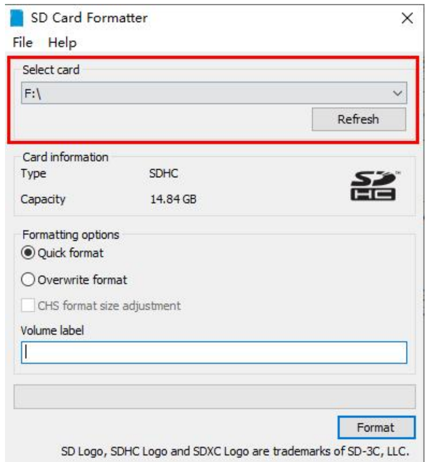
3) Then format the TF card

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

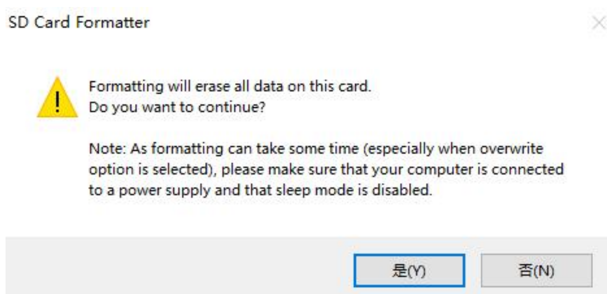
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

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

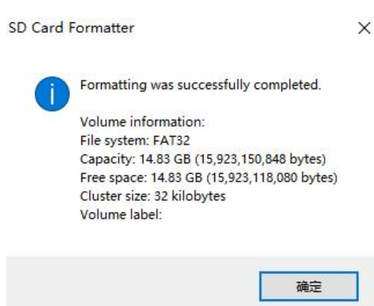
c. If only a TF card is inserted into the computer, the drive letter of the TF card will be displayed in the "**Select card**" column. If multiple USB storage devices are inserted into the computer, you can select the corresponding drive letter of the TF card through the drop-down box



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



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



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

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

a. The download page of Win32Diskimager is

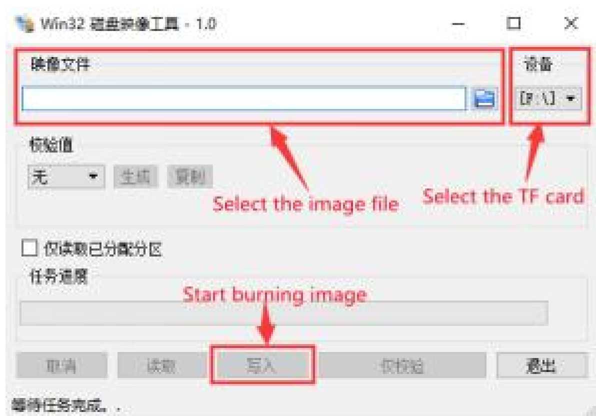
http://sourceforge.net/projects/win32diskimager/files/Archive/

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

a) First select the path of the image file

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

c) Finally click "**Write**" to start 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 image mentioned here specifically refers to the images of Linux distributions such as Debian, Ubuntu, `OpenWRT` or `OPi OS Arch` downloaded from the Orange Pi data download page, and the Ubuntu PC refers to the personal computer with the Ubuntu system installed.

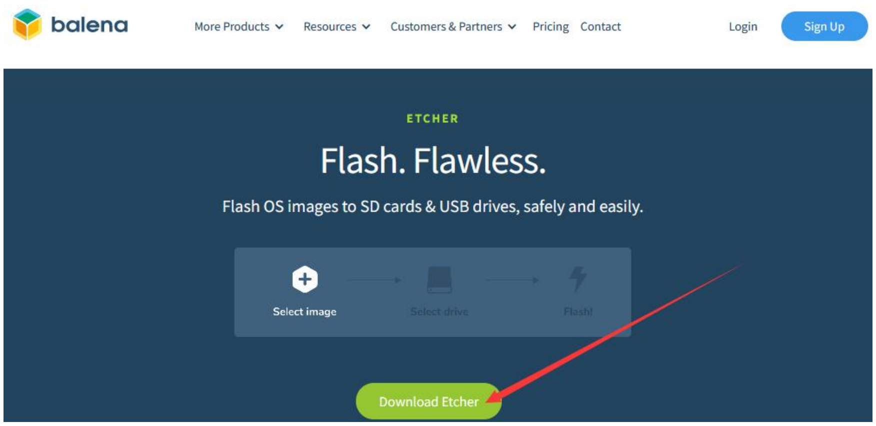
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brands

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

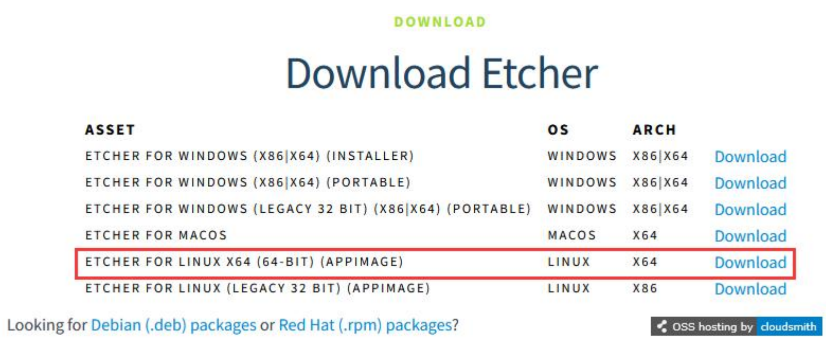
3) Download the balenaEtcher software, the download address is

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

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

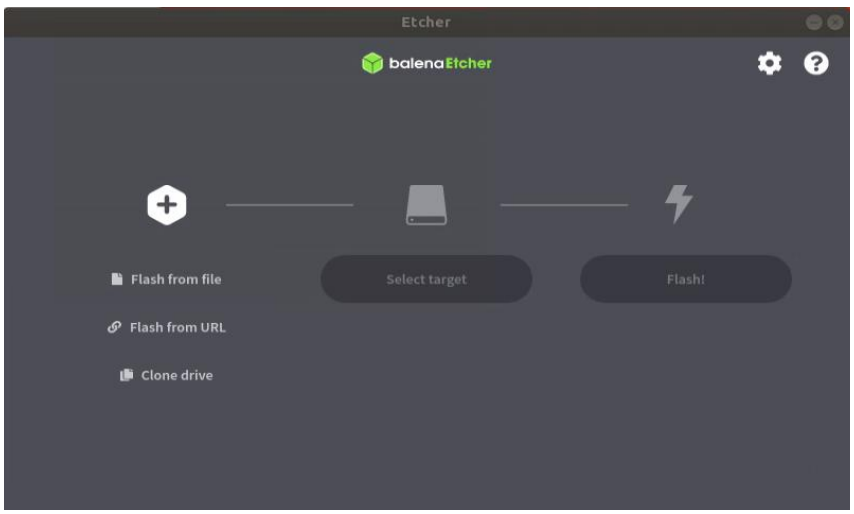
```
test@test:~$ 7z x Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_Linux5.10.160.7z
test@test:~$ ls Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_Linux5.10.160.*
```

```
Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_Linux5.10.160.7z
Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_Linux5.10.160.sha    #checksum file
Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_Linux5.10.160.img    #mirror file
```

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

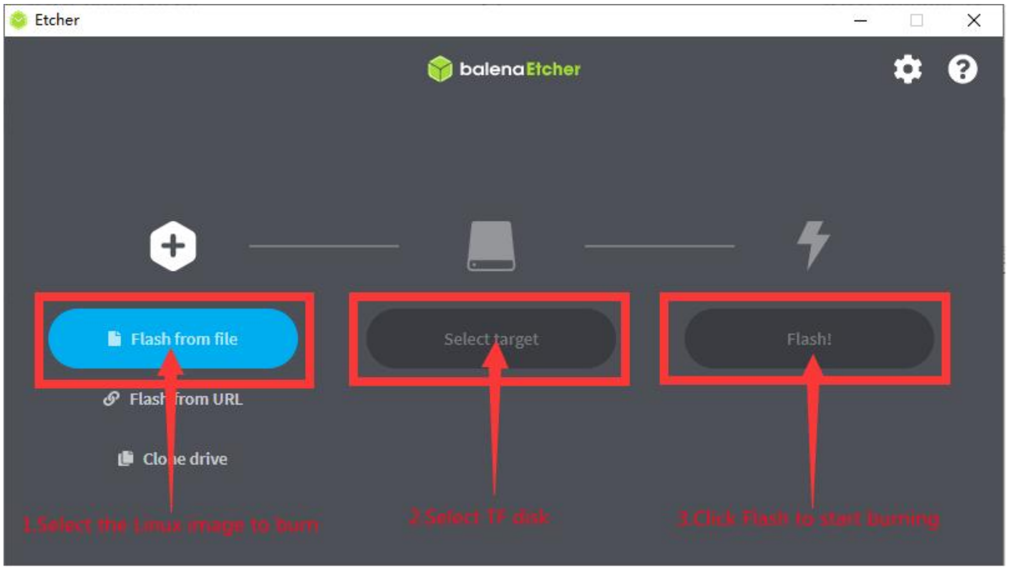
```
test@test:~$ sha256sum -c *.sha
Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_Linux5.10.160.img: OK
```

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

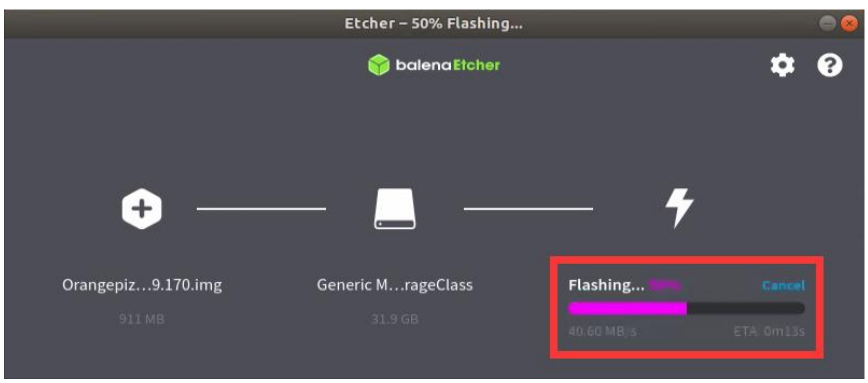


9) The specific steps of using balenaEtcher to burn the Linux image are as follows

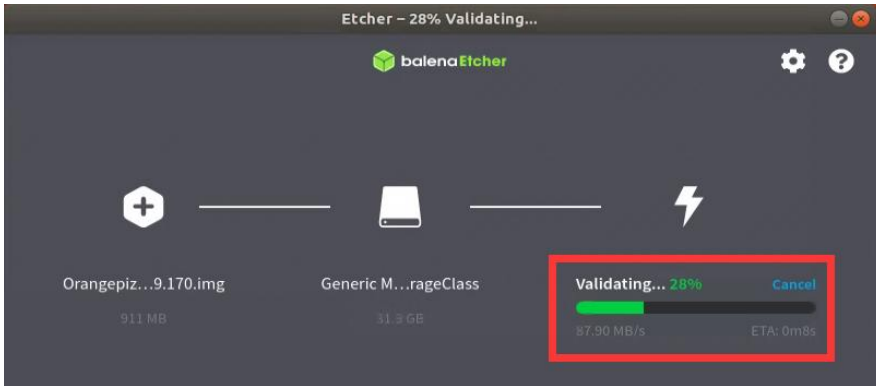
a. First select the path of the Linux image file to be burned
b. Then select the drive letter of the TF card
c. Finally, click Flash to start burning the Linux image to the TF card



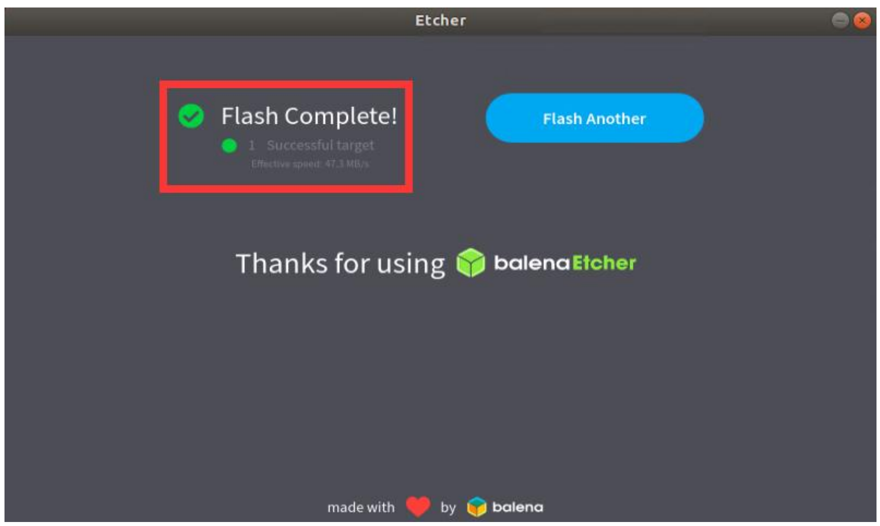
10) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card



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



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



2. 5. How to burn Linux image to eMMC

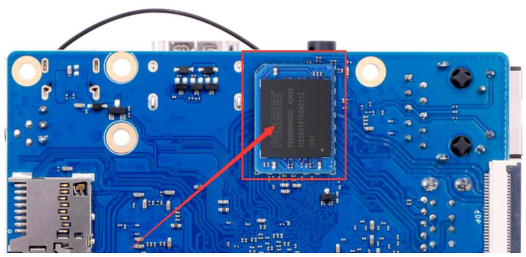
2. 5. 1. Using RKDevTool to burn the Linux image into eMMC

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

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

1) The development board reserves the expansion interface of the eMMC module.

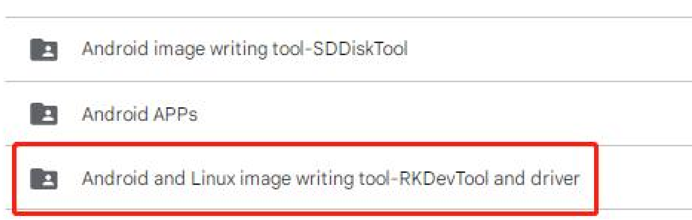
Before burning the system to the eMMC, you first need to purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC module to the development board. The eMMC module and the method of plugging into the development board are as follows:



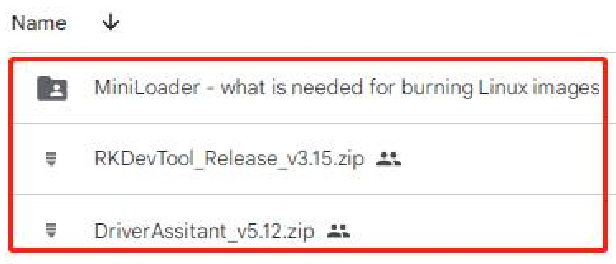
2) You also need to prepare a good quality USB2.0 male-to-male data cable

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 data download page**

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



b. Then download all the files below



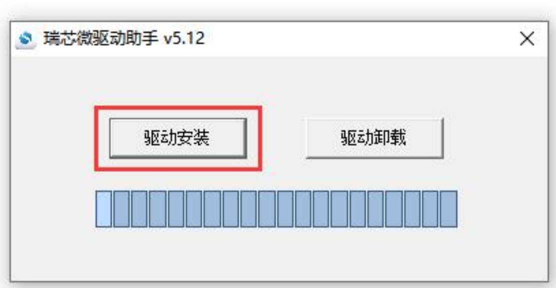
Note that the "MiniLoader-things needed to burn the Linux image" folder is hereinafter referred to as the MiniLoader folder.

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

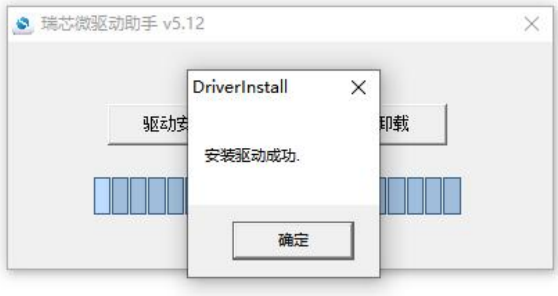
5) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

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

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



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

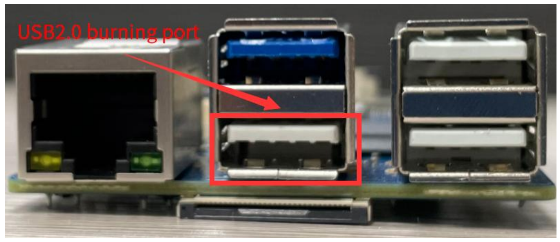


7) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

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**"

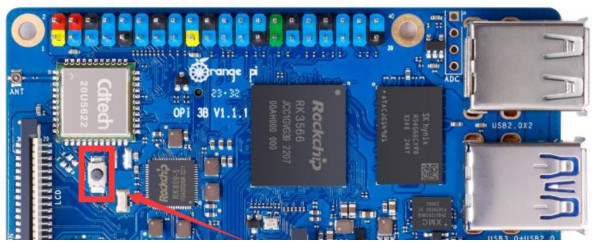
9) Then start burning the Linux image into eMMC

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

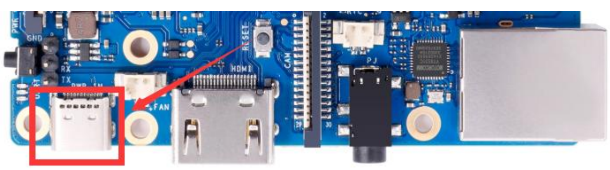


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

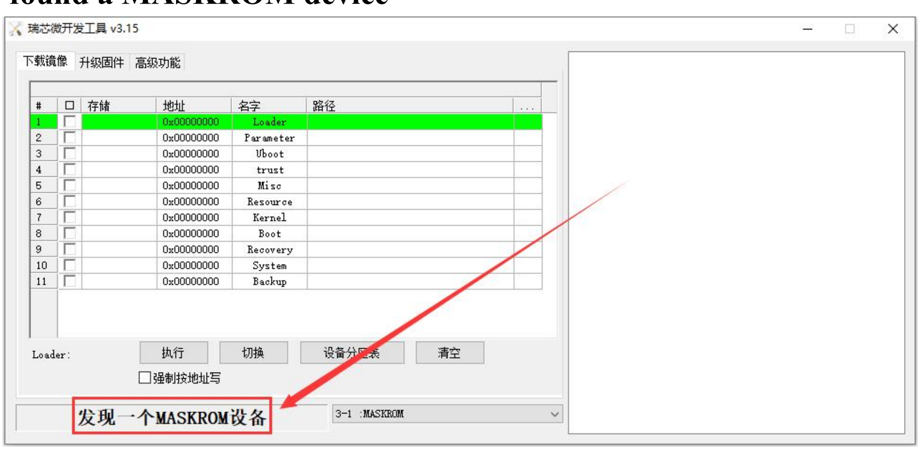
c. Then press and hold the MaskROM button on the development board, the position of the MaskROM button on the development board is shown in the figure below:



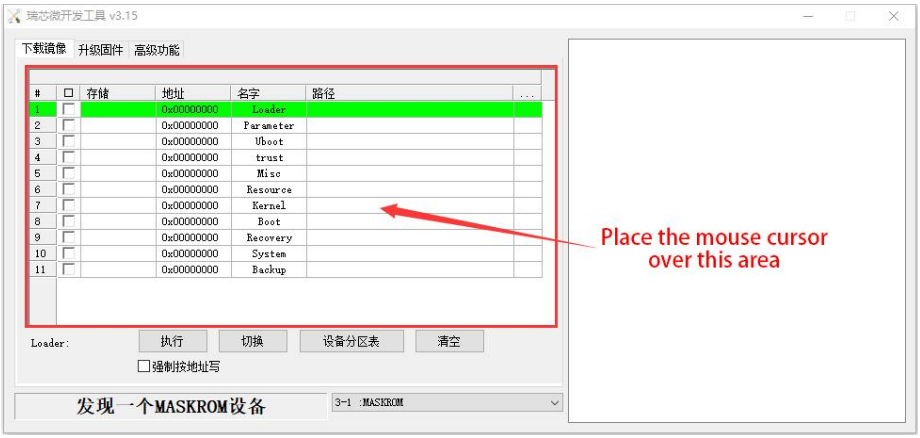
d. Then connect the power supply of the Type-C interface to the development board, and power on, and then release the MaskROM button



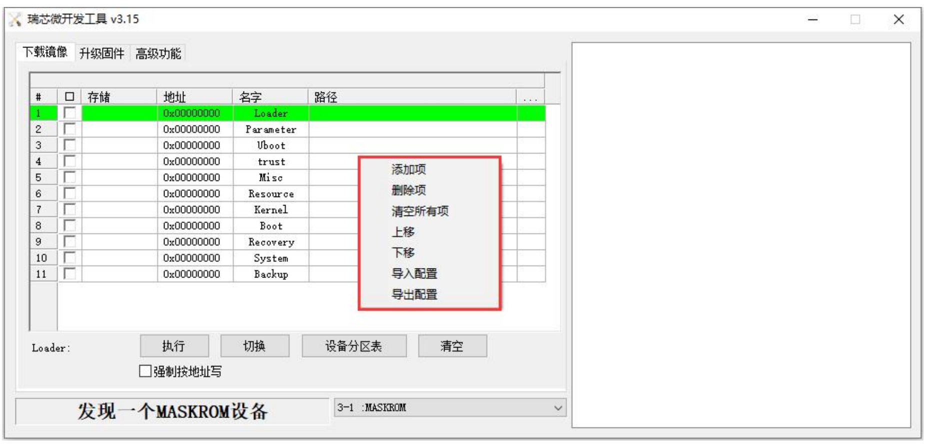
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt **"found a MASKROM device"**



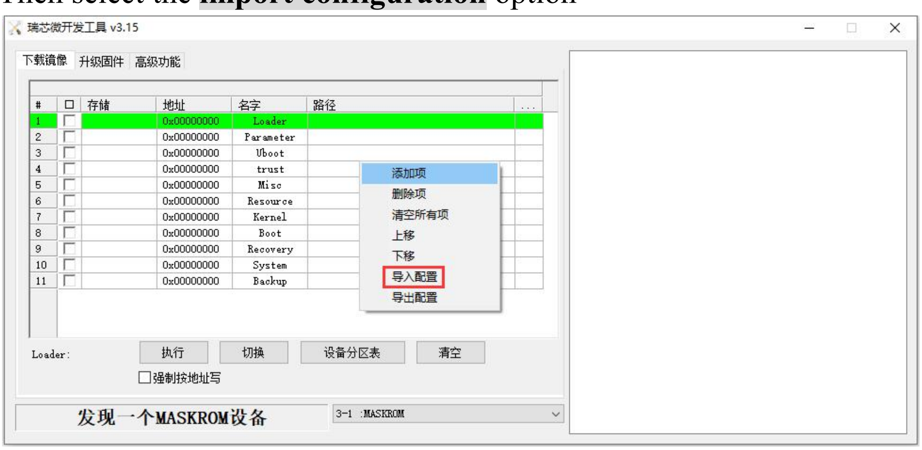
f. Then place the mouse cursor in the area below



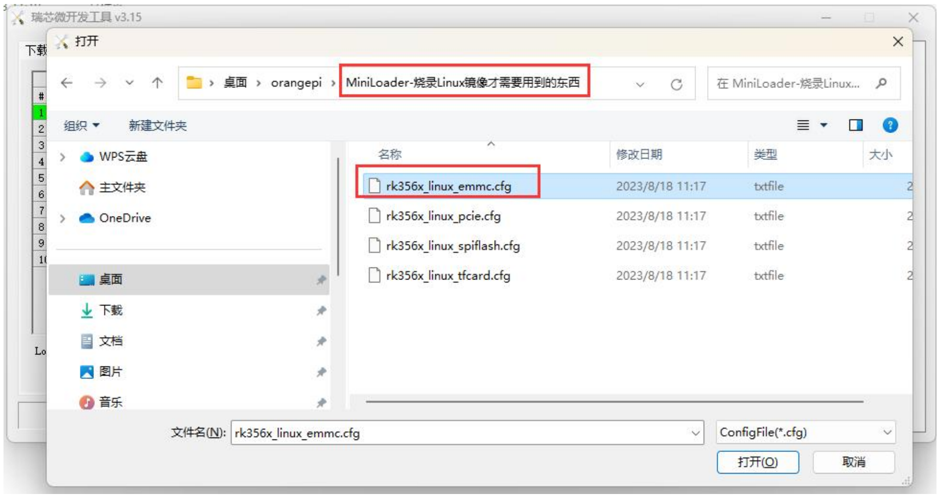
g. Then click the right mouse button and the selection interface shown in the figure below will pop up



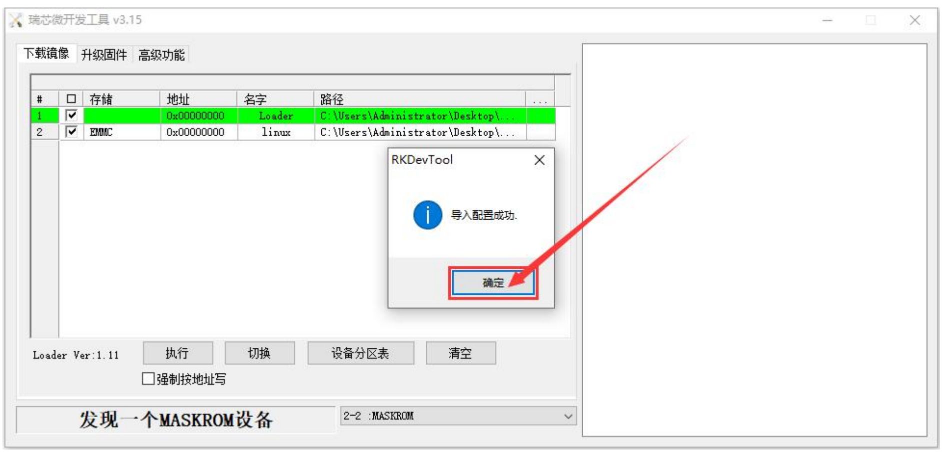
h. Then select the **import configuration** option



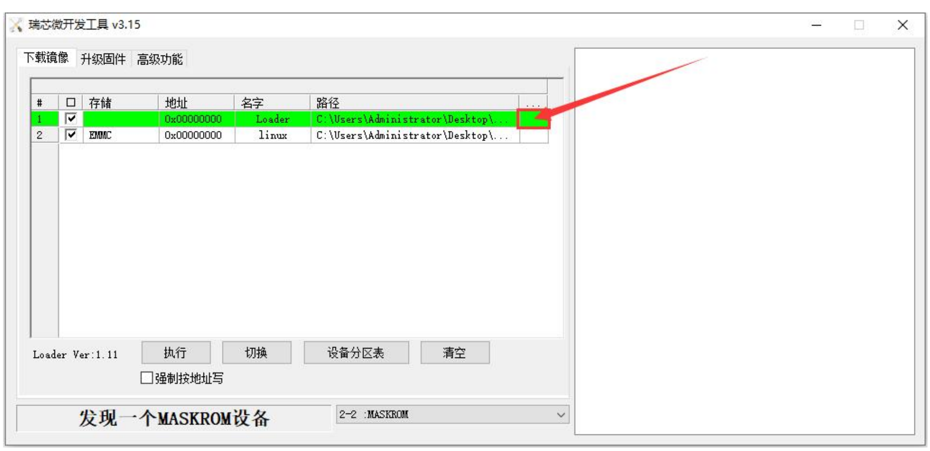
i. Then select the **rk356x_linux_emmc.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and click **Open**



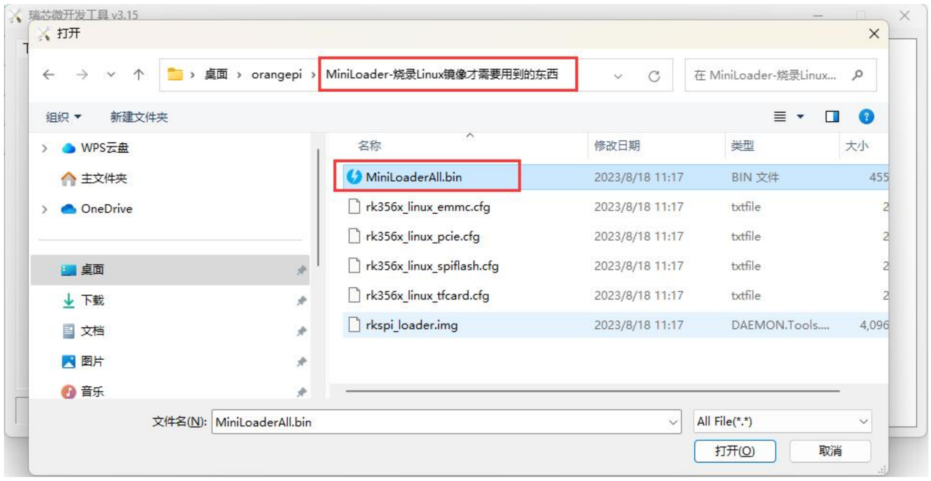
j. Then click **OK**



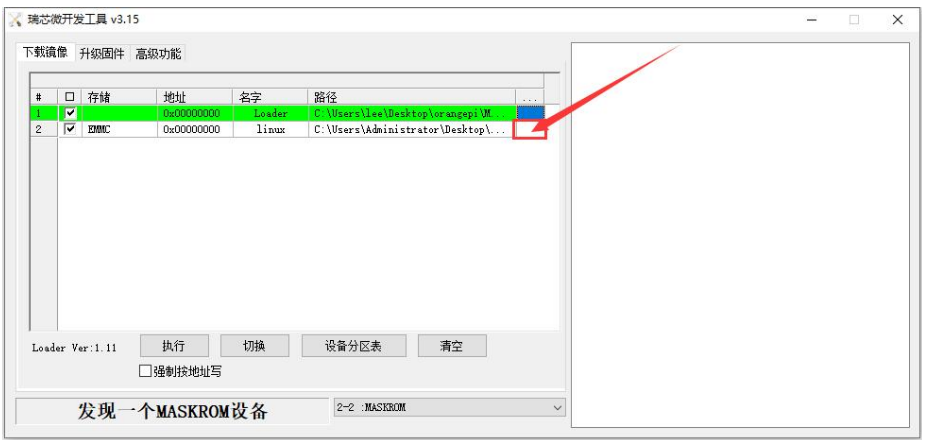
k. Then click the position shown in the figure below



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

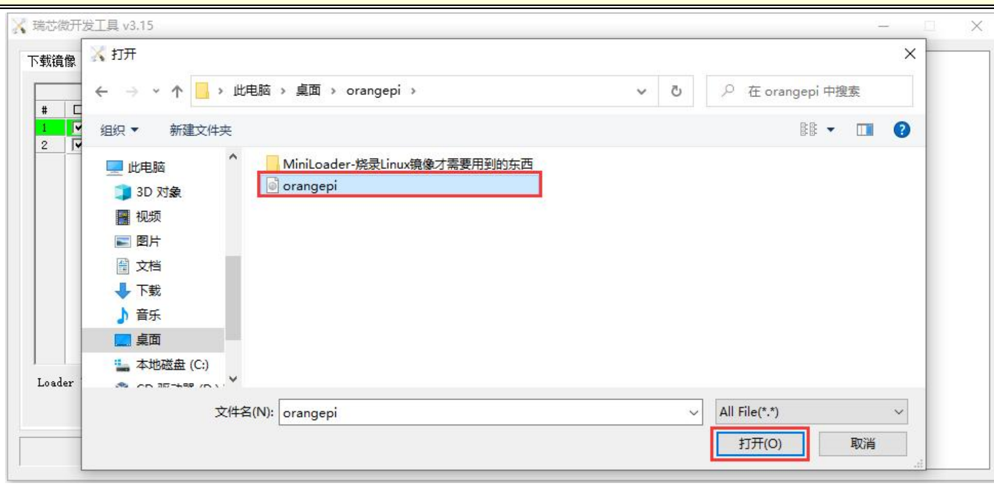


m. Then click the position shown in the figure below

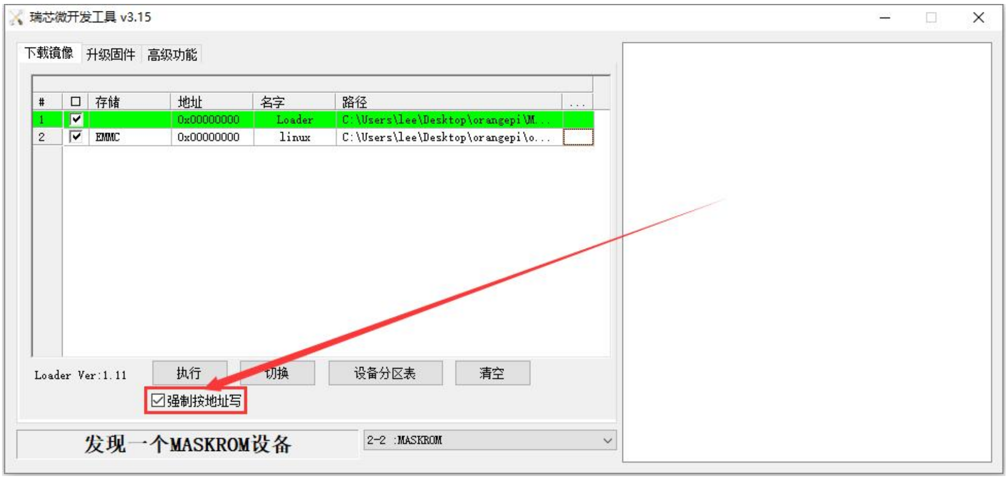


n. Then select the path of the Linux image you want to burn, and then click **Open**

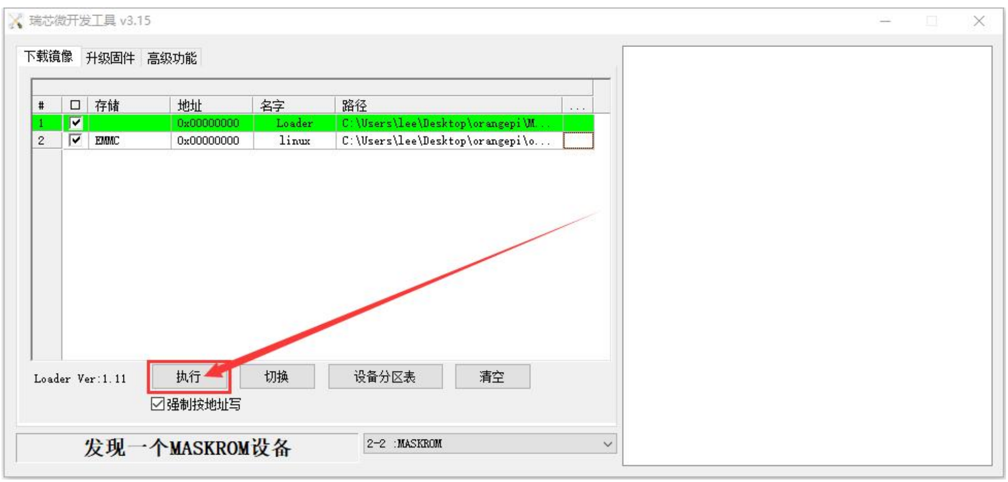
Before burning the image, it is recommended to rename the Linux image to be burned to orangepi.img or other shorter names, so that you can see the percentage value of the burning progress when burning the image.



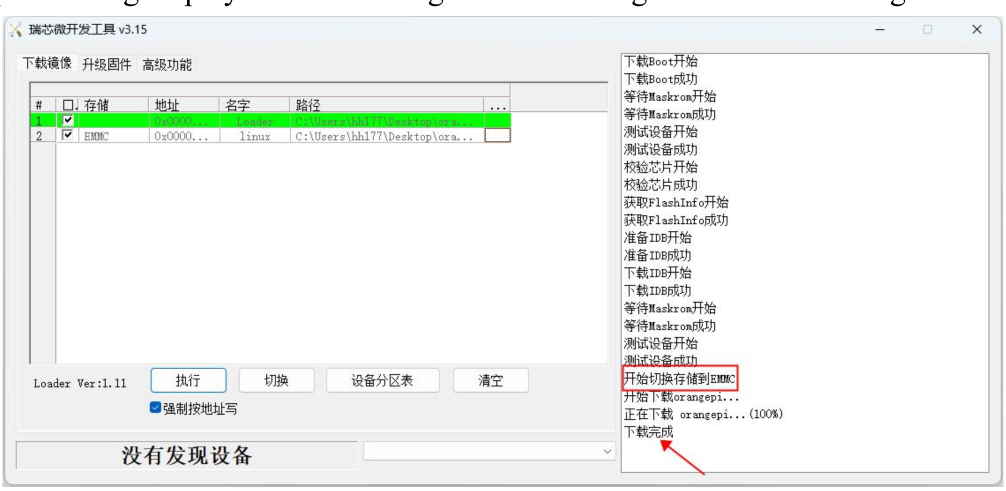
o. Then please check the option to **force writing by address**



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



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



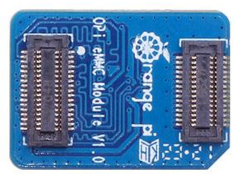
r. After burning the Linux image into the eMMC, the Linux system will start automatically.

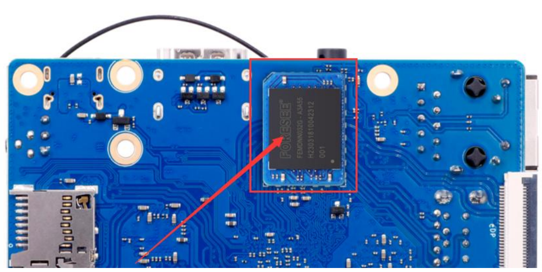
Note, after burning the image into eMMC, if the test finds that it cannot be started, please clear the SPIFlash and try again. For the method of clearing SPIFlash, please refer to the method of using RKDevTool to clear SPIFlash.

2. 5. 2. Use the dd command to burn the Linux image into eMMC

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

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





2) Using the dd command to burn the Linux image to eMMC needs to be completed with the help of a TF card, so first you need to burn the Linux image to the TF card, and then use the TF card to start the development board to enter the Linux system. For the method of burning the Linux image to the TF card, please refer to the instructions in the two sections of **the method of burning the Linux image to the TF card based on the Windows PC** and **the method of burning the Linux image to the TF card based on the Ubuntu PC**.

3) After using the TF card to start the Linux system, we first upload the decompressed

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

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

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepi3b_x.x.x_debian_bullseye_desktop_xfce_Linux5.10.160.img
```

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

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

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

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

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

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

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

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

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

a. In the following command, the if= parameter is followed by the full path where the Linux image is stored + the name of the Linux image (such as **the name of**

/home/orangepi/Desktop/Linux image). Because we have entered the path of the Linux image above, we only need to fill in the name of the Linux image.

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

```
sudo dd bs=1M if=Orangepi3b_x.x.x_debian_bullseye_desktop_xfce_Linux5.10.160.img of=/dev/mmcblk0 status=progress

sudo sync
```

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

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

8) After successfully burning the Linux image of the development board to the eMMC, you can use the **poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the Linux system in the eMMC will be started.

Note, after burning the image into eMMC, if the test finds that it cannot be started, please clear the SPIFlash and try again. For the method of clearing SPIFlash, please refer to the method of using RKDevTool to clear SPIFlash.

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

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

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

2. 6. 1. Using RKDevTool to burn

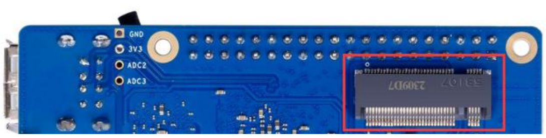
1) First, you need to prepare an NVMe SSD. The PCIe supported by the M.2 slot of the development board is PCIe2.0x1, and the theoretical maximum speed is 500MB/s.

PCIe3.0 and PCIe4.0 NVMe SSDs are also available, but the highest speed is only PCIe2.0x1.

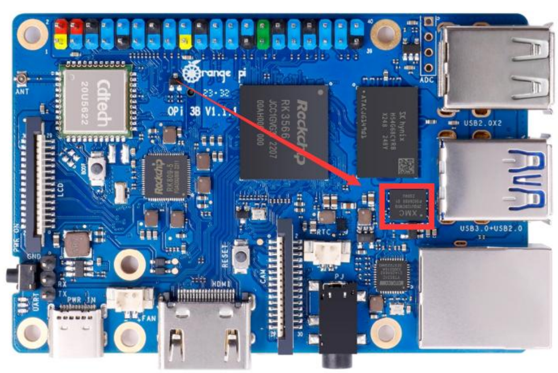
a. The M.2 2230 SSD is as follows

b. The M.2 2242 SSD is as follows

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



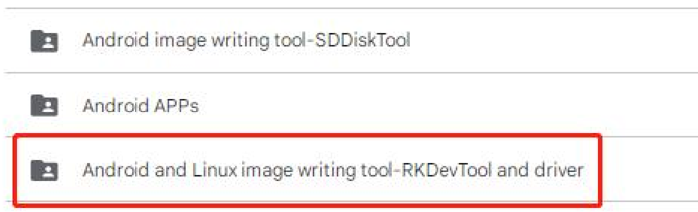
3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming



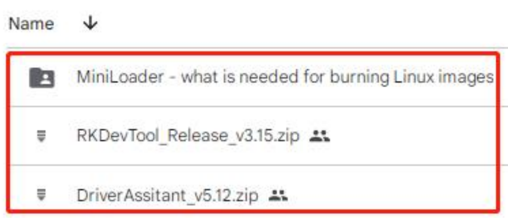
4) Then you need to prepare a good quality USB2.0 male-to-male data cable

5) Then download the Rockchip driver **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 data download page of Orange Pi, first select the **official tool**, and then enter the following folder



b. Then download all the files below



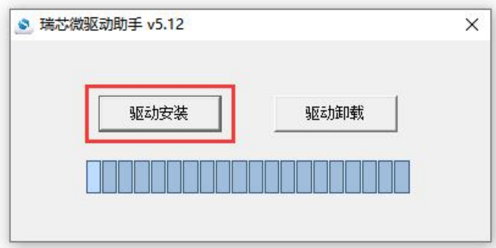
Note that the "MiniLoader-things needed to burn the Linux image" folder is hereinafter referred to as the MiniLoader folder.

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

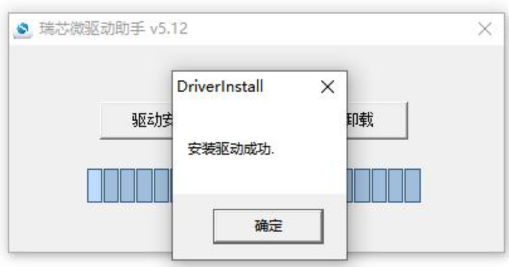
7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

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

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

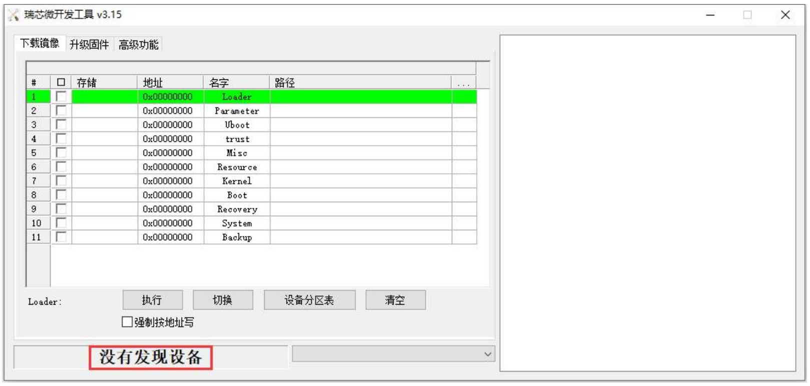


b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button.



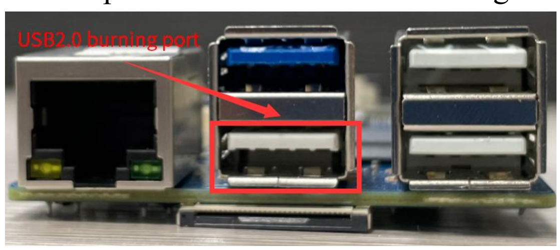
9) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

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



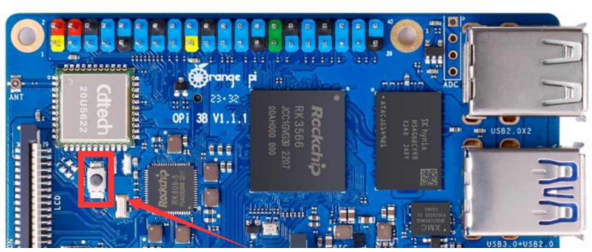
11) Then start burning the Linux image to the SSD

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

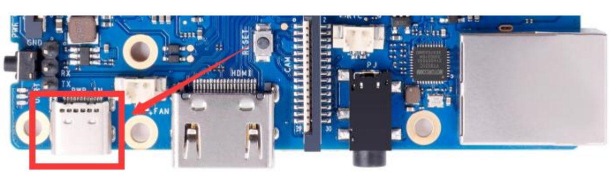


b. Make sure that the development board is not connected to the power supply and inserted into the TF card and eMMC

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:

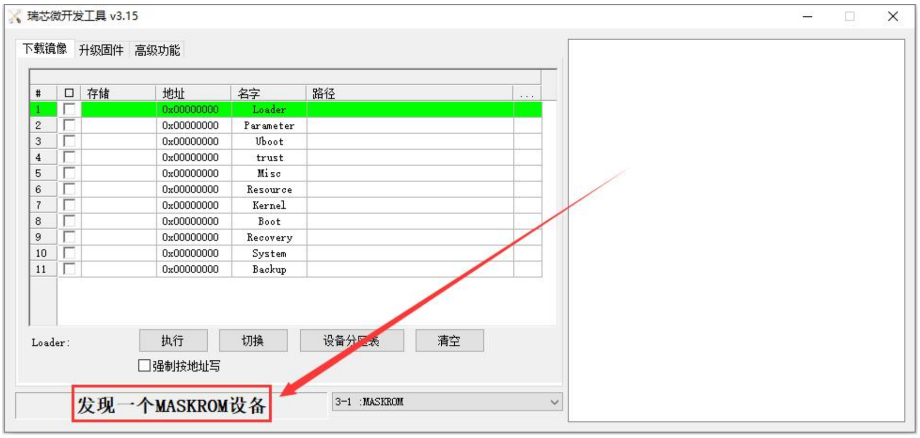


d. Then connect the power supply of the Type-C interface to the development board, and power on, and then release the MaskROM button

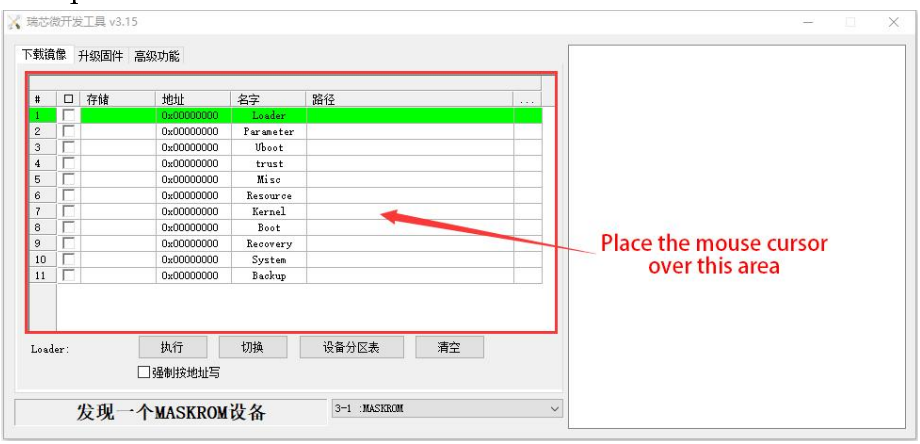


e. If the previous steps are successful, the development board will enter the

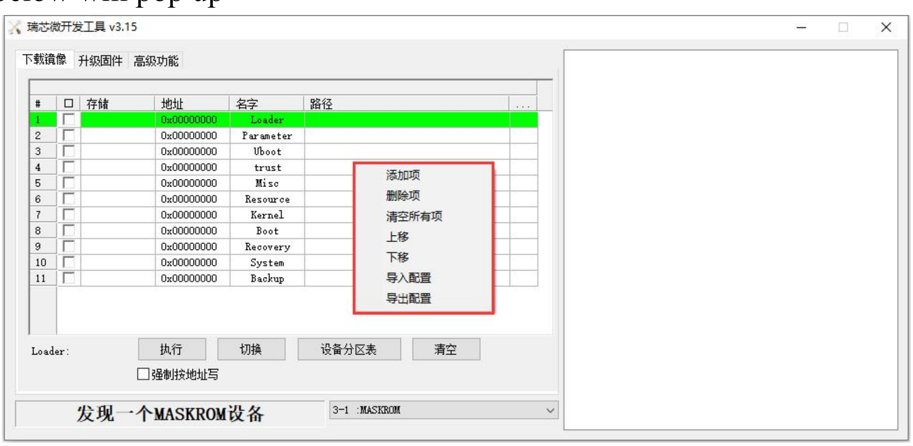
MASKROM mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



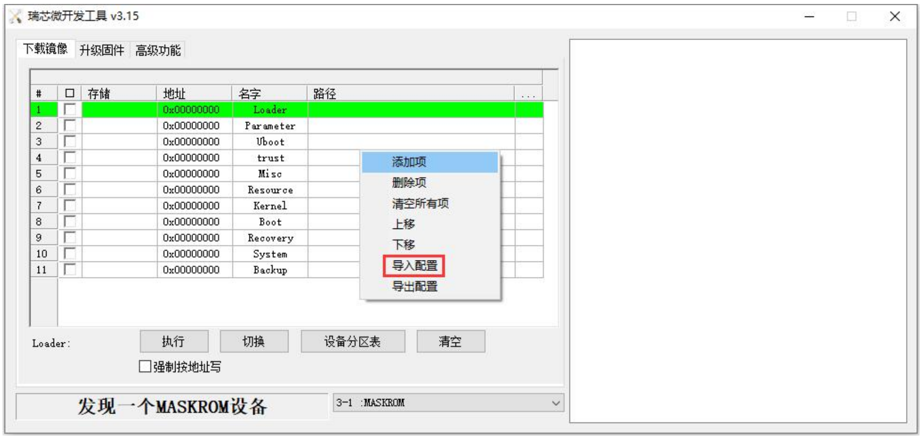
f. Then place the mouse cursor in the area below



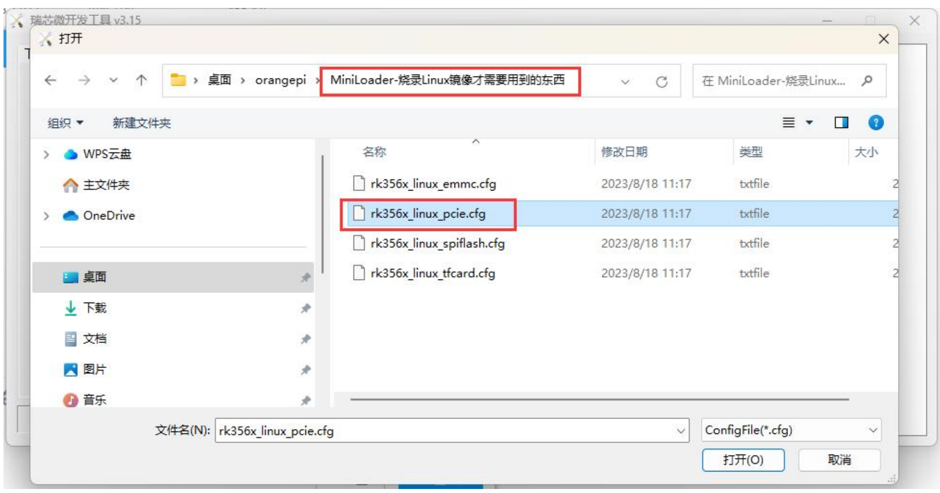
g. Then click the right mouse button and the selection interface shown in the figure below will pop up



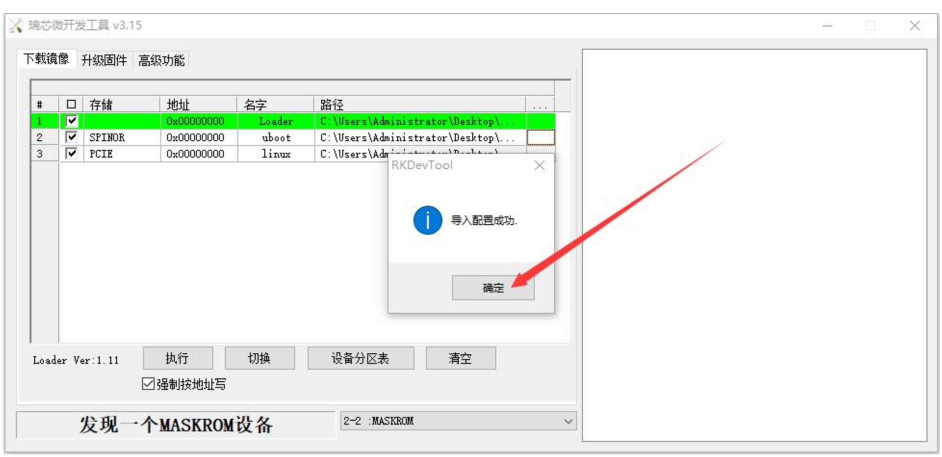
h. Then select the **import configuration** option



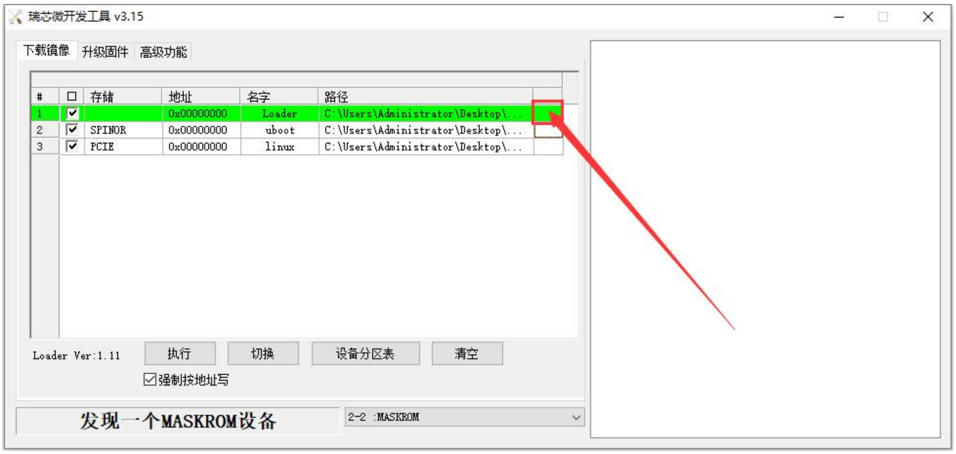
i. i. Then enter the **MiniLoader** folder downloaded earlier, then select the **rk356x_linux_pcie.cfg** configuration file, and click **Open**



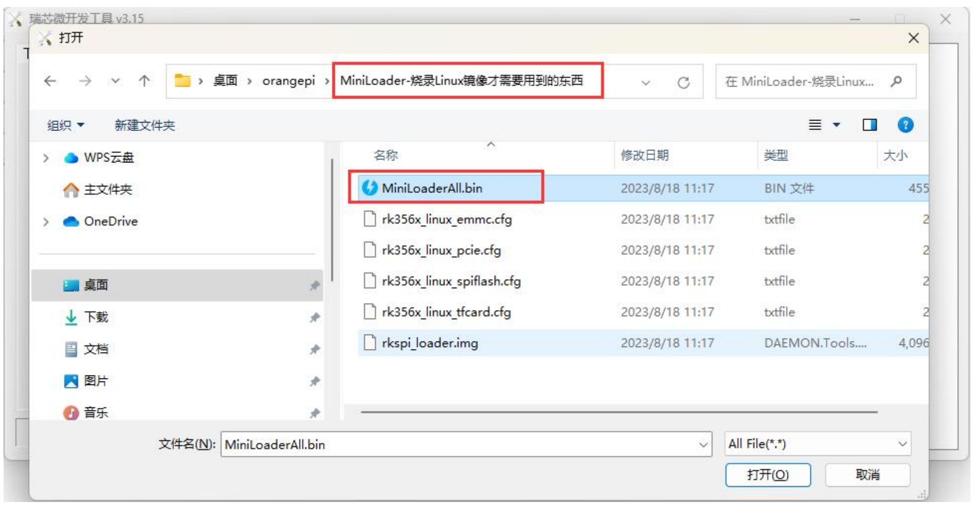
j. Then click **OK**



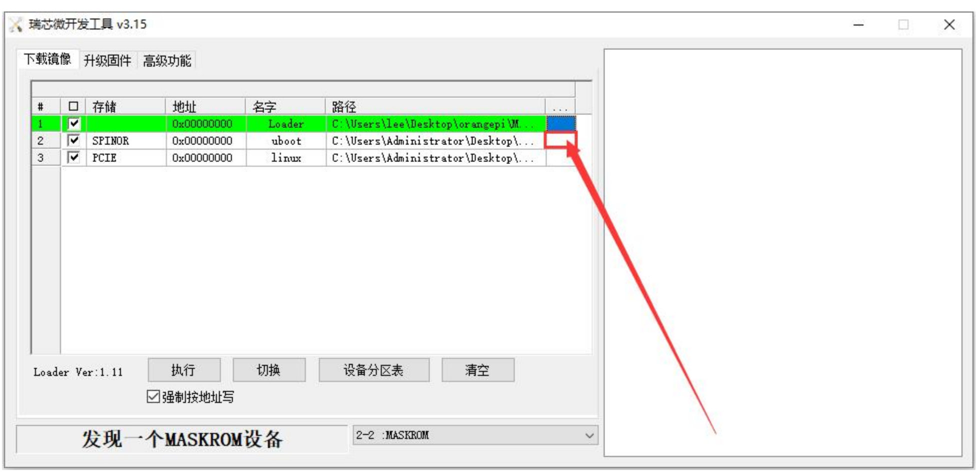
k. Then click the position shown in the figure below



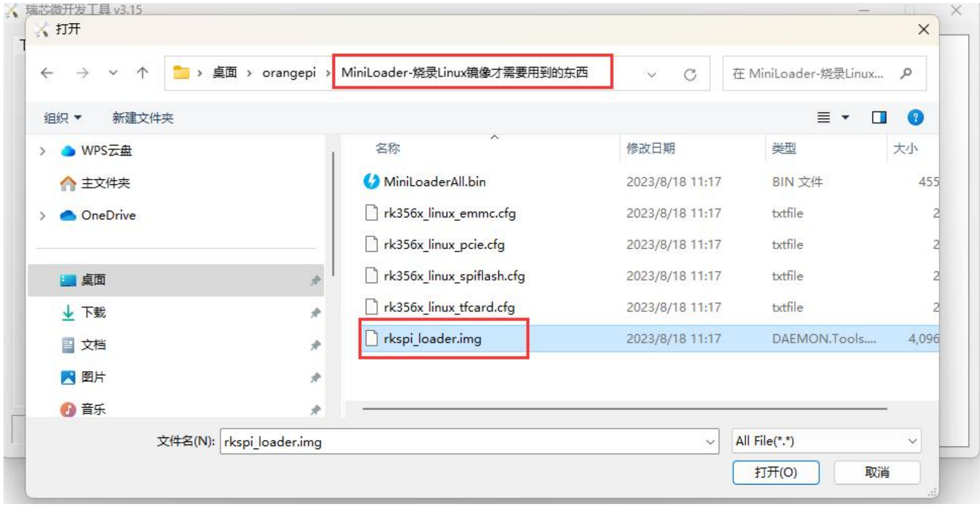
l. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click to **open**



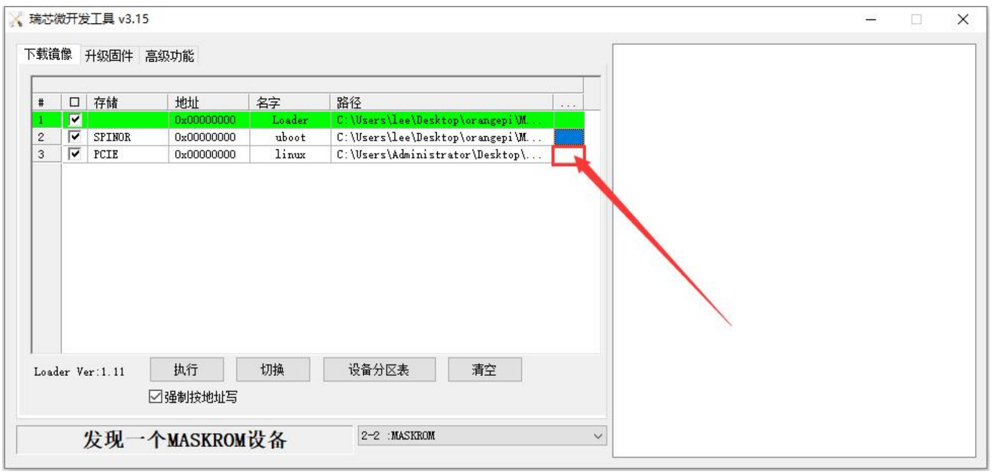
m. Then click the position shown in the figure below



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

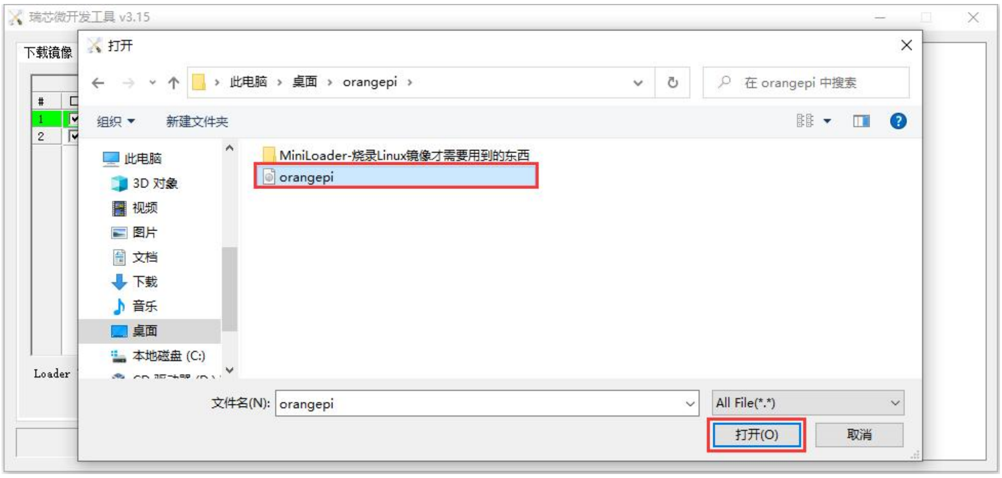


o. Then click the position shown in the figure below

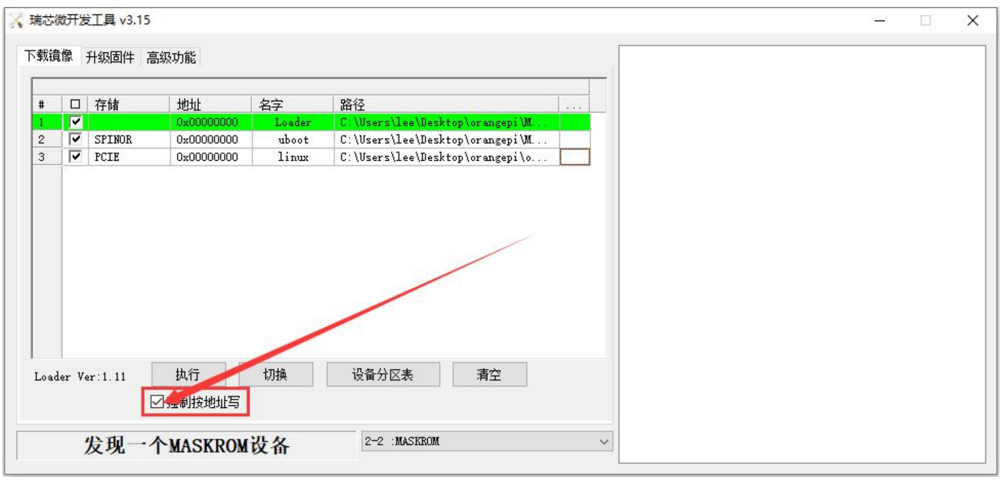


p. Then select the path of the Linux image you want to burn, and then click **Open**

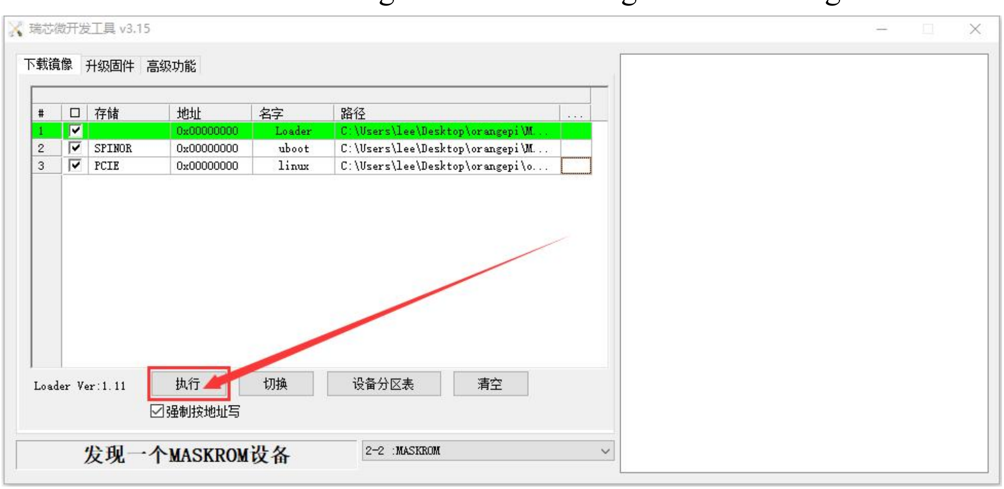
Before burning the image, it is recommended to rename the Linux image to be burned to orangepi.img or other shorter names, so that you can see the percentage value of the burning progress when burning the image.



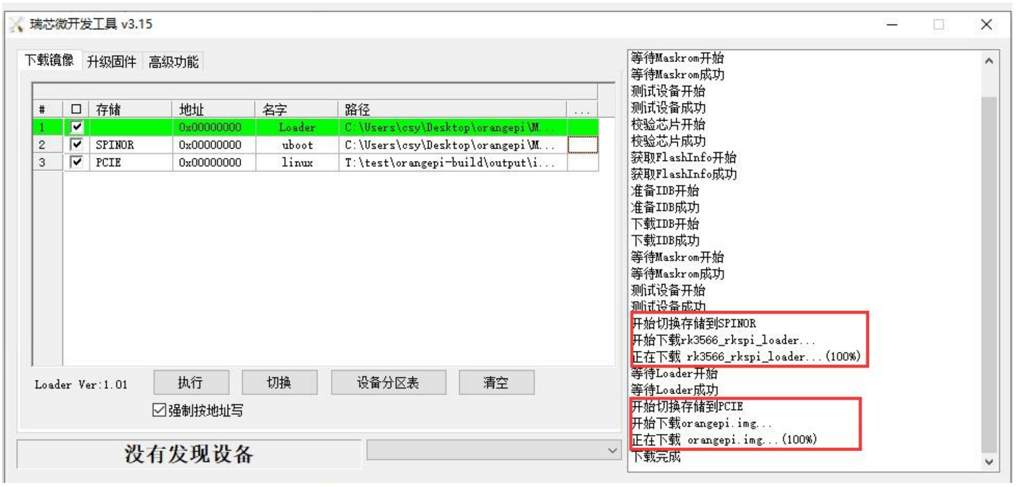
q. Then please check the option to force writing by address



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



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



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

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

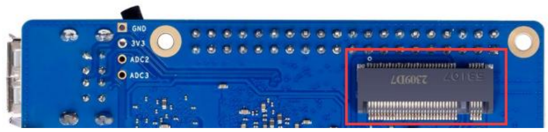
2.6.2. The method of using the dd command to burn

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

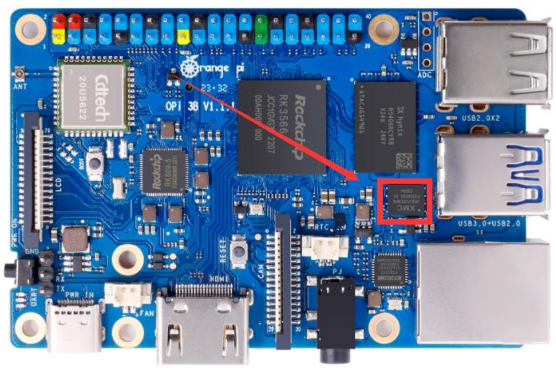
a. The M.2 2230 SSD is as follows

b. The M.2 2242 SSD is as follows

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



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



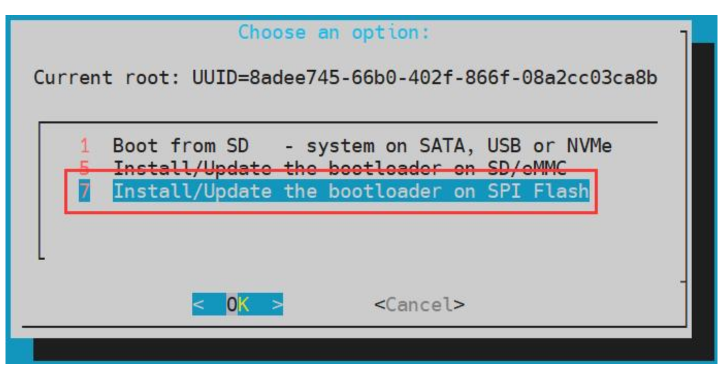
4) Burning the Linux image to SPIFlash+NVMe SSD requires a TF card, so first you need to burn the Linux image to the TF card, and then use the TF card to start the development board to enter the Linux system. For the method of burning the Linux image to the TF card, please refer to the instructions in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

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

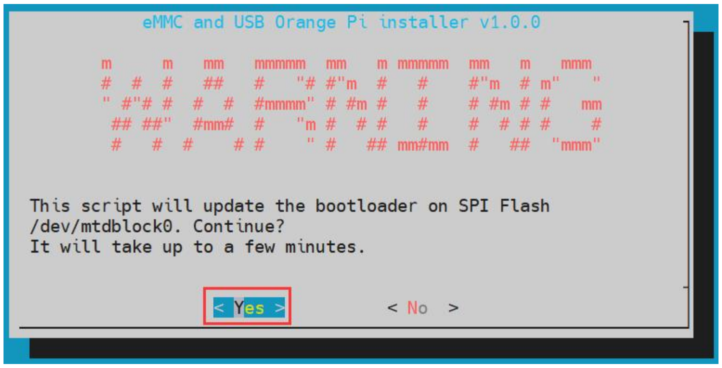
a. Run nand-sata-install first, **ordinary users remember to add sudo permission**

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

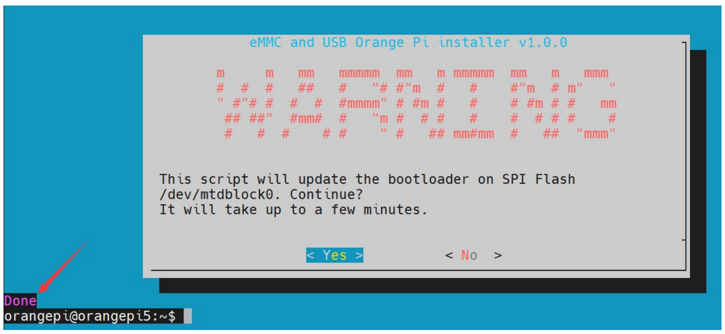
b. Then select **7 Install/Update the bootloader on SPI Flash**



c. Then select **<Yes>**



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



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

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

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
Orangepi3b_x.x.x_debian_bullseye_desktop_xfce_Linux5.10.160.img
```

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

1. For the method of using the serial port to log in to the terminal, please refer to

the instructions in the section on how to use the debugging serial port.
2. Use ssh to remotely log in to the Linux system, please refer to the instructions in the section of SSH remote login to the development board.
3. If HDMI, LCD and other display screens are connected, you can open a command line terminal on the desktop.

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

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

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

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

2) Then we can use the dd command to clear 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
```

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

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

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

```
sudo dd bs=1M if=Orangepi3b_x.x.x_debian_bullseye_desktop_xfce_Linux5.10.160.img of=/dev/nvme0n1 status=progress

sudo sync
```

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

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

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

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

a. 128GB NVMe SSD

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

b. 2TB NVMe SSD

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.8G  8.0K  3.8G   1% /dev
tmpfs           769M  1.4M  768M   1% /run
/dev/nvme0n1p2  1.9T  4.1G  1.8T   1% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/zram2      3.7G   76K  3.5G   1% /tmp
```

```
/dev/nvme0n1p1   256M    90M   166M   36% /boot
/dev/zram1       194M    15M   165M    9% /var/log
tmpfs            769M    60K   769M    1% /run/user/1000
tmpfs            769M    48K   769M    1% /run/user/0
```

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

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

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

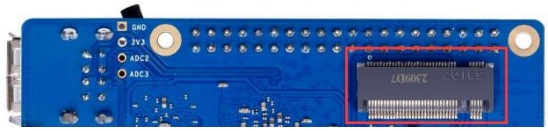
2. 6. 3. How to use balenaEtcher software to burn

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

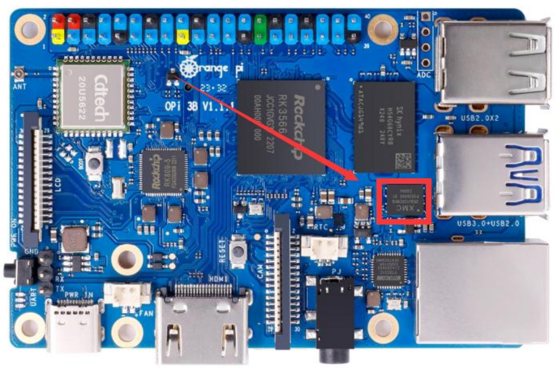
a. The M.2 2230 SSD is as follows

b. The M.2 2242 SSD is as follows

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



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



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

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

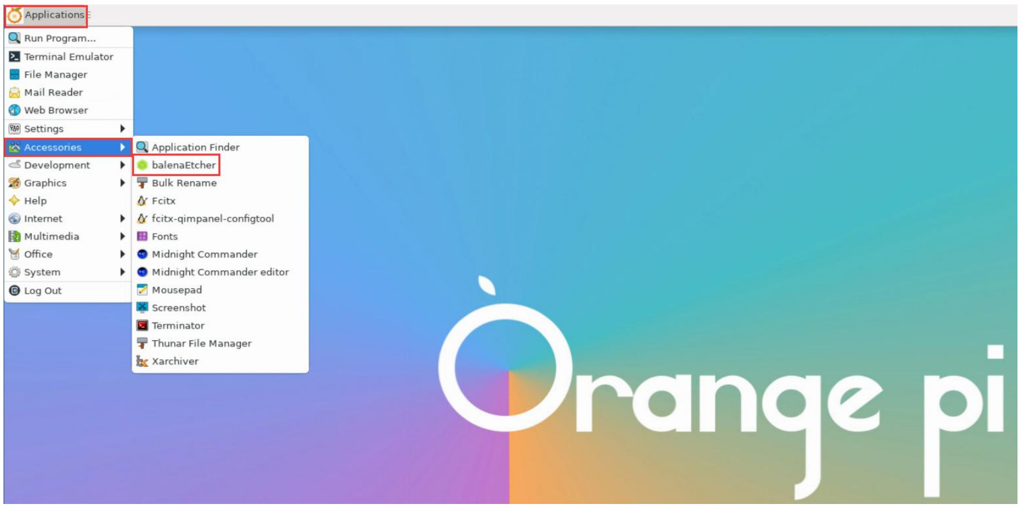
```
orangepi@orangepi:~/Desktop$ sudo fdisk -l | grep "nvme0n1"
Disk /dev/nvme0n1: 1.86 TiB, 2048408248320 bytes, 4000797360 sectors
```

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

```
orangepi@orangepi:~/Desktop$ lspci
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) The balenaEtcher has been pre-installed in the Linux image, and the opening method is as follows:

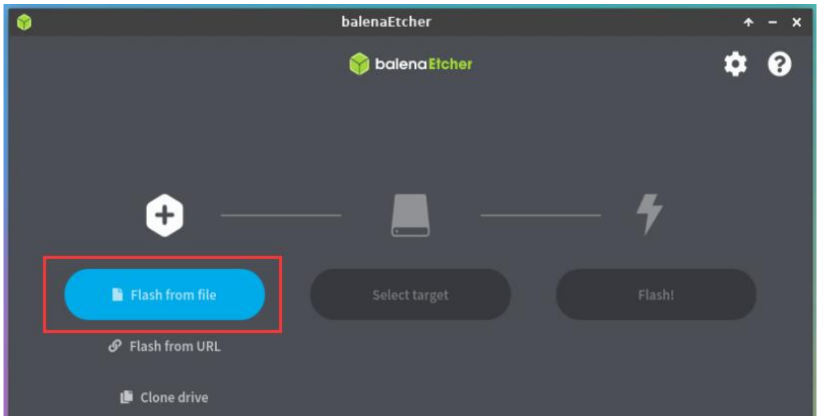


If it is not pre-installed, for how to download and install the arm64 version of balenaEtcher, please refer to the instructions in the section on how to download and install the arm64 version of balenaEtcher.

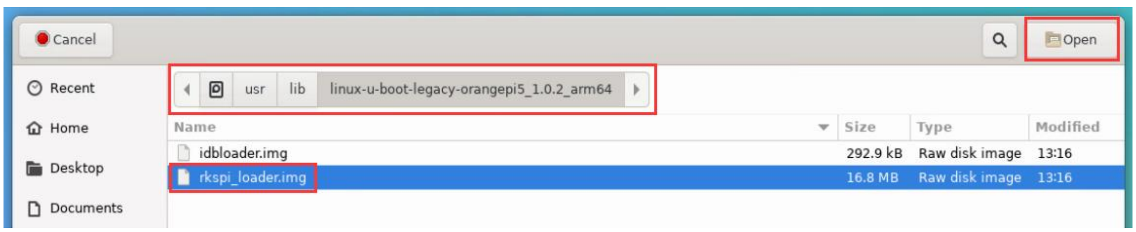
7) The interface after balenaEtcher is opened is as follows:

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

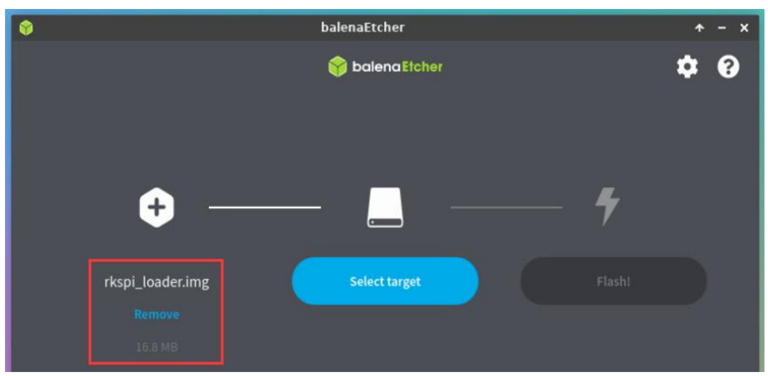
a. First click on **Flash from file**



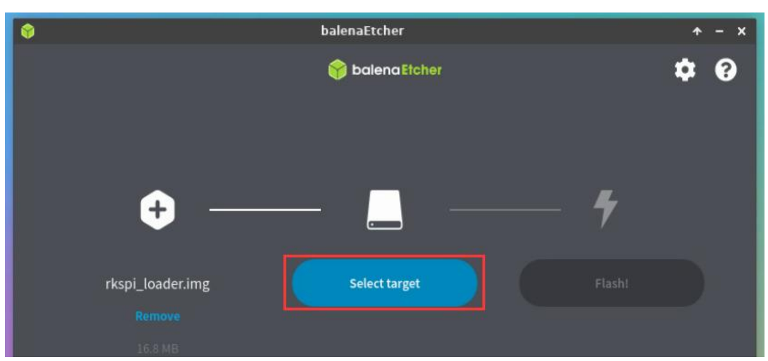
b. Then enter the **/usr/lib/Linux-u-boot-legacy-orangepi3b_1.x.x_arm64** directory, select **rkspi_loader.img**, and click **Open** to open



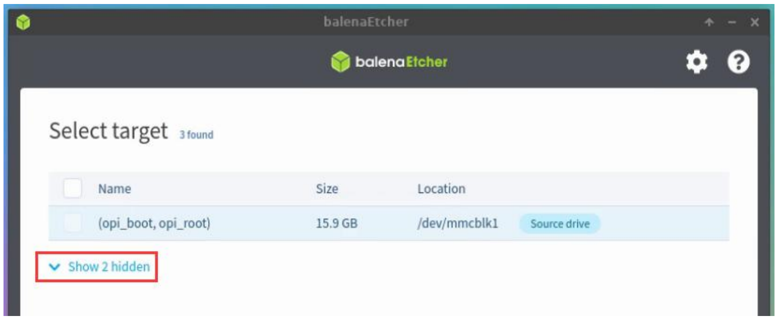
c. The interface after opening **rkspi_loader.img** is as follows:



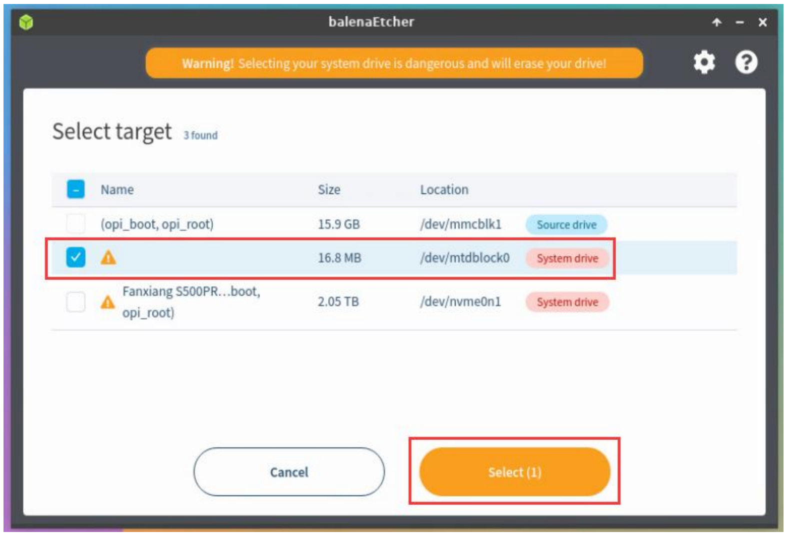
d. Then click **Select target**



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

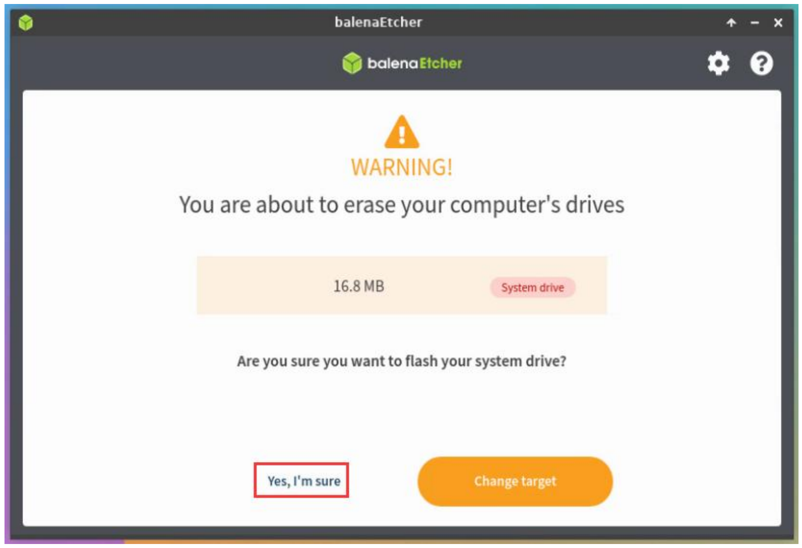


f. Then select the device name of SPI Flash **/dev/mtdblock0**, and click **Select**

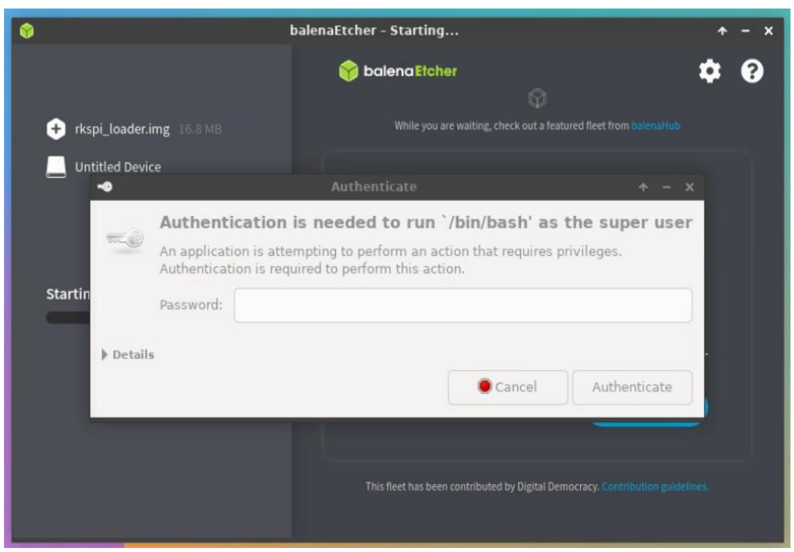


g. Then click **Flash**

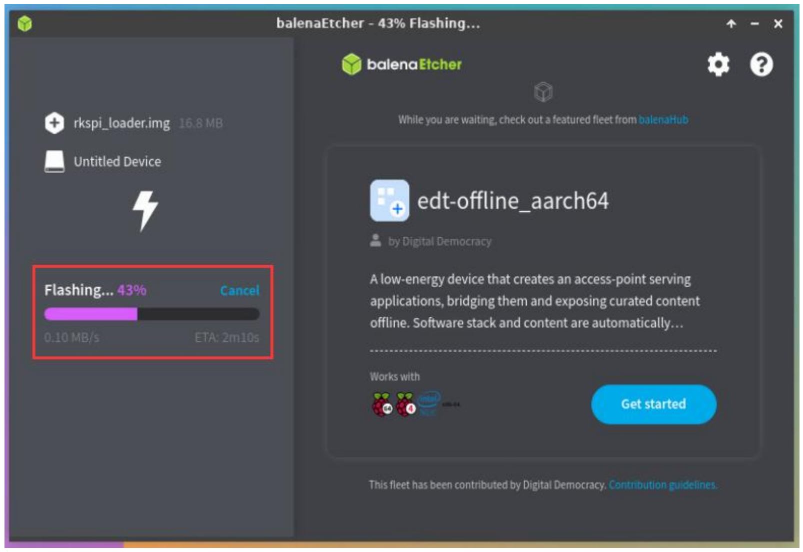
h. Then click **Yes, I'm sure**



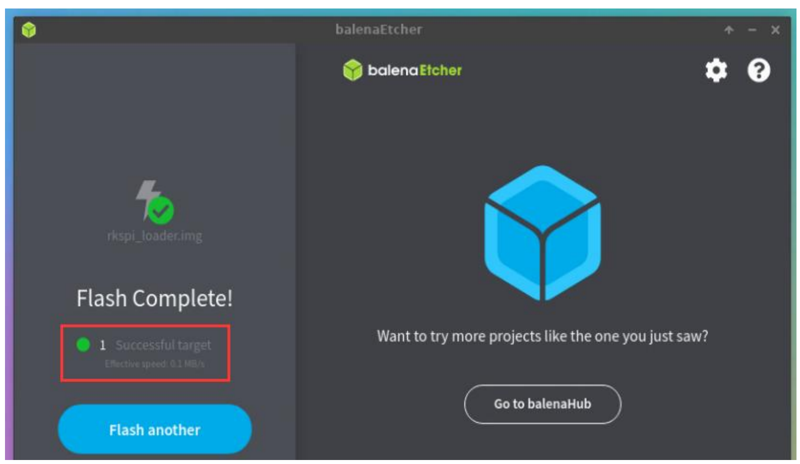
i. Then enter the password **orangepi** of the development board Linux system, and it will start burning the u-boot image into the SPI Flash



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

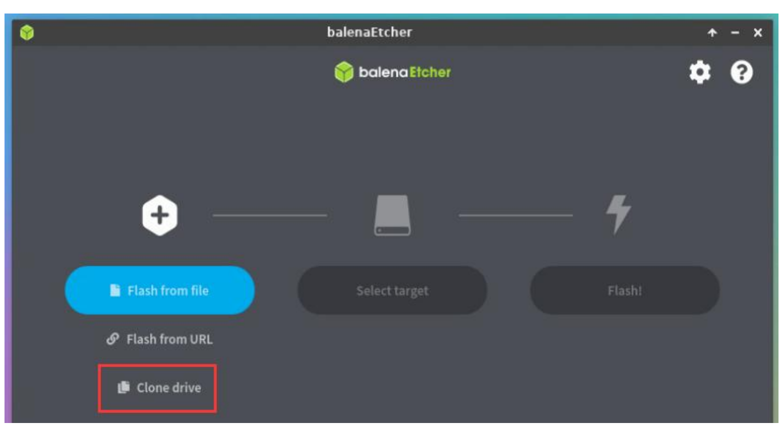


k. The display after burning is as follows:

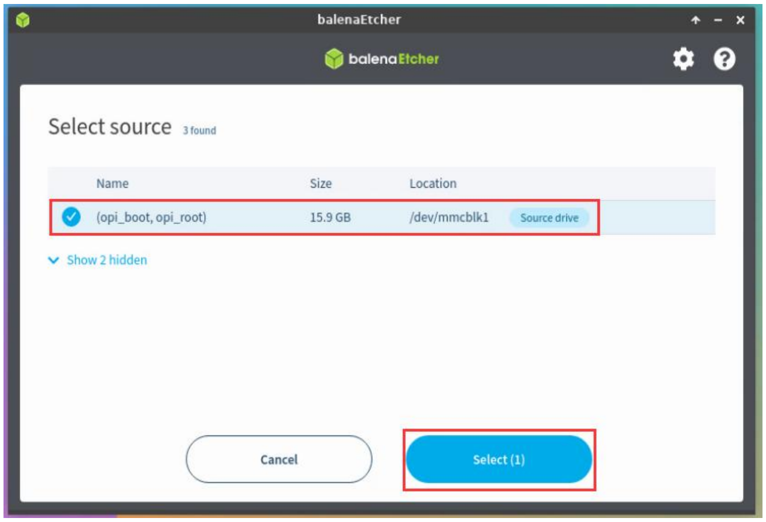


9) The method of burning the Linux system in the TF card to the NVMe SSD (this method is equivalent to cloning the system in the TF card to the NVMe SSD)

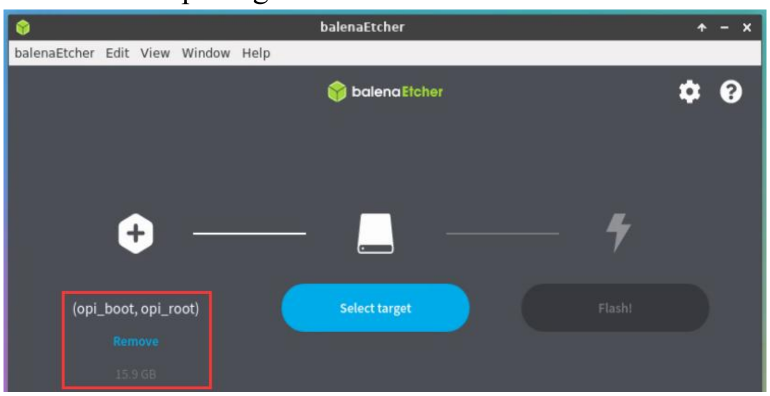
a. First click **Clone drive**



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



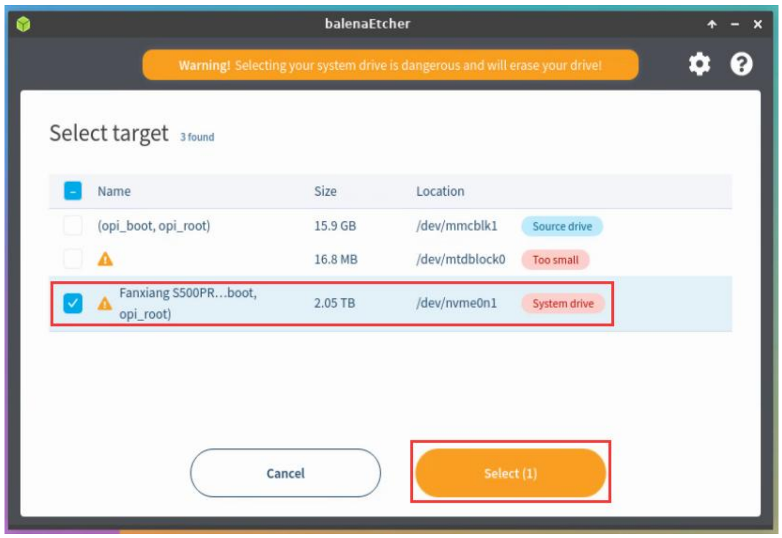
c. The interface after opening the TF card is as follows:



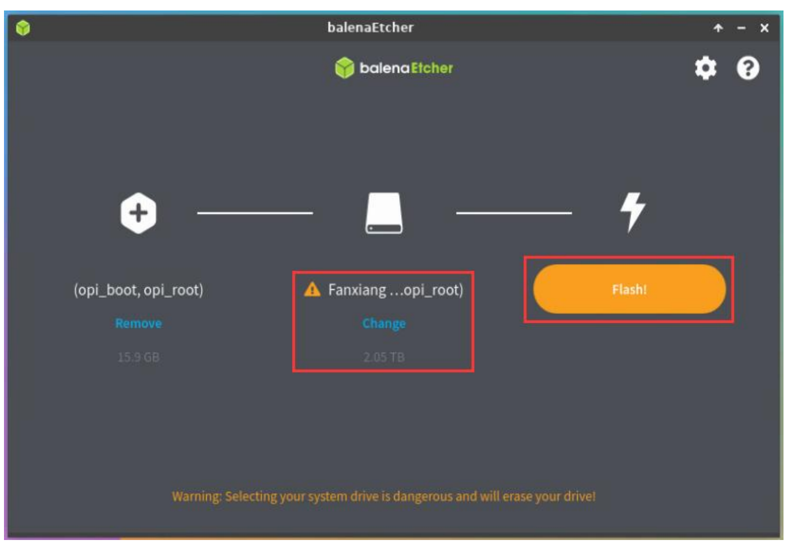
d. Then click **Select target**

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

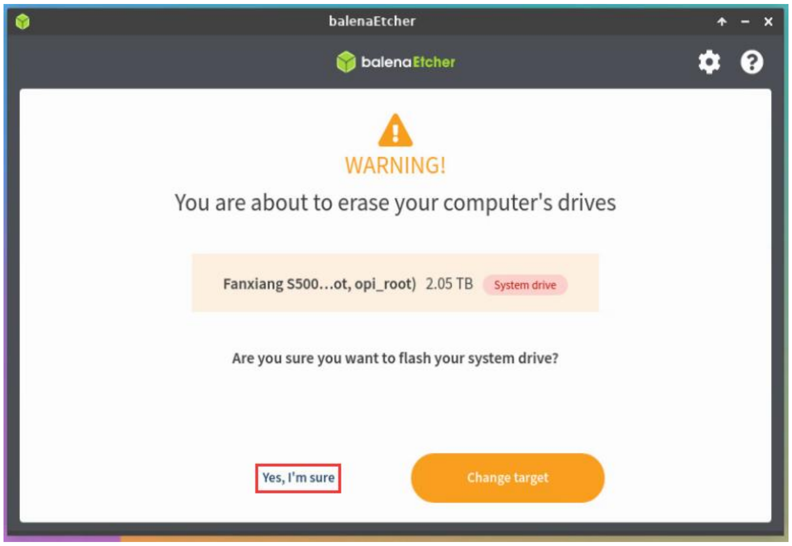
f. Then select the device name of the NVMe SSD **/dev/nvme0n1**, and click **Select**



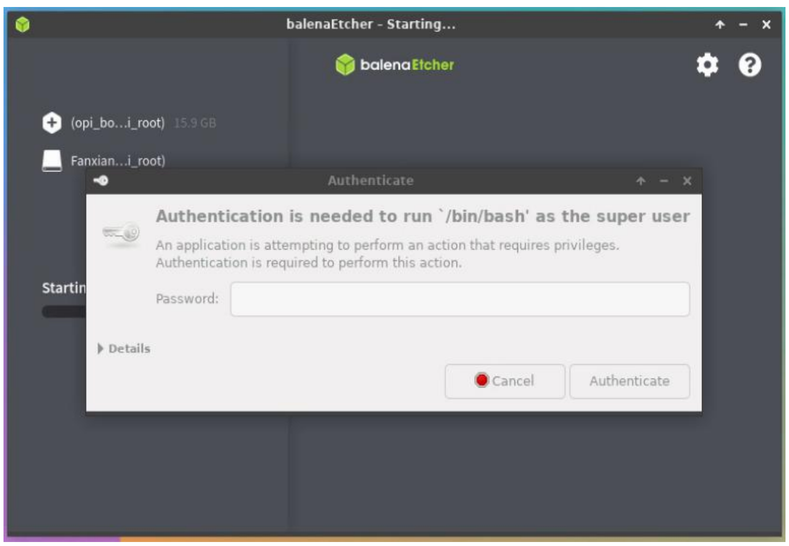
g. Then click **Flash**



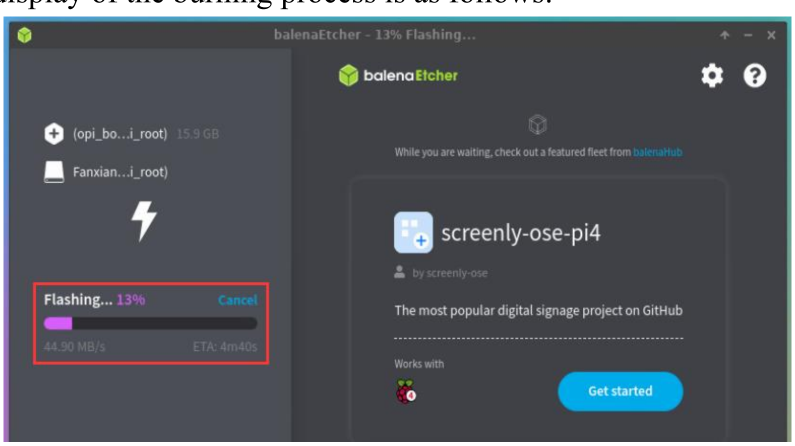
h. Then click **Yes, I'm sure**

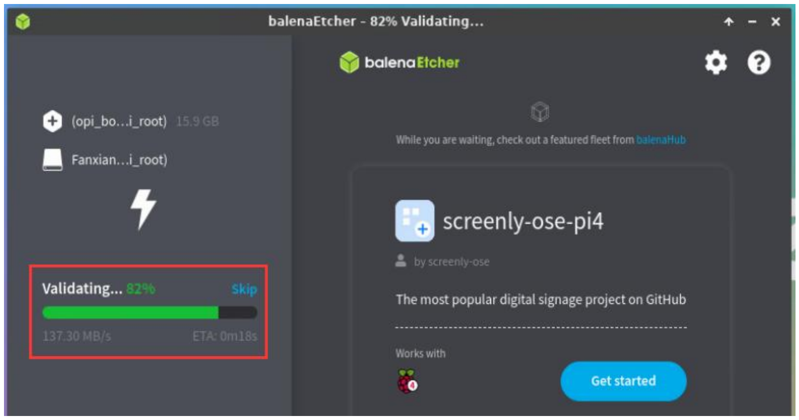


i. Then enter the password orangepi of the Linux system on the development board, and it will start burning the Linux image to the SSD

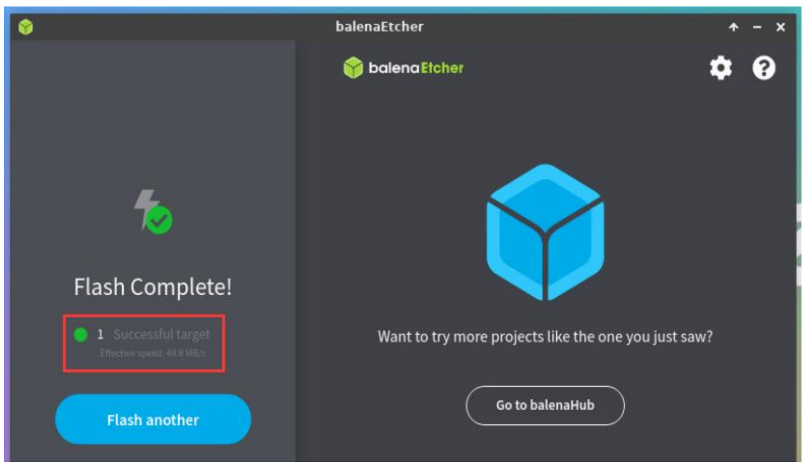


j. The display of the burning process is as follows:





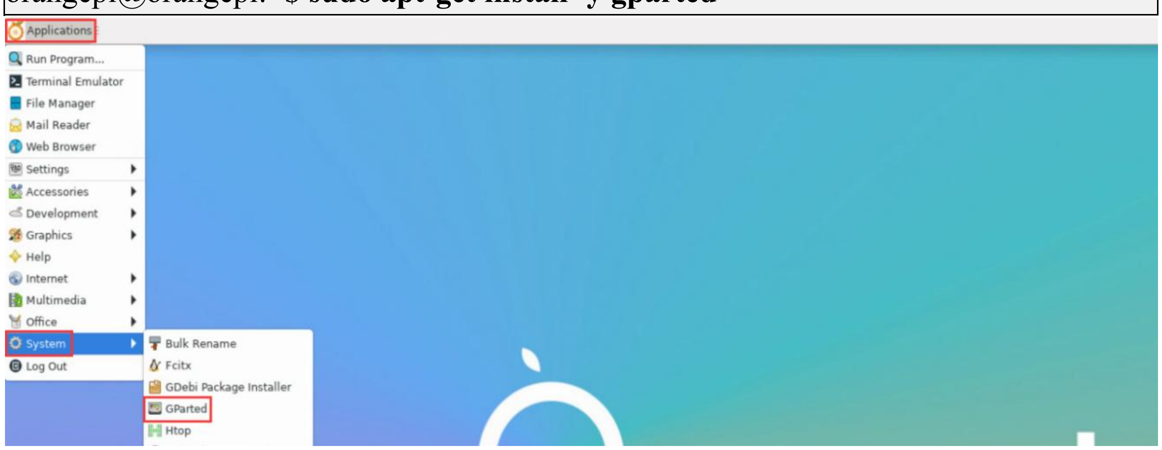
k. The display after burning is as follows:



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

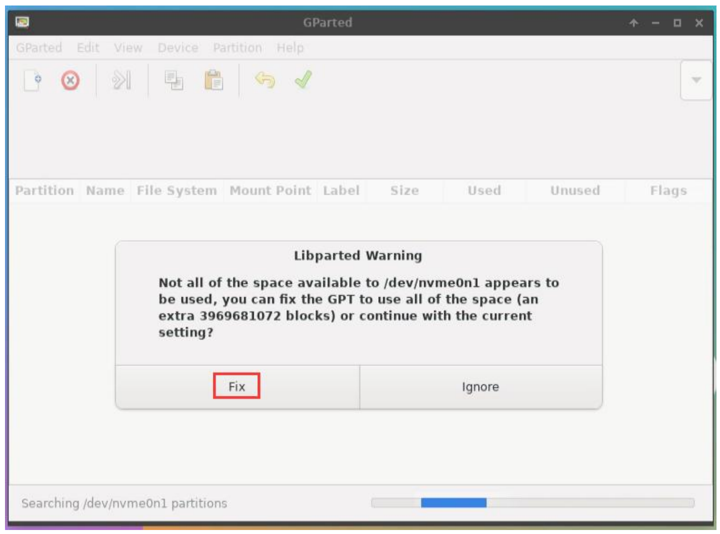
a) Open **GParted** first, if the system does not have Gparted pre-installed, please use the apt command to install it

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

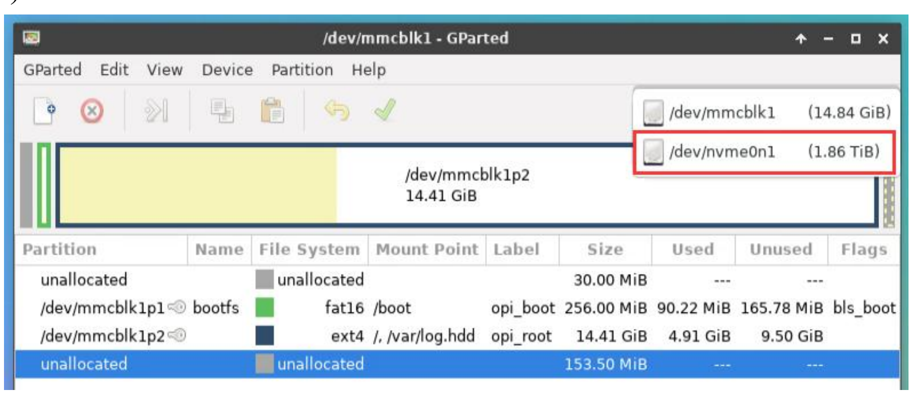


b) Then enter the password orangepi of the Linux system, and click **Authenticate**

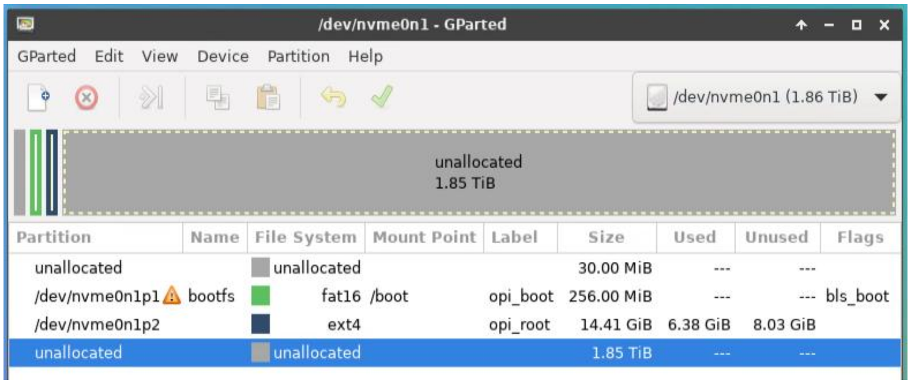
c) Then click **Fix**



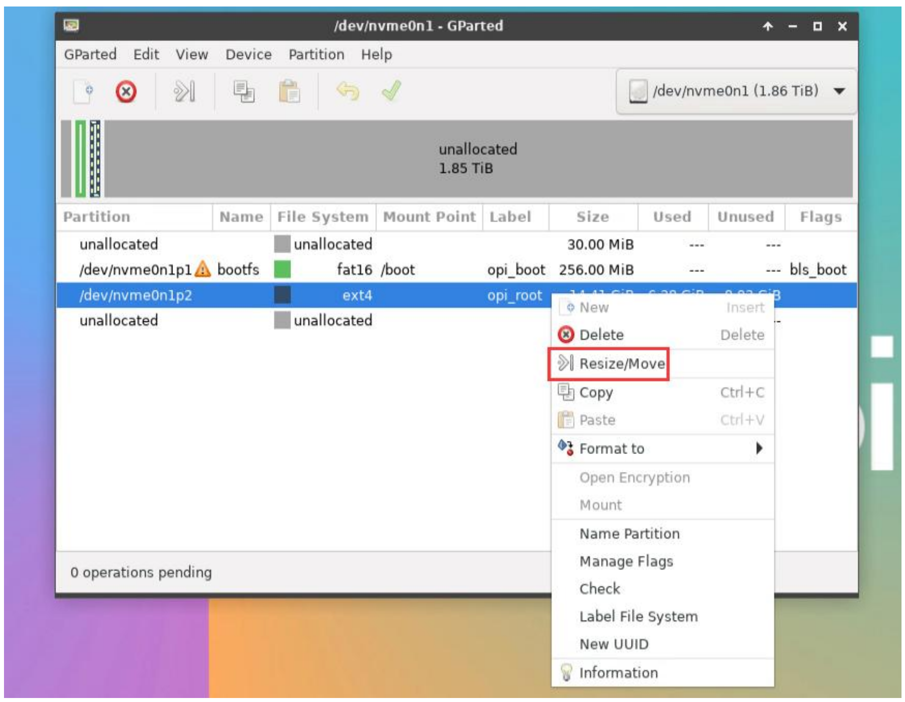
d) Then select NVMe SSD



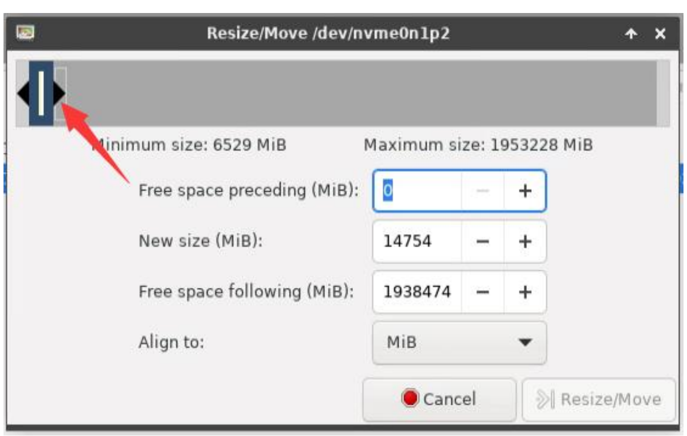
e) The display interface after selecting NVMe SSD is as follows:



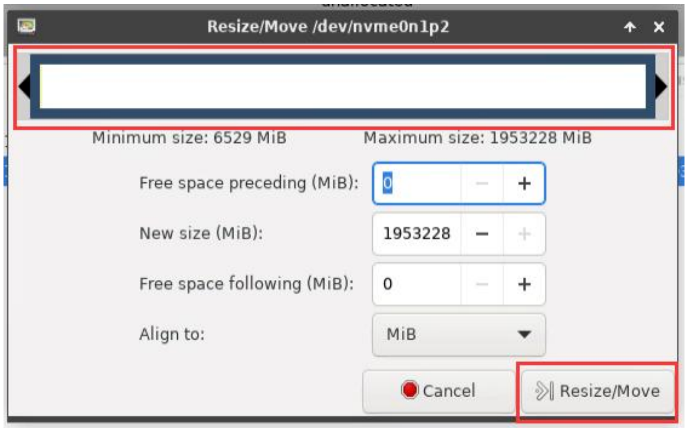
f) Then select the **/dev/nvme0n1p2** partition, click the right button again, and then select **Resize/Move**



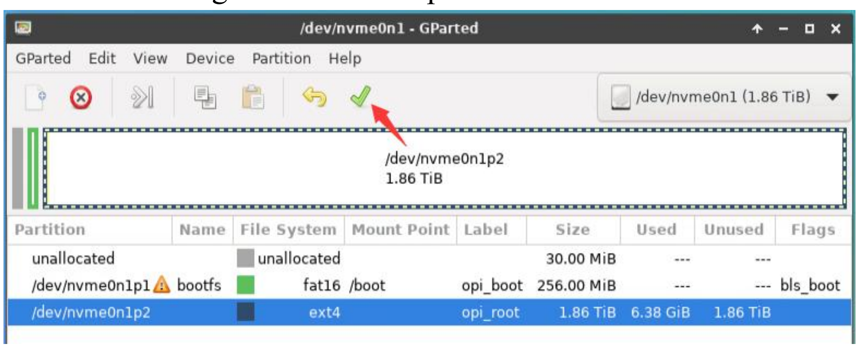
g) Then drag the capacity to the maximum at the position shown in the figure below



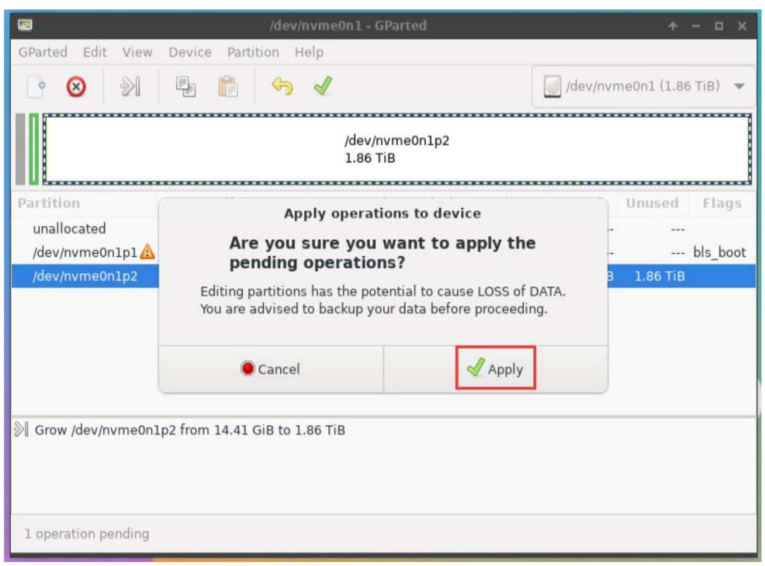
h) Then click **Resize/Move**



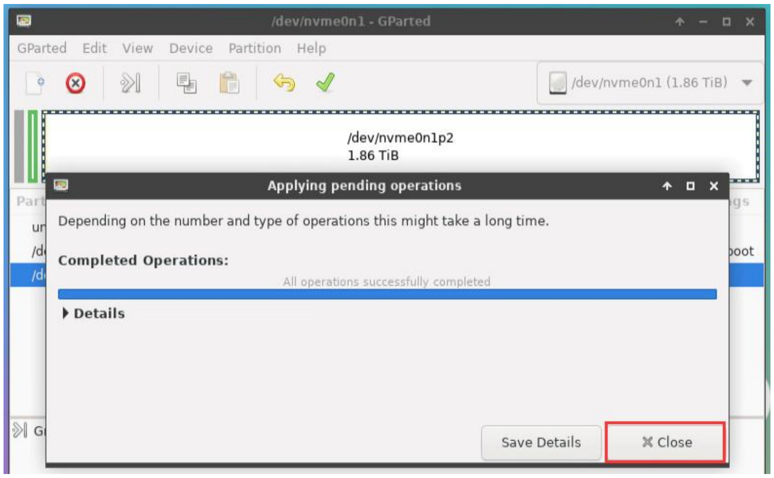
i) Then click the green √ in the position below



j) Then click **Apply**



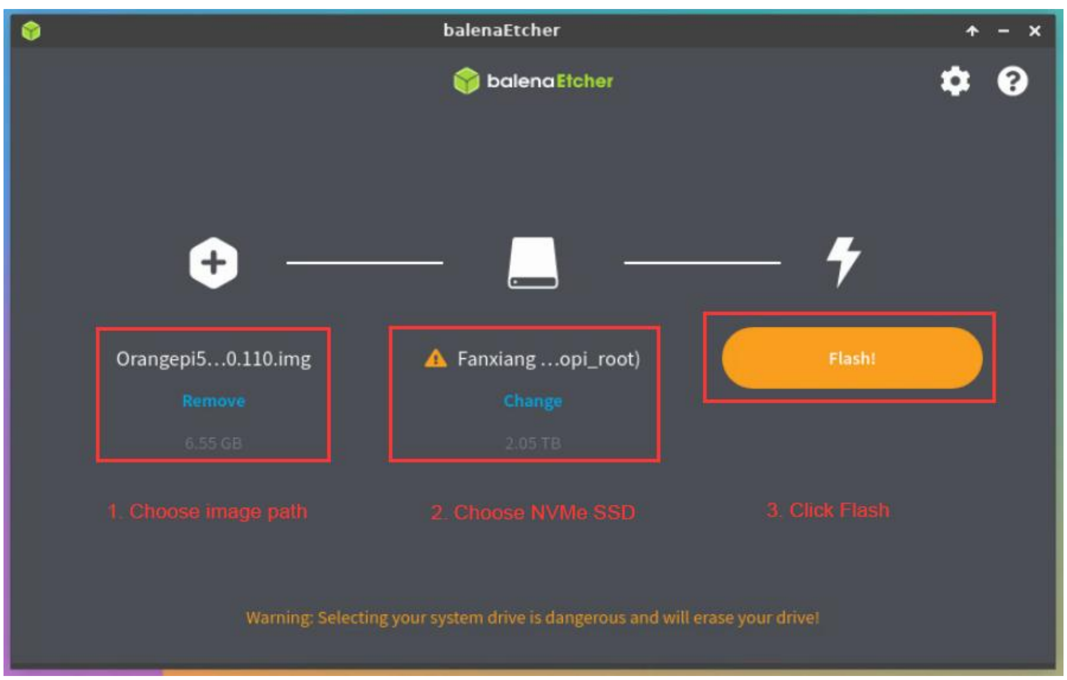
k) Then click **Close** to close



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

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

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



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

2. 7. How to burn Android image to TF card

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

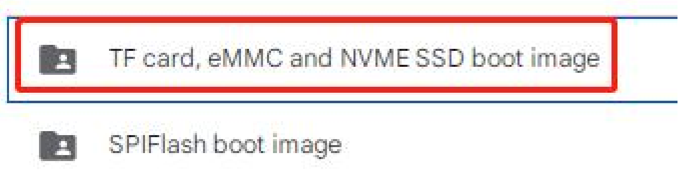
1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card of SanDisk and other brands

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

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

4) Then download the Android image from **Orange Pi's 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 startup image** folder to download



b. After entering the **TF card and eMMC startup image** folder, 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

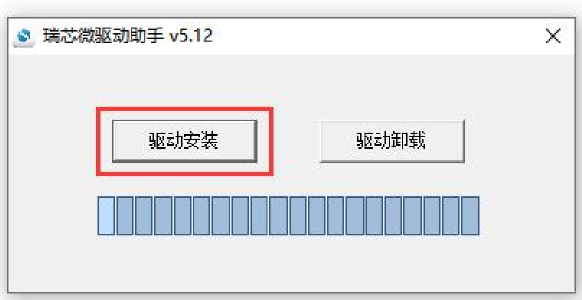
b) If you want to use lcd screen, please choose image with lcd

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

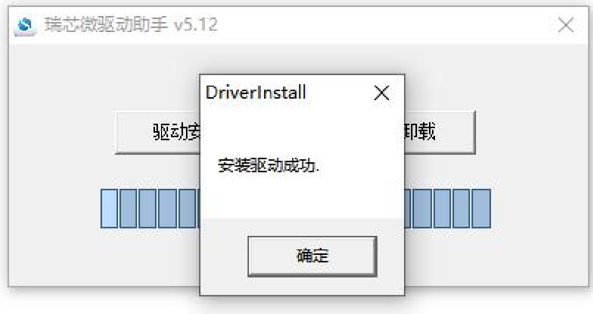
6) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

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

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



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

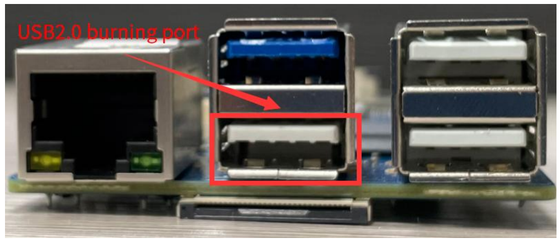


8) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

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

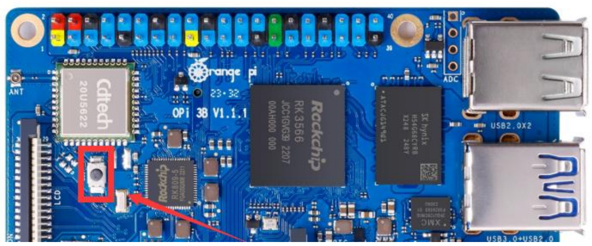
10) Then start burning the Android image to the TF card

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

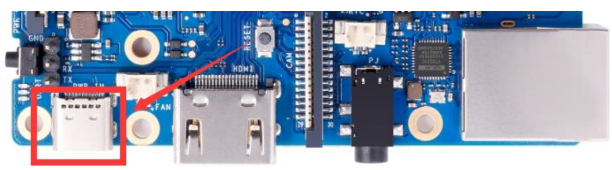


b. Then make sure that the development board is not inserted into the TF card and not connected to the power supply

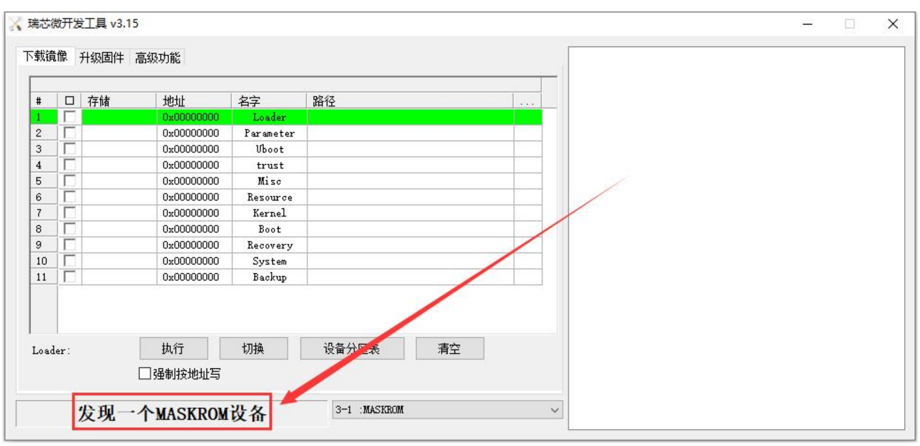
c. Then press and hold the MaskROM button on the development board, the position of the MaskROM button on the development board is shown in the figure below:



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

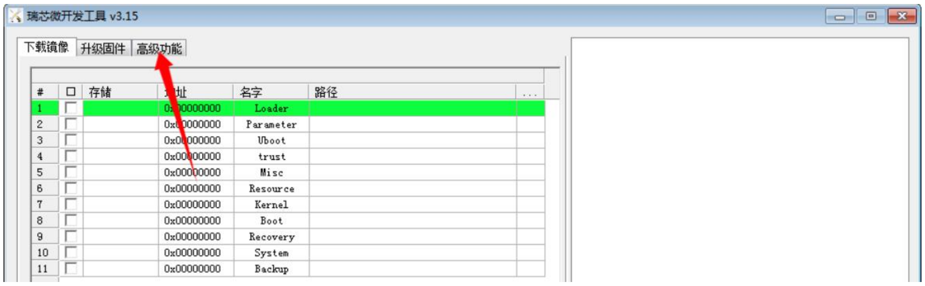


e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"

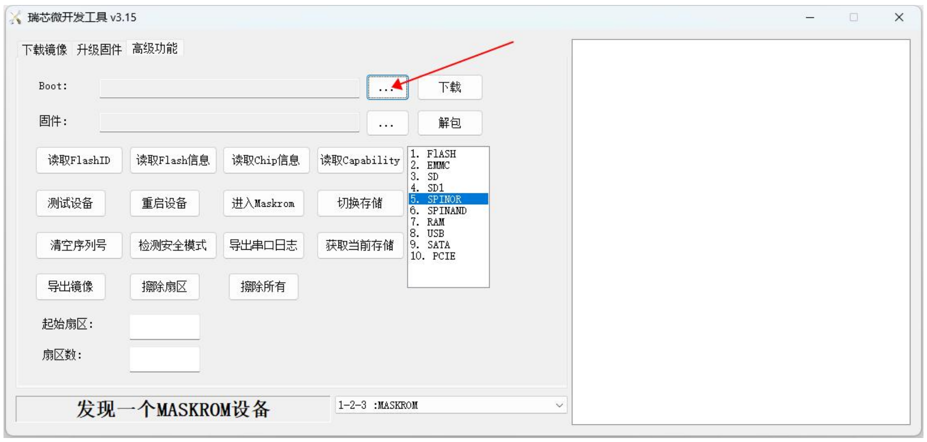


f. Then insert the TF card into the development board

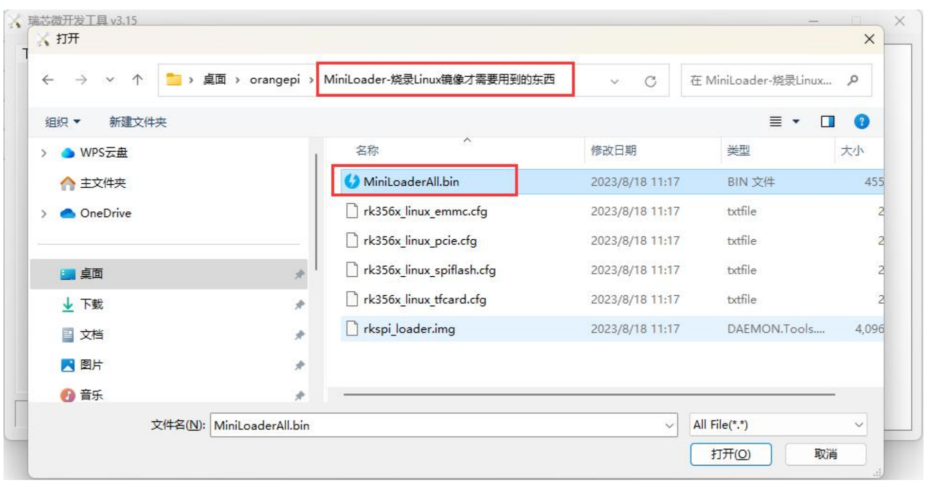
g. Then please select **advanced features**



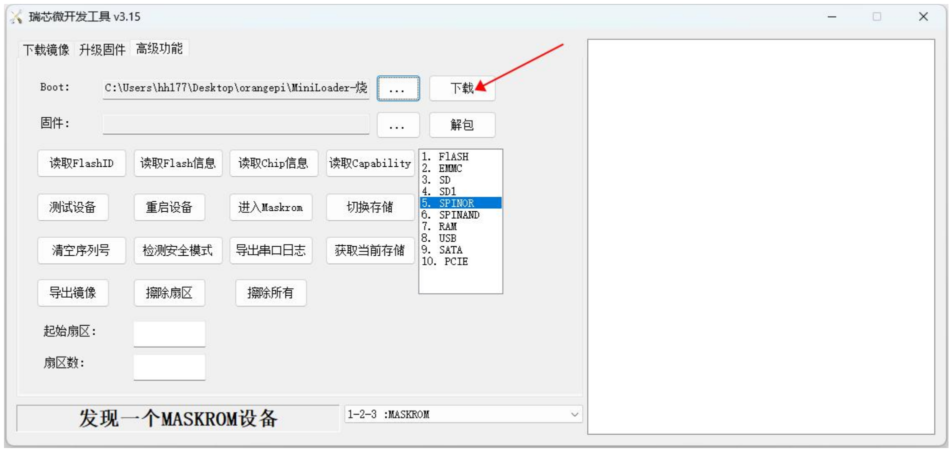
h. Then click the position shown in the figure below



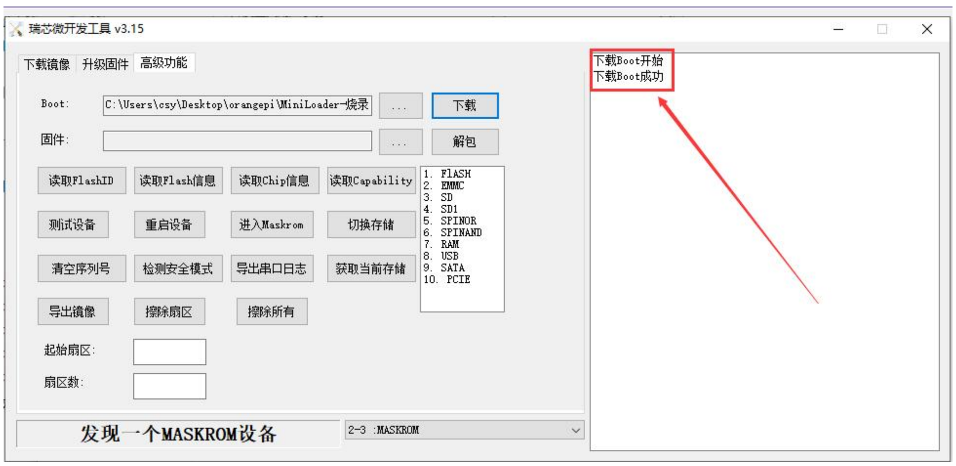
i. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and click to open



j. Then click **download**

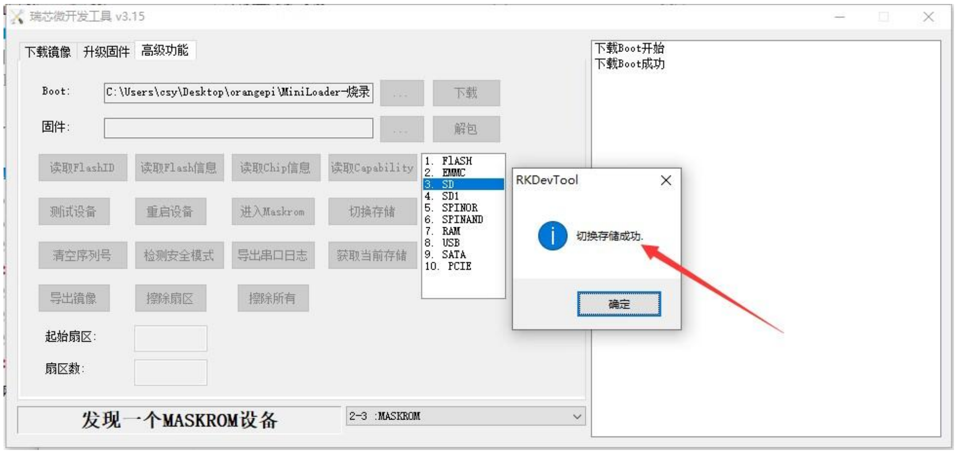


k. The display after downloading **MiniLoaderAll.bin** is shown in the figure below



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

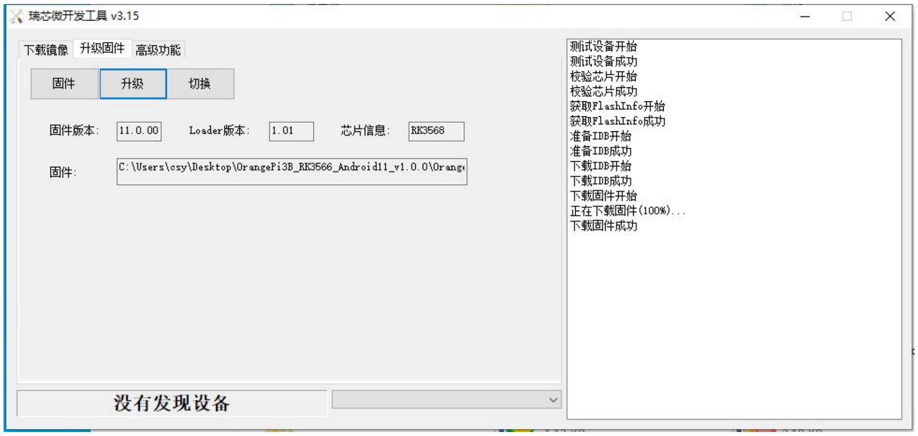
m. The display of successful switching is shown in the figure below



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

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

p. Finally, click the "**Upgrade**" button to start burning, and the log during the burning process is shown in the figure below. After burning is completed, the Android system will start automatically.



2.7.2. How to use SDDiskTool to burn Android image to TF card

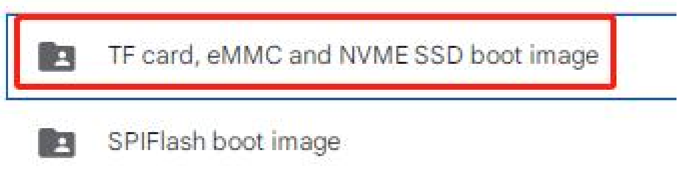
1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card of SanDisk and other brands

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

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

4) Then download **the Android11 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 startup image** folder to download



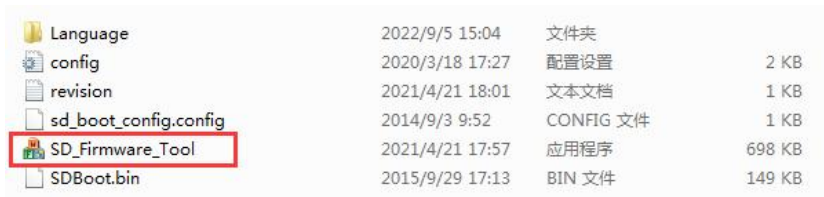
b. After entering the **TF card and eMMC boot image** folder, you can see the following two images, the difference between them is:

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

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

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

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



7) After opening **SDDiskTool**, if the TF card is recognized normally, the inserted disk device will be displayed in the "**Select Removable Disk Device**" column. P**lease 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, click the **restore disk button** in SDDiskTool, or use the **SD Card Formatter** mentioned above to format the TF card

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

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

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

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

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

2. 8. How to burn Android image to eMMC

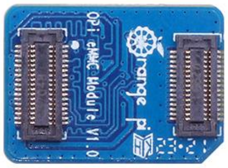
Note, after burning the image into eMMC, if the test finds that it cannot be started, please clear the SPIFlash and try again. For the method of clearing SPIFlash, please refer to the method of using RKDevTool to clear SPIFlash.

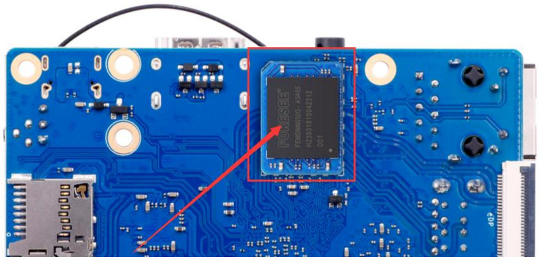
2. 8. 1. Method of burning Android image into eMMC through USB2.0 burning port

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

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

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



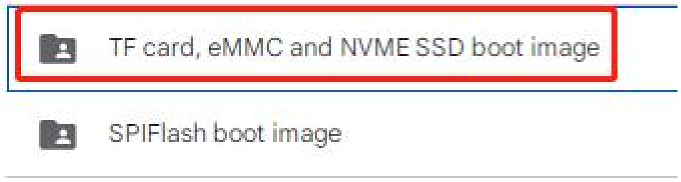


2) You also need to prepare a good quality USB2.0 male-to-male data cable

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

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

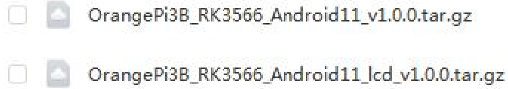
   a. After opening the download link of the Android image, you can see the following two types of Android images, please select the image in the **TF card and eMMC startup image folder** to download



   b. After entering the **TF card and eMMC startup image folder**, 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 lcd screen, please choose image with lcd



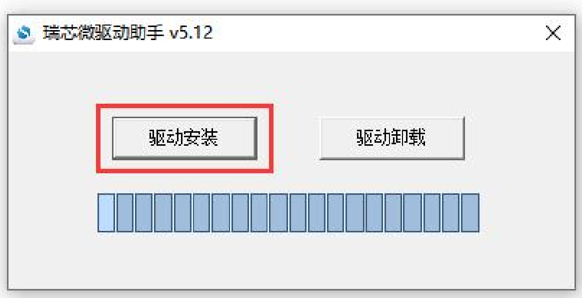
5) Then use decompression software to decompress the compressed package of the

downloaded Android image. Among the decompressed files, the file ending with ".img" is the Android image file, and the size is more than 1GB

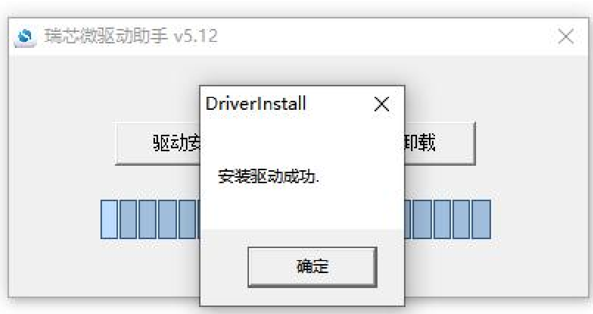
6) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

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

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

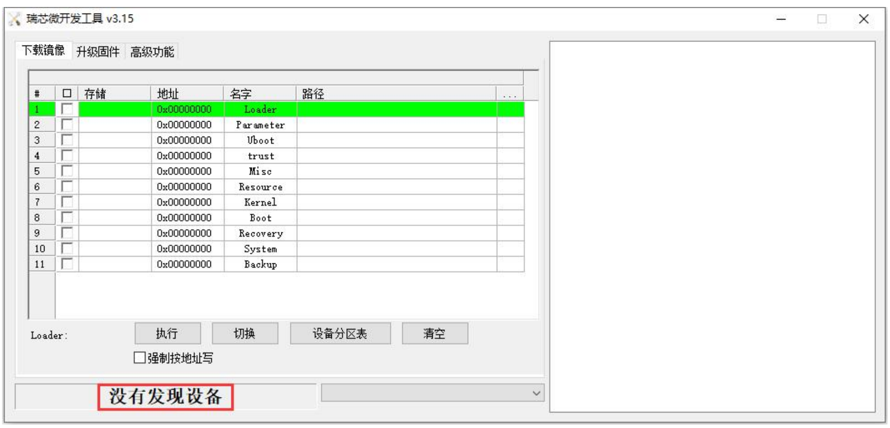


b. After waiting for a period of time, a pop-up window will prompt "**driver installed successfully**", and then click the "**OK**" button.



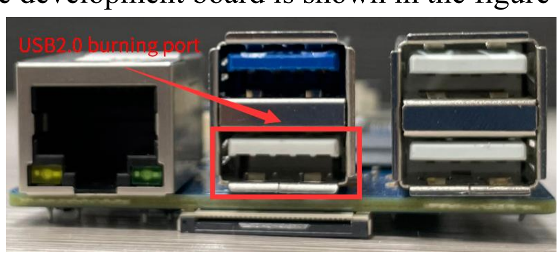
8) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

9) After opening the **RKDevTool** burning tool, because the computer is 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**"



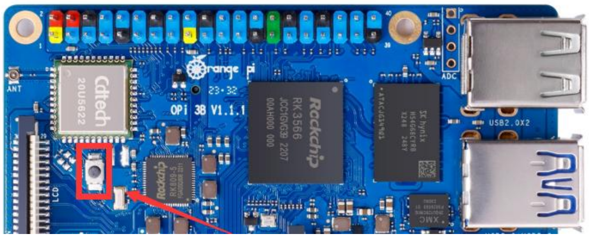
10) Then start burning the Android image into eMMC

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

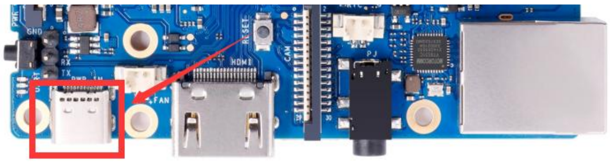


b. Then make sure that the development board is not inserted into the TF card and not connected to the power supply

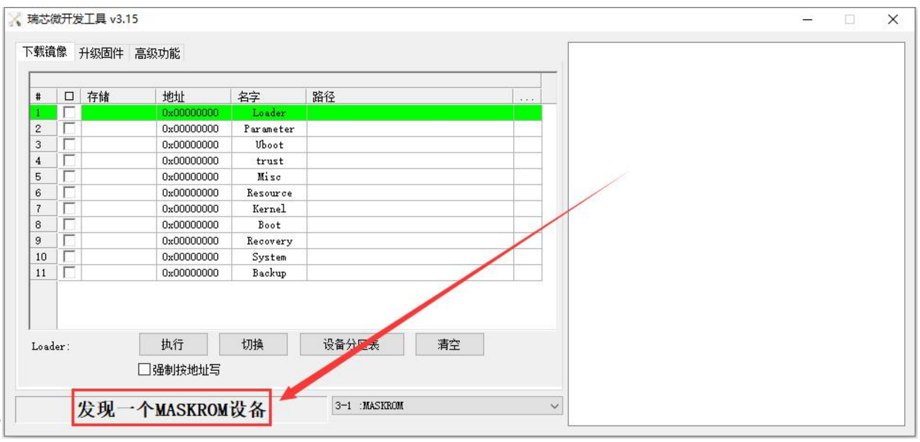
c. Then press and hold the MaskROM button on the development board, the position of the MaskROM button on the development board is shown in the figure below:



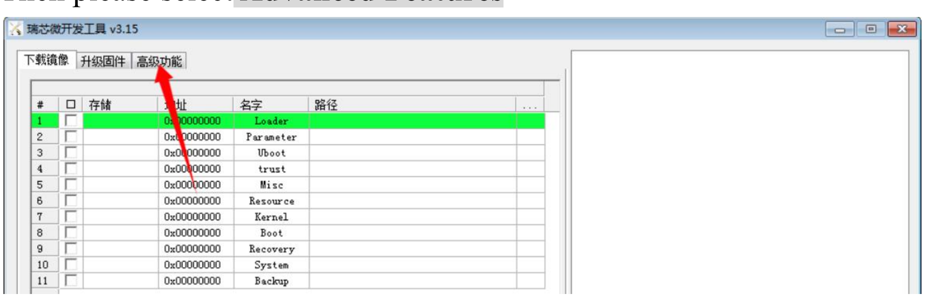
d. Then connect the power supply of the Type-C interface to the development board



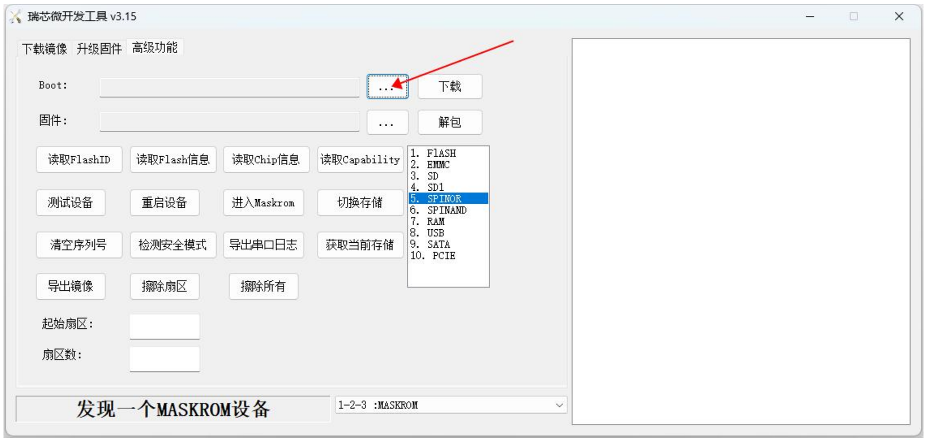
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt **"found a MASKROM device"**



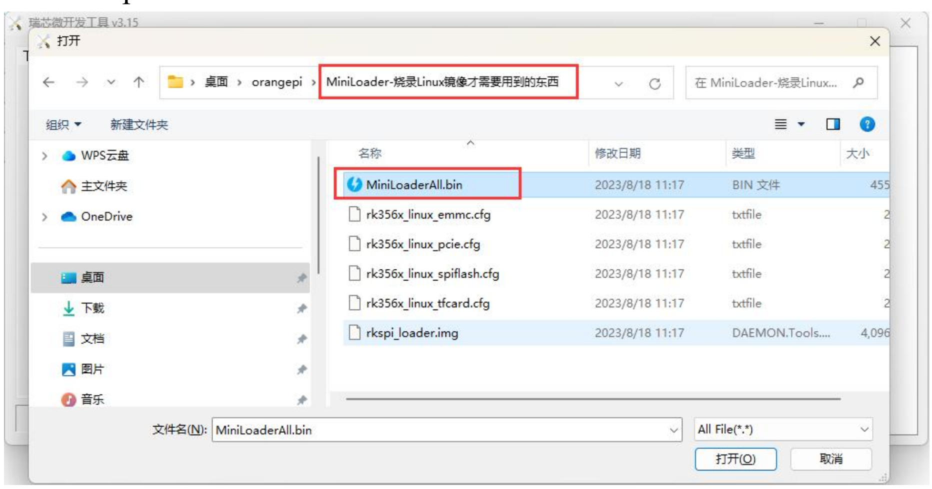
f. Then please select **Advanced Features**



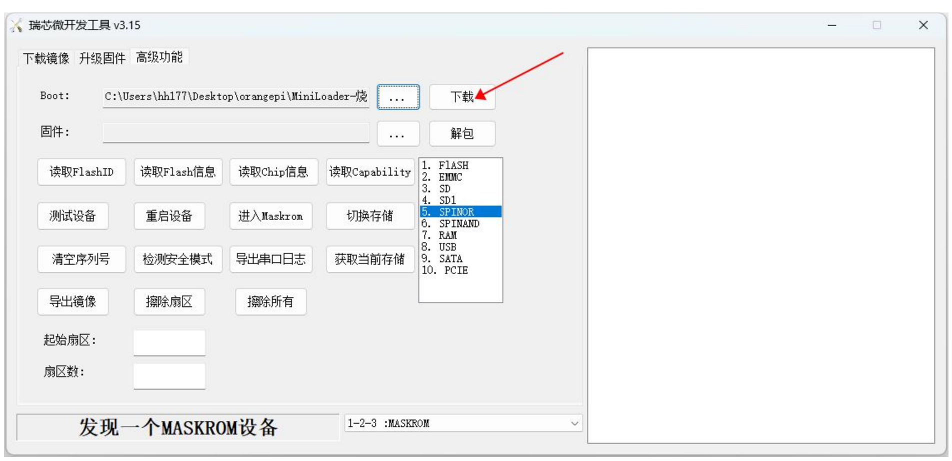
g. Then click the position shown in the figure below



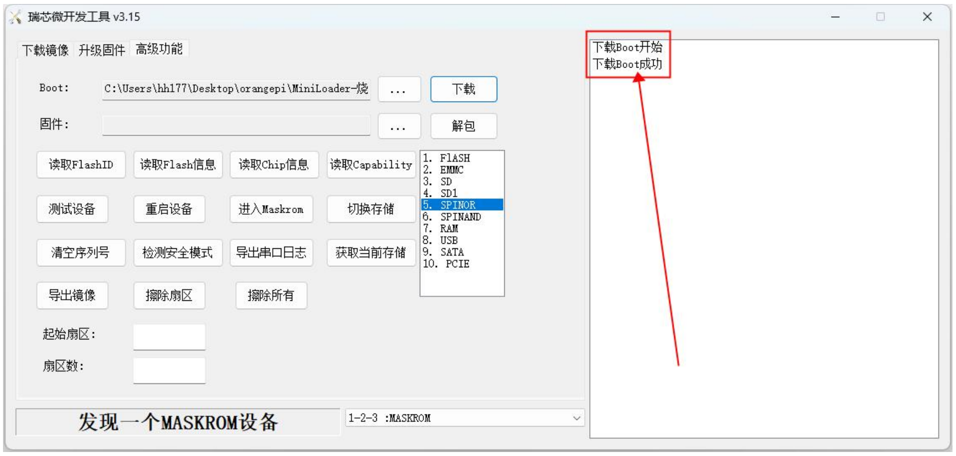
h. Select **MiniLoaderAll.bin** in the MiniLoader folder downloaded earlier, and click to open.



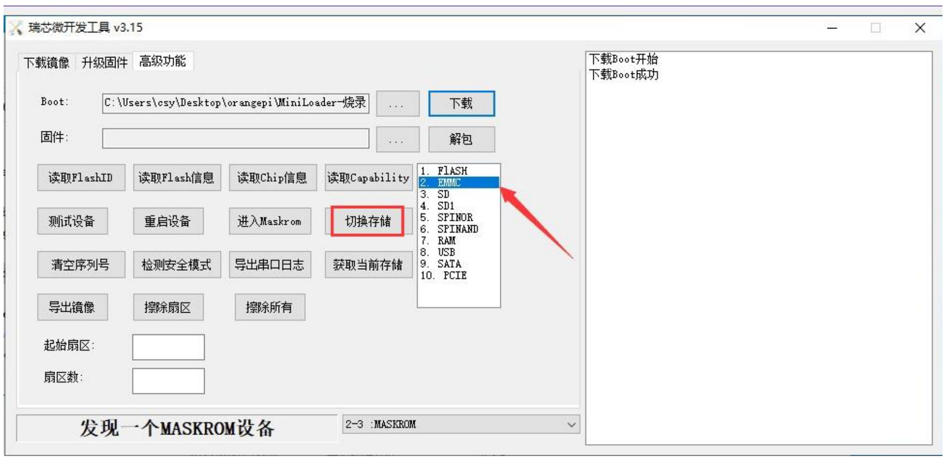
i. Then click **Download**



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 display of successful switching is shown in the figure below

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

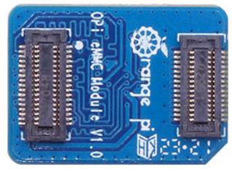
n. Then click the "**Firmware**" button to select the path of the Android image that needs to be burned

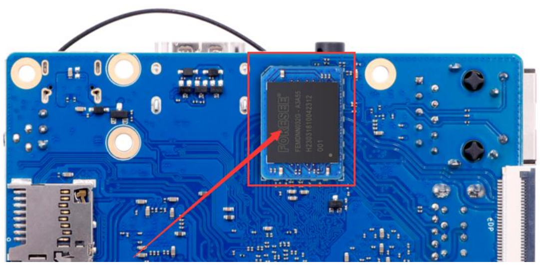
o. Finally, click the "**Upgrade**" button to start burning, and the log during the burning process is shown in the figure below. After burning is completed, the Android system will start automatically.

2. 8. 2. How to burn Android11 image into eMMC via TF card

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

1) The development board reserves an eMMC expansion interface. Before programming the system to eMMC, you first need to purchase an eMMC module that matches the eMMC interface of the development board. Then install the eMMC module to the development board. The eMMC module and the method of plugging into the development board are as follows:





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

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

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

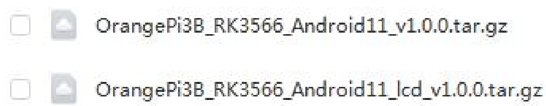
5) Then download the Android image from **Orange Pi's download page**

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 startup image folder** to download

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

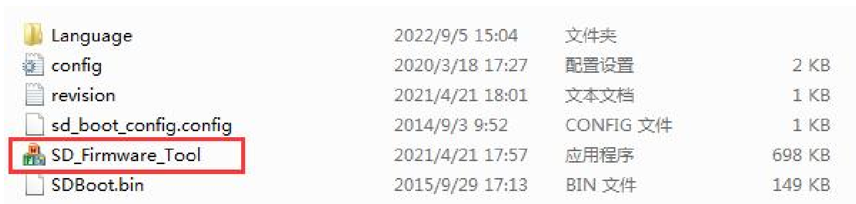
   a) The first image is dedicated to HDMI display and supports 4K display. If you don't use LCD screen, please download the image without lcd

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



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

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



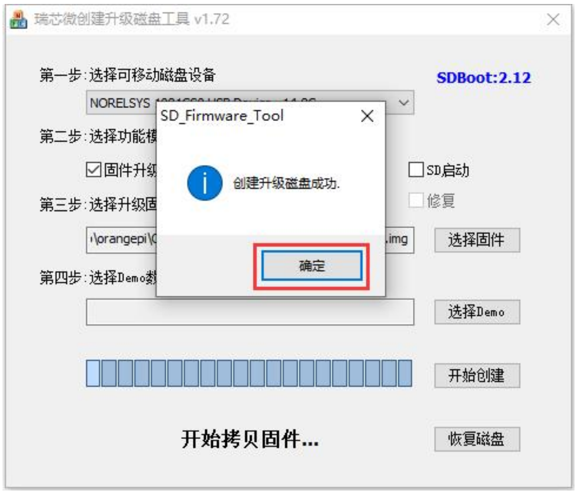
8) After opening **SDDiskTool**, if the TF card is recognized normally, the inserted disk device will be displayed in the "**Select Removable Disk Device**" column. **Please make sure that the displayed disk device is consistent with the drive letter of the TF card you want to burn**, if there is no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

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

10) Then start to write the Android image into the TF card

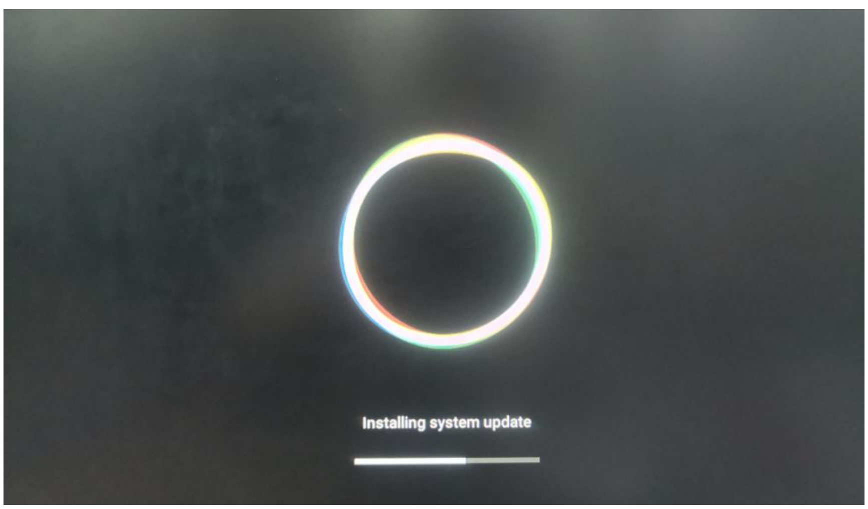
a. First confirm that the displayed drive letter is the drive letter corresponding to the TF card under "**Select Removable Disk Device**"
b. Then select "**Firmware Upgrade**" in "**Select Function Mode**"
c. Then select the path of the Android firmware in the "**Select Upgrade Firmware**" column
d. Finally click the "**Start Create**" button to start burning

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

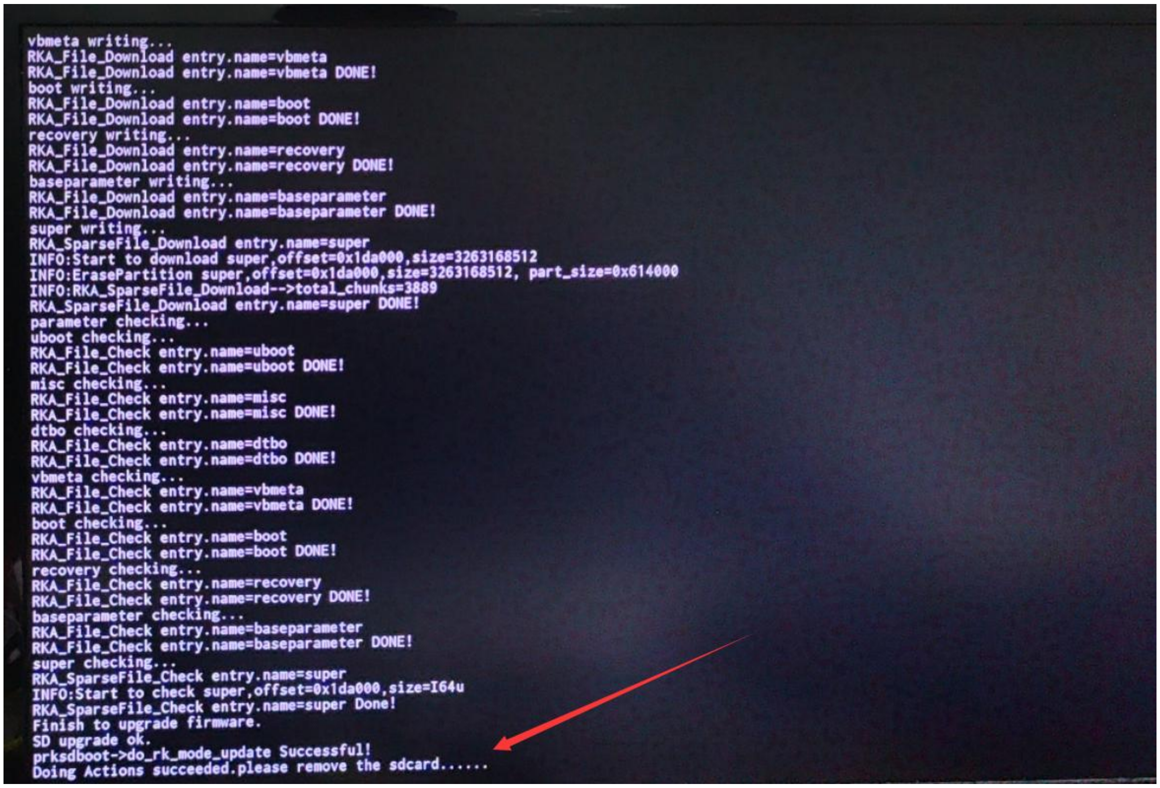


12) Then pull out the TF card from the computer and insert it into the development board. After the development board is powered on, it will automatically start burning the Android image in the TF card to the eMMC of the development board.

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



14) 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.



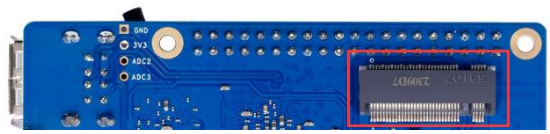
2. 9. How to burn Android image to SPIFlash+NVMe SSD

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

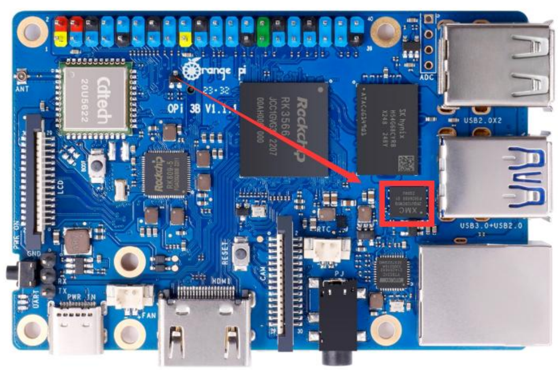
1) First, you need to prepare an NVMe SSD solid state drive
    a. The M.2 2230 SSD is as follows

    b. The M.2 2242 SSD is as follows

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



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

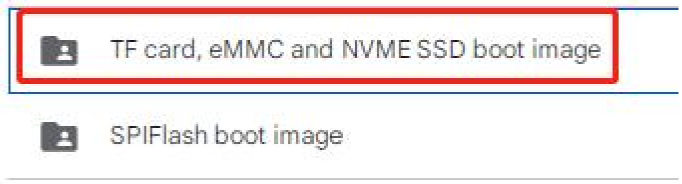


4) You also need to prepare a good quality USB2.0 male-to-male data cable

5) Then download Rockchip driver **DriverAssitant_v5.12.zip** and burning tool **RKDevTool_Release_v3.15.zip** from **Orange Pi's 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 S**PIFlash-NVME SSD boot image folder** to download

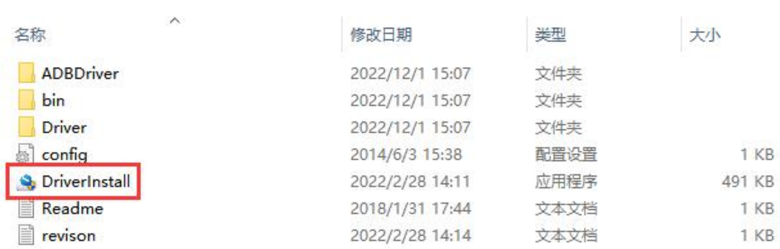


b. After entering the **SPIFlash-NVME SSD boot image folder**, you can see the following two images. Their differences are:

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

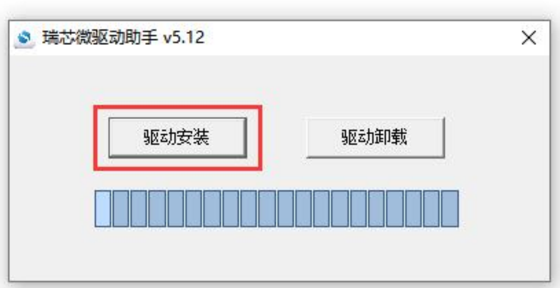
b) If you want to use LCD screen, please choose image with lcd

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

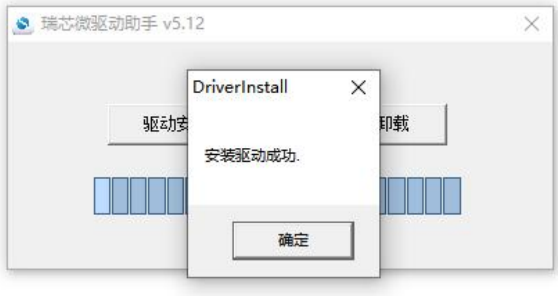


8) After opening **DriverInstall.exe**, the steps to install the Rockchip driver are as follows

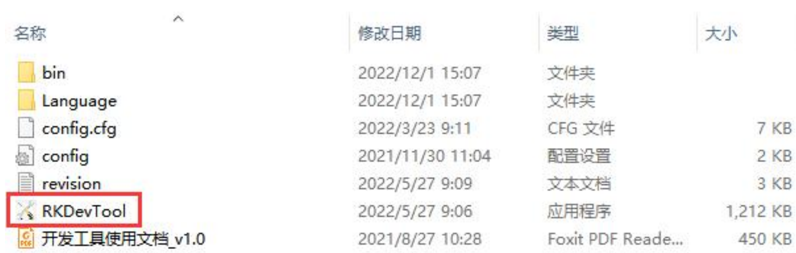
a. Click the "**Driver Installation**" button



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



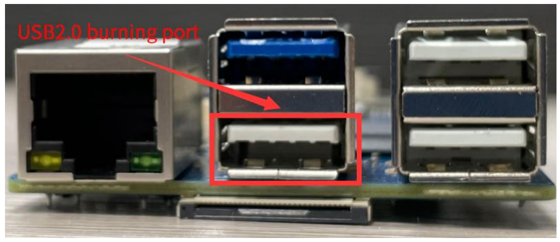
9) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it



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

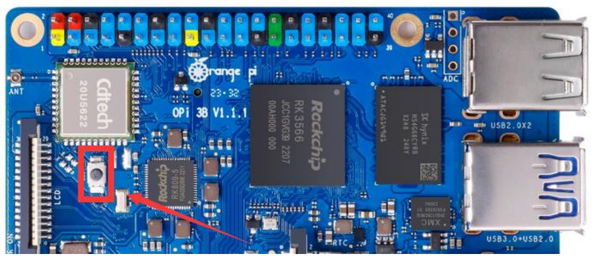
11) Then start burning the Android image to SPIFlash+NVMe SSD

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

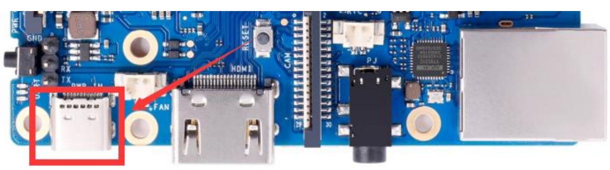


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

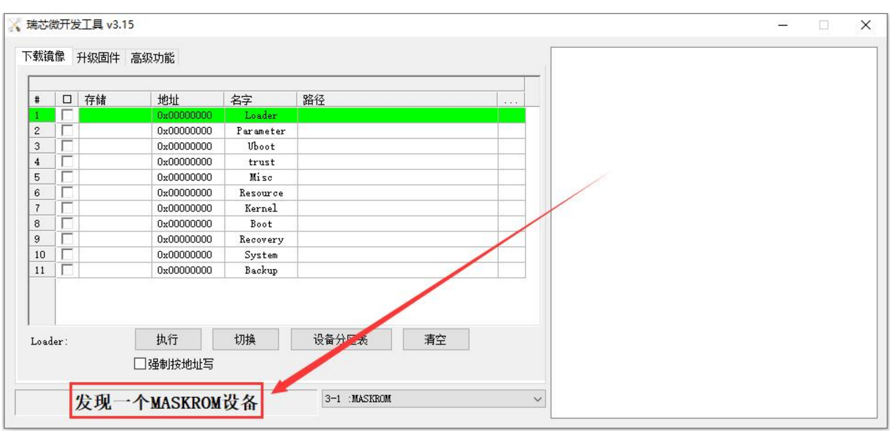
c. Then press and hold the MaskROM button on the development board, the position of the MaskROM button on the development board is shown in the figure below:



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



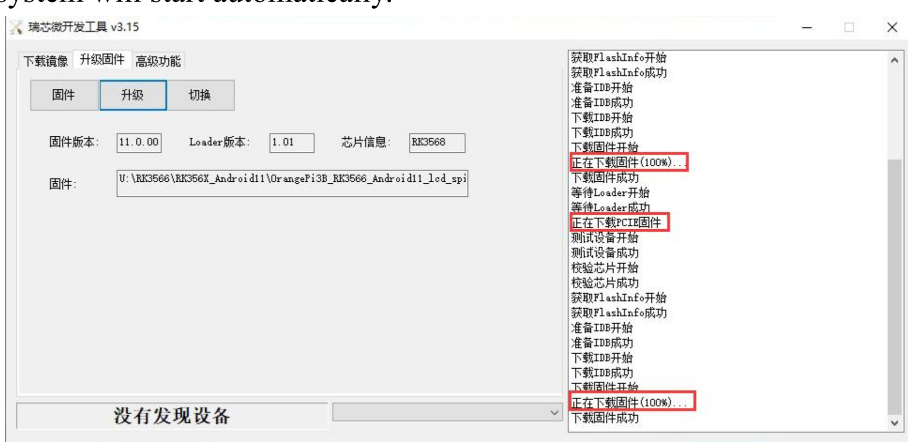
e. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



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

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

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



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

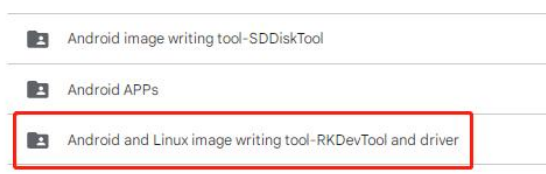
Note that all operations below are performed on a Windows computer.

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

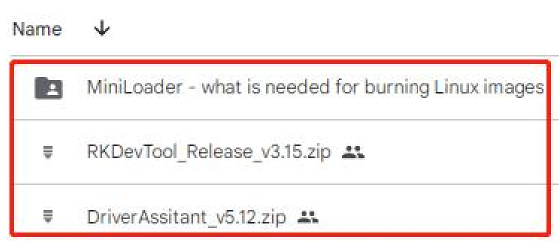
2) You also need to prepare a good quality USB2.0 male-to-male data cable

3) Then download Rockchip microdriver **DriverAssitant_v5.12.zip** and 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



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

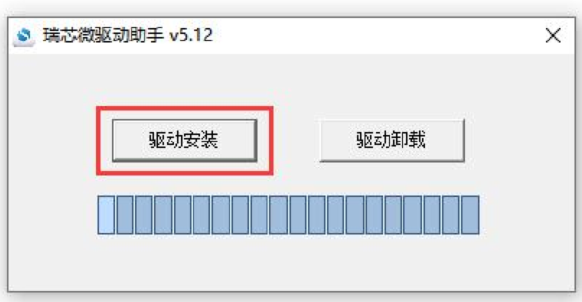
5) Then use decompression software to decompress the compressed package of the

downloaded OPi OS (OH) image

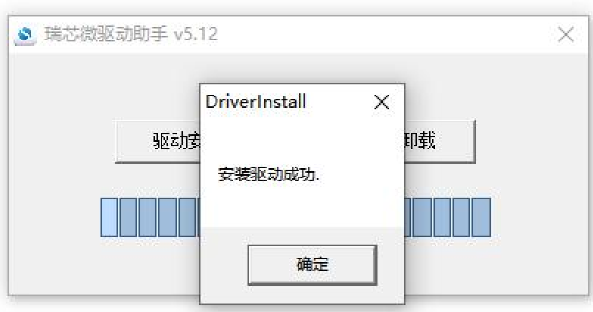
6) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

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

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

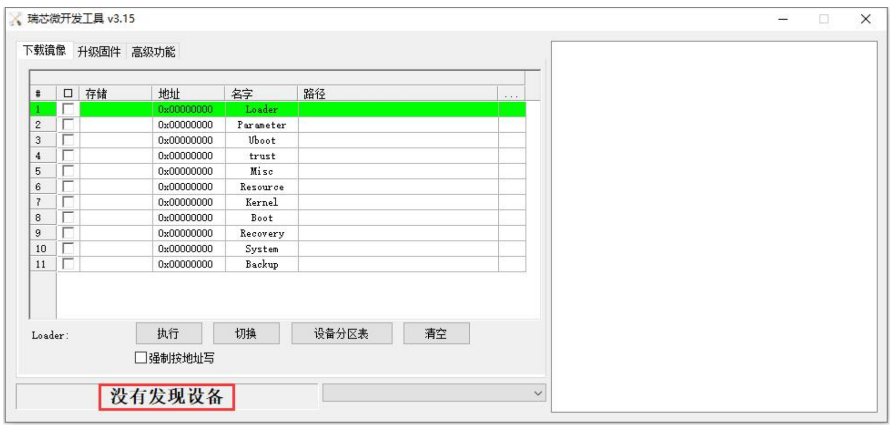


b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



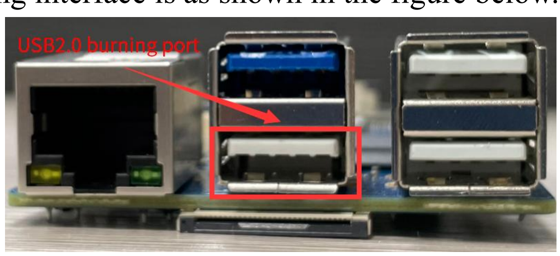
8) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

9) 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.



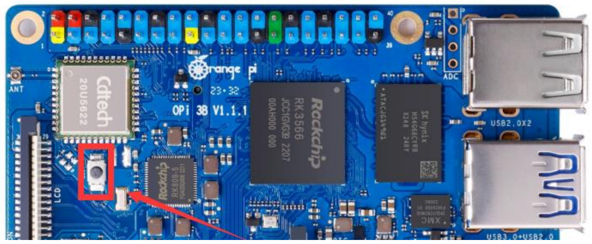
10) Then start burning the OPi OS (OH) image to the TF card

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

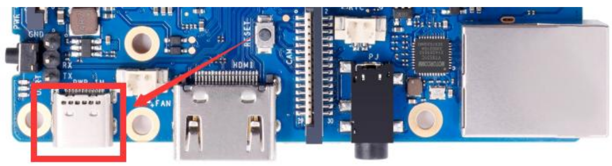


b. **Then make sure that the TF card is not inserted into the development board and the power supply is not connected.**

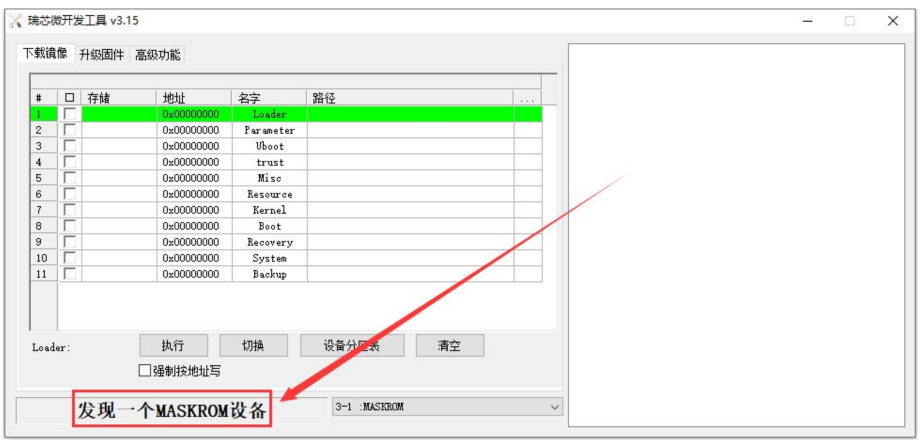
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



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

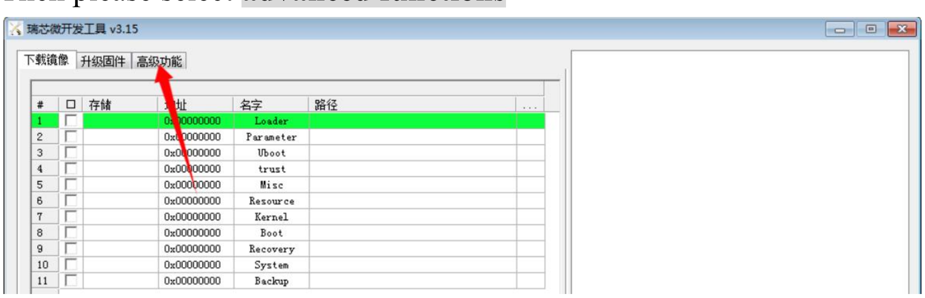


e. If the previous steps go well, the development board will enter **MASKROM** mode at this time, and the interface of the burning tool will prompt "**A MASKROM device was found**"

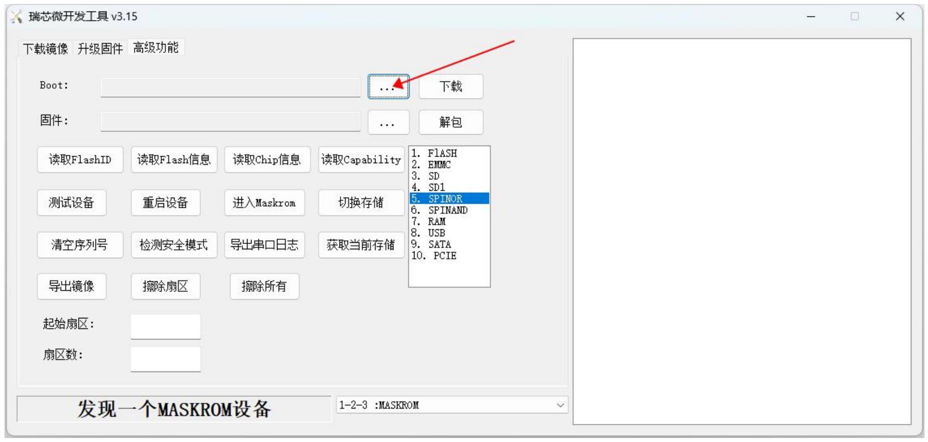


f. Then insert the TF card into the development board

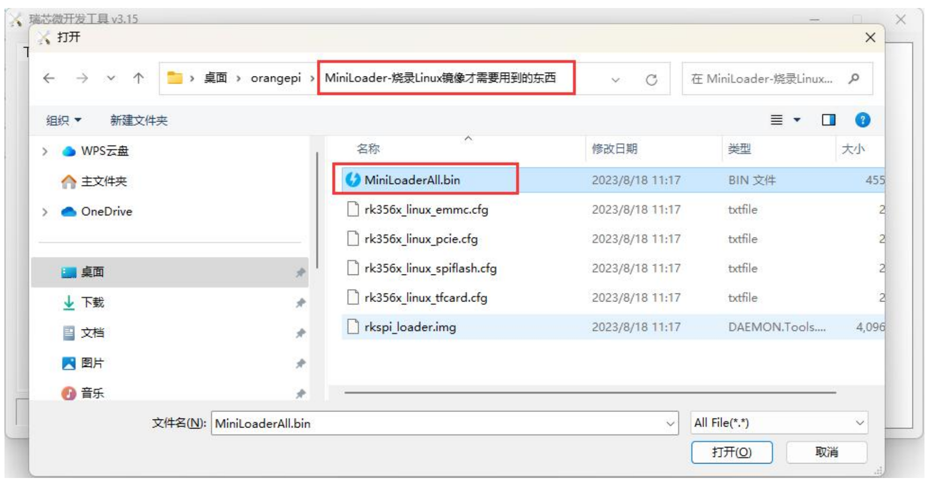
g. Then please select **advanced functions**



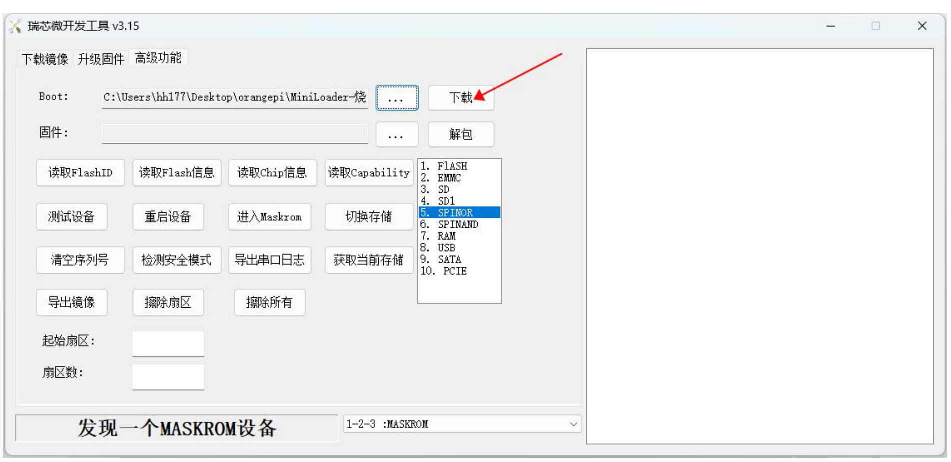
h. Then click the location shown in the picture below



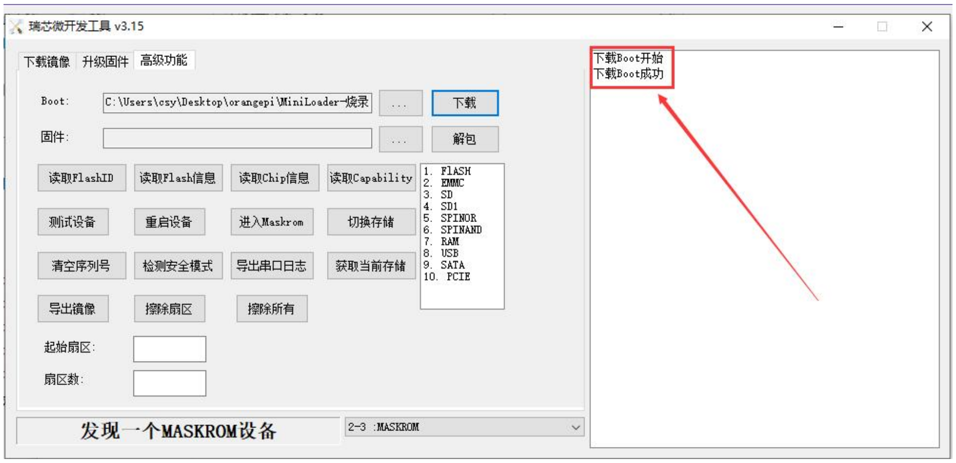
i. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click Open



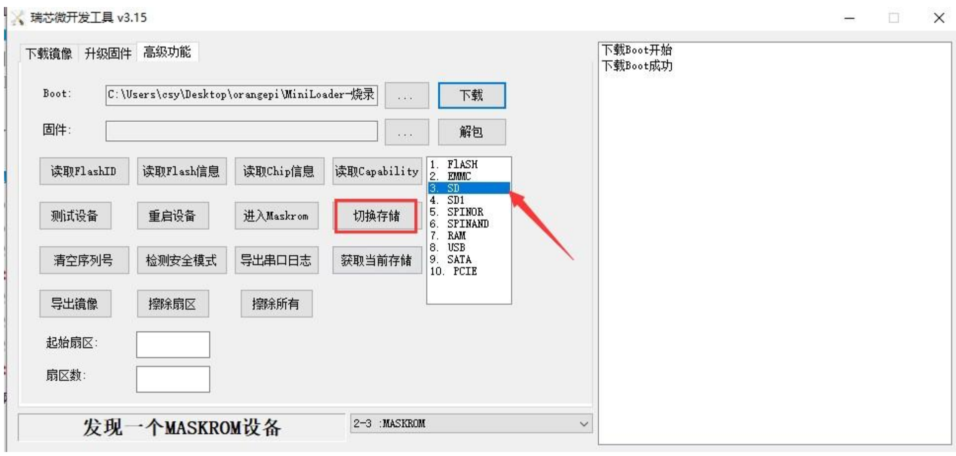
j. Then click **download**



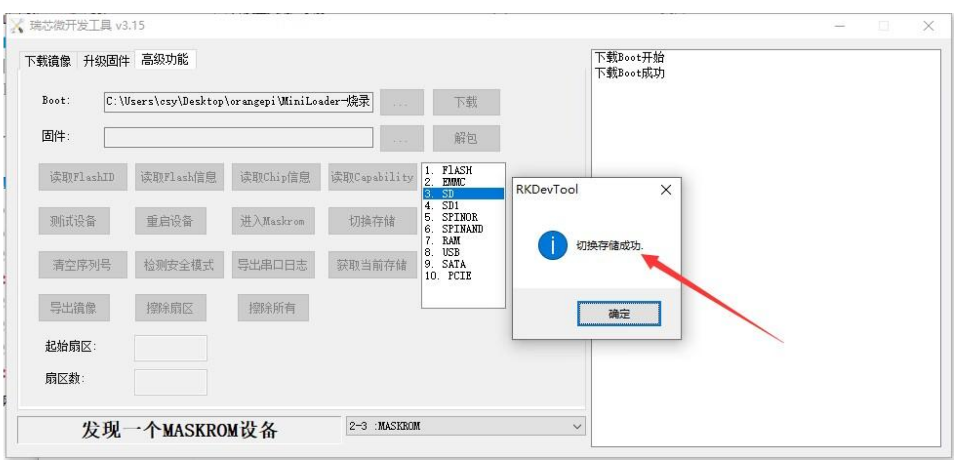
k. After downloading **MiniLoaderAll.bin**, the display is as shown below



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



m. The successful switching is displayed as shown below



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

o. Then click the "**Firmware**" button to select the path of the OPi OS (OH) image that needs to be burned.

p. Finally, click the "**Upgrade**" button to start burning. The log during the burning process is as shown below. After the burning is completed, the OPi OS (OH) system will automatically start.

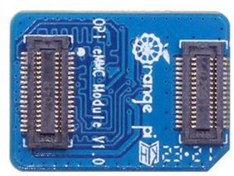
2.11. Method to burn Orange Pi OS (OH) image into eMMC

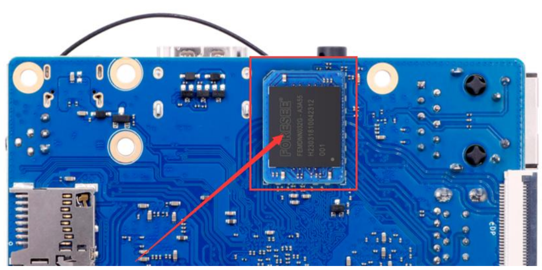
> **Note that after burning the image into eMMC, if you test and find that it cannot be started, please clear SPIFlash and try to start again. For the method of clearing SPIFlash, please refer to the section "How to clear SPIFlash using RKDevTool".**

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

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

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



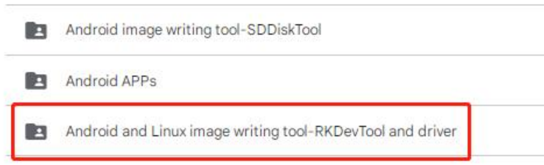


2) You also need to prepare a good quality USB2.0 male-to-male data cable

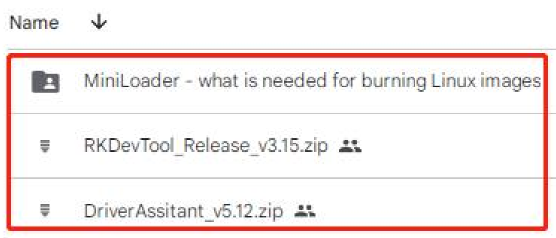
3) Then download Rockchip microdriver **DriverAssitant_v5.12.zip** and burning tool

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



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

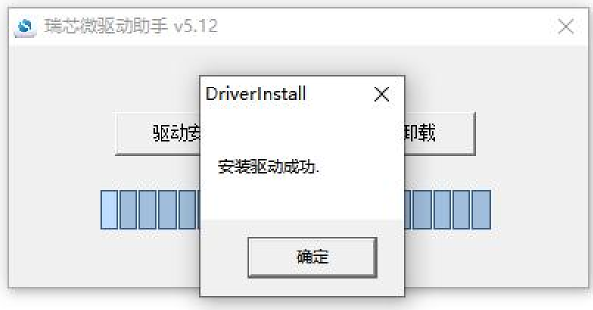
5) Then use decompression software to decompress the compressed package of the downloaded OPi OS (OH) image

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

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

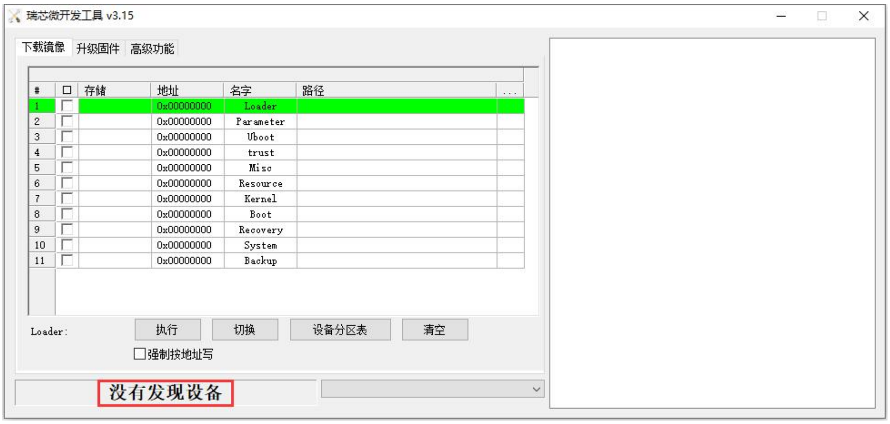
a. Click the "**Driver Installation**" button

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



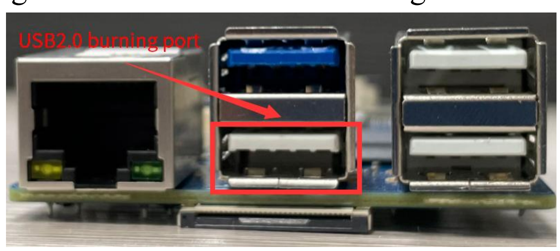
8) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

9) 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.



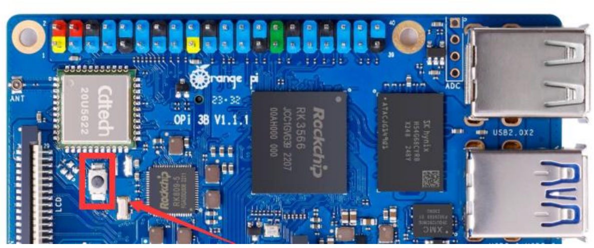
10) Then start burning the OPi OS (OH) image into eMMC

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.

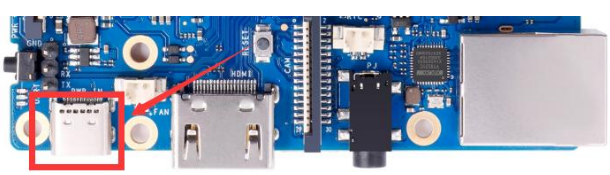


b. Then make sure that the TF card is not inserted into the development board and the power supply is not connected.

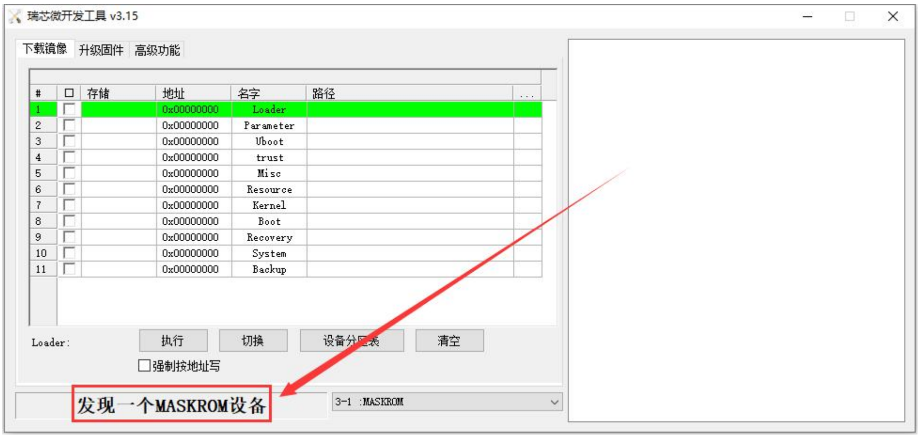
c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



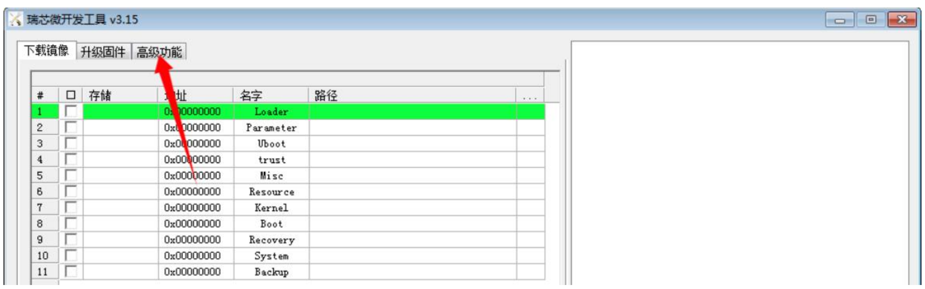
d. Then connect the development board to the power supply of the Type-C interface and power on



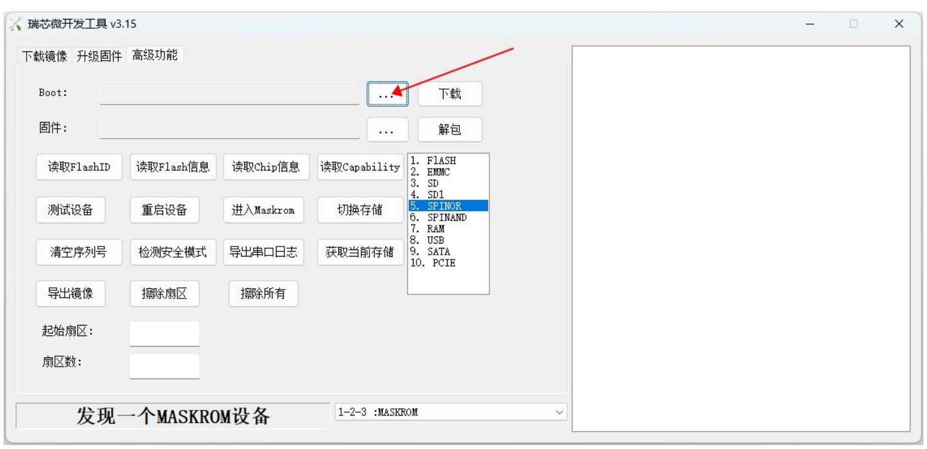
e. If the previous steps go well, the development board will enter **MASKROM** mode at this time, and the interface of the burning tool will prompt "**A MASKROM device was found**"



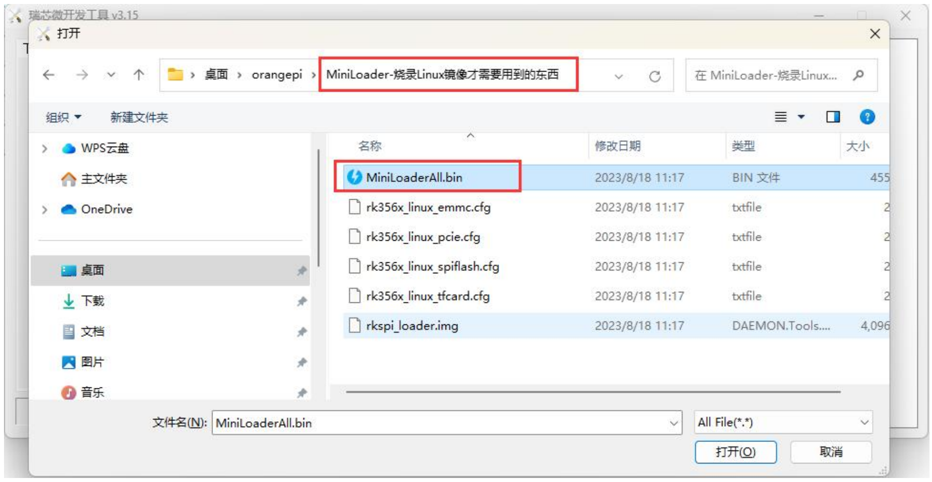
f. Then please select **advanced functions**



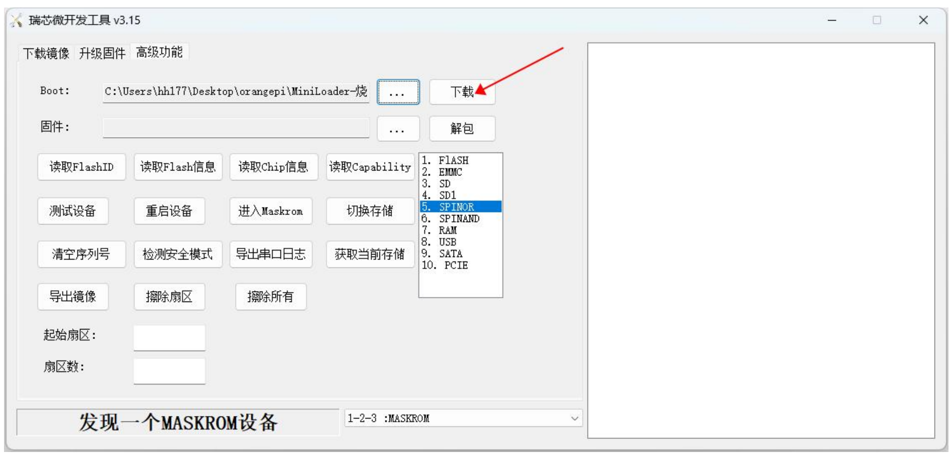
g. Then click the location shown in the picture below



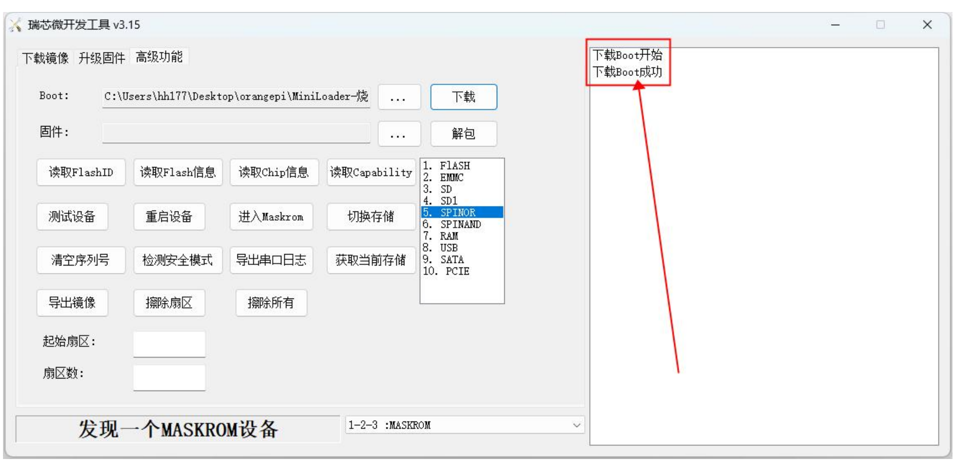
h. Then select **MiniLoaderAll.bin** in the **MiniLoader** folder downloaded earlier, and then click to open



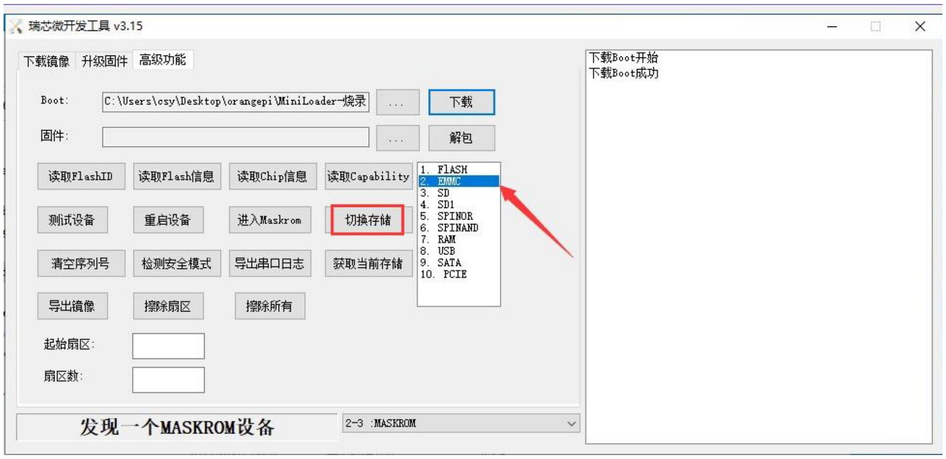
i. Then click **Download**



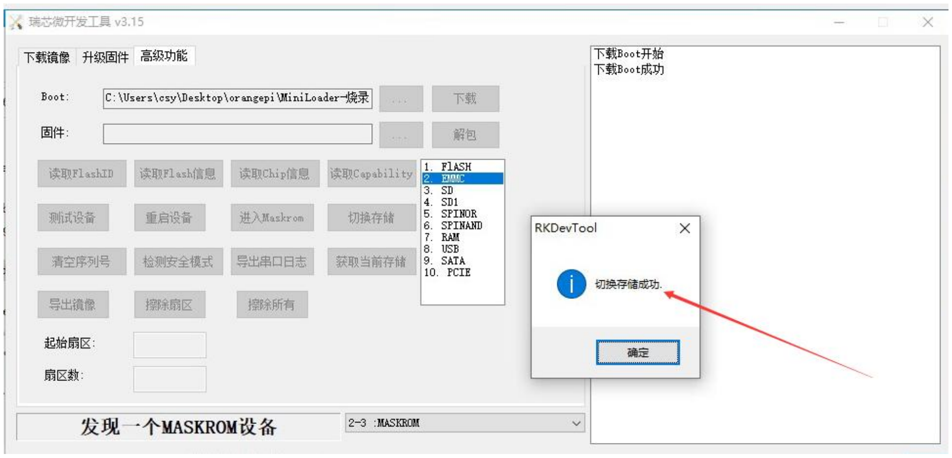
j. After downloading **MiniLoaderAll.bin**, the display is as shown below



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



l. The successful switching is displayed as shown below.



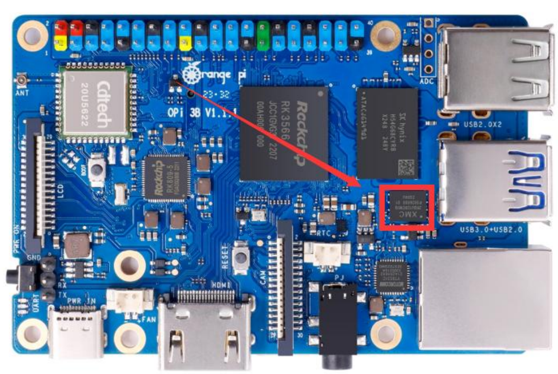
m. Then click the "**Upgrade Firmware**" column of the burning tool

n. Then click the "**Firmware**" button to select the path of the OPi OS (OH) image that needs to be burned.

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

2. 12. Using RKDevTool to clear SPIFlash

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

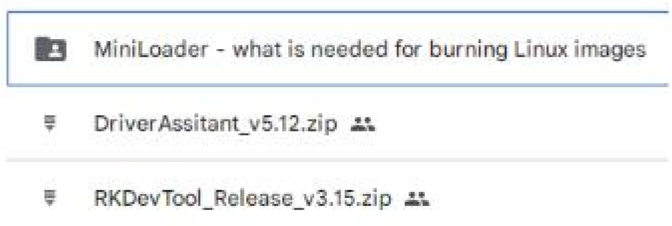


2) First, you need to prepare a good quality USB2.0 male-to-male data cable

3) Then download the Rockchip driver **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 download page of Orange Pi, first select the official tool, and then enter the following folder

b. Then download all the files below

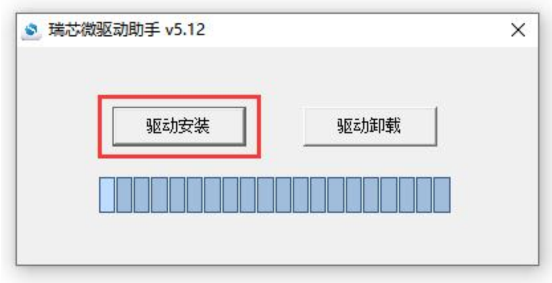


Note that the "MiniLoader-things needed to burn the Linux image" folder is hereinafter referred to as the MiniLoader folder.

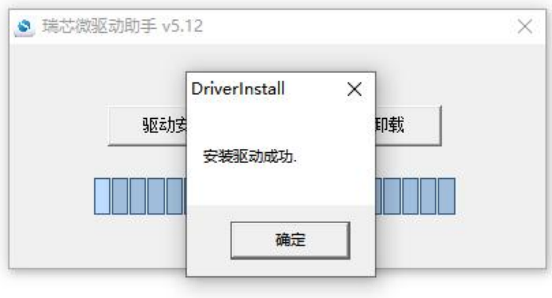
4) Then use decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

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

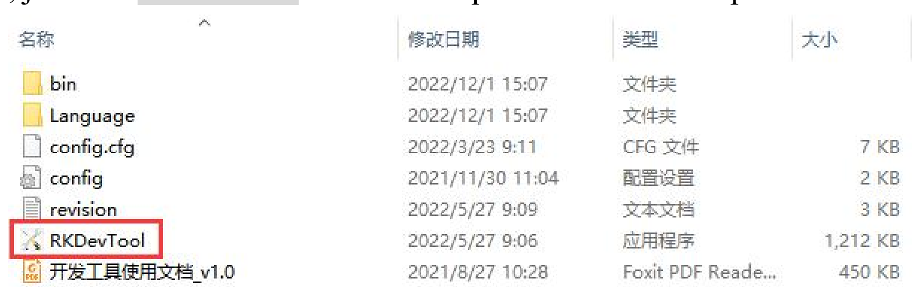
a. Click the "**Driver Installation**" button



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



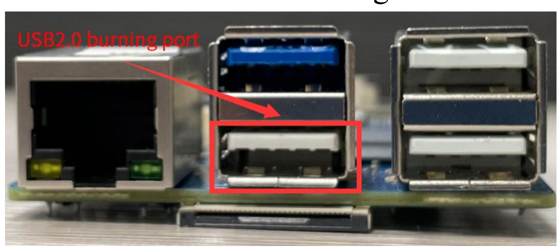
6) Then decompress **RKDevTool_Release_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it



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

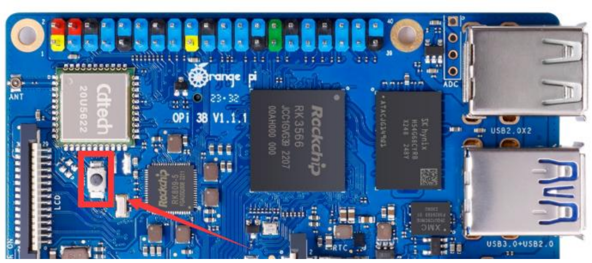
8) Then you can start to clear the content in SPI FLASH

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

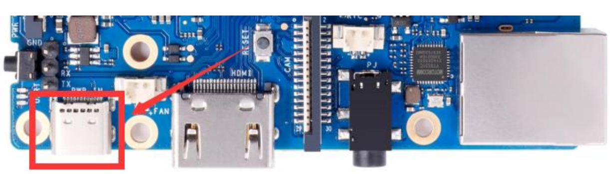


b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

c. 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:

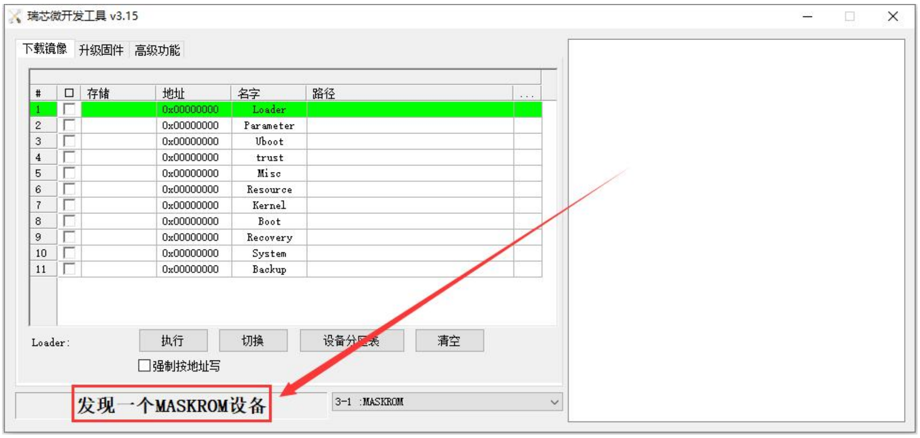


d. Then connect the power supply of the Type-C interface to the development board, and power on, and then release the MaskROM button

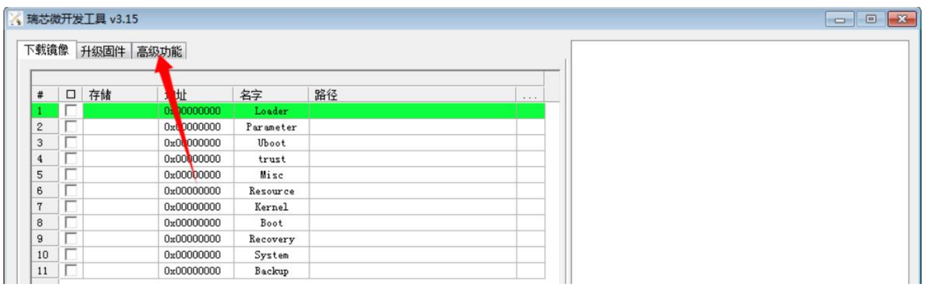


e. If the previous steps are successful, the development board will enter the

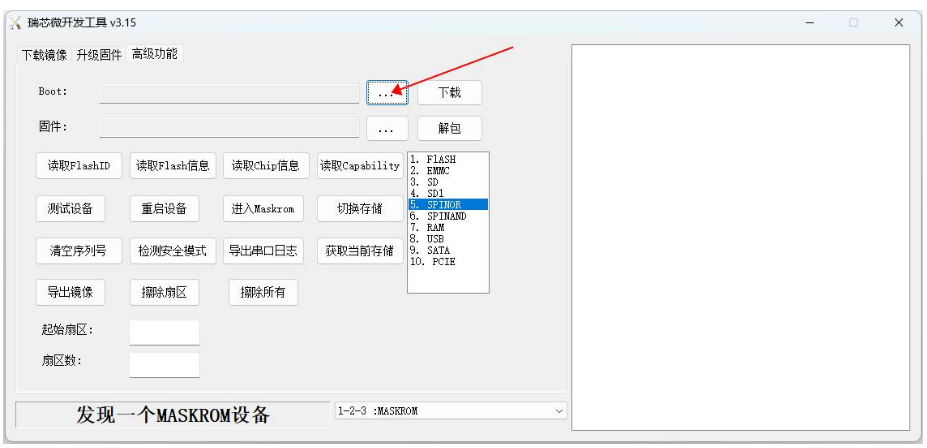
MASKROM mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"



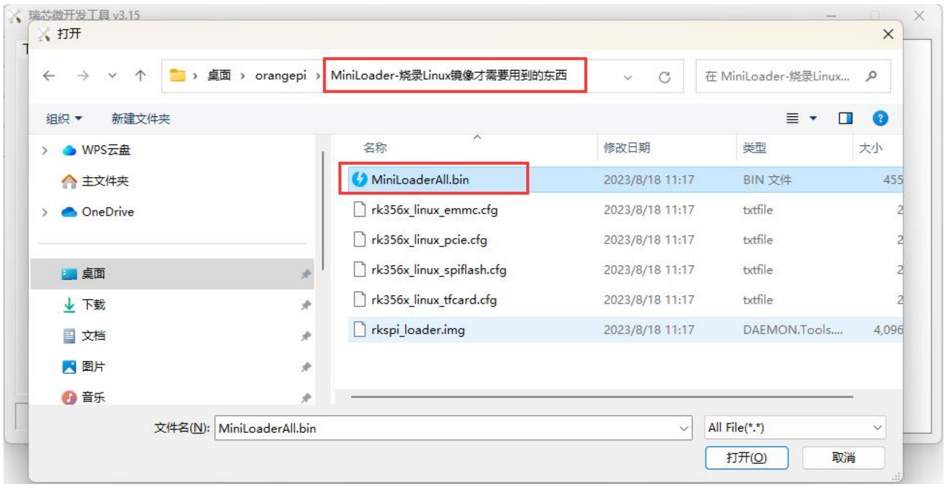
f. Then please select **Advanced Features**



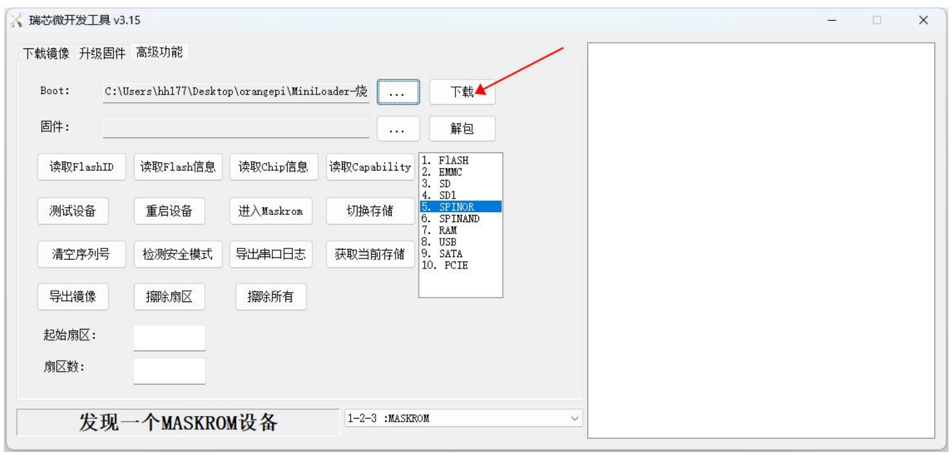
g. Then click the position shown in the figure below



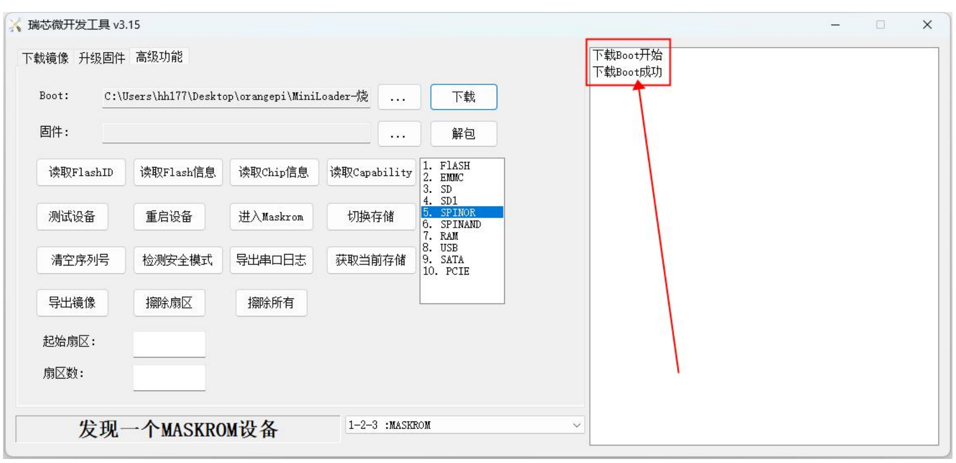
h. Select **MiniLoaderAll.bin** in the **MiniLoader** folder you downloaded earlier, and click Open



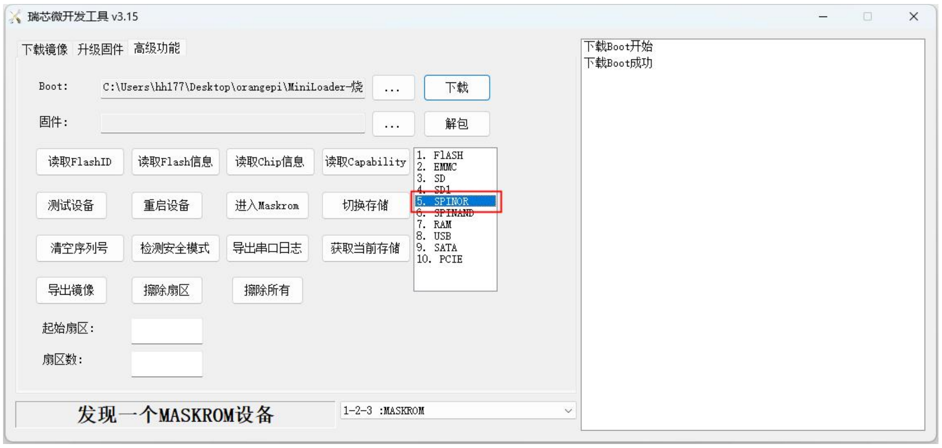
i. Then click **Download**



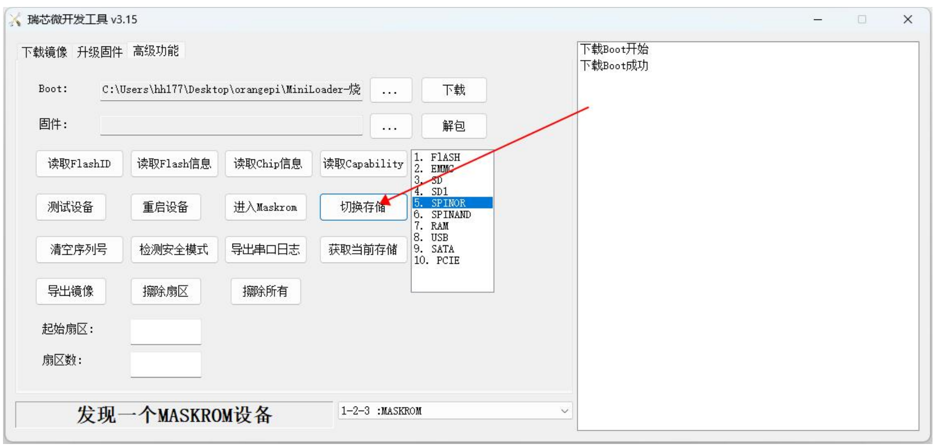
j. The display after downloading **MiniLoaderAll.bin** is shown in the figure below



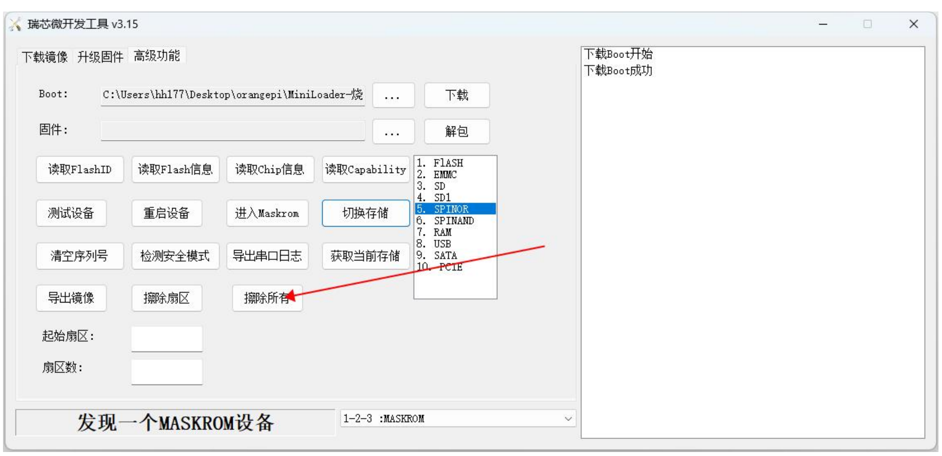
k. Then select the storage device as **SPINOR**



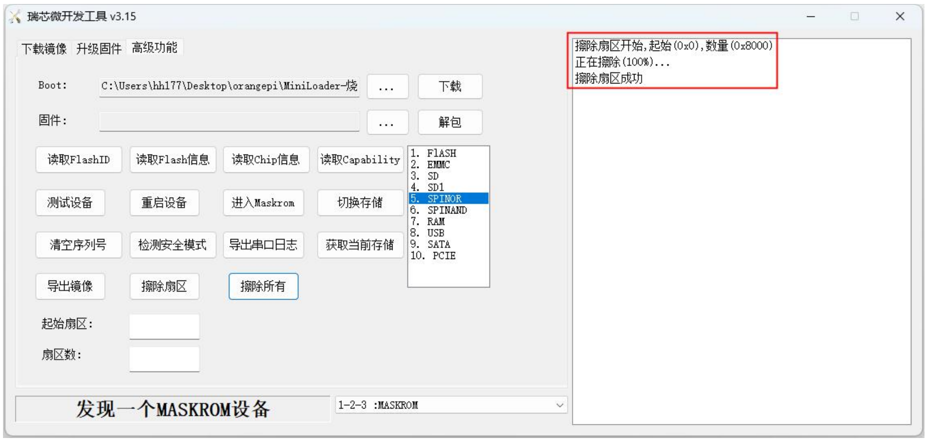
l. Then click Switch **Storage**



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



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



2. 13. Start the Orange Pi development board

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

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

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

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

5) 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, as this will burn out the development board.

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

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

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

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

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

2. 14. How to use the debugging serial port

2. 14. 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 USB to TTL module.

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

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

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

a. 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 TX** of the development board

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

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

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

2. 14. 2. How to use the debugging serial port on the Ubuntu platform

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

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

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

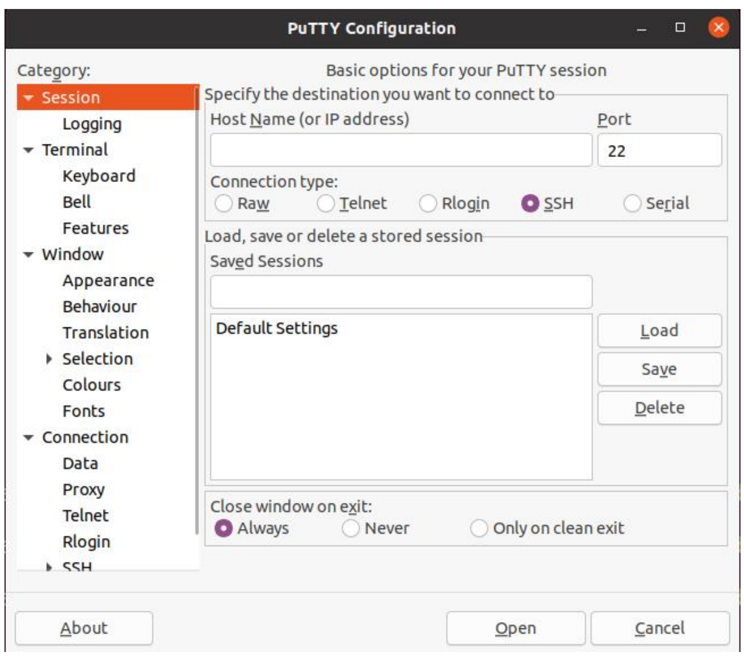
2) Then use the following command to install putty on Ubuntu PC

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

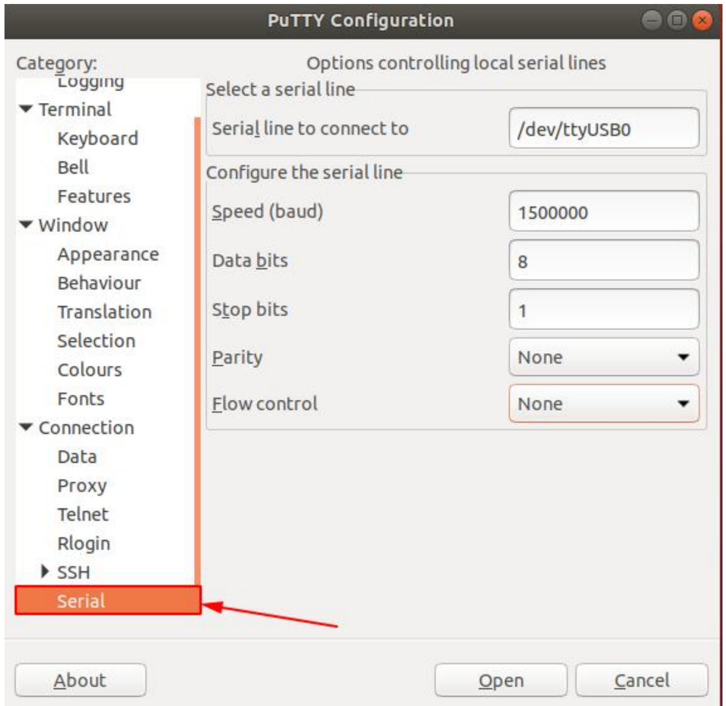
3) Then run putty, **remember to add sudo permission**

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

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



5) First select the setting interface of the serial port

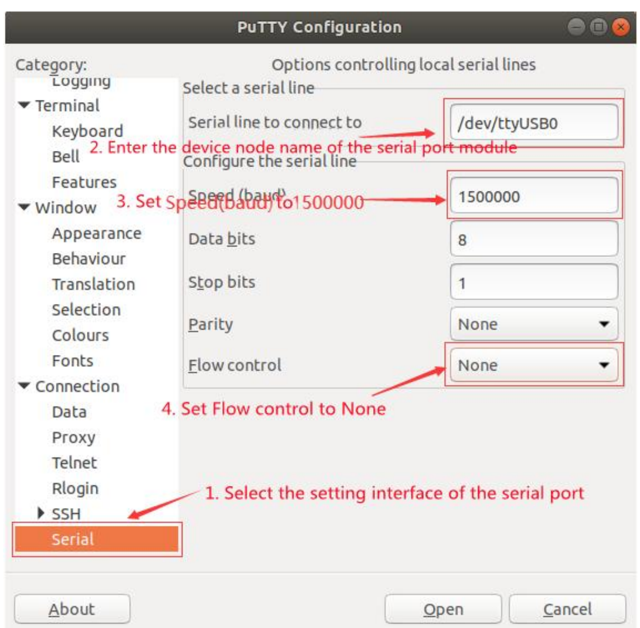


6) Then set the parameters of the serial port

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

b. Set Speed(baud) as **1500000** （Serial port baud rate）

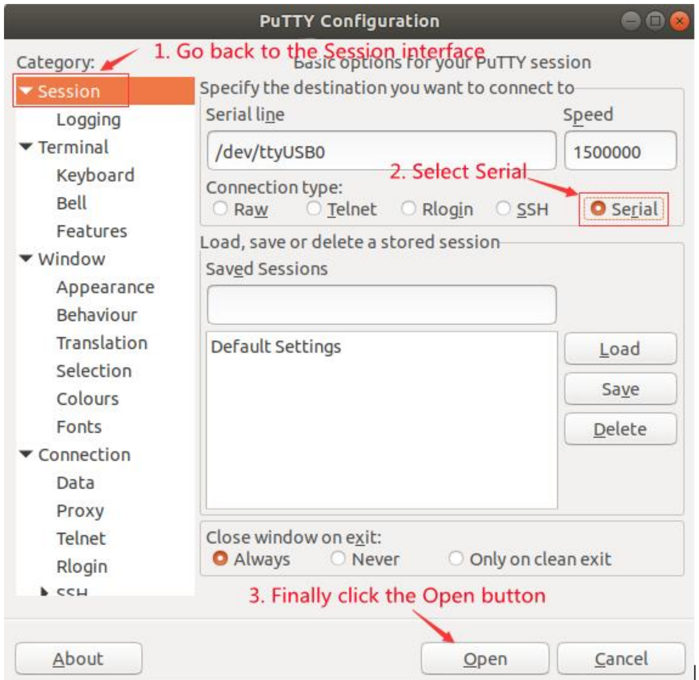
c. Set Flow control as None



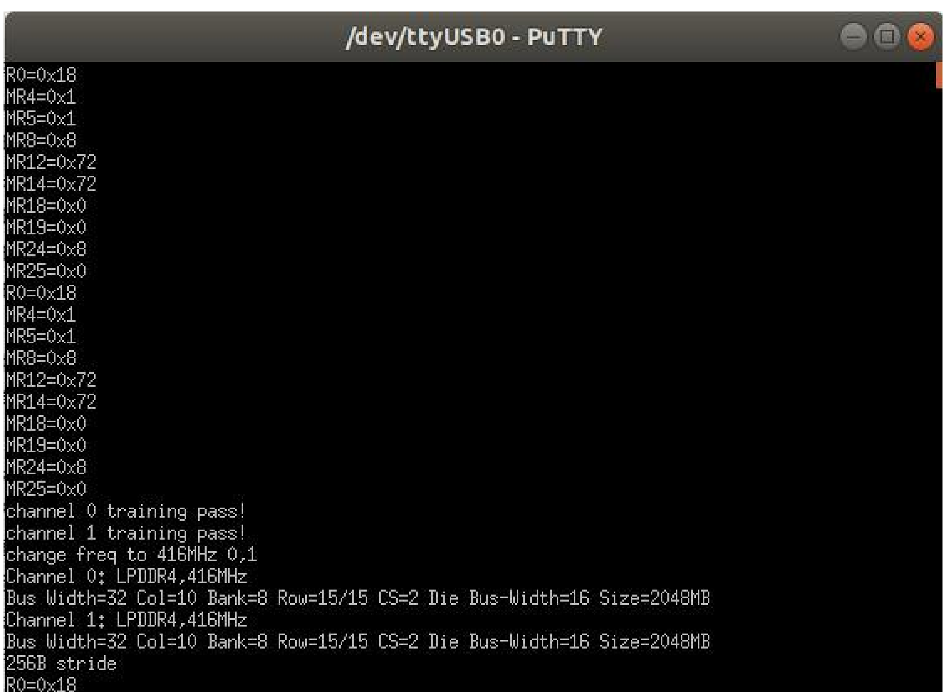
7) After setting the setting interface of the serial port, return to the Session interface

a. First select the Connection type as Serial

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



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



2. 14. 3. How to use the debugging serial port on Windows platform

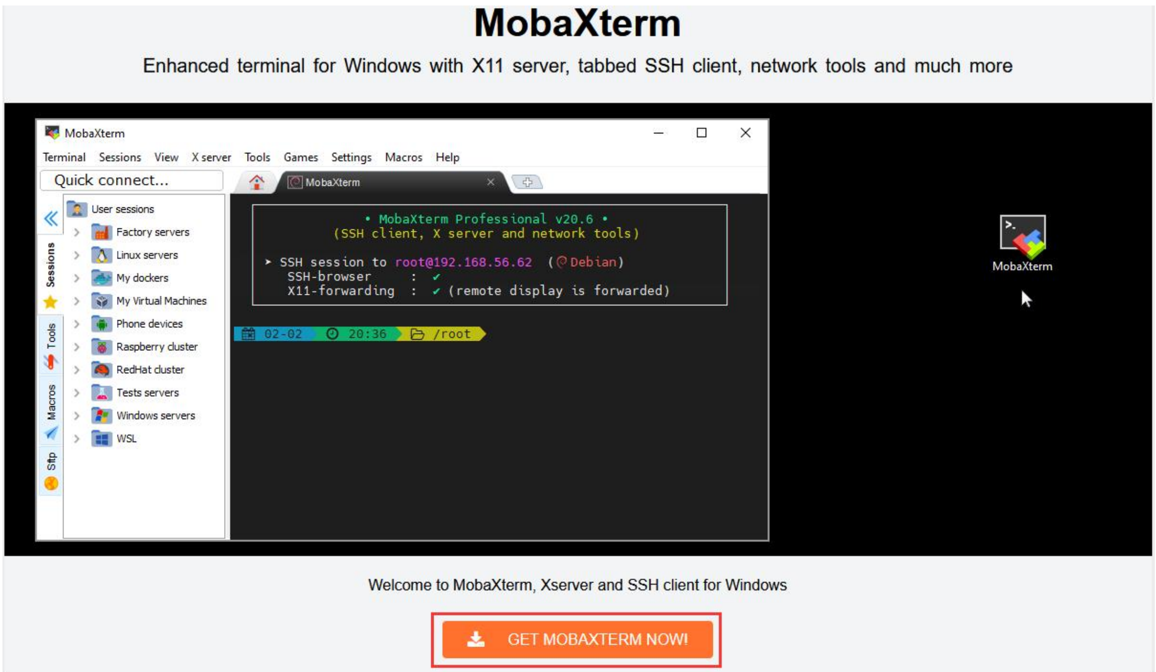
There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without buying a serial number.

1) Download MobaXterm

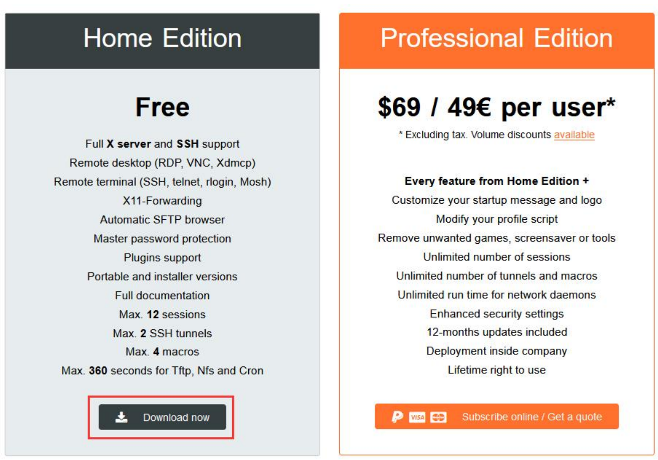
a. Download MobaXterm website as follows

```
https://mobaxterm.mobatek.net
```

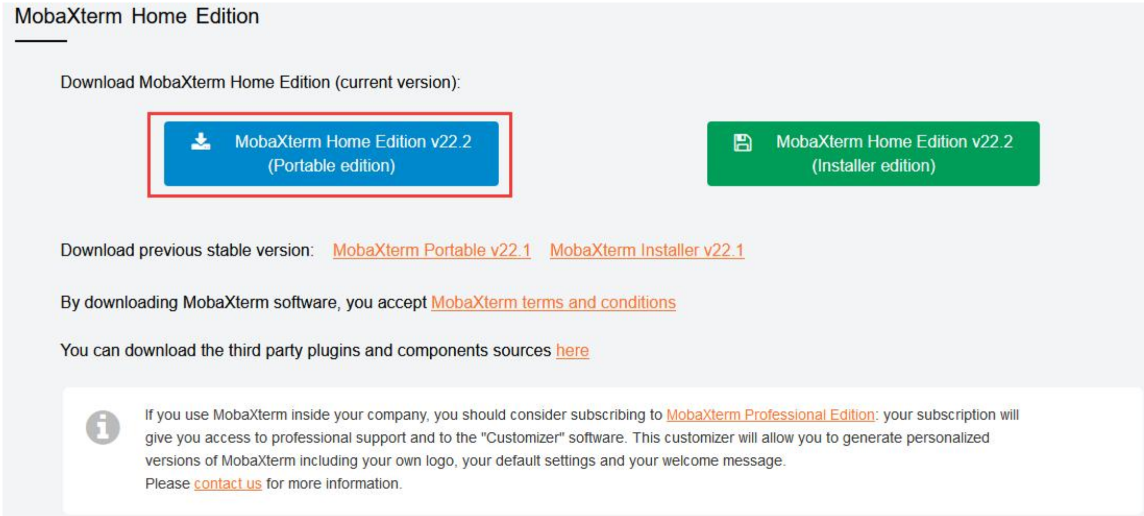
b. After entering the MobaXterm download page, click **GET XOBATERM NOW!**



c. Then choose to download the Home version



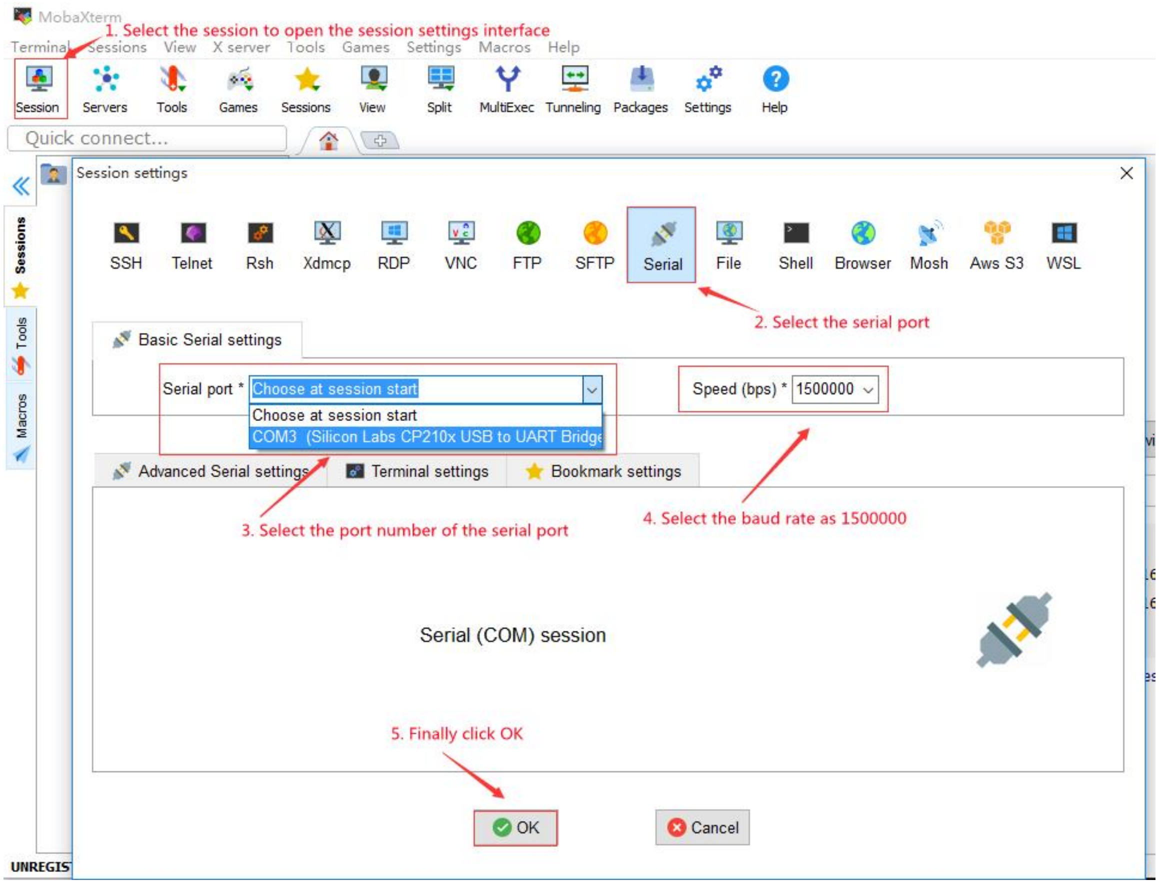
d. Then select the Portable version. After downloading, you don’t need to install it, just open it and use it



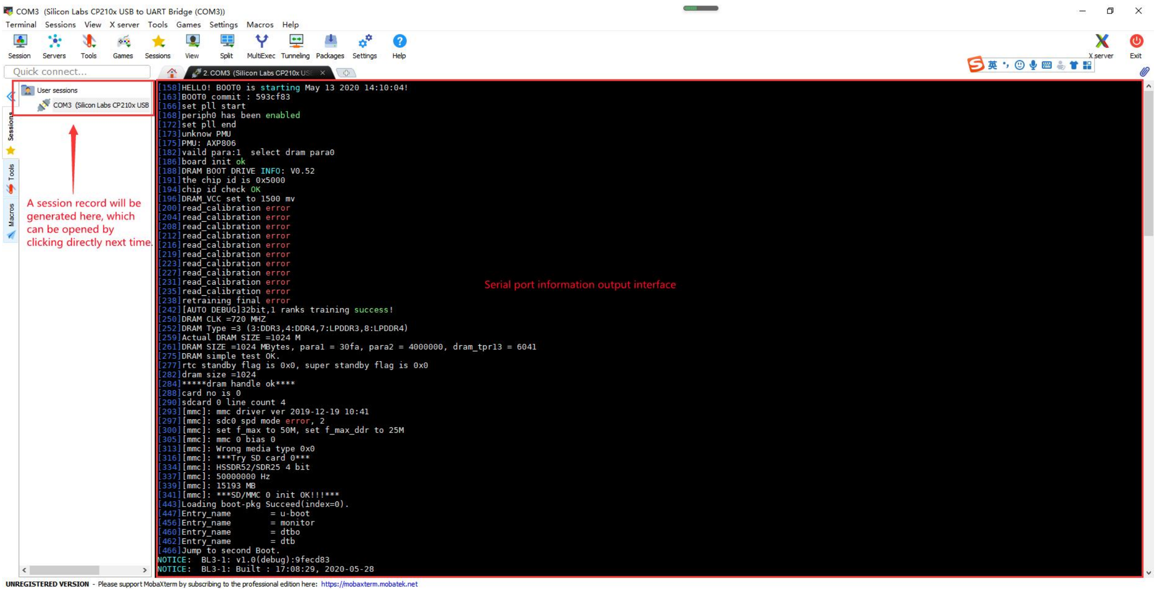
2) After downloading, use decompression software to decompress the downloaded compressed package, you can get the executable software of MobaXterm, and then double-click to open

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

a. Open the session settings interface
b. Select the serial port type
c. Select the port number of the serial port (select the corresponding port number according to the actual situation)
d. Select the baud rate of the serial port as **1500000**
e. Finally click the "**OK**" button to complete the setting



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



2.15. 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 for power supply. If you need to use the 5V pin in the 40pin interface to power the development board, please make sure 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 back to the Type-C power supply.

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

Please purchase the power cord shown in the picture above by yourself

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

a. The USB A port of the power cord shown in the above picture needs to be plugged into the 5V/3A power adapter connector (**please do not plug into the computer's USB port 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 inserted into the GND pin of the 40pin interface

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

3. Instructions for use of Ubuntu/Debian Server and Xfce desktop system

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

3. 1. Supported Linux image types and kernel versions

Linux image type	kernel version	server version	desktop version
Debian 11 - Bullseye	Linux5.10	support	support
Ubuntu 20.04 - Focal	Linux5.10	support	support
Ubuntu 22.04 - Jammy	Linux5.10	support	support
Debian 11 - Bullseye	Linux6.6	support	support
Debian12 - Bookworm	Linux6.6	support	support
Ubuntu 22.04 - Jammy	Linux6.6	support	support

3. 2. Linux System adaptation

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

Function	Debian11	Ubuntu20.04	Ubuntu22.04
USB2.0x3	OK	OK	OK
USB3.0x1	OK	OK	OK
M.2 NVMe SSD Start	OK	OK	OK
WIFI	OK	OK	OK
Bluetooth	OK	OK	OK
GPIO（40pin）	OK	OK	OK
UART（40pin）	OK	OK	OK
SPI（40pin）	OK	OK	OK
I2C（40pin）	OK	OK	OK
PWM（40pin）	OK	OK	OK
PWM fan interface	OK	OK	OK

3pin Debug serial port	OK	OK	OK
EMMC	OK	OK	OK
TF card start	OK	OK	OK
HDMI Video	OK	OK	OK
HDMI Audio	OK	OK	OK
OV5647 Camera	The kernel driver is OK, 3A is not adjusted		
LCD	OK	OK	OK
Edp Display	OK	OK	OK
Gigabit Ethernet port	OK	OK	OK
Network port status light	OK	OK	OK
headphone playback	OK	OK	OK
headphone recording	OK	OK	OK
LED Light	OK	OK	OK
RTC	OK	OK	OK
GPU	OK	OK	OK
NPU	OK	OK	OK
VPU	OK	OK	OK
watchdog test	OK	OK	OK
Chromium Hard solution video	OK	OK	OK

3. 2. 2. Linux6.6 system adaptation situation

Function	Debian11	Debian12	Ubuntu22.04
USB2.0x3	OK	OK	OK
USB3.0x1	OK	OK	OK
M.2 NVMe SSD Start	OK	OK	OK
WIFI	OK	OK	OK
Bluetooth	OK	OK	OK
GPIO（40pin）	OK	OK	OK
UART（40pin）	OK	OK	OK
SPI（40pin）	OK	OK	OK
I2C（40pin）	OK	OK	OK
PWM（40pin）	OK	OK	OK
PWM fan interface	OK	OK	OK
3pin Debug serial port	OK	OK	OK

EMMC	**OK**	**OK**	**OK**
TF card start	**OK**	**OK**	**OK**
HDMI Video	**OK**	**OK**	**OK**
HDMI Audio	**OK**	**OK**	**OK**
OV5647 Camera	**NO**	**NO**	**NO**
LCD	**NO**	**NO**	**NO**
eDP Display	**NO**	**NO**	**NO**
Gigabit Ethernet port	**OK**	**OK**	**OK**
Network port status light	**OK**	**OK**	**OK**
headphone playback	**NO**	**NO**	**NO**
headphone recording	**NO**	**NO**	**NO**
LED Light	**OK**	**OK**	**OK**
RTC	**OK**	**OK**	**OK**
GPU	**OK**	**OK**	**OK**
NPU	**NO**	**NO**	**NO**
VPU	**NO**	**NO**	**NO**
watchdog test	**NO**	**NO**	**NO**

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

1) In this manual, all commands that need to be entered in the Linux system will be marked with the following box

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

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

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

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

b. **root@orangepi:~$** The prompt indicates that this command is entered in **the Linux system of the development board**. The $ at the end of the prompt indicates that the current user of the system is a normal user. When executing a privileged command, you need to add **sudo**

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

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

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

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

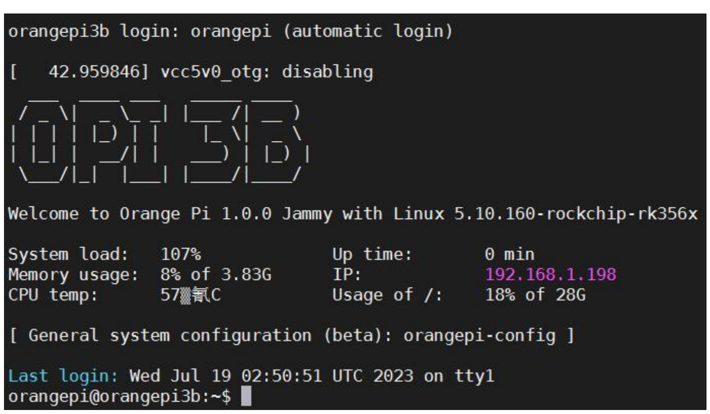
Account	Passport
root	orangepi
orangepi	orangepi

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

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

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

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



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

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

3) Use the following command to disable automatic login terminal

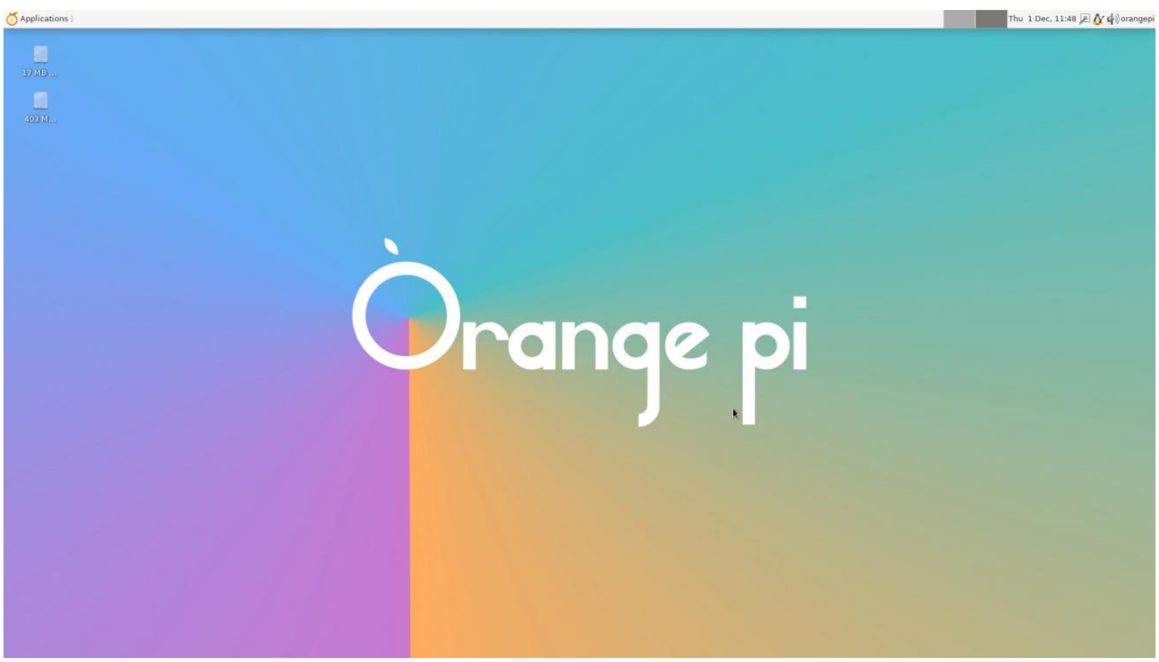
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

3.4.5. The method of disabling the desktop in the Linux desktop version system

1) First enter the following command on 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:

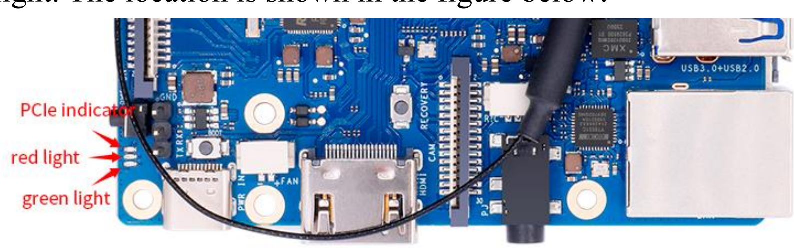
a. First enter the following command on the command line，**Please remember to add sudo permission**

```
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 blinking after the kernel is started, which is controlled by software.

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

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

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

a. First enter the setting directory of the green light

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

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

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

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

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

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

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

3. 6. Network connection test

3. 6. 1. Ethernet port test

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

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

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

Note that in the following command, Debian12 of linux5.10 needs to modify eth0 to end1, and Debian12 of linux6.6 needs to modify eth0 to end0.

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

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

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

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

1. Connect the HDMI 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's network port through the router's management interface. However, in this method, some people often cannot see the IP address of the development board normally. If you can't see it, the debug method looks like this:

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

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

C) Try another router (I have encountered many problems with the router, such as the router cannot assign the IP address normally, or the IP address has been assigned normally but cannot be seen in the router);

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

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

4) The command to test the network connectivity is as follows, the **ping** command can be interrupted through the shortcut key of **Ctrl+C** (Here is an uppercase I, not a lowercase L)

```
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 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 the following three ways

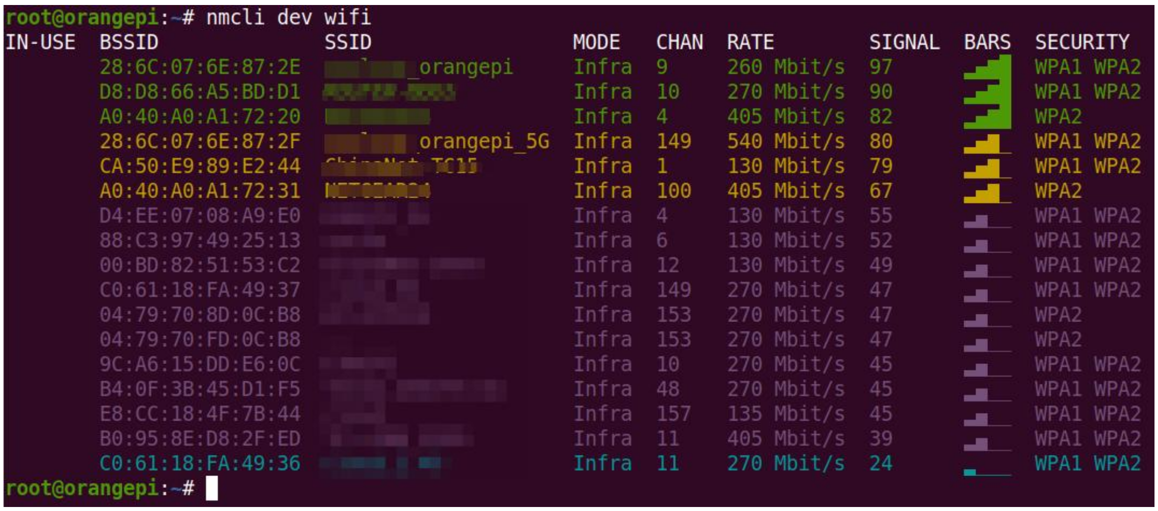
a. If the development board is connected with a network cable, you can **remotely log in to the Linux system through ssh**

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

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

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



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

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

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

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

4) You can view the IP address of 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. The server image connects to WIFI in a graphical way

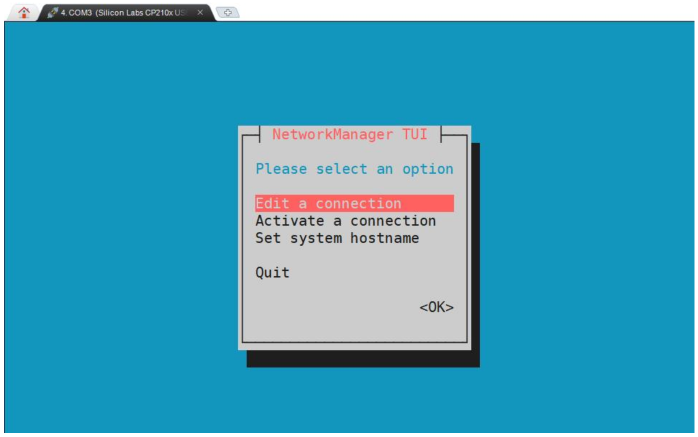
1) First log in to the Linux system, there are the following three ways

a. If the development board is connected with a network cable, you can remotely log in to **the Linux system through ssh**
b. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the Linux system (please use MobaXterm for the serial port software, and the minicom cannot display the graphical interface)
c. If the development board is connected to the HDMI display, you can log in to the Linux system through the HDMI display terminal

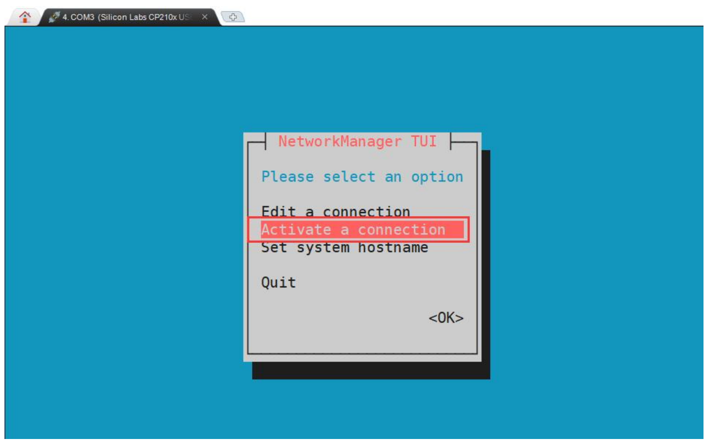
2) Then enter the nmtui command in the command line to open the wifi connection interface

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

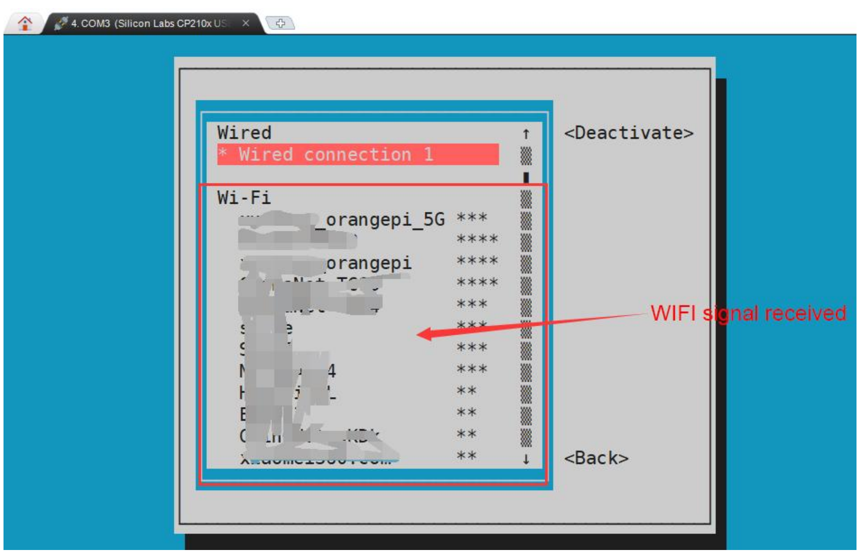
3) Enter the nmtui command to open the interface as shown below



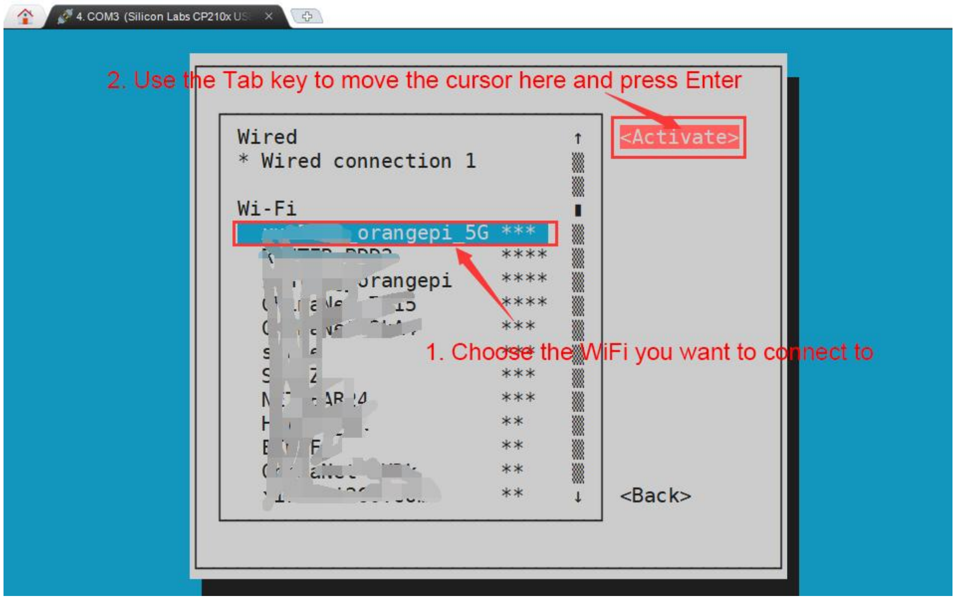
4) Select **Activate a connect** and press Enter



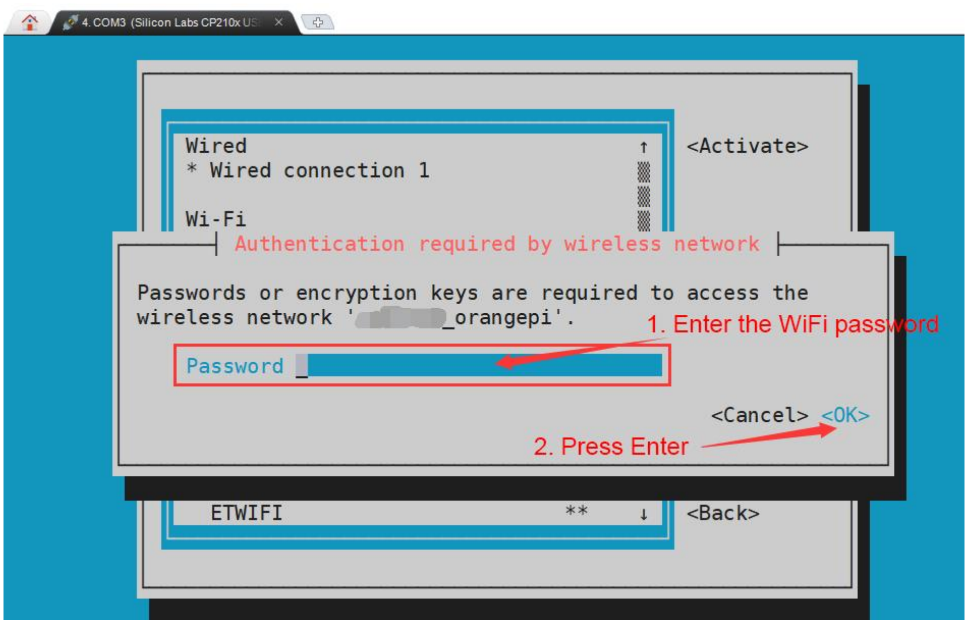
5) Then you can see all the searched WIFI hotspots



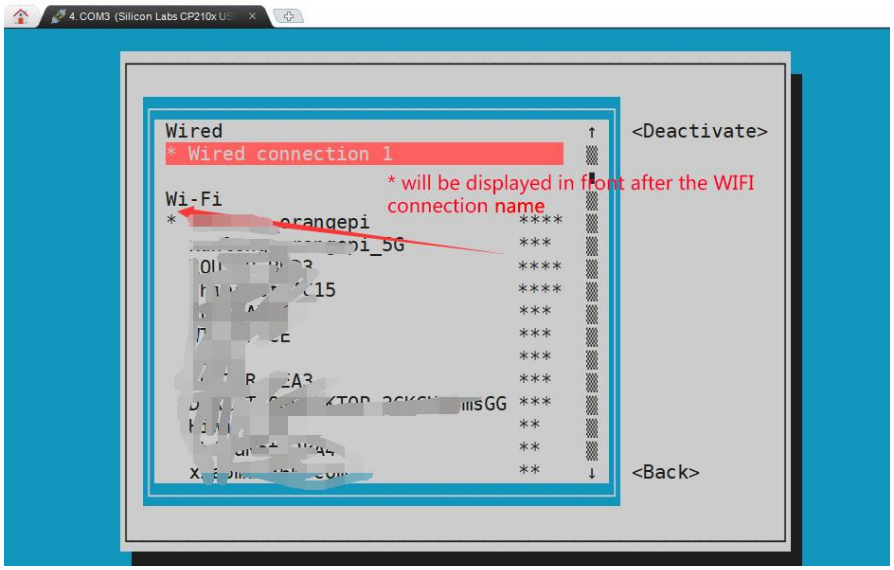
6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the cursor on **Activate** and press Enter



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



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



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

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

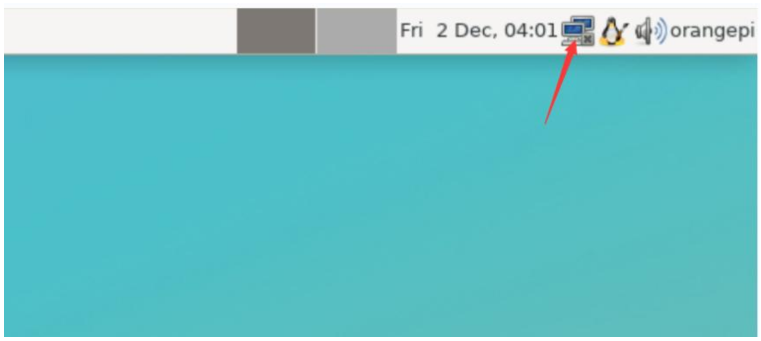
10) Use the **ping** command to test the connectivity of the wifi network, and the **ping** command can be interrupted through the shortcut key **Ctrl+C**

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

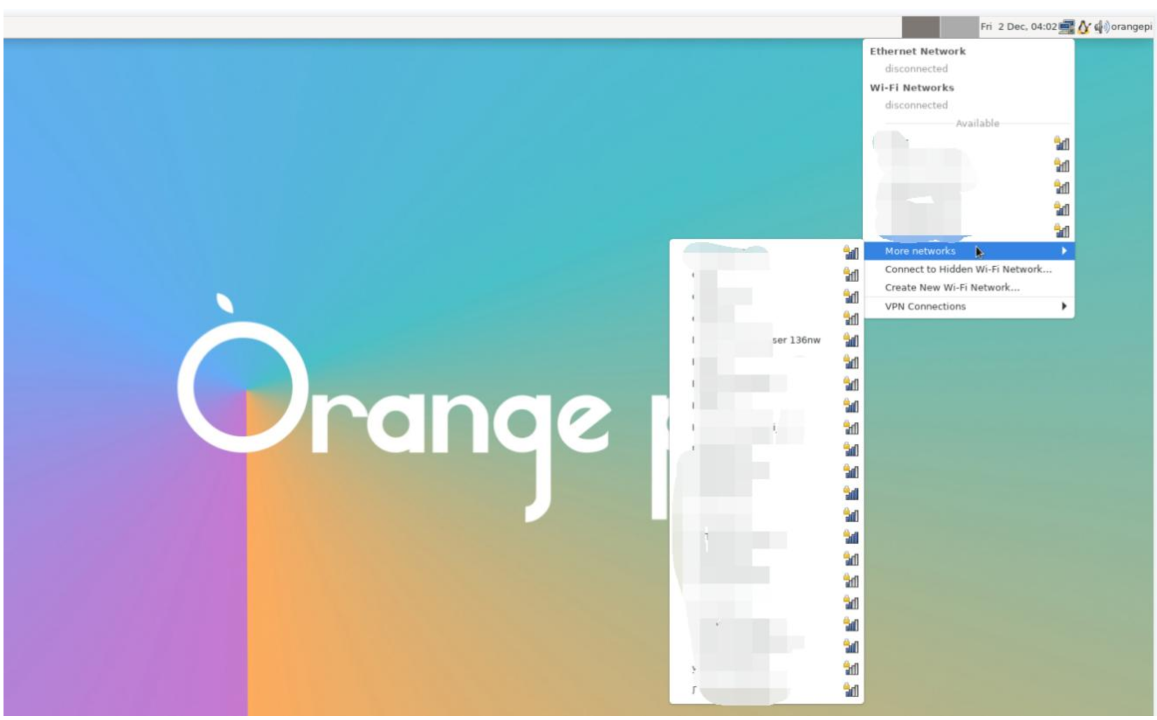
```
^C
--- www.orangepi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

3.6.2.3. Test method of desktop image

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

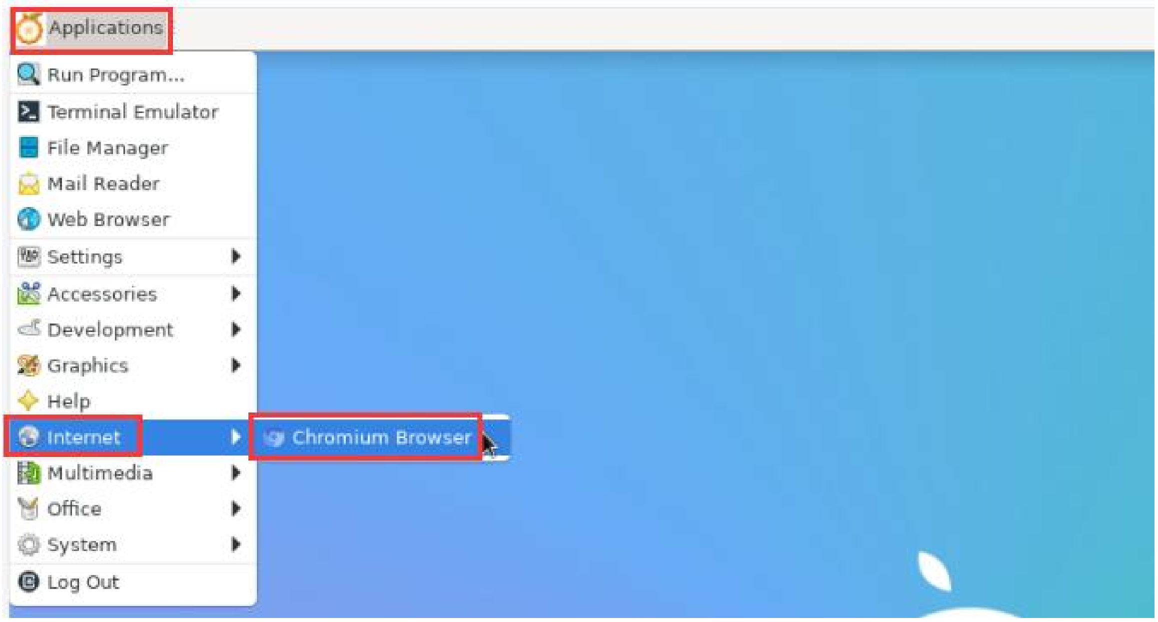


2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to.

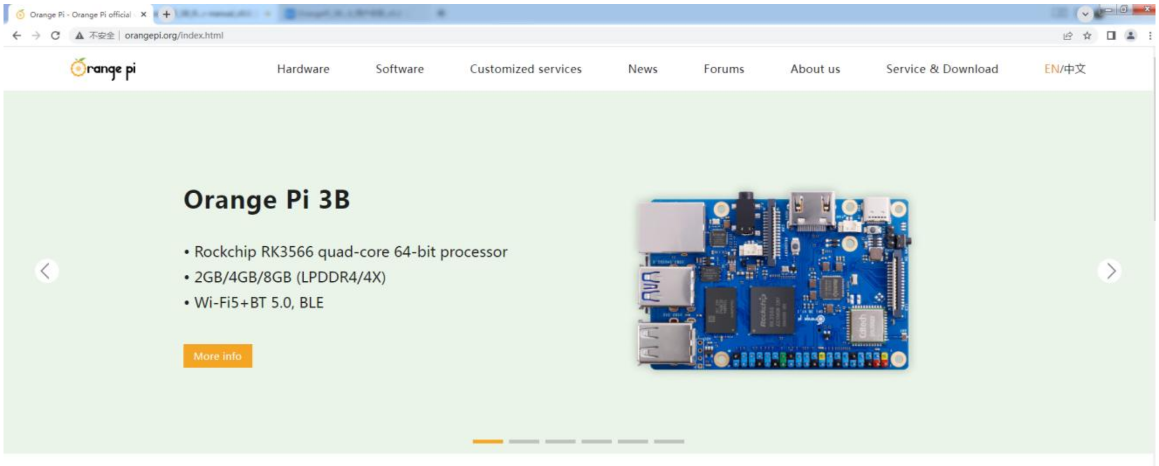


3) Then enter the password of the WIFI hotspot, and then click **Connect** to start connecting to WIFI

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



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



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

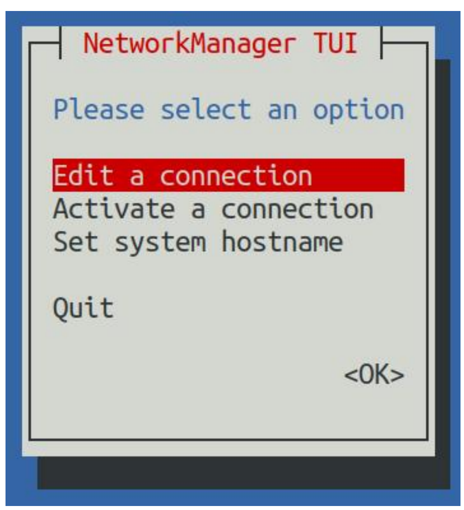
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file.

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

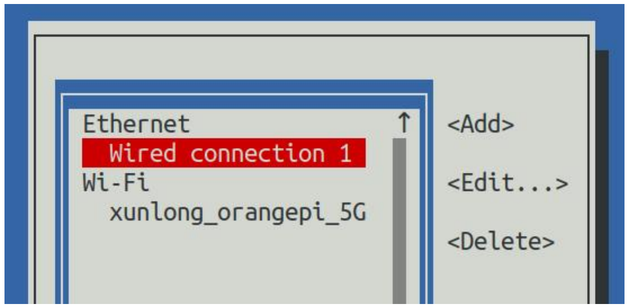
1) First run the **nmtui** command

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

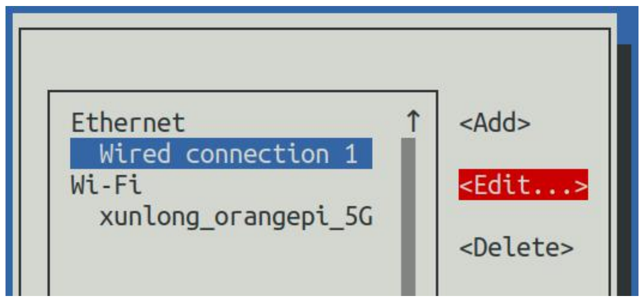
2) Then select **Edit a connection** and press Enter



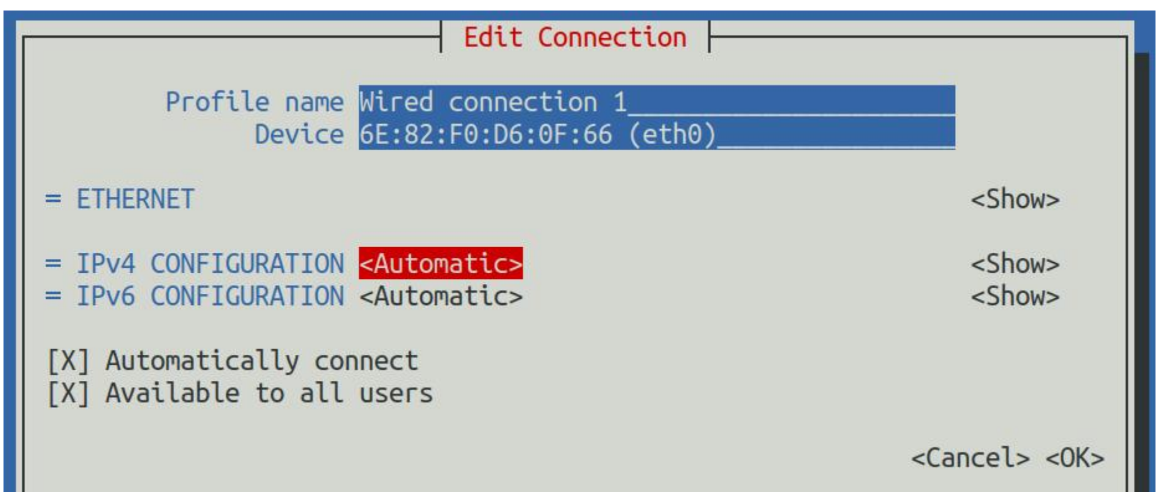
3) Then select the network interface that needs to set a static IP address, for example, to set the static IP address of the **Ethernet** interface, select **Wired connection 1**.



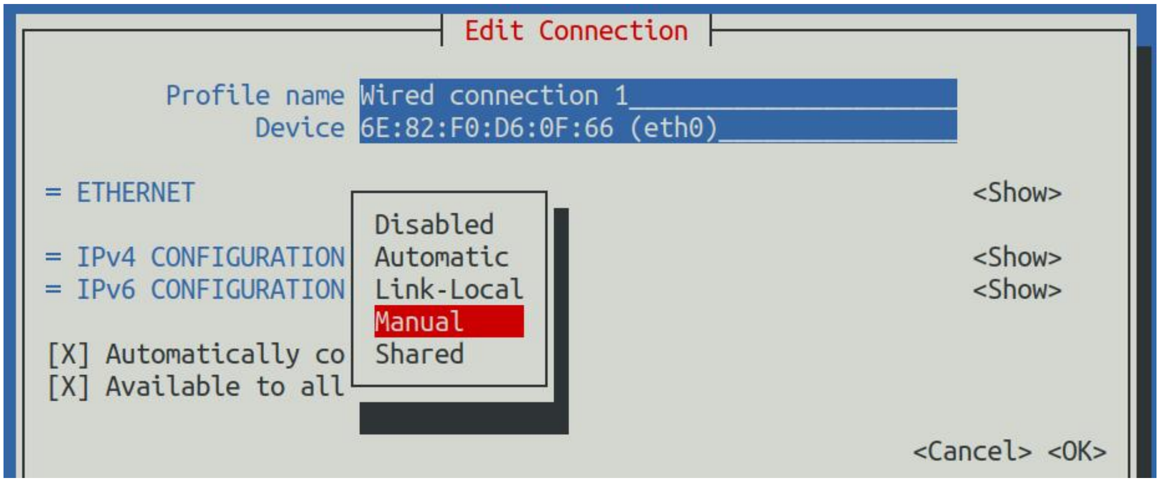
4) Then select **Edit** with the **Tab** key and press the Enter key



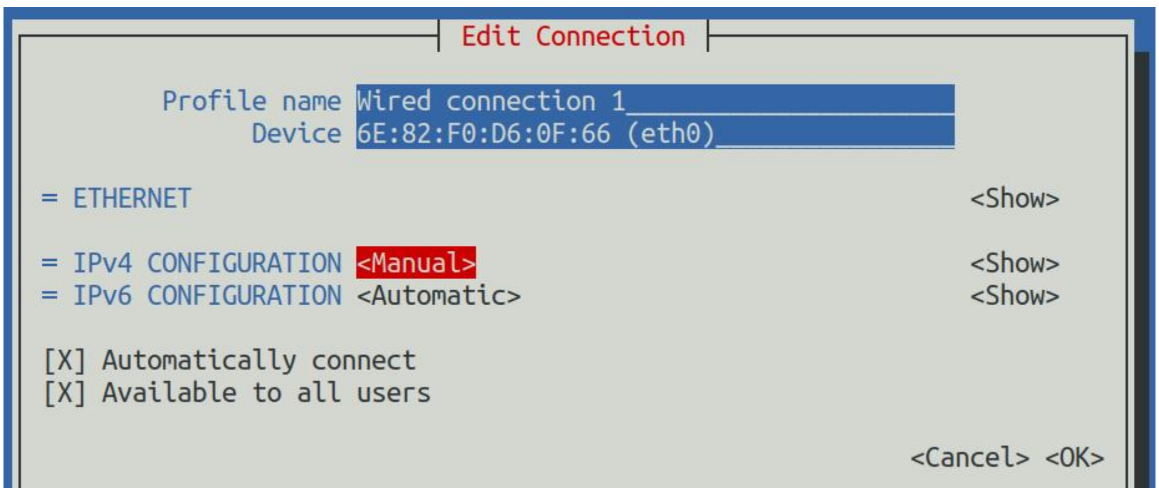
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4



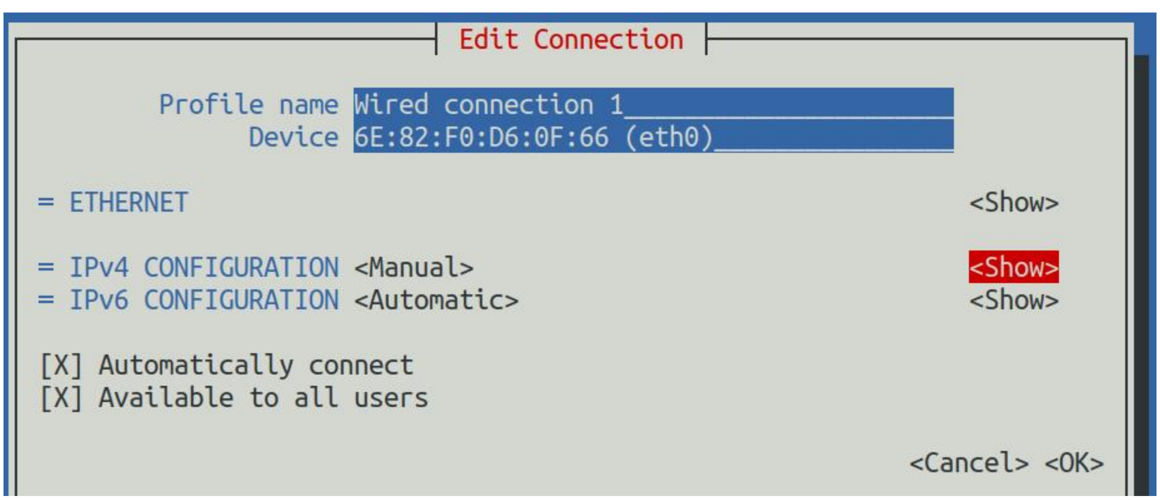
6) Then press Enter, select **Manual** with the up and down arrow keys, and press Enter to confirm



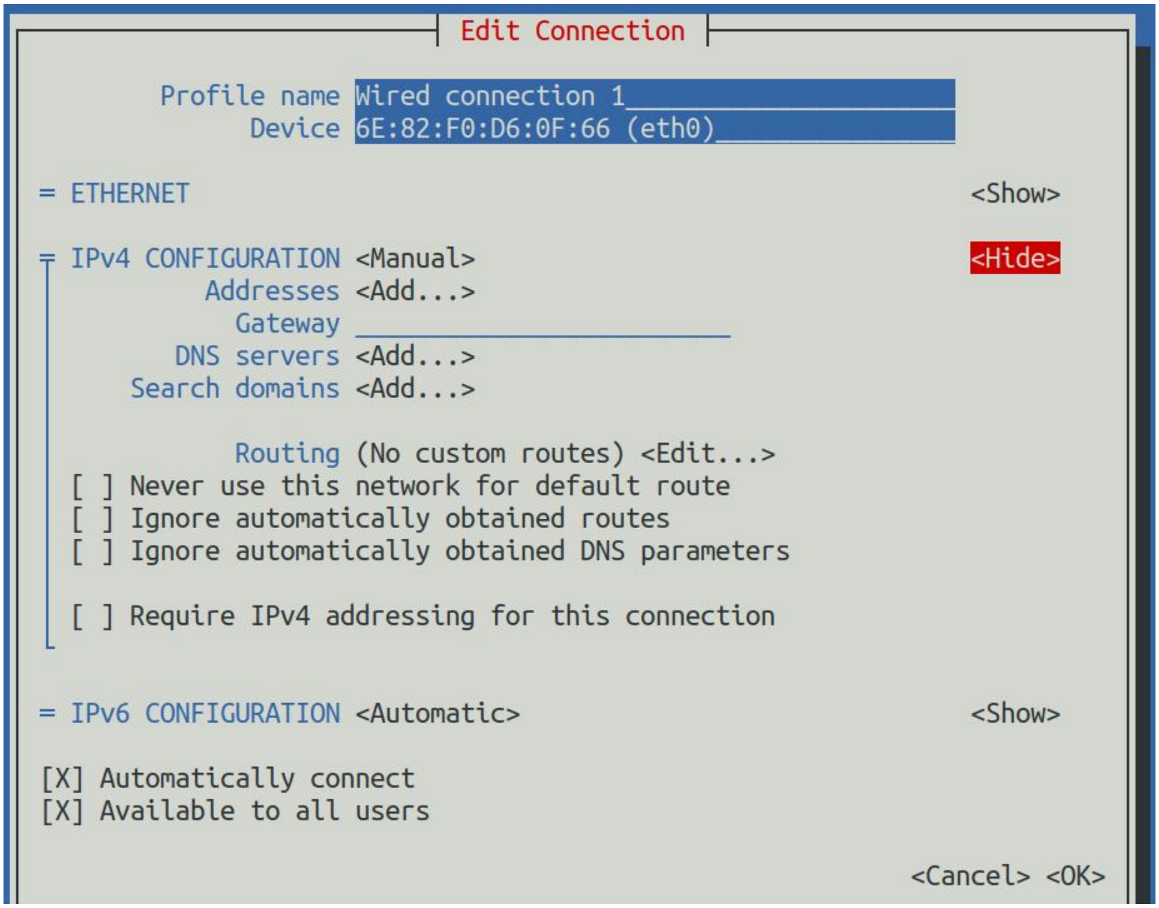
7) The display after selection is shown in the figure below



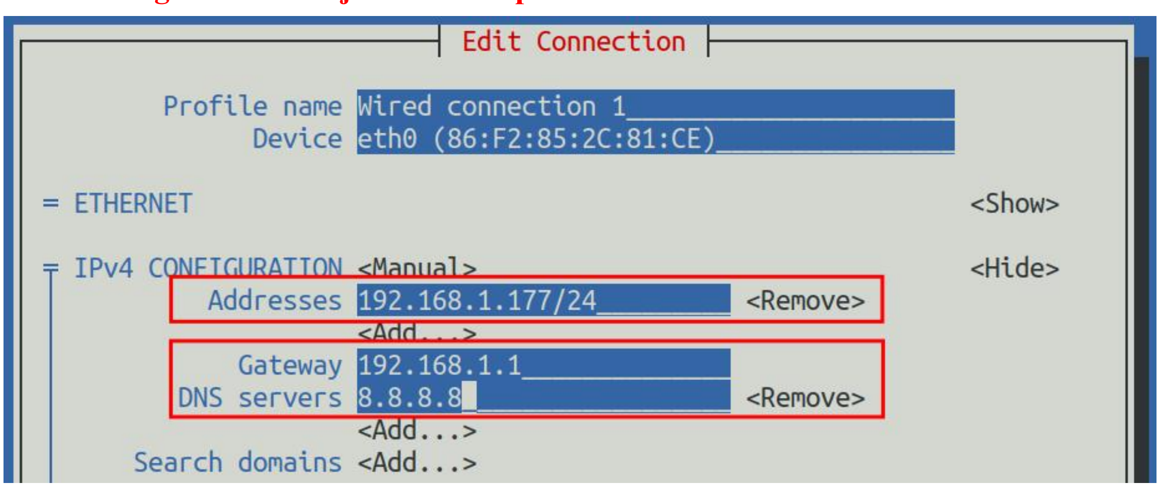
8) Then move the cursor to the **<Show>**



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

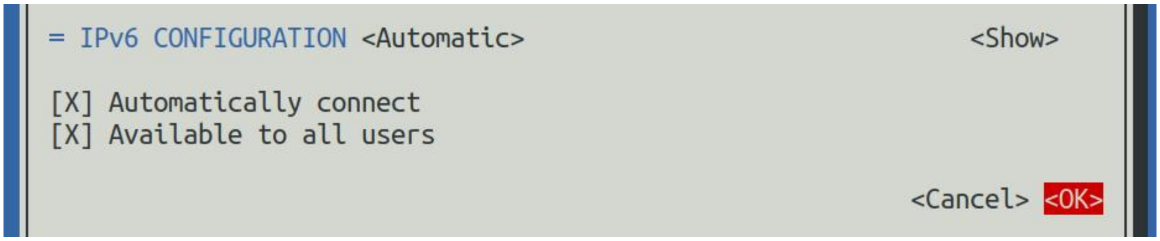


10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the position shown in the figure below (there are many other setting options in it, please explore by yourself)，**Please set it according to your specific needs, the value set in the figure below is just an example**

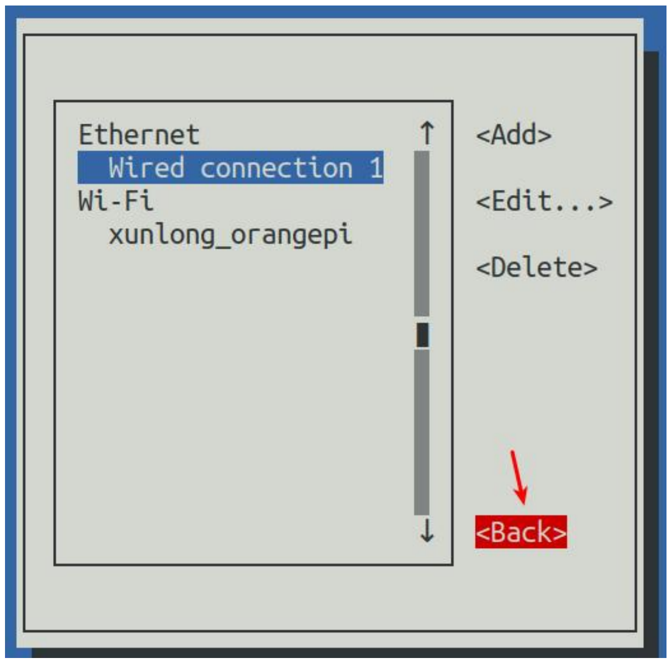


11) After setting, move the cursor to**<OK>** in the lower right corner, and press Enter to

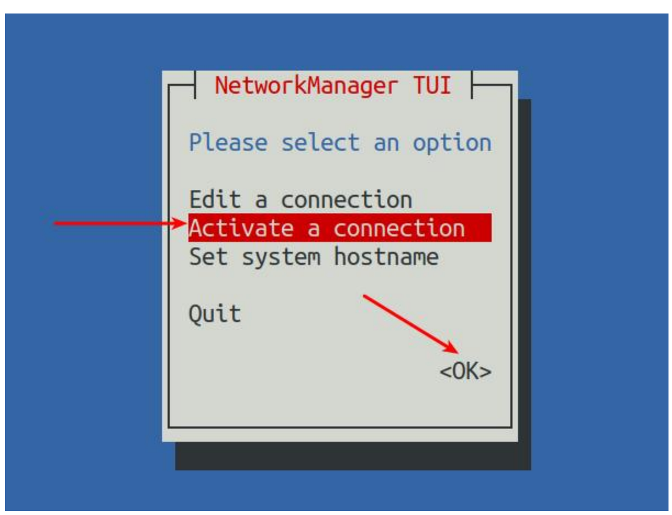
confirm



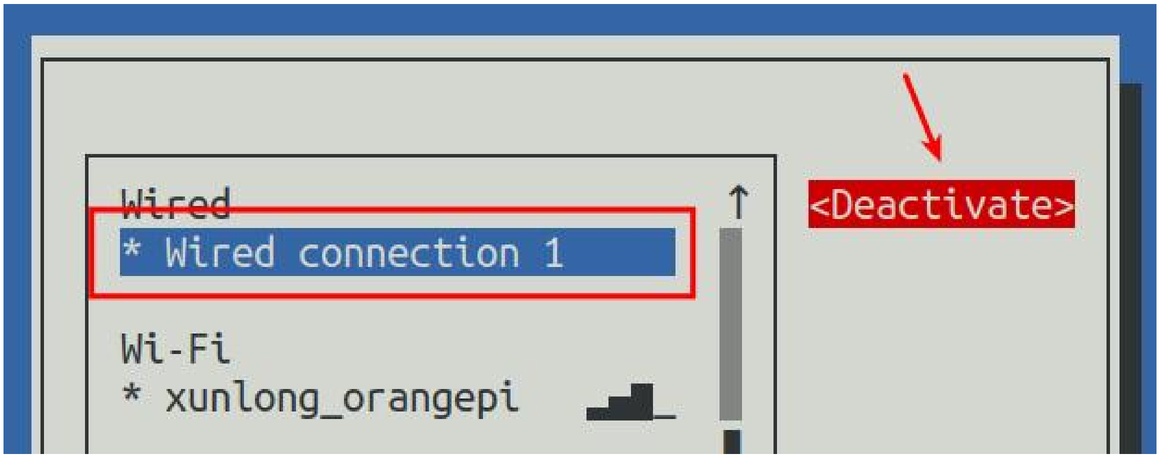
12) Then click**<Back>** to return to the previous selection interface



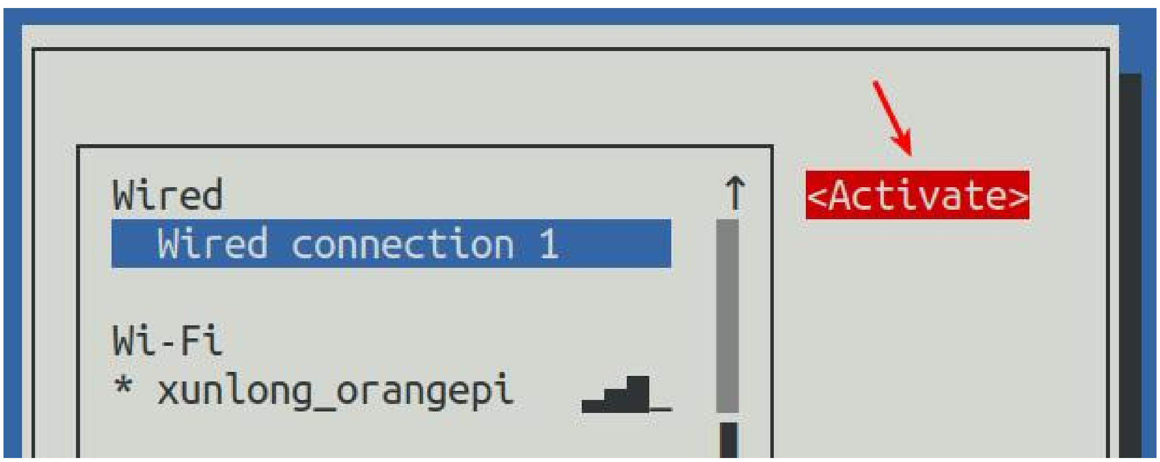
13) Then select **Activate a connection**, then move the cursor to**<OK>**, and finally click Enter



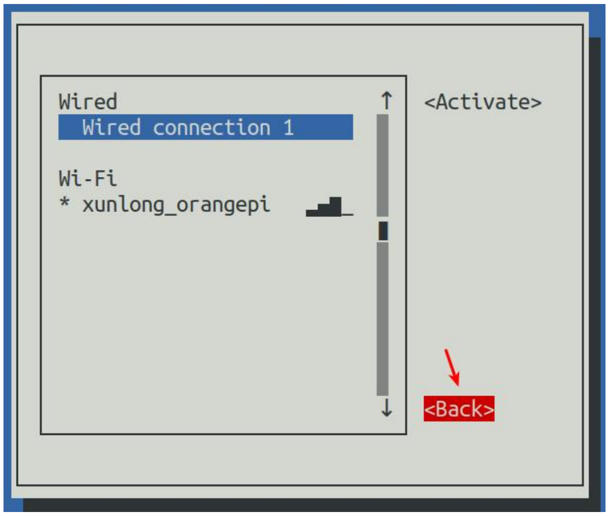
14) Then select the network interface that needs to be set, such as **Wired connection 1**, then move the cursor to**<Deactivate>**, and press Enter to disable **Wired connection 1**

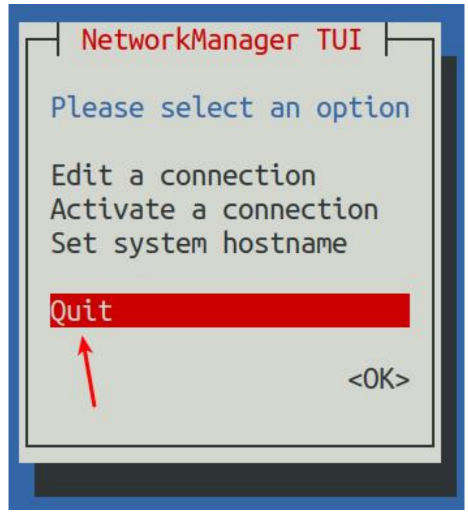


15) Then please do not move the cursor, and then press the Enter key to re-enable **Wired connection 1**, so that the static IP address set earlier will take effect



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

```
orangepi@orangepi:~$ ping 192.168.1.47 -I eth0
PING 192.168.1.47 (192.168.1.47) from 192.168.1.188 eth0: 56(84) bytes of data.
64 bytes from 192.168.1.47: icmp_seq=1 ttl=64 time=0.233 ms
64 bytes from 192.168.1.47: icmp_seq=2 ttl=64 time=0.263 ms
64 bytes from 192.168.1.47: icmp_seq=3 ttl=64 time=0.273 ms
64 bytes from 192.168.1.47: icmp_seq=4 ttl=64 time=0.269 ms
64 bytes from 192.168.1.47: icmp_seq=5 ttl=64 time=0.275 ms
```

```
^C
--- 192.168.1.47 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4042ms
rtt min/avg/max/mdev = 0.233/0.262/0.275/0.015 ms
```

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

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

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

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

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

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

3) Then enter the following command, where

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

b. After **ipv4.addresse** 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 the root user to log in to the system. Before logging in with ssh, you first need to ensure that the Ethernet or wifi network is connected, and then use the ip addr command or check the router to obtain the IP address of the development board.

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

1) Obtain the IP address of the development board

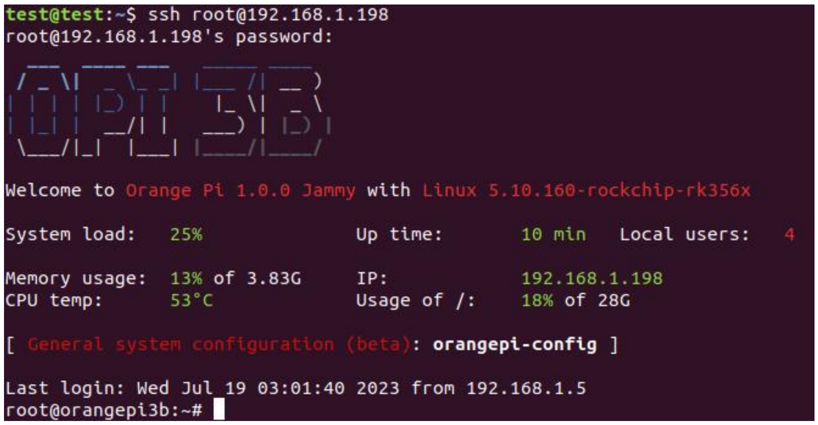
2) Then you can remotely log in to the Linux system through the ssh command

```
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 look for other reasons.

3) After successfully logging in to the system, the display is as shown in the figure below



If ssh cannot log in to the Linux system normally, please first check whether the IP address of the development board can be pinged. If the ping is ok, you can log in to the Linux system through the serial port or HDMI display and then enter the following command on the development board and try again. Is it possible to connect:

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

If it still doesn't work, try to reset the system.

3.7.2. SSH remote login development board under Windows

1) First obtain the IP address of the development board

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

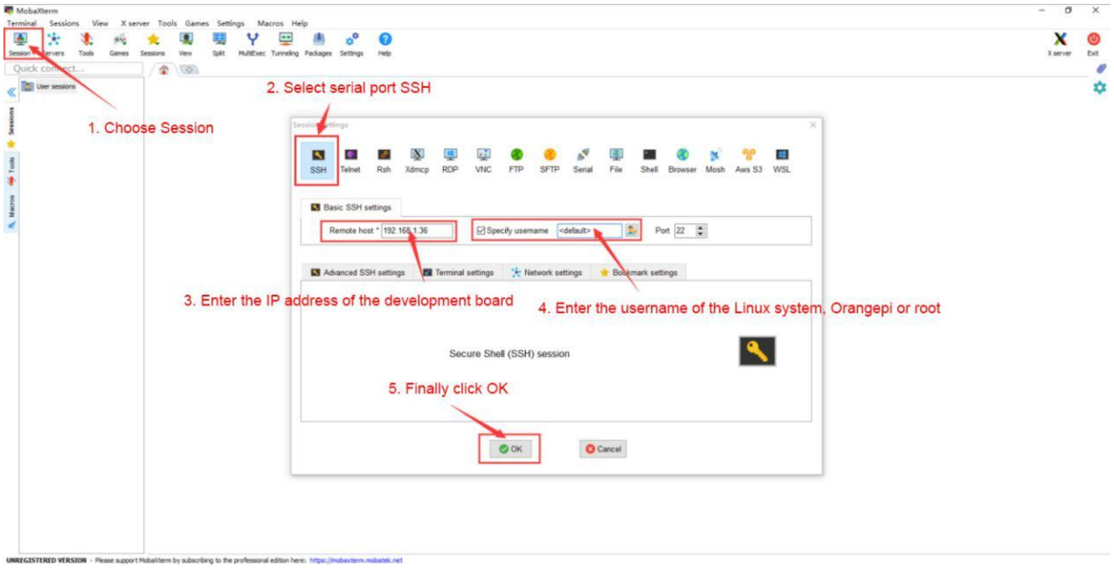
a. Open **Session**

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

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

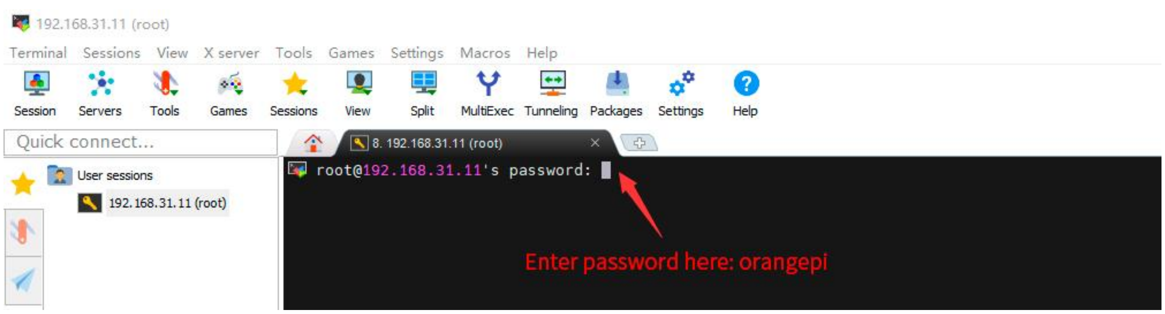
d. Then enter the user name **root** or **orangepi** of the Linux system in **Specify username**

e. Finally click **OK**

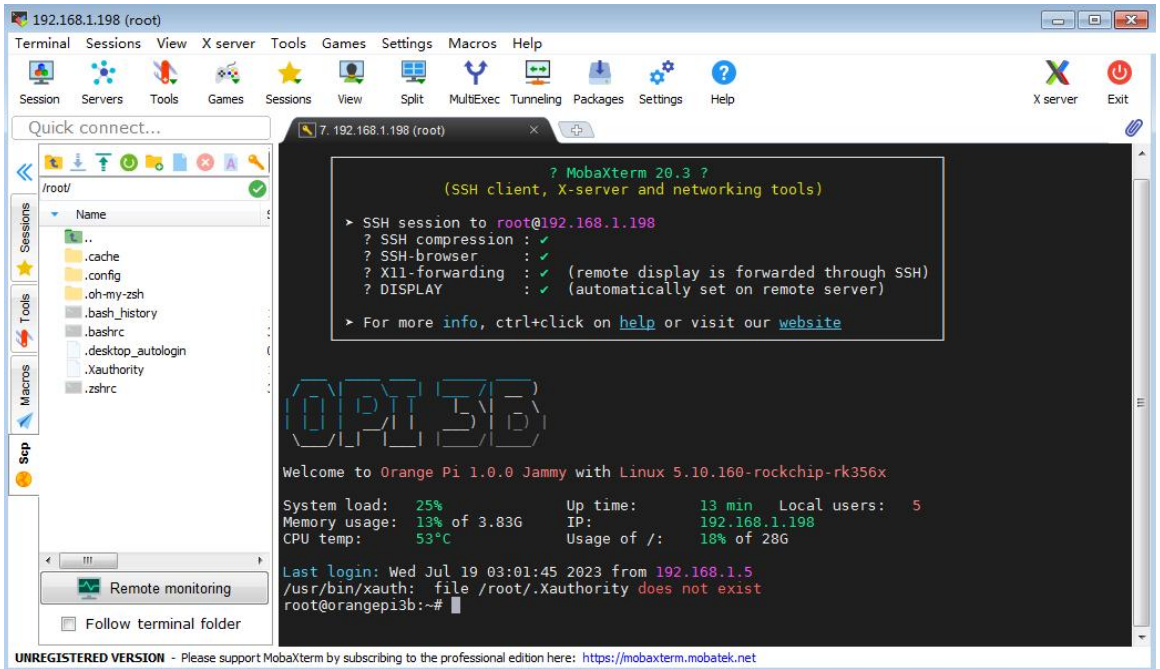


3) Then you will be prompted to enter a password. The default passwords for root and orangepi users are orangepi

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



4) After successfully logging in to the system, the display is as shown in the figure below



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

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

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

1) Use the scp command to upload files from the Ubuntu PC to the Linux system of the development board. The specific commands are as follows

a. **file_path**: need to be replaced with the path of the file to be uploaded

b. **orangepi**: It is the user name of the Linux system of the development board, and it can also be replaced with other ones, such as root

c. **192.168.xx.xx**: It is the IP address of the development board, please modify it according to the actual situation

d. **/home/orangepi**: The path in the Linux system of the development board, which can also be modified to other paths

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

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

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

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

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

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

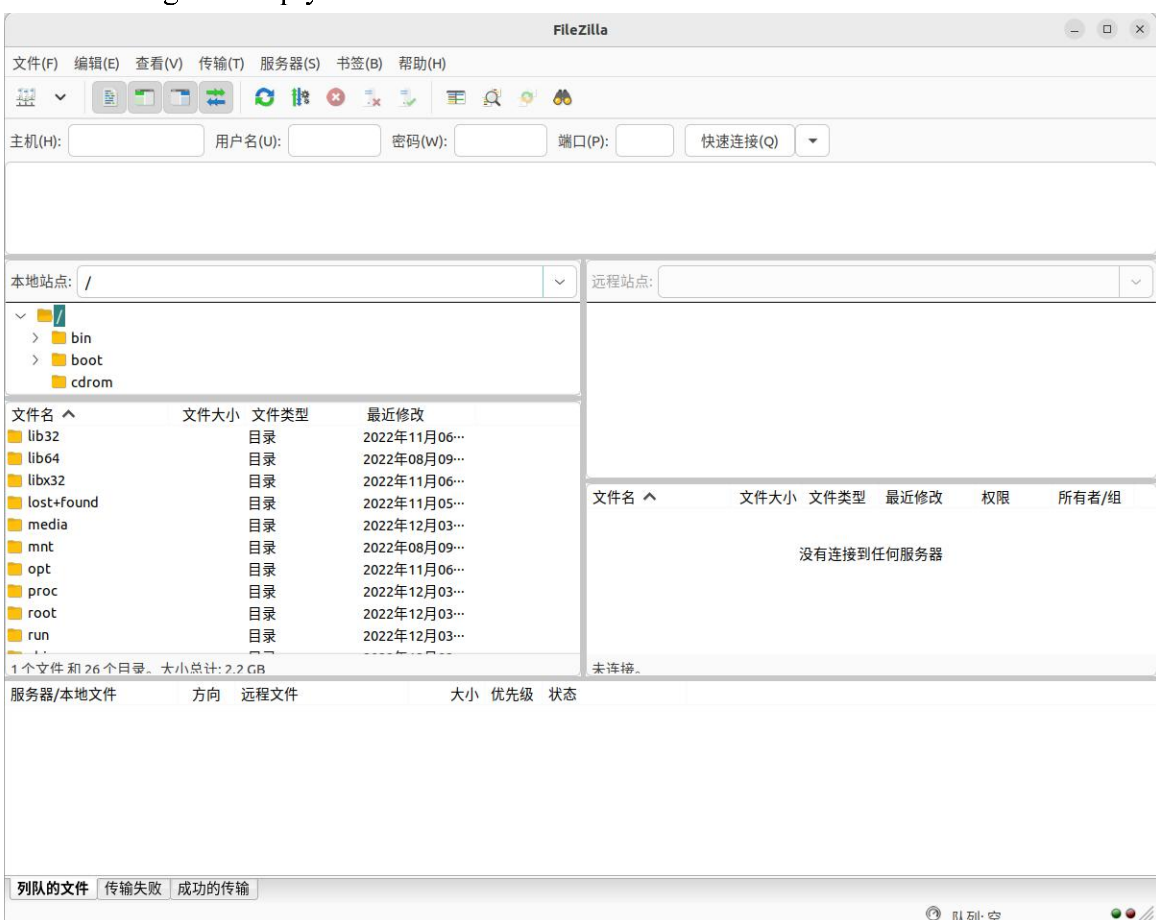
1) First install filezilla in Ubuntu PC

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

2) Then use the following command to open filezilla

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

3) The interface after filezilla is opened is as follows, and the display under the remote site on the right is empty

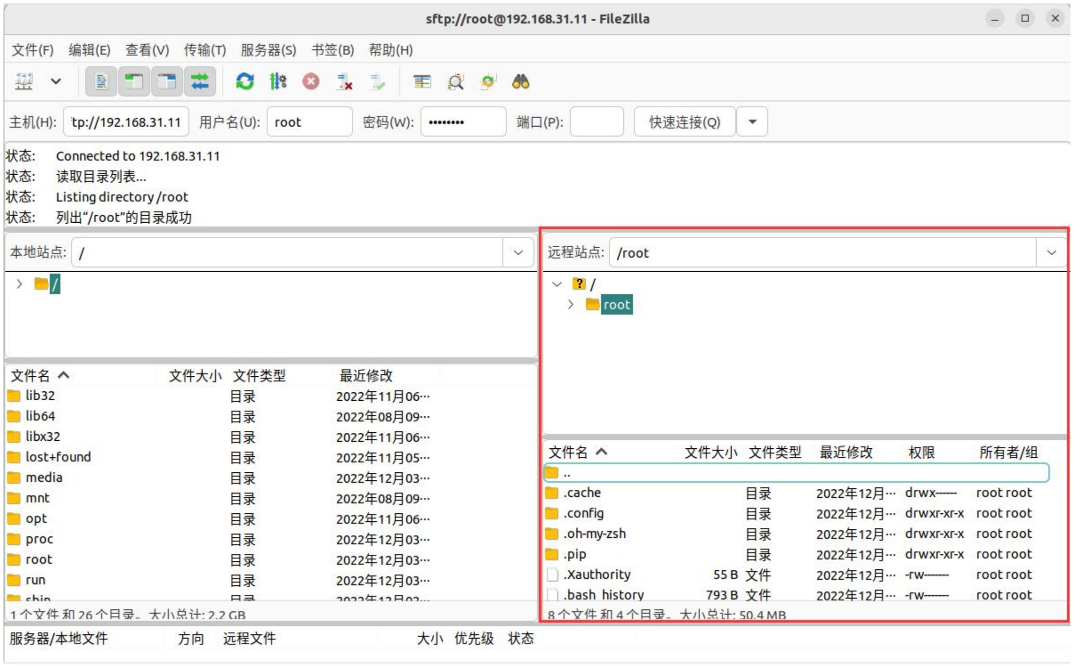


4) The method of connecting the development board is shown in the figure below

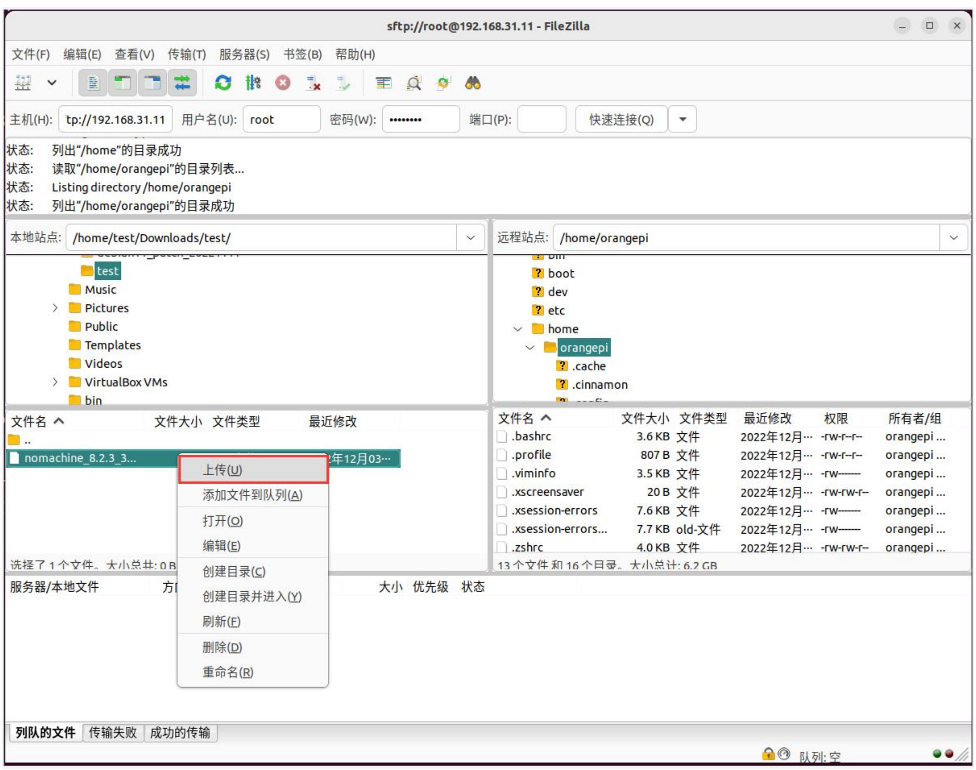
5) Then choose to **save the password**, and then click **OK**

6) Then choose to always **trust this host**, and then click **OK**

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



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



9) After the upload is complete, you can go to the corresponding path in the Linux system

of the development board to view the uploaded file

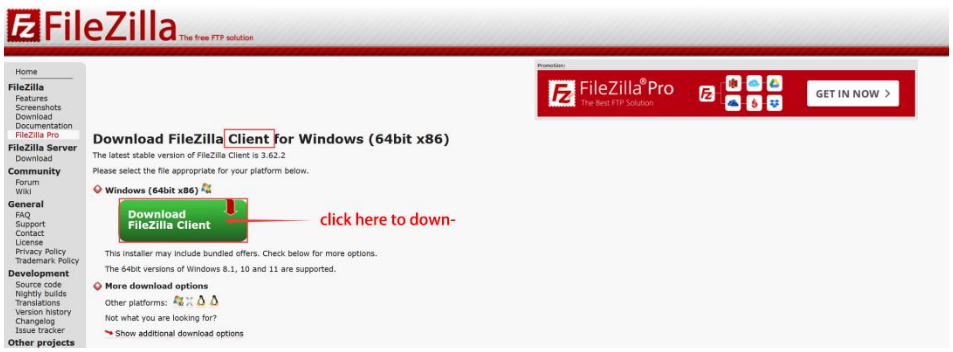
10) The method of uploading a folder is the same as that of uploading a file, so I won't go into details here

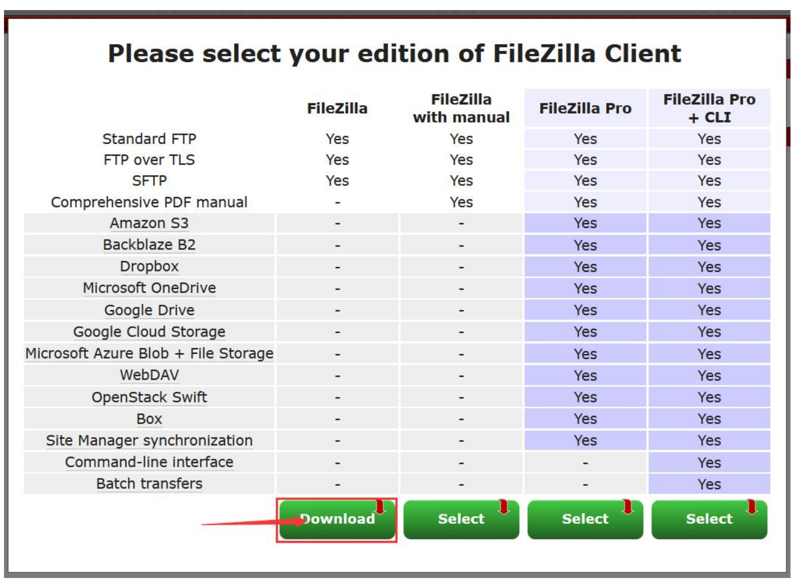
3. 8. 2. The method of uploading files from Windows PC to the Linux system of the development board

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





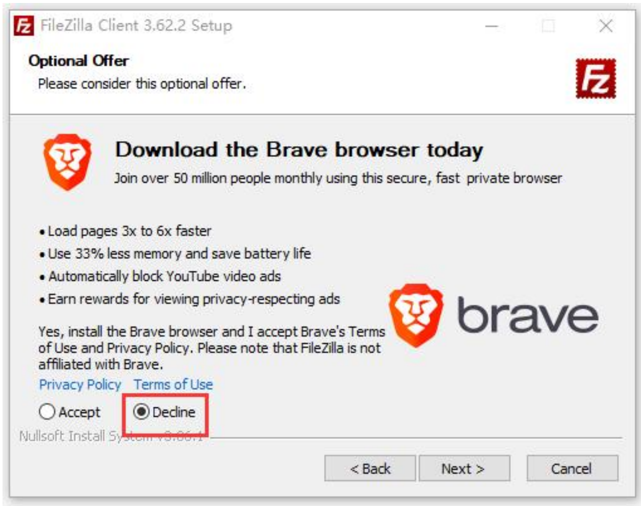
Please select your edition of FileZilla Client

	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

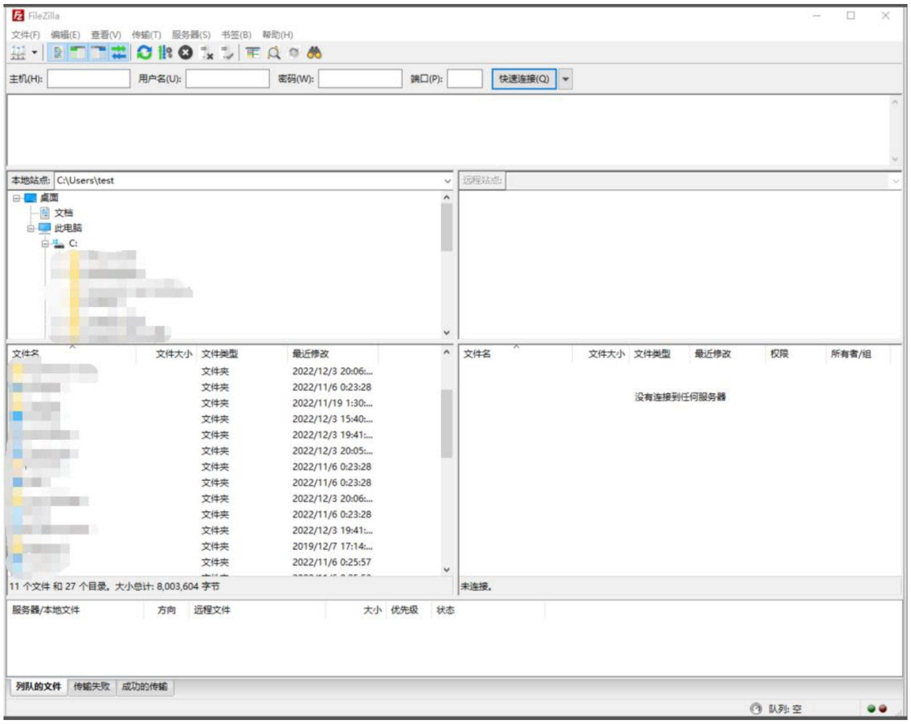
2) The downloaded installation package is as follows, and then double-click to install directly

FileZilla_Server_1.5.1_win64-setup.exe

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

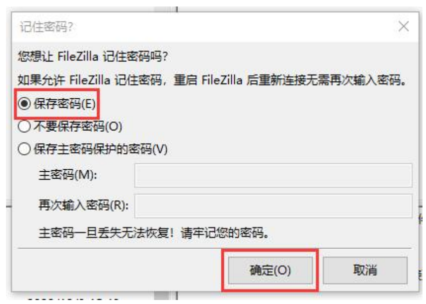


3) The interface after filezilla is opened is as follows, and the display under the remote site on the right is empty

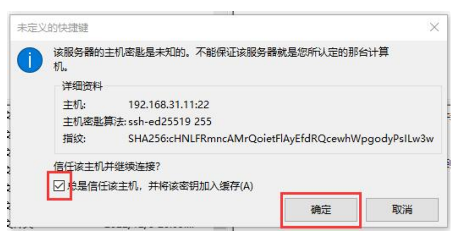


4) The method of connecting the development board is shown in the figure below:

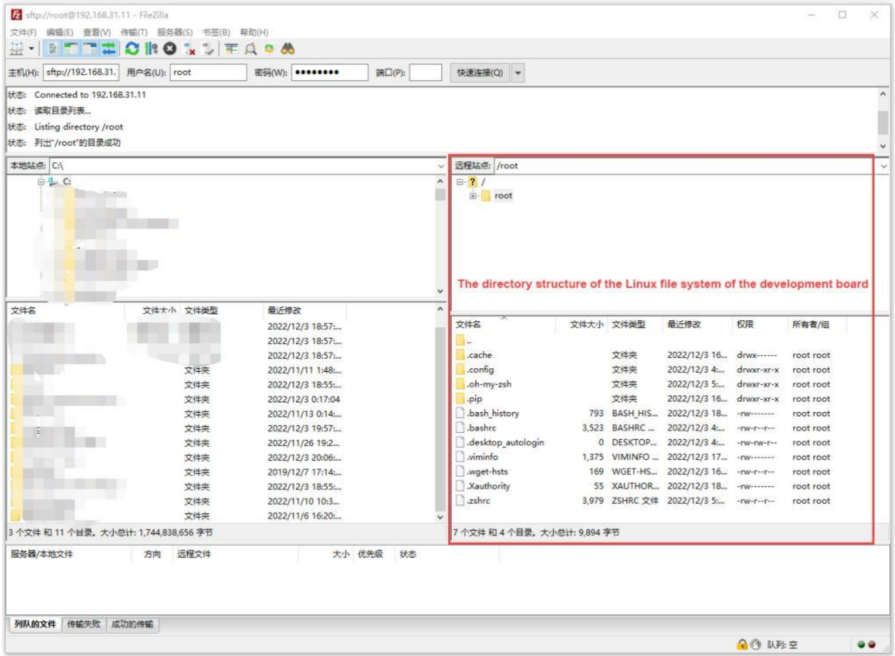
5) Then choose to **save the password**, and then click **OK**



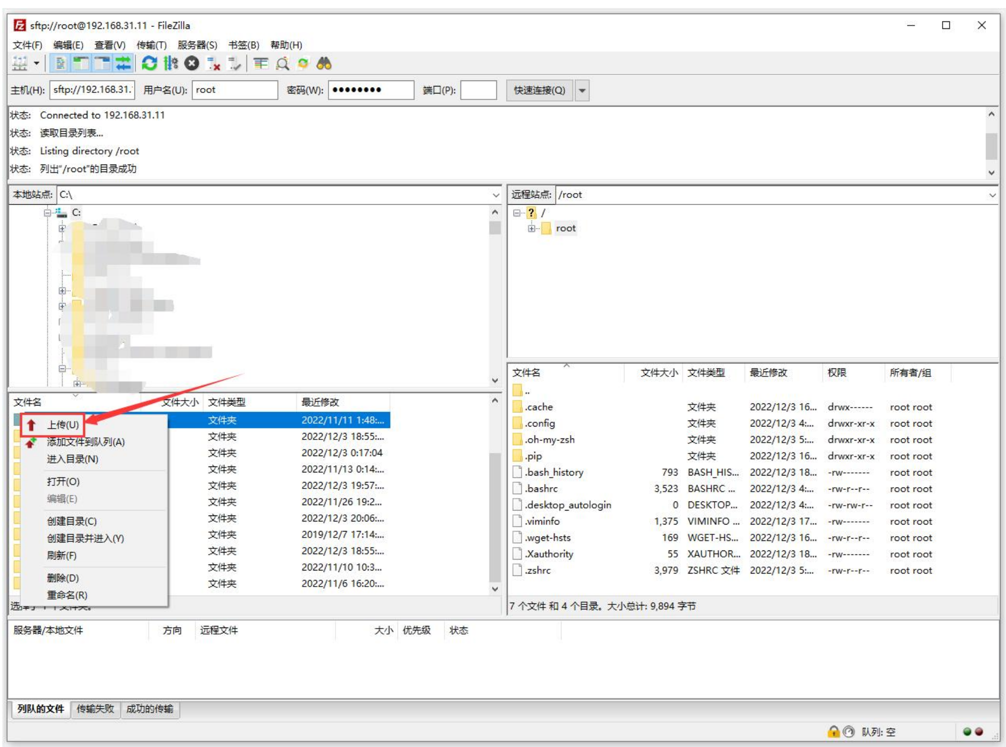
6) Then choose to always **trust this host**, and then click **OK**



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



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



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

10) The method of uploading a folder is the same as that of uploading a file

3.9. HDMI test

3.9.1. HDMI display test

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

2) After starting the Linux system, if the HDMI monitor has image output, it means that the HDMI interface is in normal use

Note that although many notebook computers have an HDMI interface, the HDMI interface of the notebook generally only has the output function, and does not have the function of HDMI in, that is to say, the HDMI output of other devices cannot be displayed on the notebook screen.

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

When the HDMI is not displayed, please check whether the HDMI cable is plugged in tightly. After confirming that there is no problem with the connection, you can change a different screen and try to see if it is displayed.

3.9.2. HDMI to VGA display test

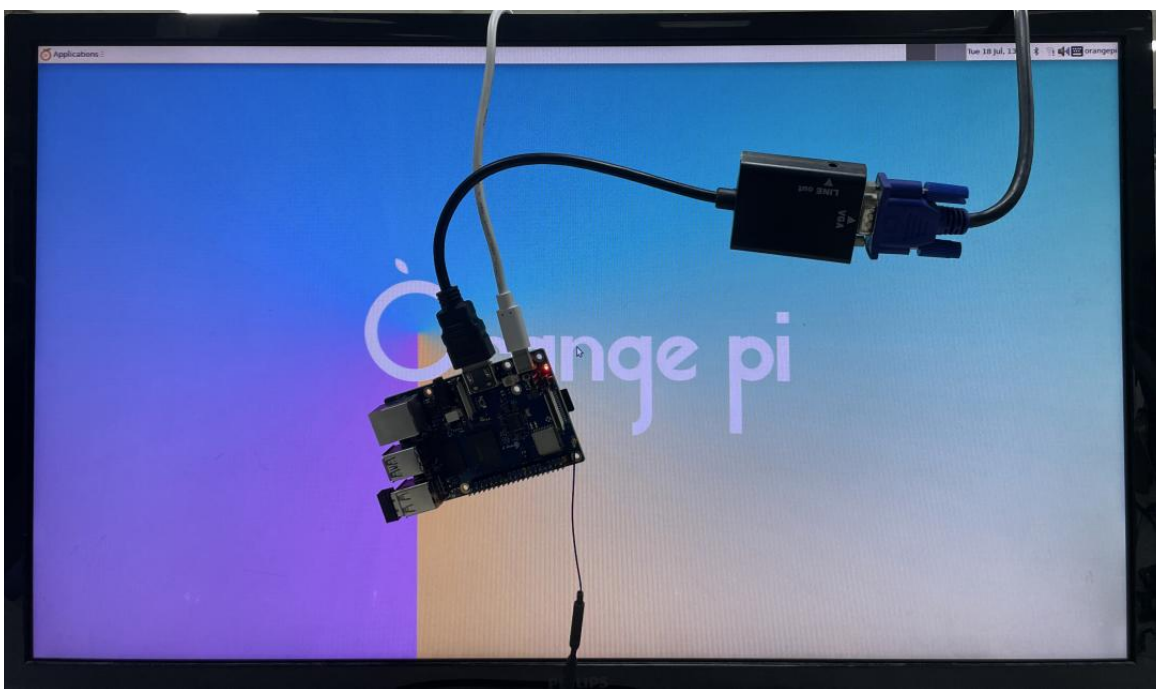
1) First, you need to prepare the following accessories

a. HDMI to VGA converter

b. A VGA cable

c. A monitor or TV that supports VGA interface

2) HDMI to VGA display test as shown below

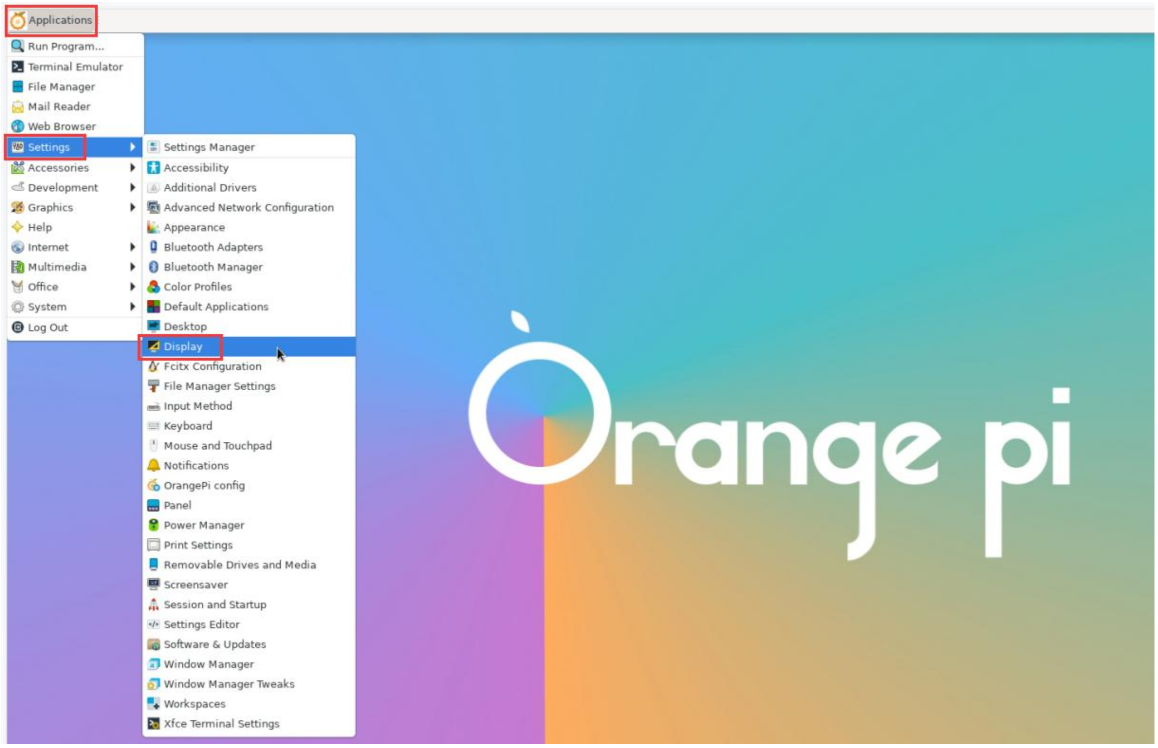


When using HDMI to VGA display, the development board and the Linux system of the development board do not need to make any settings, only the HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI to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

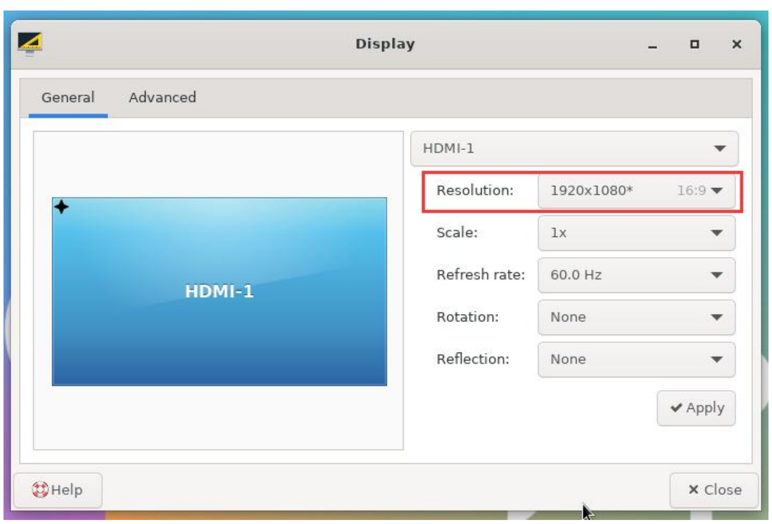
converter, VGA cable and monitor.

3. 9. 3. HDMI resolution setting method

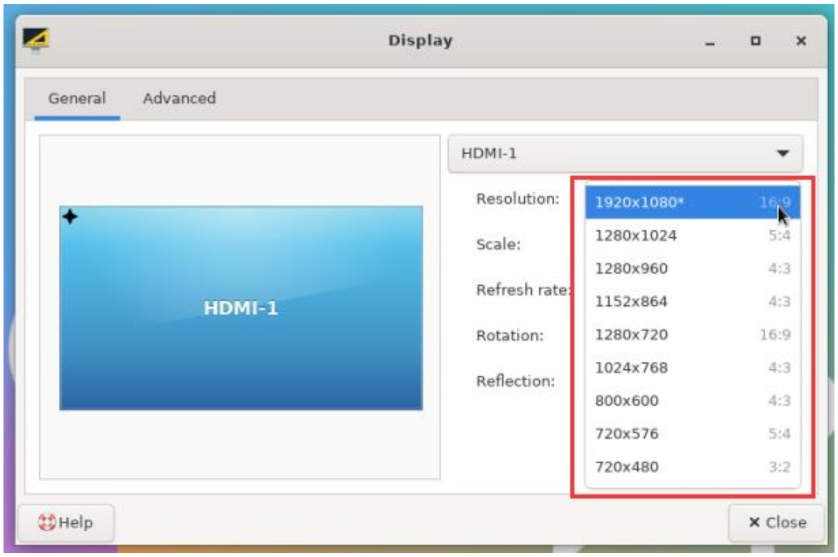
1) First open **Display** in **Settings**



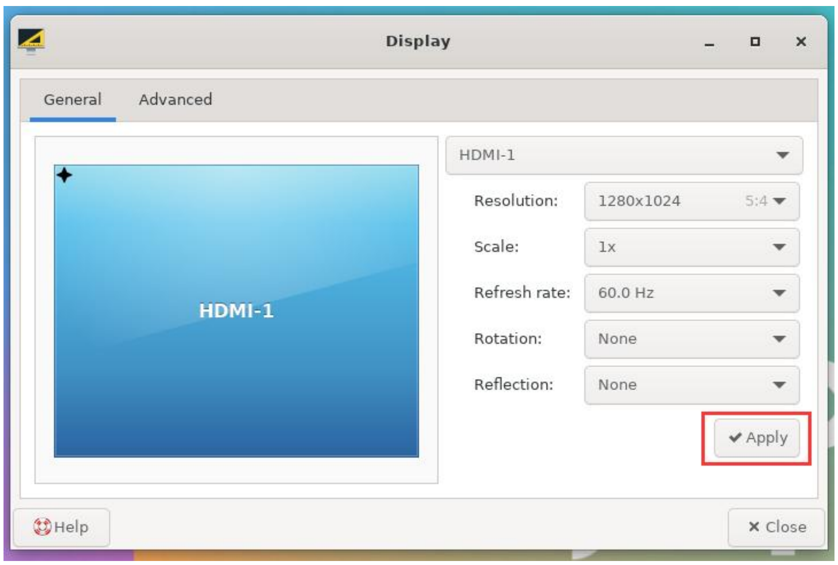
2) Then you can see the current resolution of the system



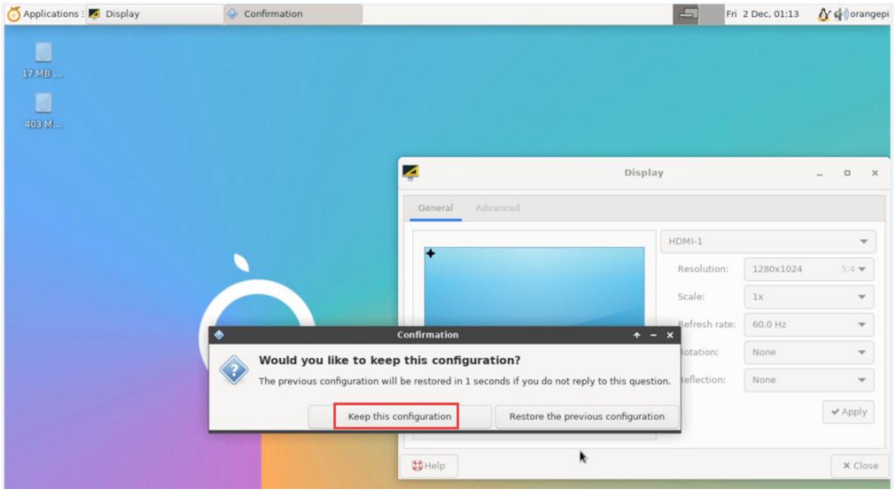
3) Click the drop-down box of Resolution to see all resolutions currently supported by the monitor



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



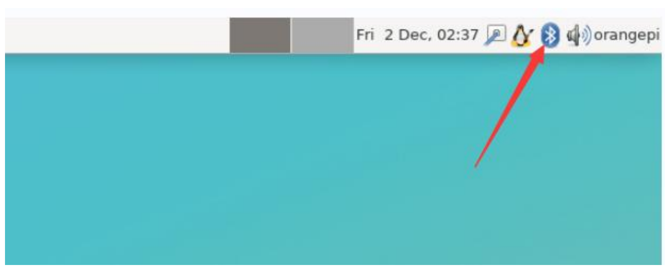
5) After the new resolution is set, select **Keep the configuration**



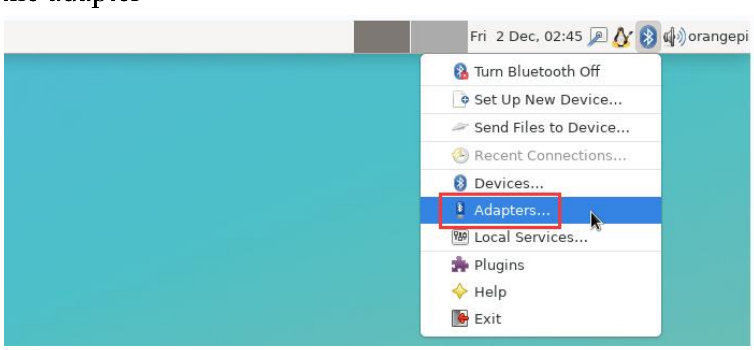
3. 10. How to use Bluetooth

3. 10. 1. Test method of desktop image

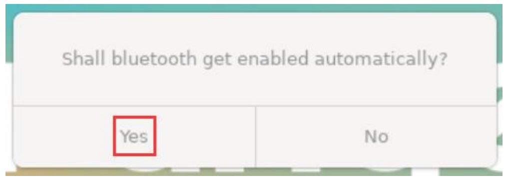
1) Click the Bluetooth icon in the upper right corner of the desktop.



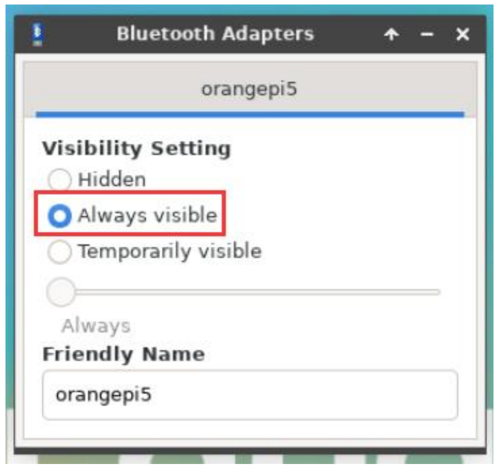
2) Then select the adapter



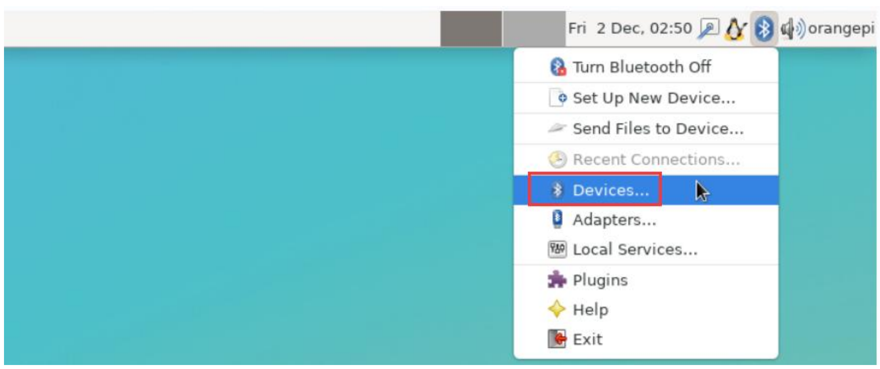
3) If there is a prompt on the following interface, please select **Yes**



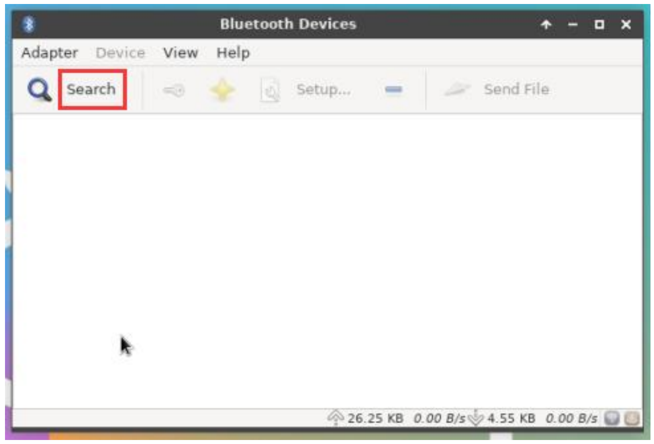
4) Then set **Visibility Setting** as **Always visible** in the Bluetooth adapter settings interface, and then close it



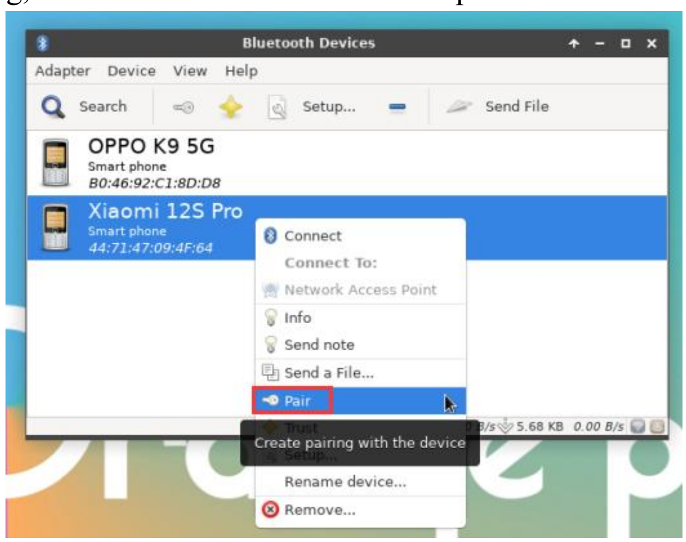
5) Then open the configuration interface of the Bluetooth device



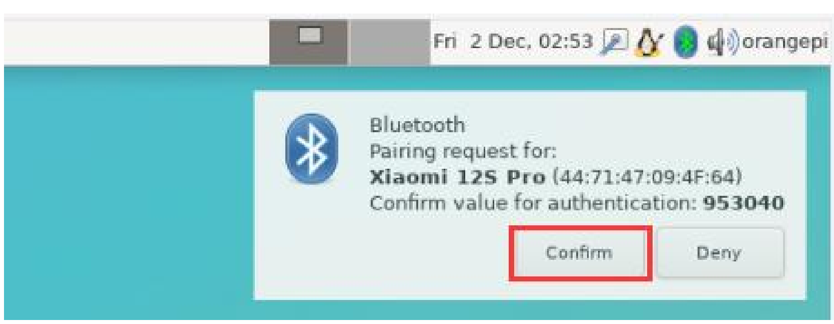
6) Click **Search** to start scanning the surrounding Bluetooth device



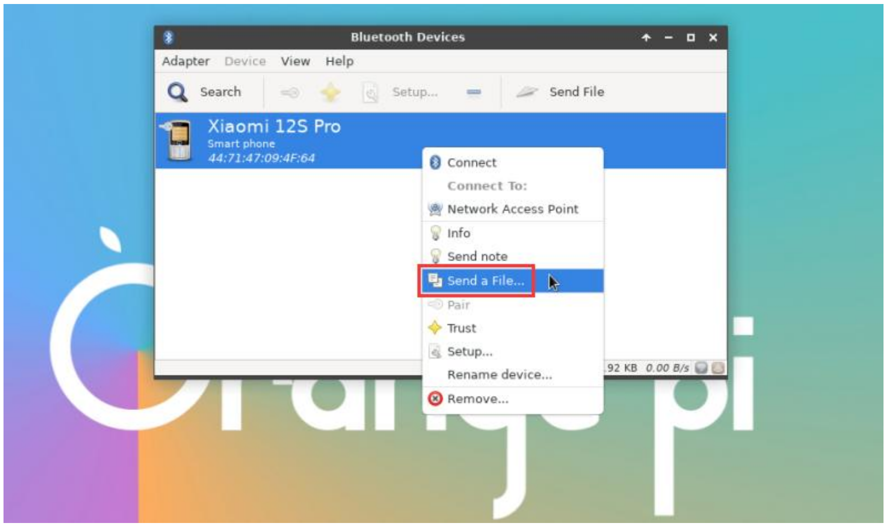
7) Then select the Bluetooth device you want to connect to, and then click the right button of the mouse to pop up the operation interface for this Bluetooth device, select **Pair** to start pairing, and the demonstration here is to pair with an Android phone



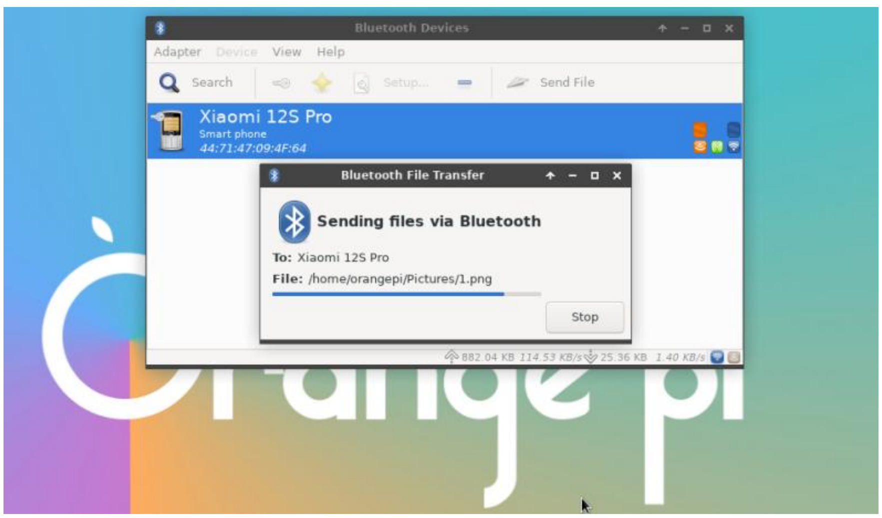
8) When pairing, the pairing confirmation box will pop up in the upper right corner of the desktop. Select **Confirm** to confirm. At this time, the mobile phone also needs to be confirmed



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



10) The interface of the sending picture is shown below



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 the USB mouse or keyboard to test

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

2) Connect Orange PI development board to HDMI display

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

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

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

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

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

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

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

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

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

3. 11. 3. USB wireless network card test

The currently **tested** USB wireless network cards are shown below. Please test it by yourself for other models of USB wireless network cards. If you cannot use it, you need to transplant the corresponding USB wireless network card driver

No.	Model	
1	RTL8723BU Support 2.4G WIFI+BT4.0	

2	RTL8811 Support 2.4G +5G WIFI	
3	RTL8821CU Support 2.4G +5G WIFI Support BT 4.2	

3. 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 RTL8723BU WIFI device node through the **sudo ifconfig** command. Please refer to **the WIFI connection test** chapter for WIFI connection and testing methods

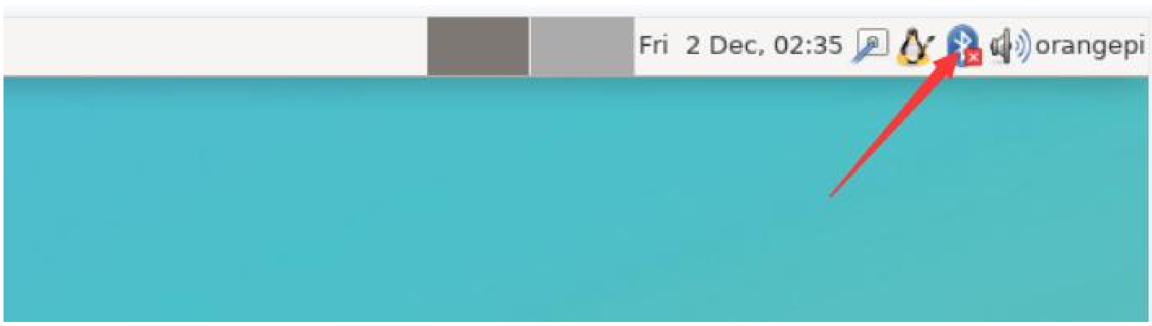
```
orangepi@orangepi:~$ sudo ifconfig wlx0013eff458ae
wlx0013eff458ae: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:13:ef:f4:58:ae  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0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

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

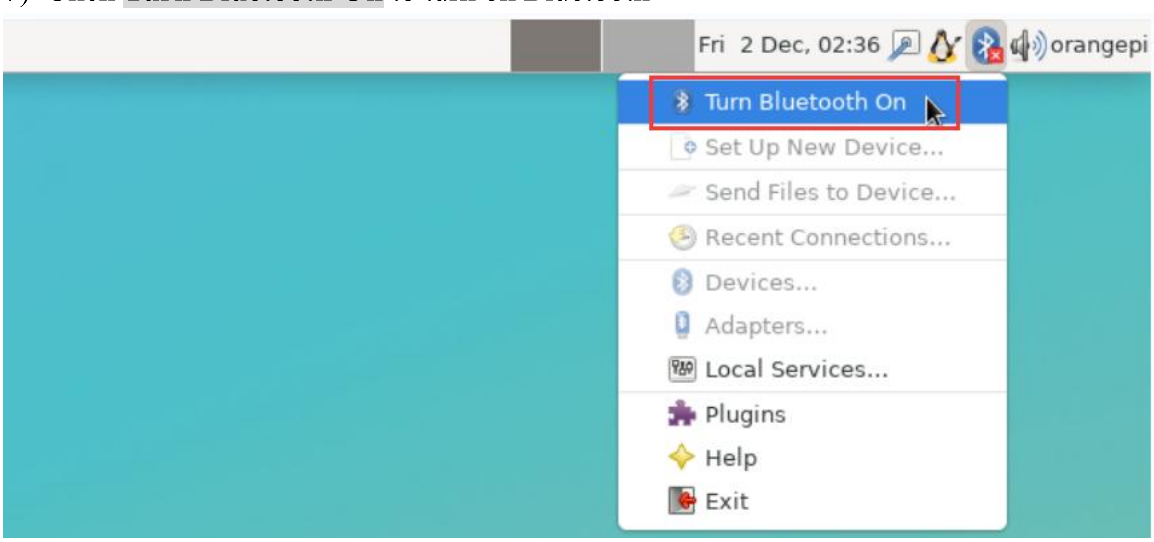
5) Then you can see the USB Bluetooth device through the **hciconfig** command

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

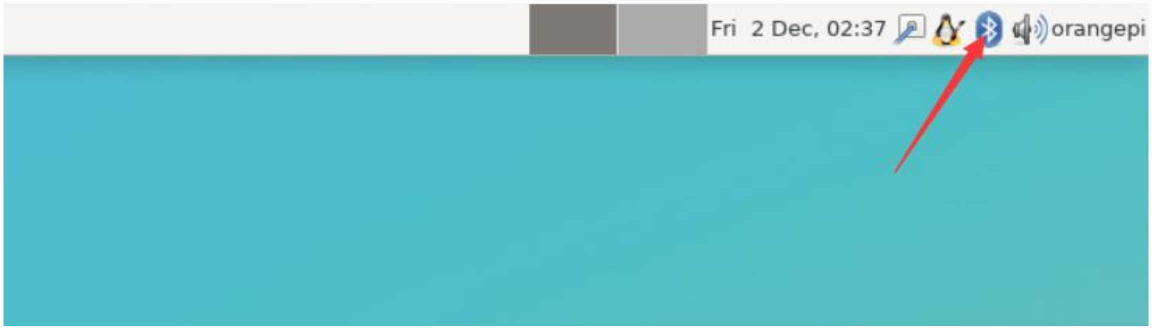
6) You can also see the Bluetooth icon on the desktop. At this time, Bluetooth has not been opened, so a red **x** will be displayed



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



8) The display after turning on Bluetooth is as follows



9) Please refer to the **Bluetooth use method** for Bluetooth test method, so I won't go into details 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 modules related to RTL8811 WIFI, and you can see that the following kernel modules have been automatically loaded through the lsmod command

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

3) Through the dmesg command, you can see the loading information of the RTL8811 module

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

4) Then you can see the WiFi device node through the **sudo ifconfig** command. Please refer to the **WiFi connection test chapter** for WIFI connection and testing methods. I won't go into details here

```
orangepi@orangepi:~$ sudo ifconfig wlx1cbfced9d260
wlx1cbfced9d260: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 1c:bf:ce:d9:d2:60  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

3. 11. 4. USB Camera Test

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

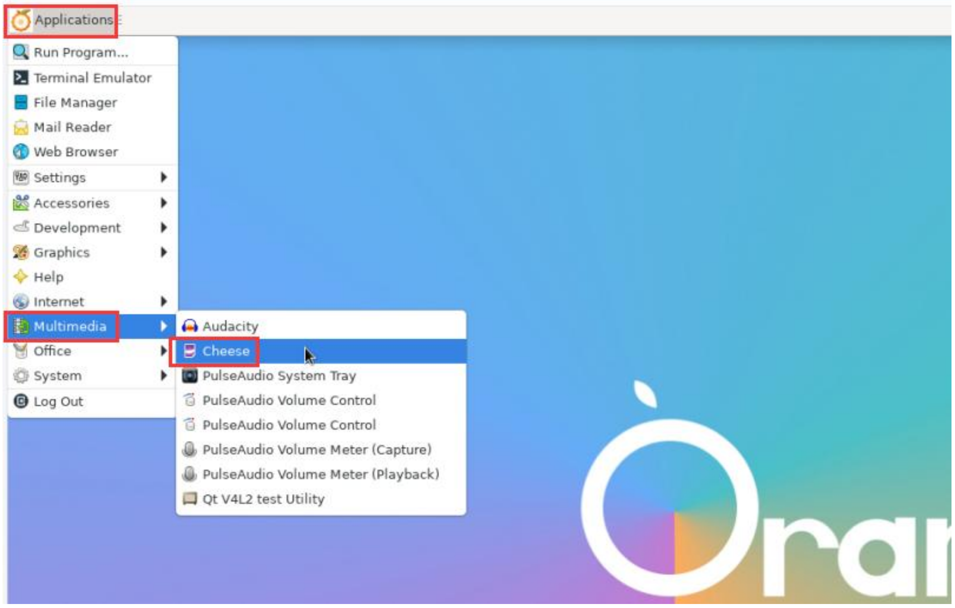
2) You can see that the USB camera's device node information is /dev/video0 through the v4l2-ctl command

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

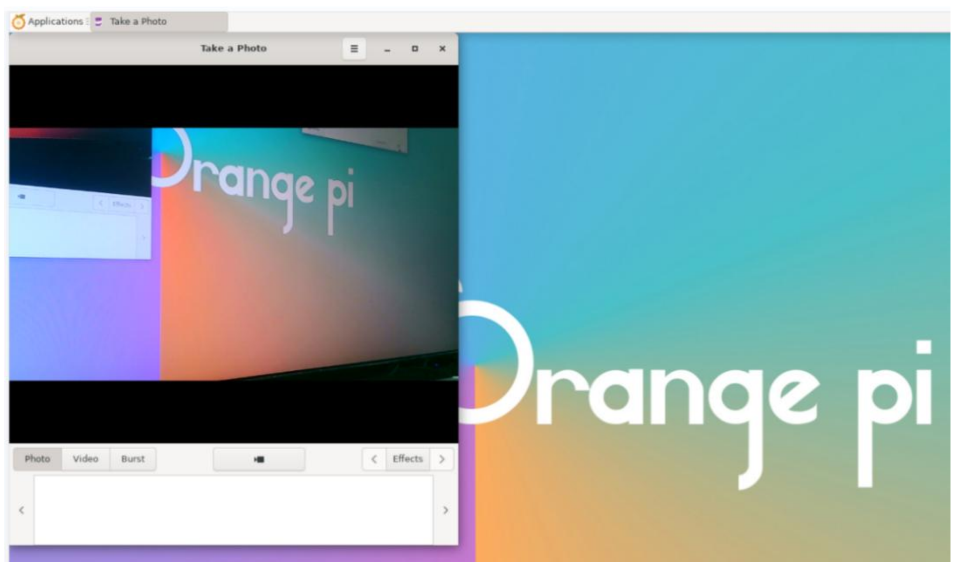
Note that the l in v4l2 is a lowercase letter l, not the number 1.

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

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



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



4) How to test the USB camera using fswebcam

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

```
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

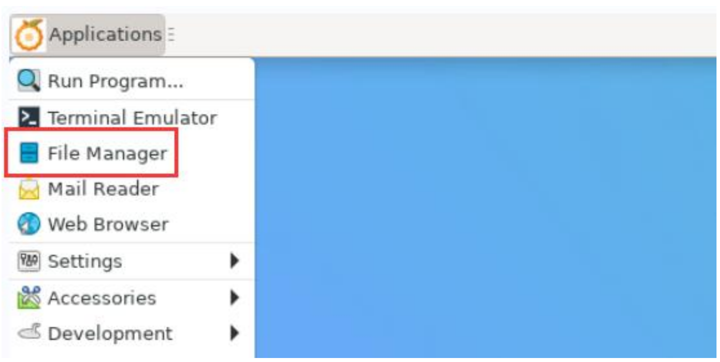
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test （Modify the IP address and path according to the actual situation）
```

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

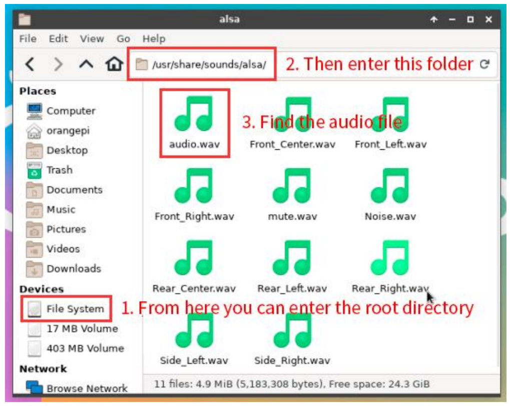
3. 12. Audio Test

3. 12. 1. Test audio methods in the desktop system

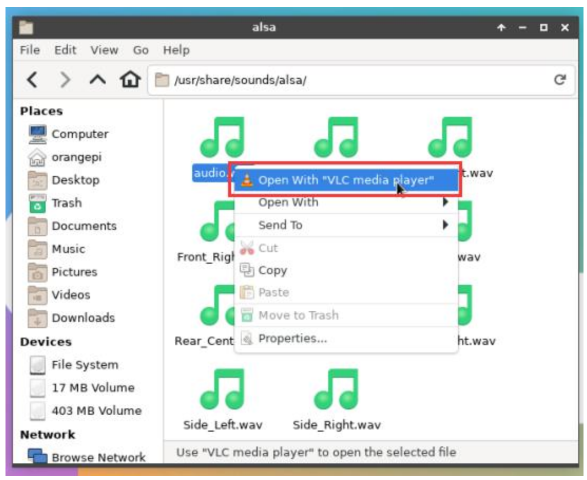
1) First open the file manager



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

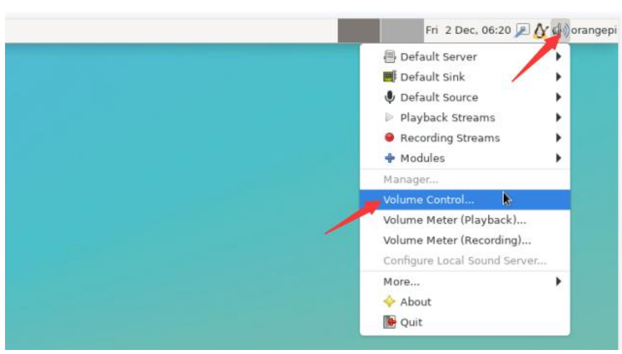


3) Then select the audio.wav file, right click and select open with vlc to start playing

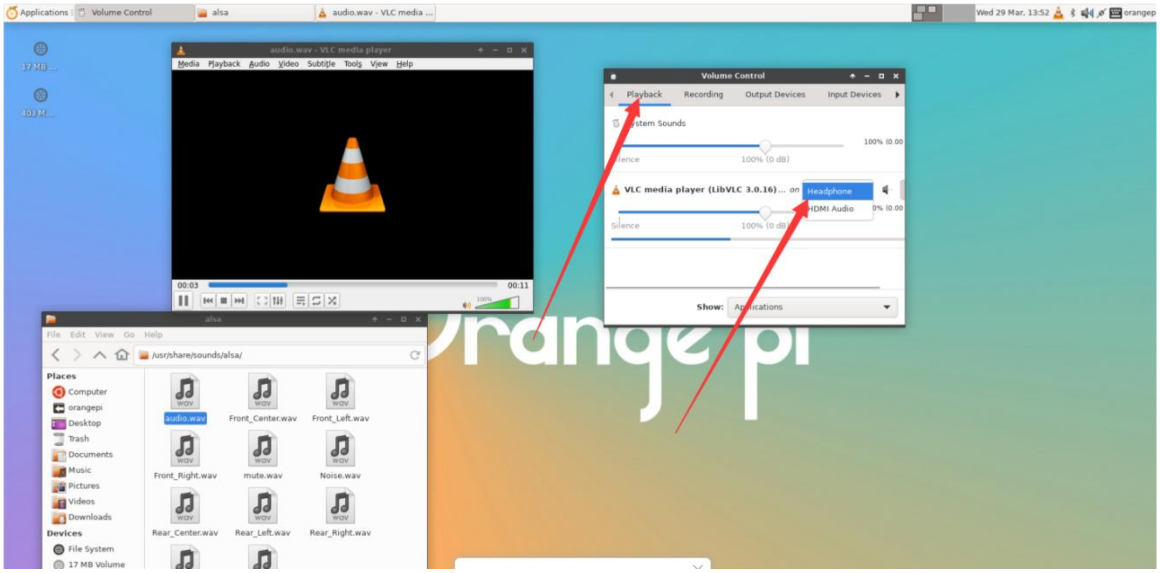


4) Methods to switch different audio equipment such as HDMI playback and headset playback

a. First open the volume control interface



b. When playing audio, the audio equipment options that play software can be used will be displayed in **Playback**. As shown in the figure below, which audio equipment you need to play here can be set.

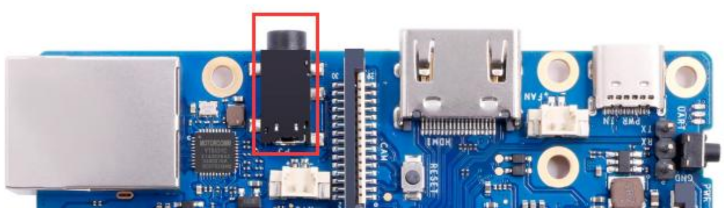


3. 12. 2. How to play audio with commands

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

Note that linux6.6 is not supported yet.

1) First insert the headset 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
   a. The output of a.linux5.10 system is as follows. 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.
   a. Playback command for linux5.10 system:

```
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 Play Test

1) First use the HDMI to HDMI cable to connect the Orange PI development board to the TV (other HDMI displays need to ensure that the audio can be played)

2) Then check the HDMI sound card serial number. From the output below, you can know that the HDMI sound card is **card 1**.
   a. The output of the linux5.10 system is as follows, card 1 is the HDMI sound card device

```
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
```

   b. The output of the b.linux6.6 system is as follows. Card 0 is the HDMI sound card device

```
orangepi@orangepi:~$ aplay -l
```

```
**** List of PLAYBACK Hardware Devices ****
card 0: HDMI [HDMI], device 0: fe400000.i2s-i2s-hifi i2s-hifi-0 [fe400000.i2s-i2s-hifi i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: RK809 [Analog RK809], device 0: fe410000.i2s-rk817-hifi rk817-hifi-0 [fe410000.i2s-rk817-hifi rk817-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio file that comes with the system. If the sound can be heard on the HDMI display or TV, it means that the hardware can be used normally.

a. Playback command for linux5.10 system:

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

b. Playback command for linux6.6 system

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

3. 12. 3. Use the command to test the recording method

Note that linux6.6 is not supported yet.

1) The Orange Pi 3B 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.

a. Commands for linux5.10 system:

```
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)
```

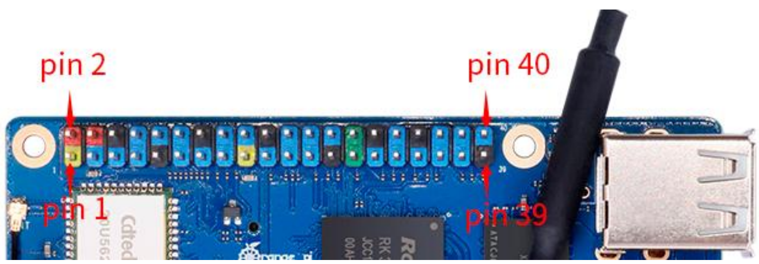
```
gpu_thermal-virtual-0
Adapter: Virtual device
temp1:           +43.8°C
```

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 explanation

2) Orange Pi 3B Development board 40 Pin interface pins, please refer to the figure below



3) The function of the Orange Pi 3B development board 40 PIN interface pins is shown in the table below.

a. Below is a complete pins 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 following form is a picture on the left half of the full table above, which can be seen 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 following form is a picture on the right half of the top table above, which can be seen 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		

4) There are a total of **28** GPIO ports in the 40pin interface. The voltage of all GPIO ports is **3.3v**

3.15. How to install wiringOP

Note that wiringOP has been pre-installed in the Linux image released by Orange Pi. Unless the code of wiringOP is updated, there is no need to re-download,

compile and install, and 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@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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 |     IN | 1 | 27 || 28 | 1 | IN     | 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 |
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
```

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 3B needs to download the code of the wiringOP next branch, please don't miss the parameter of -b next。

If there is a problem with the download code from GitHub, you can use the wiringOP source code that comes with the Linux image directly, 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@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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 |     IN | 1 | 27 || 28 | 1 | IN     | 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 |
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
```

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

Note that if you need to set overlays to open multiple configurations at the same time, please use a space to write in one line like the following space.

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 method of running blink_all_gpio program is shown below:

orangepi@orangepi3b:~$ **sudo blink_all_gpio** #Remember to add sudo permissions
[sudo] password for orangepi: #You need to enter password 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@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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 as the output mode, the third parameter needs to enter the serial number of the wPi corresponding to the pins

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

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

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

Using gpio readall, you can see the value of the No. 7 pin (V) to 0

```
root@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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
```

Using gpio readall, you can see the value of No. 7 pin (V) into 1

```
root@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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. 40pin GPIO port pull-down resistance setting method

Note that the 4 GPIO pins below Orange Pi 3B are invalid because there are 3.3V on the outside, so setting drop-down is invalid. Other pins can be set normally

```
root@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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 |     IN | 1 | 27 || 28 | 1 | IN     | 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 |
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
```

1) Below the No. 11 pin — corresponding to GPIO 3_C6-corresponding wPi serial number 5—to demonstrate how to pull-down resistance of the GPIO port

```
root@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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 of all, you need to set the GPIO port as the input mode. The third parameter needs to enter the serial number of the wPi corresponding to the pins

```
root@orangepi:~/wiringOP# gpio mode 5 in
```

3) After the setting is set to input mode, execute the following command to set the GPIO port as the pull-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

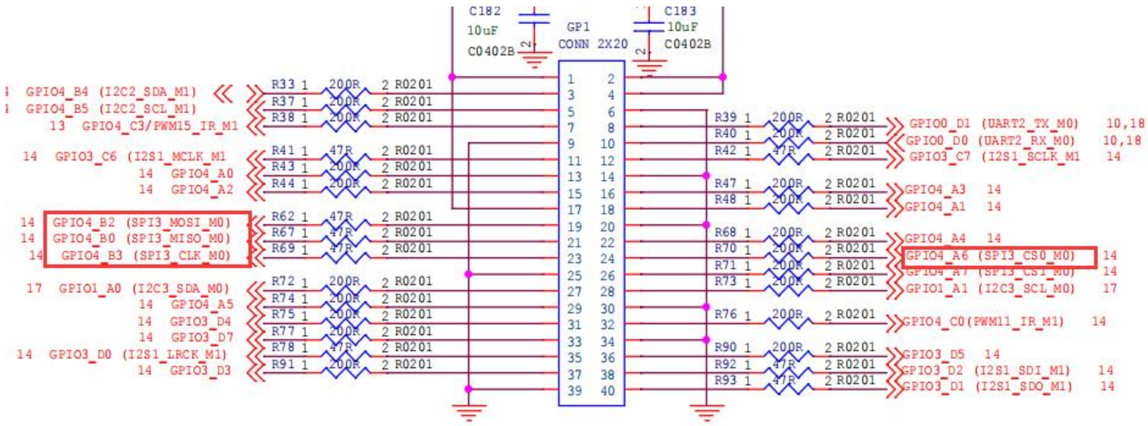
```
root@orangepi:~/wiringOP# gpio mode 5 down
```

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) From the schematic diagram of the 40PIN interface, the SPI available for Orange Pi 3B is spi3



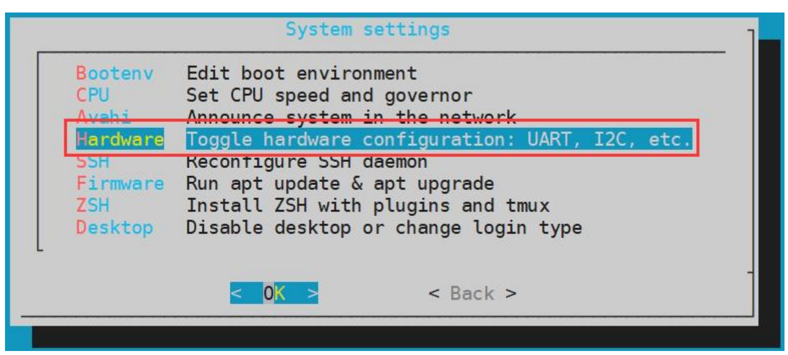
2) In the linux system, the SPI in the 40 pin is closed by default, and it needs to be opened manually before it can be used. The detailed steps are as follows:

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

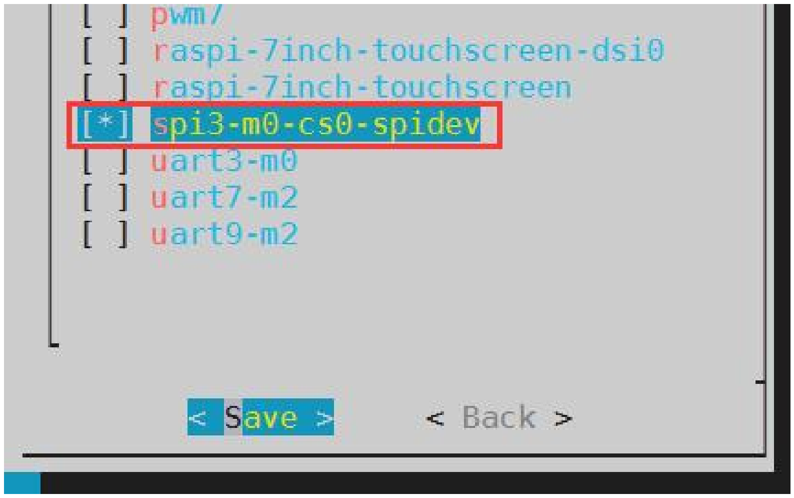
```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**

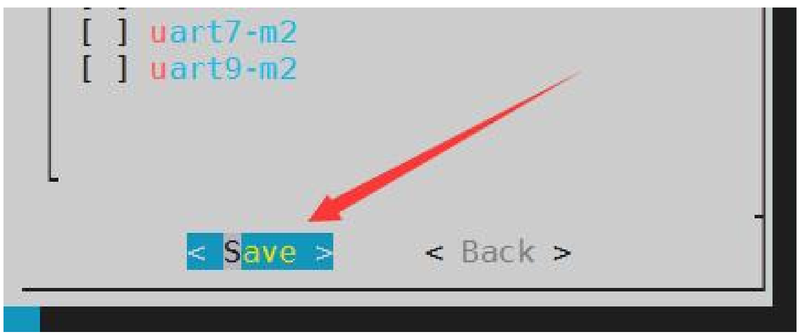
c. Then select **Hardware**



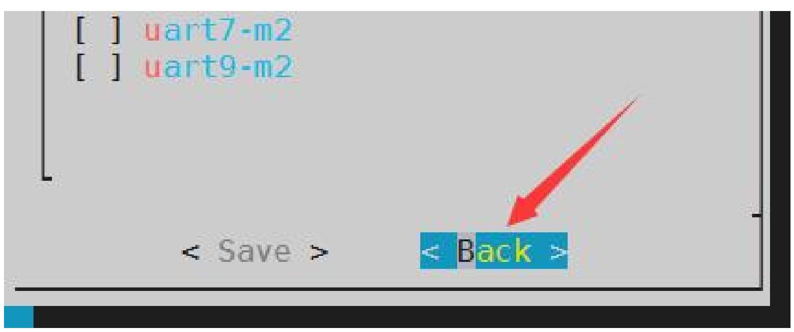
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select the SPI configuration you want to open



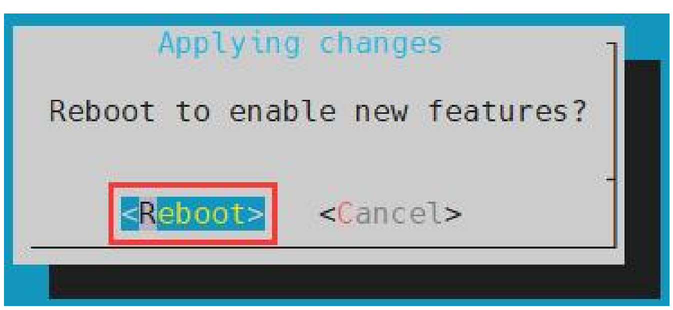
e. Then select **<Save>** to save



f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration 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, 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 then run the output of spidev_test as follows, you can see the sending and receiving same data

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/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) From the table below, the I2C available for Orange Pi 3B is I2C2, I2C3, and I2C4 a total of three groups of I2C bus.

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, and i2c3_m0 and uart3_m0 are also referenced. Open at the same time, i2c3_m0 and uart3_m0 are also pin multiplexed, and both cannot be opened at the same time

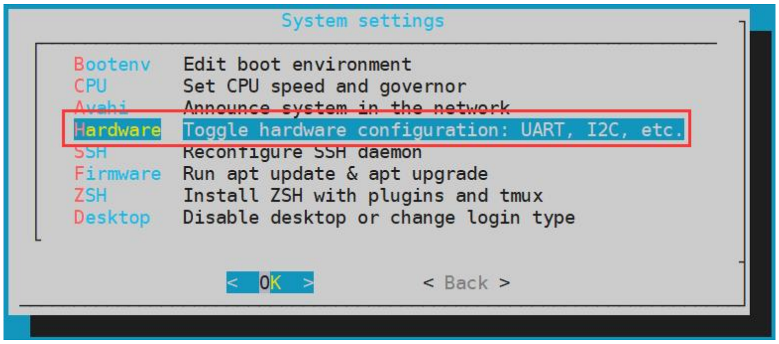
2) In the linux system, the I2C bus in the 40 pin is closed by default, and it needs to be opened manually to use it. The detailed steps are as follows:

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

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

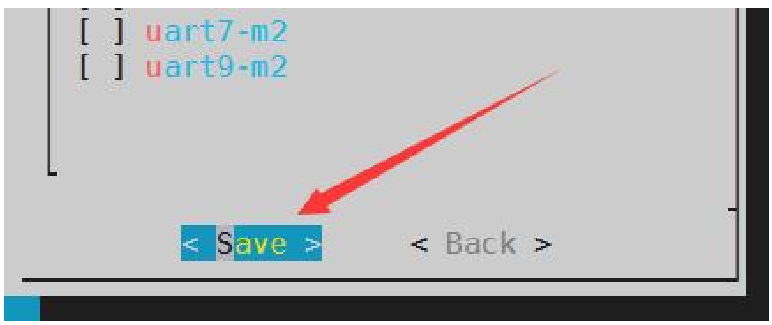
b. Then select **System**

c. Then select **Hardware**

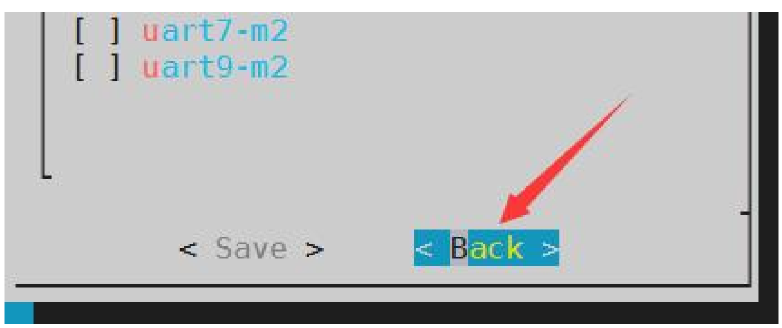


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select the I2C configuration you want to open

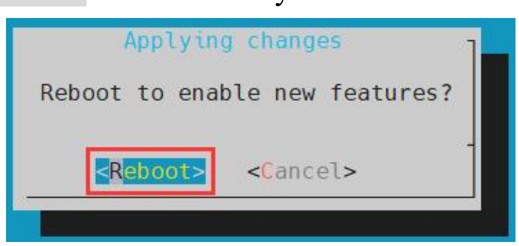
e. Then select **<Save>** to save



f. Then select **<Back>**



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



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

```
orangepi@orangepi:~# ls /dev/i2c-*
/dev/i2c-0    /dev/i2c-2    /dev/i2c-3    /dev/i2c-4    /dev/i2c-6
```

4) Then connect a i2c device on 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
```

```
root@orangepi3b:~# i2cdetect -y 2
     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: -- -- -- -- -- -- -- --
```

3. 16. 5. 40pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 3B 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.

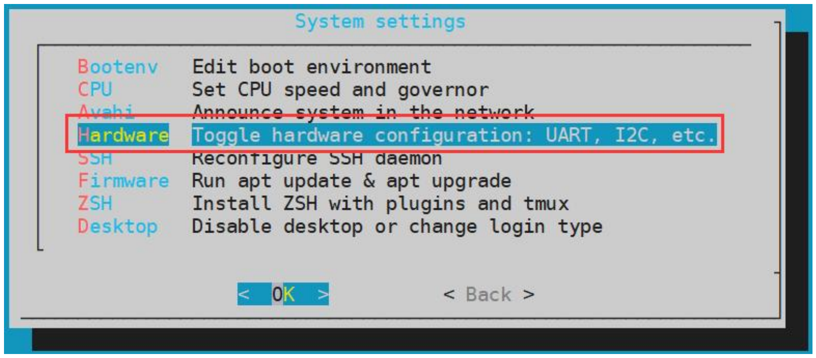
2) In the linux system, the UART in the 40 pins is closed by default, and it needs to be opened manually before it can be used. The detailed steps are as follows:

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

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

b. Then select **System**

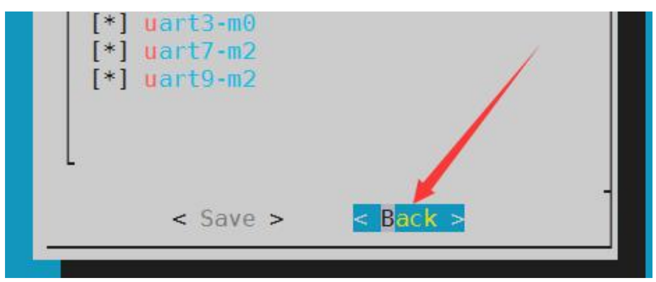
c. Then select **Hardware**



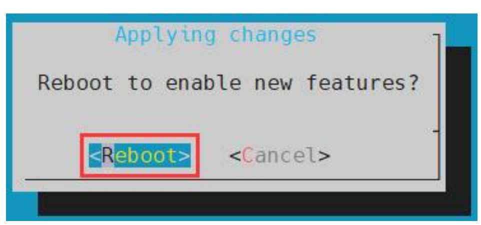
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select the UART configuration you want to open

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

f. Then select **<Back>**



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



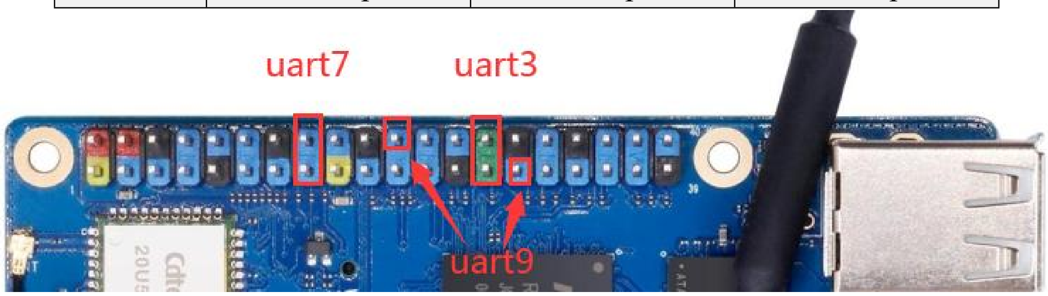
3) After entering the Linux system, first confirm whether there is a device node

corresponding to uart under/dev

```
orangepi@orangepi:~# ls /dev/ttyS*
/dev/ttyS1    /dev/ttyS3    /dev/ttyS7    /dev/ttyS9
```

4) tThen start to test the uart interface, first use the DuPont line to short-circuit the rx and tx of the uart interface to be tested

	uart3	uart7	uart9
Tx pin	Corresponding to the 28 pin	Corresponding to the 16 pin	Corresponding to the 29 pin
Rx Pin	Corresponding to the 27 pin	Corresponding to the 15 pin	Corresponding to the 22 pin



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 the 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 the 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 the password here.

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

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

1) As can be seen from the table below, Orange Pi 3B has two pwm channels available pwm11 and pwm15

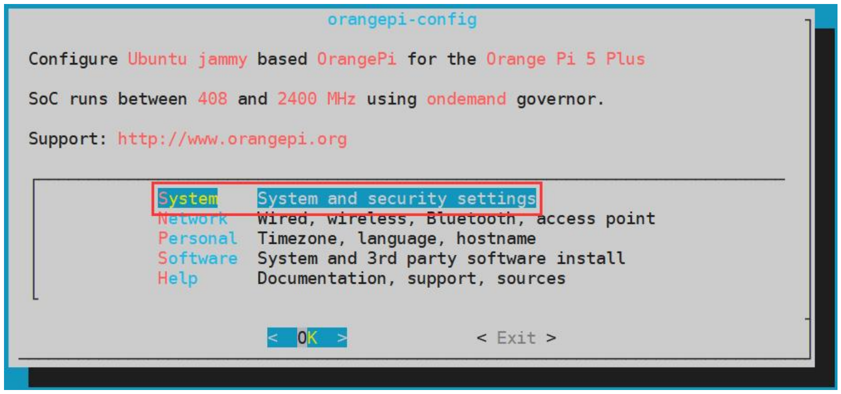
复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
		3.3V		1	2		5V		
	I2C2_SDA_M1	GPIO4_B4	140	3	4		5V		
	I2C2_SCL_M1	GPIO4_B5	141	5	6		GND		
	PWM15_IR_M1(fe700030)	GPIO4_C3	147	7	8	25	GPIO0_D1	UART2_TX_M0	
		GND		9	10	24	GPIO0_D0	UART2_RX_M0	
		GPIO3_C6	118	11	12	119	GPIO3_C7		
		GPIO4_A0	128	13	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:

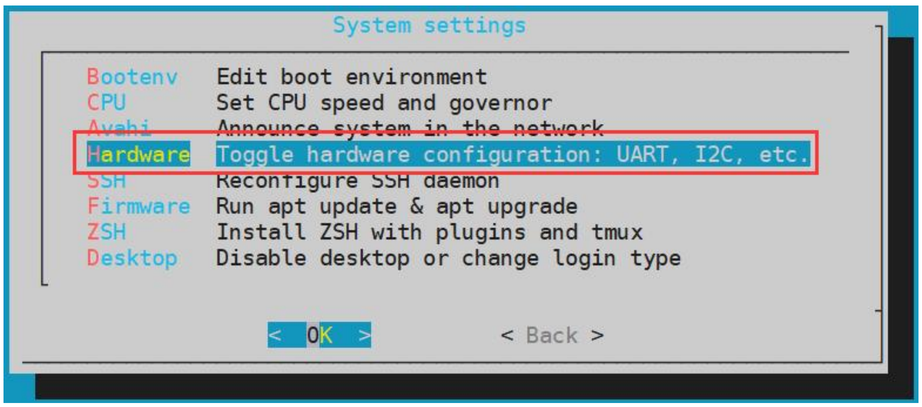
a. First run **orangepi-config**, ordinary users remember to add **sudo** permission

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

b. Then select **System**

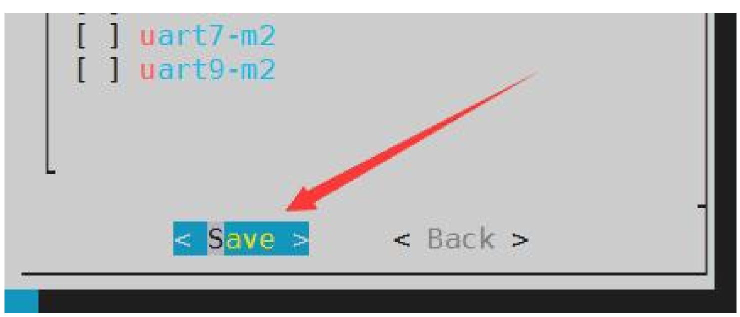


c. Then select **Hardware**

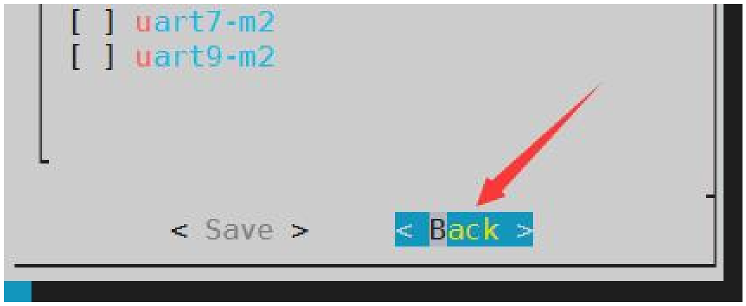


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select the PWM configuration you want to open

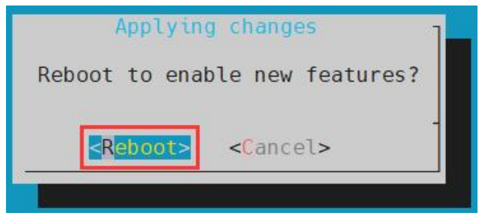
e. Then select **<Save>** to save



f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration 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 pwmchip above corresponds to pwm11, let's check the output of the **ls /sys/class/pwm/ -l** command first, as shown below:

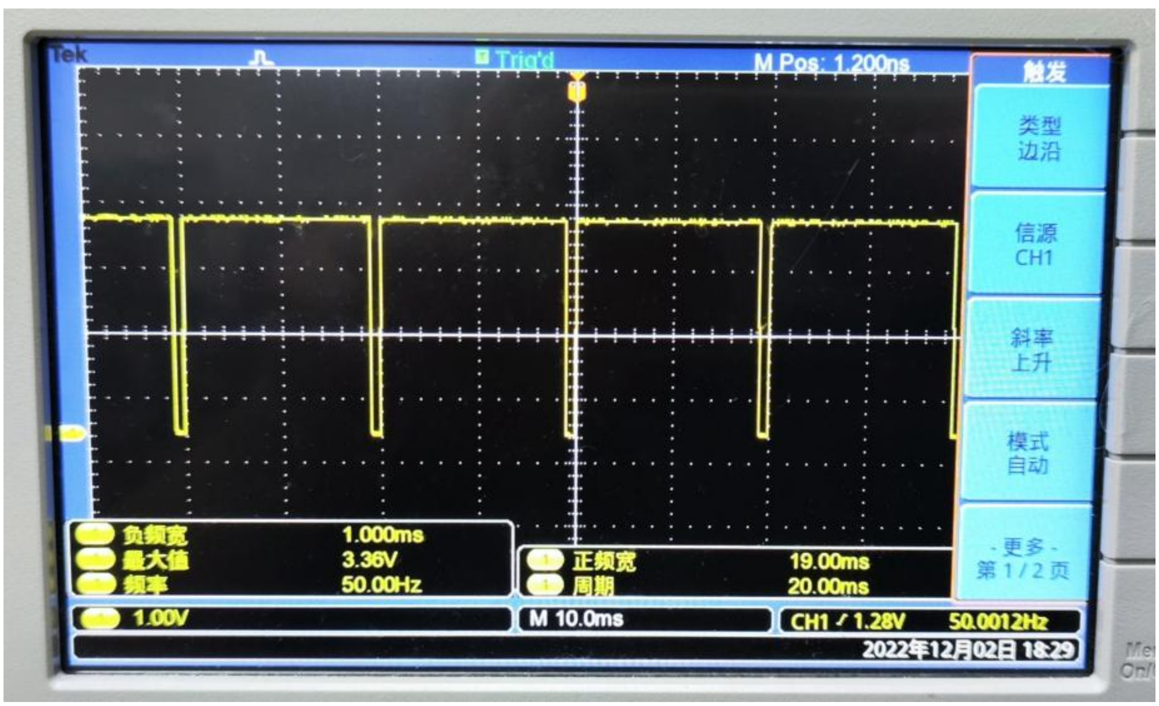
```
orangepi@orangepi3b:~$ ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Jan  1  1970 pwmchip0 -> ../../devices/platform/fe6e0030.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Jan  1  1970 pwmchip1 -> ../../devices/platform/fe6f0030.pwm/pwm/pwmchip1
```

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) Other pwm test methods are similar to the test method of pwm11 demonstrated above.

3. 17. How to use wiringOP hardware PWM

Before using the wiringOP hardware PWM function, please download the latest wiringOP source code and then compile and install it. The wiringOP preinstalled in the v1.0.2 version of the Linux image cannot use the hardware PWM function.

For how to download and install wiringOP, please refer to the instructions in the section How to Install wiringOP.

The development board can use a total of 2 PWM channels, PWM15_IR_M1 and PWM11_IR_M1. The locations of their pins are as shown in the figure below:

3. 17. 1. How to set PWM using wiringOP's gpio command

3. 17. 1. 1. Set the corresponding pin to PWM mode

1) The corresponding relationship between PWM pin and wPi serial number is as shown in the following table:

PWM Pin	wPi serial number
PWM15_IR_M1	2
PWM11_IR_M1	21

2) The command to set the pin to PWM mode is as follows, taking PWM15_IR_M1 as an example. The third parameter needs to enter the serial number of the wPi corresponding to the PWM15_IR_M1 pin.

```
orangepi@orangepi:~$ gpio mode 2 pwm
```

3) After the pin is set to PWM mode, it will output a square wave with a frequency of 200Hz and a duty cycle of 50% by default. At this time, we use an oscilloscope to measure the corresponding PWM pin and you can see the following waveform.

3. 17. 1. 2. Method of adjusting PWM duty cycle

1) The calculation formula of PWM duty cycle is as follows. We can adjust the PWM duty cycle by setting the values of CCR and ARR.

PWM duty cycle = CCR / ARR

In:

The value range of CCR is 0~65535, and the default value is 500.

The value range of ARR is 0~65535, and the default value is 1000.

It should be noted that our CCR value needs to be smaller than the ARR value because the duty cycle cannot be greater than 1.

When setting CCR > ARR, the following error message will be prompted:

gpio: CCR should be less than or equal to ARR (XXX)

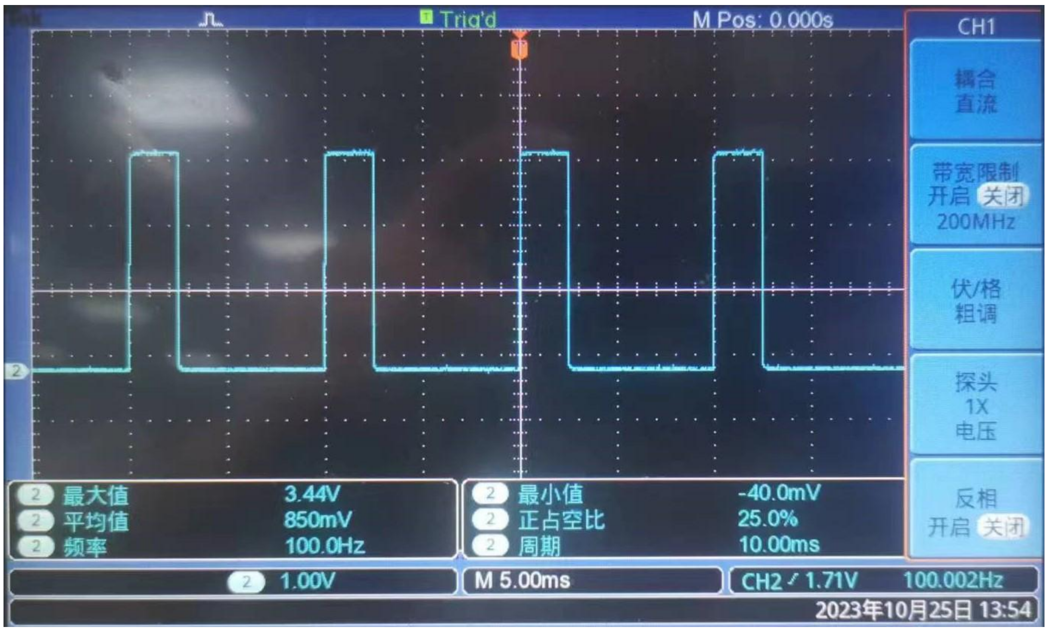
When setting ARR < CCR, the following error message will be prompted:

gpio: ARR should be greater than or equal to CCR (XXX)

2) We can use the following command to set the ARR of the PWM15_IR_M1 pin to 2000

```
orangepi@orangepi:~$ gpio pwmr 2 2000
```

3) After running the above command, you can observe through the oscilloscope that the PWM duty cycle changes from the default 50% (500/1000) to 25% (500/2000)



4) We can use the following command to set the CCR of the PWM15_IR_M1 pin to 1000

```
orangepi@orangepi:~$ gpio pwm 2 1000
```

5) After running the above command, you can observe through the oscilloscope that the PWM duty cycle changes from 25% (500/2000) to 50% (1000/2000)

3. 17. 1. 3. Method of adjusting PWM frequency

The calculation formula of PWM frequency is as follows:

PWM frequency = clock source frequency / (frequency division factor * ARR)

In:

The default value of the clock source frequency is 24000000Hz.

The value range of the frequency division coefficient is 2 ~ 512, and the default value is 120.

The value range of ARR is 0 ~ 65535, and the default value is 1000.

The default value of PWM frequency is 24000000 / (120 * 1000) = 200Hz.

It should be noted that if the frequency division coefficient is set to an odd number, the actual frequency division coefficient is the set value minus one.

3. 17. 1. 3. 1. Method to adjust PWM frequency by setting frequency division coefficient

1) We can use the following command to set the frequency division coefficient of the PWM15_IR_M1 pin to 4

```
orangepi@orangepi:~$ gpio pwmc 2 4
```

2) According to the above formula, the calculated value of PWM frequency is 6000Hz. It can be observed through the oscilloscope that the measured value of PWM frequency is 6010Hz, and the error can be ignored.

3. 17. 1. 3. 2. Method of directly setting the PWM frequency

1) We can use the **gpio pwmTone** command to set the frequency of the PWM pin. For example, use the following command to set the PWM frequency of the PWM15_IR_M1 pin to 500Hz.

```
orangepi@orangepi:~$ gpio pwmTone 2 500
```

When setting the PWM frequency, you need to ensure:

The set frequency value is < 24000000 / (frequency division factor * 2).

For example, the default frequency division coefficient is 120. If the frequency division coefficient is not modified, the set frequency value should be less than 100000Hz.

If the setting value is too large, the following error will appear:

```
gpio: The PWM frequency you set is too high to be possible
```

2) Then you can observe through the oscilloscope that the PWM frequency becomes 500Hz

3. 17. 2. How to use the PWM test program

1) In the example directory of wiringOP, there is a program named pwm.c. This program demonstrates how to operate PWM using the PWM-related API in wiringOP.

```
orangepi@orangepi:~$ cd wiringOP/examples/
orangepi@orangepi:~/wiringOP/examples$ ls pwm.c
pwm.c
```

2) The command to compile **pwm.c** into an executable program is as follows

```
orangepi@orangepi:~/wiringOP/examples$ gcc -o pwm pwm.c -lwiringPi
```

3) Then you can execute the PWM test program. When executing the PWM test program, you need to specify the PWM pin. For example, you can use the following command to test the PWM15_IR_M1 pin:

```
orangepi@orangepi:~/wiringOP/examples$ sudo ./pwm 2
```

4) After the pwm program is executed, the following contents will be tested in sequence:

a. Adjust the PWM duty cycle by setting ARR

b. Adjust the PWM duty cycle by setting CCR
c. Adjust the PWM frequency by setting the frequency division coefficient
d. Directly set the PWM frequency

5) After each test is completed, the current pwm waveform will be maintained for 5 seconds. After all test contents are completed, a new round of testing will be restarted.

6) The detailed execution process of the PWM test program is as follows:

a. Adjust the PWM duty cycle by setting ARR: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 50% to 25%, maintains it for 5 seconds, and then PWM The waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 25% to 50% and remains for 5 seconds.
b. Adjust the PWM duty cycle by setting CCR: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 50% to 100%, maintains it for 5 seconds, and then PWM The waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 100% to 50% and remains for 5 seconds.
c. Adjust the PWM frequency by setting the frequency division coefficient: It can be observed through the oscilloscope that the PWM frequency first changes to 12000Hz, and then the PWM waveform changes every 0.5 seconds. After changing 9 times, the PWM frequency changes to 1200Hz and remains for 5 seconds. Then the PWM waveform changes every 0.5 seconds. After changing 9 times, the PWM frequency changes to 12000Hz and remains for 5 seconds.
d. Directly set the PWM frequency: It can be observed through the oscilloscope that the PWM frequency first changes to 2000Hz, and then the PWM frequency increases by 2000Hz every two seconds. After changing 9 times, the PWM frequency changes to 20000Hz and remains for 5 seconds.

3. 18. 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. 18. 1. wiringOP-Python installation method

1) First install the dependency package

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python

Note that the following git clone--recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on wiringOP. Please make sure that the download process does not report an error due to network problems。

If there is a problem with the download code from GitHub, you can use the wiringOP-Python source code that comes with the Linux image directly, 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 a help information output, it means that Wiringop-Python is successfully installed. Press the **q** key to exit the interface of the help

information

```
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) Test whether the wiringOP-Python is installed successfully under the Python command line is shown below：

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. Enter the following command to view the help information of wiringOP-Python, and press the **q** key to exit the interface of the help information

```
>>> 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. 18. 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@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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 |     IN | 1 | 27 || 28 | 1 | IN     | 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 |
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
```

1) The following takes pin 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@orangepi3b:~# gpio readall
 +------+-----+----------+--------+---+   PI3B   +---+--------+----------+-----+------+
 | 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 command line of python3 are as follows:

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

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

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

```
>>> wiringpi.digitalWrite(2, GPIO.LOW)
```

e. Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

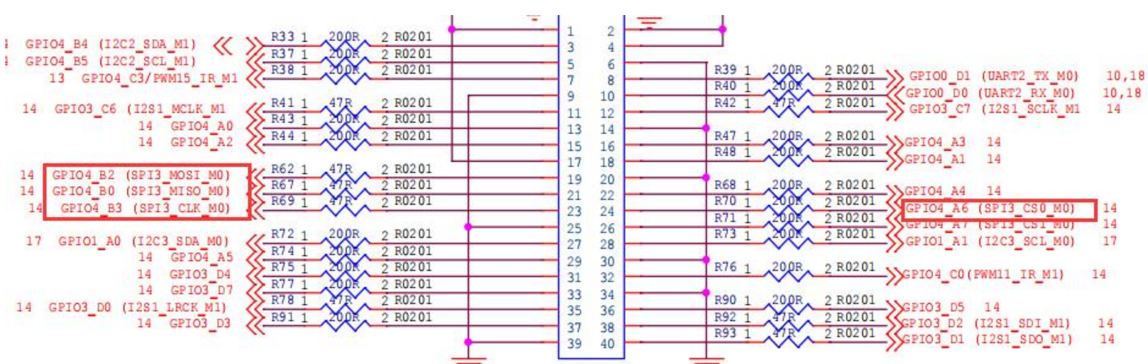
```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

4) The method of wiringOP-Python setting GPIO high and low levels in python code can refer to the **blink.py** test program in the examples below. The **blink.py** test program will set the voltage of all GPIO ports in the 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. 18. 3. 40pin SPI test

1) From the schematic diagram of the 40pin interface, the SPI available for Orange Pi 3B is spi3

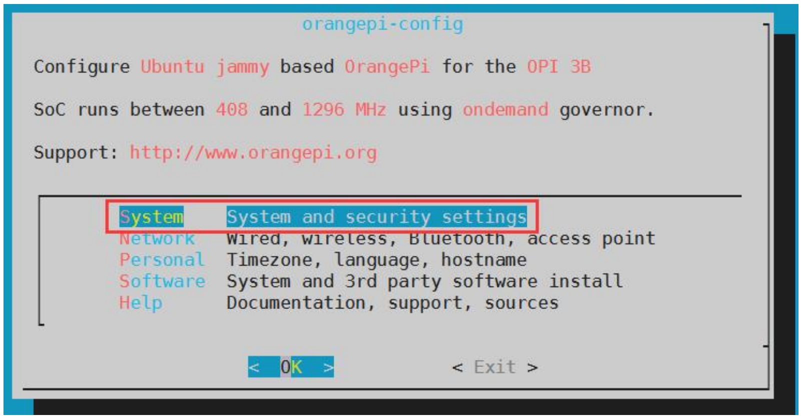


2) In the linux system, the SPI in the 40 pin is closed by default, and it needs to be opened manually before it can be used. The detailed steps are as follows:

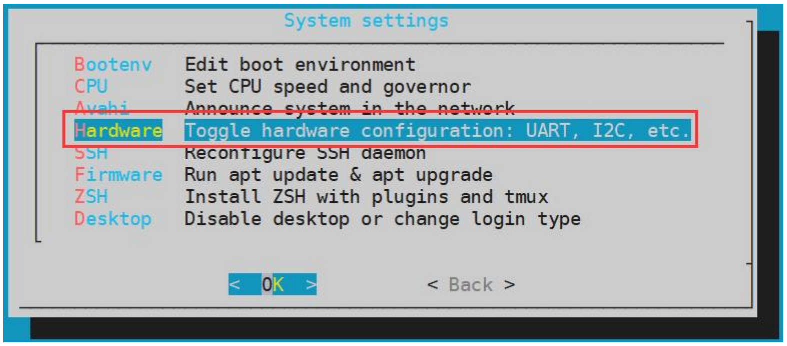
a. First run **orangepi-config**, ordinary users remember to add **sudo** permission

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

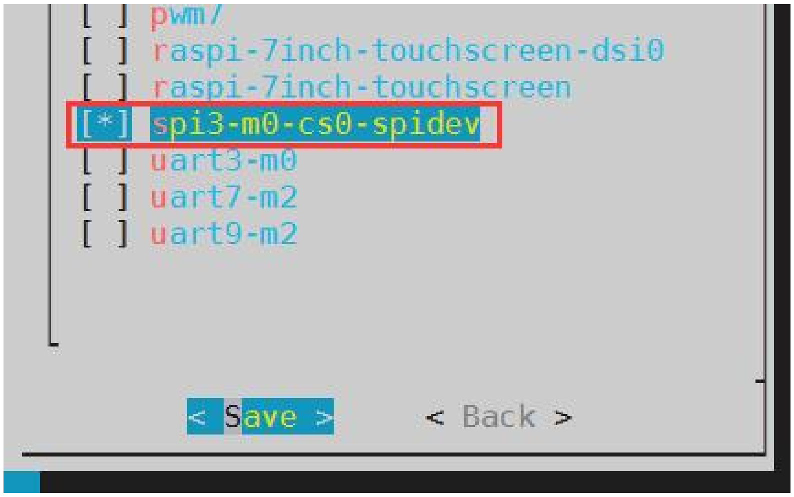
b. Then select **System**



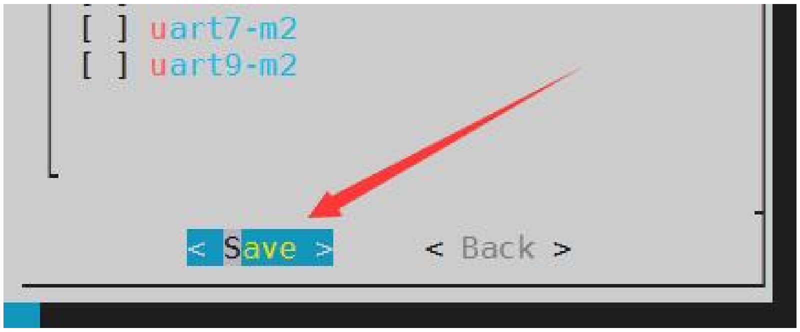
c. Then select **Hardware**



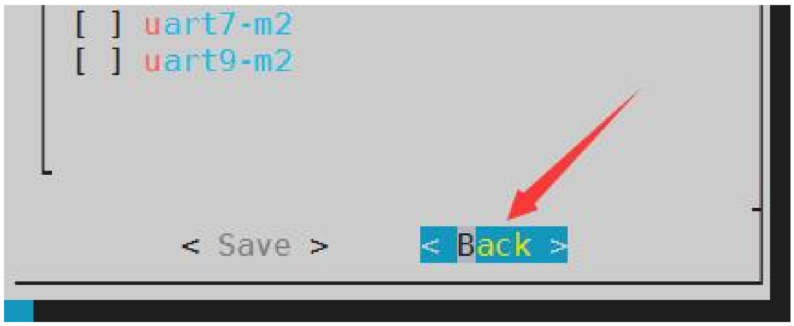
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select the SPI configuration you want to open



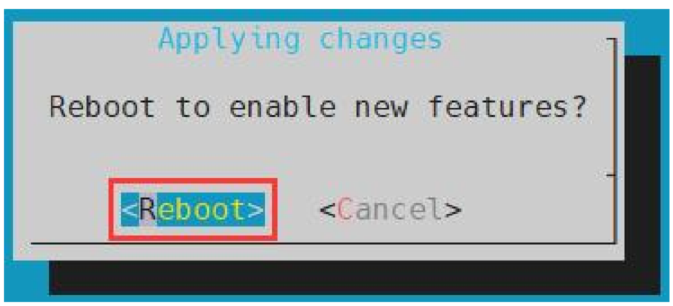
e. Then select **<Save>** to save



f. Then select **<Back>**



g. Then select **<Reboot>** to restart the system to make the configuration 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 the 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 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. 18. 4. 4.40pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi 3B 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

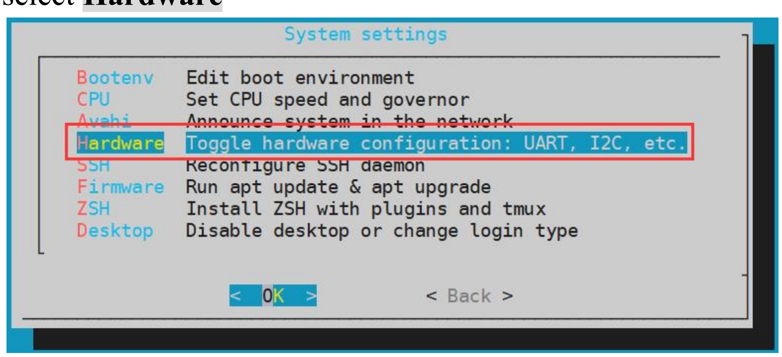
2) In the linux system, the I2C bus in the 40 pin is closed by default, and it needs to be opened manually to use it. The detailed steps are as follows:

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

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

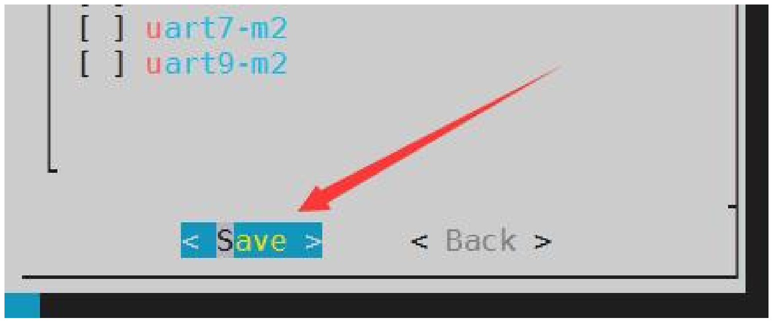
b. Then select **System**

c. Then select **Hardware**

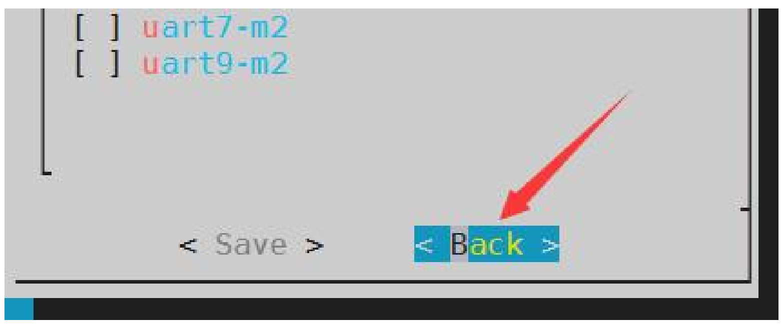


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select the I2C configuration you want to open

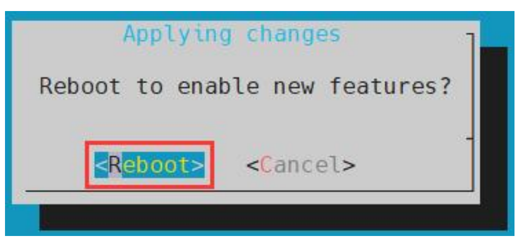
e. Then select **<Save>** to save



f. Then select **<Back>**



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

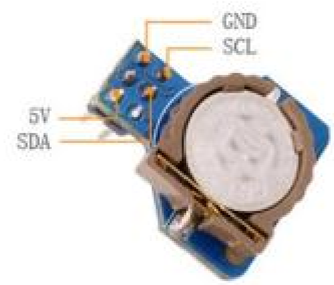


3) After starting the Linux system, first confirm that there is an i2c device node under /dev

```
orangepi@orangepi:~# ls /dev/i2c-*
/dev/i2c-0    /dev/i2c-2    /dev/i2c-3    /dev/i2c-4    /dev/i2c-6
```

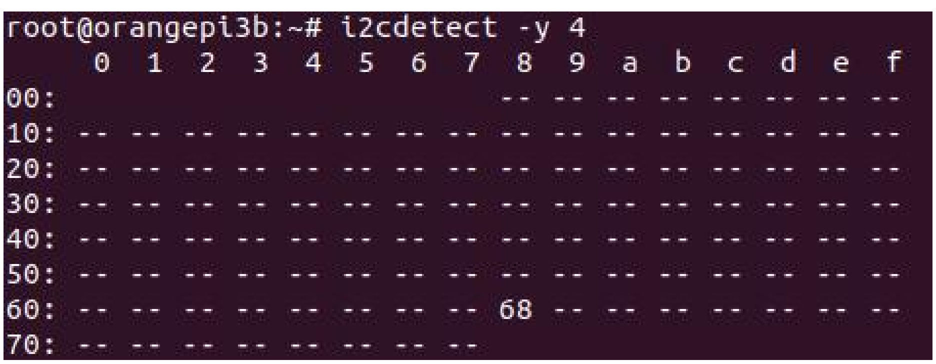
4) Then connect a I2C device on the I2C pin of the 40PIN connector. Here is an example of the DS1307 RTC module.

	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 the 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@orangepi3b:~# i2cdetect -y 4
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

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. 18. 5. 40pin's UART test

1) As can be seen from the table below, the available uarts for Orange Pi 3B 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.

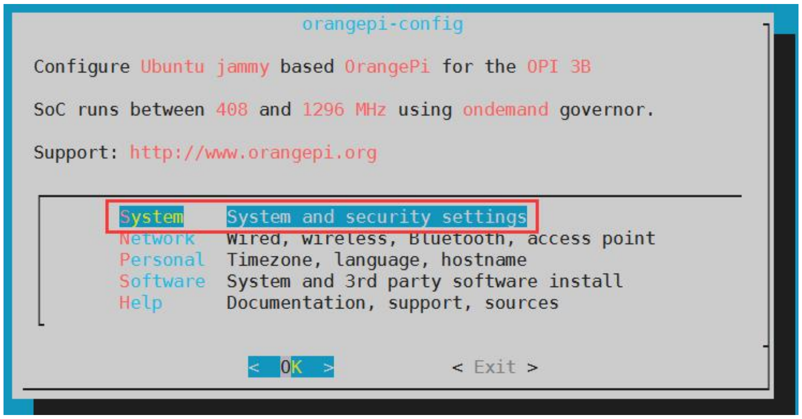
Please note that uart2_m0 is used as the debugging serial port of the system by default. If the configuration of uart2 is turned on, the debugging serial port function will not be available.

2) In the linux system, the UART in the 40 pins is closed by default, and it needs to be opened manually before it can be used. The detailed steps are as follows:

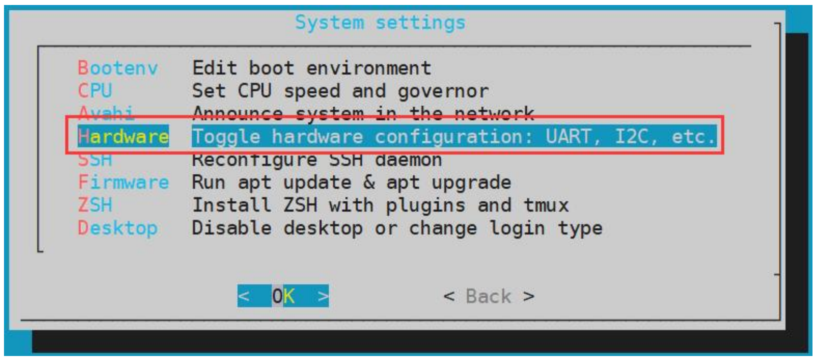
a. First run **orangepi-config**, ordinary users remember to add **sudo** permission

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

b. Then select **System**

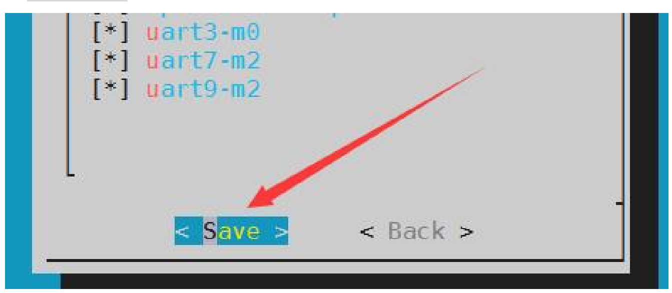


c. Then select **Hardware**

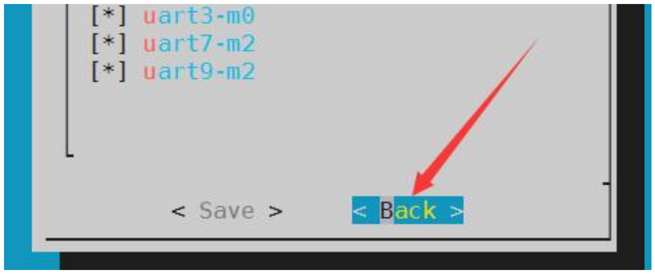


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select the UART configuration you want to open

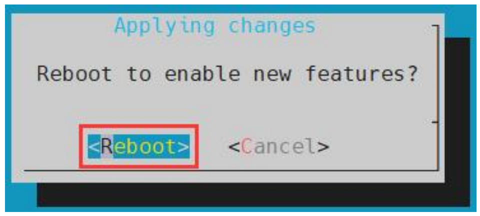
e. Then select **<Save>**to save



f. Then select **<Back>**



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



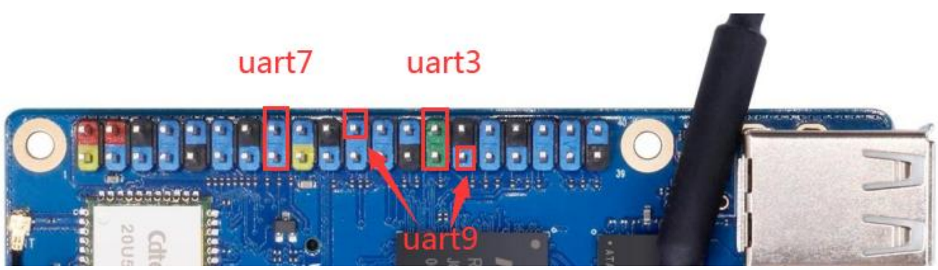
3) After entering the Linux system, first confirm whether there is a device node corresponding to uart under /dev

```
orangepi@orangepi:~# ls /dev/ttyS*
/dev/ttyS1    /dev/ttyS3    /dev/ttyS7    /dev/ttyS9
```

Note that /dev/ttyS0 in linux6.6 is uart9.

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 the 28 pin	Corresponding to the 16 pin	Corresponding to the 29 pin
rx Pin	Corresponding to the 27 pin	Corresponding to the 15 pin	Corresponding to the 22 pin



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. 19. Hardware watch the door dog test

Note that linux6.6 is not supported yet.

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
```

3.20. Check the serial number of the 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
```

3.21. The method of downloading and installing the balenaEtcher version of arm64

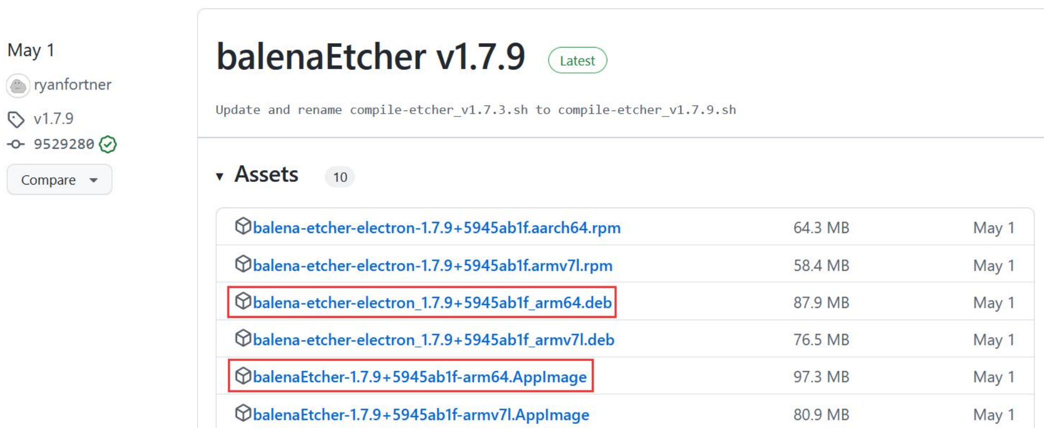
1) The download address of Balenaetcher ARM64 version is：

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

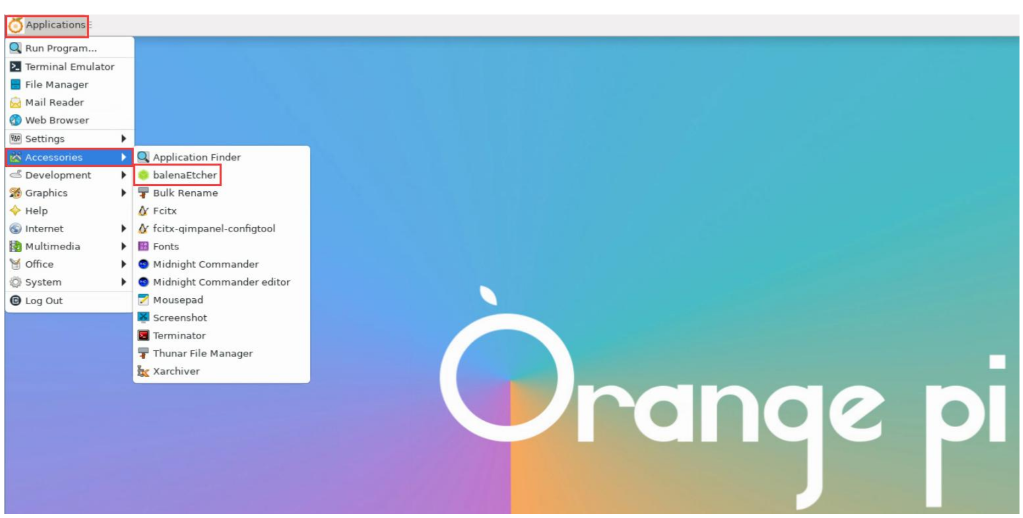


2) How to install and use the deb version of Balenaetcher：

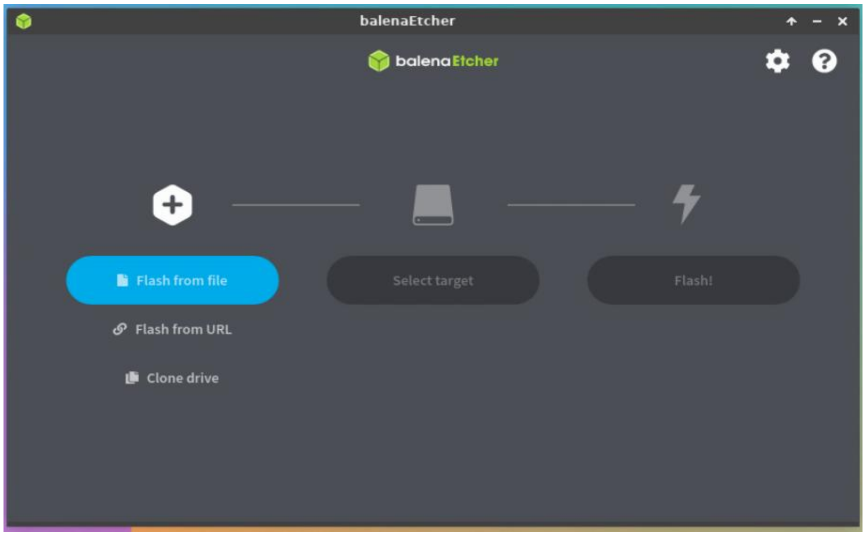
a. The deb version of balenaEtcher installation command is as follows:

```
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:

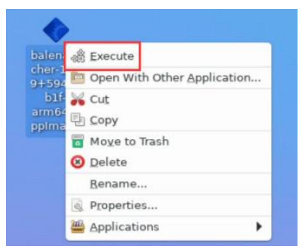


3) How to use the AppImage version of balenaEtcher：

a. First add permissions to Balenaetcher

```
orangepi@orangepi:~/Desktop$ chmod +x balenaEtcher-1.7.9+5945ab1f-arm64.AppImage
```

b. Then select the AppImage version balenaEtcher right-click the mouse, and then click Execute to open balenaEtcher



3. 22. The installation method of the Bt-Panel Linux panel

Bt-Panel Linux panel is a server management software that improves operation and maintenance efficiency, and supports one-click configuration of more than 100 server management functions such as LAMP/LNMP/cluster/monitoring/website/FTP/database/JAVA (excerpted from the official website of the Bt-Panel)

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

```
orangepi@orangepi:~$ sudo sed -i 's/nosuid/&,size=2G/' /etc/fstab
```

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

2) After restarting, you can see that the size of the **/tmp** space has become 2G.

```
orangepi@orangepi:~$ df -h | grep "/tmp"
tmpfs            2.0G   12K   2.0G    1% /tmp
```

3) Then enter the following command in the Linux system to start the installation of the Bt-Panel

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

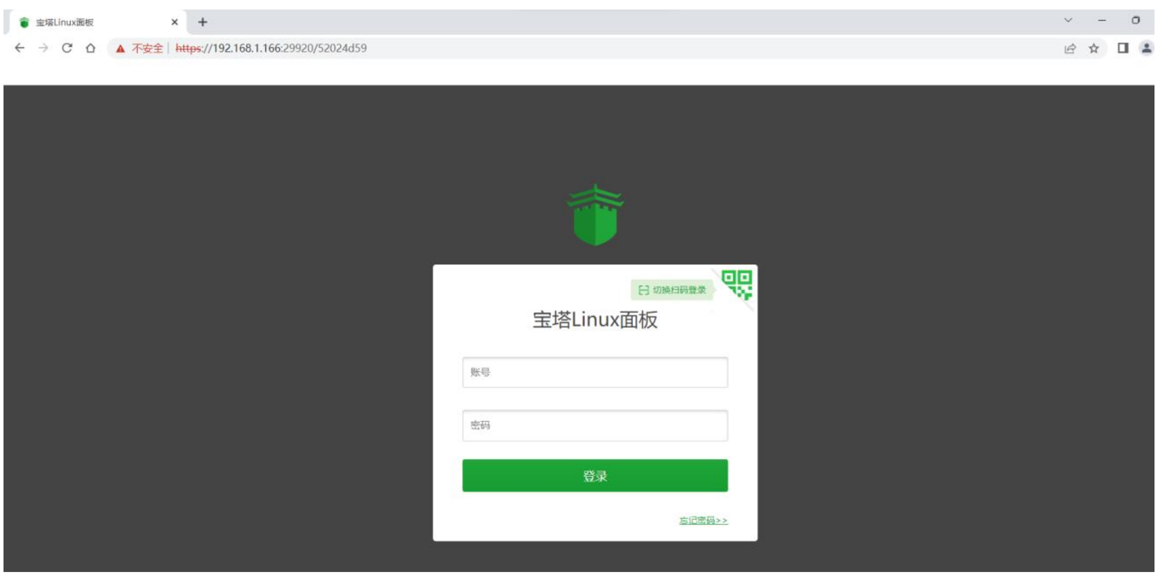
4) Then the Bt-Panel installation program reminds whether to install the **Bt-Panel** to the **/www** folder, and enter y at this time

```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| The WebPanel URL will be http://SERVER_IP:8888 when installed.
+----------------------------------------------------------------------

Do you want to install Bt-Panel to the /www directory now?(y/n): y
```

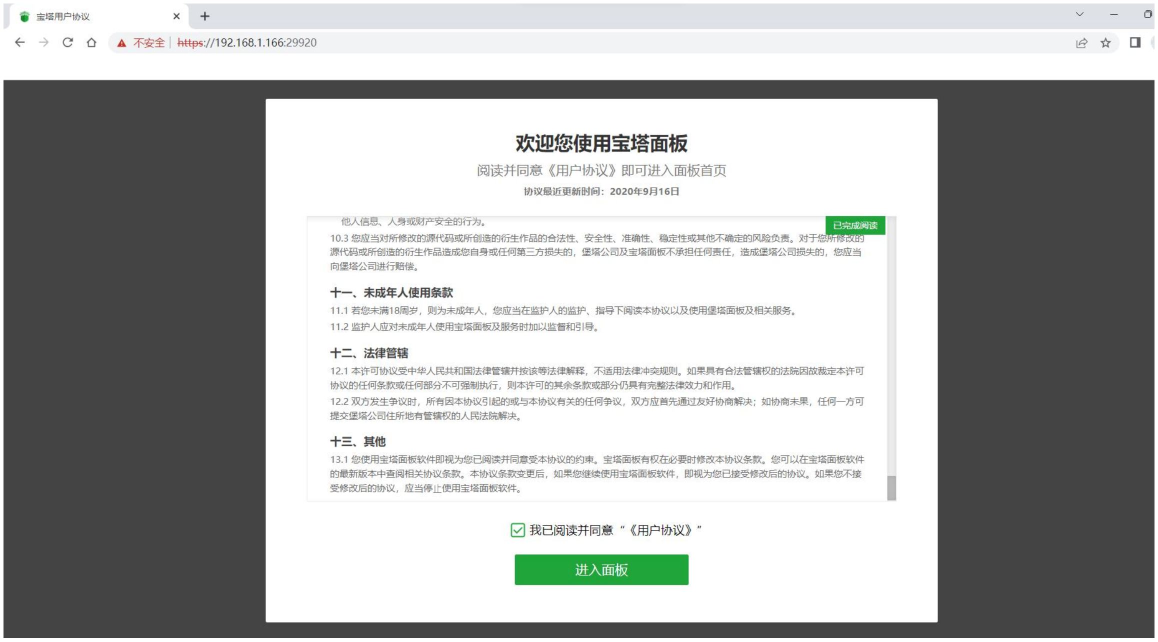
5) Then you have to wait patiently. When you see the printing information below the terminal output, it means that the Bt-Panel has been installed. The entire installation process takes about 34 minutes. There may be some differences according to the difference in network speed

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

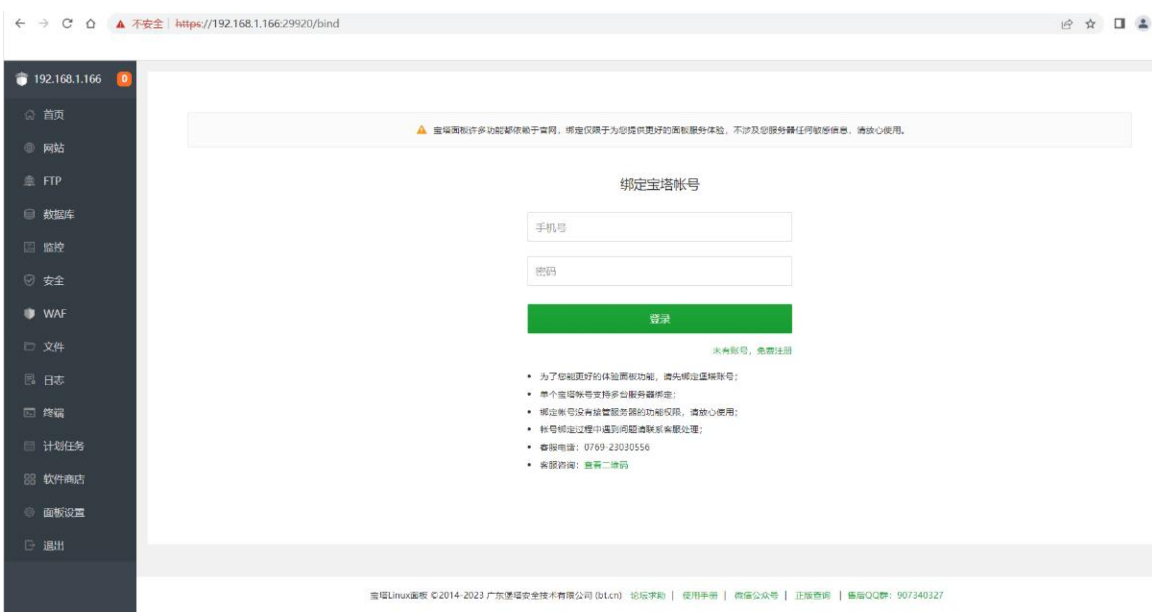
6) At this time, enter the **panel address** displayed above in the browser to open the login interface of the Bt-Panel Linux panel, and then enter the **username** and **password** displayed in the corresponding position to log in to the Bt-Panel



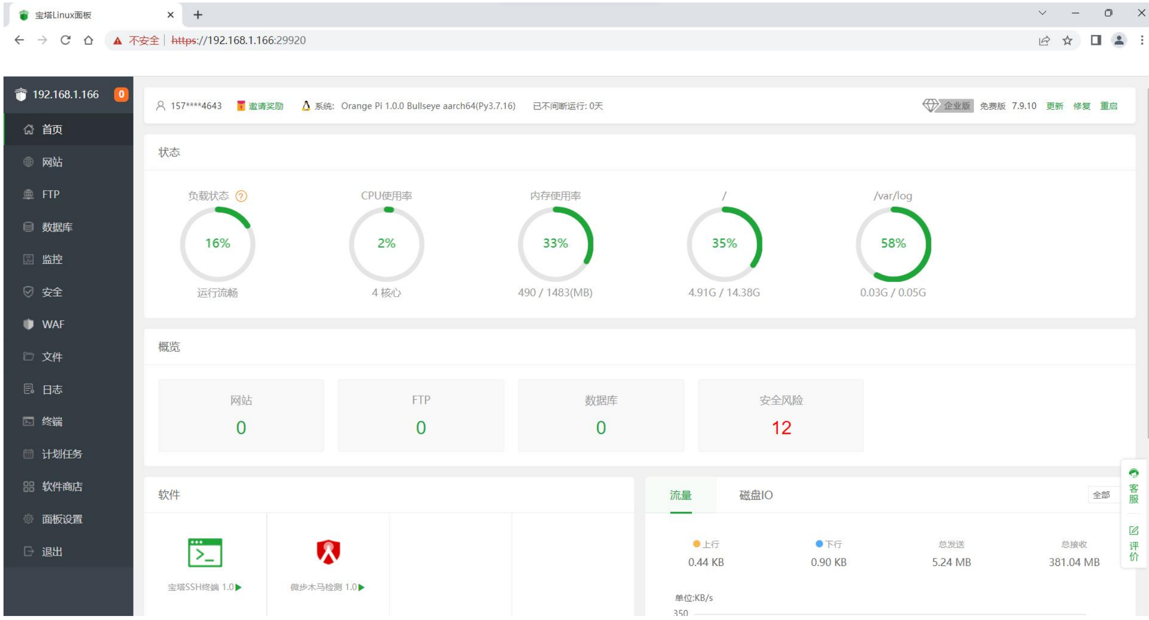
7) After successfully logging into the Bt-Panel, the following welcome interface will pop up. First, please read the user notice in the middle and drag it to the bottom, then you can select "I have agreed and read the "User Agreement"", and then click "Enter the Panel" You can enter the Bt-Panel



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



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



10) More functions of the Bt-Panel 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. Set the Chinese environment and install Chinese input method

Note that before installing the Chinese input method, please make sure that the Linux system used in the development board is the desktop version system。

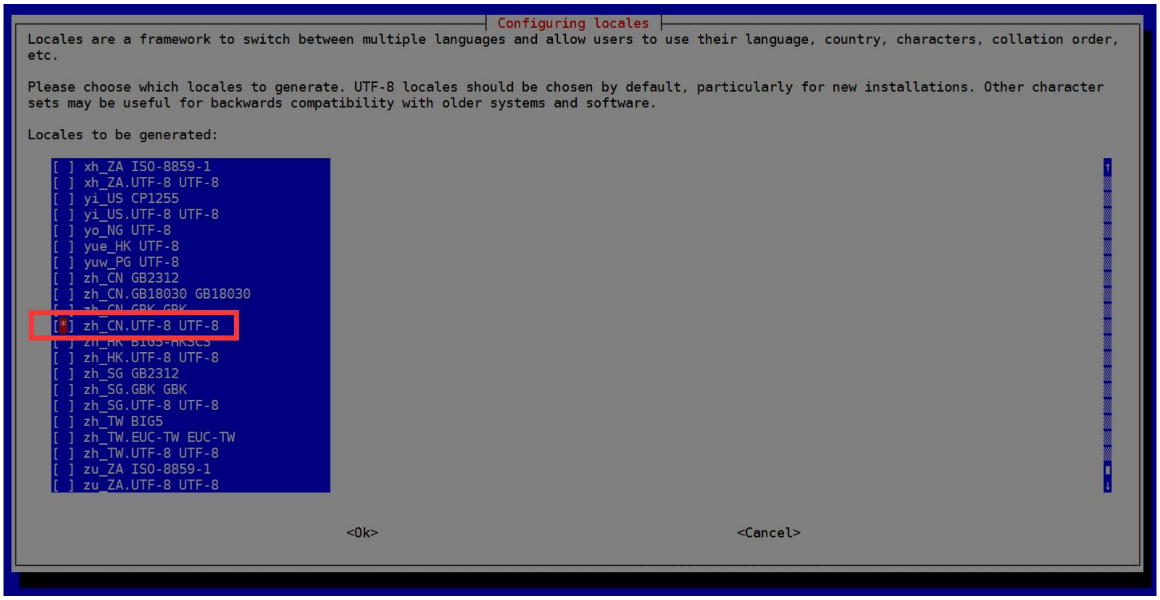
3. 23. 1. Debian system installation method

1) First set the default **locale** as Chinese

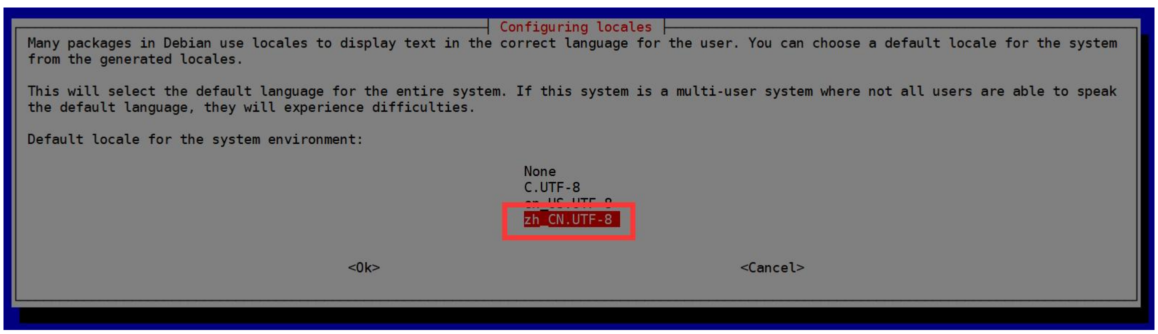
a. Enter the command below to start configured **locale**

orangepi@orangepi:~$ **sudo dpkg-reconfigure locales**

b. Then select **zh_CN.UTF-8 UTF-8** in the pop-up interface (to move up and down through the upper and lower direction buttons on the keyboard, select it through the space key, and finally move the cursor to **<OK>** through the TAB key, then press the ENTER key)



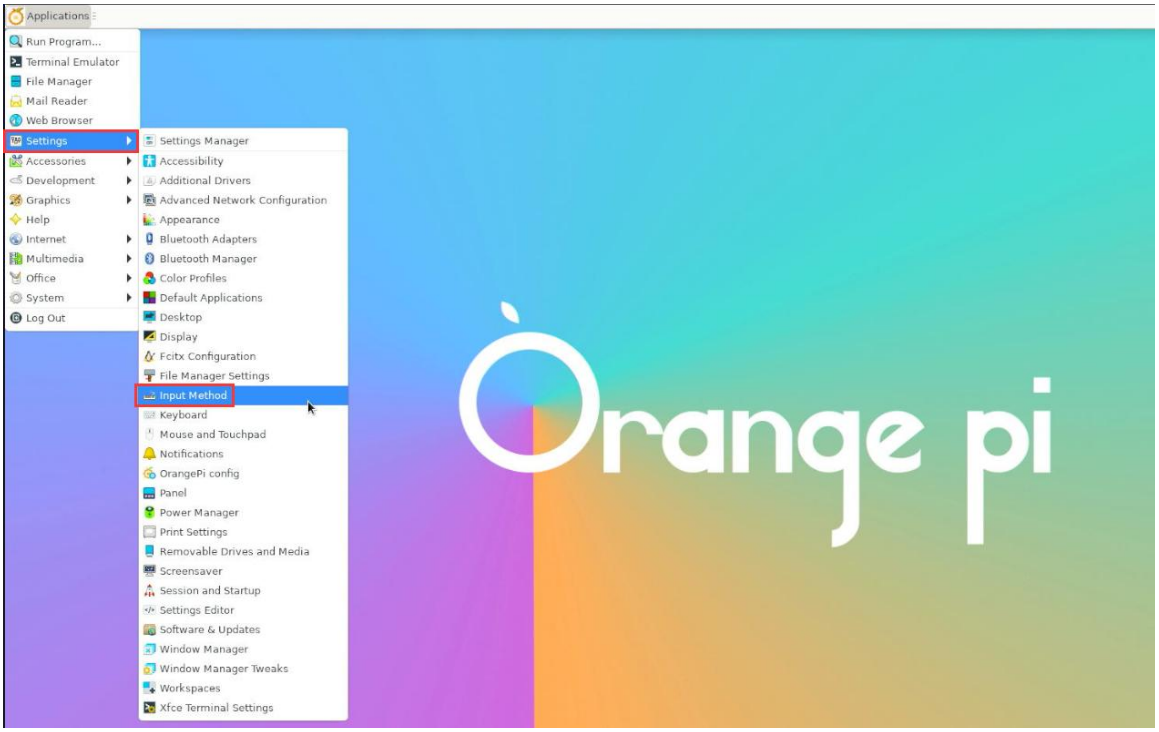
c. Then set the default **locale** as **zh_CN.UTF-8**



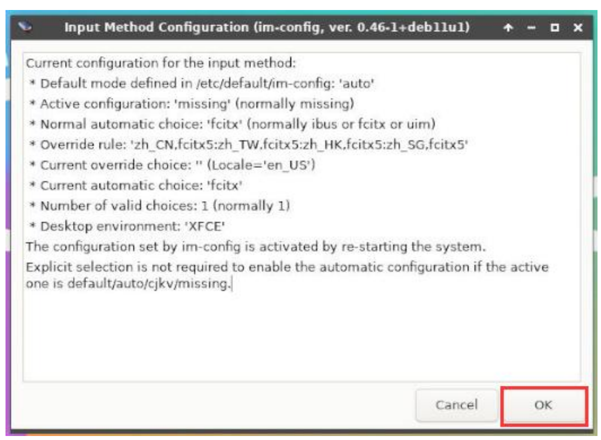
d. After exiting the interface, the **locale** settings will be started. The output displayed by the command line is shown below

```
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.
```

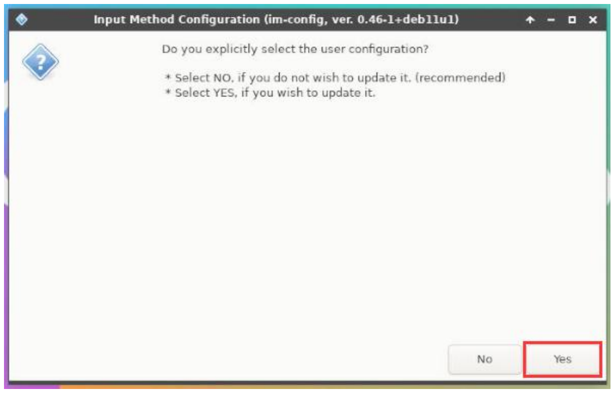
2) Then open **Input Method**



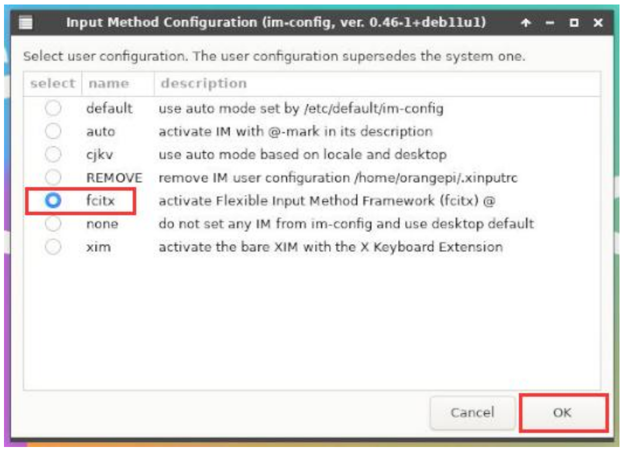
3) Then select **OK**



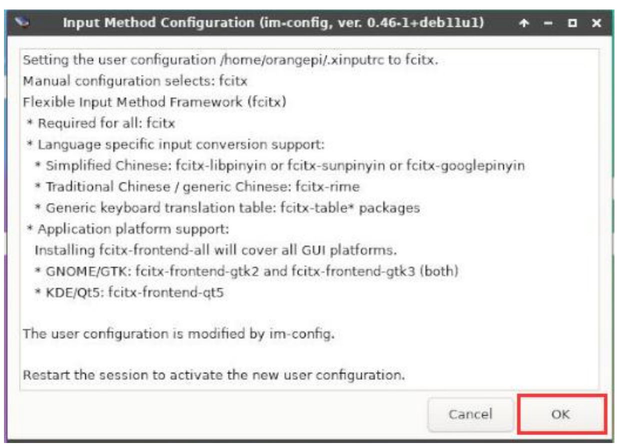
4) Then select **Yes**



5) Then select **fcitx**

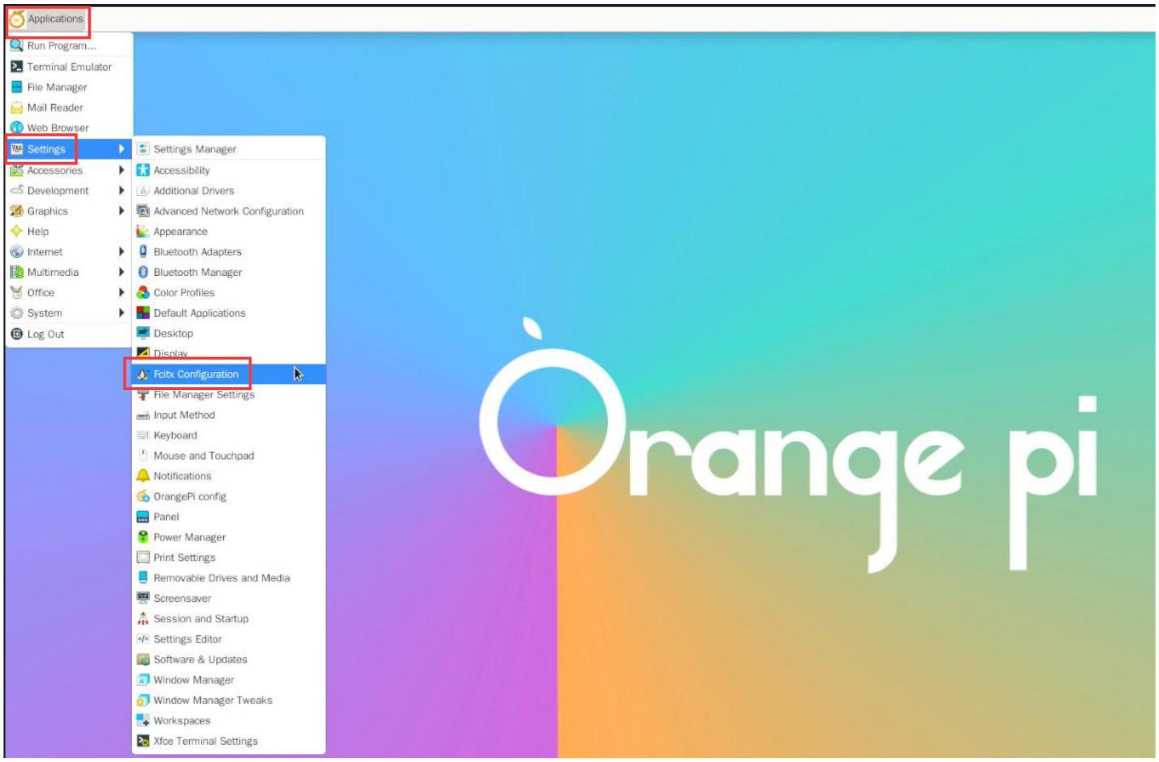


6) Then select **OK**

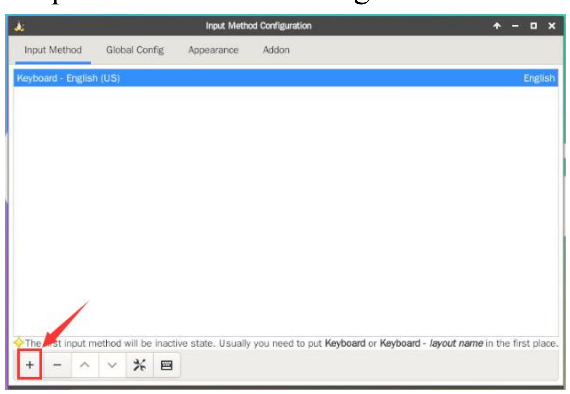


7) **Then restart the Linux system to make the configuration effective**

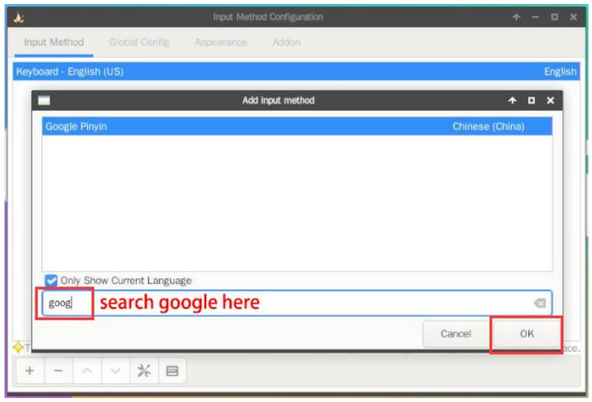
8) Then open **Fcitx configuration**



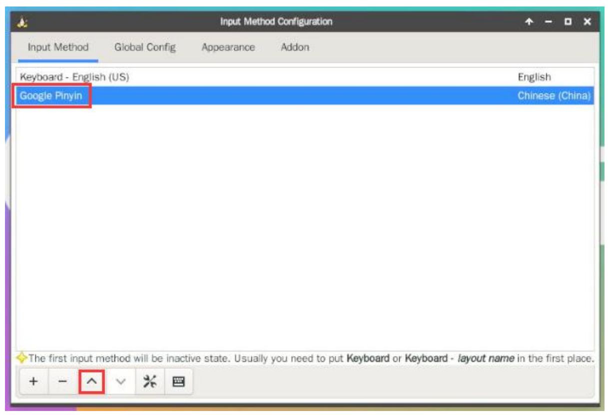
9) Then click the + of the position shown in the figure below

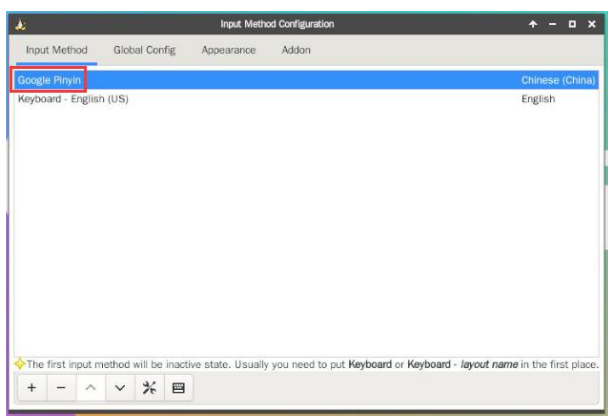


10) Then search **Google Pinyin** and click **OK**



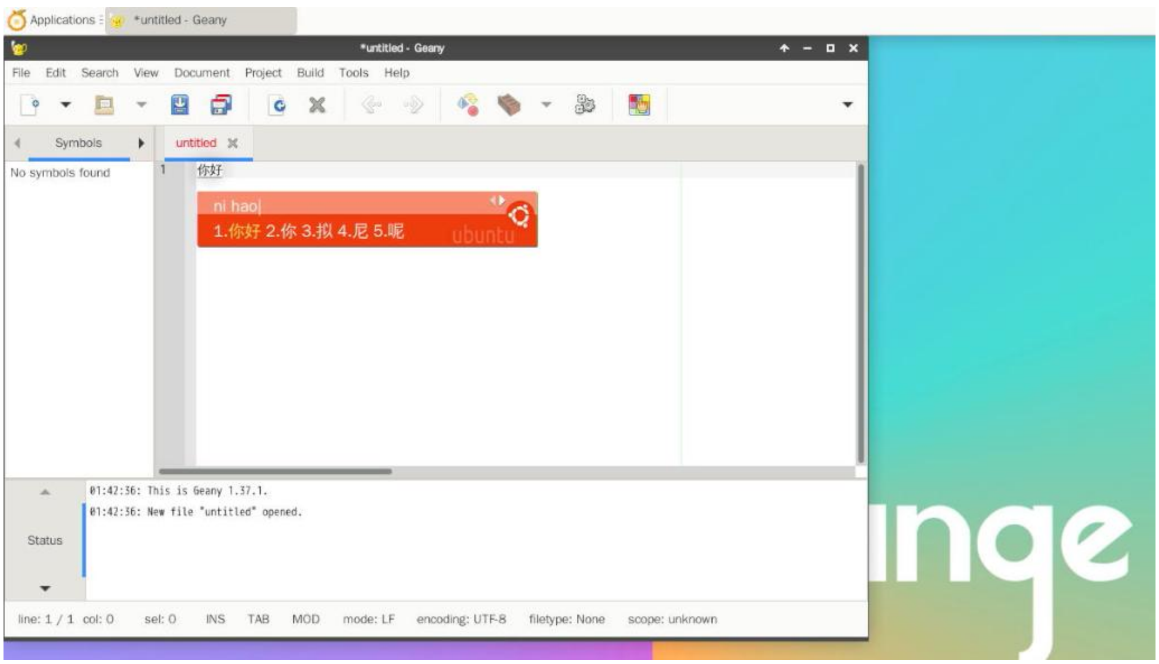
11) Then put **Google Pinyin** to the front





12) Then open the **Geany** editor to test the Chinese input method

13) The Chinese input method test is shown below



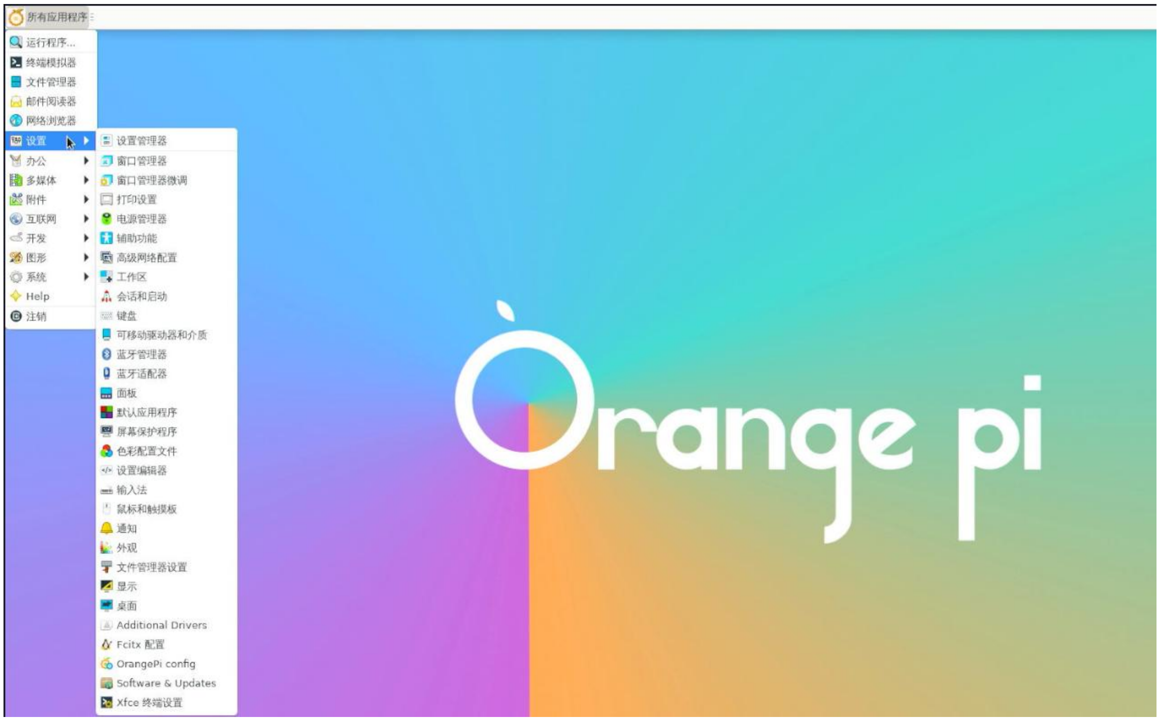
14) You can switch the Chinese and English input method through the **Ctrl+Space** shortcut

15) If the entire system is required as Chinese, the variables in **/etc/default/locale** can be set 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
```

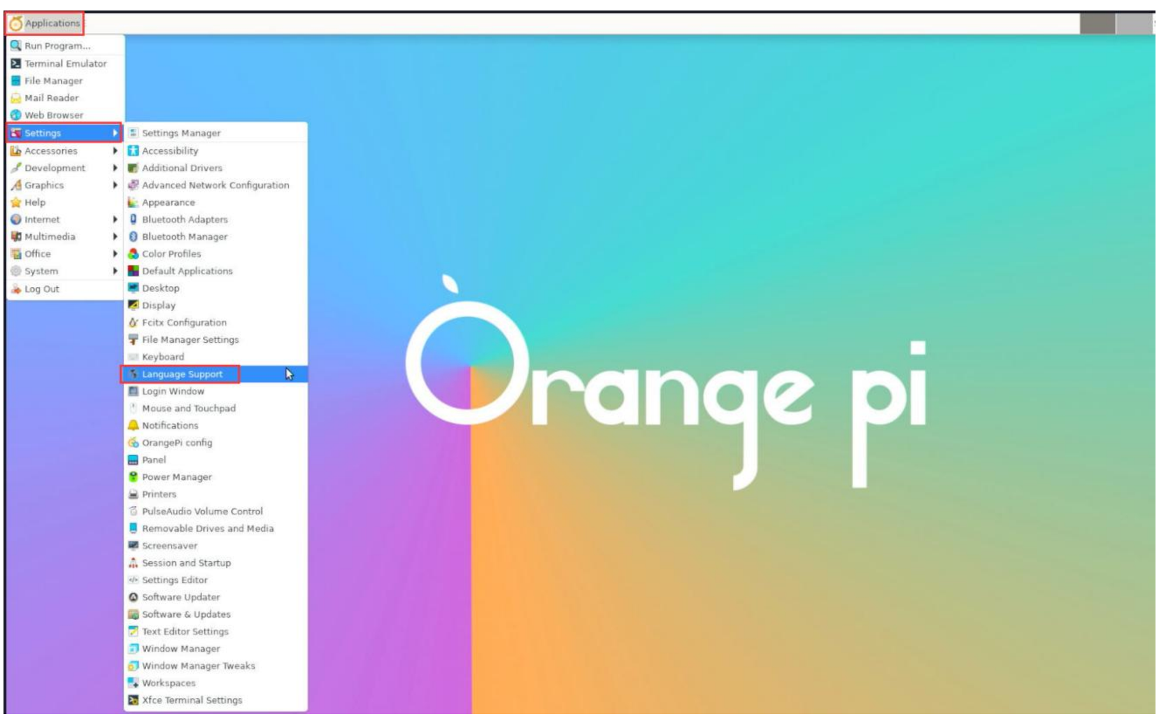
```
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** to see the system displayed as Chinese

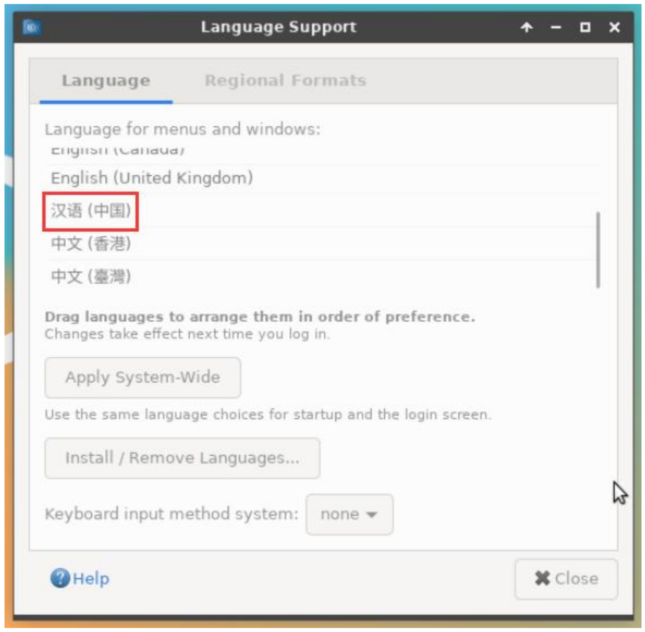


3. 23. 2. The installation method of Ubuntu 20.04 system

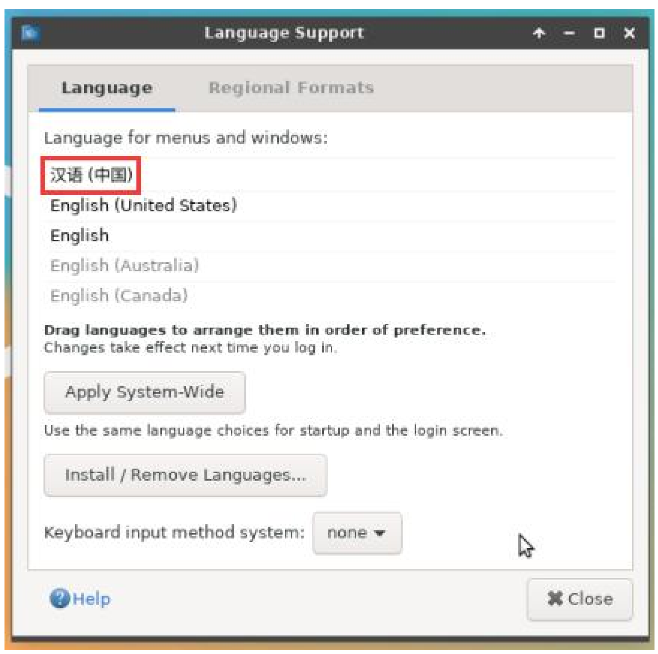
1) First open **Language Support**



2) Then find **Chinese (China)** option

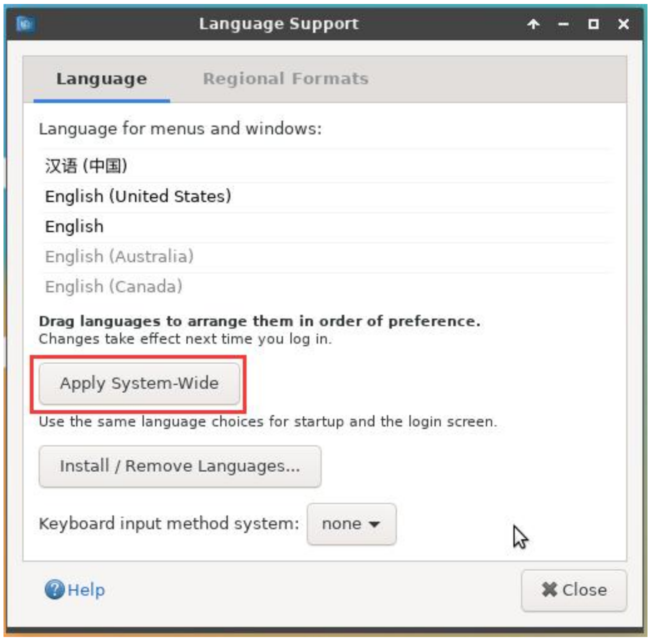


3) Then use the left mouse button to select **Chinese (China)** and hold it down, and then drag it up to the beginning. The display is shown below：

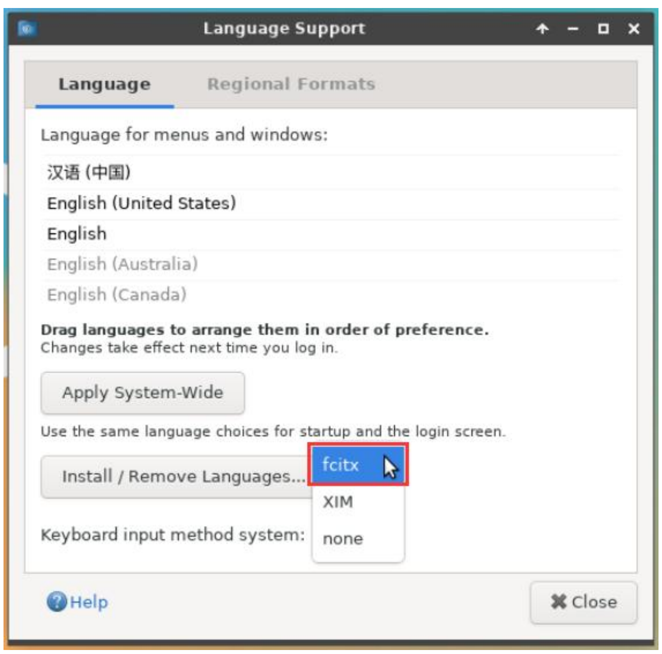


Note that this step is not easy to drag, please try more patiently。

4) Then select the **Apply System-Wide** to apply the Chinese settings to the entire system



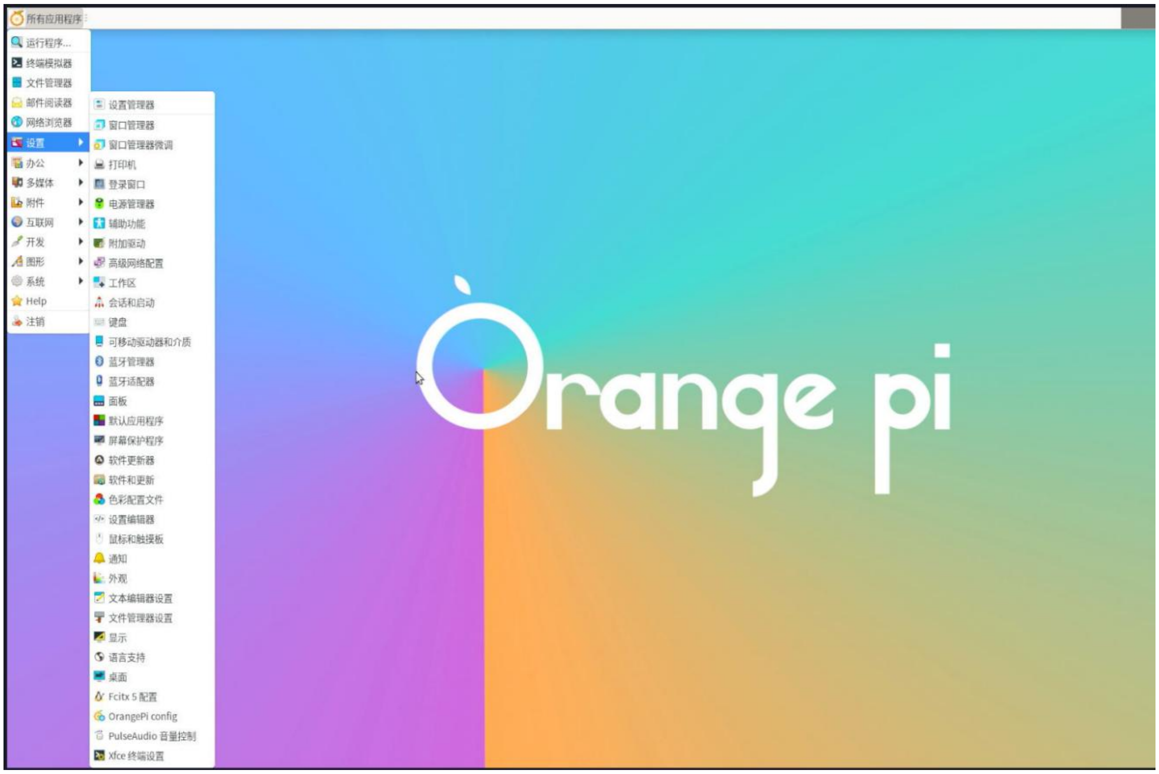
5) Then set the **Keyboard input method system** to **fcitx**



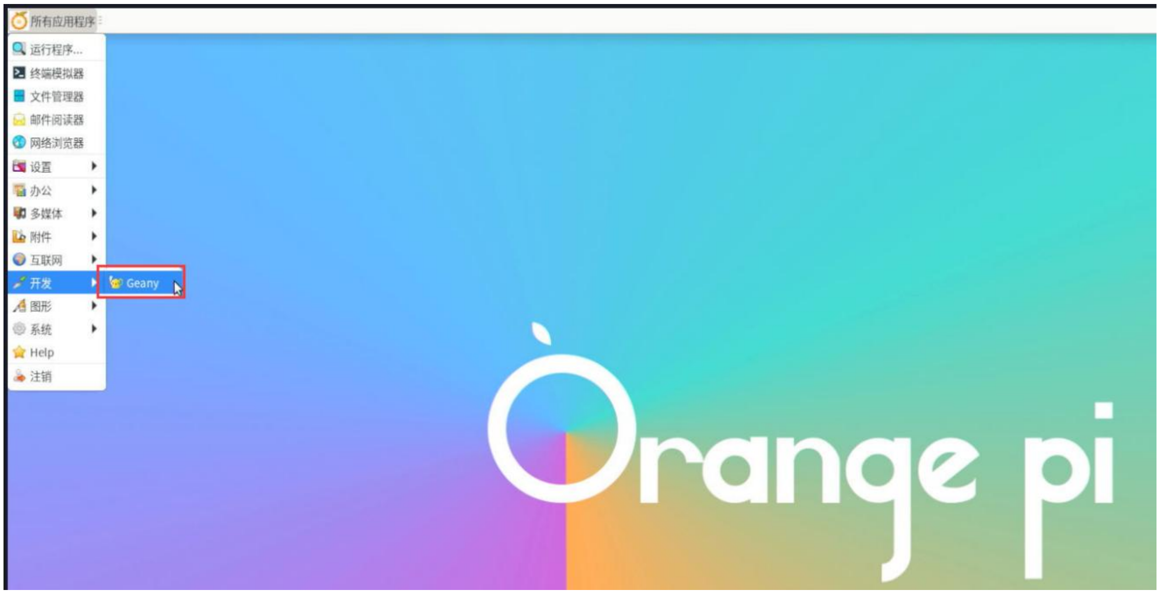
6) **Then restart the Linux system to make the configuration effective**

7) After re-entering the system, please do **not ask me again** at the interface below, and then determine whether the standard folder should be updated as Chinese based on your preference.

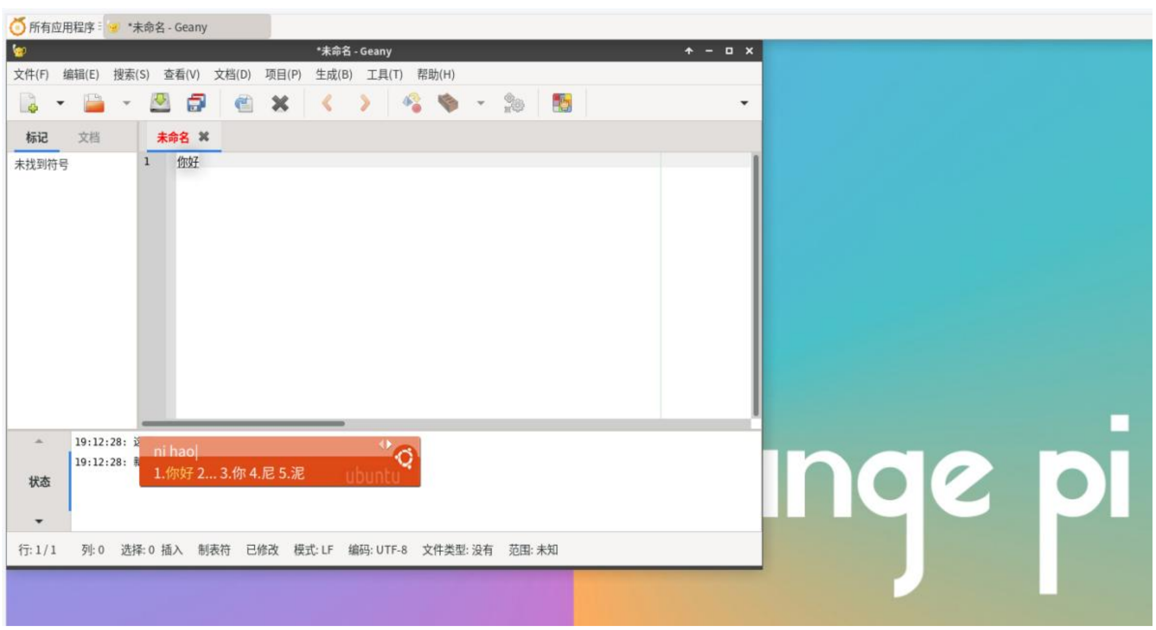
8) Then you can see that the desktop is displayed as Chinese



9) Then we can open the **Geany** to test Chinese input method , and the way to open is shown in the figure below

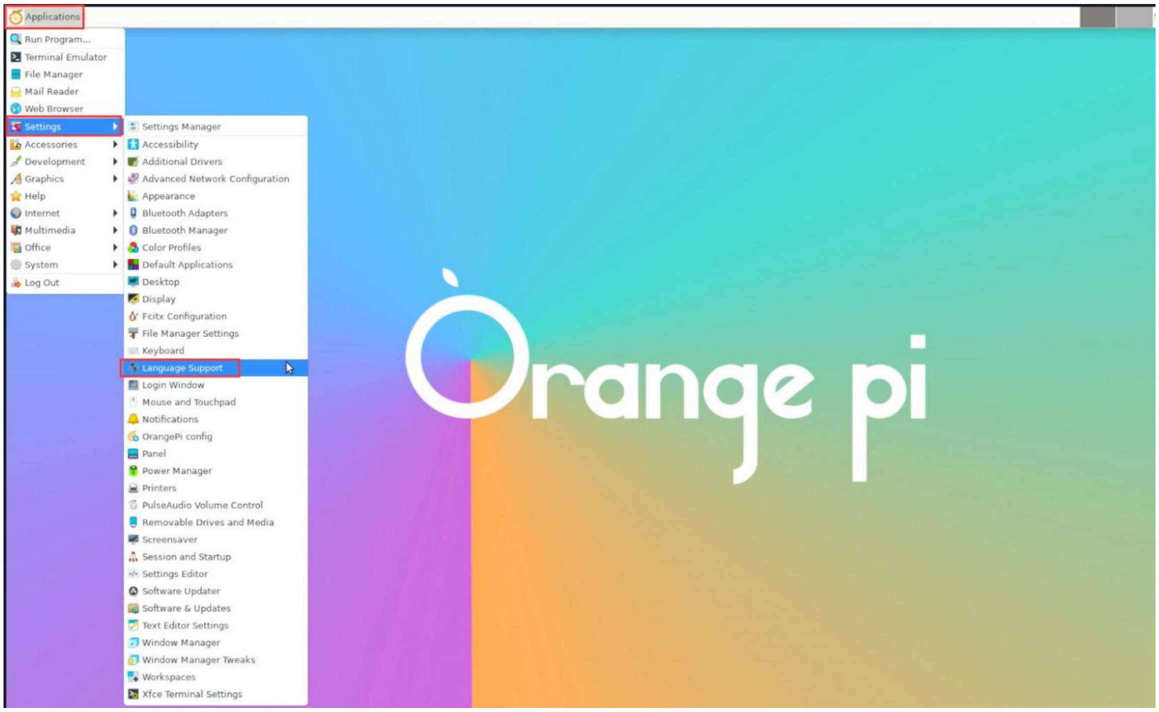


10) After opening **Geany**, the default is an English input method. We can switch into Chinese input method through the **Ctrl+Space** shortcut keys, and then we can enter Chinese

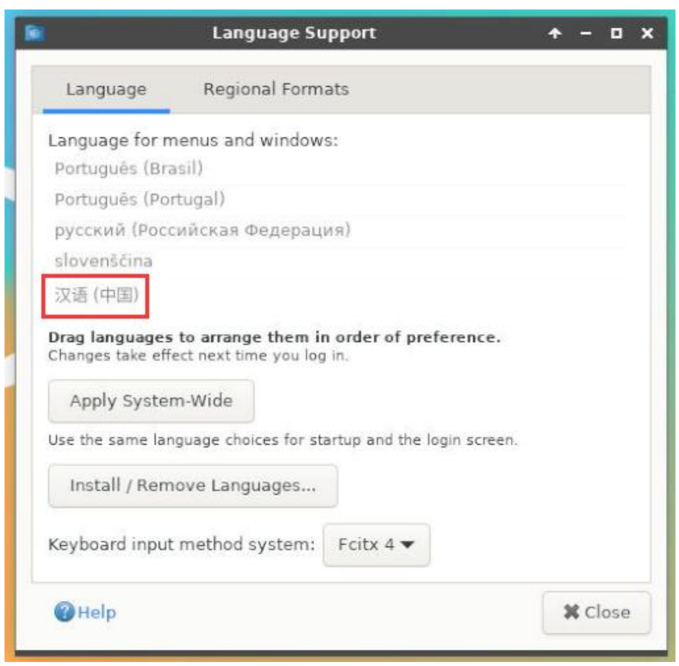


3. 23. 3. The installation method of ubuntu 22.04 system

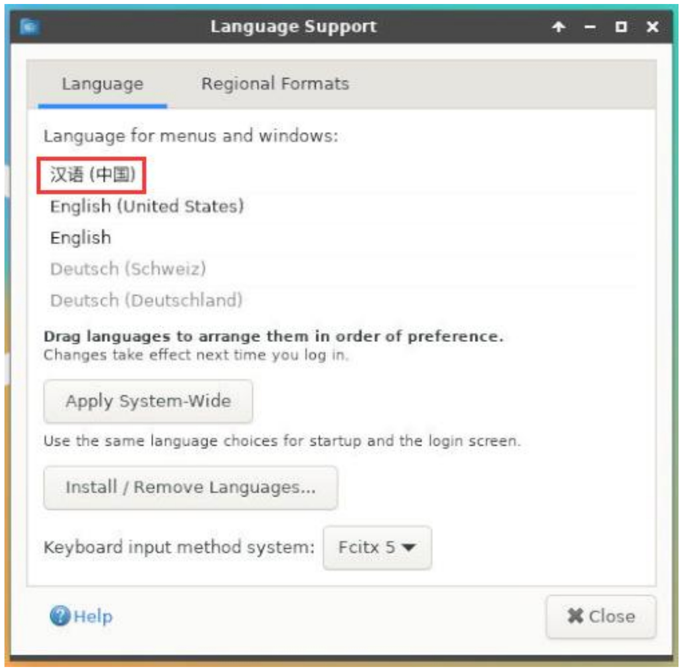
1) First open **Language Support**



2) Then find **Chinese (China)** option

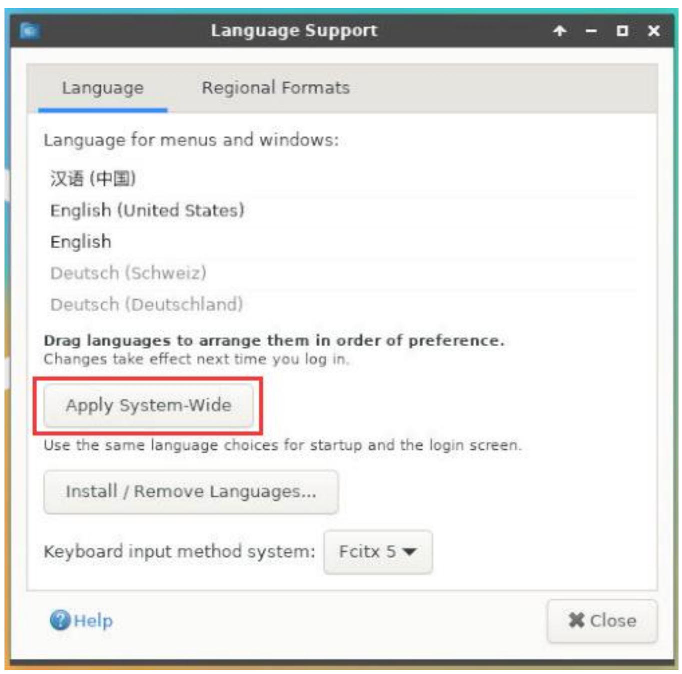


3) Then use the left mouse button to select **Chinese (China)** and hold it down, and then drag it up to the beginning. The display after dragging is shown in the figure below：



Note that this step is not easy to drag, please try more patiently.

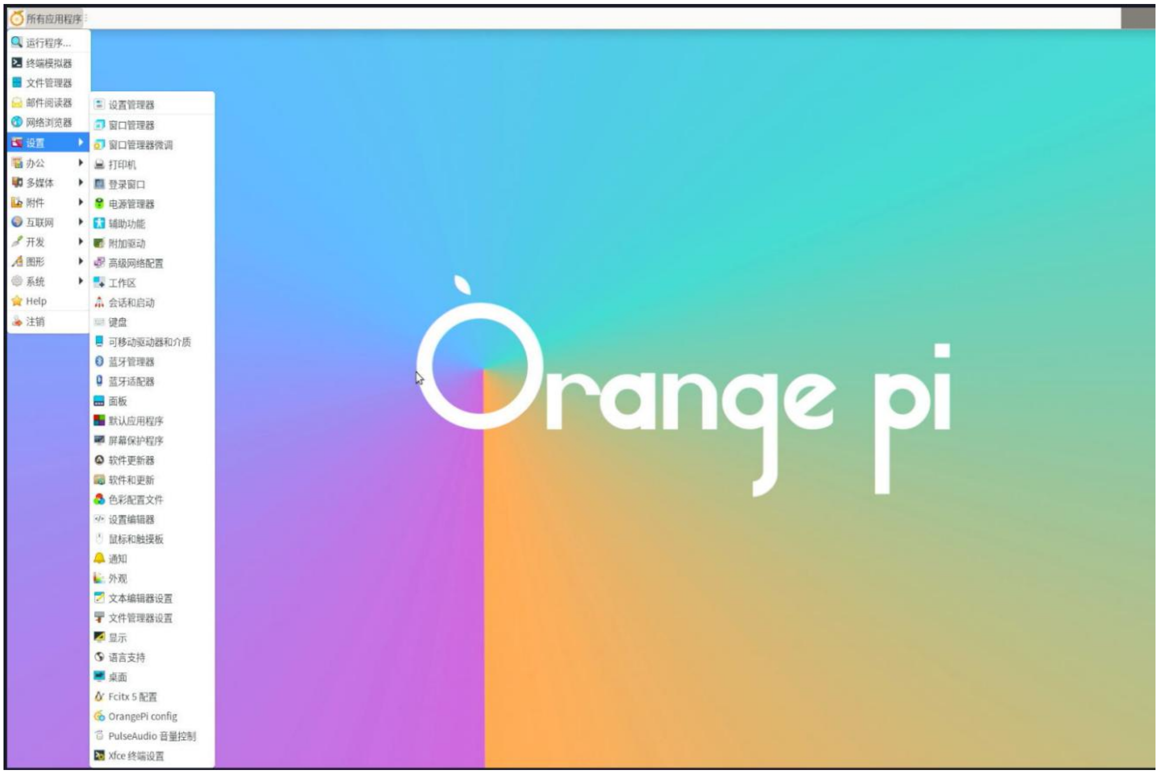
4) Then select the **Apply System-Wide** to apply the Chinese settings to the entire system



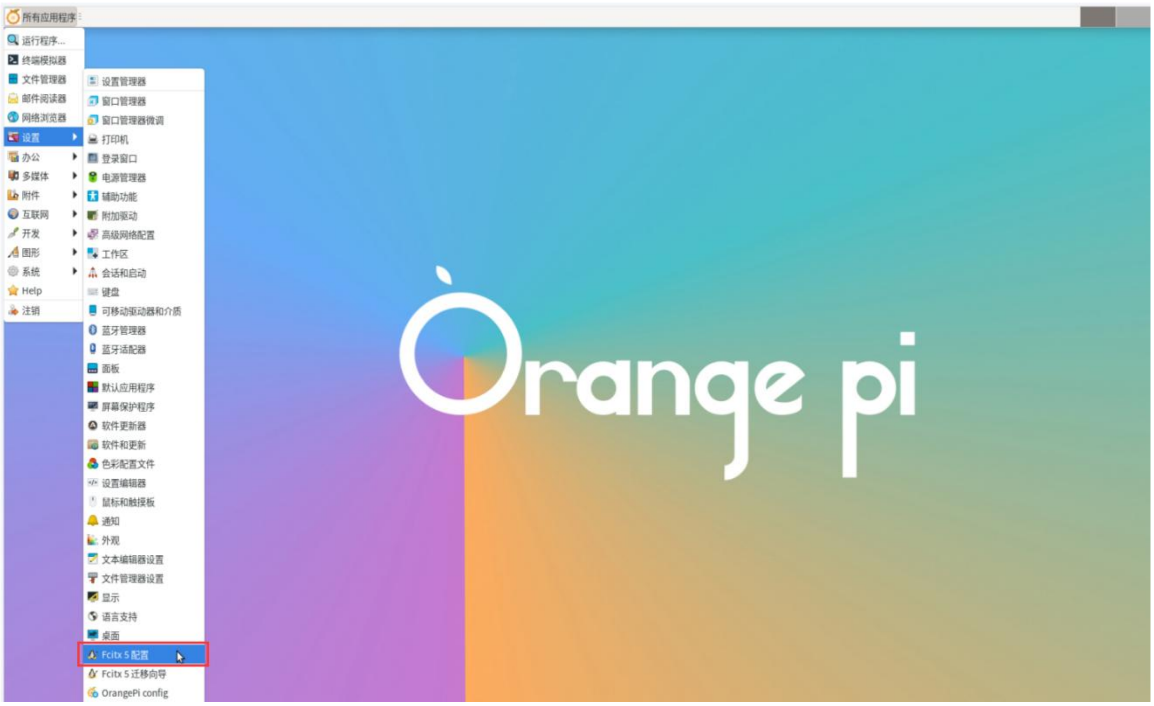
5) **Then restart the Linux system to make the configuration effective**

6) After re-entering the system, please **do not ask me again** at the interface below, and then determine whether the standard folder should be updated as Chinese based on your preference

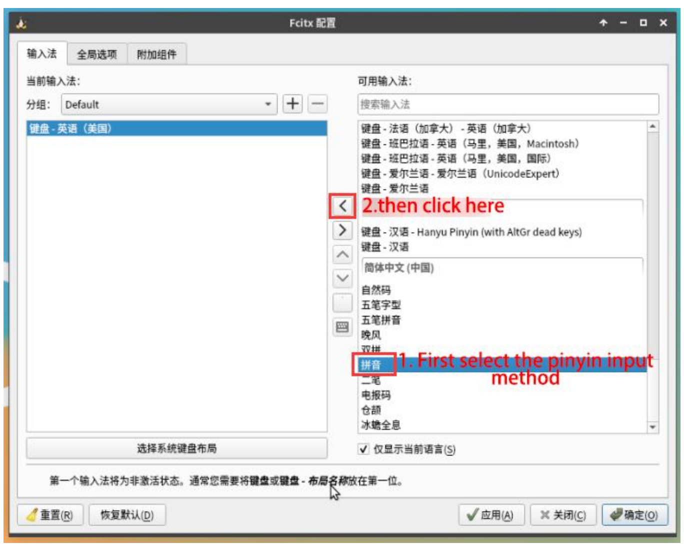
7) Then you can see that the desktop is displayed as Chinese



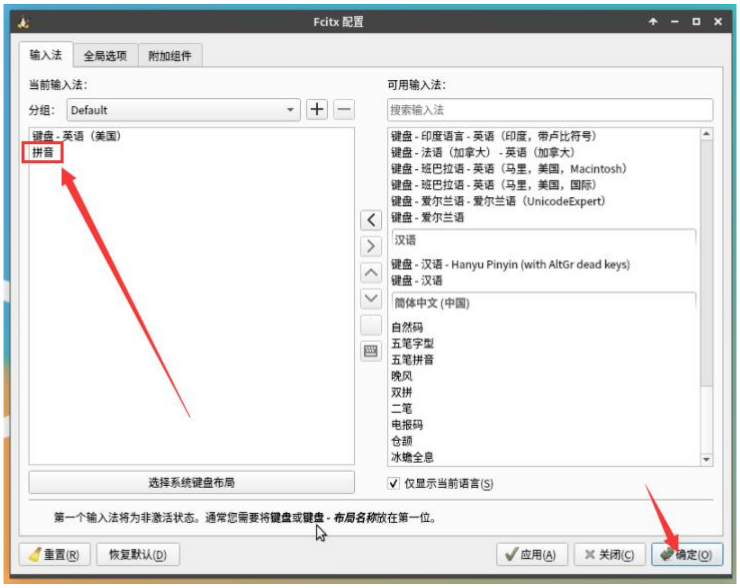
8) Then open the Fcitx5 configuration program



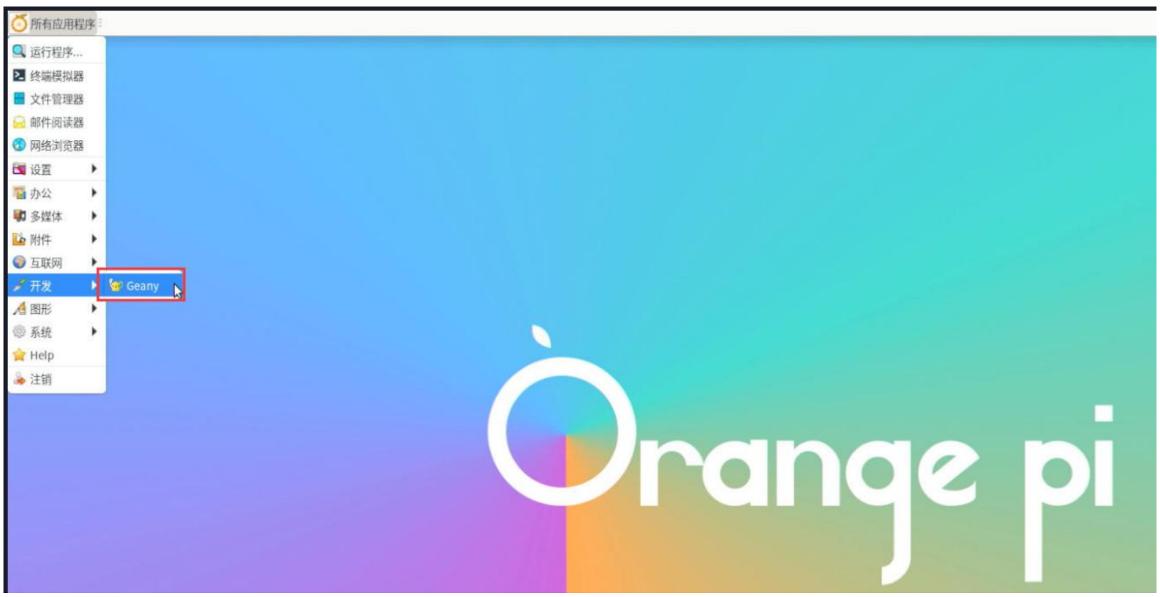
9) Then choose to use Pinyin input method



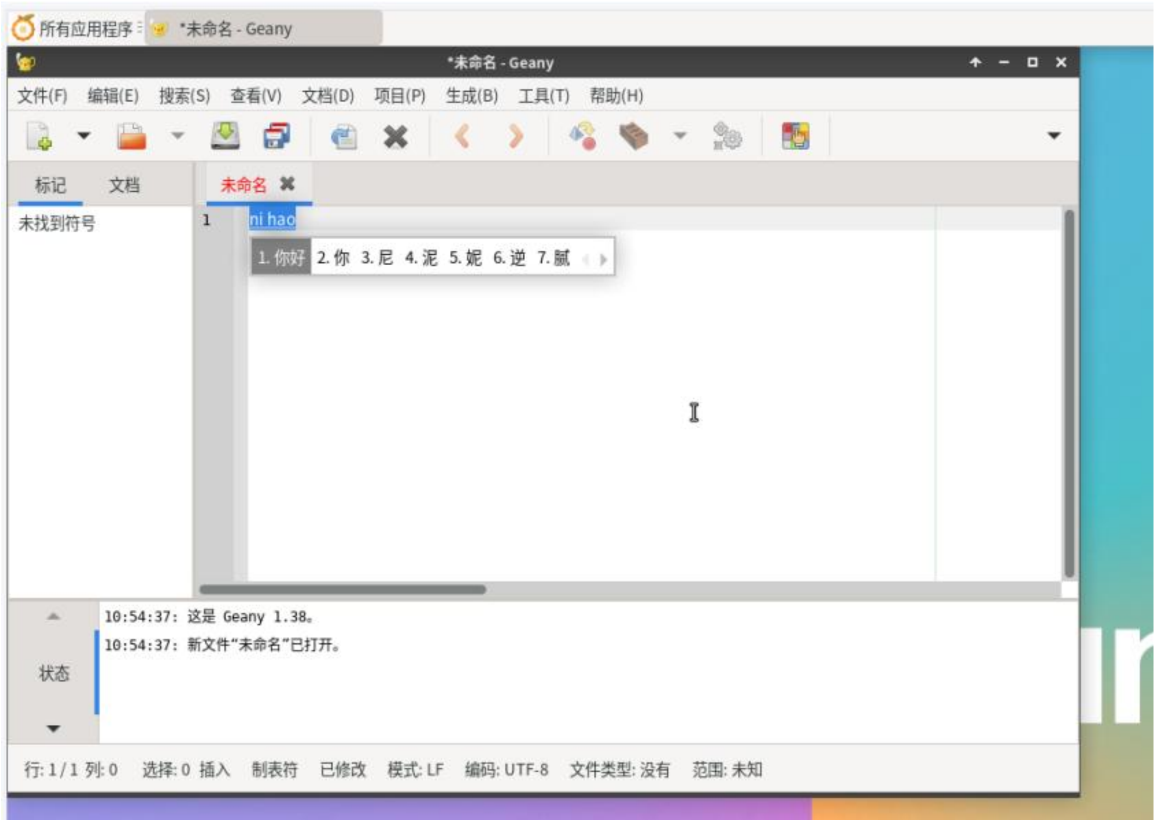
10) The interface after the selection is shown below, then click OK



11) Then we can open the **Geany** to test Chinese input method, and the way to open is shown in the figure below



12) After opening **Geany**, it is still an English input method by default. We can switch into Chinese input methods through the **Ctrl+Space** shortcut keys, and then we can enter Chinese



3. 24. How to remotely log in to the Linux system desktop method

3. 24. 1. Use NoMachine remote login

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.

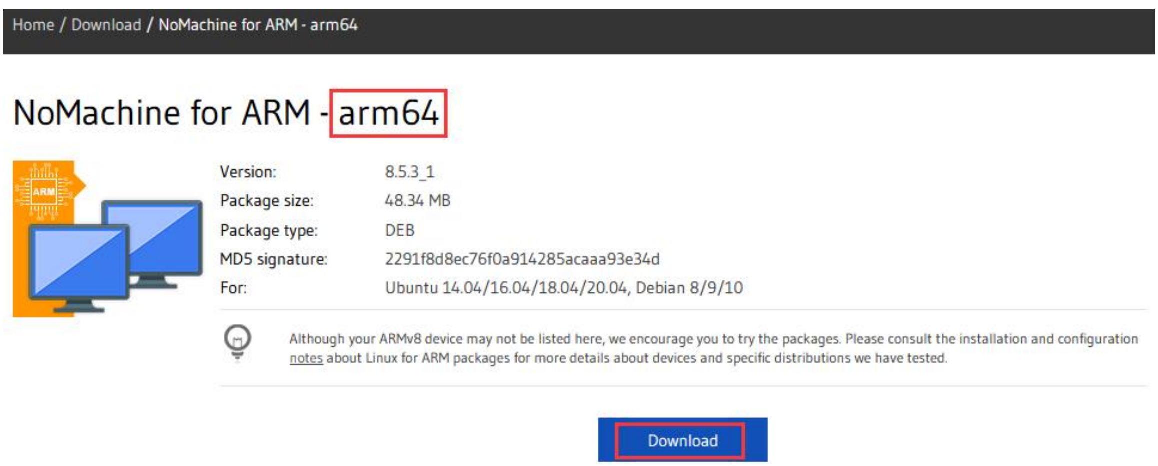
Before operation, please make sure that the Windows computer and the development board are in the same LAN, and you can log in to the Ubuntu or Debian system of the development board through ssh.

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:

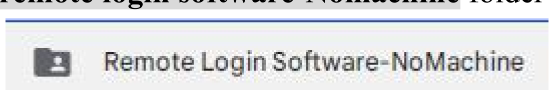
Note that this download link may change, please look for the deb package of the Armv8/Arm64 version.

https://downloads.nomachine.com/download/?id=116&distro=ARM

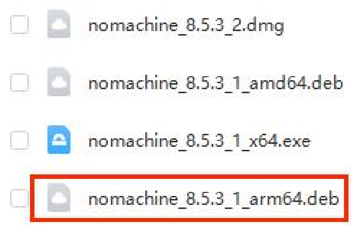


b. In addition, you can also download the installation package to **NoMachine** in the **official tools**.

First enter the **remote login software-Nomachine** folder



Then download the ARM64 version of the DEB installation package



c. Then upload the downloaded **nomachine_x.x.x_x_arm64.deb** to the Linux system of the development board.

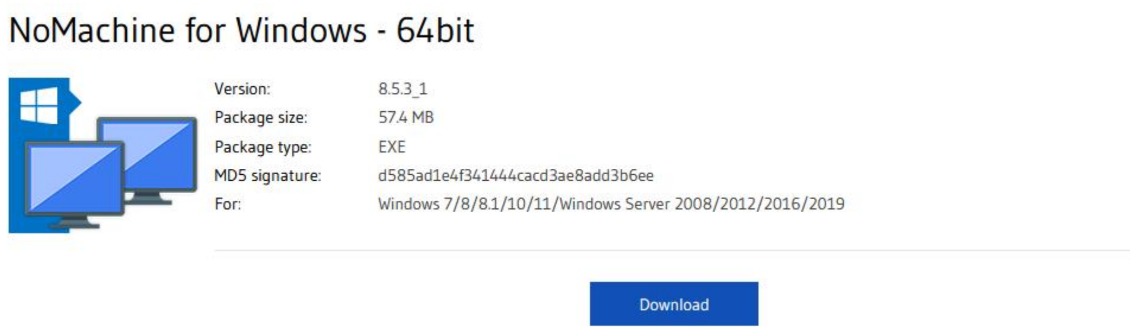
d. Then use the following command to install **NoMachine** in the Linux system of the development board

orangepi@orangepi:~$ **sudo dpkg -i nomachine_x.x.x_x_arm64_arm64.deb**

2) Then download the installation package of the Windows version of the NoMachine software, the download address is as follows

Note that this download link may change.

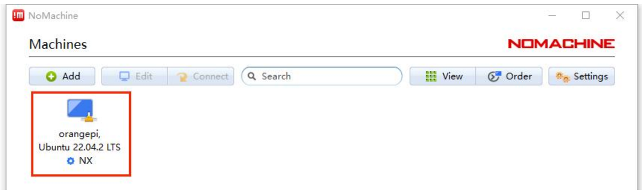
https://downloads.nomachine.com/download/?id=8



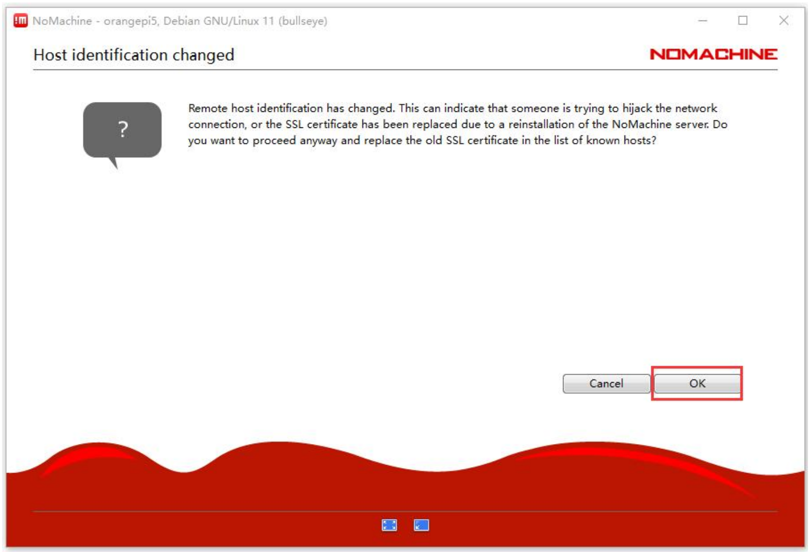
3) Then install nomachine in Windows. **Please restart the computer after installation**

4) Then open **NoMachine** in Window

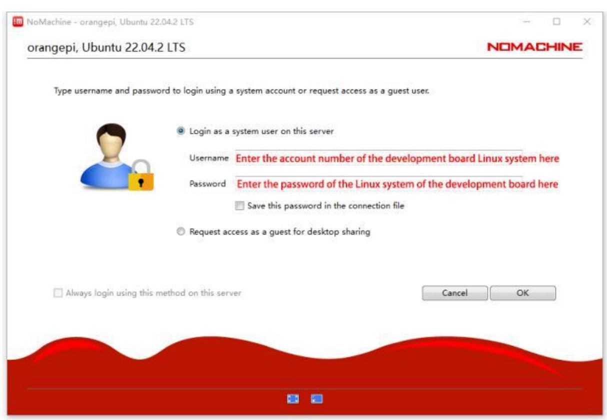
5) After NoMachine starts, it will automatically scan other devices installed with NoMachine in the LAN. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices, and then click the position shown in the red box in the figure below You can start to log in to the Linux system desktop of the development board



6) Then click **OK**

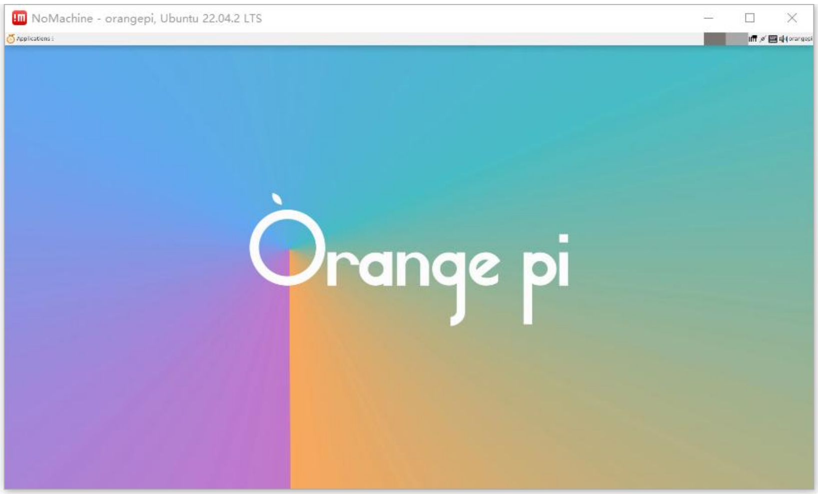


7) Then enter the username and password of the Linux system in the corresponding position in the figure below, and then click **OK** to start logging in



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 remote login

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.

Ubuntu 20.04 tests many problems with VNC, please do not use this method。

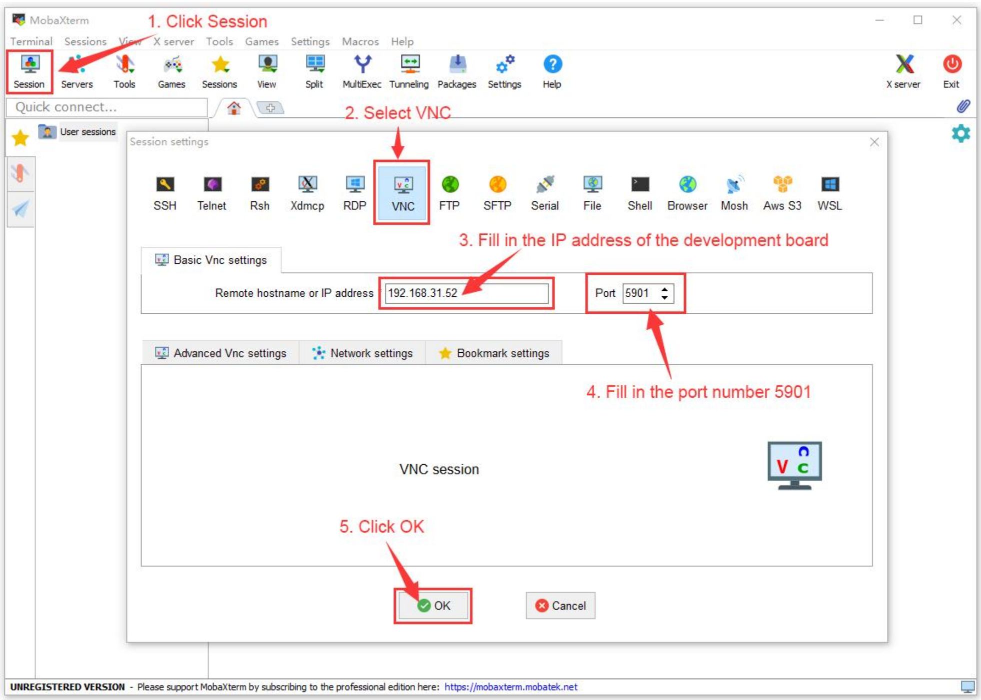
1) First run the **set_vnc.sh** script settings, and **remember to add Sudo permissions**

```
orangepi@orangepi:~$ sudo set_vnc.sh
You will require a password to access your desktops.

Password:          #Set the VNC password here, 8 -bit characters
Verify:            #Set the VNC password here, 8 -bit characters
Would you like to enter a view-only password (y/n)? n
xauth:   file /root/.Xauthority does not exist

New 'X' desktop is orangepi3b:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi3b:1.log

Killing Xtightvnc process ID 3047

New 'X' desktop is orangepi3b:1
```

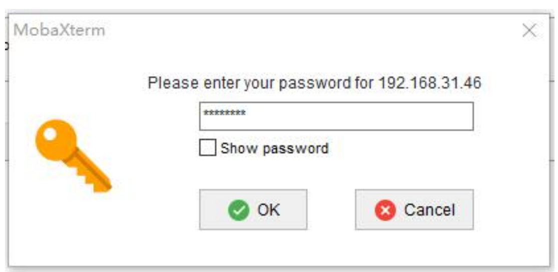
```
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi3b:1.log
```

2) The steps to use the MobaXterm software to connect to the Linux system desktop of the development board are as follows:

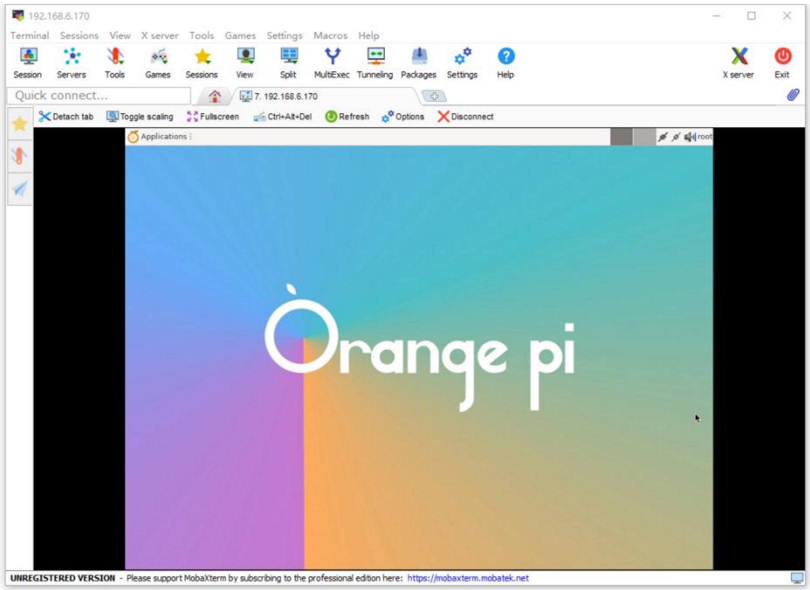
a. First click on Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm



b. Then enter the password of the previously set VNC



c. The interface after the login is shown as shown in the figure below, and then the desktop of the Linux system can be remotely operated



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 C language of **Hello_world.c** program

```
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. **hello_world.py** program in Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bullseye does not install Java compilation tools and operating environment by default

a. You can use the following command to install openjdk, the latest version in Debian Bullseye is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

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

c. Edit the **hello_world.java** of the Jave version

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
```

```
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 25. 2. Debian Bookworm system

1) Debian Bookworm is installed with a 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 the **hello_world.py** program in Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The result of running **hello_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bookworm does not install Java compilation tools and operating environment by default

a. You can use the following command to install openjdk, the latest version in Debian Bookworm is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

b. After installation, you can check the version of Java

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

c. Write the Java version of **hello_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 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 a.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. **hello_world.c** program to write 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 defaults to install Python3

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. **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's compilation tool and operating environment without the installation of

Java 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. Edit the **hello_world.java** of Jave version

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 25. 4. Ubuntu Jammy system

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

a. The version of gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.2.0-19ubuntu1) 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 of 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 is installed with Python3 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. Edit **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 defaults to compile tools and operating environments that are not installed in Java

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. Edit the **hello_world.java** of the Jave version

```
orangepi@orangepi:~$ vim hello_world.java
```

```
public class hello_world
{
        public static void main(String[] args)
        {
                System.out.println("Hello World!");
        }
}
```

d. Then compile and run **hello_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

3. 26. QT installation method

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

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

2) The QT version number will be automatically printed after installation

a. Ubuntu20.04 comes with QT version **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. Ubuntu22.04 comes with QT version **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. Debian11 comes with QT version **5.15.2**

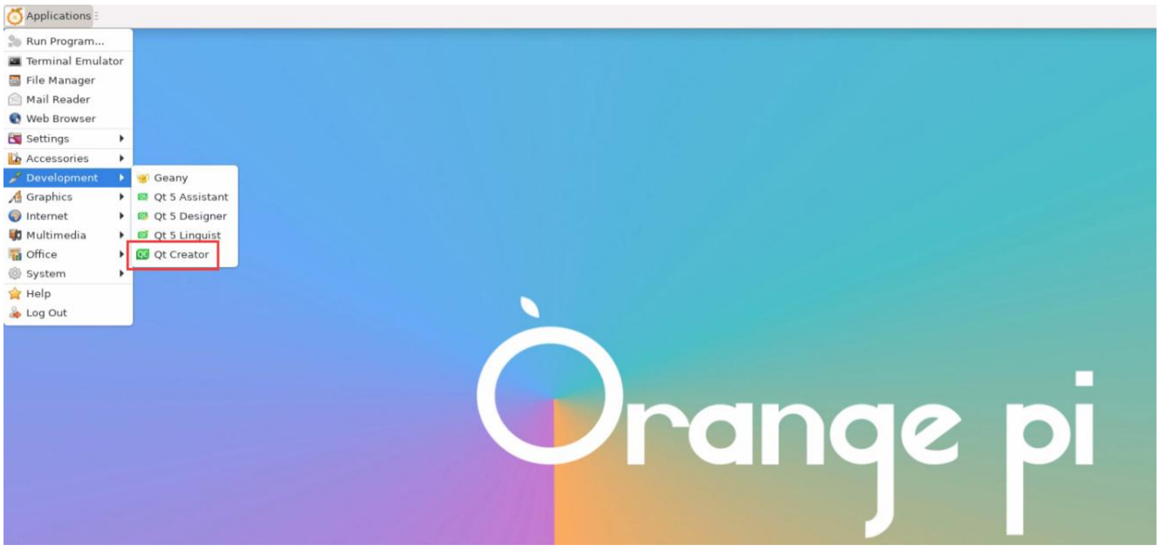
```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

d. Debian12 comes with QT version **5.15.8**

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

```
......
QMake version 3.1
Using Qt version 5.15.8 in /usr/lib/aarch64-linux-gnu
```

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



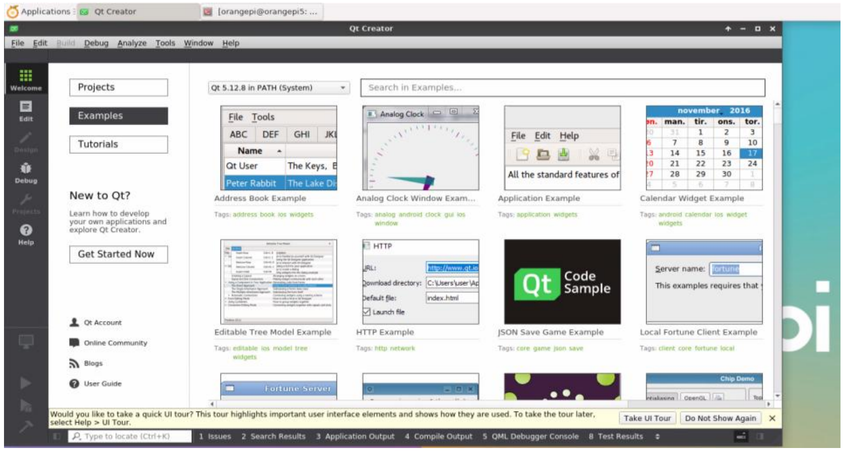
QT Creator can also be opened using the following command

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

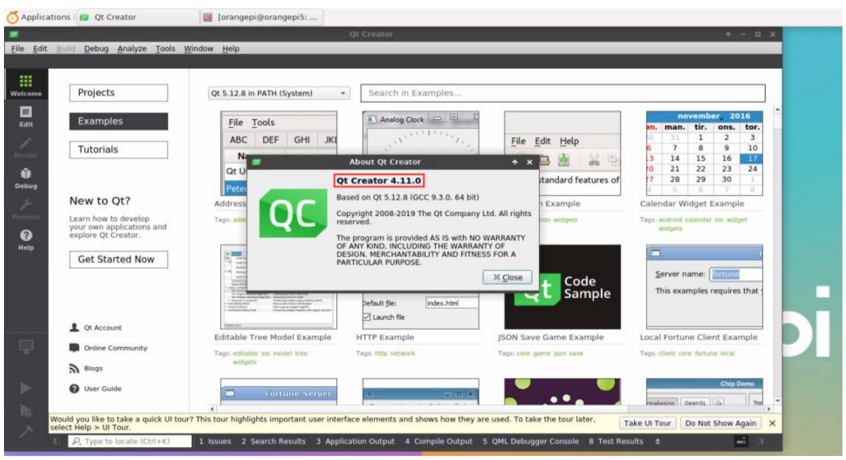
During the startup of QT and QT applications, if the following error is displayed, ignore it. This error has no impact on application running.

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

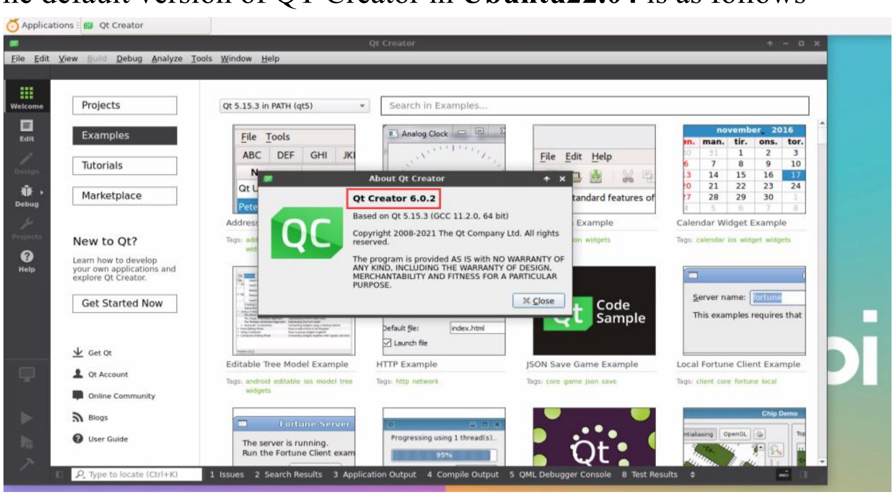
4) The interface after QT Creator is opened is as follows



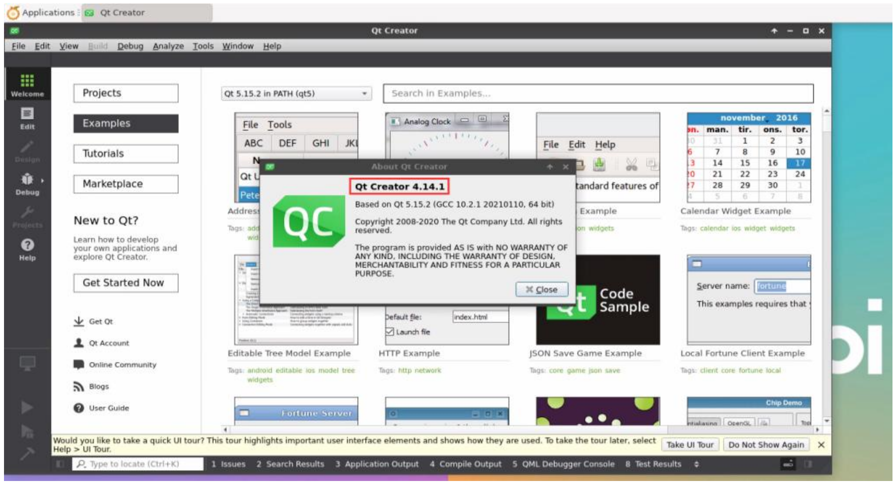
5) The QT Creator version is shown below
   a. The default version of QT Creator in **Ubuntu20.04** is as follows



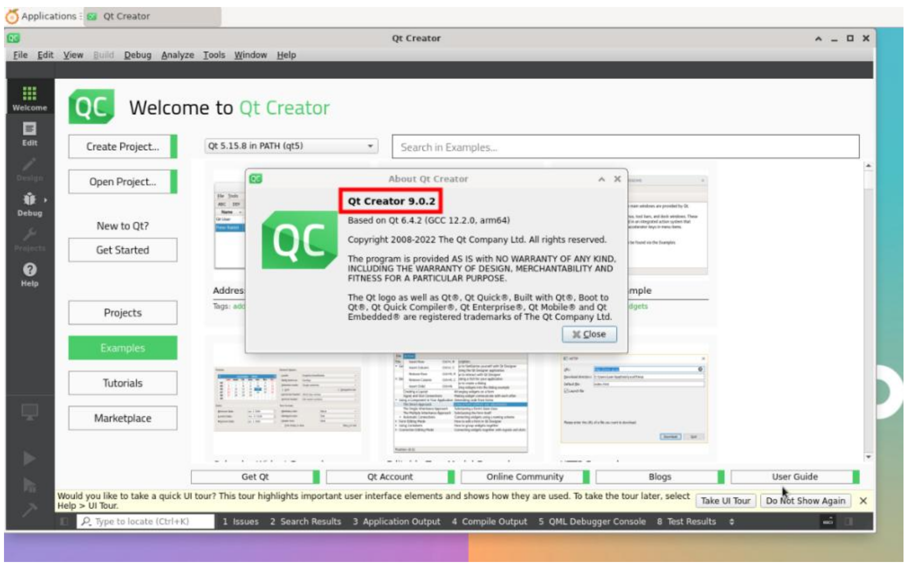
   b. The default version of QT Creator in **Ubuntu22.04** is as follows



   c. The default version of QT Creator in **Debian11** is as follows

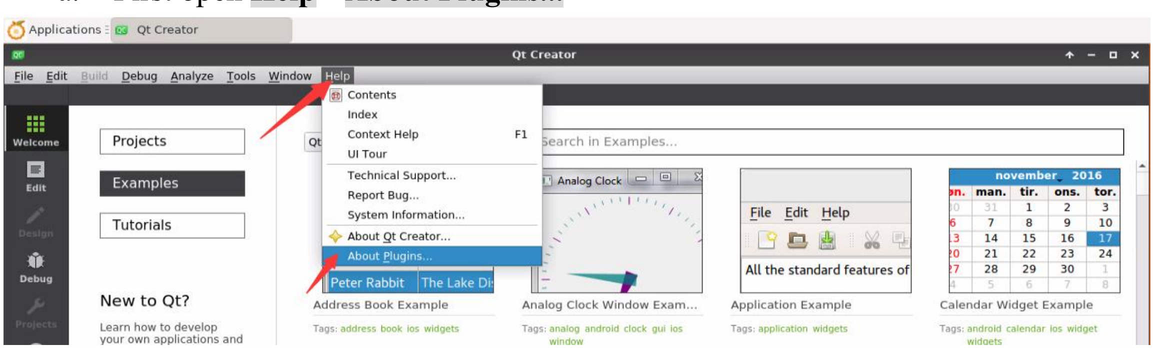


d. The default version of QT Creator in **Debian12** is as follows

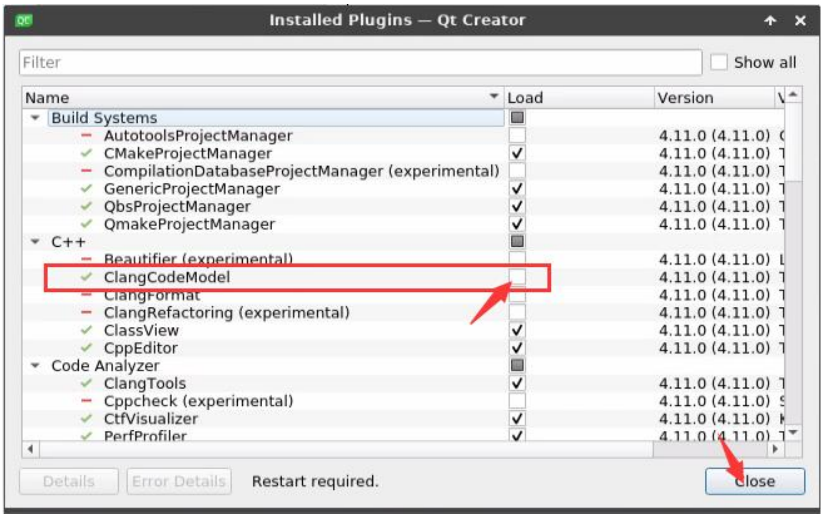


6) Then set QT

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



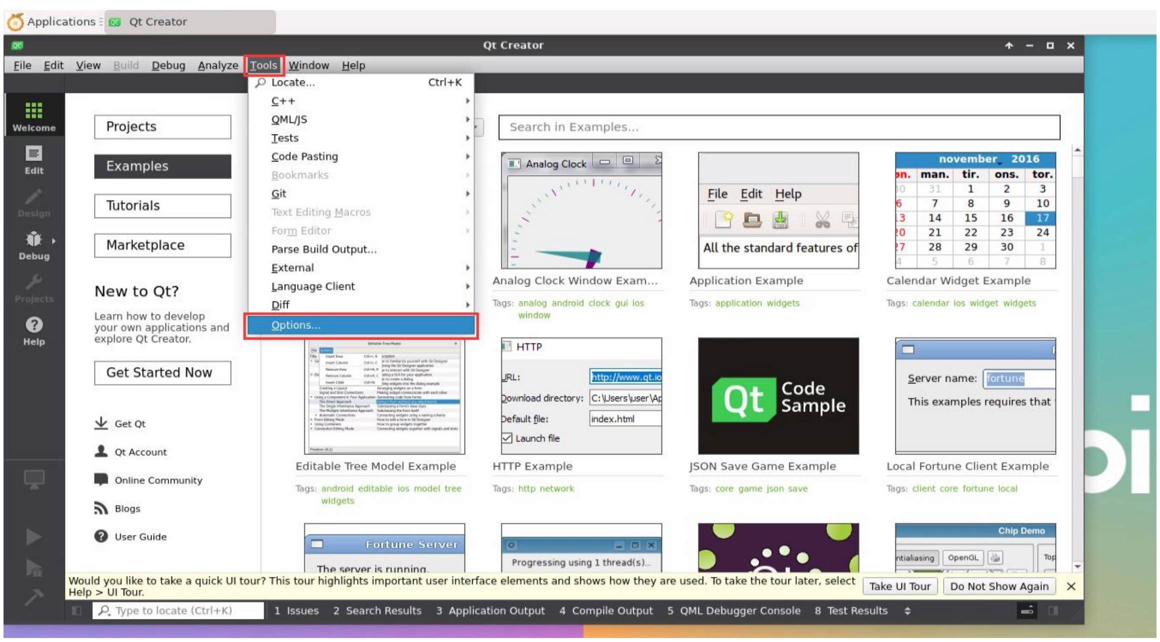
b. Then remove the check box for **ClangCodeModel**

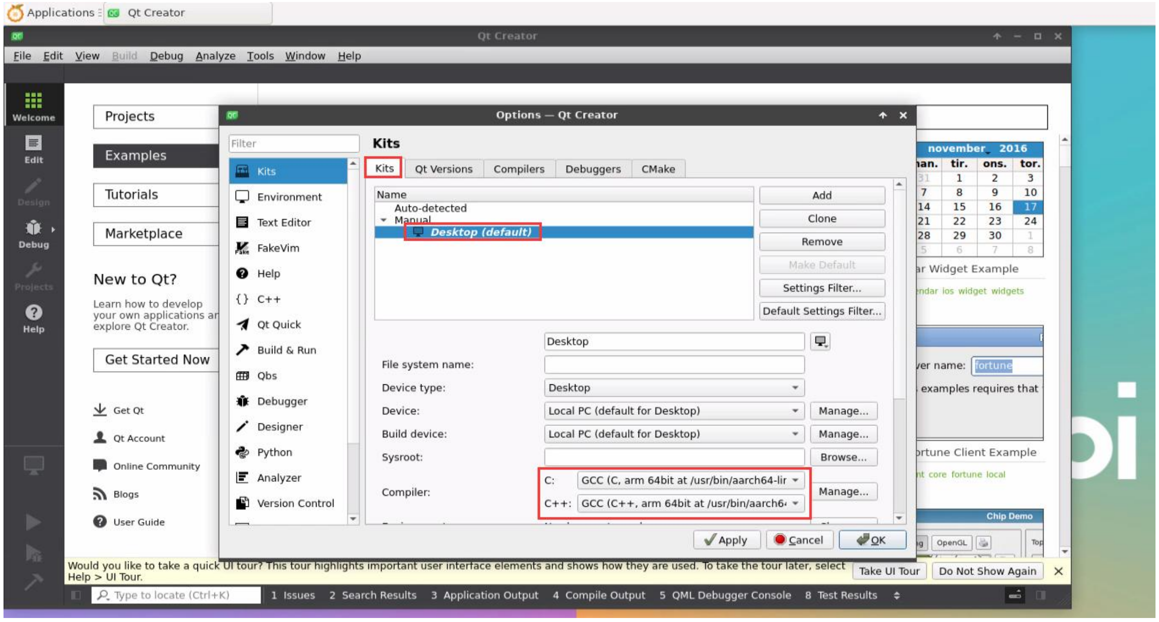


c. **Restart QT Creator after the Settings are complete**

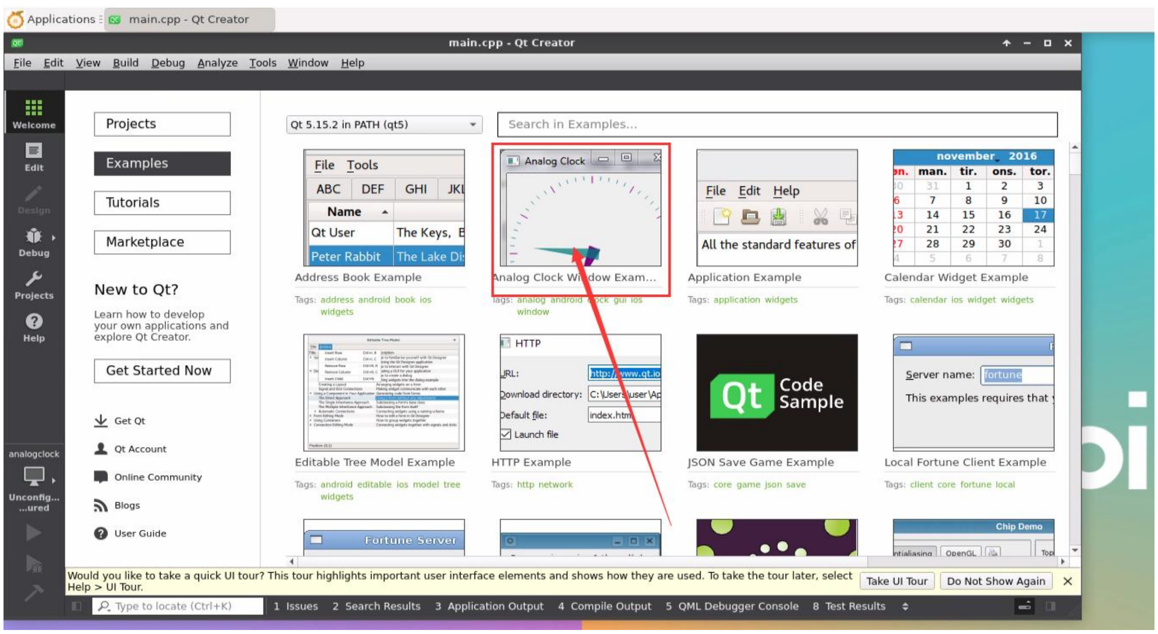
d. Then make sure that QT Creator uses the GCC compiler, if the default is Clang, change it to GCC

Debian12 Please skip this step.

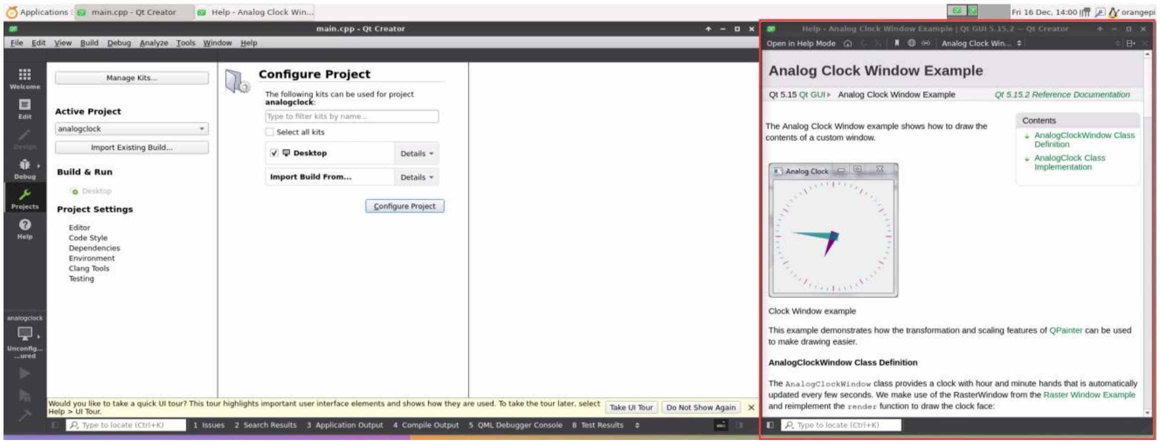




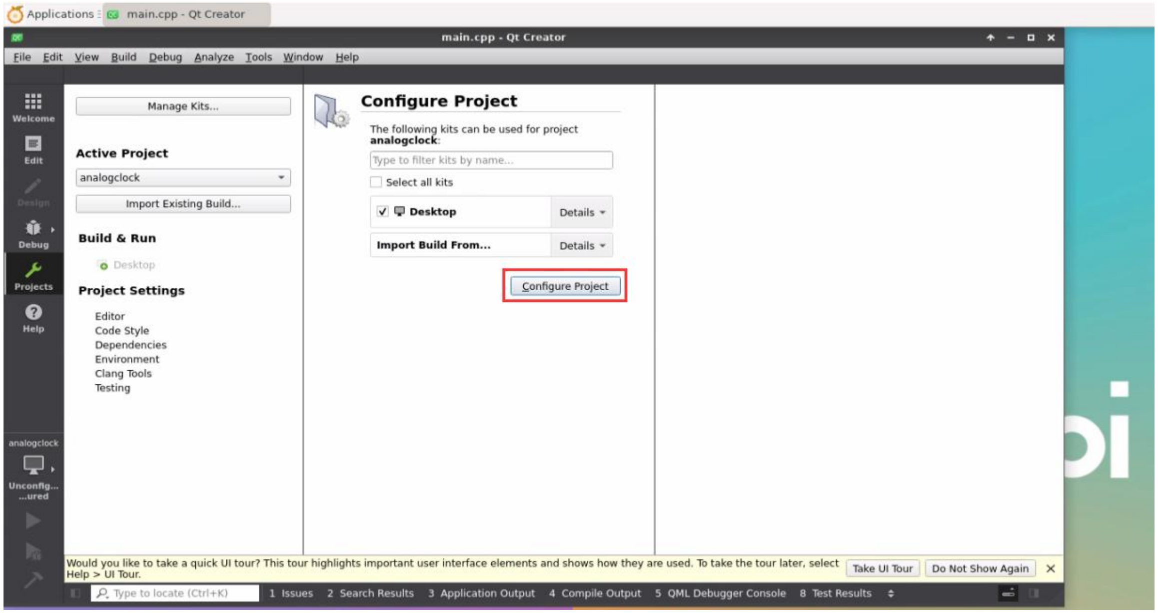
7) You can then open a sample code



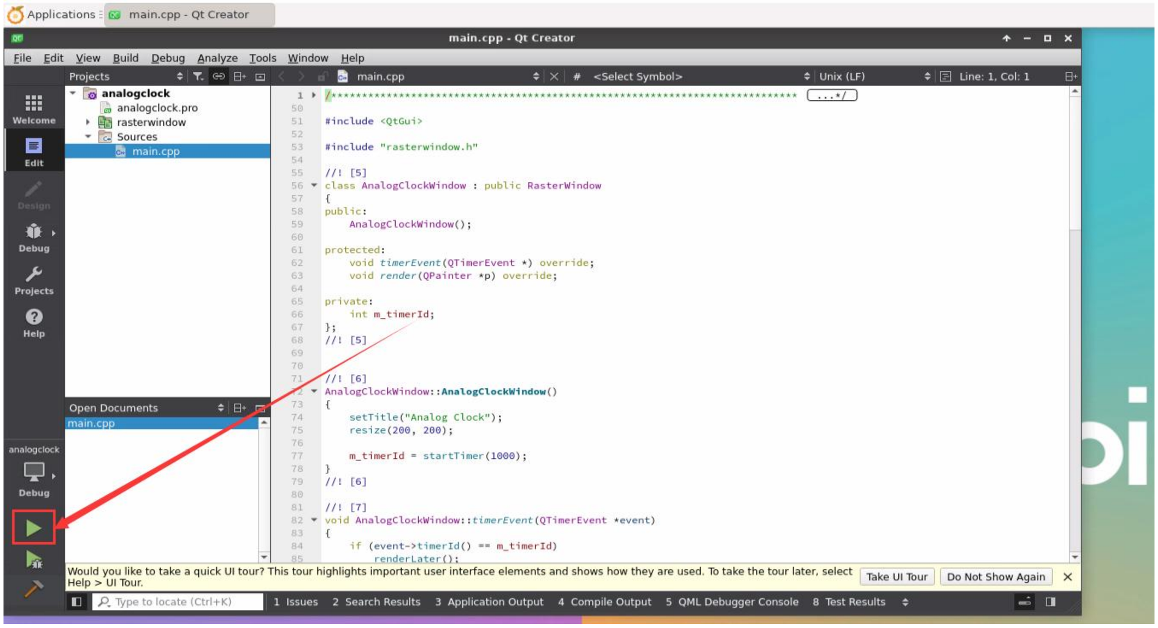
8) Clicking on the example code will automatically open the corresponding instruction document, you can carefully read the instructions



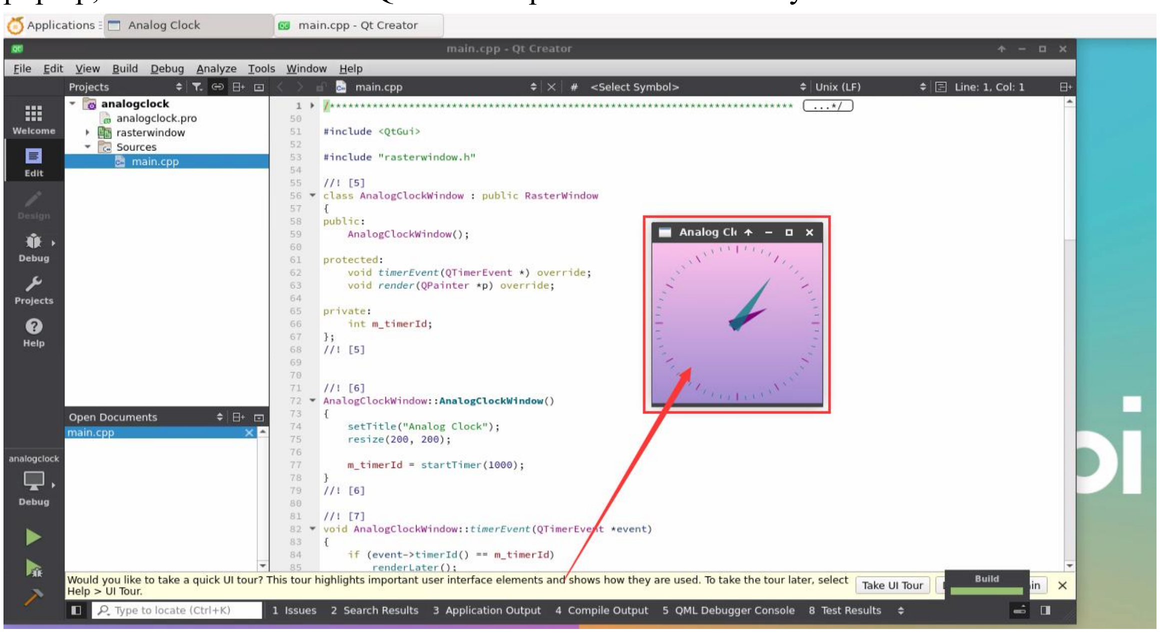
9) Then click **Configure Project**



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



11) After waiting for a period of time, the interface shown in the following figure will pop up, which indicates that QT can compile and run normally



12) Reference documents

https://wiki.qt.io/Install_Qt_5_on_Ubuntu
https://download.qt.io/archive/qtcreator
https://download.qt.io/archive/qt

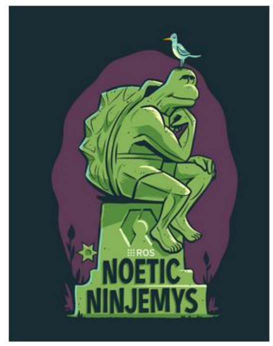
3. 27. ROS Installation Method

3. 27. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The current active version of ROS 1 is as follows, the recommended version is **Noetic Ninjemys**

Active ROS 1 distributions

Recommended



Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

http://docs.ros.org
https://wiki.ros.org/Distributions

2) The official installation document link of ROS 1 **Noetic Ninjemys** is as follows:

http://wiki.ros.org/noetic/Installation/Ubuntu

3) In the 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 script below to install ros1

orangepi@orangepi3b:~$ **install_ros.sh ros1**

5) Before using the ROS tool, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code

Note that running the following command needs to ensure that the development board can access github normally, otherwise an error will be reported due to network problems.

The install_ros.sh script will try to modify /etc/hosts and automatically run the following commands. However, this method cannot guarantee normal access to github every time. If the following error is displayed after installing ros1 in install_ros.sh, please find other ways to allow the Linux system of the development board to access github normally, and then manually run the following Order.

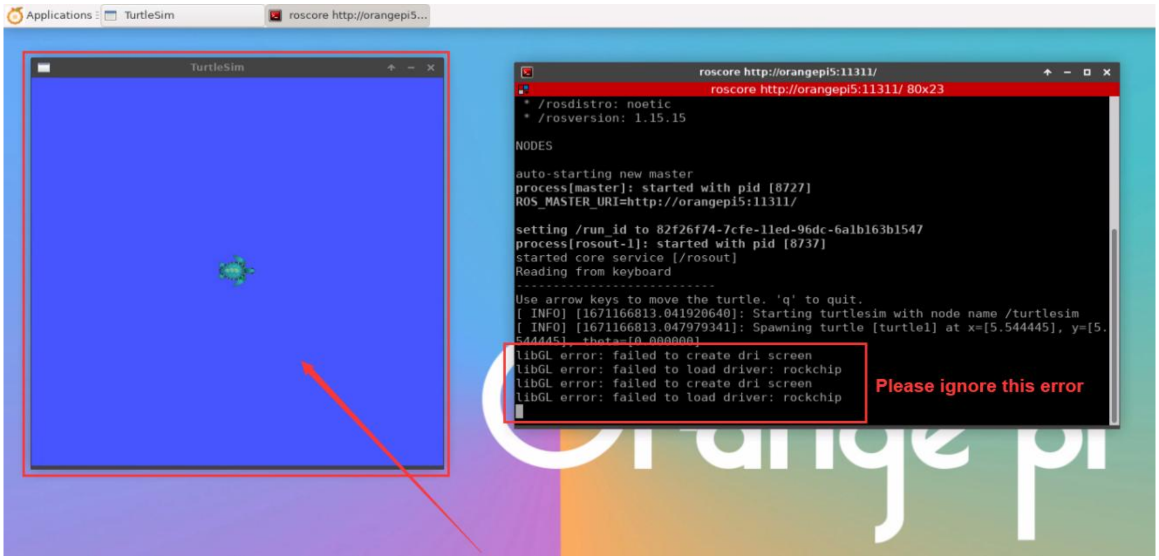
https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
ERROR: error loading sources list:
The read operation timed out

orangepi@orangepi:~$ **source /opt/ros/noetic/setup.bash**
orangepi@orangepi:~$ **sudo rosdep init**
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

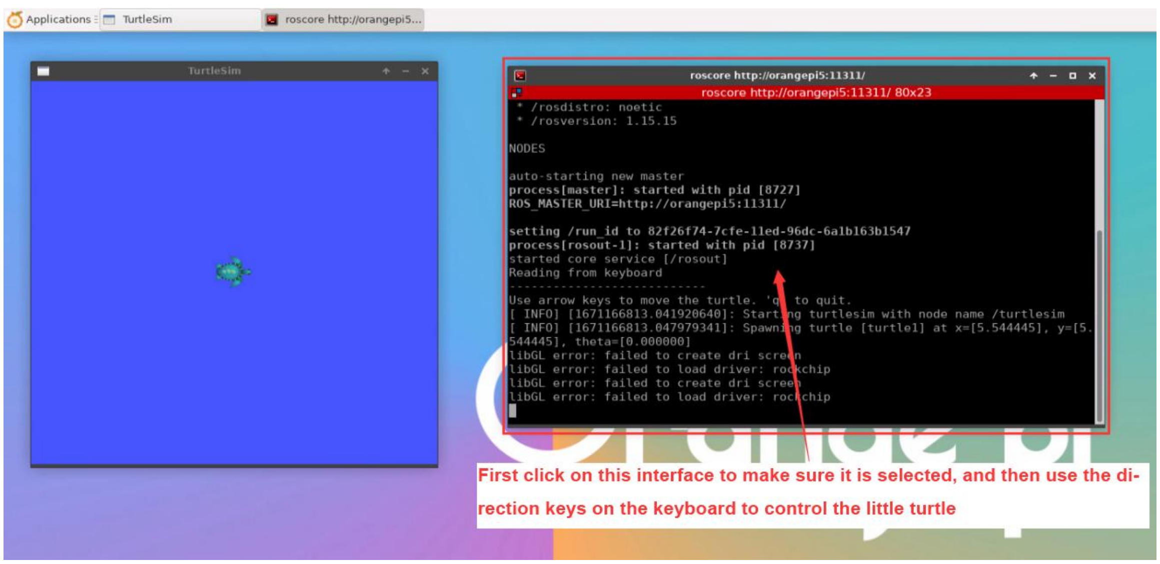
```
        rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

6) Then open a command line terminal window on the desktop, and then use the test_ros.sh script to start a small turtle routine to test whether ROS can be used normallyorangepi@orangepi:~$ **test_ros.sh**

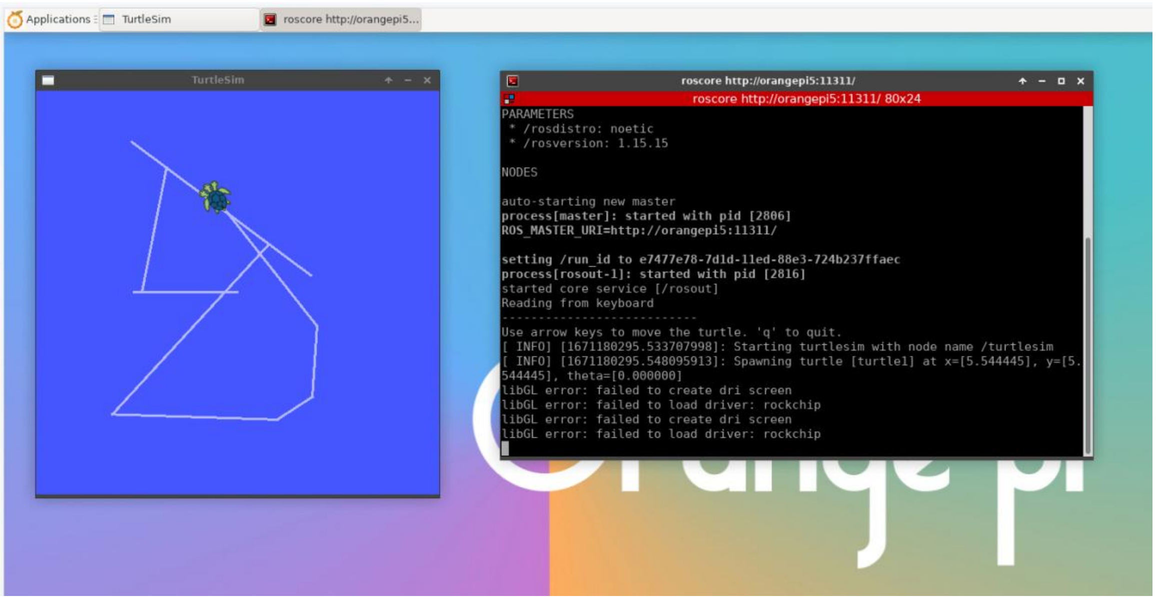
7) After running the **test_ros.sh** script, a little turtle as shown in the figure below will pop up



8) Then please keep the terminal window just opened at the top

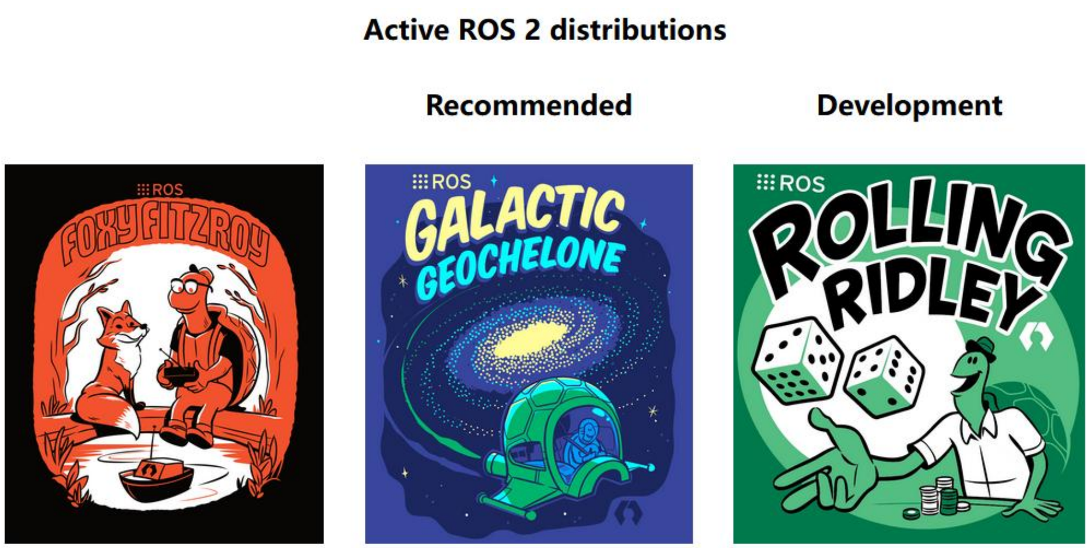


9) At this time, press the direction keys on the keyboard to control the little turtle to move up, down, left, and right



3. 27. 2. How to install ROS 2 Galactic on Ubuntu 20.04

1) The current active version of ROS 2 is as follows, 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) It is recommended to use Ubuntu20.04 in the official installation document of ROS 2 **Galactic Geochelone**, so please ensure that the system used by the development board is **Ubuntu20.04 desktop version**. There are several ways to install ROS 2. The following shows how to install ROS 2 **Galactic Geochelone** using **Debian packages**

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

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

5) 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.
```

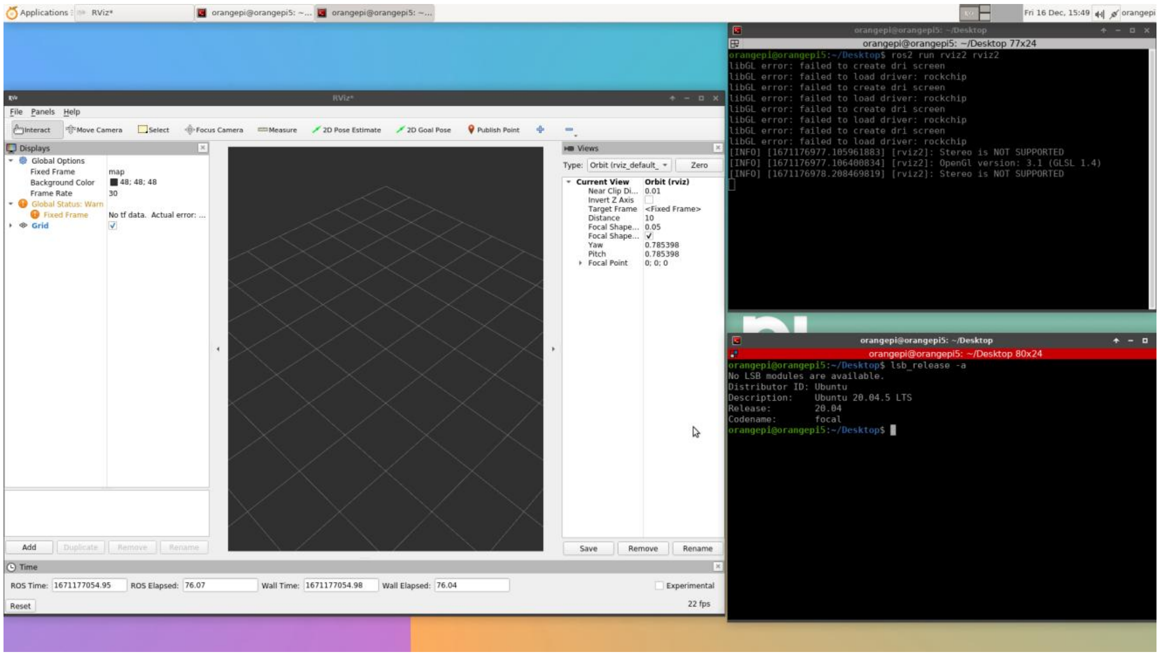
6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepi3b:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
```

orangepi@orangepi:~$ **ros2 run rviz2 rviz2**



8) For the usage of ROS, please refer to the documentation of ROS 2

http://docs.ros.org/en/galactic/Tutorials.html

3. 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
```

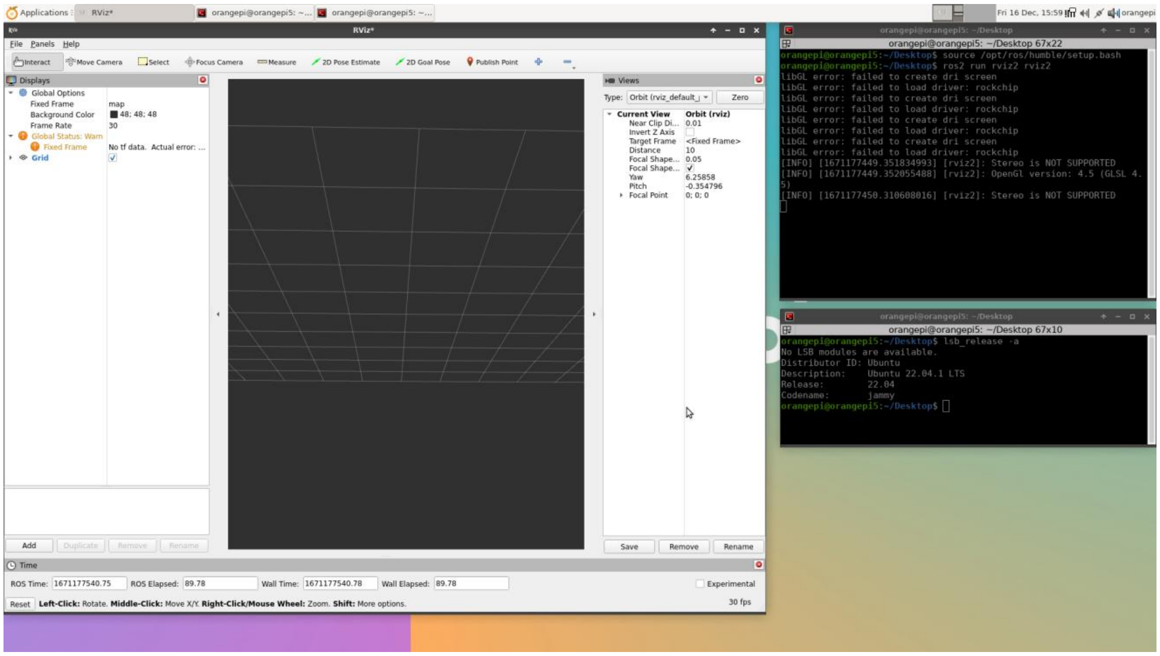
```
  daemon        Various daemon related sub-commands
  doctor        Check ROS setup and other potential issues
  interface     Show information about ROS interfaces
  launch        Run a launch file
  lifecycle     Various lifecycle related sub-commands
  multicast     Various multicast related sub-commands
  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@orangepi3b:~$ 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");
```

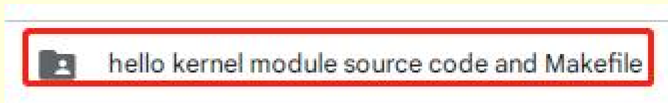
b. Then write the Makefile for compiling the hello kernel module, as follows:

```
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 the Raspberry PI's 5-inch screen

3. 29. 1. Assembly method of Raspberry PI 5-inch screen

Note that linux6.6 is not supported yet.

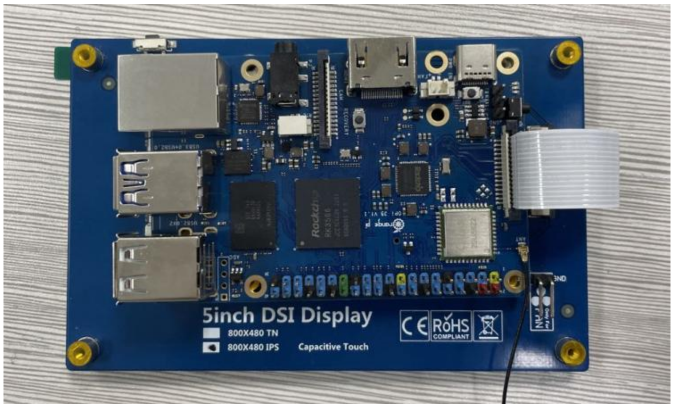
1) First prepare the required accessories

a. Raspberry PI 5-inch MIPI LCD display + touch screen

b. 15pin MIPI cable

2) Then connect the 15pin MIPI cable to the Raspberry PI 5-inch screen in the way shown below (note the orientation of the insulation surface)

3) Finally connect to the LCD interface of the Orange Pi 3B development board



3. 29. 2. Open the Raspberry PI 5-inch screen configuration method

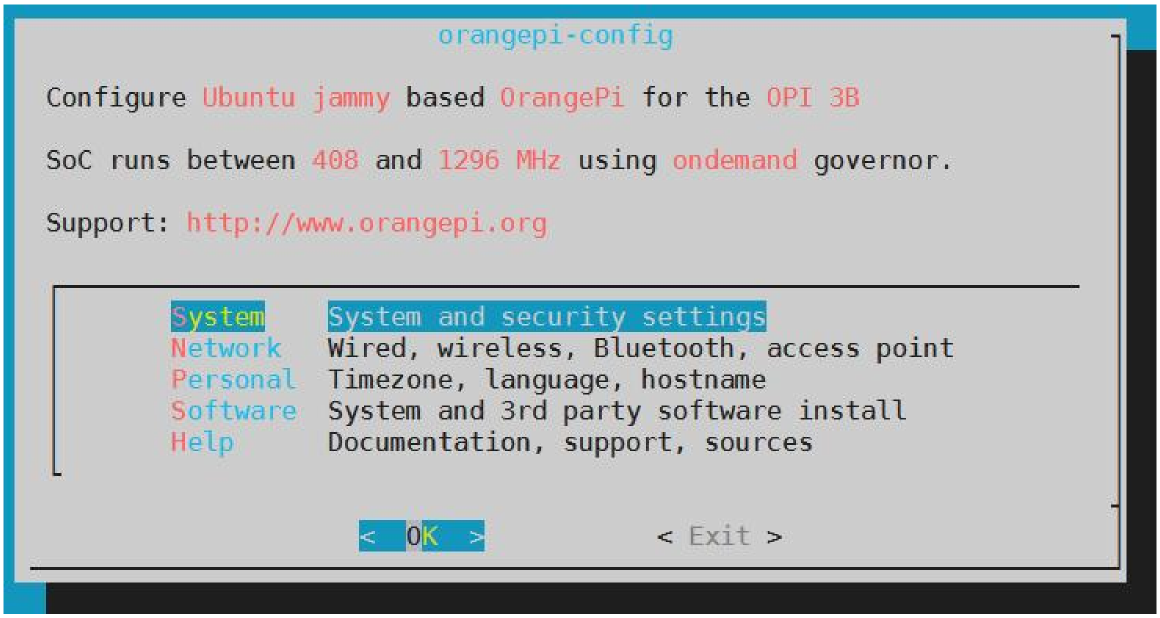
1) The Linux image does not open the 5-inch screen of the Raspberry PI by default. If you need to use the 5-inch screen of the Raspberry PI, you need to open it manually.

2) The steps to open the mipi lcd configuration are as follows:

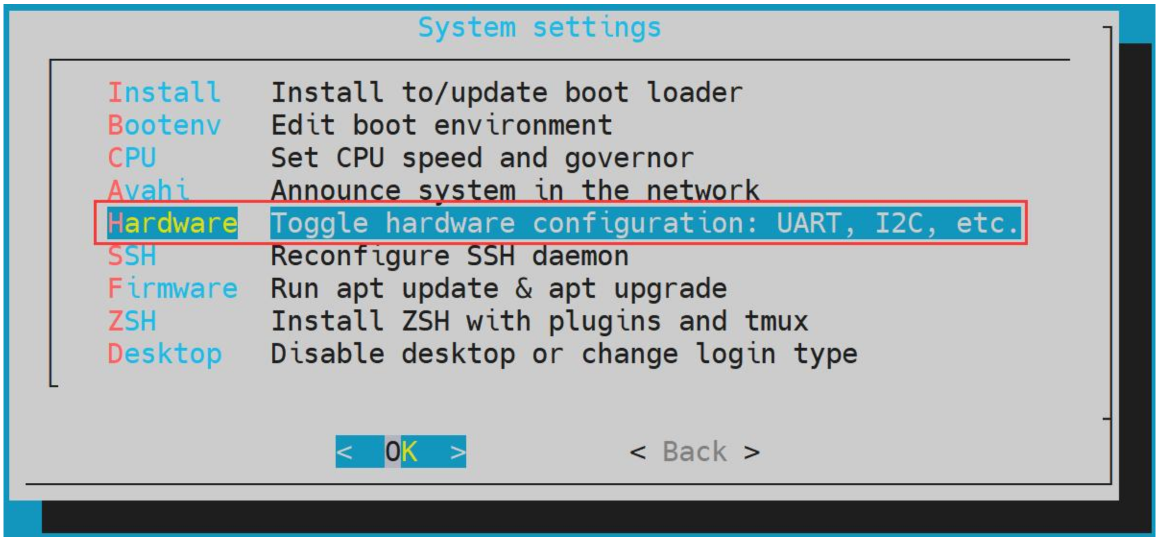
a. First run **orangepi-config**, ordinary users remember to add **sudo** permission

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

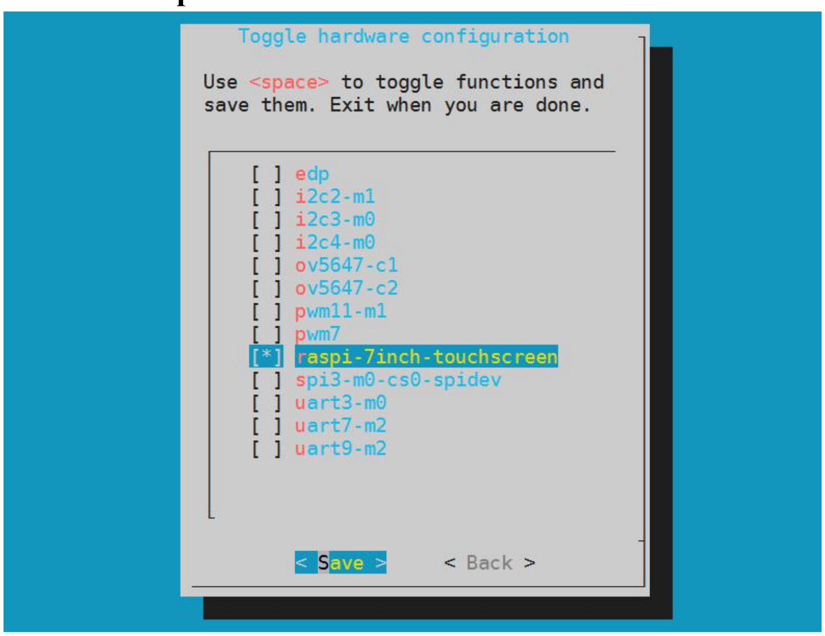
b. Then select **System**



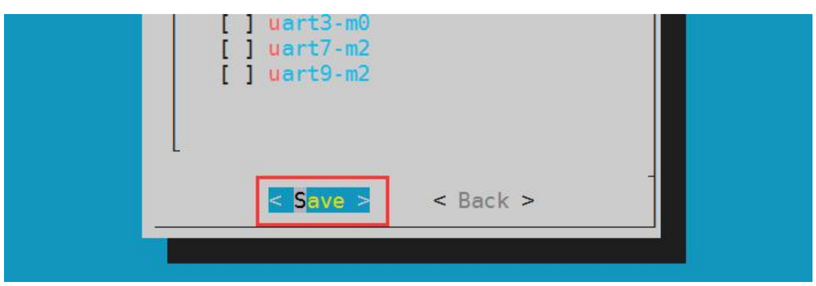
c. Then select **Hardware**



d. Then use the arrow keys of the keyboard to navigate to raspi-7inch-touchscreen, and then use the **space** to select

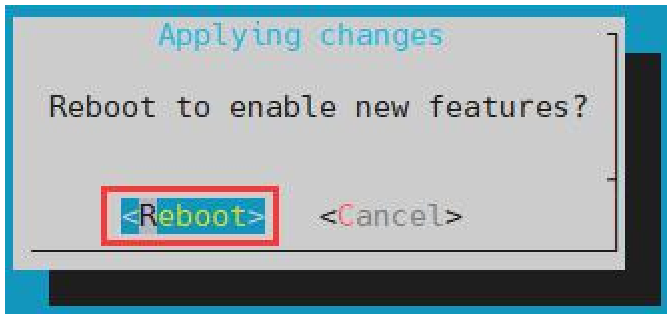


e. Then select **<Save>** to save



f. Then select **<Back>**

g. Then select **<Reboot>** to restart the system 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 lcd screen display 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

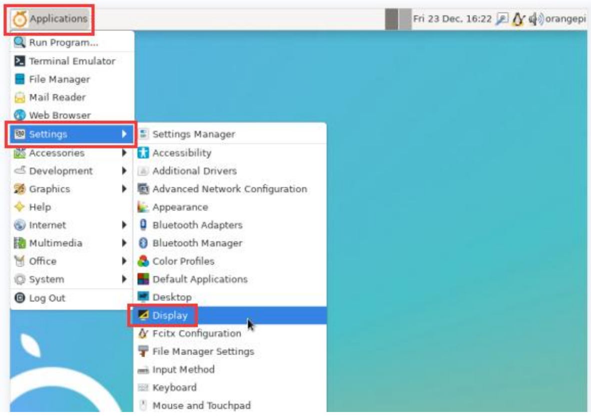
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=lcd1
extraargs=cma=64M fbcon=rotate:3
```

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

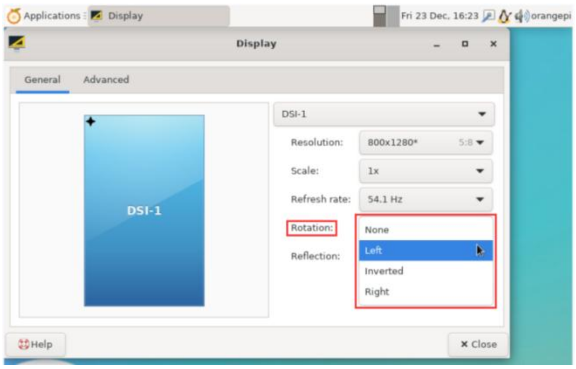
3. 29. 4. Method of rotating display and touch direction of desktop version image

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

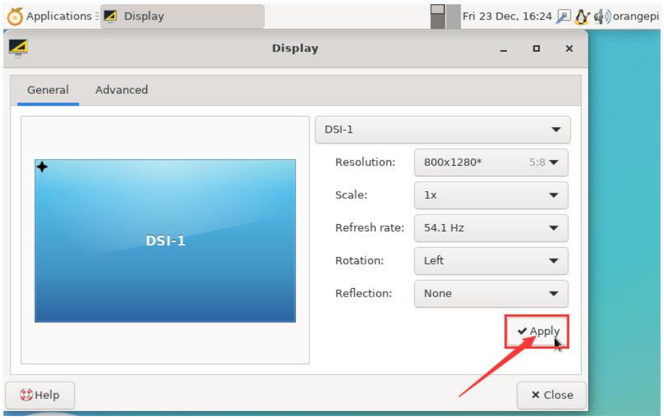


2) Then select the direction you want to rotate in **Rotation**

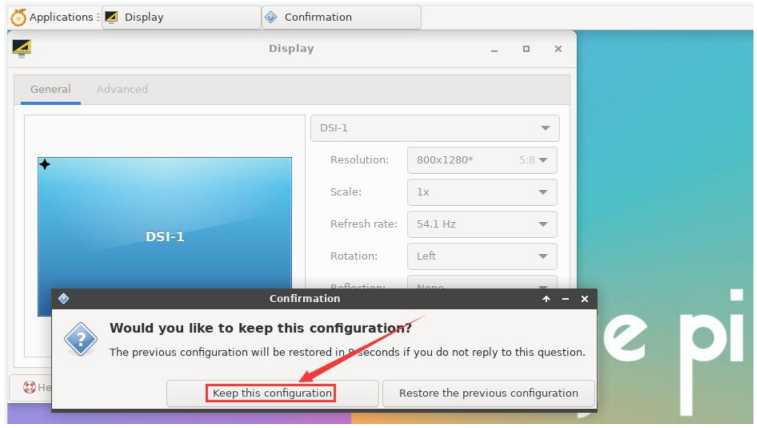
   a. **None**: no rotation
   b. **Left**: rotate left 90 degrees
   c. **Inverted**: Flip up and down, which is equivalent to rotating 180 degrees
   d. **Right**: rotate right 90 degrees



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

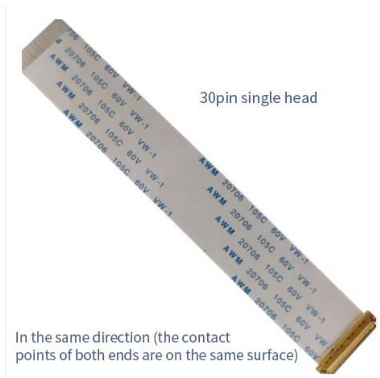
https://wiki.ubuntu.com/X/InputCoordinateTransformation

3. 30. How to use the eDP screen

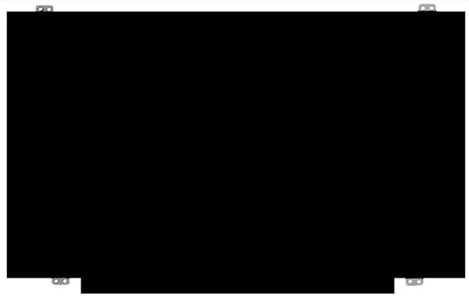
3. 30. 1. Assembly method of eDP screen

Note that linux6.6 is not supported yet.

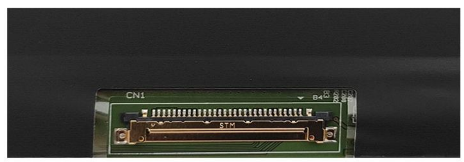
1) Currently only one eDP screen is compatible, including the following accessories:
   a. 0.5 pitch 30pin single-head cable in the same direction



   b. eDP display


front


back


30pins EDP interface

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.

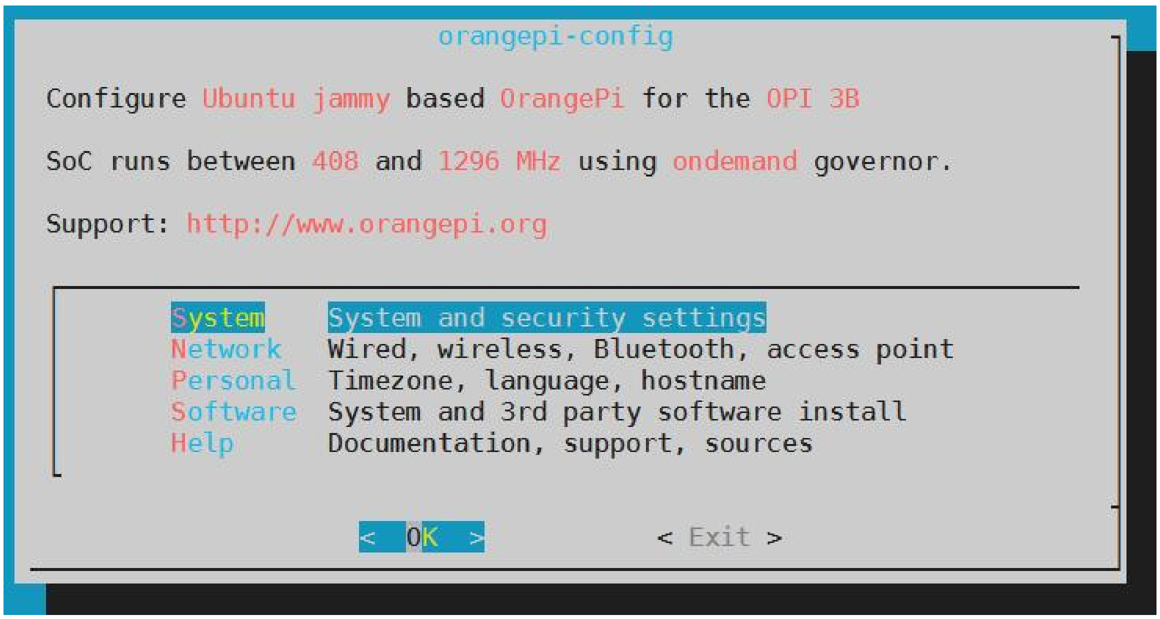
1) By default, the Linux image does not have the configuration to open the eDP screen. If you need to use the eDP screen, you need to open it manually.

2) The steps to open the eDP screen configuration are as follows:

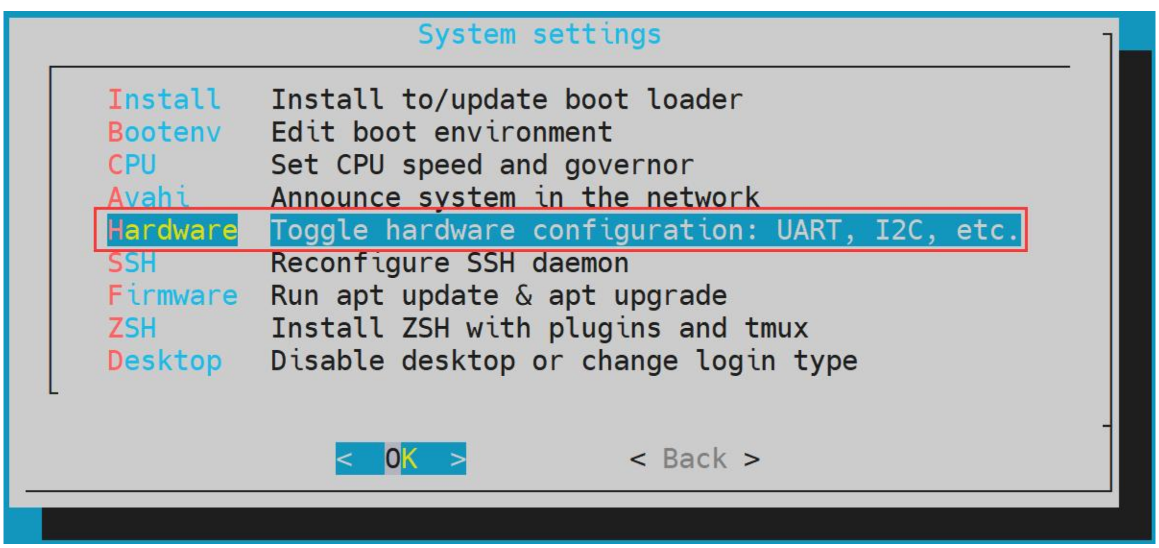
a. First run **orangepi-config**, ordinary users remember to add **sudo** permission

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

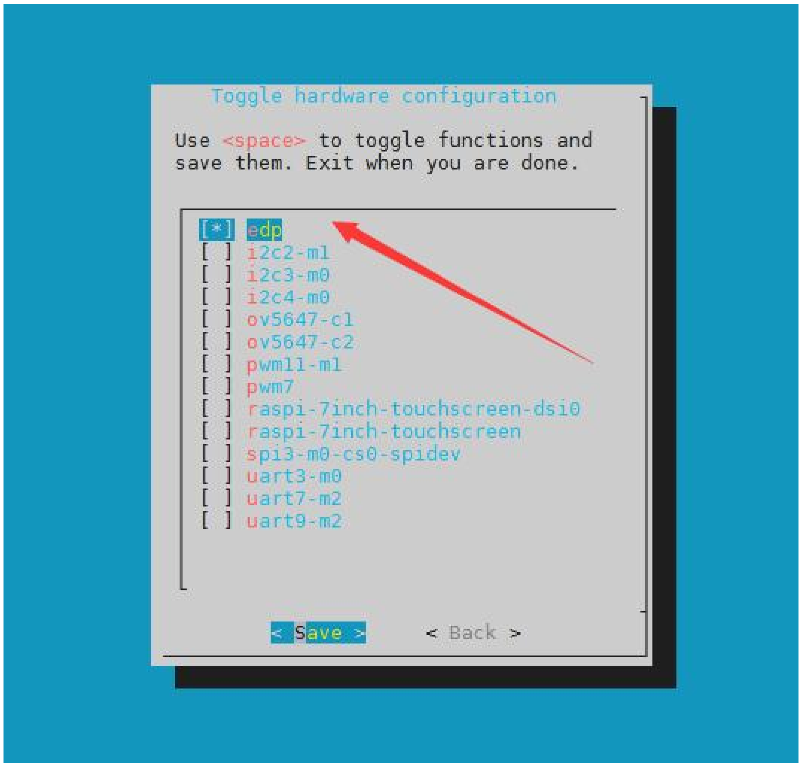
b. Then select **System**



c. Then select **Hardware**



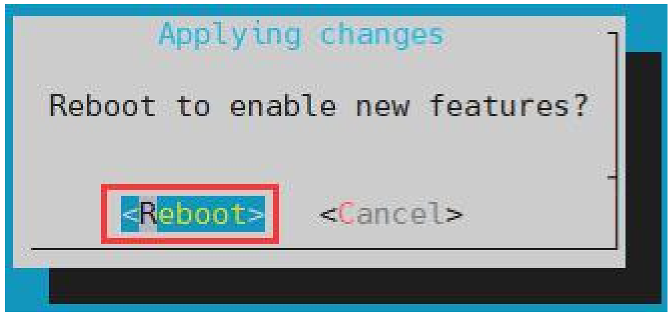
d. Then use the arrow keys on the keyboard to navigate to edp, and then use the **space** to select



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

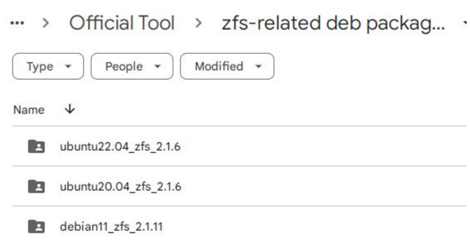
Note that linux6.6 is not supported yet.

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

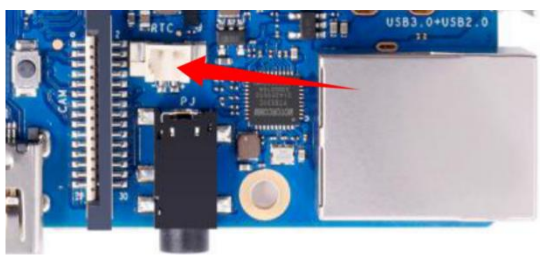
root@orangepi:/pool1$ **sudo tar -cf text.tar /var/log/ /etc/**

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 use RTC

1) An RTC battery interface is reserved on the development board. The location is as follows:：



2) The RTC battery that needs to be purchased is as follows, the interface is 2pin, 1.25mm pitch

3) The RTC chip used on the development board is HYM8563TS. The characteristics of this chip are:

a. Wide operating voltage range: 1.0~5.5v

b. Low sleep current: typical value is 0.25 µ A (VDD =3.0V, TA =25° C)

4) After connecting the RTC battery to the development board, use the following method to test whether the RTC is working normally:

a. First boot into the system, and then record the current time of the system.

b. Then use the poweroff command to shut down the system gracefully

c. Then unplug the power, make sure the development board is not connected to network cables and wireless WIFI, and wait a few more minutes.

d. Then start the system again. If you see that the time has moved forward a few minutes after entering the system, it means

The RTC module and battery are working normally

5) The command to view RTC information through the procfs interface of the Linux system is

```
orangepi@orangepi:~$ cat /proc/driver/rtc
rtc_time            : 10:10:36
rtc_date            : 2023-10-19
alrm_time           : 00:00:00
alrm_date           : 1999-12-16
alarm_IRQ           : no
alrm_pending        : no
update IRQ enabled        : no
periodic IRQ enabled      : no
periodic IRQ frequency    : 1
max user IRQ frequency    : 64
24hr                : yes
```

3. 34. Testing method of GPU in Linux6.6 system

Note that the desktop wallpaper may display a black screen after the GPU is turned on, so the GUP is turned off by default.

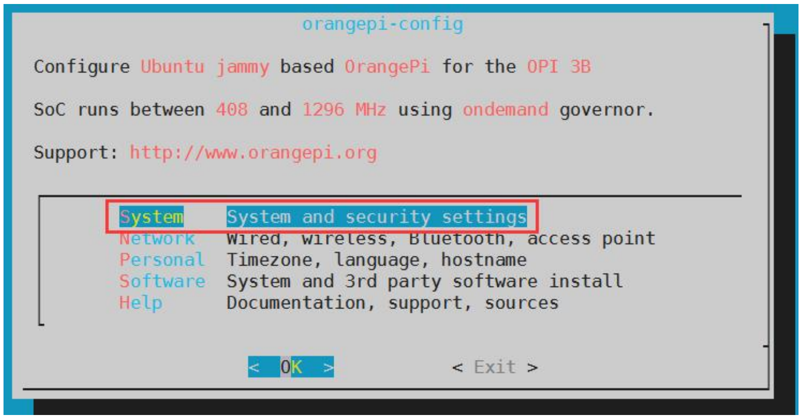
GPU is only available in Ubuntu22.04 and Debian12.

1) In Linux 6.6 system, the GPU is turned off by default and needs to be turned on manually before it can be used. Detailed steps are as follows:

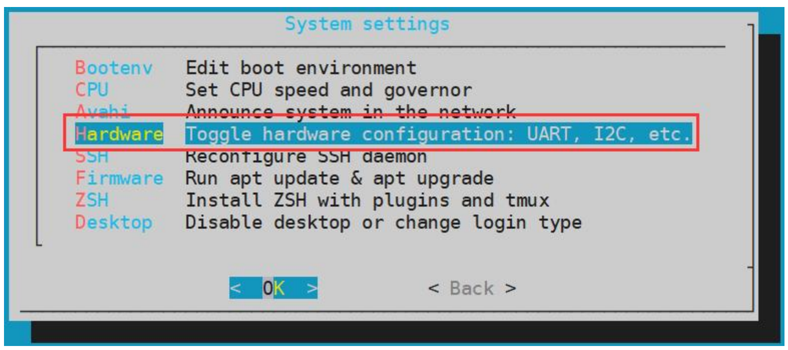
a. First run **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

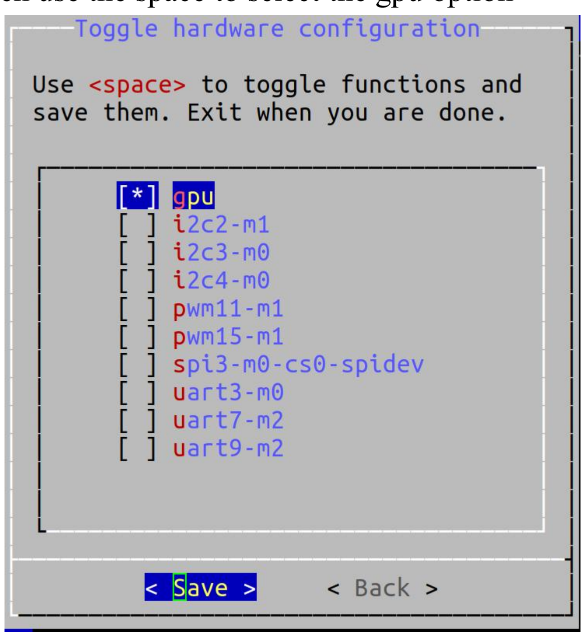
b. Then select System



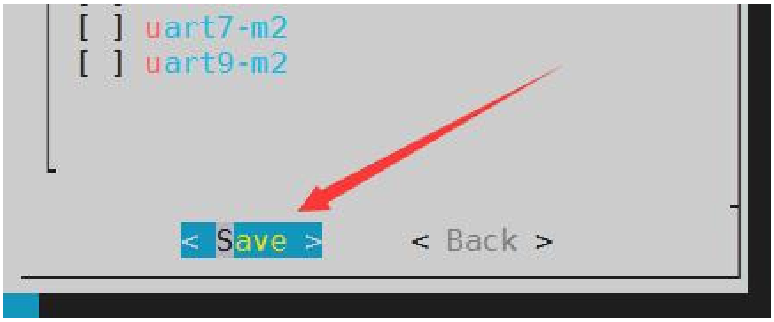
c. Then select Hardware



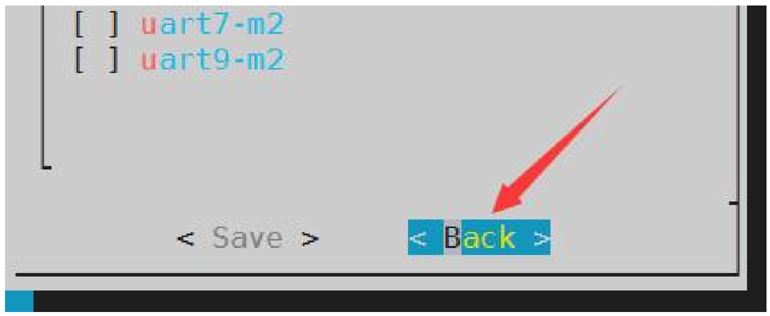
d. Then use the keyboard' s arrow keys to locate the location shown in the picture below, and then use the space to select the gpu option



e. Then select <Save> to save



f. Then select <Back>



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

2) Open a terminal on the desktop and enter the glmark2 command. If you can see GL_VENDOR followed by **Panfrost**, it means a GPU is used.

```
orangepi@orangepi:~$ glmark2
=======================================================
    glmark2 2023.01
=======================================================
    OpenGL Information
    GL_VENDOR:      Panfrost
    GL_RENDERER:    Mali-G52 r1 (Panfrost)
    GL_VERSION:     3.1 Mesa 22.3.6
    Surface Config: buf=32 r=8 g=8 b=8 a=8 depth=24 stencil=0 samples=0
    Surface Size:   800x600 windowed
=======================================================
```

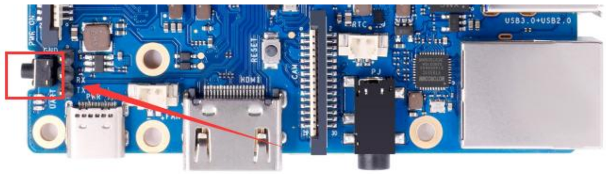
3. 35. How to shut down and restart the development board

1) During the running of the Linux system, if you directly unplug the Type-C power supply and cut off the power, the file system may lose some data or be damaged.

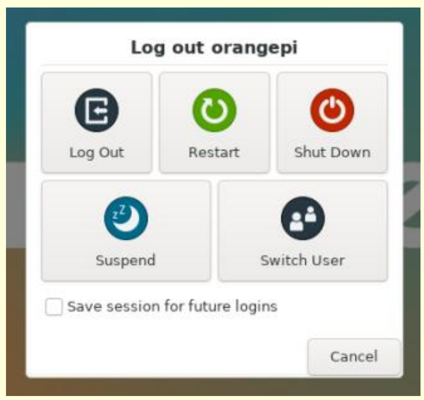
Therefore, please use the **poweroff** command to shut down the Linux system of the development board before cutting off the power. Then unplug the power supply.

```
orangepi@orangepi:~$ sudo poweroff
```

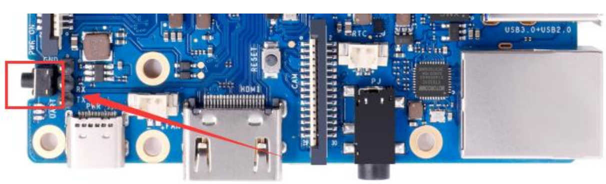
2) In addition, the development board is equipped with a power on/off button, and you can also **short press** the power on/off button on the development board to shut down.



Note that after pressing the power button on the Linux desktop version, the confirmation box shown in the figure below will pop up. You need to click the Shut Down option before Shut Down.



3) After shutting down, short press the power button on the development board to turn it on.



4) The command to restart the Linux system is

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

4. 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.

```
orangepi@orangepi:~$ lsb_release -a
No LSB modules are available.
Distributor ID:	Ubuntu
Description:	Ubuntu 22.04.1 LTS
Release:	22.04
Codename:	jammy
```

2) Since the source codes such as the kernel and U-boot are stored on GitHub, it is very important to ensure that the development board can download codes from GitHub normally when compiling the image.

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
```

4.1.2. Compile with x64 Ubuntu22.04 computer

1) The Linux SDK, **orangepi-build**, supports running on computers with **Ubuntu 22.04** installed, so before downloading orangepi-build, please make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on the computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please 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 **VMware** 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
```

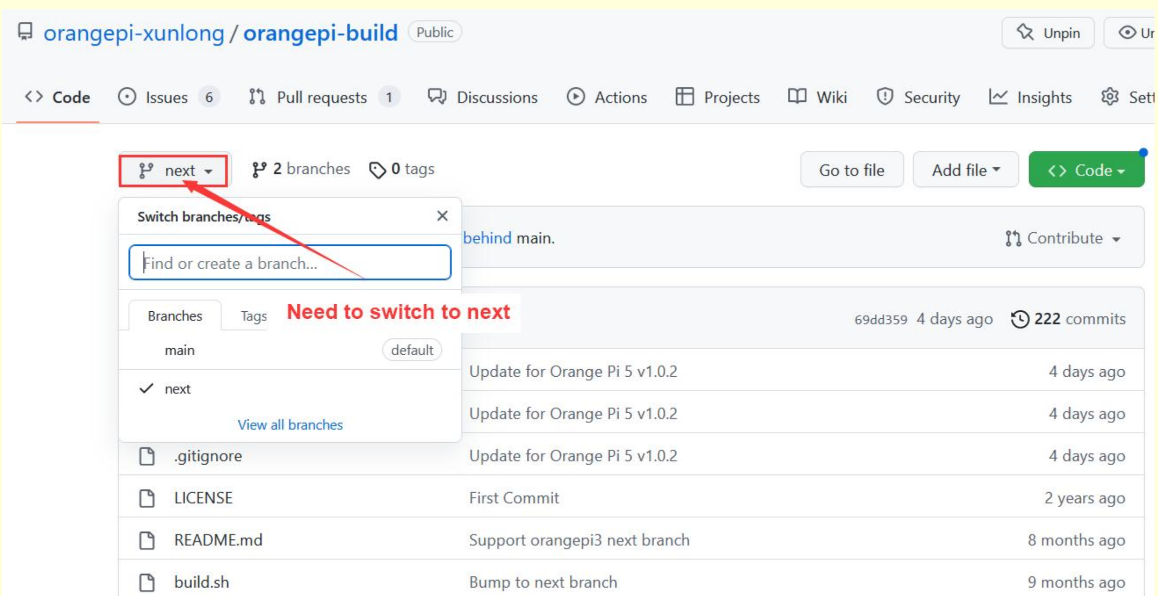
4. 2. Get the source code of Linux sdk

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
current	u-boot 2017.09	Linux6.6

The branch mentioned here is not the same thing as the branch of orangepi-build source code, please don't get confused. This branch is mainly used to distinguish different kernel source code versions.

We define the Linux5.10 bsp kernel currently provided by RK as the legacy branch. We define the latest Linux6.6 kernel as the current branch.

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

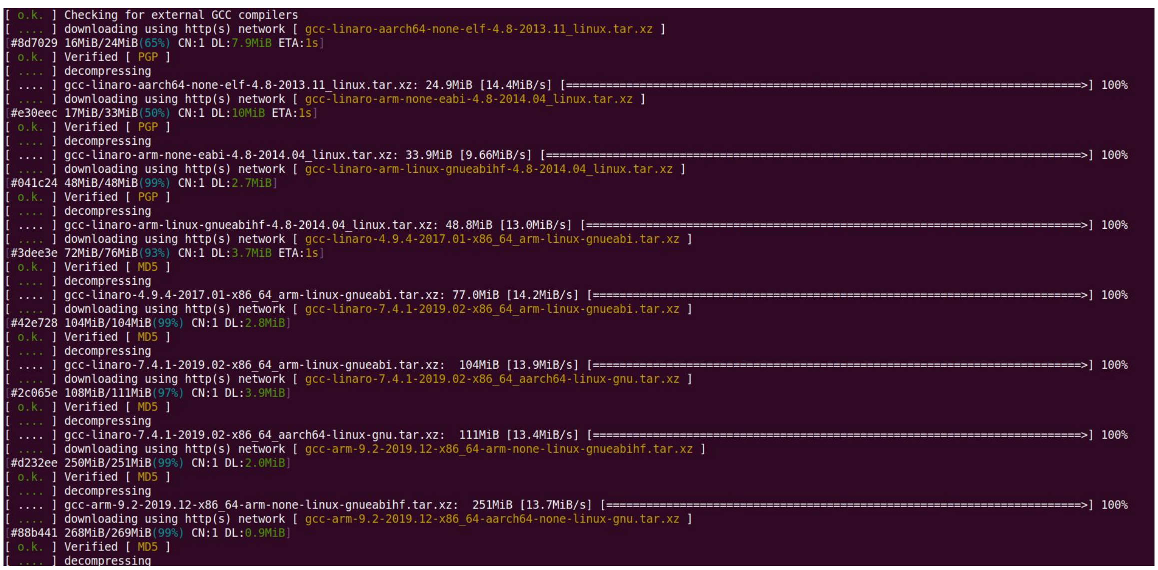
```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

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) After downloading, the orangepi-build warehouse does not contain the source code of the Linux kernel, u-boot and cross-compilation tool chain. The source code of the Linux kernel and u-boot is stored in an independent git warehouse.

a. The git warehouse where the Linux kernel source code is stored is as follows. Please note that the branch of the linux-orangepi warehouse is switched to

a) Linux5.10

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk35xx
```

b) Linux6.6

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-6.6-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. 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. The folder named orange-pi-6.6-rk35xx stores the source code. It is the kernel source code of the current branch of the RK3566 series development board. Please do not modify the name of the folder of the kernel source code manually. If it is 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

```
Choose an option
Compile image | rootfs | kernel | u-boot

          U-boot package
          Kernel package
          Rootfs and all deb packages
          Full OS image for flashing
```

3) Then select the model of the development board

```
Choose an option
Please choose a Board.

  orangepi3-lts    Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
  orangepizero2    Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
  orangepizero3    Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
  orangepizero2w   Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
  orangepi4        Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
  orangepi4-lts    Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
  orangepi800      Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
  orangepi5        Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
  orangepi5b       Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT eMMC
  orangepi5plus    Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C WiFi/BT NVMe eMMC
  orangepicm4      Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
  orangepi3b       Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT

                 <Select>                    <Exit>
```

4) Then select u-boot to select the branch type. Both the current branch and the legacy branch will compile the code of the u-boot v2021.07 version that needs to be used.

```
Choose an option
Select the target kernel branch

      current Recommended. Come with best support
      legacy  Old stable / Legacy
```

5) Then it will start to compile u-boot. Some of the information prompted during compilation is as follows:

a. u-boot source code version

[o.k.] Compiling u-boot [**v2017.09**]

b. 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 ]
```

6) 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
```

7) 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
```

8) 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"
```

9) 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
```

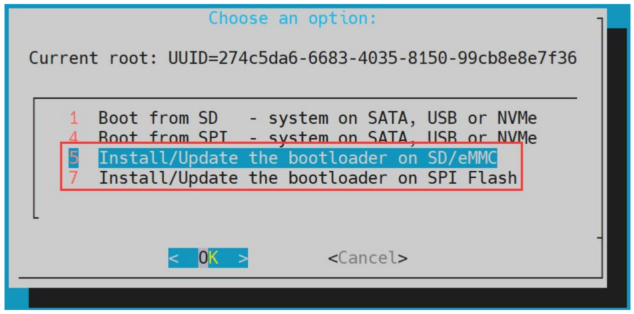
c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi3b_1.0.0_arm64.deb
```

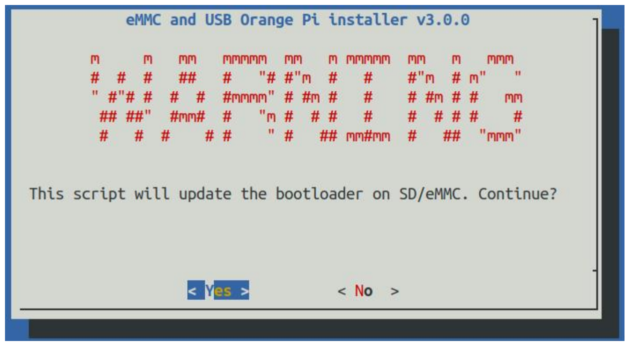
d. Then run the nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

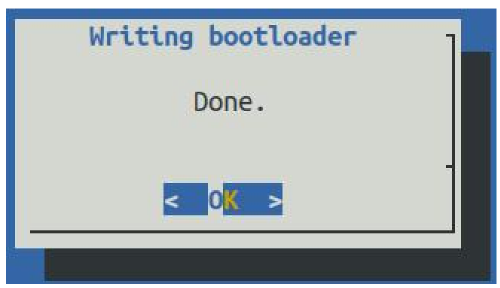
e. Then select **5 Install/Update the bootloader on SD/eMMC** to update the u-boot in the TF card or **7 Install/Update the bootloader on SPI Flash** to update the u-boot in the SPI Flash



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

10) 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

```
Choose an option
Compile image | rootfs | kernel | u-boot

                U-boot package
                Kernel package
                Rootfs and all deb packages
                Full OS image for flashing
```

3) Then select the model of the development board

```
Choose an option
Please choose a Board.

    orangepi3-lts  Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
    orangepizero2  Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
    orangepizero3  Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
    orangepizero2w Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
    orangepi4      Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
    orangepi4-lts  Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
    orangepi800    Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
    orangepi5      Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
    orangepi5b     Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT eMMC
    orangepi5plus  Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C WiFi/BT NVMe eMMC
    orangepicm4    Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
    orangepi3b     Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT

                  <Select>                        <Exit>
```

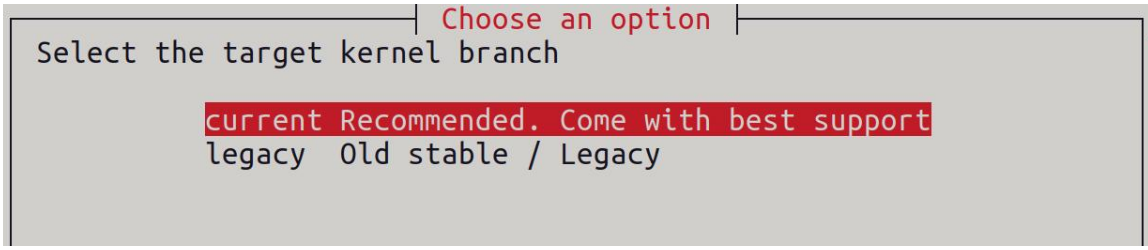
4) Then it will prompt whether to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If you need to modify the kernel configuration, select the second one.

```
Choose an option
Select the kernel configuration.

          Do not change the kernel configuration
          Show a kernel configuration menu before compilation
```

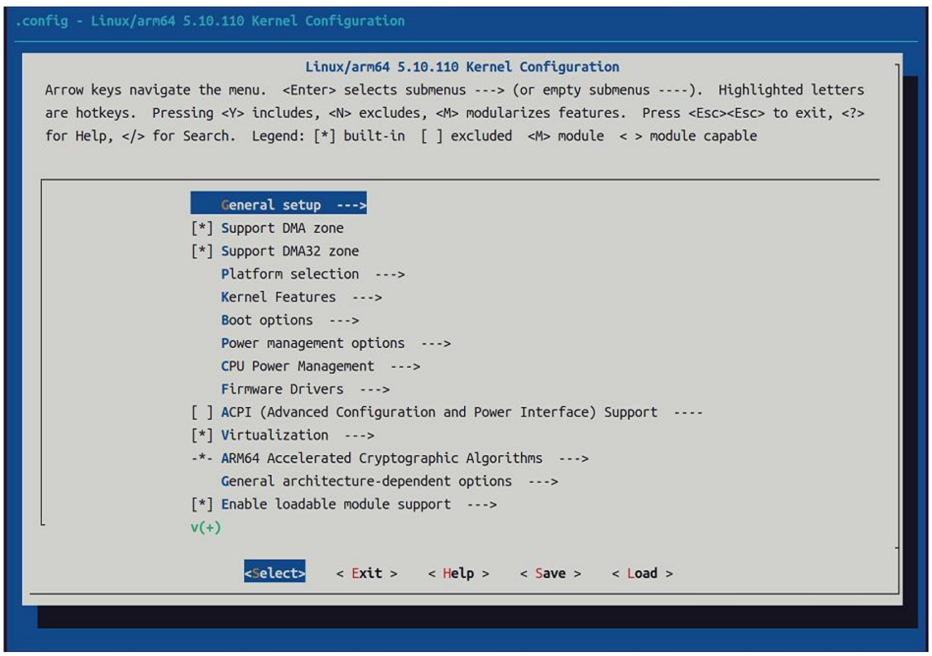
5) Then select the branch type of the kernel source code

a. The legacy branch will compile the linux5.10 kernel source code

b. The current branch will compile the linux6.6 kernel source code



6) If you choose to display the kernel configuration menu (the second option) in step 4, the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration, save and exit after modification. Yes, after exiting, the kernel source code will be compiled



a. If you do not need to modify the configuration options of the kernel, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily block the pop-up kernel configuration interface

```
test@test:~/orangepi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

b. You can also set **KERNEL_CONFIGURE=no** in the **orangepi-build/userpatches/config-default.conf** configuration file, which can permanently disable this function

c. If the following error is displayed when compiling the kernel, it is because the terminal interface of the Ubuntu PC is too small to display the **make menuconfig** interface. Please maximize the terminal of the Ubuntu PC and run the build.sh script again

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

7) Part of the information prompted when compiling the legacy branch kernel source code is explained 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**]

8) 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
```

9) 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
```

10) 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"
```

11) If the kernel has been modified, the following method can be used to update the kernel and kernel modules of the development board Linux system

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
```

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

```
Choose an option
Compile image | rootfs | kernel | u-boot

U-boot package
Kernel package
Rootfs and all deb packages
Full OS image for flashing
```

3) Then select the model of the development board

```
Choose an option
Please choose a Board.

orangepi3-lts    Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
orangepizero2    Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
orangepizero3    Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
orangepizero2w   Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
orangepi4        Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi4-lts    Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi800      Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
orangepi5        Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
orangepi5b       Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT eMMC
orangepi5plus    Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C WiFi/BT NVMe eMMC
orangepicm4      Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
orangepi3b       Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT

<Select>                    <Exit>
```

4) Then select the branch type of the kernel source code. Currently, the rootfs maintained by the kernel source code uses the same set.

```
Choose an option
Select the target kernel branch

current Recommended. Come with best support
legacy  Old stable / Legacy
```

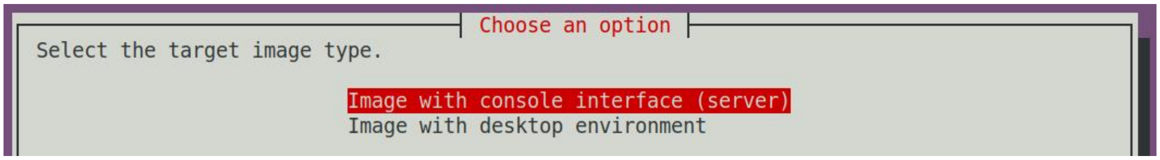
5) Then select the type of rootfs

```
Choose a release package base
Select the target OS release package base

bullseye Debian 11 Bullseye
jammy    Ubuntu jammy 22.04 LTS
```

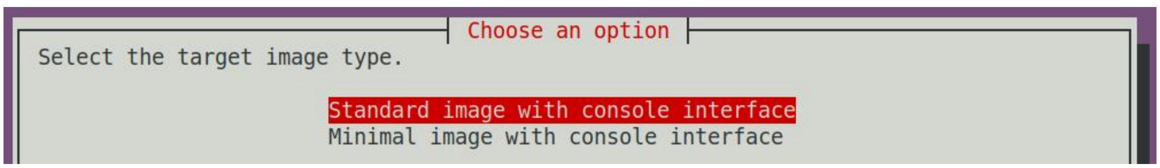
6) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small

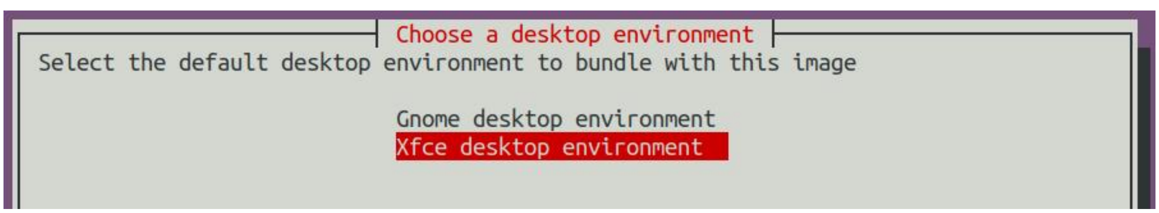
b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

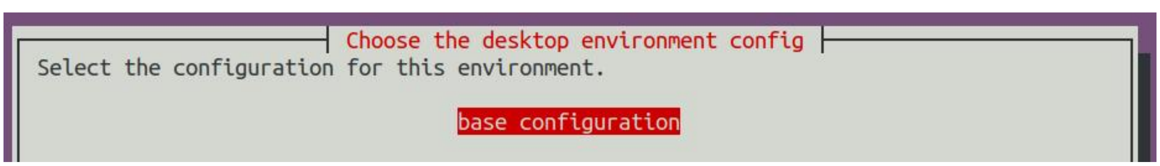


7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**

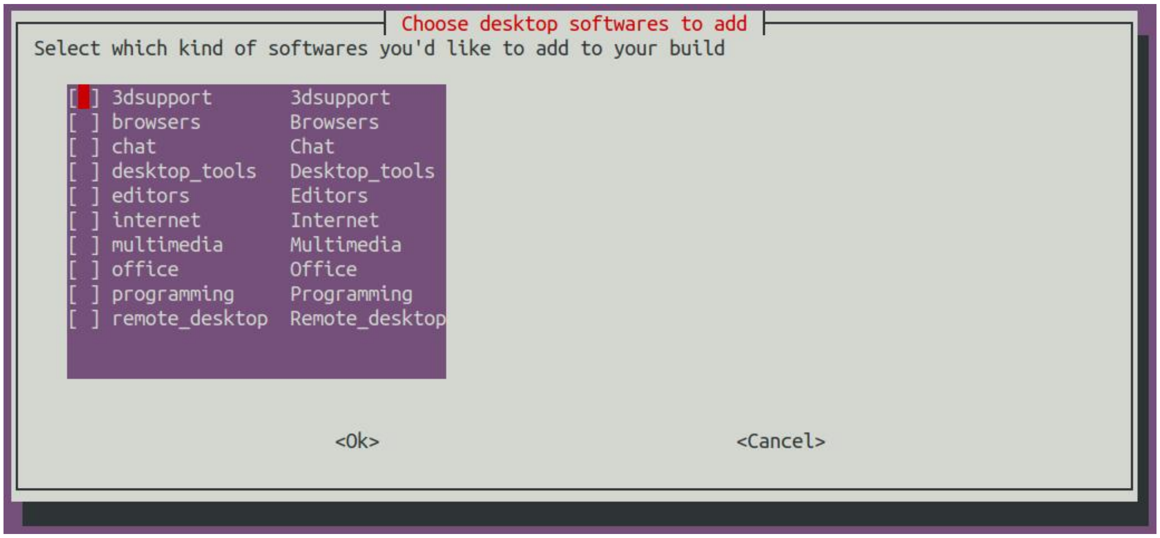


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, and Debian Bullseye mainly maintains XFCE and KDE desktops





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



9) Then it will start to compile rootfs, and some of the information prompted during compilation are as follows

a. The type of rootfs

[o.k.] local not found [Creating new rootfs cache for **jammy**]

b. The storage path of the compiled rootfs compressed package

[o.k.] Target directory [**external/cache/rootfs**]

c. The name of the rootfs compressed package generated by compilation

[o.k.] File name [**jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**]

d. The time used for compilation

[o.k.] Runtime [**13 min**]

10) View the rootfs compressed package generated by compilation

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is the rootfs compressed package, the meaning of each field of the name is

a) **jammy** indicates the type of Linux distribution of rootfs

b) **xfce** means rootfs is the type of desktop version, if it is **cli**, it means the type of server version

c) **arm64** represents the architecture type of rootfs

d) **f930ff6ebbac1a72108a2e100762b18f** is the MD5 hash value generated by the package names of all software packages installed by rootfs. As long as the list of software packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to generate Determine whether rootfs needs to be recompiled

b. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.current
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list
```

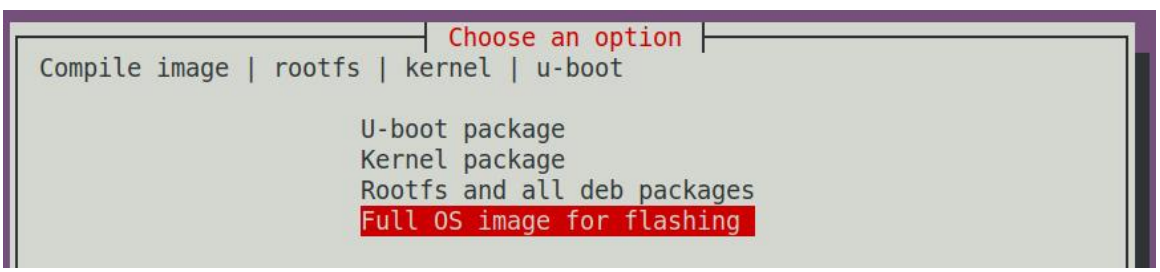
11) If the required rootfs already exists under **external/cache/rootfs**, then compiling rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has If there is rootfs available in the cache, use it directly, which can save a lot of download and compilation time.

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

```
Choose an option
Please choose a Board.

    orangepi3-lts  Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
    orangepizero2  Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
    orangepizero3  Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
    orangepizero2w Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
    orangepi4      Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
    orangepi4-lts  Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
    orangepi800    Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
    orangepi5      Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
    orangepi5b     Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT eMMC
    orangepi5plus  Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C WiFi/BT NVMe eMMC
    orangepicm4    Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
    orangepi3b     Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT

                  <Select>                              <Exit>
```

4) Then select the branch type of the kernel source code
    a. The legacy branch will compile the linux5.10 kernel source code
    b. The current branch will compile the linux6.6 kernel source code

```
Choose an option
Select the target kernel branch

         current Recommended. Come with best support
         legacy  Old stable / Legacy
```

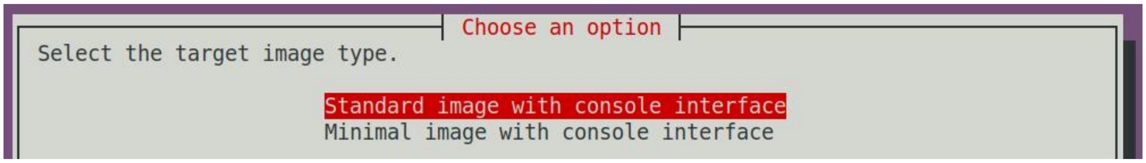
5) Then select the type of rootfs

```
Choose a release package base
Select the target OS release package base

         bullseye Debian 11 Bullseye
         jammy    Ubuntu jammy 22.04 LTS
```

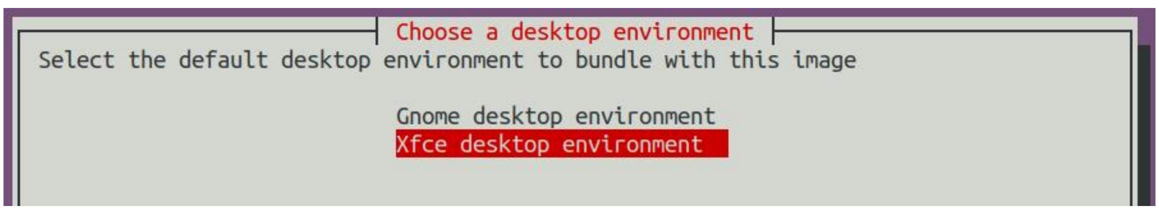
6) Then select the type of image
    a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small
    b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

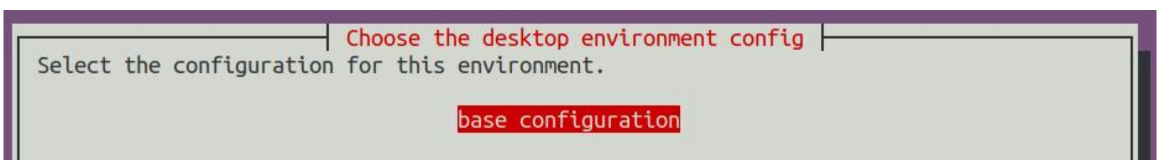
```
Choose an option
Select the target image type.

         Image with console interface (server)
         Image with desktop environment
```

7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**

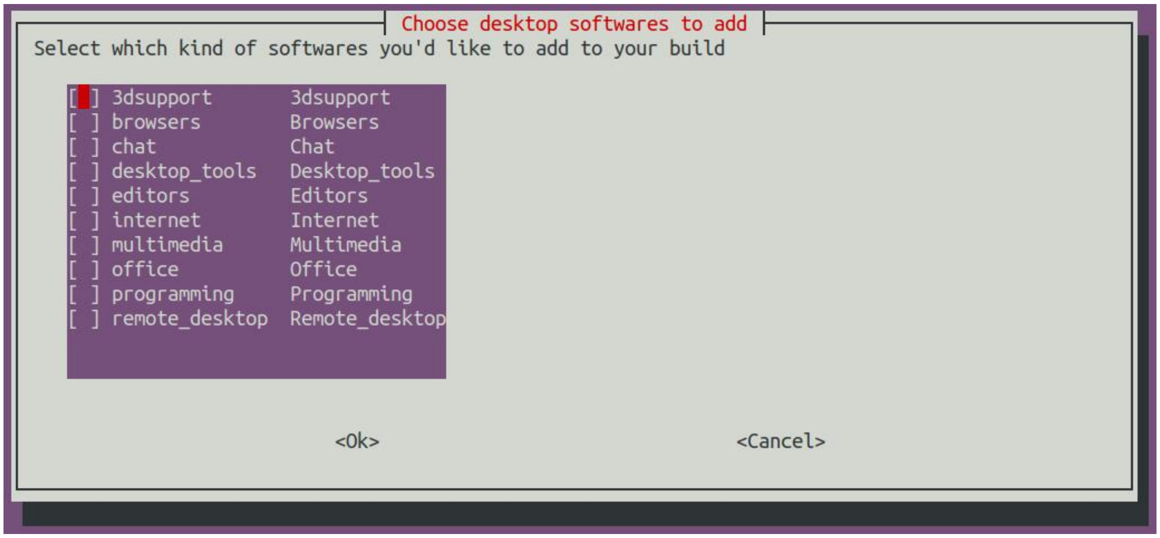


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, and Debian Bullseye mainly maintains XFCE and KDE desktops





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



9) Then it will start to compile the Linux image. The general process of compilation is as follows

a. Initialize the compilation environment of Ubuntu PC and install the software packages required for the compilation process

b. Download the source code of u-boot and Linux kernel (if cached, only update the code)

c. Compile u-boot source code and generate u-boot deb package

d. Compile the Linux source code and generate Linux-related deb packages

e. Make the deb package of Linux firmware

f. Make the deb package of the orangepi-config tool

g. Create a deb package supported by the board

h. If you are compiling the desktop image, you will also create desktop-related deb packages

i. Check whether the rootfs has been cached, if not, recreate the rootfs, if it has been cached, directly decompress and use

j. Install the previously generated deb package into rootfs

k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.

l. Then make an image file and format the partition, the default type is ext4

m. Then copy the configured rootfs to the mirrored partition

n. Then update initramfs

o. Finally, write the bin file of u-boot into the image through the dd command

10) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/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
WIFI	OK
Bluetooth	OK
GPIO（40pin）	OK
UART（40pin）	OK
SPI（40pin）	OK

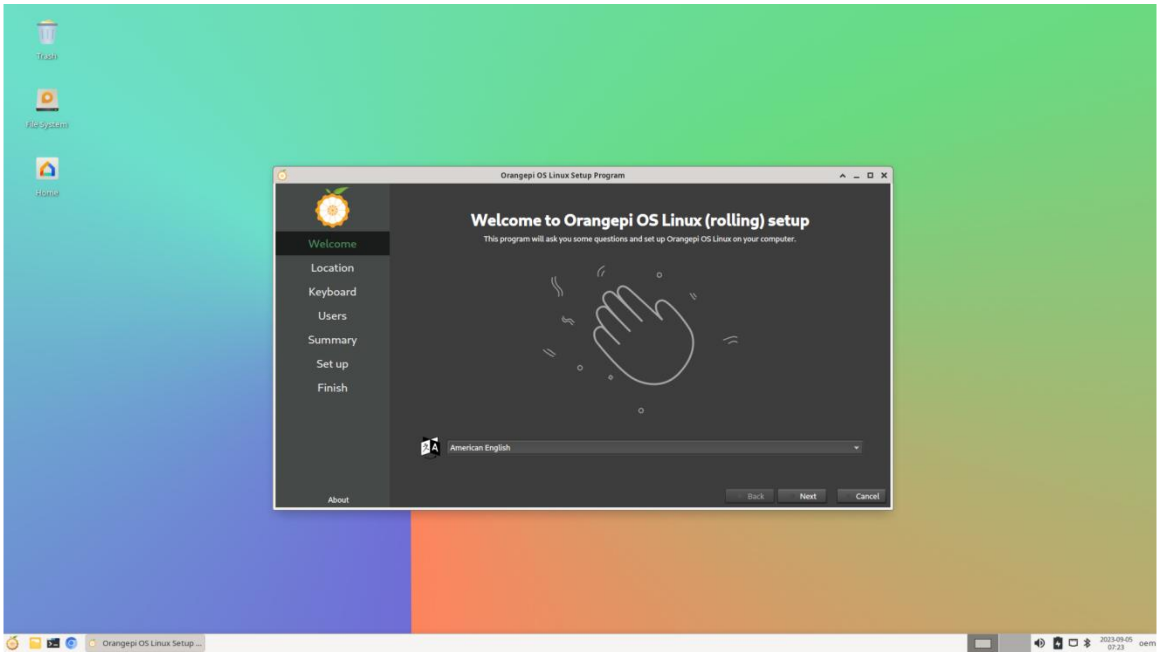
I2C (40pin)	OK
PWM (40pin)	OK
3pin debugging serial port	OK
eMMC start	OK
TF card start	OK
HDMI video	OK
HDMI audio	OK
Raspberry Pi 5 inch screen display	OK
Raspberry Pi 5-inch screen touch function	OK
eDP display	OK
OV5647 camera	The kernel driver is OK, 3A is not adjusted
Gigabit Ethernet port	OK
Network 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

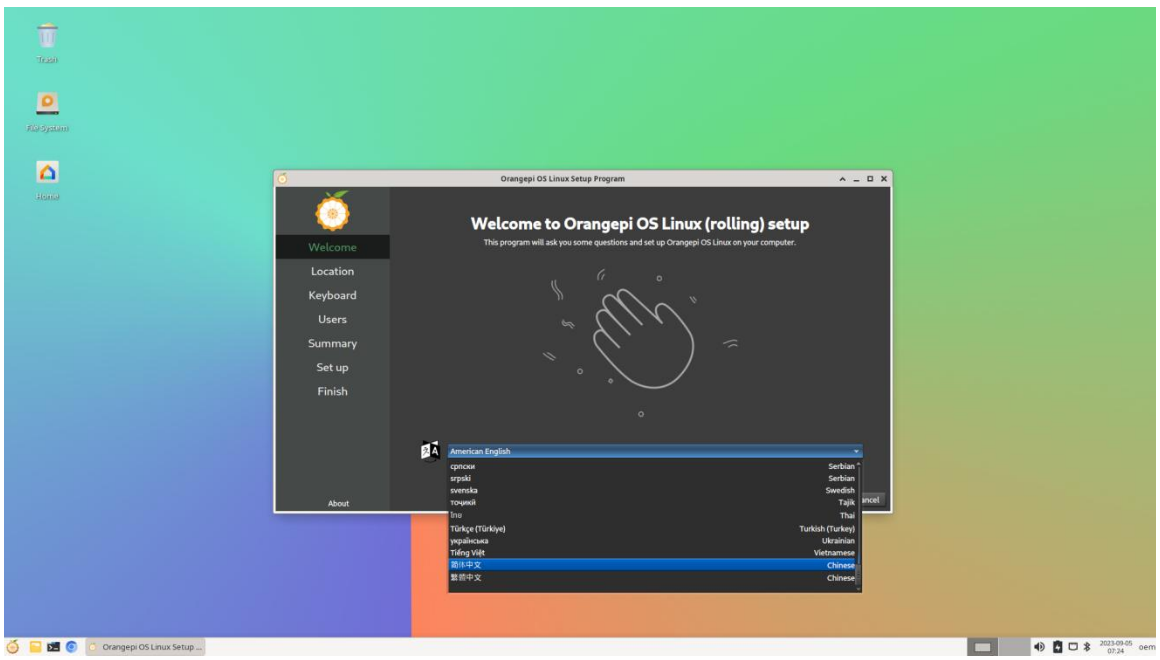
First of all, please note that the OPi OS Arch system does not have a default orangepi user and password, so after burning and starting the system, it is impossible to log in remotely through the serial port and ssh directly (not even the root user). This is different from Ubuntu and Debian systems.

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:

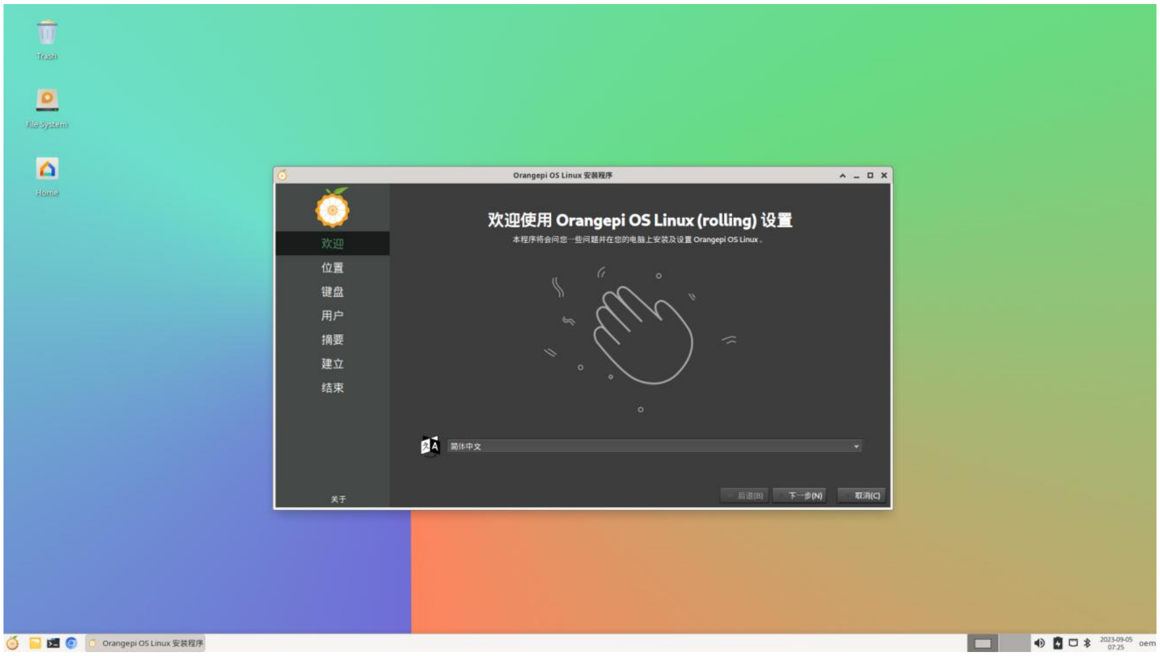
a) After burning the system for the first time and enter the desktop, you will see the user wizard program shown in the figure below



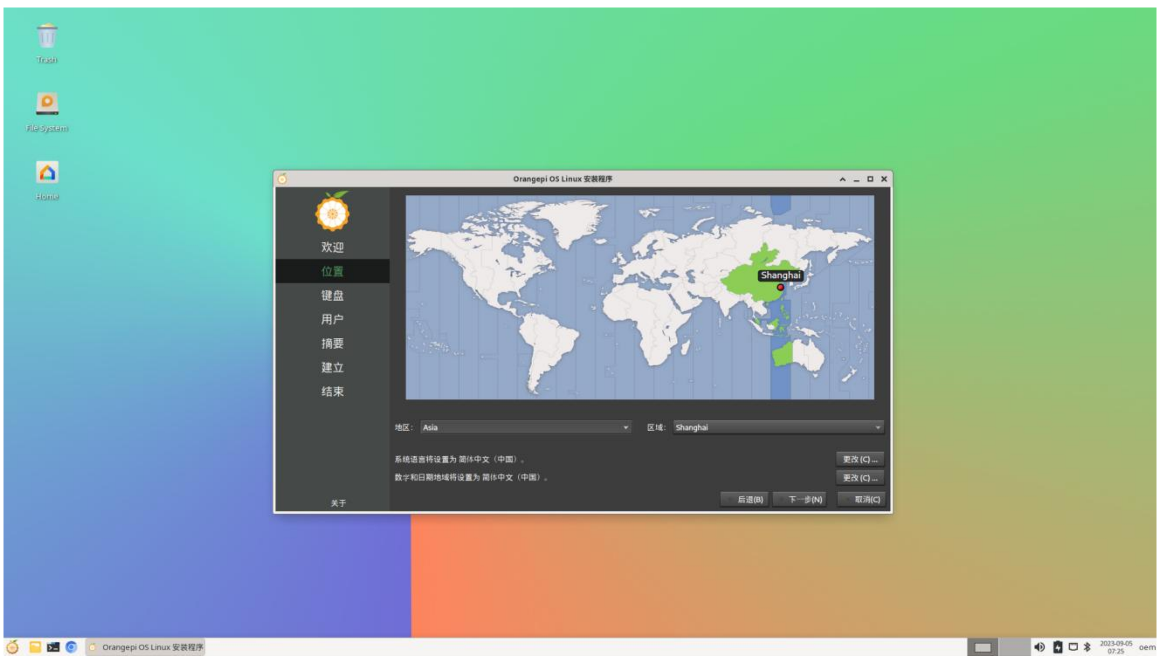
b) First you need to choose the desired language



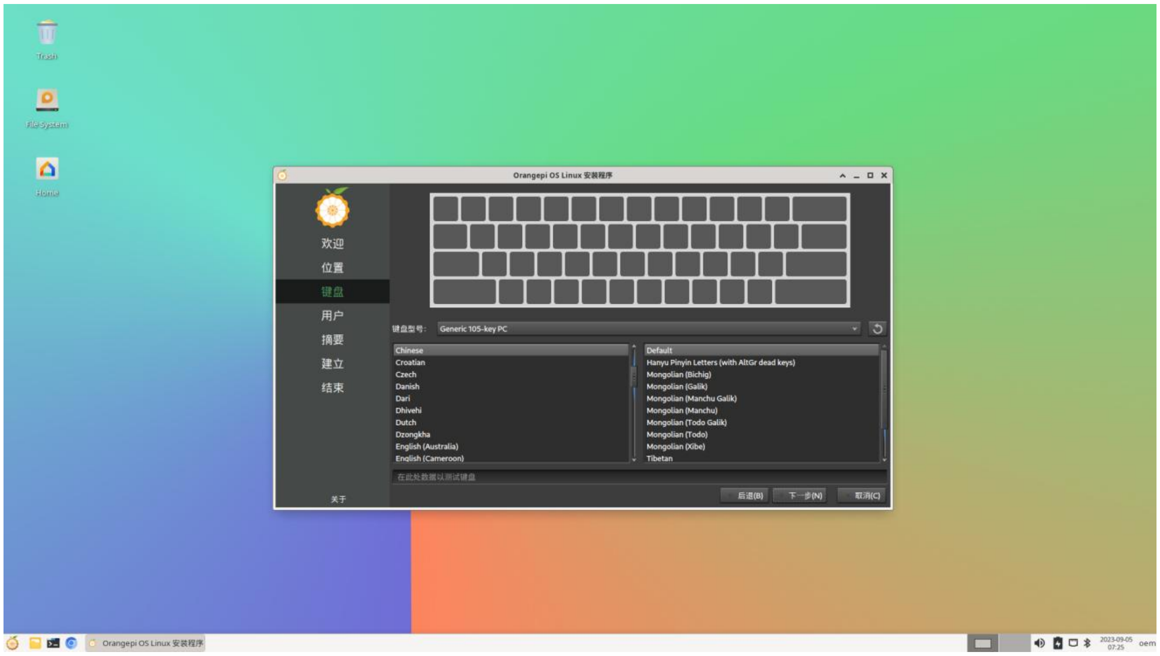
c) After selecting the language, the user guide will immediately switch to the corresponding language interface, such as the Chinese display as shown below



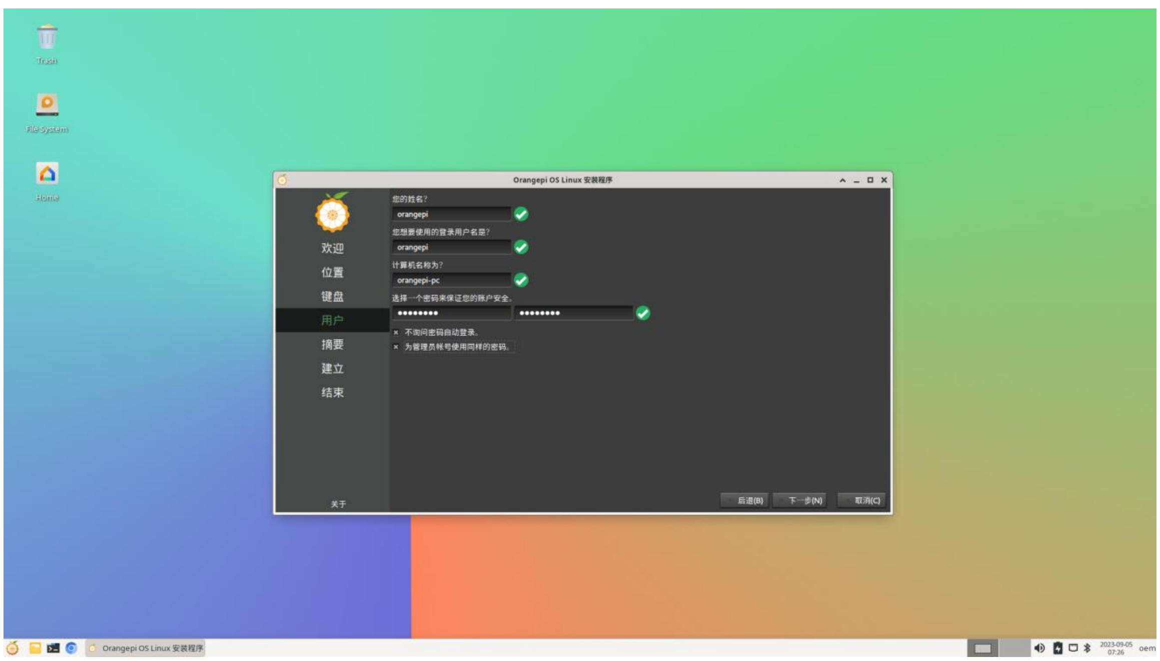
d) Then select the area



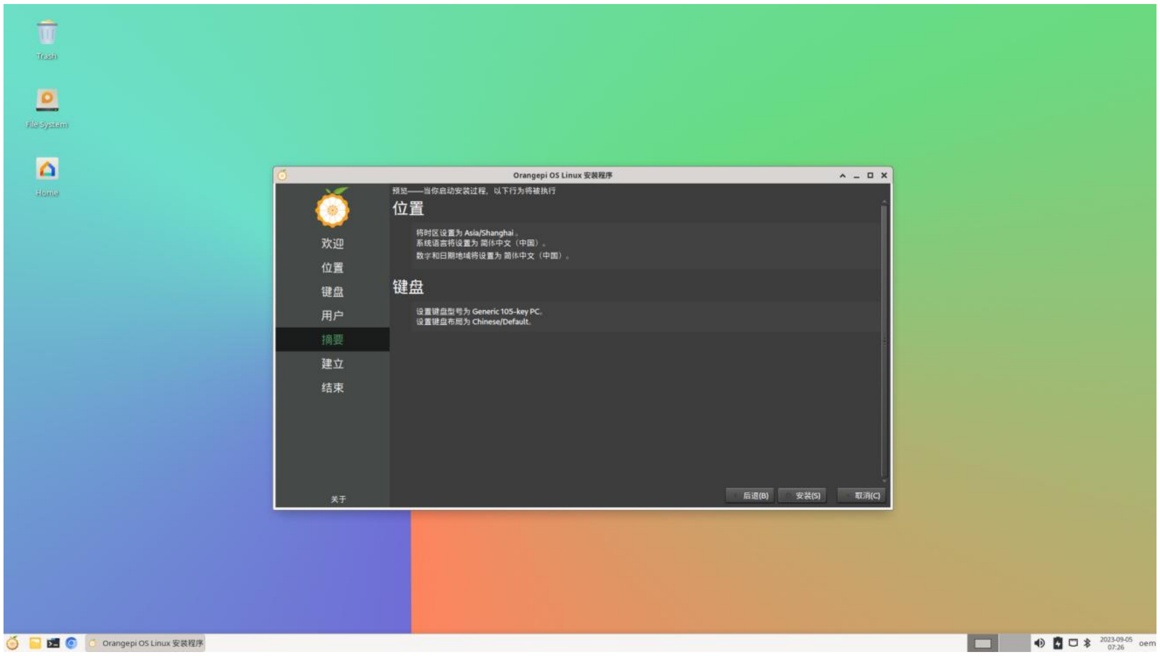
e) Then select the keyboard model



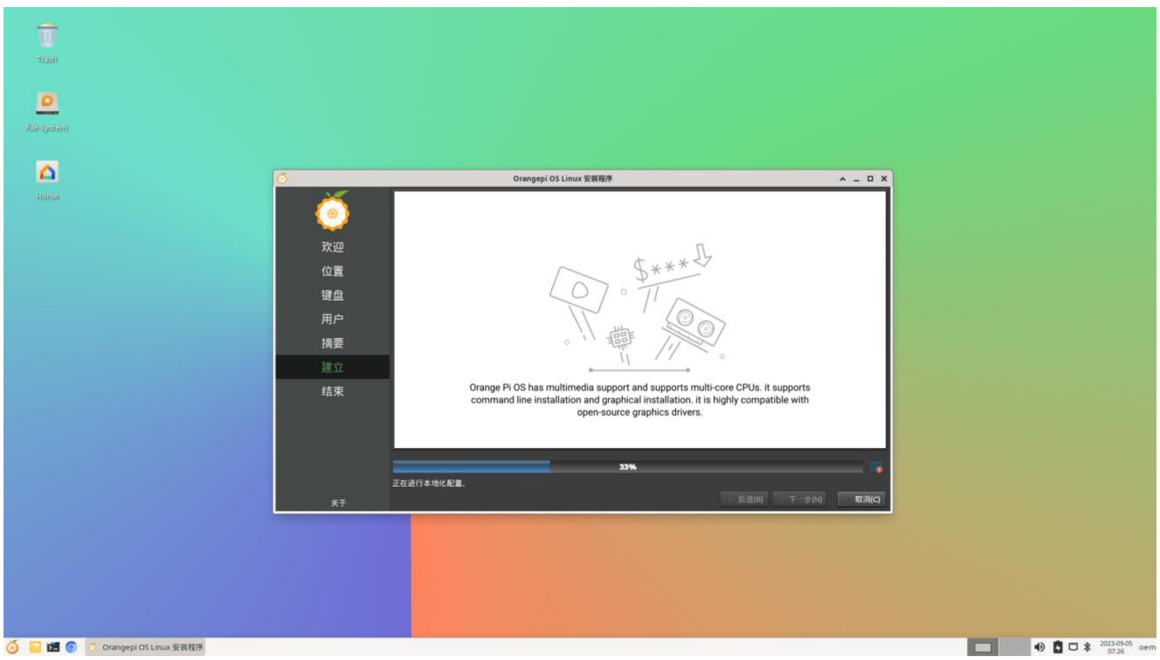
f) Then create a new user name and set a password



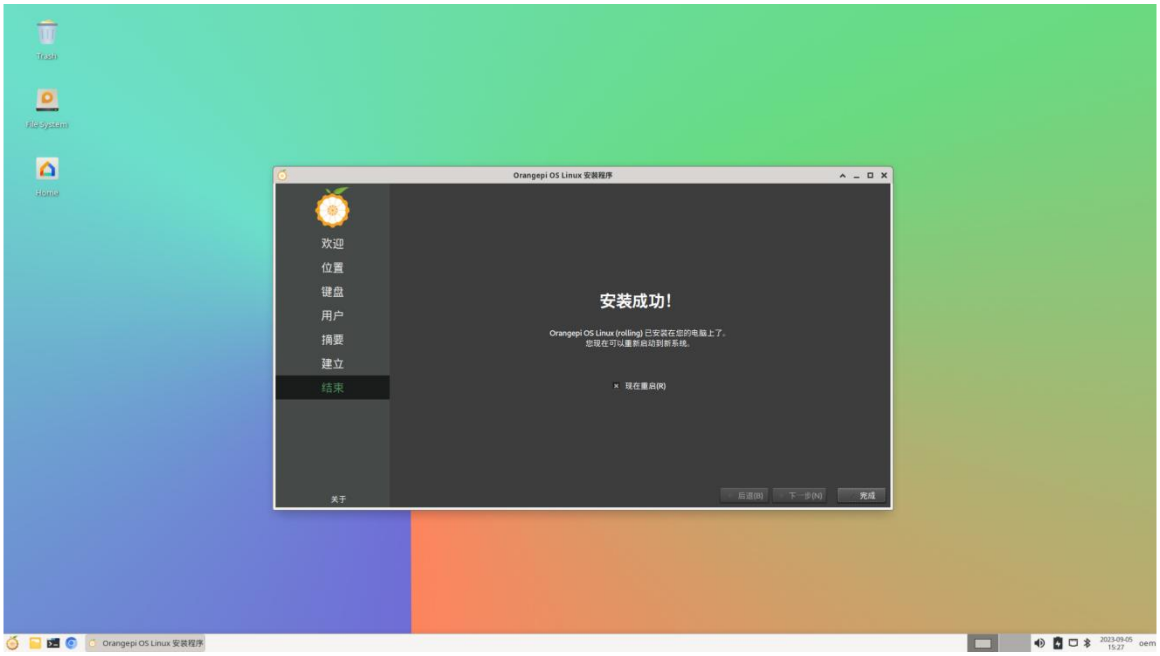
g) Then make sure that there is no problem with the selection, and then click the **install** button



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:

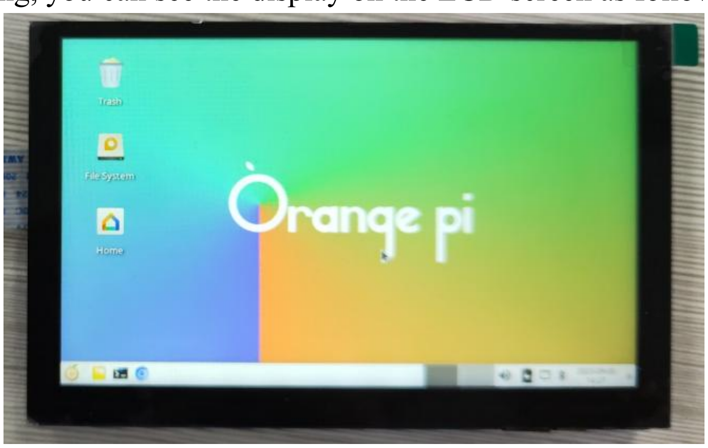
a. First add the following configuration in **/boot/extlinux/extlinux.conf**

```
[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. Orange Pi OS OH system usage instructions

6. 1. Orange Pi OS OH system function adaptation status

功能	OPi OS OH
USB2.0x3	OK
USB3.0x1	OK
SPIFlash+M.2 NVMe SSD Boot	NO
WIFI	OK
Bluetooth	NO
3pin Debug serial port	OK
eMMC start	OK
TF card start	OK
HDMI video	OK
HDMI audio	NO
Raspberry Pi 5 inch screen display	NO
Raspberry Pi 5-inch screen touch function	NO
eDP display	NO
OV5647 camera	NO
Gigabit Ethernet port	OK
Network port status light	OK
Network port status light	OK
headphone recording	NO
LED lights	OK
GPU	OK
NPU	NO
VPU	NO

Currently, the 8GB memory version of the development board can only use 4GB

of memory in the OPi OS OH system. Please pay special attention to this.

6. 2. How to use the Gigabit Ethernet port

1) First use a network cable to connect the development board and router

2) Then use the ifconfig command in the debugging serial port to see the IP address assigned by the router to the development board network port.

```
# ifconfig eth0
eth0      Link encap:Ethernet   HWaddr 4e:fc:9d:f3:67:26   Driver rk_gmac-dwmac
          inet addr:192.168.1.189   Bcast:192.168.1.255   Mask:255.255.255.0
          inet6 addr: fe80::4cfc:9dff:fef3:6726/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
          RX packets:378 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30663 TX bytes:2904
          Interrupt:45

#
```

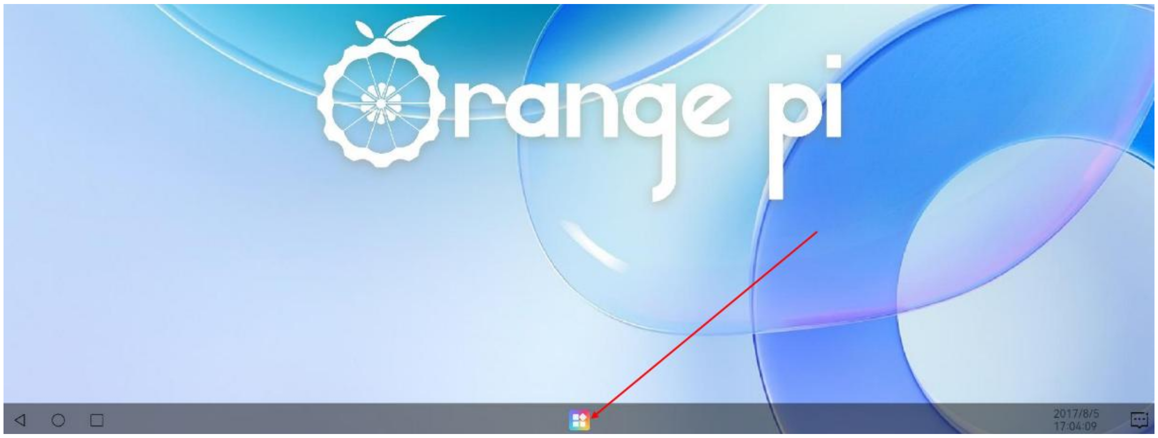
3) Then use the ping command to test whether the network can be used normally

```
# ping www.orangepi.cn
Ping www.orangepi.cn (123.57.147.237): 56(84) bytes.
64 bytes from 123.57.147.237: icmp_seq=1 ttl=0 time=42 ms
64 bytes from 123.57.147.237: icmp_seq=2 ttl=0 time=43 ms
64 bytes from 123.57.147.237: icmp_seq=3 ttl=0 time=44 ms

--- 123.57.147.237 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss
round-trip min/avg/max = 0/0/43 ms
#
```

4) In addition to checking the IP address of the network port in the command line, you can also check the IP address of the network port in the OH settings. The method is

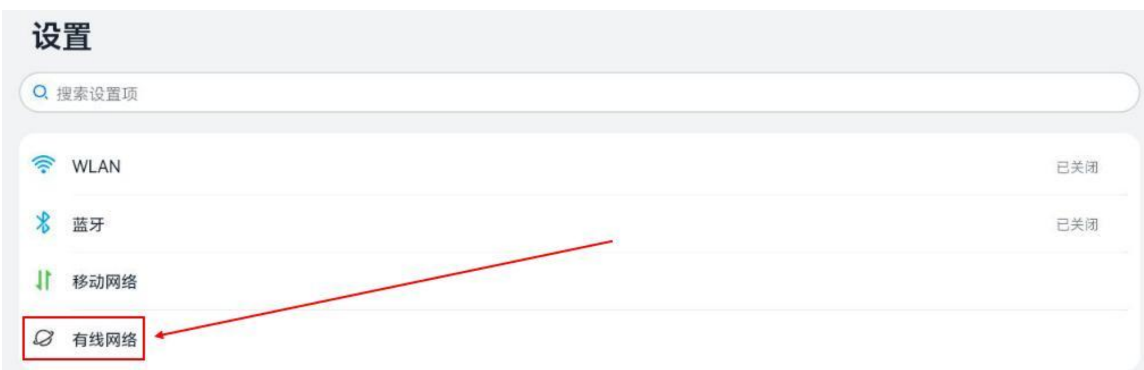
as follows

a. First click on the **application list**

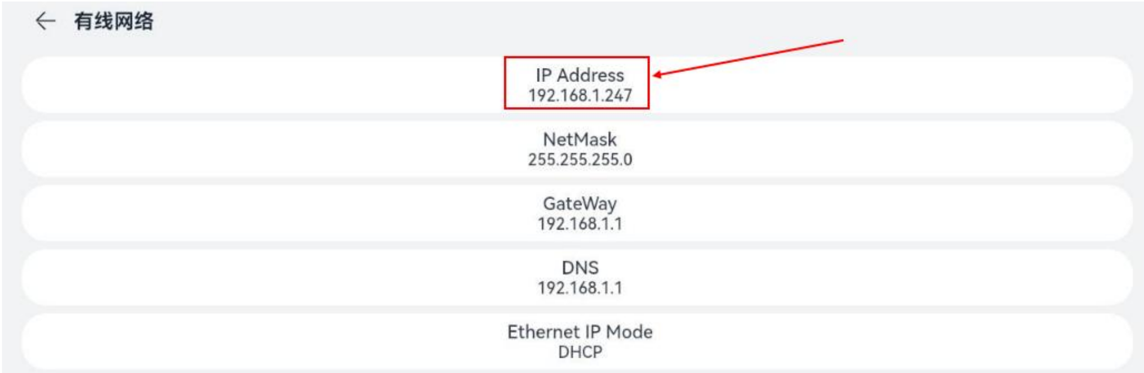


b. Then open **settings**

c. Then select **wired network**



d. Then you can see the IP address of the network port and other information.

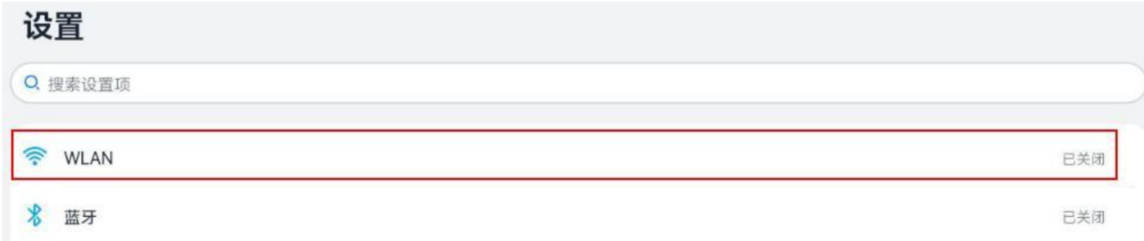


How to use WIFI

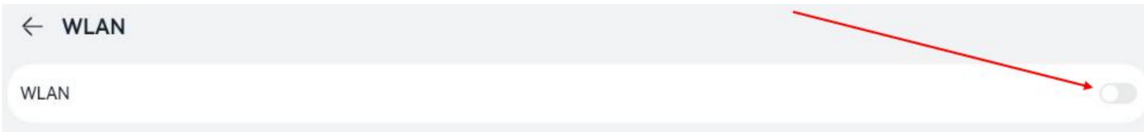
1) First click on the **application list**

2) Then open **settings**

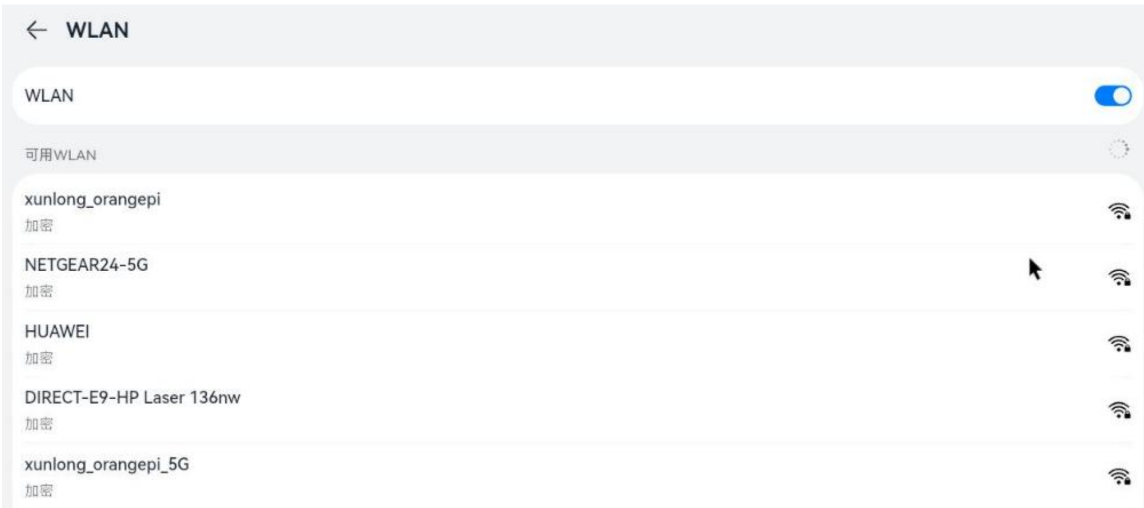
3) Then select **WLAN**



4) Then turn on **WLAN**



5) Then connect to the searched WIFI hotspot



7. Android 11 operating system instructions

7. 1. Supported Android versions

Android version	Kernel version
Android 11	Linux4.19

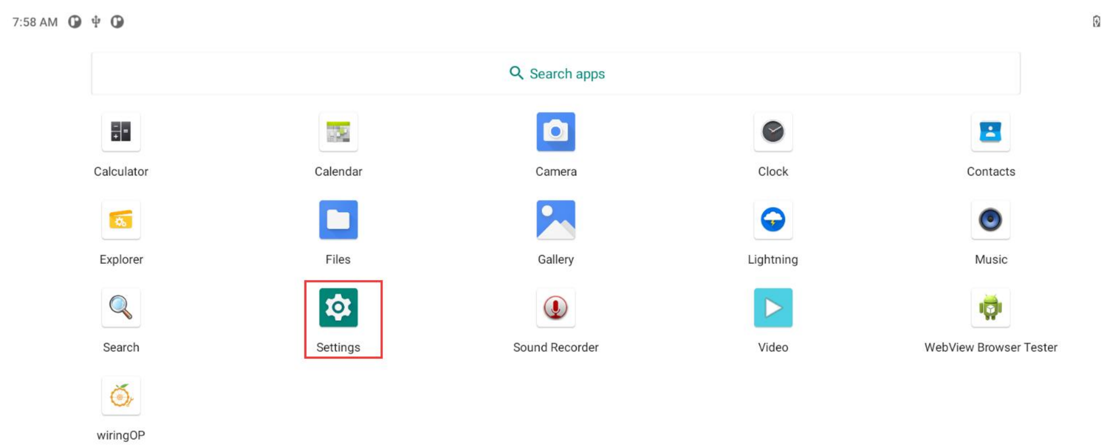
7. 2. Android Function Adaptation

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

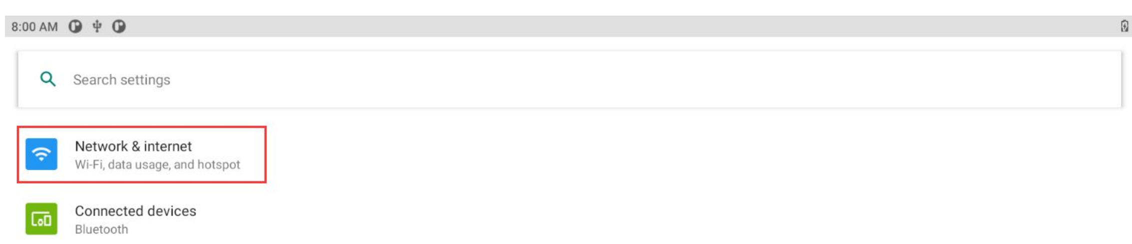
Headphone playback	**OK**
Headphone recording	**OK**
LED Light	**OK**
GPU	**OK**
NPU	**OK**
VPU	**OK**
RTC	**OK**

7. 3. WIFI connection test method

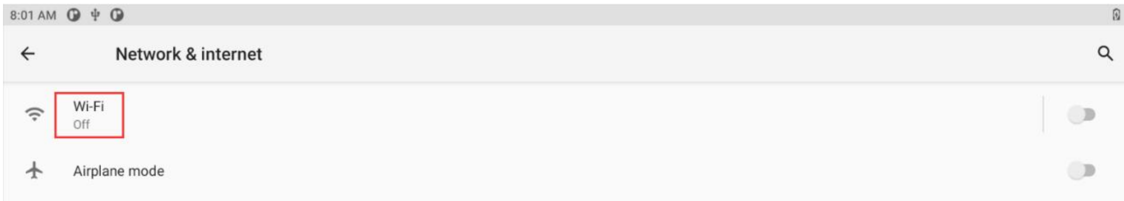
1) First click enter **Setting**



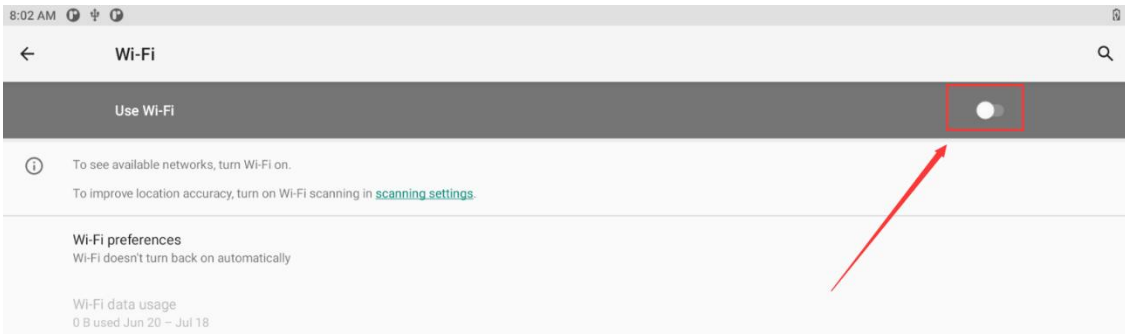
2) Then select **Network & internet**



3) Then select **Wi-Fi**

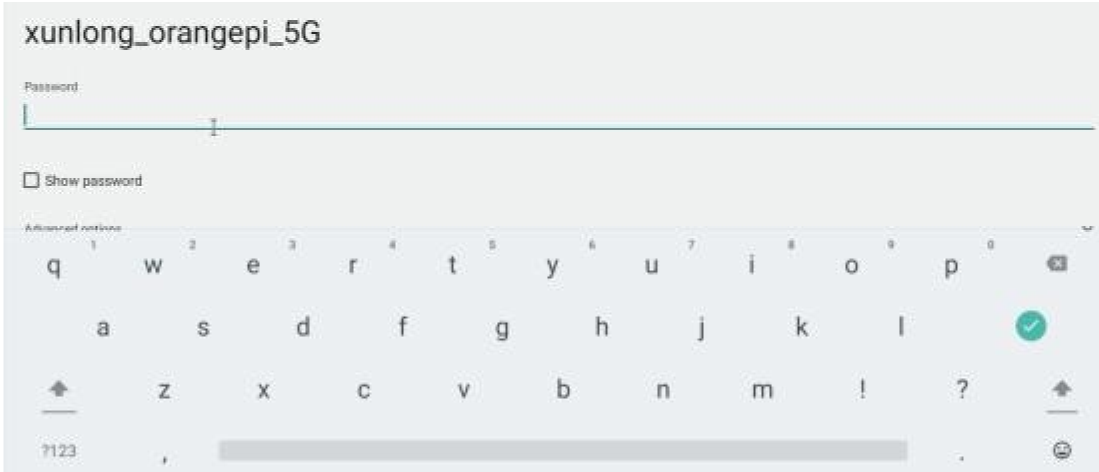


4) Then turn on the **Wi-Fi** switch



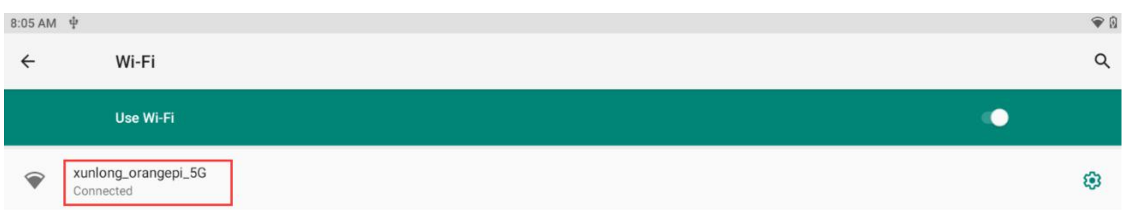
5) After turning on **Wi-Fi**, if everything is normal, you can scan for nearby Wi-Fi hotspots

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

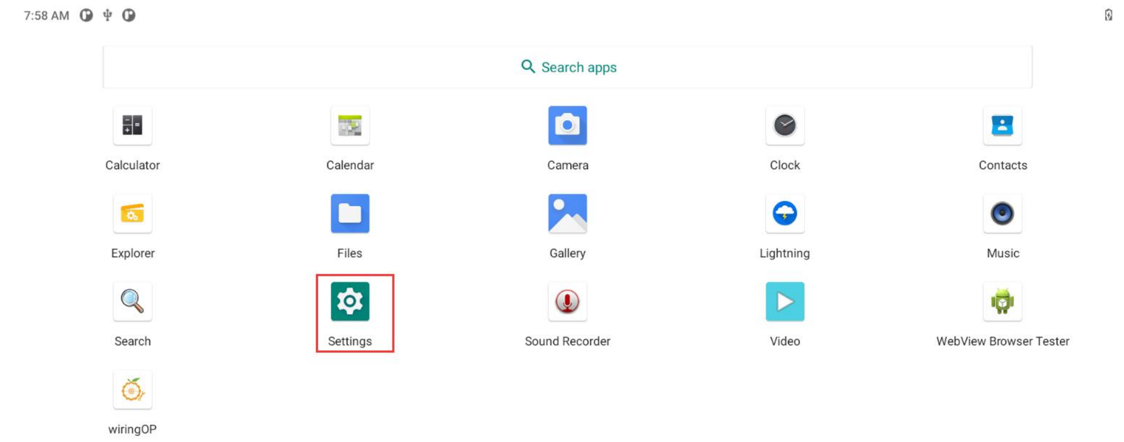
8) After the Wi-Fi connection is successful, the display is as shown in the figure below:



7. 4. How to use Wi-Fi hotspot

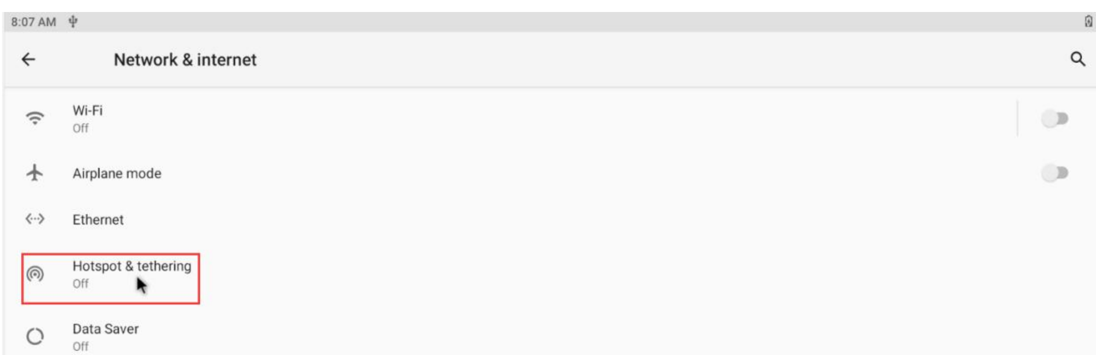
1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally

2) Then select **Settings**

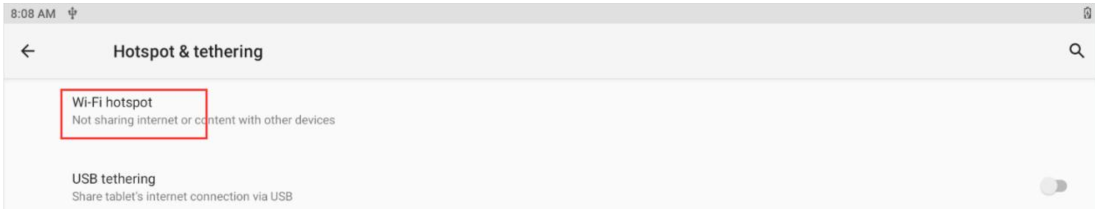


3) Then select **Network & internet**

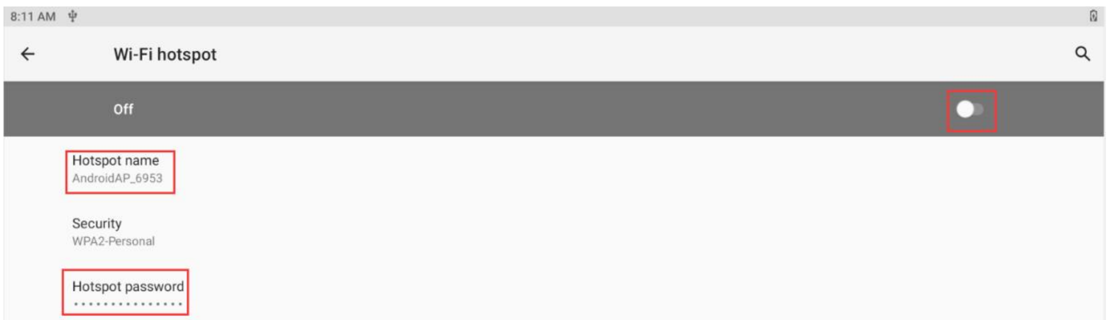
4) Then select **Hotspot & tethering**



5) Then select **Wi-Fi hotspot**



6) Then 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)



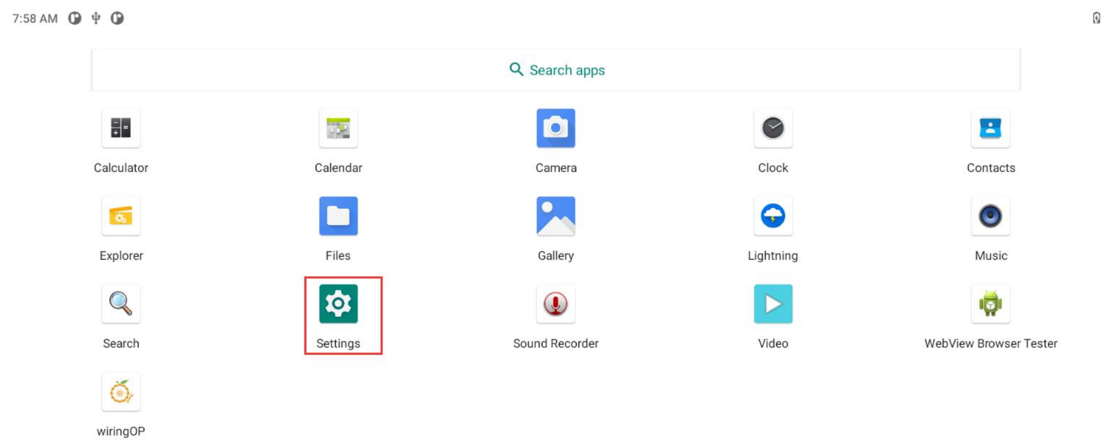
7) At this time, you can take out your mobile phone. If everything is normal, you can find the WIFI hotspot with the same name (**here AndroidAP_6953**) displayed under the **Hotspot name** in the above picture in the WI-FI list searched by the mobile phone. Then you can click **AndroidAP_6953** to connect 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

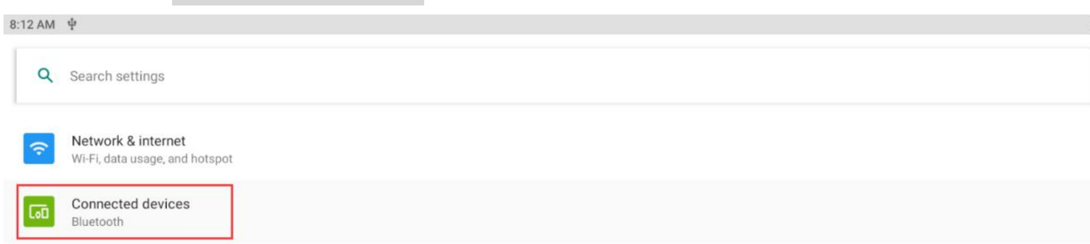
to the display of your mobile phone). At this point, you can open a webpage on your mobile phone to see if you can access the Internet. If you can open the webpage normally, it means that the **WI-FI Hotspot** of the development board can be used normally.

7. 5. Bluetooth test method

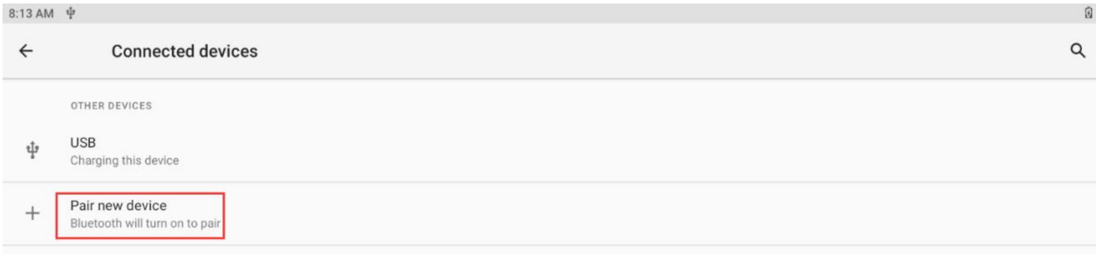
1) First click enter **Setting**



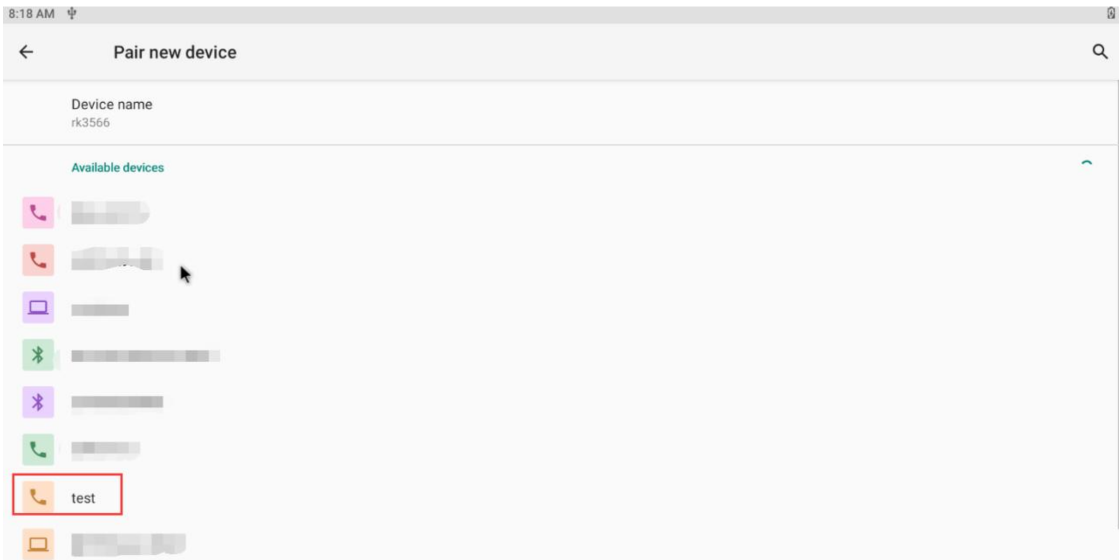
2) Then select **Connected devices**



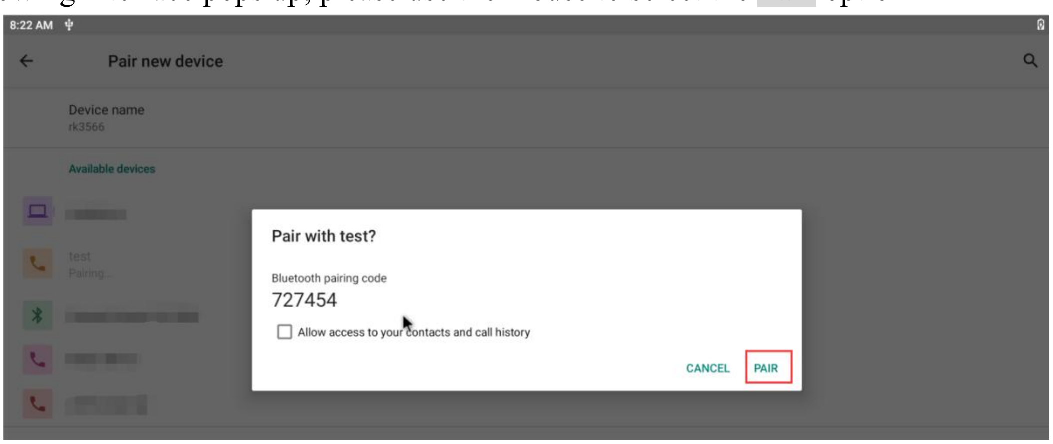
3) Then click **Pair new device** to turn on Bluetooth and start scanning the surrounding Bluetooth devices



4) The searched Bluetooth devices will be displayed under **Available devices**



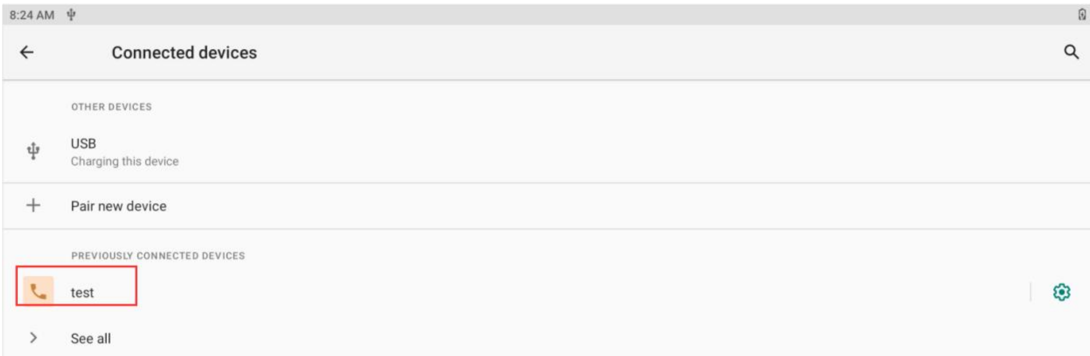
5) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option



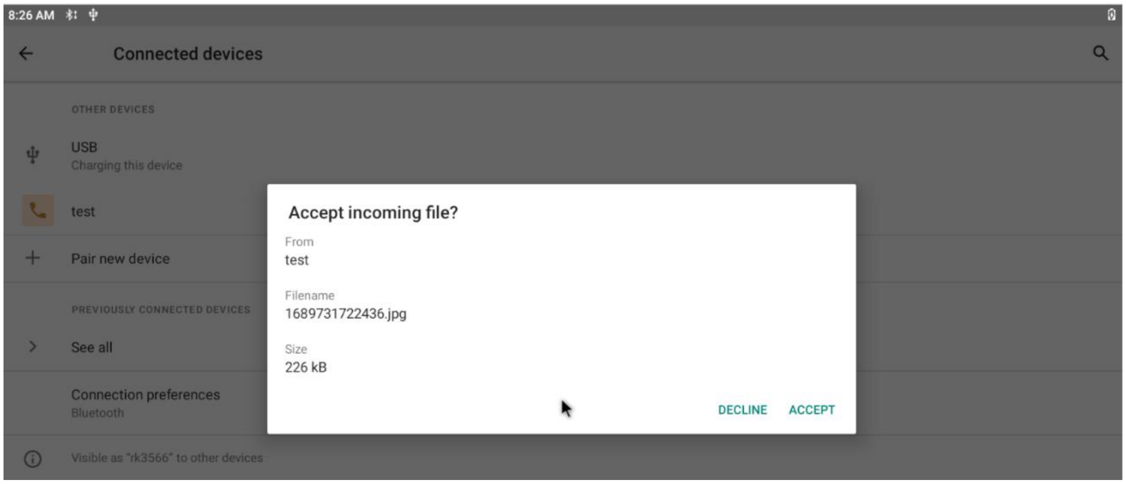
6) The test here is the configuration process of the development board and the Bluetooth of the Android mobile phone. At this time, the following confirmation interface will pop up on the mobile phone. After clicking the pairing button on the mobile phone, the

pairing process will start

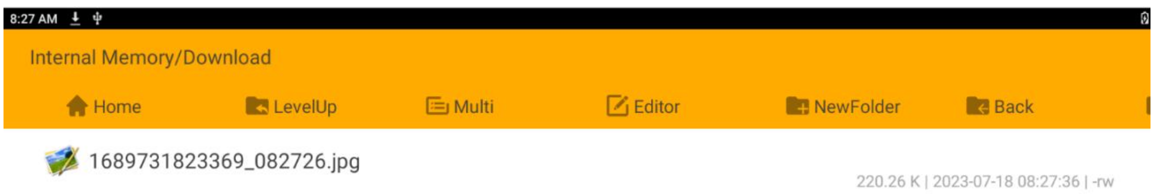
7) After the pairing is completed, you can see the paired Bluetooth device as shown in the figure below



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



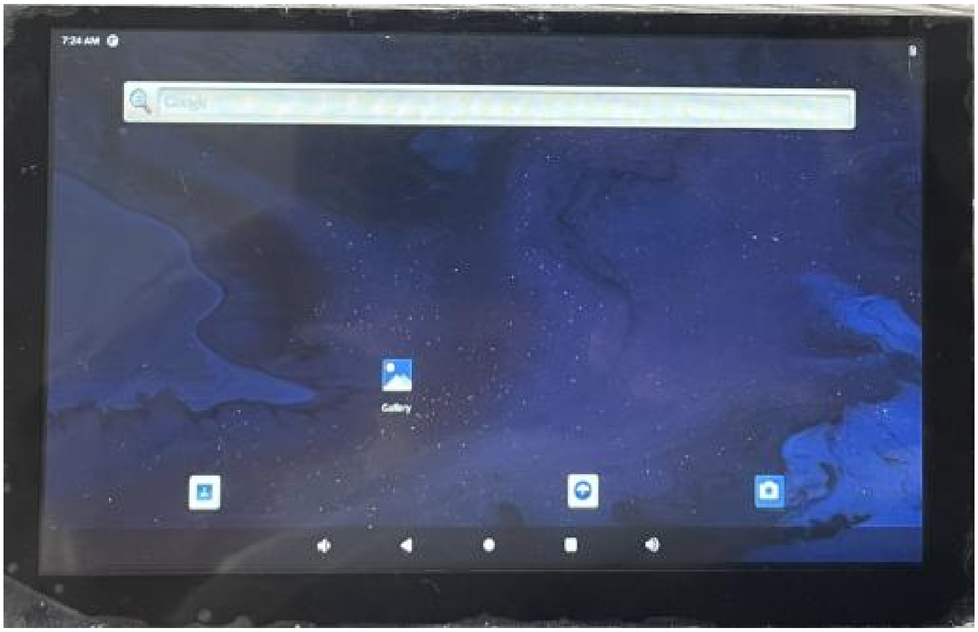
7. 6. How to use Raspberry Pi 5-inch screen

> **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.**

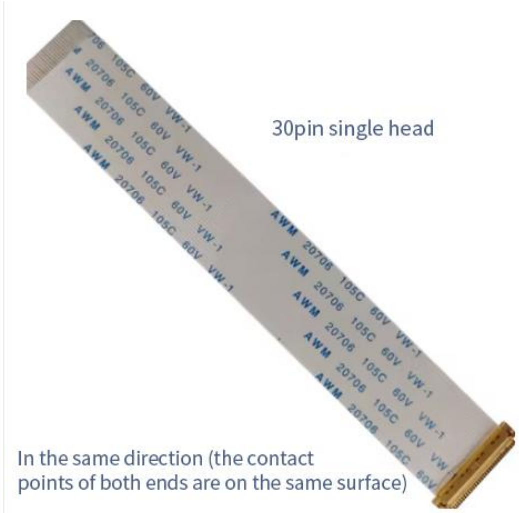


7. 7. How to use the eDP screen

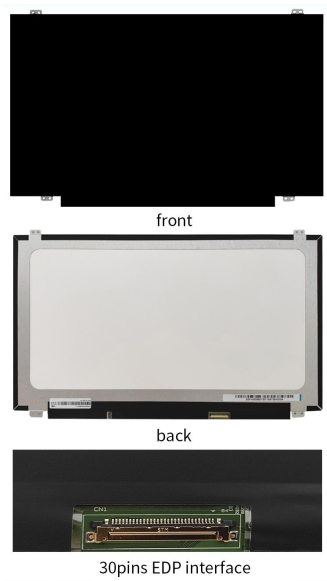
Please make sure that the image used is the following two versions of the image:
OrangePi3B_RK3566_Android11_lcd_v1.x.x.img
OrangePi3B_RK3566_Android11_spi-nvme_lcd_v1.x.x.img

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



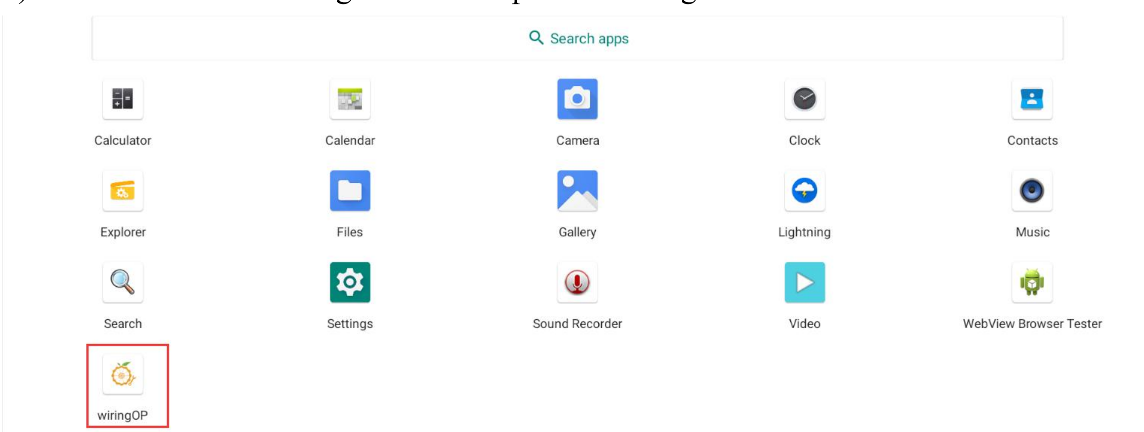
2) Connect the FPC end of the 30pin single-head codirectional cable to the eDP interface of the development board, and connect the other end to the eDP interface of the screen

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

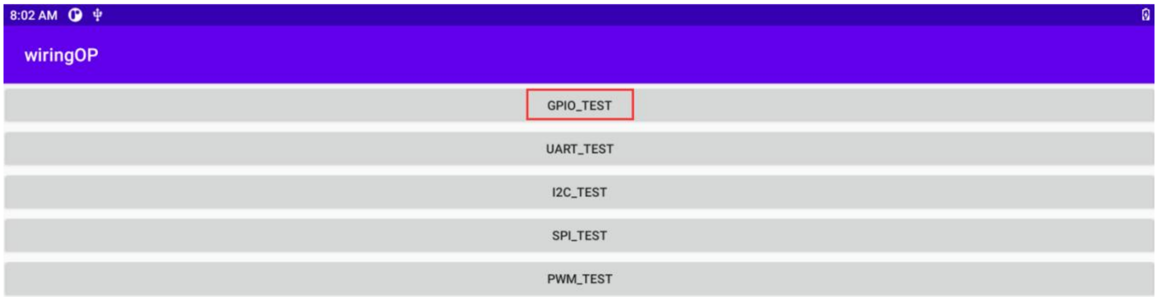
7. 8. 40pin interface GPIO, UART, SPI and PWM test

7. 8. 1. 40pin GPIO port test

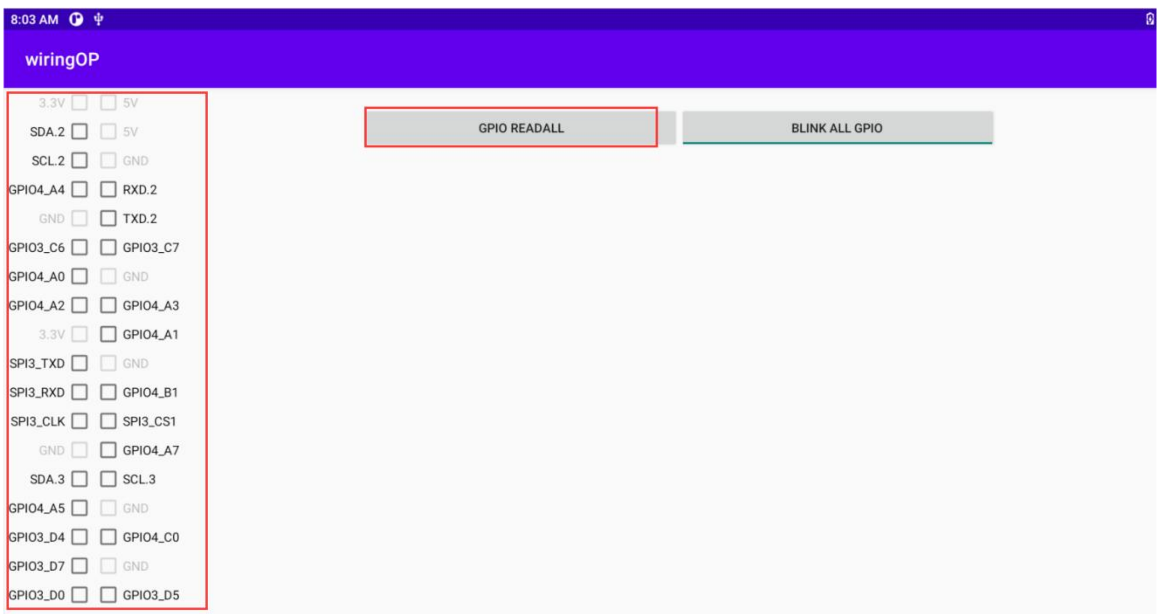
1) First click on the wiringOP icon to open the wiringOP APP



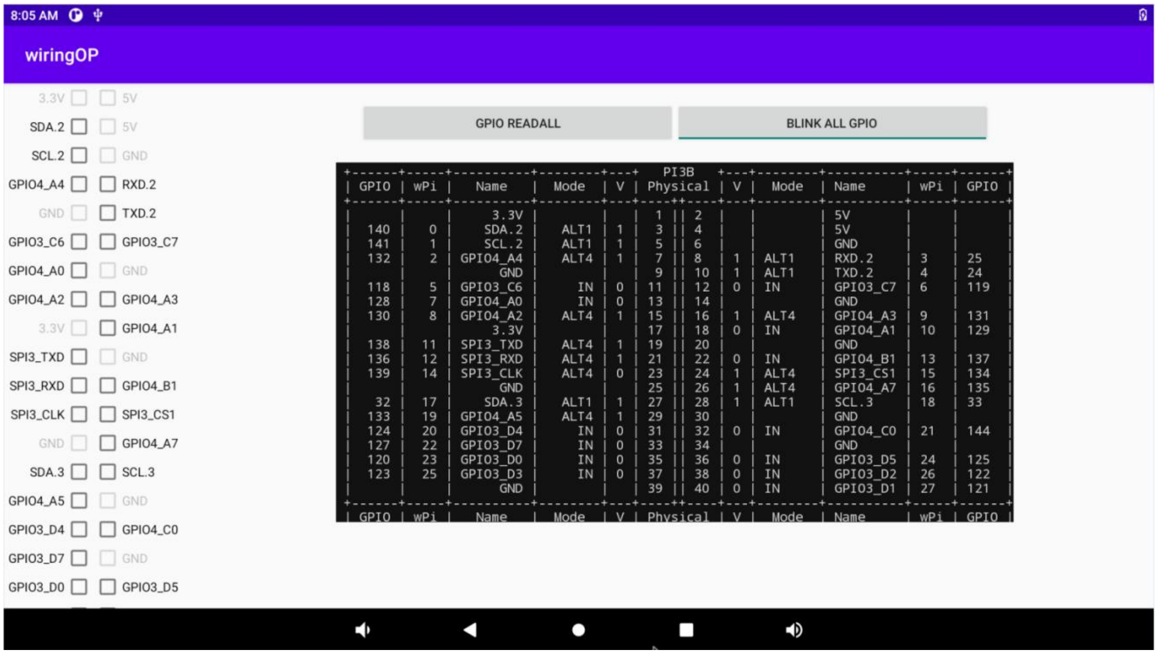
2) The main interface of wiringOP APP is displayed as shown in the figure below, and then click the **GPIO_TEST** button to open the GPIO test interface



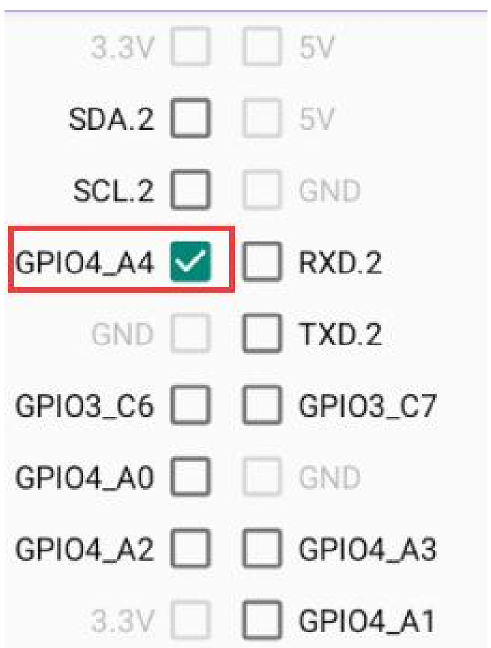
3) The GPIO test interface is shown in the figure below. The two rows of **CheckBox** buttons on the left are in one-to-one correspondence with the 40pin pins. When the **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT** mode, and the pin level will be set to high level; 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



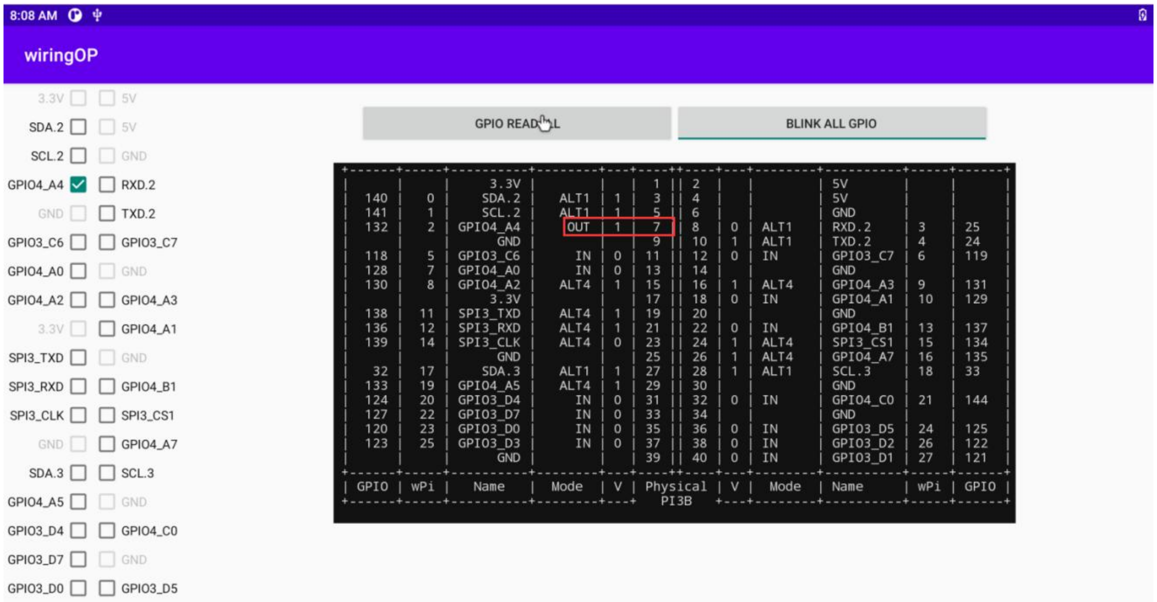
4) Then click the **GPIO READALL** button, the output information is as shown in the figure below:



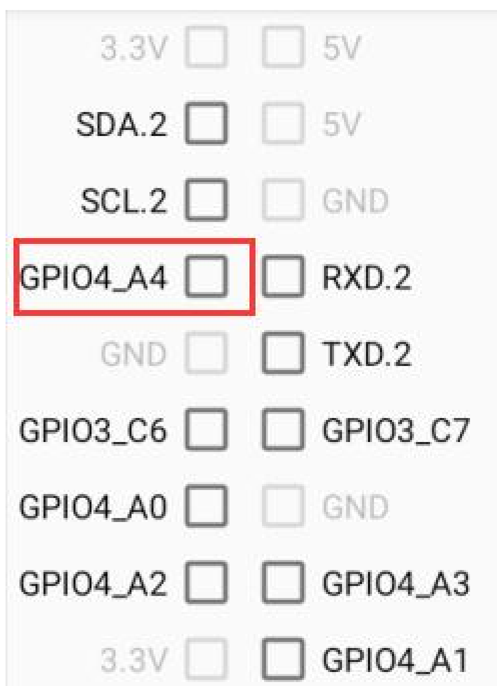
5) There are a total of 28 GPIO ports in the 40pins of the development board that can be used. The following uses pin 7 — the corresponding GPIO is GPIO4_A4 — the corresponding wPi serial number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port. First click the **CheckBox** 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

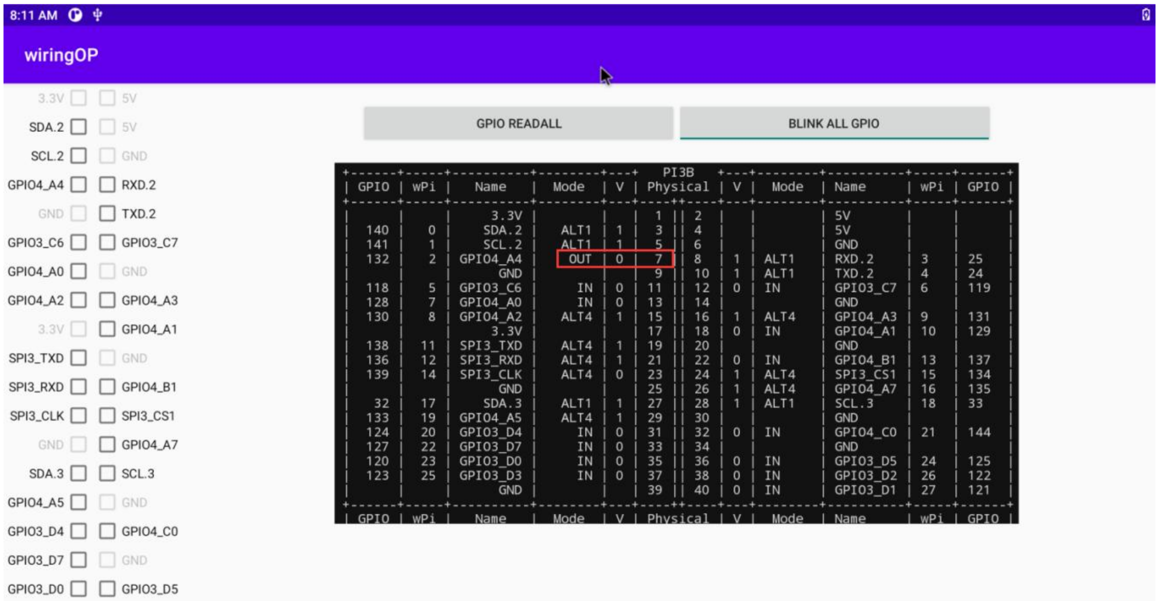


7) Click the **CheckBox** button in the figure below again to cancel the check status. Pin 7 will be set to low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level is set successfully.



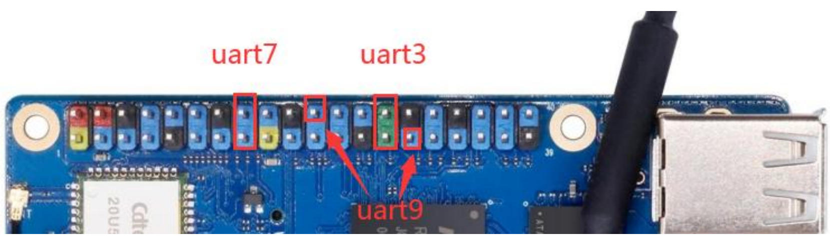
8) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is

OUT, and the pin level is low

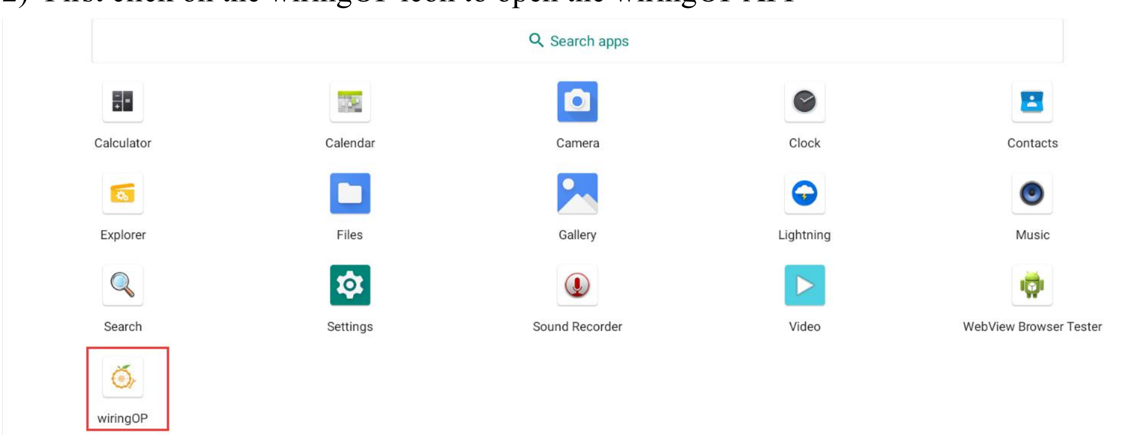


7. 8. 2. 40pin UART test

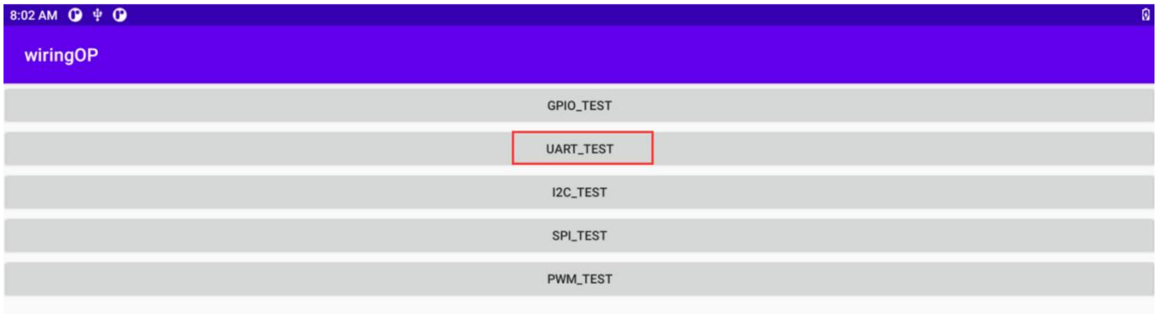
1) UART7 and UART9 are enabled by default in Android. The position of the 40pin is shown in the figure below, and the corresponding device nodes are **/dev/ttyS7** and **/dev/ttyS9** respectively



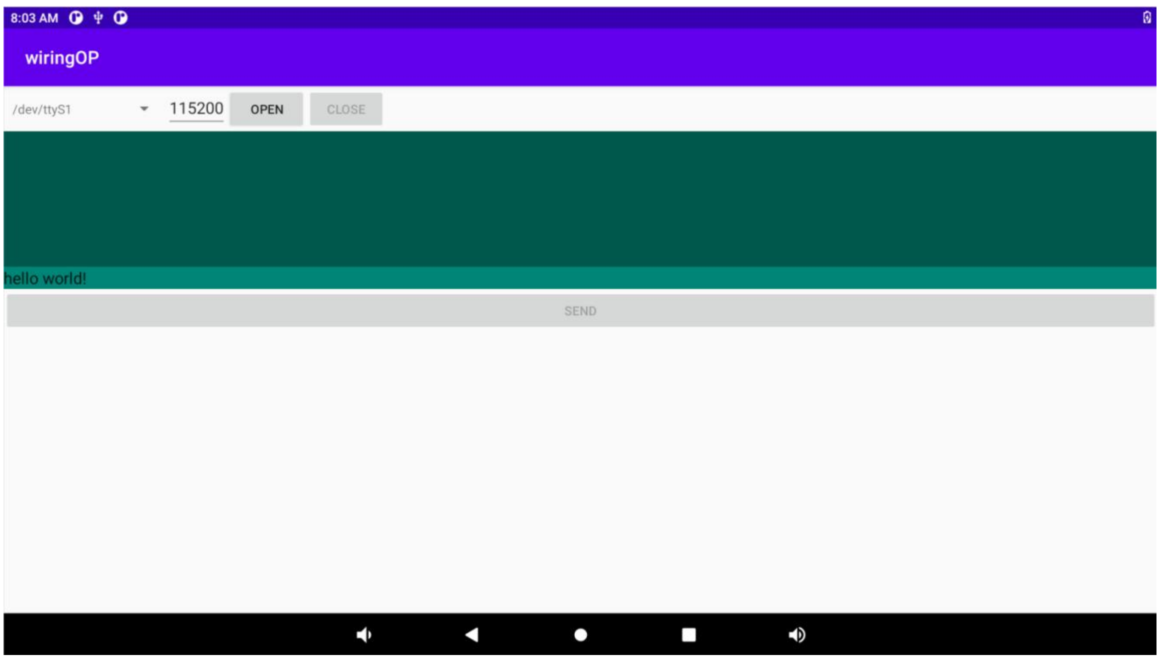
2) First click on the wiringOP icon to open the wiringOP APP



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



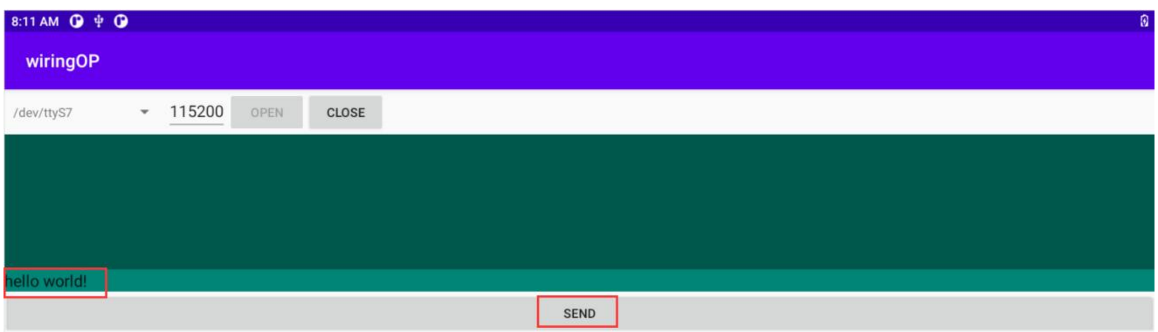
4) The serial port test interface of the APP is shown in the figure below



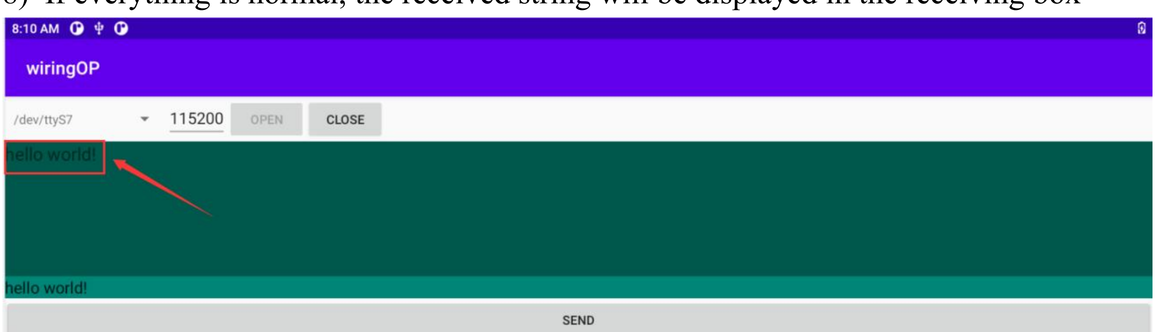
5) Take the test of **UART7** as an example below, select the **/dev/ttyS7** node in the selection box, enter the baud rate you want to set in the edit box, and then click the **OPEN** button to open the /dev/ttyS7 node. After the opening is successful, the **OPEN** button becomes unselectable, and the **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



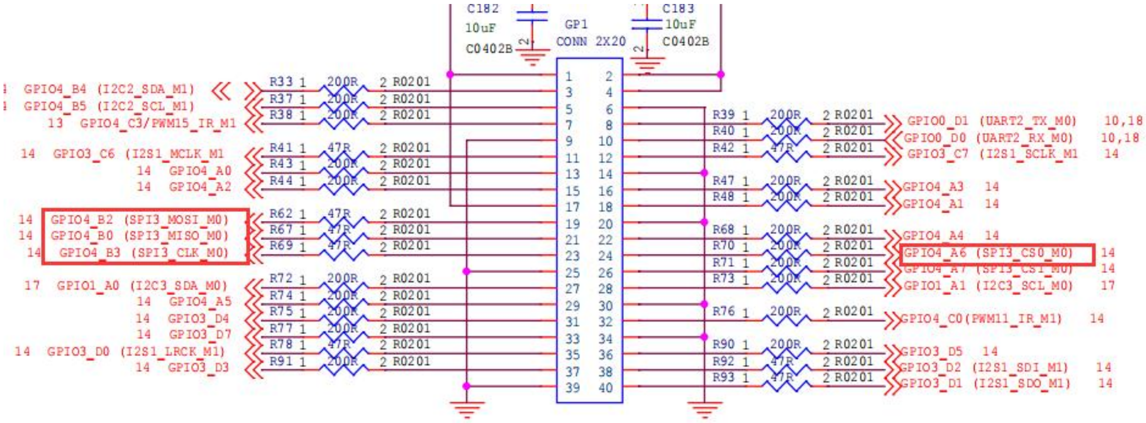
8) If everything is normal, the received string will be displayed in the receiving box



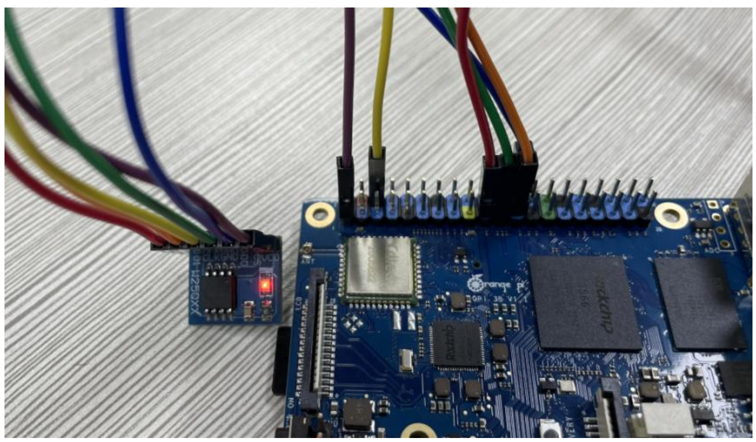
7.8.3. 40pin SPI test

1) According to the schematic diagram of the 40pin interface, the spi available for

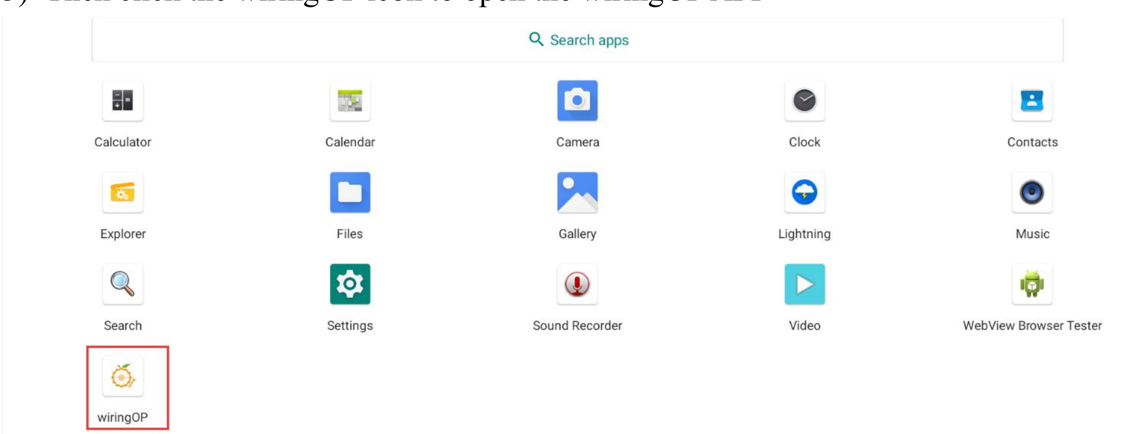
Orange Pi 3B is spi3



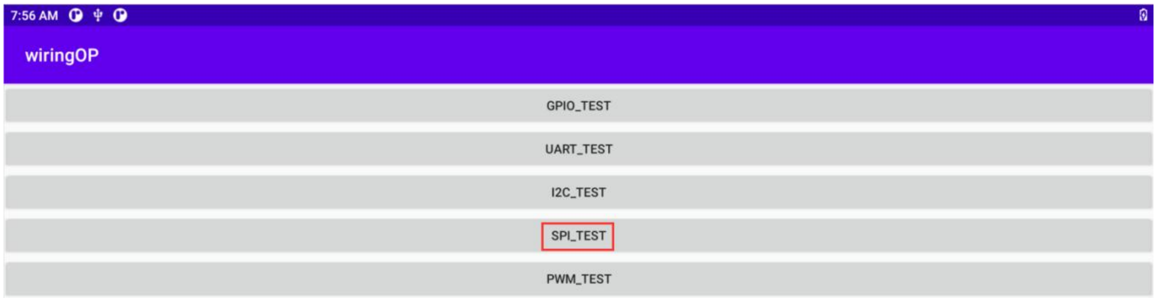
2) Here, the SPI interface is tested through the w25q64 module. First, the w25q64 device is connected to the SPI3 interface



3) Then click the wiringOP icon to open the wiringOP APP

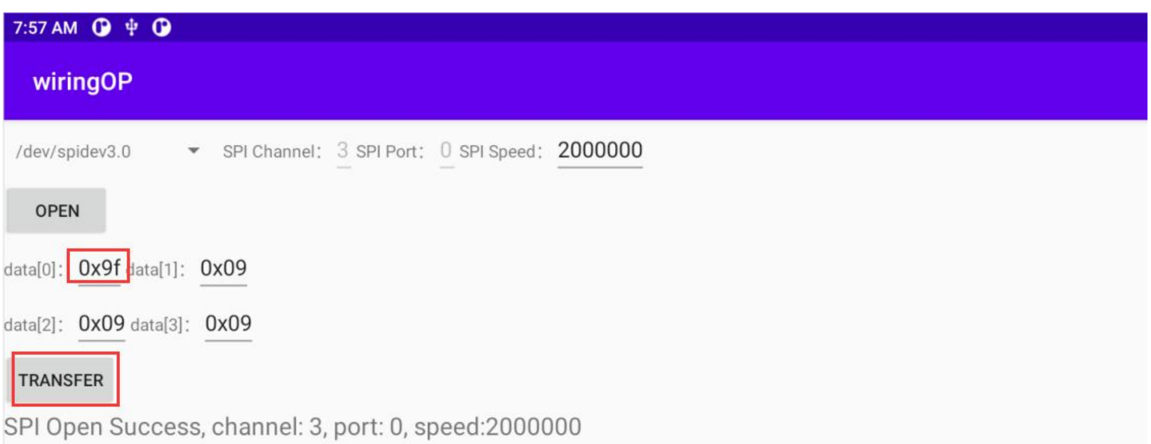


4) The main interface of wiringOP APP is displayed as shown in the figure below, click the SPI_TEST button to open the SPI test interface

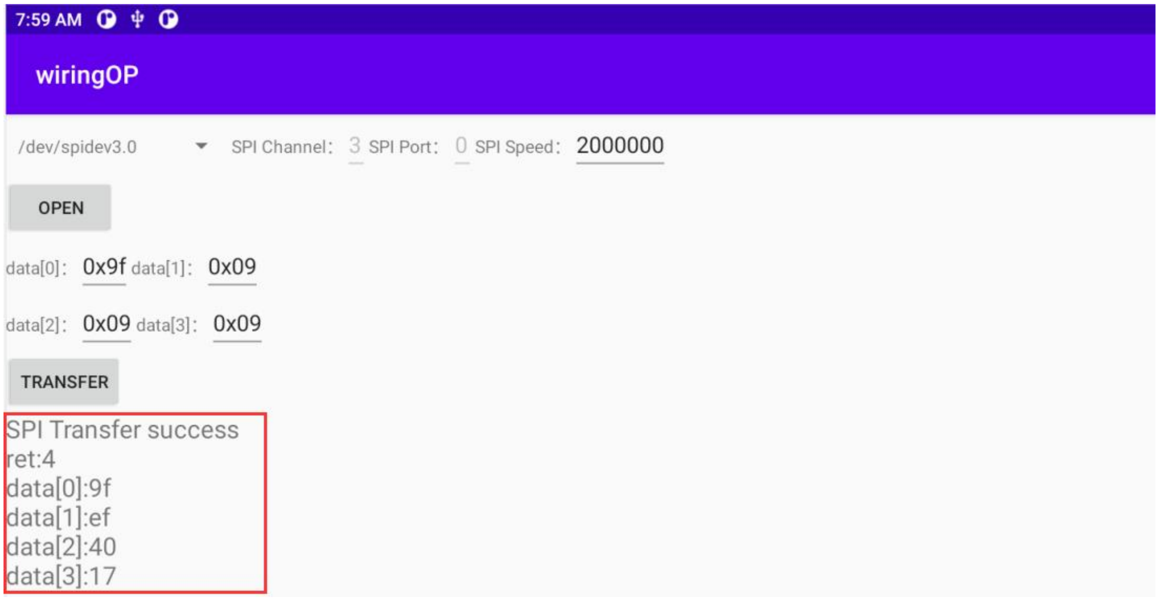


5) Then click the **OPEN** button to initialize the SPI

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

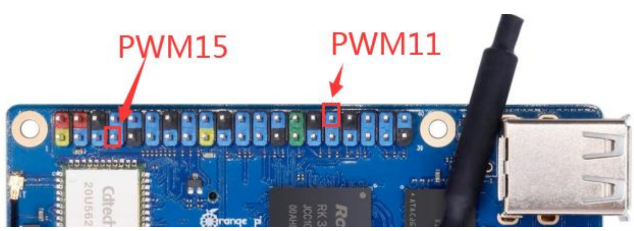


8) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, corresponding to the value read above (h stands for hexadecimal)

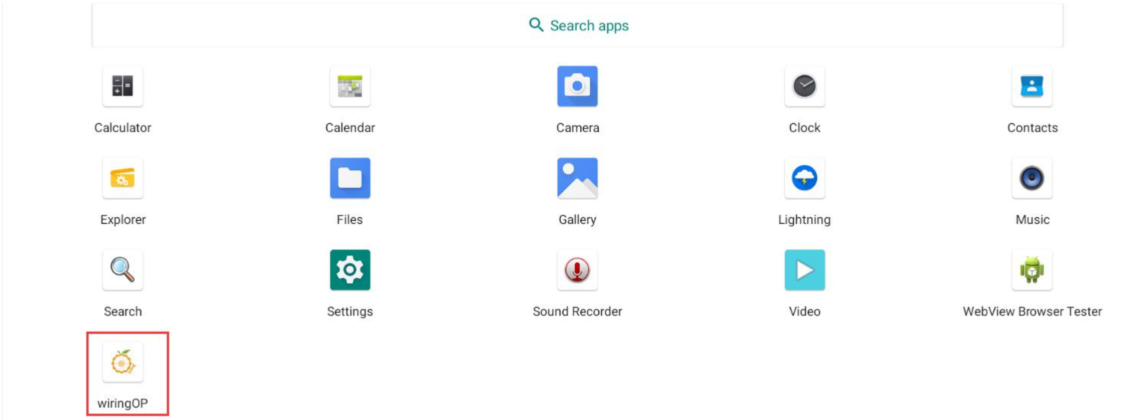
MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

7. 8. 4. 40pin PWM test

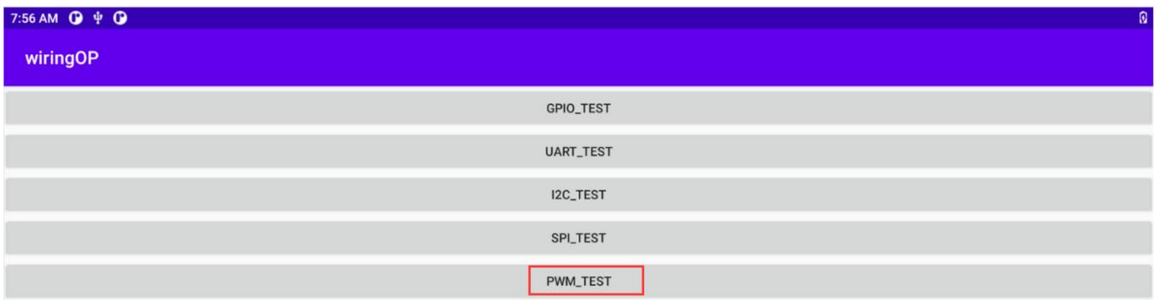
1) Android enables **PWM11** by default, and the corresponding pin is located at 40pin as shown in the figure below



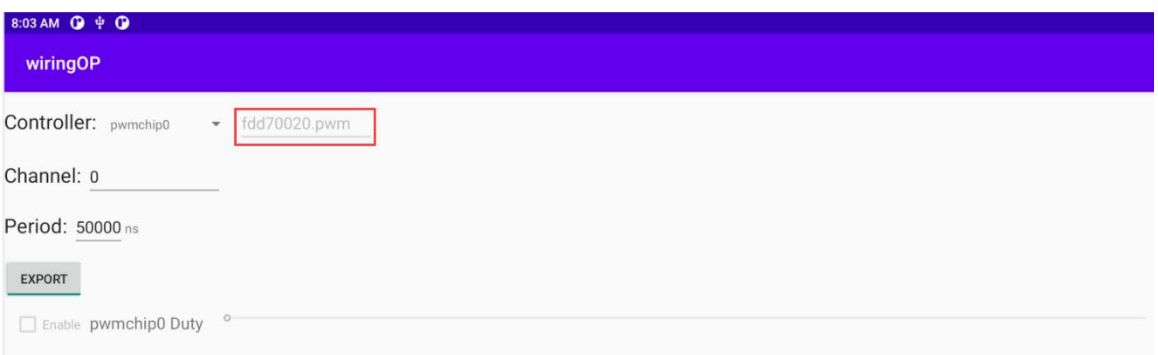
2) First click on the wiringOP icon to open the wiringOP APP



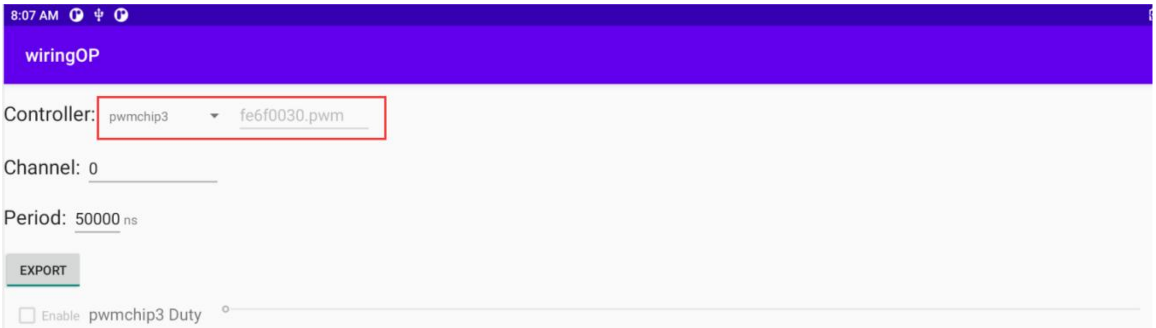
3) Then click the **PWM_TEST** button on the main interface of wiringOP to enter the PWM test interface



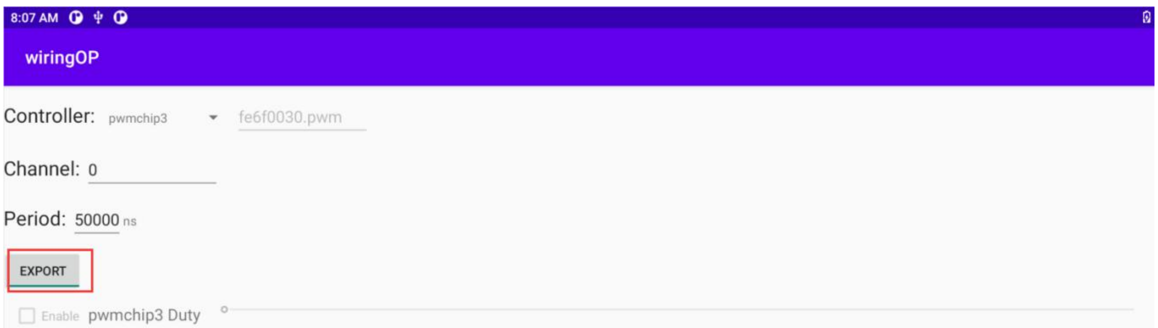
4) The base address corresponding to PWM11 is **fe6f0030**, here pwmchip0 shows **fdd70020.pwm** on the right, then you need to click the drop-down option to select other pwmchips until **fe6f0030.pwm** is displayed on the right



5) When the drop-down option selects **pwmchip3**, the corresponding base address of PWM11 is **fe6f0030** on the right

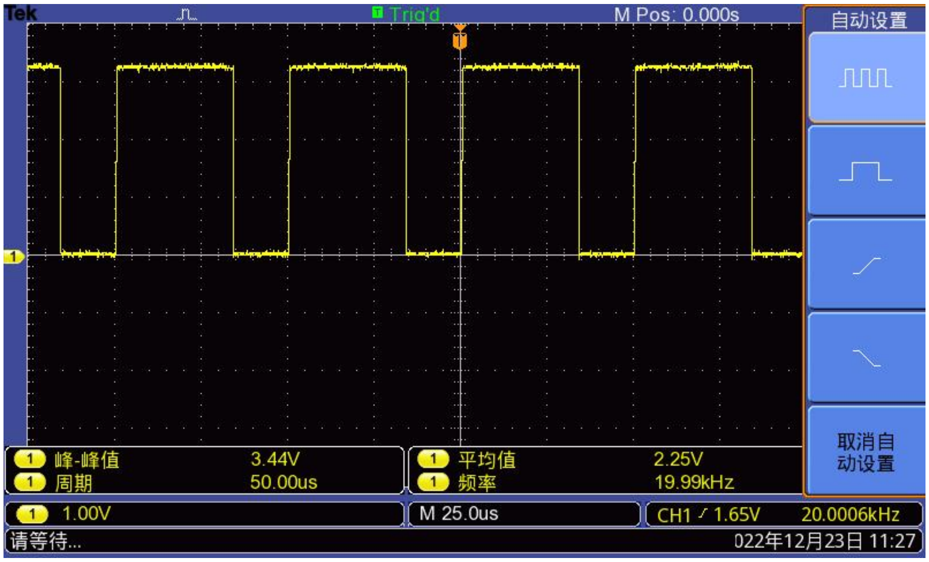


6) Then confirm the PWM channel, the default is channel 0, and confirm the PWM cycle, the default configuration is **50000ns**, converted to PWM frequency is **20KHz**, you can modify it yourself, click the **EXPORT** button to export **PWM11**



7) Then drag the drag bar below to change the PWM duty cycle, and then check Enable to output the PWM waveform

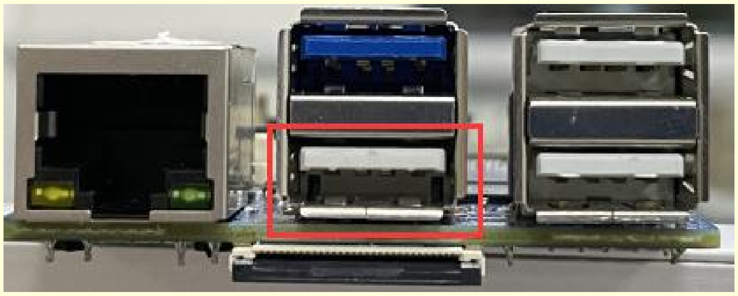
8) Then use an oscilloscope to measure the No. 32 pin in the 40pin of the development board, and you can see the following waveform



7.9. How to use ADB

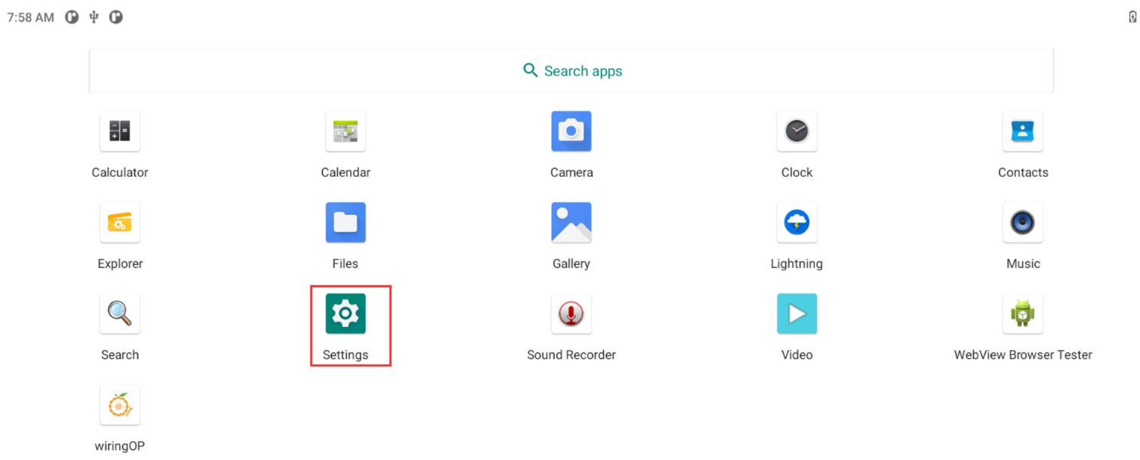
7.9.1. The method of USB OTG mode switching

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.

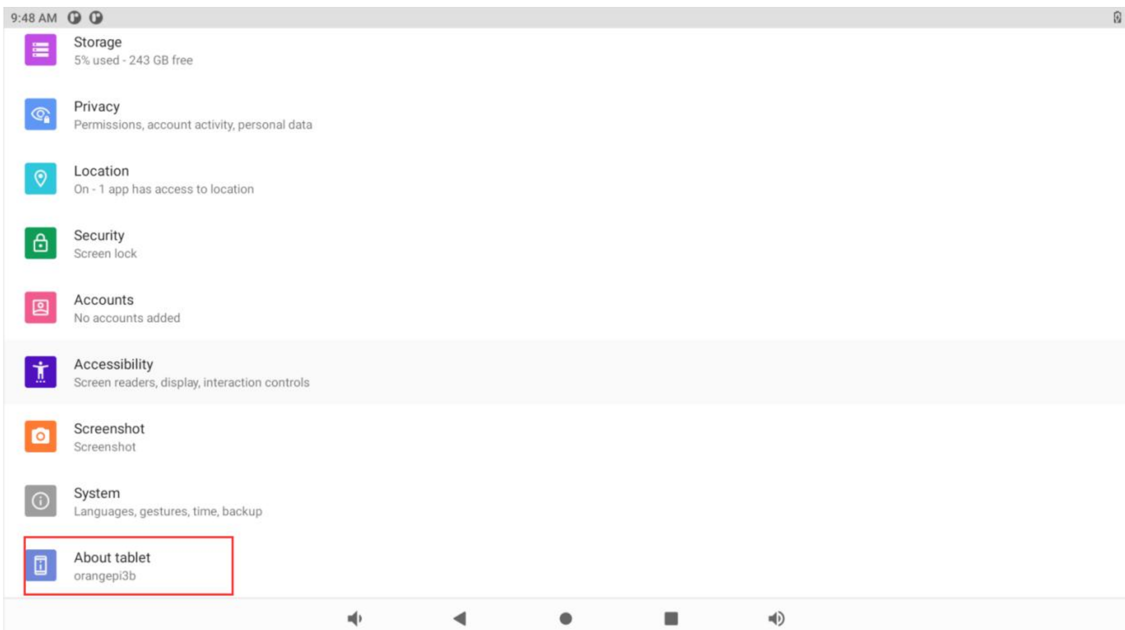


The USB OTG interface defaults to Host mode, which can be used to connect USB devices such as mouse and keyboard. If you want to use ADB, you need to manually switch to Device mode.

1) First open Settings



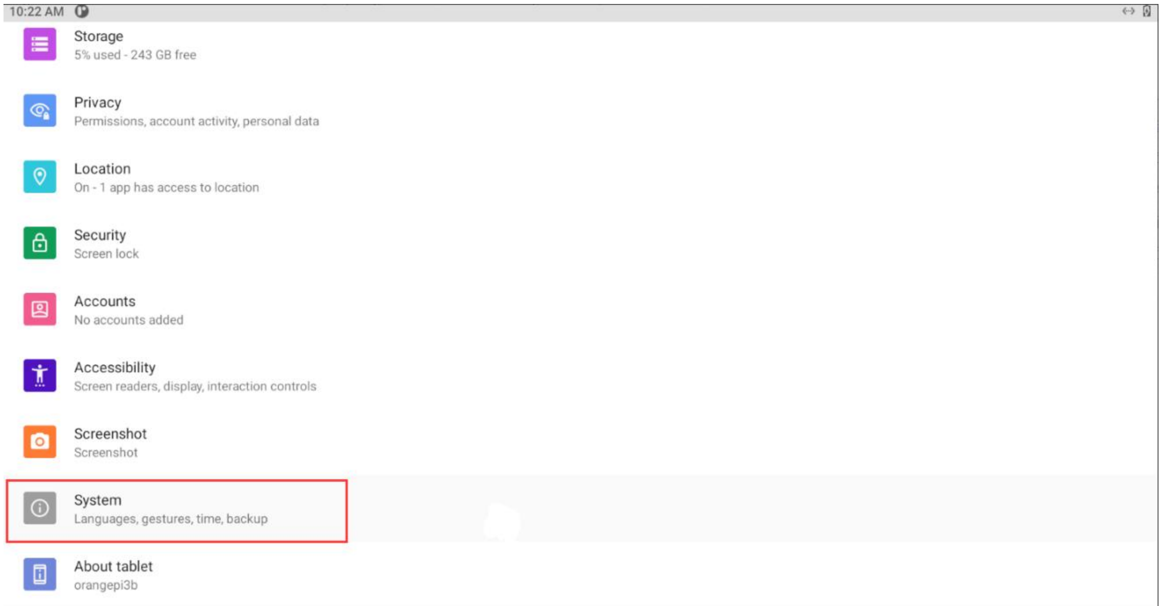
2) Then select **About tablet**



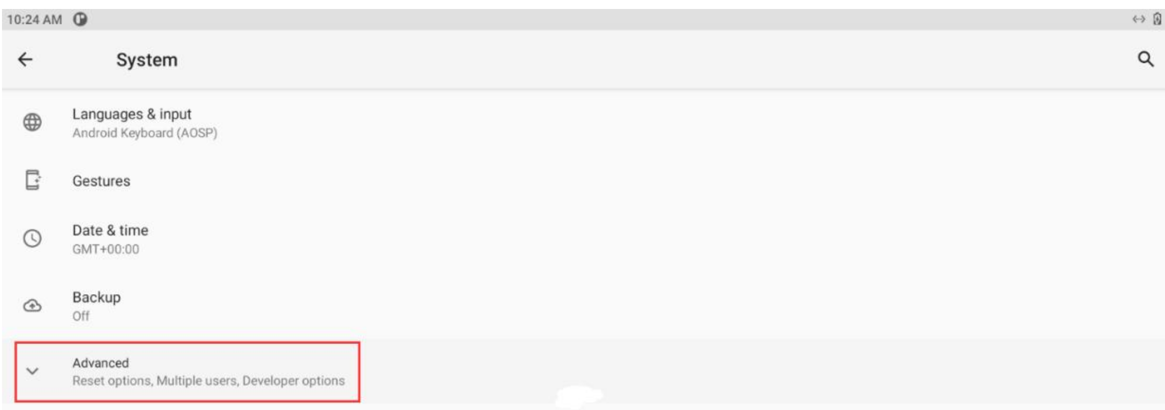
3) Then click the **Build number** menu bar several times with the mouse until the prompt **You are now a developer!** appears

4) Then click to return to the previous menu

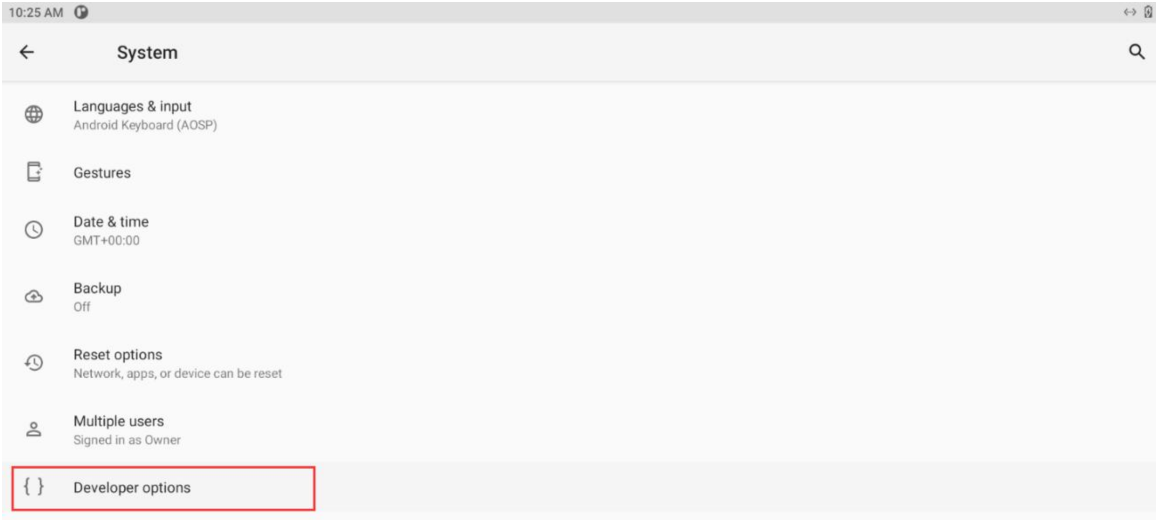
5) Then select **System**



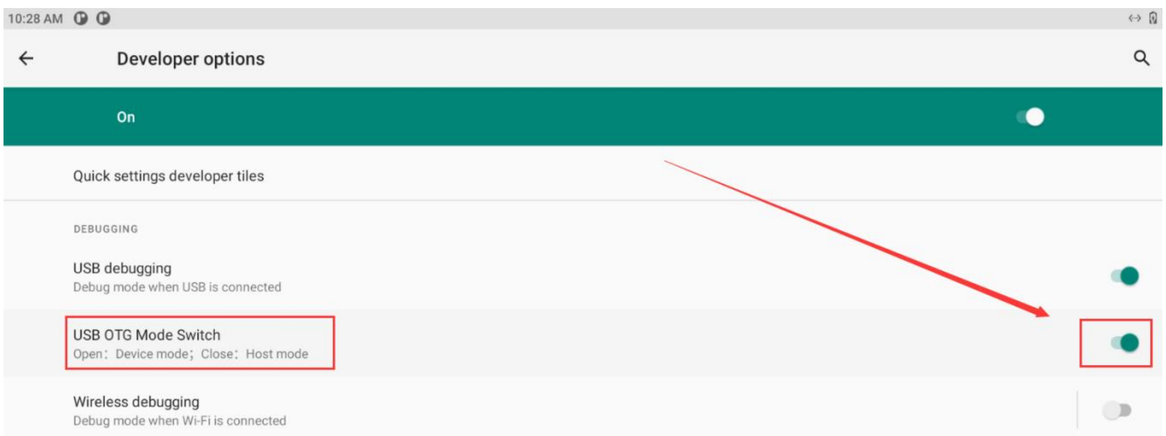
6) Then select **Advanced**



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**

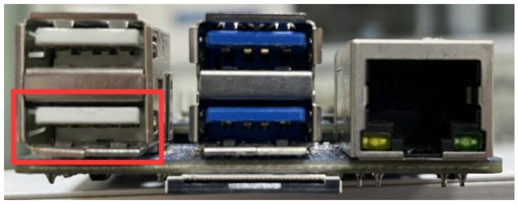


7. 9. 2. Use the data cable to connect to adb debugging

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/
```

7. 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:/ #
```

8. Instructions for using the OpenWRT system

8. 1. OpenWRT Version

OpenWRT Version	Kernel version
snapshot	Linux6.1

8. 2. OpenWRT adaptation situation

Function	OpenWRT
USB2.0x3	OK
USB3.0x1	OK
3pin debugging serial port	OK
TF card startup	OK
UWE5622 WIFI	NO
M.2 NVMe SSD boot	OK
Network port	OK
Network port status light	OK
RTL8811 USB network card	OK
USB to wired network card	OK
LED light	OK
FAN interface	OK
eMMC expansion interface	OK

8. 3. Start expanding rootfs for the first time

1) When the OpenWRT system is started for the first time, the resize-rootfs.sh script will be executed to expand the rootfs, and it will automatically restart after the expansion is completed.

2) After logging in to the system, you can use the **df -h** command to check the size of rootfs. If it is consistent with the actual capacity of the storage device (TF card, eMMC or NVME SSD), it means that the automatic expansion is running correctly.

```
root@OpenWrt:~# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                14.8G     14.7G     91.6M  99% /
tmpfs                   495.5M      6.1M    489.4M   1% /tmp
tmpfs                   512.0K         0    512.0K   0% /dev
/dev/root                14.8G     14.7G     91.6M  99% /opt/docker
```

8. 4. How to log in to the system

8. 4. 1. Log in through serial port

1) First, for the use of the debugging serial port, please refer to the chapter on **how to use the debugging serial port**.

2) The OpenWrt system will automatically log in as the root user by default, and the display interface is as follows

```
OpenWrt login: root


BusyBox v1.36.1 (2023-11-23 13:47:57 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt SNAPSHOT, r24453+3-a39a49e323
 -----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@OpenWrt:~#
```

8. 4. 2. Log in to the system via SSH

Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) If the Ubuntu system is installed on the computer, you can execute the following command to log in to the system through SSH. By default, you can log in directly without a password

```
test@ubuntu:~$ ssh root@192.168.2.1
```

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

5) If the computer is installed with Windows system, you can refer to the method of **SSH remote login to the development board under Windows** in the Linux system instruction manual to log in.

8. 4. 3. Log in to the LuCI management interface

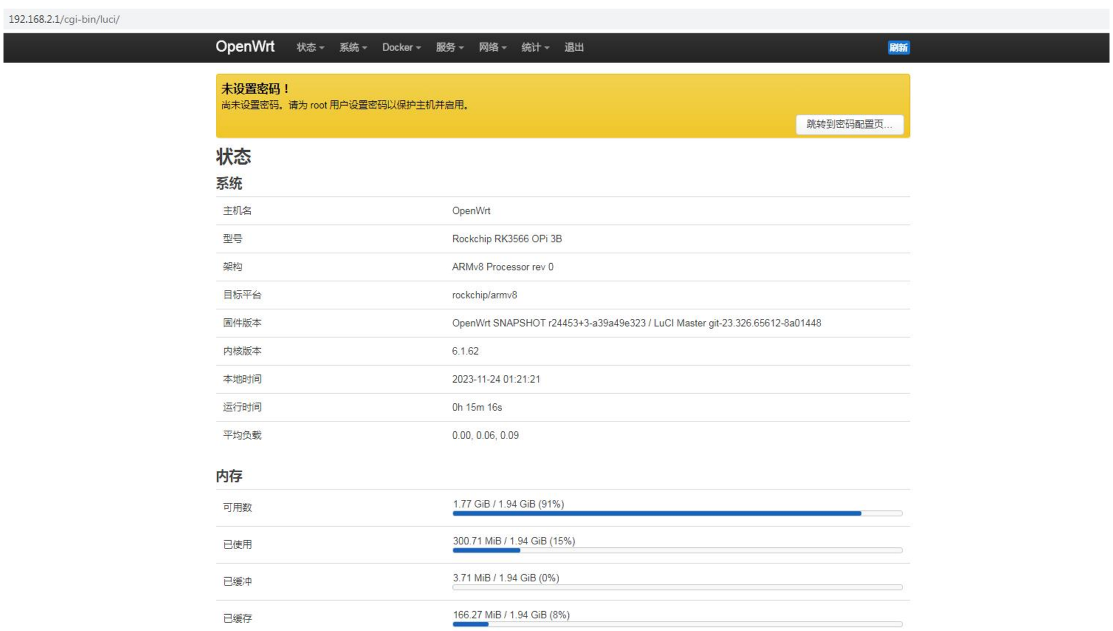
Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) Enter the IP address **192.168.2.1** in the browser on your computer to log in to the LuCI interface

4) The OpenWrt system does not set a password by default, so just click the login button. After successful login, the interface will be displayed as shown below.



8. 4. 4. Log in to the terminal through the LuCI management interface

Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through

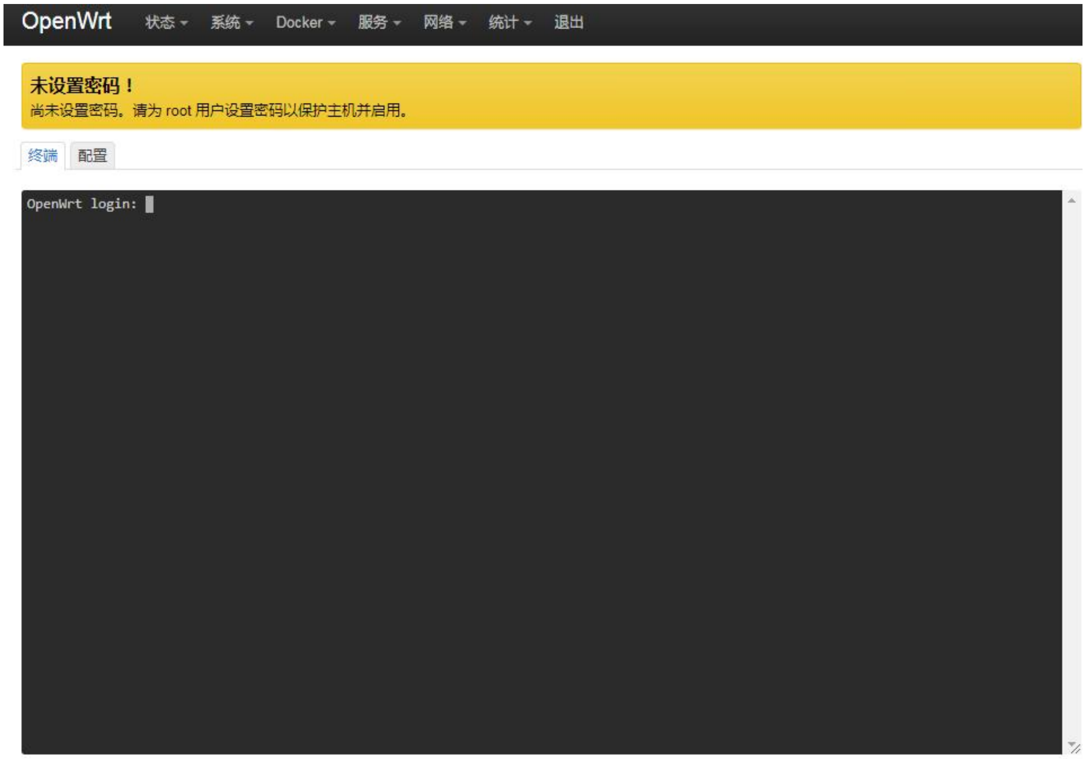
DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

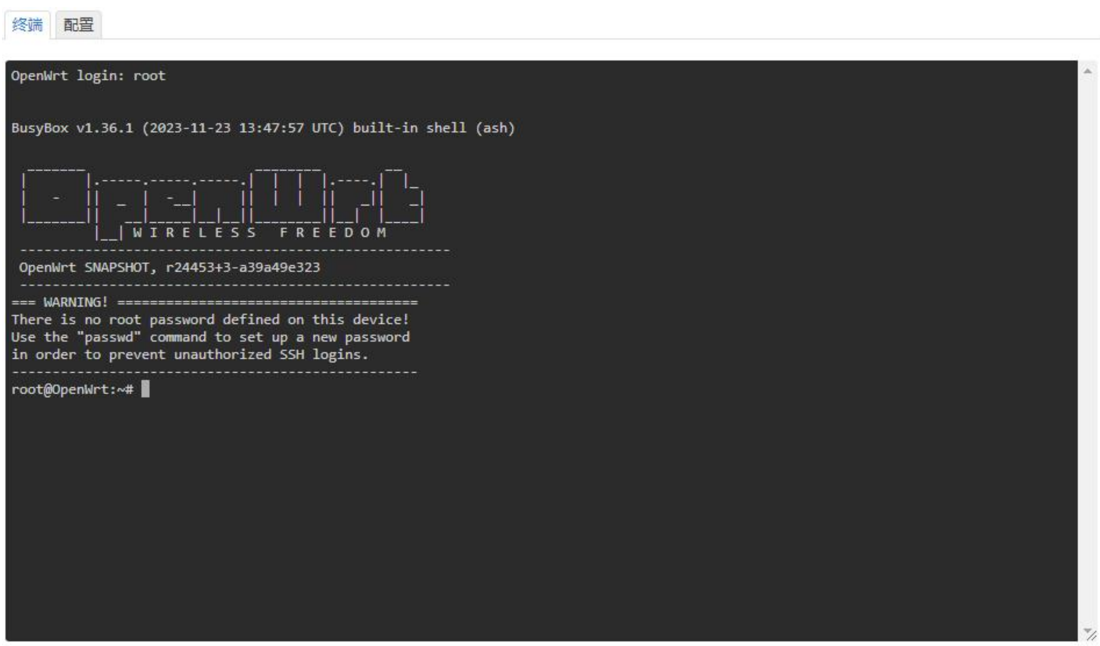
3) Enter the IP address **192.168.2.1** in the browser on your computer to log in to the LuCI interface

4) Select **"Terminal"** in the **"Service"** column of the navigation bar and click to enter

5) At this time, the terminal display interface is as shown below



6) Enter the username **root** to log in



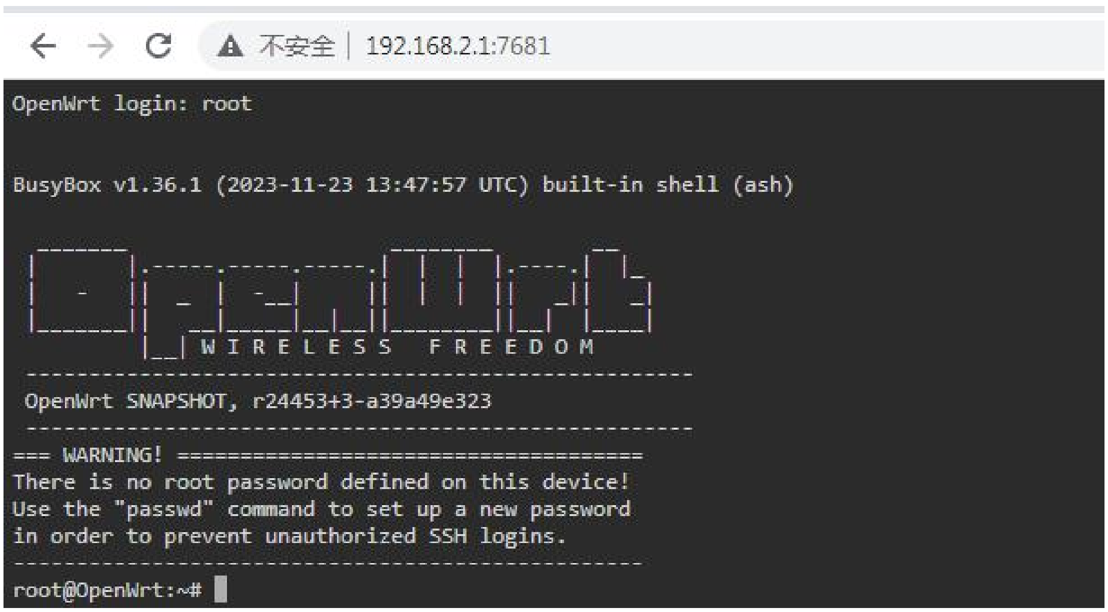
8. 4. 5. Log in to the terminal using IP address + port number.

Please note that in the OpenWrt system of Orange Pi 3B, the onboard network port is configured as a LAN port function by default.

1) First use a network cable to connect the LAN port of the board to the network port of the computer so that the computer's network port can obtain the IP address through DHCP.

2) The default LAN port IP of the board is set to **192.168.2.1**, so at this time the computer can obtain an IP address starting with **192.168.2**

3) Then enter **192.168.2.1:7681** in the browser to log in to the OpenWRT terminal



8.5. How to modify the LAN port IP address through the command line

1) In the OpenWrt system, a command line tool uci is provided, which can easily modify, add, delete and read the contents in the configuration file. For detailed instructions, please refer to the **official documentation**

2) First use the following command to obtain the network configuration. The corresponding configuration file is **/etc/config/network**. You can see that the value of **network.lan.ipaddr** is **192.168.2.1**

```
root@OpenWrt:~# uci show network
...
```

```
network.lan=interface
network.lan.device='br-lan'
network.lan.proto='static'
network.lan.ipaddr='192.168.2.1'
network.lan.netmask='255.255.255.0'
network.lan.ip6assign='60'
....
```

3) Then enter the following command to modify the **network.lan.ipaddr** item

```
root@OpenWrt:~# uci set network.lan.ipaddr='192.168.100.1'
```

4) Then enter the following command to complete the submission, that is, write it to the configuration file

```
root@OpenWrt:~# uci commit
```

If the IP address in red font is consistent with the one to be set, the modification is successful.

```
root@OpenWrt:~# cat /etc/config/network
...
config interface 'lan'
        option device 'br-lan'
        option proto 'static'
        option netmask '255.255.255.0'
        option ip6assign '60'
        option ipaddr '192.168.100.1'
...
```

5) Restart the network through ubus. For instructions on using ubus, please refer to the **official documentation**.

```
root@OpenWrt:~# ubus call network restart
```

6) At this time, enter the command and you can see that the IP of the LAN port is already **192.168.100.1**

```
root@OpenWrt:~# ifconfig br-lan
br-lan    Link encap:Ethernet  HWaddr FE:55:13:A3:EF:E7
```

```
          inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fd60:c4cd:1033::1/60 Scope:Global
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:370 (370.0 B)
```

8. 6. How to change the root password

8. 6. 1. Modification through command line

1) First enter passwd root on the system command line. The following prompt message will appear. At this time, you can enter the password you want to set and press the Enter key to confirm.

```
root@OpenWrt:/# passwd root
Enter new UNIX password:
```

2) You will then be prompted to re-enter your password. At this time, enter your password again to confirm and press Enter.

```
Retype password:
```

3) The successful modification is displayed as follows

```
passwd: password for root changed by root
```

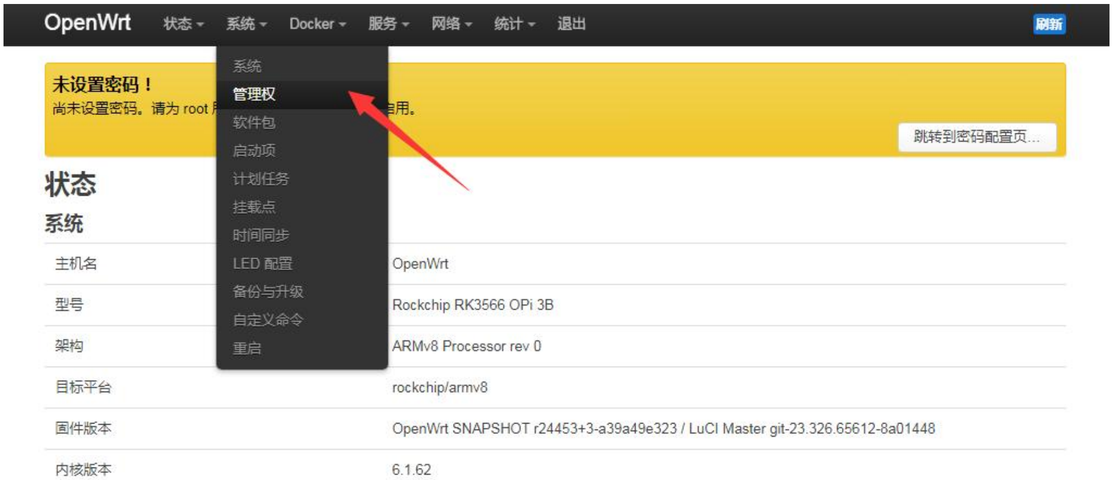
8. 6. 2. Modify through LuCI management interface

1) First refer to **logging in to the LuCI management interface** to enter the OpenWRT management interface.

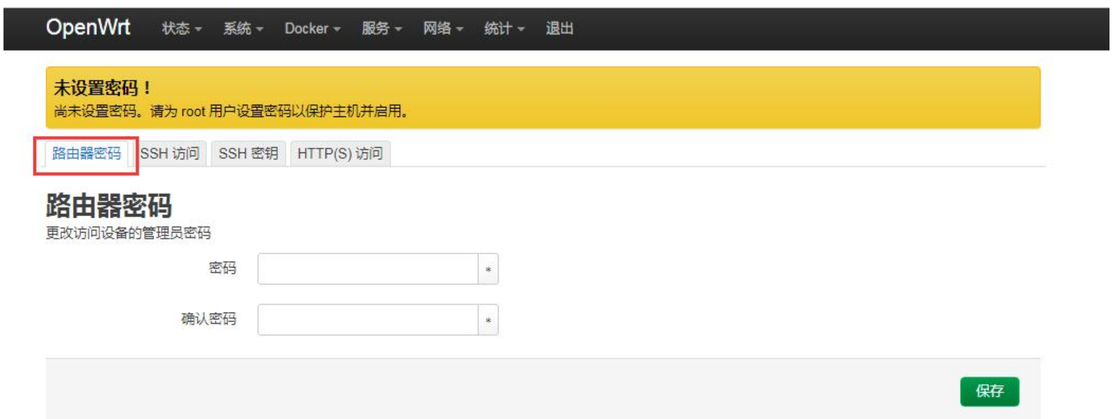
2) Then follow the steps below to change the password

a. Find the "**System**" option in the navigation bar and click

b. In the vertical column options below the system, select "**Management Rights**" and click



c. Select the "**Router Password**" option on the Tab page



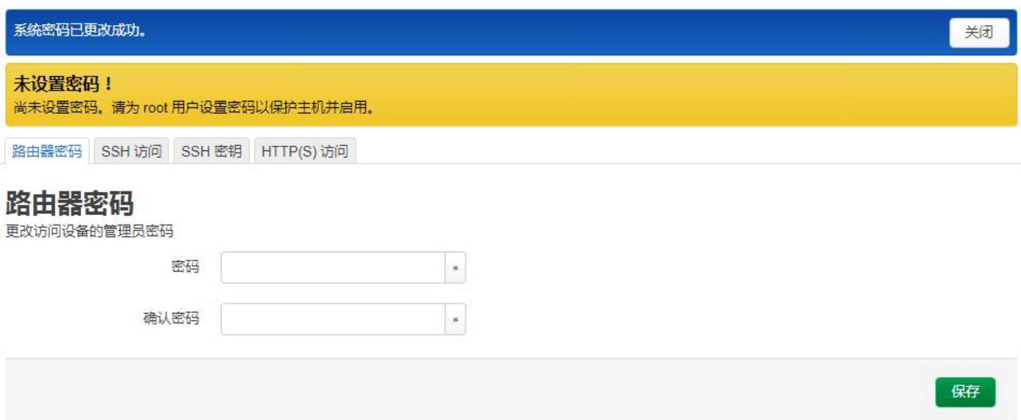
3) Modify and save the router password

a. Enter the password you set in the "**Password**" and "**Confirm Password**" dialog boxes (if you are not sure whether the password is entered correctly, you can click the "*" icon behind the dialog box to display the input characters)

b. Click "**Save**" to save the newly modified password.

Note: In the "Password" and "Confirm Password" dialog boxes, the passwords entered twice must be consistent.

4) After the password is successfully changed, a pop-up box showing "**System password has been changed successfully**" will pop up. At this time, you will need a password to log in to OpenWRT.



8.7. USB interface test

8.7.1. Mount USB storage device from command line

1) First insert the USB disk into the USB interface of the Orange Pi development board

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

```
root@OpenWrt:~# cat /proc/partitions | grep "sd*"
major minor    #blocks    name
    8              0      15126528 sda
```

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

```
root@OpenWrt:~# mount /dev/sda /mnt/
root@OpenWrt:~# ls /mnt/
```

```
test.txt
```

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

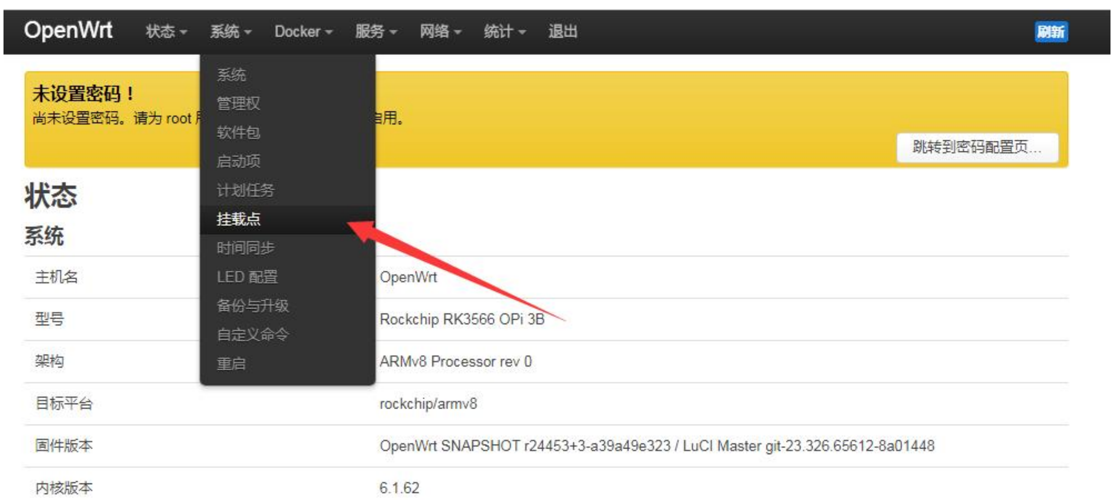
```
root@OpenWrt:~# df -h | grep "sd"
/dev/sda              14.4G      187.2M       14.2G     1% /mnt
```

8. 7. 2. Mount USB storage device in LuCI management interface

1) First connect the U disk (or other storage device) to the development board through USB2.0

2) Then log in to **the LuCI management interface** to enter the LuCI management interface.

3) Then in the LuCI management interface, click "System->Mount Point" to enter the mount point configuration interface



4) Then follow the steps below to add a mount point

a. Find "**Mount Point**" at the bottom of the **mount point global settings interface**.

b. Below the **mount point**, select the "Add" button and click to enter

c. The following pop-up window interface will pop up.

d. Then you can start mounting the storage device

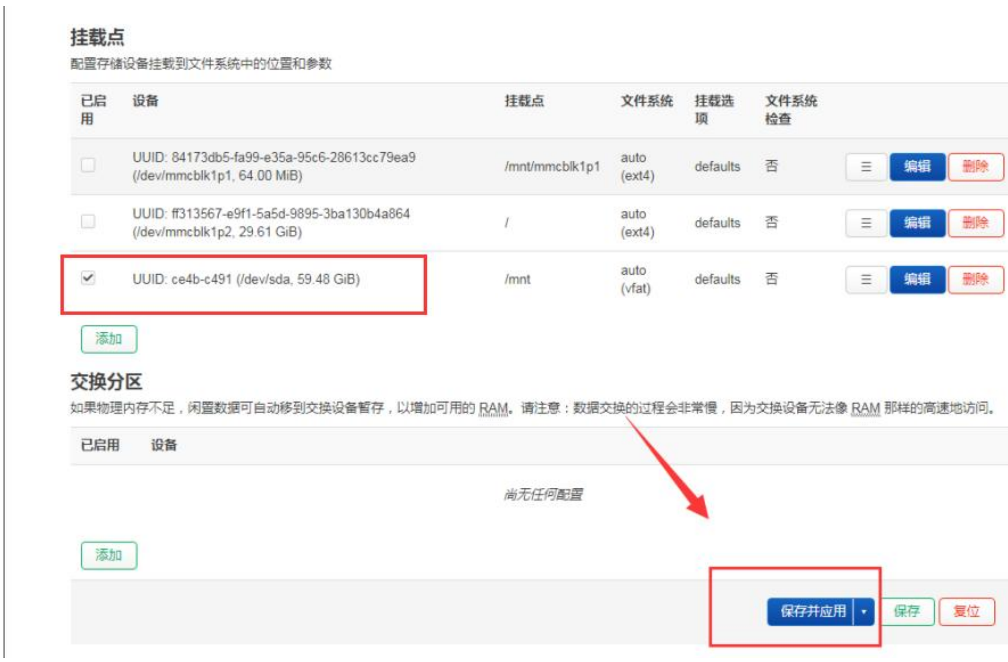
a) Check "**Enabled**"

b) Select the actual connected device /dev/sda in the General Settings UUID column (select according to your own device)

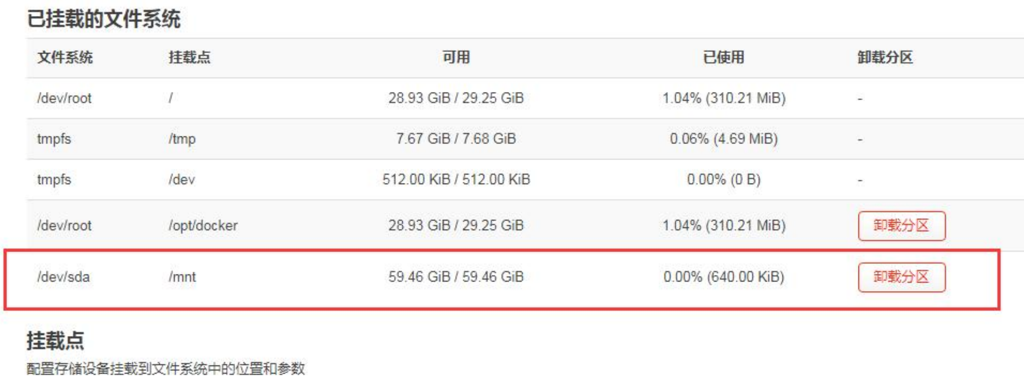
c) Select "**Custom**" in the mount point column and fill in the target directory to be mounted. Here, the **/mnt** directory is used as an example. After filling in, press **Enter** to confirm.

d) Then click the "**Save**" button in the lower right corner

5) Then you will return to the mount point global settings page. Click "Save and Apply" in the lower left corner of the page to make the mount point effective.



6) After saving, you can see that the storage device has been mounted successfully in "**Mounted File Systems**"



8. 8. USB wireless network card test

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

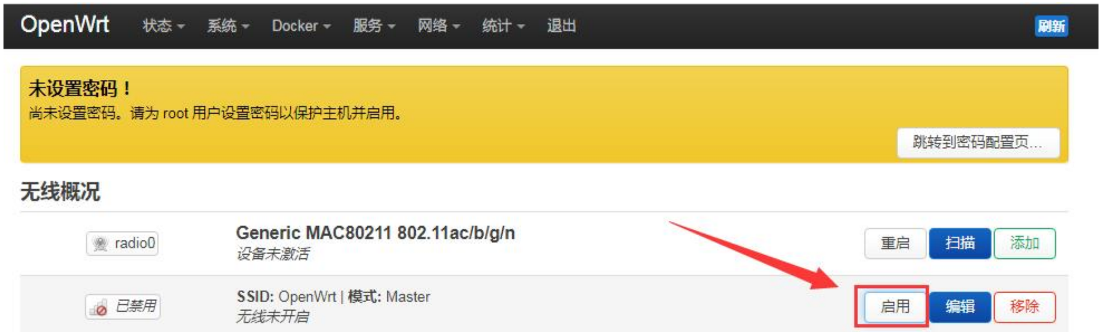
serial number	model	
1	RTL8811 Support 2.4G +5G WIFI	

8. 8. 1. Method to create WIFI hotspot using USB wireless network card

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power up the development board.

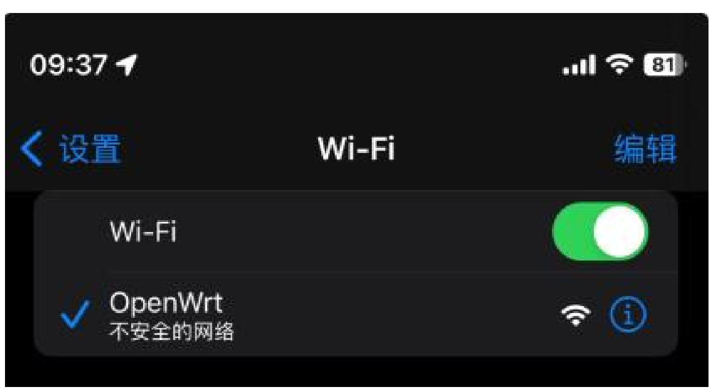
2）After the system starts, click **Network -> Wireless** to enter the wireless WiFi configuration interface. If there is no **wireless** option, it means that the USB wireless network card model is not supported by the system.

3）The default wireless configuration of the OpenWRT system is **Master** mode. We can directly click to **enable** to create an open WIFI hotspot.



4）The display interface of successfully creating a WIFI hotspot is as shown below

5）At this point, you can connect to the WIFI hotspot named **OpenWrt** through your mobile phone

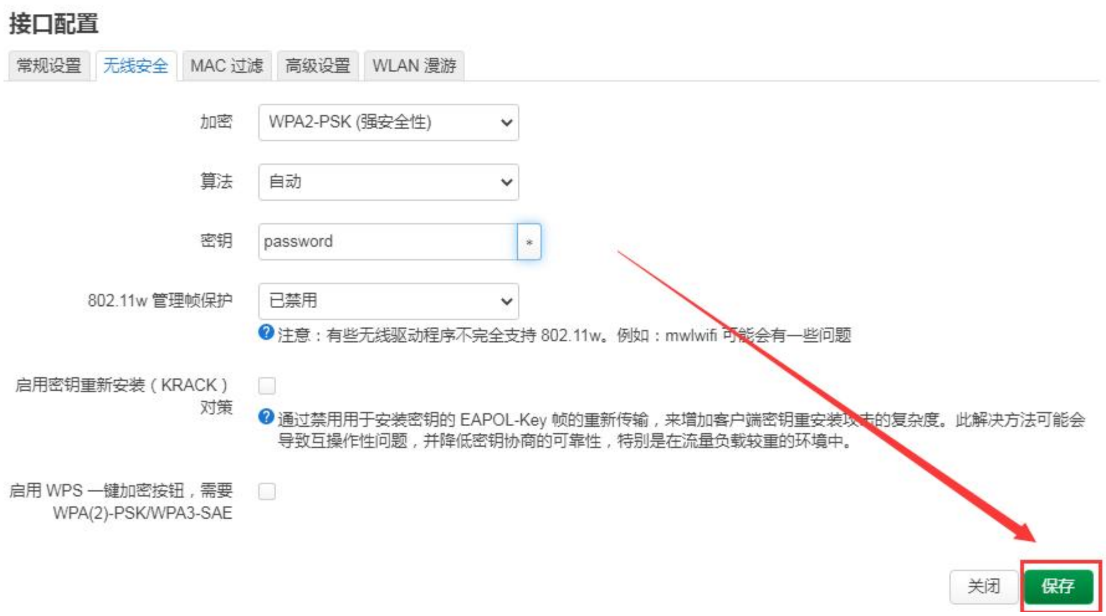


6）If you want to create a WIFI hotspot with a password, click the **Edit** option

7）In the pop-up tab, we click on the **Wireless Security** column.

8）Then in **Interface Configuration -> Wireless Security**, select **WPA2-PSK** as the encryption algorithm; set the key (wireless password) to **password**

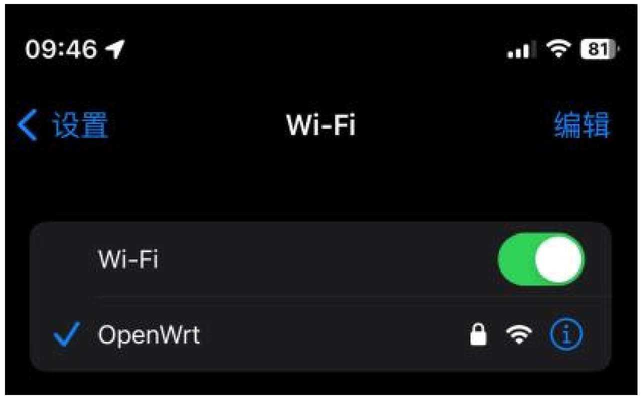
9）After the above settings are completed, click **Save** in the lower right corner of the page, and then exit the tab page



10）Then click **Save and Apply** in the lower right corner of the page and wait for the configuration to be applied.

11）The display interface of successfully creating a hotspot is as shown below. At this time, you can see that the WIFI hotspot has been encrypted.

12）Then use your mobile phone or computer to search for the WiFi corresponding to the SSID and connect. After the connection is successful, as shown in the figure below

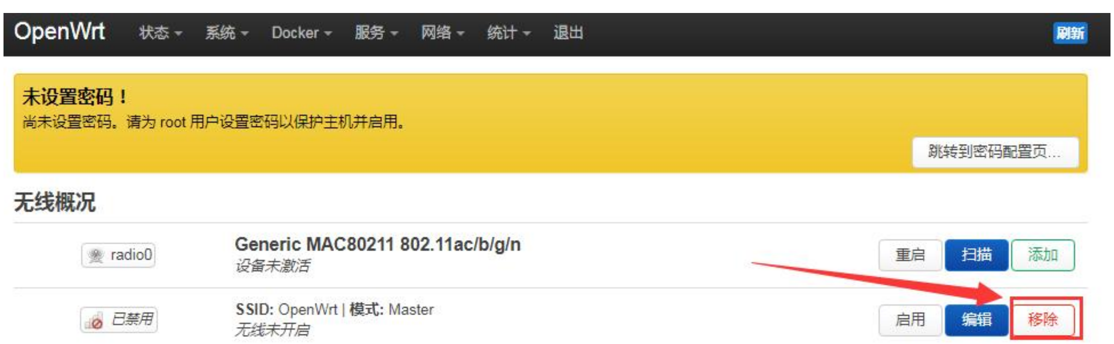


8.8.2. How to use USB wireless network card to connect to WIFI hotspot

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power up the development board.

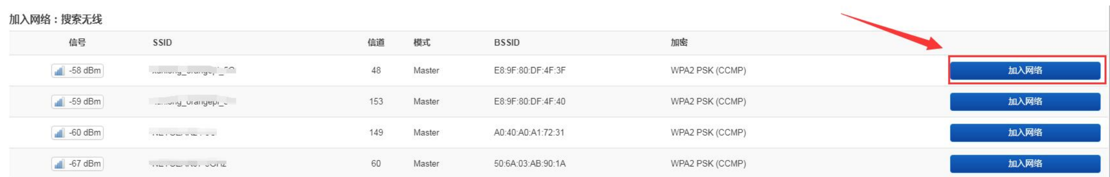
2）After the system startup is completed, click **"Network > Wireless"** to enter the wireless WiFi configuration interface.

3）First, you need to remove the default wireless configuration, click the **"Remove"** button as shown below

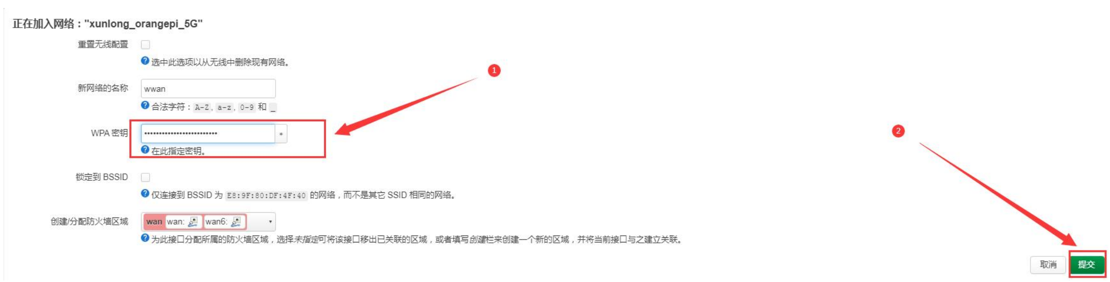


4）Then click the **"Scan"** button to scan the surrounding WiFi hotspots

5）Then the following window will pop up to display the available WiFi hotspots. Click the **"Join Network"** button to the right of the WiFi hotspot you want to connect to.



6）Then enter the password as shown in the picture below, and then click **"Submit"**



7）Then the following interface will pop up, click Save

8）Finally, you will return to the main interface of wireless configuration, click **"Save and Apply"**

9）After successfully connecting to the WiFi hotspot, the interface displays as shown below

8. 9. Installing packages via the command line

8. 9. 1. Install through opkg in the terminal

1) Update the list of available software packages

```
root@OpenWrt:/# opkg update
```

2) Get the software list

```
root@OpenWrt:/# opkg list
```

3) Install the specified software package

```
root@OpenWrt:/# opkg install <package name>
```

4) View installed software

```
root@OpenWrt:/# opkg list-installed
```

5) Uninstall the software

```
root@OpenWrt:/# opkg remove <package name>
```

8. 10. OpenWRT management interface installation package

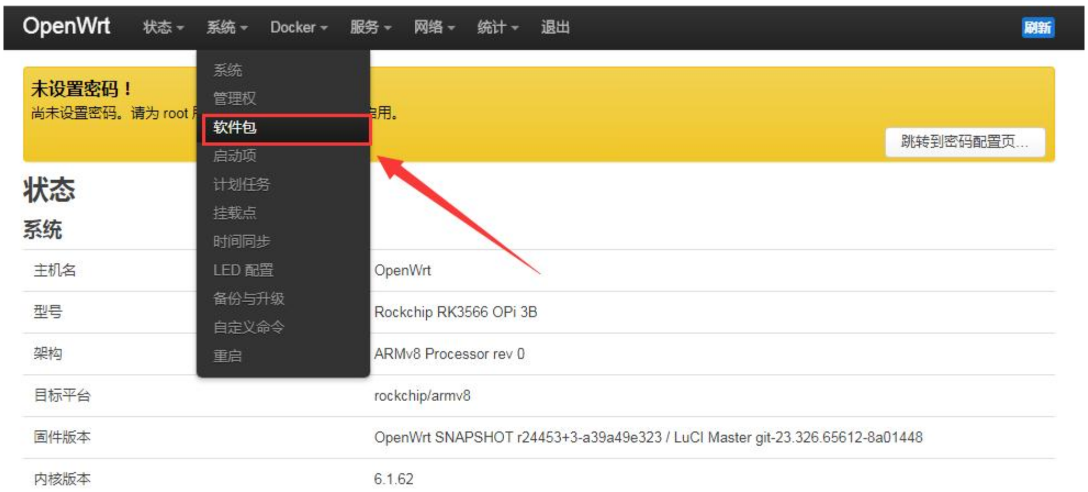
If you need to add a new software package, you can install it through the OpenWRT management interface.

8.10.1. View the list of available software packages on the system 系

1) First enter the software package management page

a. Find the "**System**" option in the navigation bar and click to enter

b. In the vertical column options below the system, select "**Software Package**" and click to enter



2) Then the main page of the software package will appear, as shown in the figure below, to obtain the list of available software

a. In the "**Action**" option of the software package, click "**Update List**" to obtain the list of available software packages.

b. In the Tab page, click "**Available**" to view the currently available software packages.

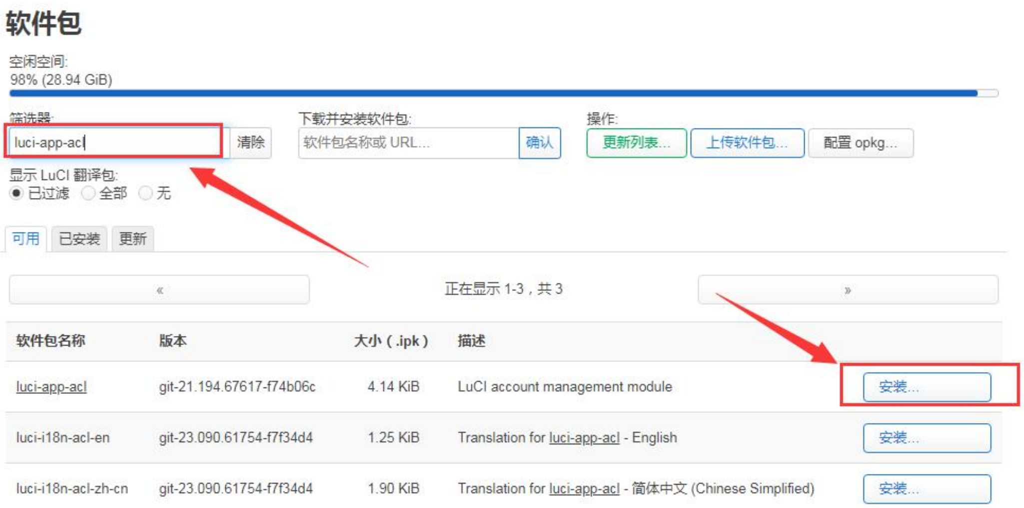
c. View the number of currently available software packages

8. 10. 2. Installation package example

1) Take the installation of the software package "**luci-app-acl**" as an example

   a. In the OpenWRT package management interface, click the filter dialog box and enter "**luci-app-acl**"

   b. In the list of software packages, you can see the version, package size and description information of the "**luci-app-acl**" software package, and then click the "**Install**" button



   c. Then the following pop-up window will appear, click "**Install**"

d. Then wait for the installation to complete

e. The installation completion display is as follows

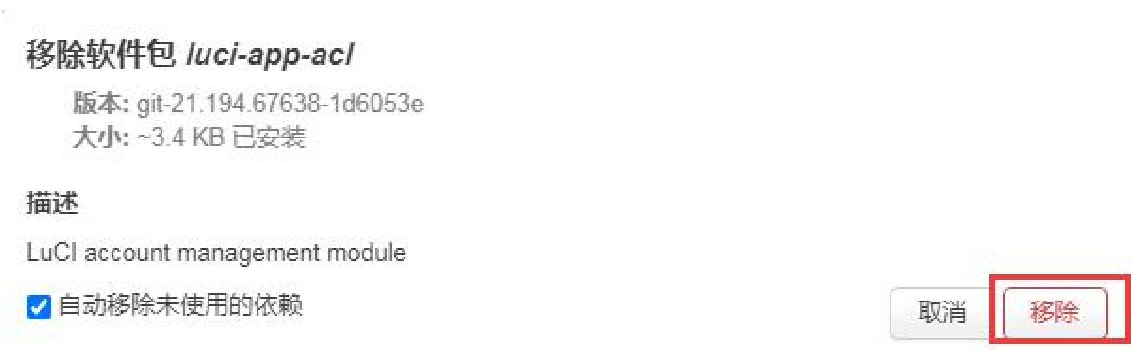
2) Check whether the software package is installed successfully
   a. In the OpenWRT package management interface, click the filter dialog box and enter "**luci-app-acl**"
   b. Select and click "**Available**" on the Tab page
   c. The "**luci-app-acl**" software package will be displayed in the software package list, and the status will be updated to "**Installed**"”

8. 10. 3. Example of removing software packages

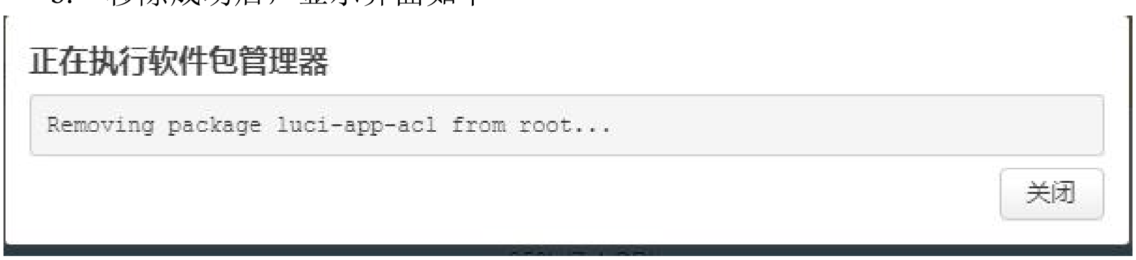
1) Take removing the software package "**luci-app-acl**" as an example
   a. In the OpenWRT package management interface, click the filter dialog box and enter "**luci-app-acl**"
   b. Select "**Installed**" on the Tab page to display the list of installed software packages.

c. Click "**Remove**" on the right to remove the corresponding software package

a. Then the pop-up window below will be displayed, click "**Remove**"



b. 移除成功后，显示界面如下



2) Check whether the software package was successfully removed

a. In the OpenWRT package management interface, click the filter dialog box and enter "**luci-app-acl**"

b. Select and click "**Installed**" on the Tab page

c. The "**luci-app-acl**" software package will not be displayed in the software package list. At this time, the "**luci-app-acl**" software package has been successfully removed.

8. 11. Using Samba network sharing

There are two main software options for OpenWRT LAN file sharing implementation, Samba and NFS. The Samba system has good compatibility, and NFS has superior performance. For users who need to use Windows devices, it is recommended to choose Samba.

1) Enter the management page of Samba network share
   a. Find the "**Service**" option in the navigation bar and click to enter
   b. In the vertical bar options below the service, select "**Network Sharing**" and click to enter

2) Select the interface that the Samba service needs to monitor
   a. Select "**General Settings**" in the navigation bar of network sharing and click to enter

b. The interface is specified according to actual needs. If you want to access through the "wan port", set it to "**wan**"

3) Set up a shared directory for network sharing

a. Click "**Add**" shared directory address in "**Shared Directory**" of "**General Settings**" of network sharing.

b. Enter the name of the shared folder as "**mmt**" under the name.

c. Under the path of the shared directory, select "/**mnt**" to set the shared directory location.

d. Check "**Browsable**" and "**Run anonymous user**"

e. Click "**Save and Apply**" to save the configuration

4) Windows 10 starts network discovery and sharing

Note: When accessing Samba and sharing under Windows 10 system, you need

to first confirm whether Windows 10 has enabled network discovery and sharing. If not, perform the following settings first.

a. Enable Samba v1/v2 access

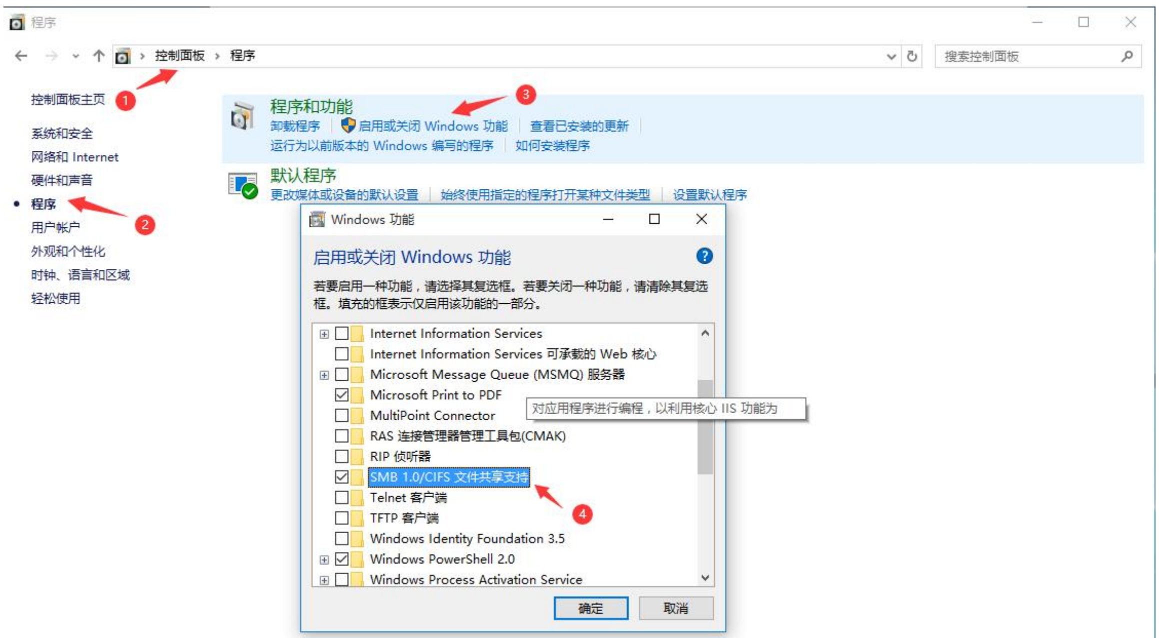
a) Enter the "Control Panel" of Windows 10

b) Click "Programs" on the left navigation bar of the control panel

c) Select "Turn Windows features on or off" in Programs and Features

d) Check "SMB 1.0/CIFS file sharing support" in the pop-up box to enable or disable Windows features.

e) Click "OK" to configure the application



b. Turn on network discovery in Windows 10

a) Enter the "Control Panel" of Windows 10

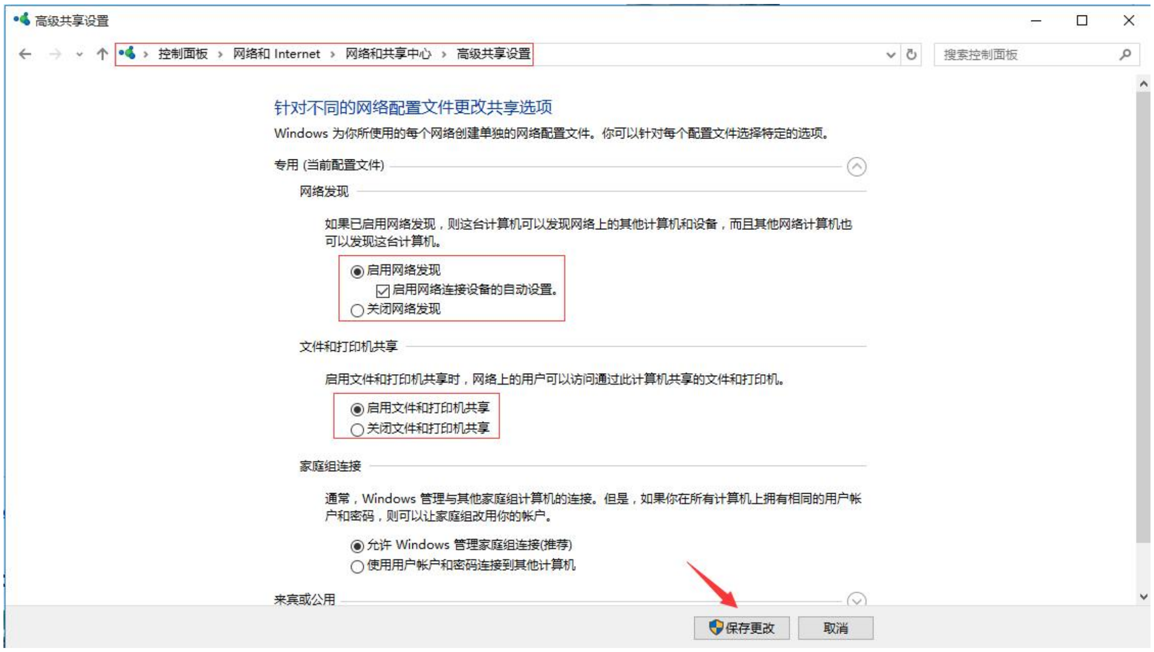
b) Select "Network and Internet" in the Control Panel

c) Then open "Network and Sharing Center"

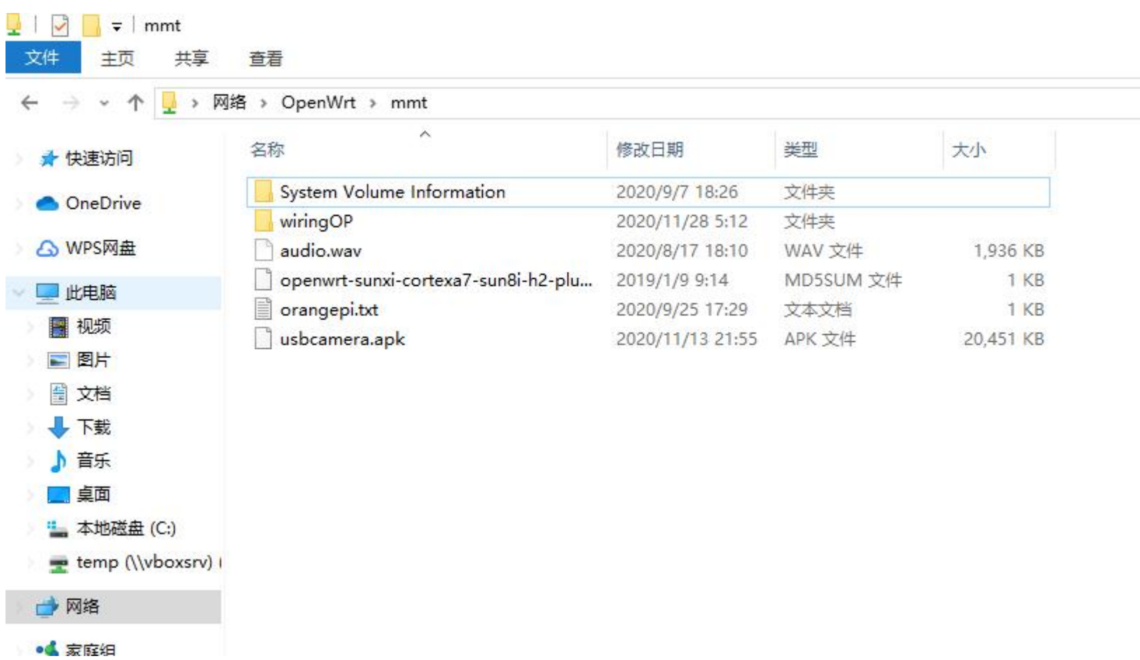
d) Click | "Advanced Sharing Settings"

e) Turn on "**Enable network discovery**" and "**Enable file and printer sharing**"

f) Click "Save Changes" to save the network discovery configuration of Windows 10



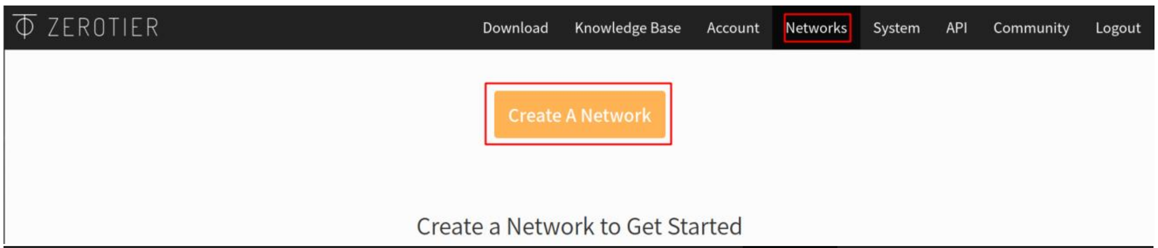
5) After the setting is completed, enter \\OpenWrt in the address bar of the resource manager to access the shared directory. The user name is root, and the password is the password set by the development board host.

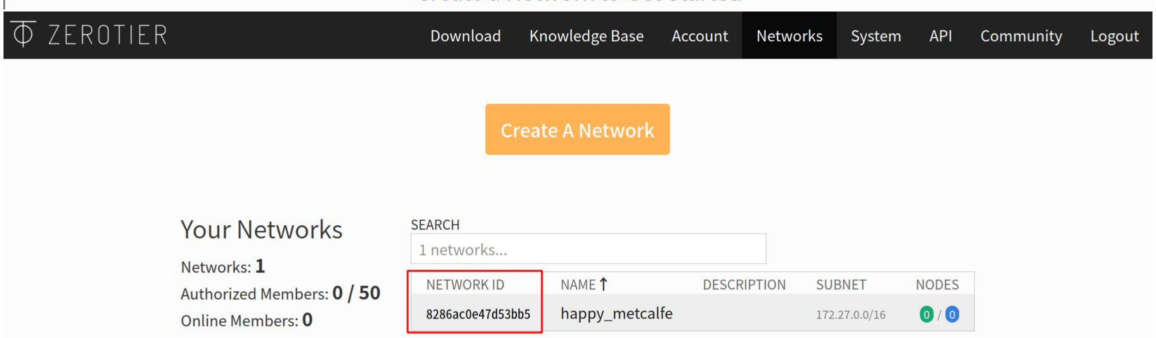


8. 12. Zerotier usage instructions

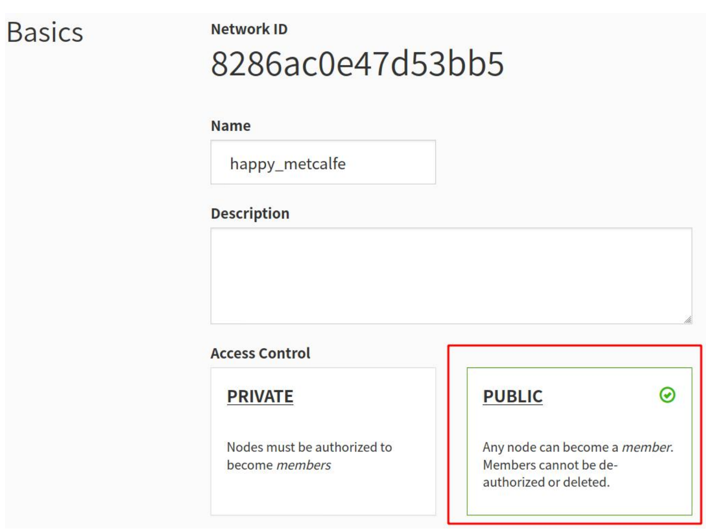
The OpenWRT system has been pre-installed with the zerotier client. After creating a virtual LAN on the zerotier official website, the client can directly join it through the Network ID. The specific operations are as follows.

1) Log in to zerotier official website **https://my.zerotier.com/network**, register and click Network->Create A Network to create a virtual LAN

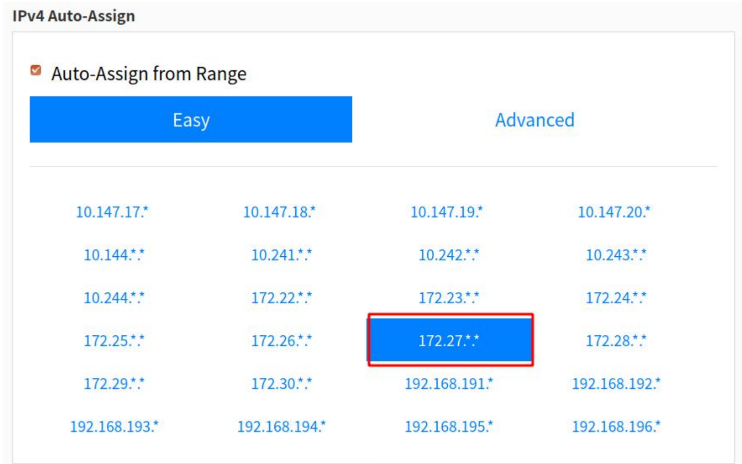




2) Click to enter the network console page and set the privacy option to public, so that the added network nodes do not need to be verified.



3) Next, you can choose the network segment for automatically assigning the address. The selected network segment here is 172.27.*.*



4) Enter the following command in the OpenWRT terminal to join the virtual LAN created above, **where 8286ac0e47d53bb5 is the Network ID of the virtual LAN created above**.

```
root@OpenWrt:/# zerotier-one -d                             #Start zerotier client
root@OpenWrt:/# zerotier-cli join 8286ac0e47d53bb5    #Join the network
```

5) Enter ifconfig in the terminal and you can see that there is a new **ztks54inm2** device with an IP address of **172.27.214.213**

```
root@OpenWrt:/# ifconfig
ztks54inm2 Link encap:Ethernet    HWaddr F6:4E:DE:BF:D8:52
           inet addr:172.27.214.213    Bcast:172.27.255.255    Mask:255.255.0.0
           inet6 addr: fe80::e82f:d0ff:fe5a:867e/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST    MTU:2800    Metric:1
           RX packets:18 errors:0 dropped:0 overruns:0 frame:0
           TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1720 (1.6 KiB)    TX byte81 (8.2 KiB)
```

6) Install the zerotier client on another device (Ubuntu 18.04 is used as an example here), execute the following command to install. After the installation is completed, you need to restart the computer.

```
test@ubuntu:~$ curl -s https://install.zerotier.com | sudo bash
```

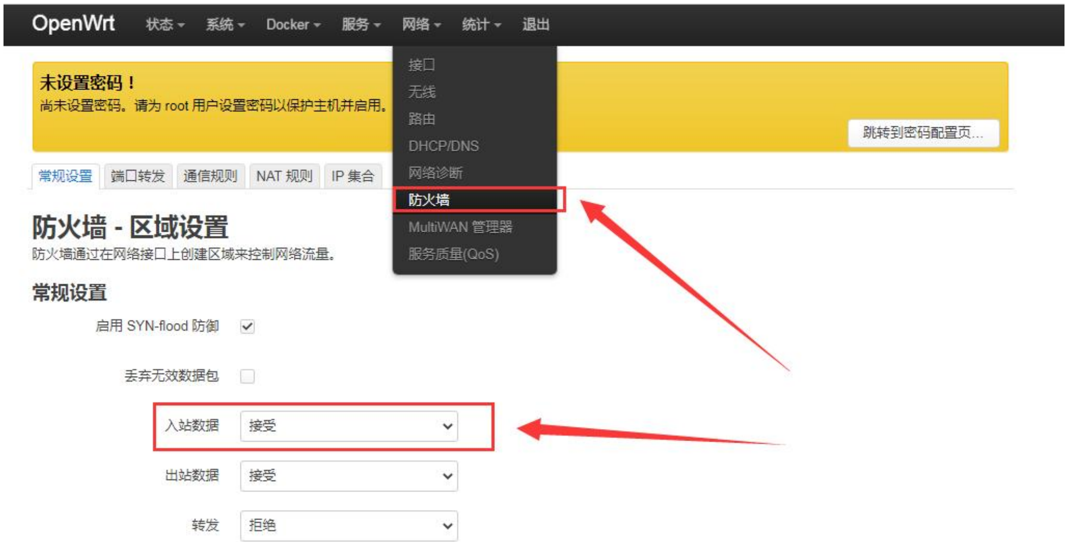
7) After restarting, join the virtual LAN according to the Network ID. You can also see that the IP address assigned by zerotier has been obtained. At this time, the Ubuntu PC and the development board are in the same LAN, and the two can communicate freely.

```
test@ubuntu:~$ sudo zerotier-cli join 8286ac0e47d53bb5
test@ubuntu:~$ ifconfig
ztks54inm2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2800
        inet 172.27.47.214  netmask 255.255.0.0  broadcast 172.27.255.255
        inet6 fe80::5ce1:85ff:fe2b:6918  prefixlen 64  scopeid 0x20<link>
        ether f6:fd:87:68:12:cf  txqueuelen 1000  (以太网)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 46  bytes 10006 (10.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

8) Test whether the two terminals can communicate

```
root@OpenWrt:/# ping 172.27.47.214 -I ztks54inm2
PING 172.27.47.214 (172.27.47.214): 56 data bytes
64 bytes from 172.27.47.214: seq=0 ttl=64 time=1.209 ms
64 bytes from 172.27.47.214: seq=1 ttl=64 time=1.136 ms
64 bytes from 172.27.47.214: seq=2 ttl=64 time=1.203 ms
64 bytes from 172.27.47.214: seq=3 ttl=64 time=1.235 ms
^C
--- 172.27.47.214 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.136/1.195/1.235 ms
```

9) If the computer cannot ping the development board, please log in to the LuCI interface and change the inbound data in the firewall configuration to accept



10) Other common commands of zerotier

```
root@OpenWrt:/# zerotier-one -d                          #Start zerotier client
root@OpenWrt:/# zerotier-cli status                      #Get address and service status
root@OpenWrt:/# zerotier-cli join # Network ID           #Join the network
root@OpenWrt:/# zerotier-cli leave # Network ID          #Leave the network
root@OpenWrt:/# zerotier-cli listnetworks                #list networks
OPENWRT_DEVICE_REVISION="v0"
OPENWRT_RELEASE="OpenWrt 22.03.4 r20123-38ccc47687"
```

9. How to compile Android11 source code

9. 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 -
```

9. 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
```

10. Compilation method of OpenWRT source code

10. 1. Download OpenWRT source code

1) First execute the following command to download the source code

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install -y git
test@test:~$ git clone https://github.com/orangepi-xunlong/openwrt.git -b main
```

2) After the OpenWRT code is downloaded, it will contain the following files and folders

```
test@test:~/openwrt$ ls
BSDmakefile   Config.in   include   Makefile   README.md   scripts   toolchain
Config   feeds.conf.default   LICENSE   package   rules.mk   target   tools
```

10. 2. Compile OpenWRT source code

1) First install the following dependent software (currently only tested on Ubuntu22.04, you need to install the following software. If you compile on other versions of the system, please install the dependent software yourself according to the error message)

```
test@test:~/openwrt$ sudo apt update
test@test:~/openwrt$ sudo apt install -y ack antlr3 asciidoc autoconf    \
automake autopoint binutils bison build-essential    \
bzip2 ccache cmake cpio curl device-tree-compiler fastjar \
flex gawk gettext gcc-multilib g++-multilib git gperf haveged \
help2man intltool libc6-dev-i386 libelf-dev libglib2.0-dev \
libgmp3-dev libltdl-dev libmpc-dev libmpfr-dev \
libncurses5-dev \libncursesw5-dev libreadline-dev libssl-dev \
libtool lrzsz mkisofs msmtp nano ninja-build p7zip p7zip-full \
patch pkgconf python2.7 python3 python3-pyelftools \
libpython3-dev qemu-utils rsync scons squashfs-tools \
subversion swig texinfo uglifyjs upx-ucl unzip \
vim wget xmlto xxd zlib1g-dev
```

2) Then execute **./scripts/feeds update -a** and **./scripts/feeds install -a** to download dependency packages

```
test@test:~/openwrt$ ./scripts/feeds update -a
test@test:~/openwrt$ ./scripts/feeds install -a
```

3) Then choose to use the configuration file of OrangePi 3B

```
test@test:~/openwrt$ cp configs/orangepi-3b-rk3566_defconfig .config
```

4) Then execute the following command to make the configuration take effect

```
test@test:~/openwrt$ make defconfig
```

5) Execute the following command to start compiling the openwrt source code

```
test@test:~/openwrt$ make V=s
```

6) After compilation is completed, the path where the image is generated is:

```
test@test:~/openwrt$ tree -L 1 bin/targets/rockchip/armv8/
```

```
bin/targets/rockchip/armv8/
├── config.buildinfo
├── feeds.buildinfo
├── openwrt-rockchip-armv8-xunlong_orangepi-3b-ext4-sysupgrade.img.gz
├── openwrt-rockchip-armv8-xunlong_orangepi-3b.manifest
├── openwrt-rockchip-armv8-xunlong_orangepi-3b-squashfs-sysupgrade.img.gz
├── packages
├── profiles.json
├── sha256sums
└── version.buildinfo

1 directory, 9 files
```

11. Appendix

11. 1. User Manual Update History

Version	Date	Update Notes
v1.0	2023-08-17	initial version
v1.1	2023-08-24	1. Compilation method of Android11 source code 2. Android11：The method of USB OTG mode switching 3. Android11：The method of using the data cable to connect to adb debugging
v1.2	2023-08-25	1. Added instructions for purchasing PCIe NVMe SSDs
v1.3	2023-09-05	1. Instructions for using the Orange Pi OS Arch system
v1.4	2023-09-21	1. Linux：How to create a WIFI hotspot through create_ap
v1.5	2023-11-08	1. How to burn Orange Pi OS (OH) image to TF card 2. How to burn Orange Pi OS (OH) image into eMMC 3. Orange Pi OS OH system usage instructions

		4. Added instructions for Linux 6.6 system
v1.6	2023-11-24	1. How to use wiringOP hardware PWM 2. Instructions for using the OpenWRT system 3. How to compile OpenWRT source code

11. 2. Image Update History

Date	Update Notes
2023-08-17	Orangepi3b_1.0.0_ubuntu_focal_server_linux5.10.160.7z Orangepi3b_1.0.0_ubuntu_jammy_server_linux5.10.160.7z Orangepi3b_1.0.0_debian_bullseye_server_linux5.10.160.7z Orangepi3b_1.0.0_debian_bookworm_server_linux5.10.160.7z Orangepi3b_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.0_debian_bookworm_desktop_xfce_linux5.10.160.7z OrangePi3B_RK3566_Android11_v1.0.0.tar.gz OrangePi3B_RK3566_Android11_lcd_v1.0.0.tar.gz OrangePi3B_RK3566_Android11_spi-nvme_v1.0.0.tar.gz OrangePi3B_RK3566_Android11_lcd_spi-nvme_v1.0.0.tar.gz * initial version
2023-08-23	Opios-arch-aarch64-xfce-opi3b-23.08-linux5.10.160.img.xz * initial version
2023-08-24	OrangePi3B_RK3566_Android11_v1.0.1.tar.gz OrangePi3B_RK3566_Android11_lcd_v1.0.1.tar.gz OrangePi3B_RK3566_Android11_spi-nvme_v1.0.1.tar.gz OrangePi3B_RK3566_Android11_lcd_spi-nvme_v1.0.1.tar.gz * Support USB OTG mode switching function
2023-08-25	Opios-arch-aarch64-xfce-opi3b-23.08.1-linux5.10.160.img.xz * Solve the problem that the app store cannot be used

2023-09-21	Orangepi3b_1.0.2_ubuntu_focal_server_linux5.10.160.7z Orangepi3b_1.0.2_ubuntu_jammy_server_linux5.10.160.7z Orangepi3b_1.0.2_debian_bullseye_server_linux5.10.160.7z Orangepi3b_1.0.2_debian_bookworm_server_linux5.10.160.7z Orangepi3b_1.0.2_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.2_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepi3b_1.0.2_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepi3b_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
2023-11-06	Opios-openharmony-4.0-beta1-aarch64-opi3b-23.11-linux5.10.img.tar.gz * initial version
2023-11-08	Orangepi3b_1.0.0_ubuntu_jammy_desktop_xfce_linux6.6.0-rc50.7z Orangepi3b_1.0.0_debian_bullseye_desktop_xfce_linux6.6.0-rc5.7z Orangepi3b_1.0.0_debian_bookworm_desktop_xfce_linux6.6.0-rc5.7z * initial version
2023-11-08	Opios-oh-4.0-beta1-aarch64-opi3b-23.11.1-linux5.10.img.tar.gz * Support TF card boot
2023-11-24	openwrt-aarch64-opi3b-23.05-linux6.1.62-ext4.img.gz * initial version