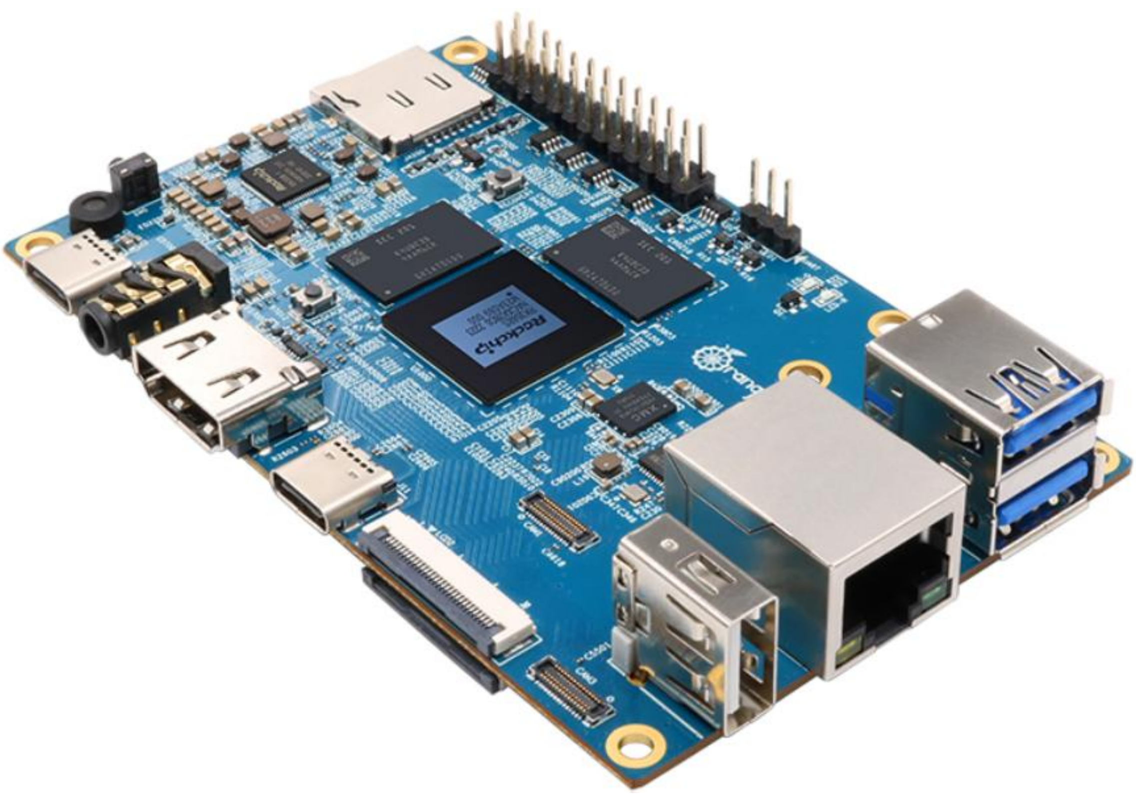
# Orange Pi 5
# User Manual



# Contents

# User Manual Update History

| Version | date | Release Notes |
|---|---|---|
| v0.1 | 2022-12-02 | initial version |
| v0.2 | 2022-12-05 | 1. How to burn Linux image to SPIFlash+NVMe SDD<br>2. Linux: How to upload files to the Linux system of the development board<br>3. Linux: How to download and install arm64 version balenaEtcher<br>4. How to burn Orange Pi OS (Droid) image to TF card<br>5. Burn Orange Pi OS (Droid) image to SPIFlash+NVMe SDD<br>6. Linux: How to log in to the desktop of the Linux system remotely |
| v0.3 | 2022-12-09 | 1. How to compile Android 12 source code<br>2. Linux: orangepi-build instructions<br>3. Linux: How to use adb |
| v0.4 | 2022-12-12 | 1.Linux: How to use SATA SSD<br>2. How to write Linux image to SPIFlash+SATA SDD<br>3. Linux: Test method of RTL8821CU USB WIFI module<br>4. Debian: How to set up Chinese environment and install Chinese input method |
| v0.5 | 2022-12-16 | 1.Linux: How to use AP6275P PCIe network card<br>2.Linux: How to install QT<br>3. How to install ROS 1 Noetic on Ubuntu 20.04<br>4. How to install ROS 2 Galactic on Ubuntu 20.04<br>5. How to install ROS 2 Humble on Ubuntu 22.04 |
| v0.6 | 2022-12-23 | 1.Linux: Method of using commands to test recording<br>2.Linux: How to install kernel header files<br>3. Linux: How to use the 10.1-inch MIPI LCD screen<br>4.Ubuntu20.04: How to set Chinese and Chinese input methods in the system<br>5.Ubuntu22.04: How to set Chinese and Chinese input methods in the system<br>6.Android12: How to burn Android image to SPIFlash+SATA SDD<br>7. Android12: How to use USB wireless network card<br>8.Android12: 26pin interface GPIO, UART, SPI and PWM test |
| v0.7 | 2023-01-06 | 1. How to burn Linux image to SPIFlash+USB storage device |

| | | |
|---|---|---|
| | | 2. Linux: How to install and use wiringOP-Python<br>3. Linux: Instructions for using the logo on and off<br>4.Linux: AP6275P PCIe network card creates WIFI hotspot method through create_ap<br>5. Ubuntu22.04: Instructions for using orangepi-build to compile the image on the development board |

# image update history

| date | Release Notes |
|---|---|
| 2022-12-02 | Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.7z<br><br>* initial version |
| 2022-12-05 | Orangepi5_1.0.2_debian_bullseye_desktop_xfce_linux5.10.110.7z<br><br>* Pre-installed with balenaEtcherhe and Gparted<br>* Pre-installed ffmpeg and mpv player<br>* Add some scripts and configuration files |
| 2022-12-09 | Orangepi5_1.0.2_debian_bullseye_server_linux5.10.110.7z<br>Orangepi5_1.0.2_ubuntu_jammy_server_linux5.10.110.7z<br>Orangepi5_1.0.2_ubuntu_jammy_desktop_xfce_linux5.10.110.7z<br><br>* initial version |
| 2022-12-12 | Orangepi5_1.0.4_debian_bullseye_server_linux5.10.110.7z<br>Orangepi5_1.0.4_debian_bullseye_desktop_xfce_linux5.10.110.7z<br>Orangepi5_1.0.4_ubuntu_jammy_server_linux5.10.110.7z<br>Orangepi5_1.0.4_ubuntu_jammy_desktop_xfce_linux5.10.110.7z<br><br>* Add rk3588-ssd-sata.dtbo<br>* Add rkspi_loader_sata.img to start the linux system on sata ssd<br>* Pre-installed usb-modeswitch package, test RLT8821CU WIFI module can be used normally |
| 2022-12-16 | Orangepi5_1.0.6_debian_bullseye_desktop_xfce_linux5.10.110.7z<br><br>* Support ov13855 camera<br>* Support open multiple mipi cameras at the same time |

| | |
|---|---|
| | * Test that qt can be installed and used normally<br>* Some scripts are pre-installed<br><br>Orangepi5_1.0.6_ubuntu_jammy_desktop_xfce_linux5.10.110.7z<br><br>* Test that ros can be installed and used normally<br>* Test that qt can be installed and used normally<br>* Some scripts are pre-installed<br>* Fix the bug that the fcitx5 configuration program cannot be opened<br><br>Orangepi5_1.0.6_ubuntu_focal_server_linux5.10.110.7z<br>Orangepi5_1.0.6_ubuntu_focal_desktop_xfce_linux5.10.110.7z<br><br>* initial version |
| 2022-12-23 | Orangepi5_1.0.8_debian_bullseye_server_linux5.10.110.7z<br>Orangepi5_1.0.8_ubuntu_jammy_server_linux5.10.110.7z<br>Orangepi5_1.0.8_ubuntu_focal_server_linux5.10.110.7z<br>Orangepi5_1.0.4_debian_bullseye_desktop_xfce_linux5.10.110.7z<br>Orangepi5_1.0.4_ubuntu_jammy_desktop_xfce_linux5.10.110.7z<br><br>* Some scripts are pre-installed<br>* Solve the error problem of kernel header file deb package installation<br><br>Orangepi5_1.0.6_ubuntu_focal_desktop_xfce_linux5.10.110.7z<br><br>* Some scripts are pre-installed<br>* Solve the error problem of kernel header file deb package installation<br>* Solve the problem that the mipi camera cannot be used<br><br>OrangePi5_RK3588S_Android12_v1.0.1.img<br>OrangePi5_RK3588S_Android12_lcd_v1.0.1.img<br>OrangePi5_RK3588S_Android12_spi-nvme_lcd_v1.0.1.img<br>OrangePi5_RK3588S_Android12_spi-nvme_v1.0.1.img<br><br>* Support OV13855 camera |

| | |
|---|---|
| | * Support RTL8211CU, RTL8822CU, RTL8723BU three USB wireless network cards<br>* Enable UART0, I2C5, SPI4, PWM15 by default<br>* Pre-installed WiringOP APP is used to operate GPIO, I2C, SPI and UART hardware resources<br><br>OrangePi5_RK3588S_Android12_spi-sata_v1.0.1.img<br>OrangePi5_RK3588S_Android12_spi-sata_lcd_v1.0.1.img<br><br>* initial version |
| 2023-01-06 | Orangepi5_1.1.0_debian_bullseye_server_linux5.10.110.7z<br>Orangepi5_1.1.0_ubuntu_focal_server_linux5.10.110.7z<br>Orangepi5_1.1.0_ubuntu_jammy_server_linux5.10.110.7z<br><br>* Pre-install create_ap, support AP6275P PCIe network card to open hotspot function<br>* Support SPIFlash+USB storage device to start Linux system (only USB3.0 interface)<br>* Open some kernel configuration<br><br>Orangepi5_1.1.0_debian_bullseye_desktop_xfce_linux5.10.110.7z<br>Orangepi5_1.1.0_ubuntu_focal_desktop_xfce_linux5.10.110.7z<br>Orangepi5_1.1.0_ubuntu_jammy_desktop_xfce_linux5.10.110.7z<br><br>* Add switch to display logo<br>* Set VOP DCLK as dynamic allocation strategy<br>* Pre-install create_ap, support AP6275P PCIe network card to open hotspot function<br>* Support SPIFlash+USB storage device to start Linux system (only USB3.0 interface)<br>* Optimize the set_lcd_rotate.sh script to solve the unusable problem in Debian11<br>* Open some kernel configuration |

# 1. Basic features of Orange Pi 5

## 1. 1. What is Orange Pi 5

Orange Pi 5 adopts Rockchip RK3588S new-generation octa-core 64-bit ARM processor, specifically quad-core A76 and quad-core A55, using Samsung 8nm LP process technology, large-core main frequency up to 2.4GHz, integrated ARM Mali -G610 MP4 GPU, embedded with high-performance 3D and 2D image acceleration modules, built-in AI accelerator NPU with a computing power of up to 6 Tops, has 4GB/8GB/16GB/32GB (LPDDR4/4x) memory, and has up to 8K display processing capabilities.

Orange Pi 5 brings out quite a lot of interfaces, including HDMI output, Type-C, M.2 PCIe2.0x1, Gigabit Ethernet port, USB2.0, USB3.0 interface and 26pin 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 5 supports Orange Pi OS, the official operating system developed by Orange Pi. At the same time, it supports Android 12.1, Debian11, Ubuntu20.04 and Ubuntu22.04 and other operating systems.

## 1. 2. Purpose of Orange Pi 5

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, because the Orange Pi 5 development board can install Linux systems such as Debian and Ubuntu, and systems such as Android, which means that we can implement it within the scope of the development board hardware and software support. Various functions.**

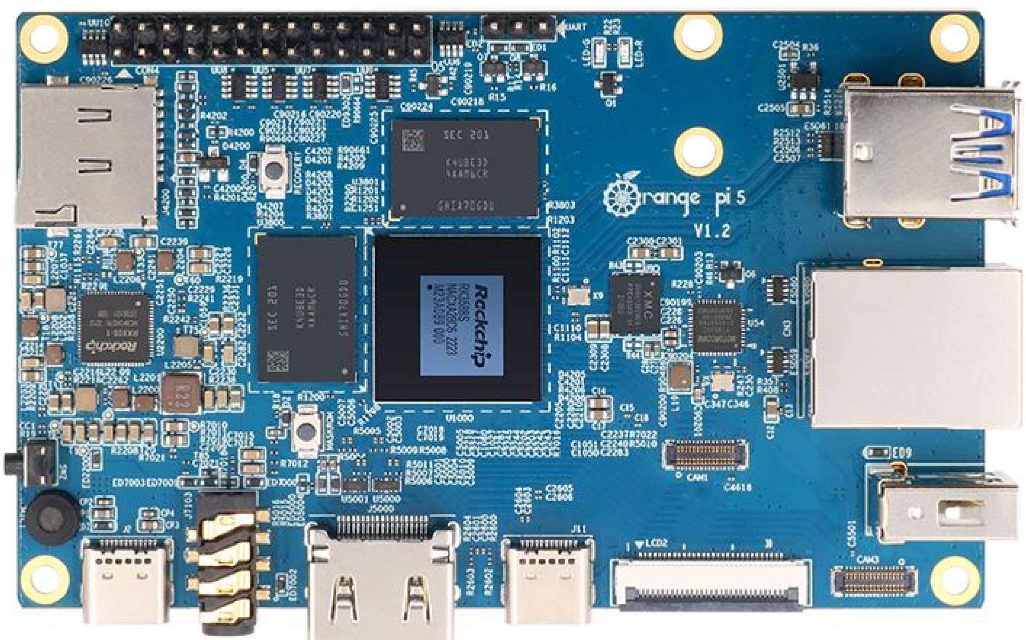
## 1. 3. Hardware features of Orange Pi 5

| **Introduction to hardware features** | |
|---|---|
| CPU | • Rockchip RK3588S (8nm LP process)<br>• 8-core 64-bit processor<br>• 4-core Cortex-A76 and 4-core Cortex-A55 core architecture<br>• The main frequency of the large core is up to 2.4GHz, and the main frequency of the small core is up to 1.8GHz |
| GPU | • Integrated ARM Mali-G610<br>• OpenGL ES1.1/2.0/3.2, OpenCL 2.2 and Vulkan 1.2 |
| NPU | • Built-in AI accelerator NPU with a computing power of up to 6 Tops<br>• Support INT4/INT8/INT16 mixed operation |
| video output | • HDMI 2.1, up to 8K @60Hz<br>• DP1.4（DisplayPort）<br>• 2 * MIPI D-PHY TX 4Lane |
| Memory | 4GB/8GB/16GB/32GB（LPDDR4/4x） |
| camera | • 1 * MIPI CSI 4Lane<br>• 2 * MIPI D-PHY RX 4Lane |
| PMU | RK806-1 |
| onboard storage | • 16MB QSPI Nor FLASH<br>• MicroSD (TF) Card slot<br>• PCIe2.0x1 M.2 M-KEY (SSD) slot |
| ethernet | 10/100/1000Mbps Ethernet (YT8531C ) |
| audio | • 3.5mm headphone jack audio in/out<br>• Onboard MIC input<br>• HDMI output |
| PCIe M.2 M-KEY | • Support PCIe WIFI6+BT5.0+BLE<br>• Support SSD |
| USB interface | 1 * USB3.0 interface<br>2 * USB2.0 interface (one of which is shared with Type-C |

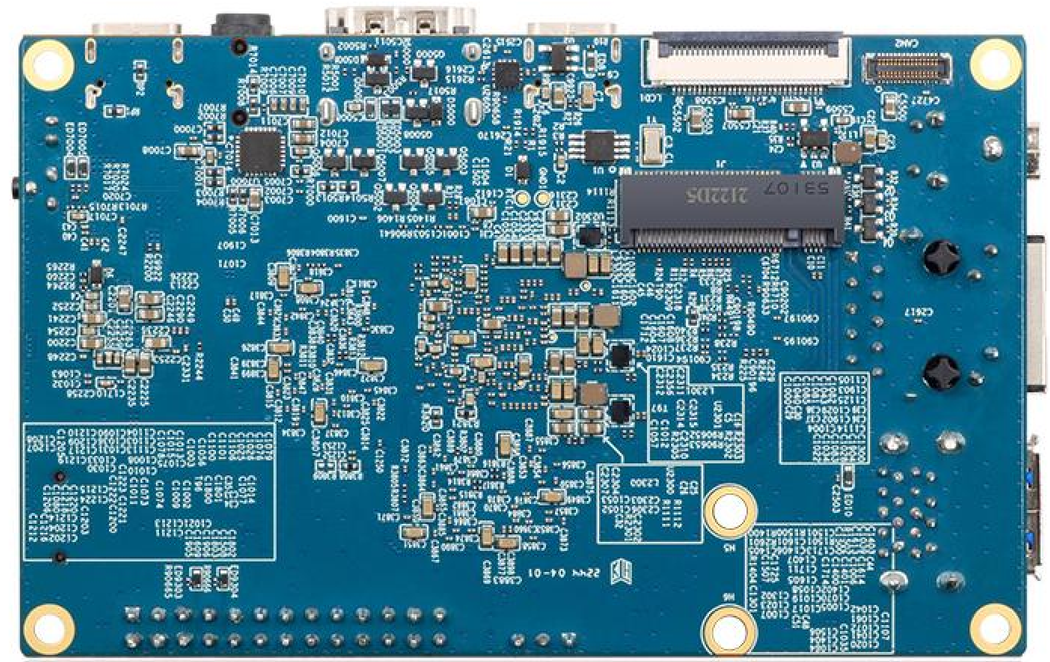
| | |
|---|---|
| | interface)<br>1 * USB3.0 Type-C interface |
| 26pin extension header | Used to expand UART, PWM, I2C, SPI, CAN and GPIO interfaces |
| debug serial port | 3pin debugging serial port |
| LED light | Power light and status light |
| button | 1 * MaskROM key, 1 * RECOVERY, 1 * switch key |
| Powered Source | Type-C power supply 5V/4A; |
| Supported OS | Orange Pi OS (Droid), Android12.1, Debian11 and other operating systems |
| **Introduction of Appearance Specifications** | |
| Product Size | 100mm*62mm |
| weight | 46g |
| range Pi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd. | |

## 1.4. Top view and bottom view of Orange Pi 5

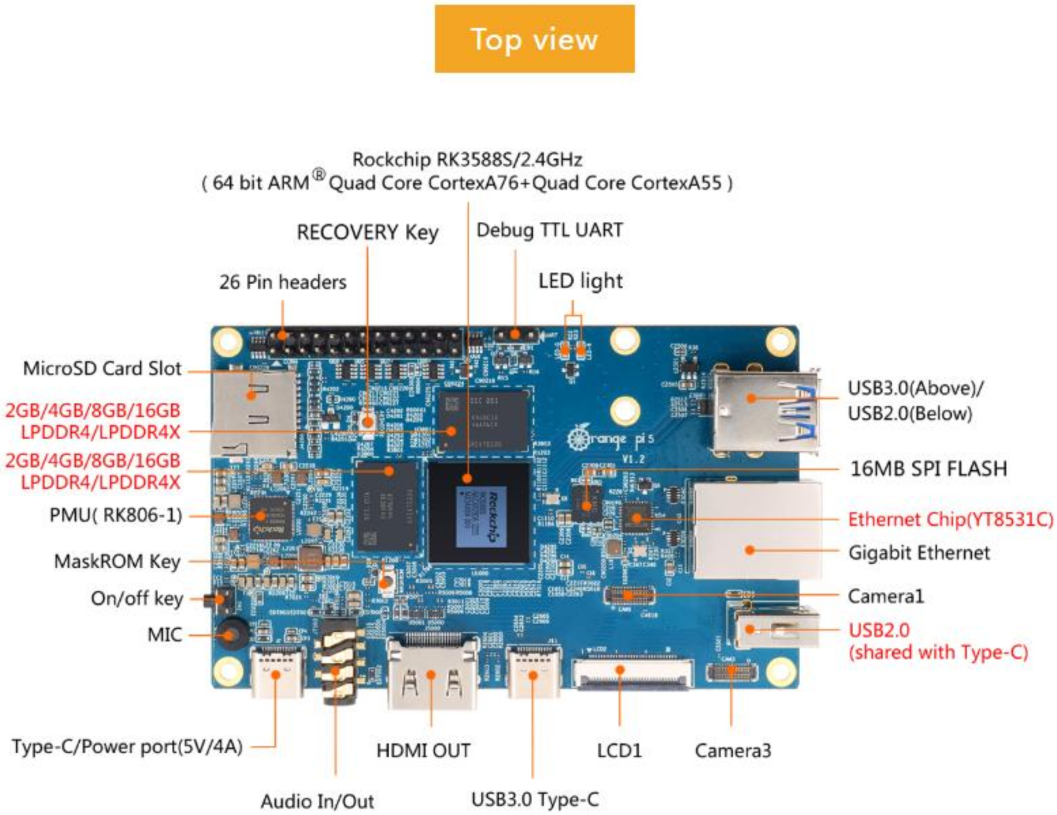
**Top view:**



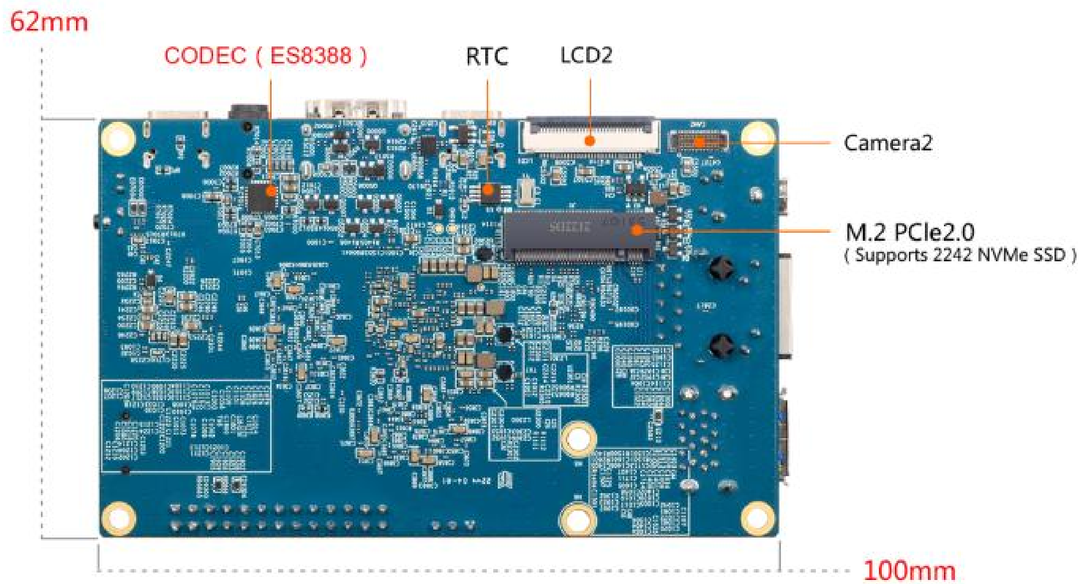

**Bottom view:**

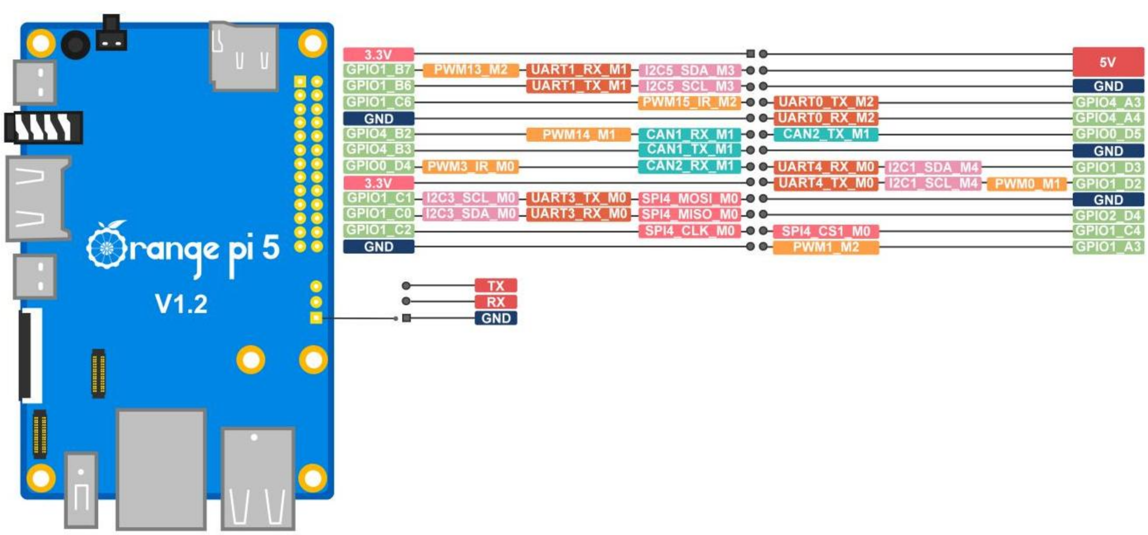


## 1.5. Interface details of Orange Pi 5



Bottom view




| |
|---|
| **The diameter of the four positioning holes is 3.0mm, and the diameter of the two M.2 PICE device fixing holes is 3.5mm.** |

# 2. Introduction to the use of the development board

## 2. 1. Prepare the required accessories

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


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


3) Display with HDMI interface


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

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

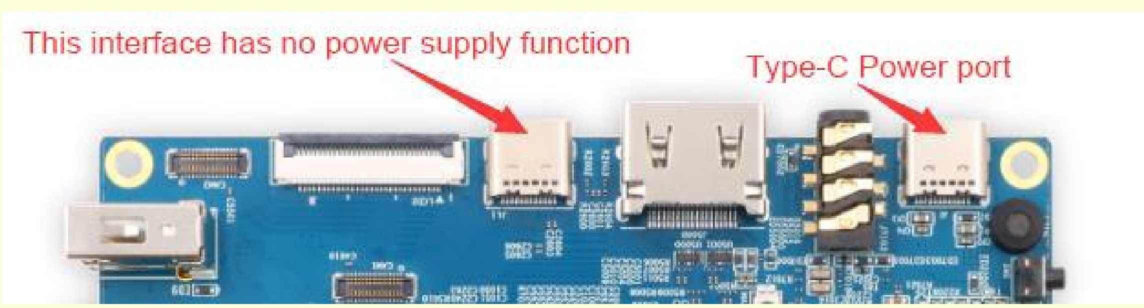
5) Type-C to HDMI cable, connect the development board to an HDMI monitor or TV for display through the Type-C interface


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

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

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

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




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

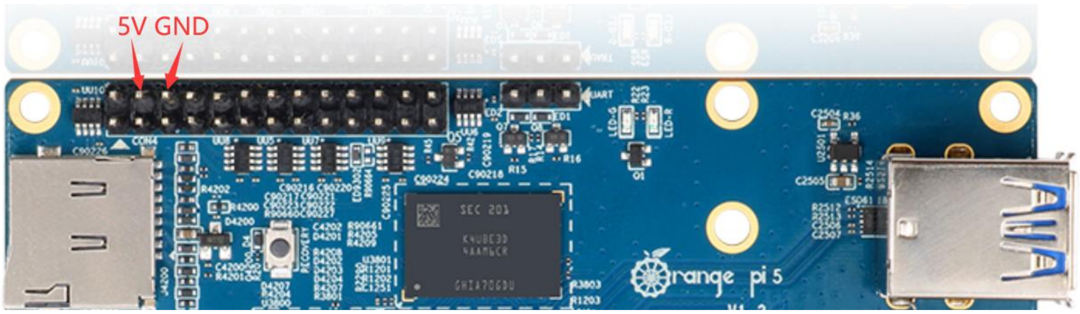
9) The mouse and keyboard of the USB interface, as long as the mouse and keyboard of the standard USB interface are acceptable, the mouse and keyboard can be used to control the Orange Pi development board


10) USB camera


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

**Note that the 5V pin on the 26pin pin header can be used directly after the development board is plugged into the power supply of the Type-C interface. No other settings are required. In addition, the output voltage of the 5V pin on the 26pin pin header cannot be adjusted and turned off by software. (no PWM**

**function).**




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

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

14) OV13850 camera with 13 million MIPI interface (picture to be added)

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

16) **3.3V** USB to TTL module and DuPont line, when using the serial port debugging function, need USB to TTL module and DuPont line to connect the development board and computer

17) 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 Chinese version is:

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

2) The website for downloading the English version is:

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

3) The information mainly includes

a. **Android source code:** saved on Google Drive
b. **Linux source code:** saved on Github
c. **User manual and schematic diagram:** saved on Google Drive
d. **Official tools:** mainly include the software that needs to be used during the use of the development board
e. **Android image:** saved on Google Drive
f. **Ubuntu image:** saved on Google Drive
g. **Debian image:** saved on Google Drive

## 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 image of Linux distributions such as Debian or Ubuntu 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

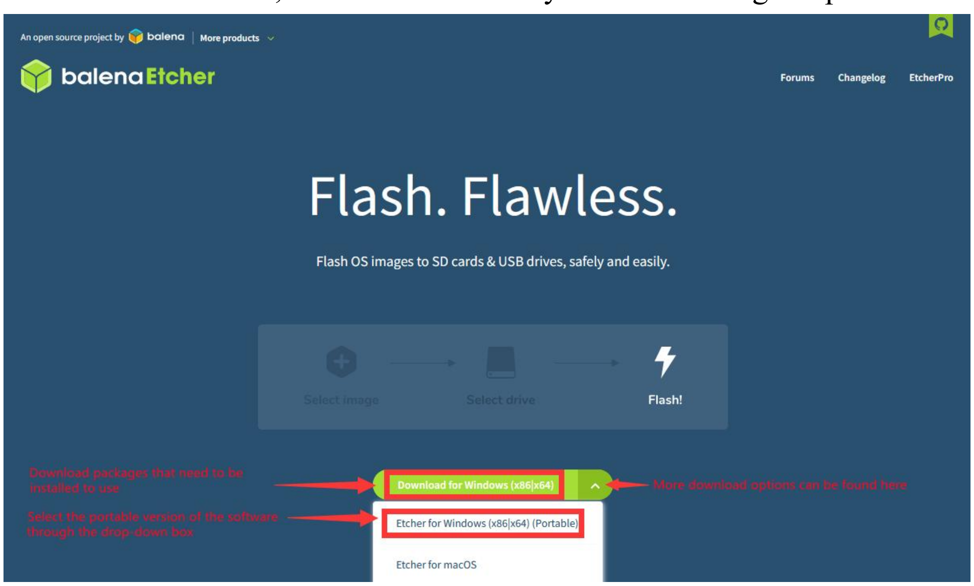
2) Then use the card reader to insert the TF card into the computer

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 more than 2GB

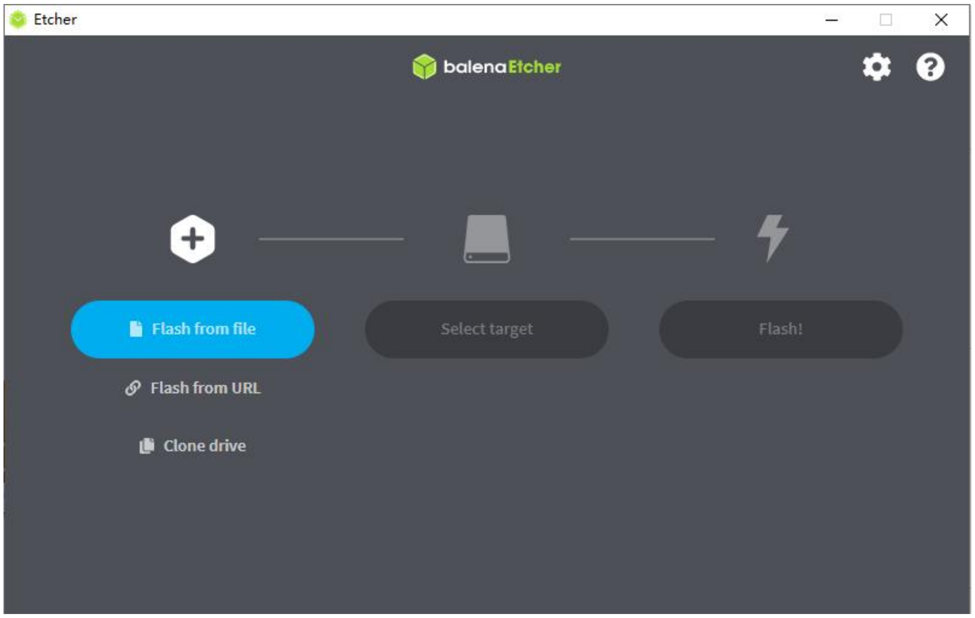
4) Then download the burning software of Linux image——**balenaEtcher**, the download address is

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

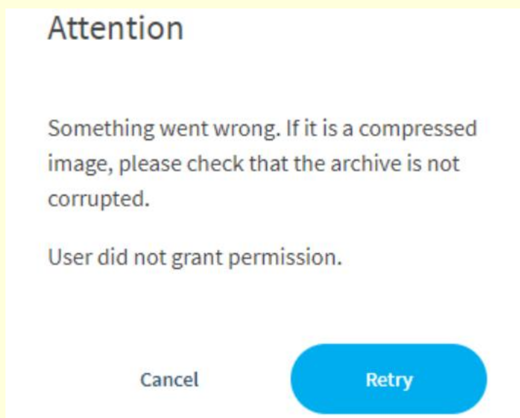
5) After entering the balenaEtcher download page, click the green download button to download the installation package of balenaEtcher. You can also select the Portable version of the balenaEtcher software through the drop-down box. The Portable version does not need to be installed, and it can be used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



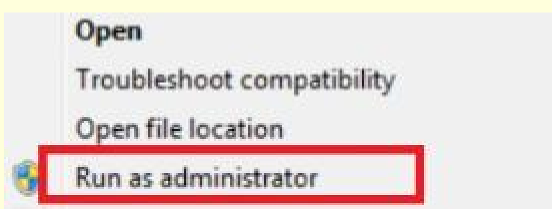

6) If the downloaded version of balenaEtcher needs to be installed, please install it before using it. If you downloaded the Portable version of balenaEtcher, just double-click to open it. The opened balenaEtcher interface is shown in the figure below




**When opening balenaEtcher, if the following error is prompted:**

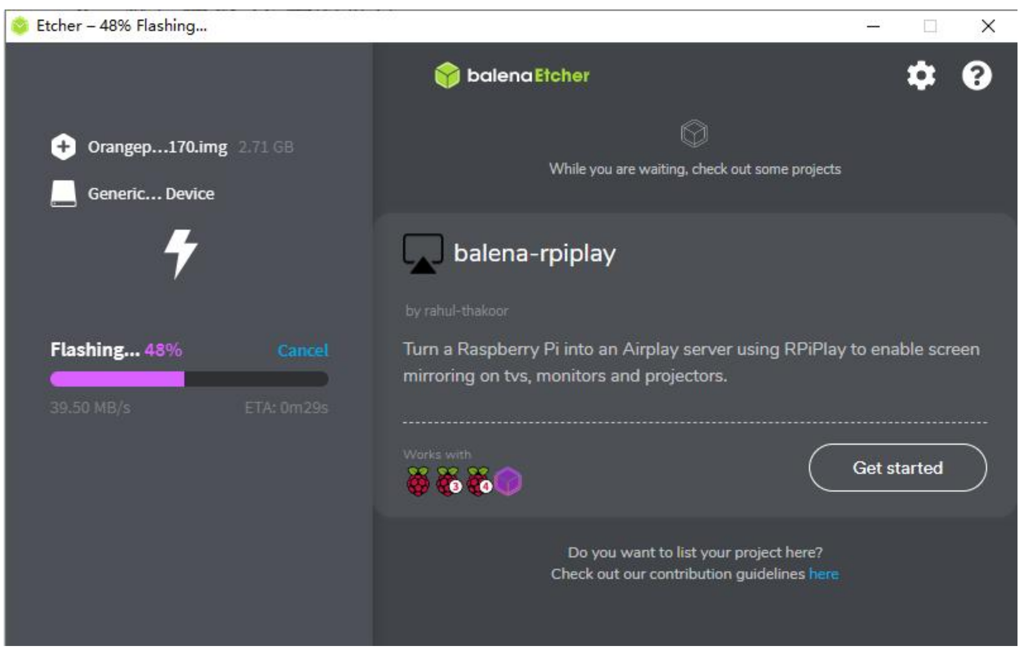



**Please select balenaEtcher, right-click, and select Run as administrator.**

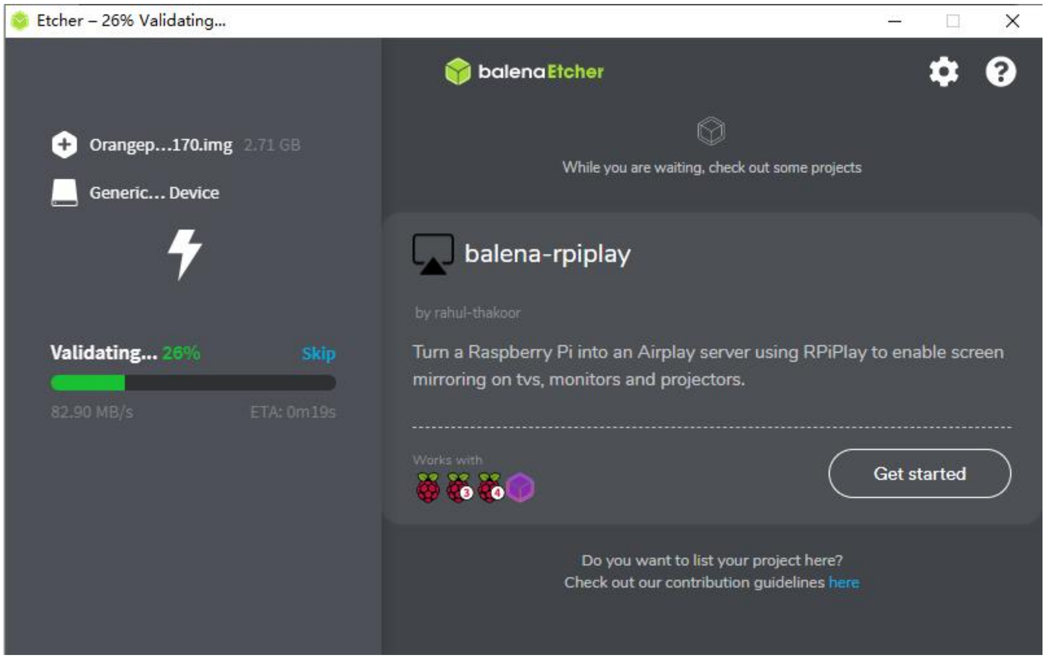



7) The specific steps to use balenaEtcher to burn the Linux image are as follows
   a. First select the path of the Linux image file to be burned
   b. Then select the drive letter of the TF card
   c. Finally, click Flash to start burning the Linux image to the TF card

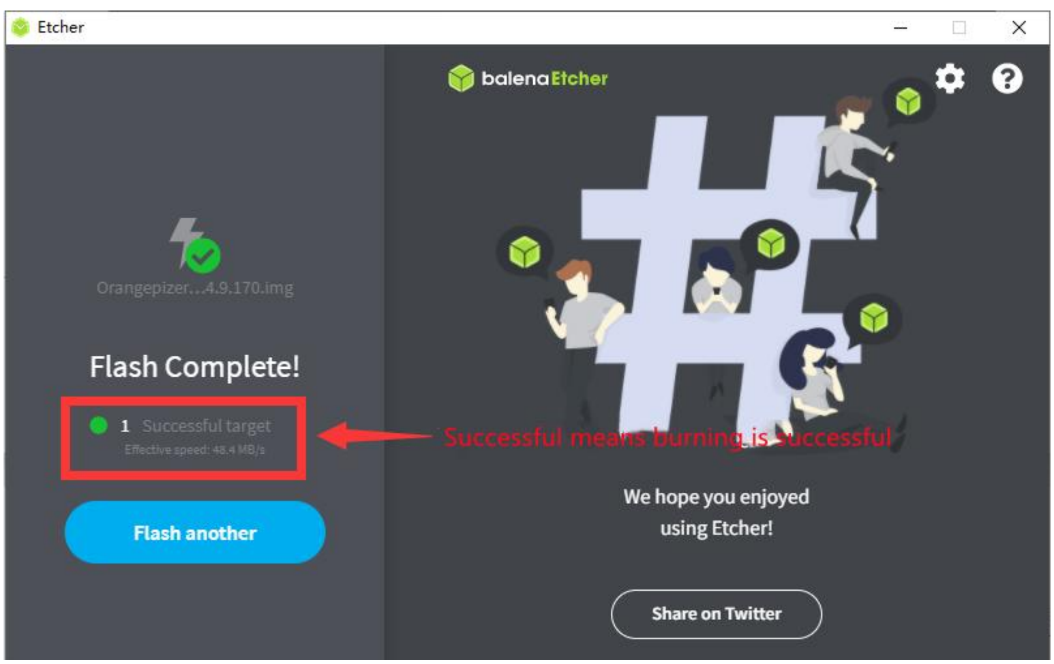

8) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card




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




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




## 2. 3. 2. How to use 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

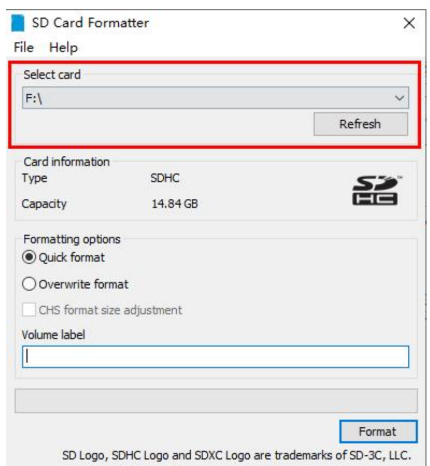
3) Then format the TF card

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

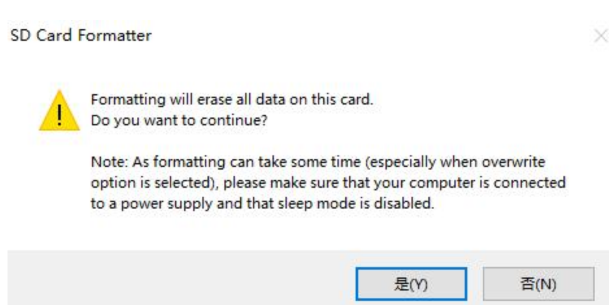
**https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip**

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

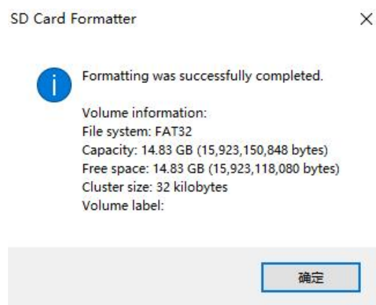
c. If only a TF card is inserted into the computer, the drive letter of the TF card will be displayed in the **"Select card"** column. If multiple USB storage devices are inserted into the computer, you can select the corresponding drive letter of the TF card through the drop-down box




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

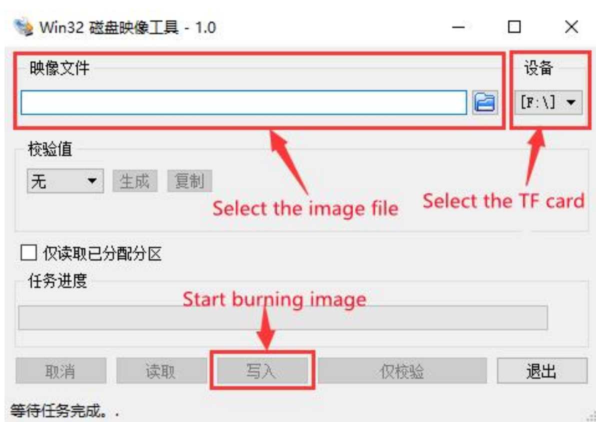
**http://sourceforge.net/projects/win32diskimager/files/Archive/**

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

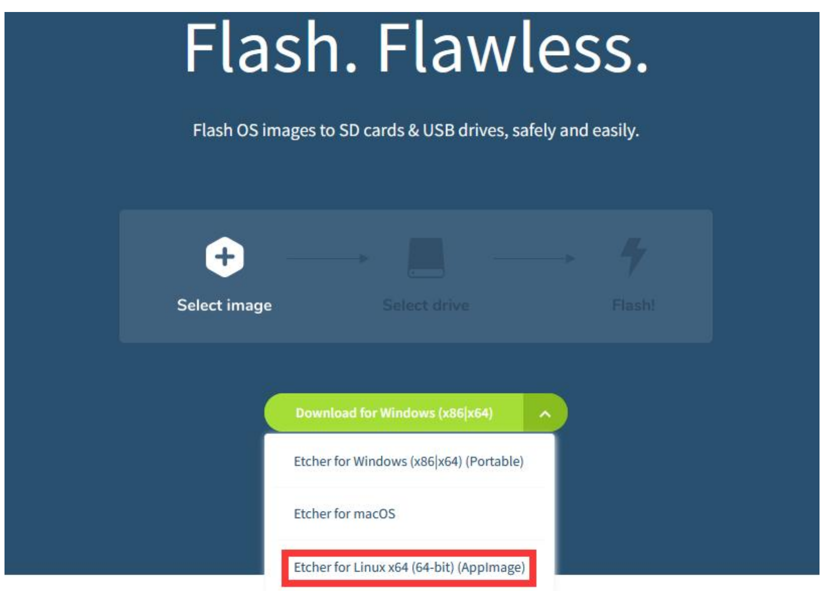
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brands

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, please select the Linux version of the software from the drop-down box to download



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

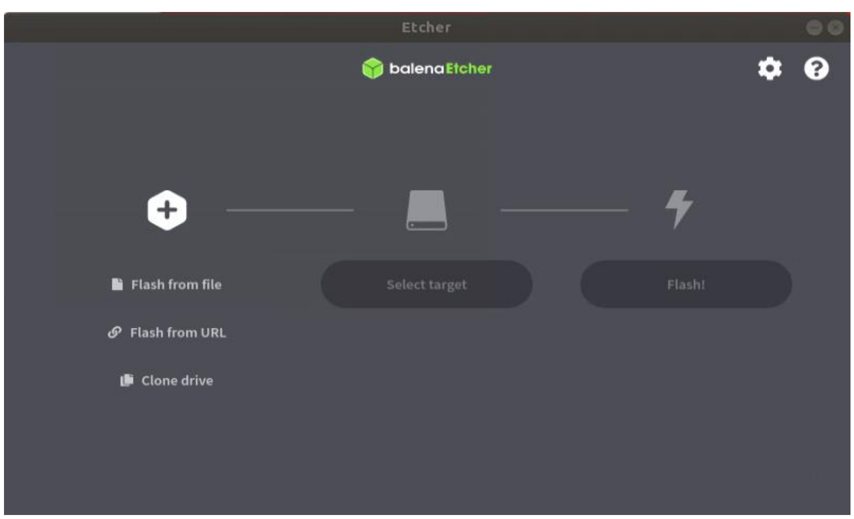
The decompression command for the compressed package ending in 7z is as follows

```
test@test:~$ 7z x Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.7z
test@test:~$ ls Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.*
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.7z
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.sha    # checksum file
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.img    #image file
```

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

```
test@test:~$ sha256sum -c *.sha
Orangepi5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.img: OK
```

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

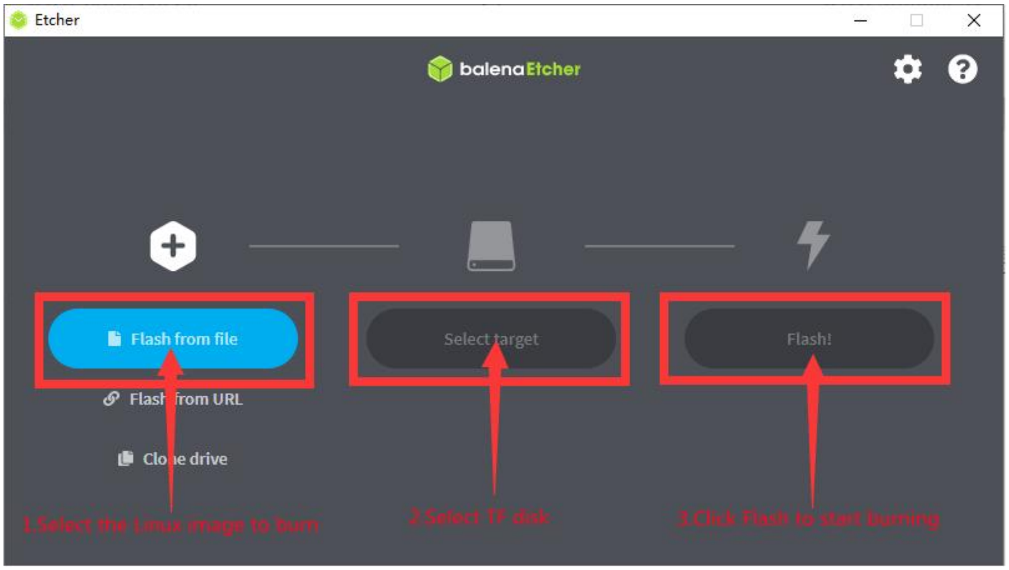



8) The specific steps to use balenaEtcher to burn the Linux image are as follows

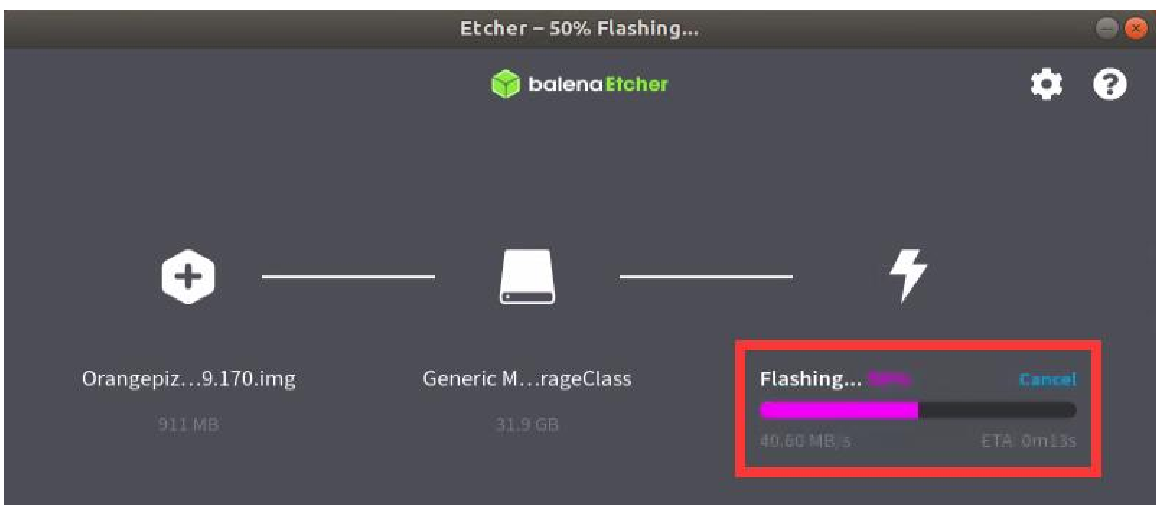
a. First select the path of the Linux image file to be burned

b. Then select the drive letter of the TF card

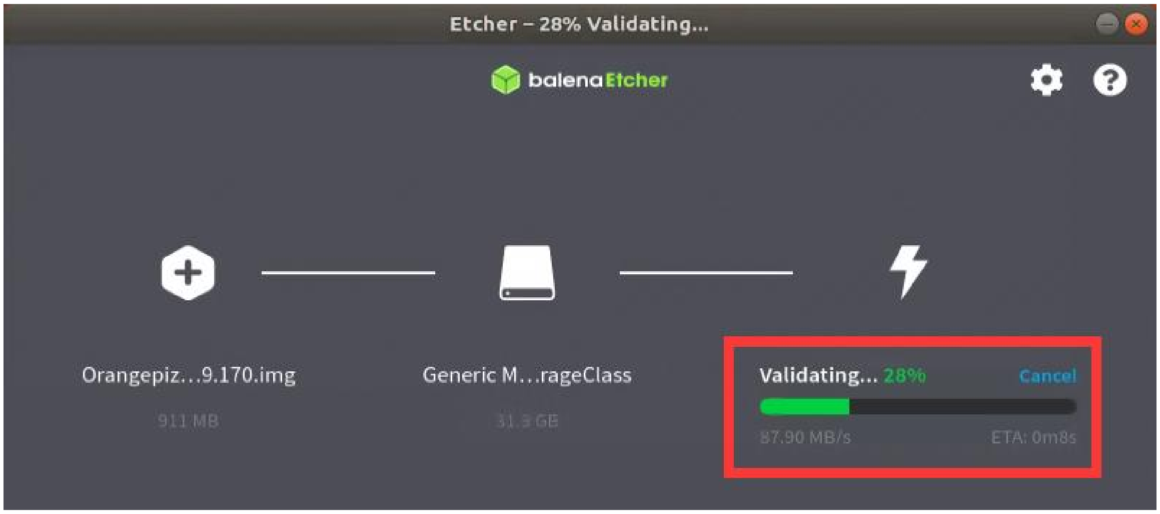
c. Finally, click Flash to start burning the Linux image to the TF card



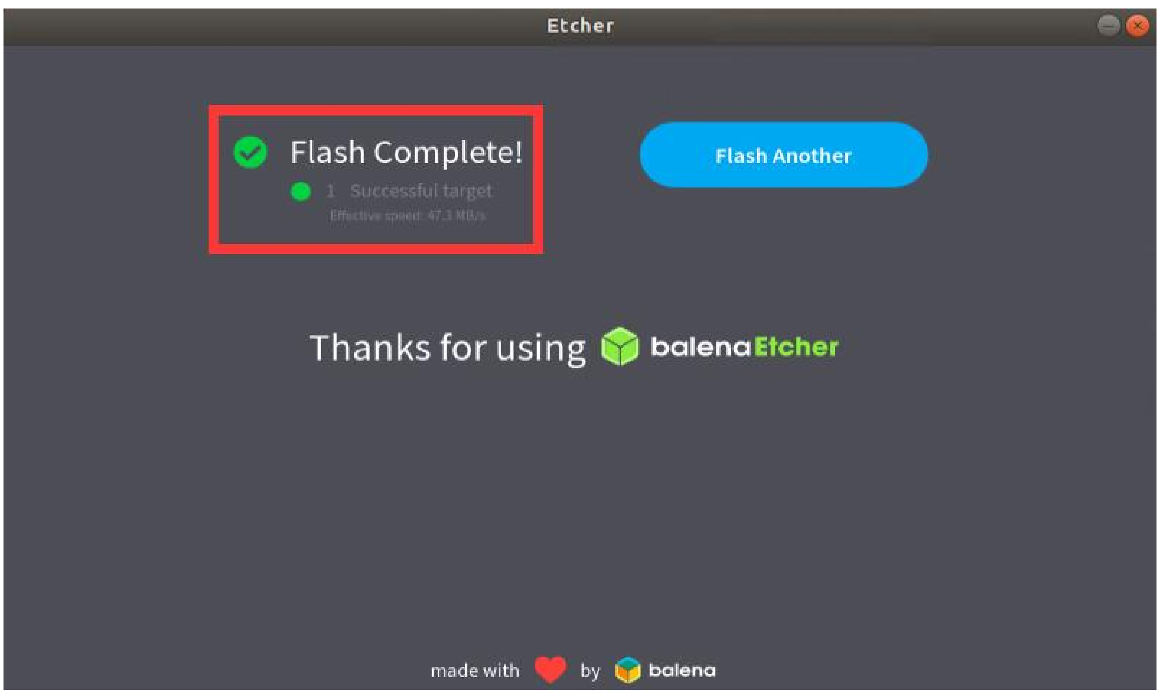

9) The interface displayed during the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card




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




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




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

**Note that the Linux image mentioned here specifically refers to the image of**

**Linux distributions such as Debian or Ubuntu downloaded from the Orange Pi data download page.**

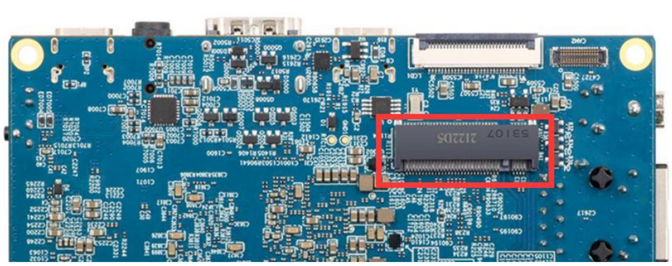
### 2. 5. 1. The method of using the dd command to burn

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.

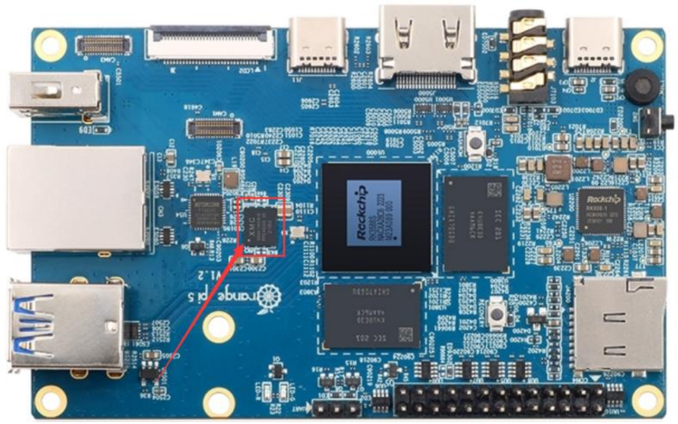
a. The M.2 2230 SSD is as follows


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



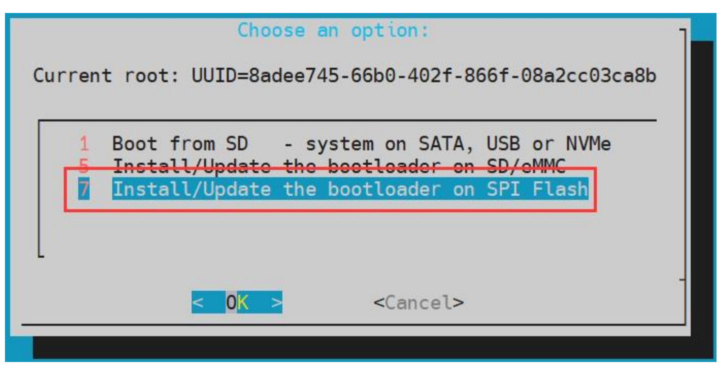
4) Burning the linux image to SPIFlash+NVMe SDD needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please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

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

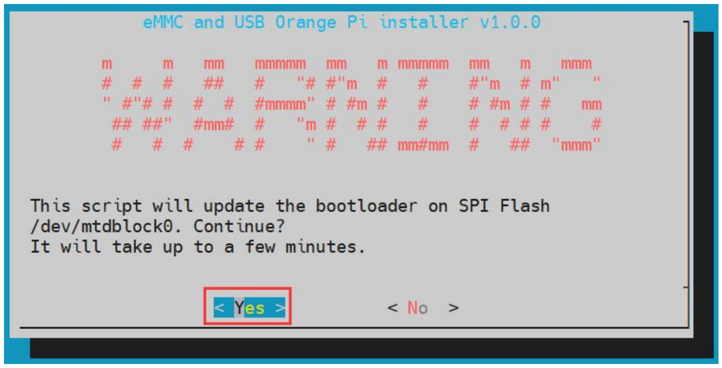
a. Run **nand-sata-install** first, **ordinary users remember to add sudo permission**

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

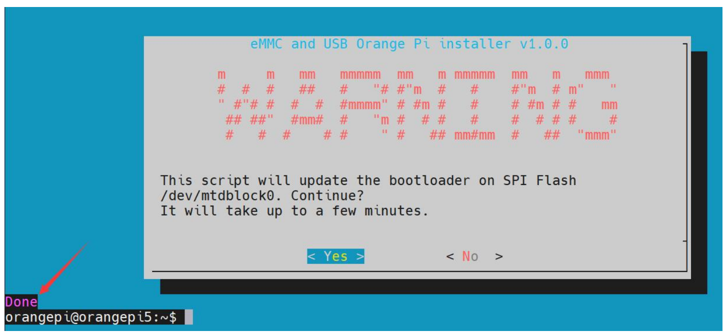
b. Then select **7 Install/Update ther bootloader on SPI Flash**




c. Then select **<Yes>**




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




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
Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.110.img
```

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

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

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

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

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

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

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

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

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

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

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

```
sudo dd bs=1M if=Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.110.img of=/dev/nvme0n1 status=progress

sudo sync
```

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

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

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

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

a. 128GB NVMe SDD

```
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 SDD

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

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

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

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

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

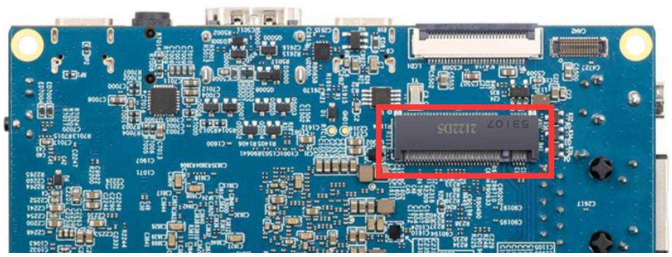
### 2. 5. 2. Using balenaEtcher software to burn

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.

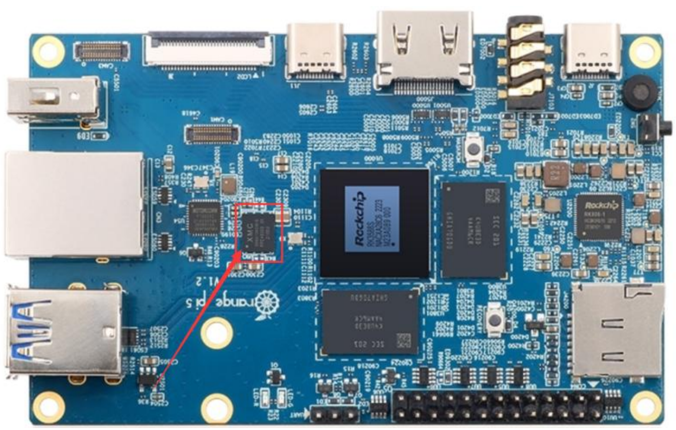
a. The M.2 2230 SSD is as follows


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

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

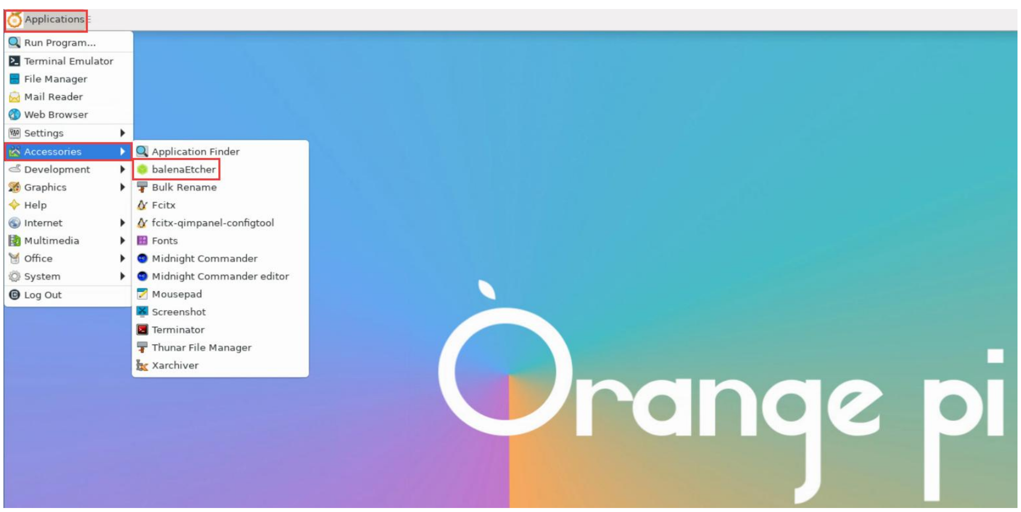
5) After booting into the linux system in the TF card, please confirm that the NVMe SSD has been properly recognized by the linux of the development board. If the NVMe SSD is recognized normally, use the **sudo fdisk -l** command to see **nvme**-related information

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

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

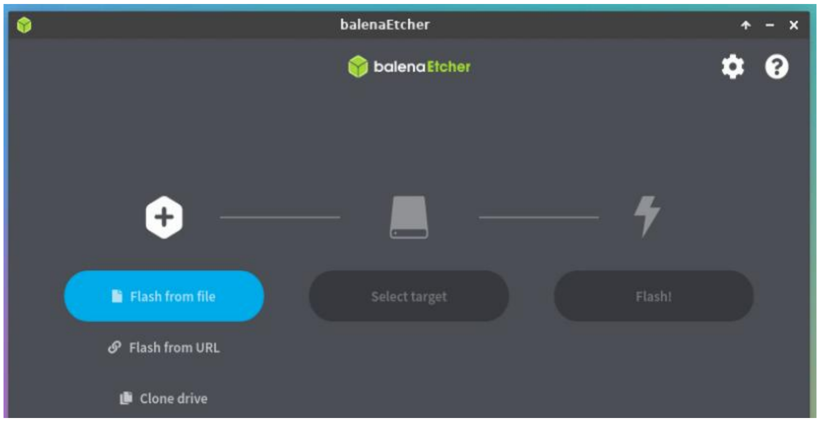
```
orangepi@orangepi:~/Desktop$ lspci
0004:40:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd Device 3588 (rev 01)
0004:41:00.0 Non-Volatile memory controller: MAXIO Technology (Hangzhou) Ltd. NVMe SSD Controller MAP1202 (rev 01)
```

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



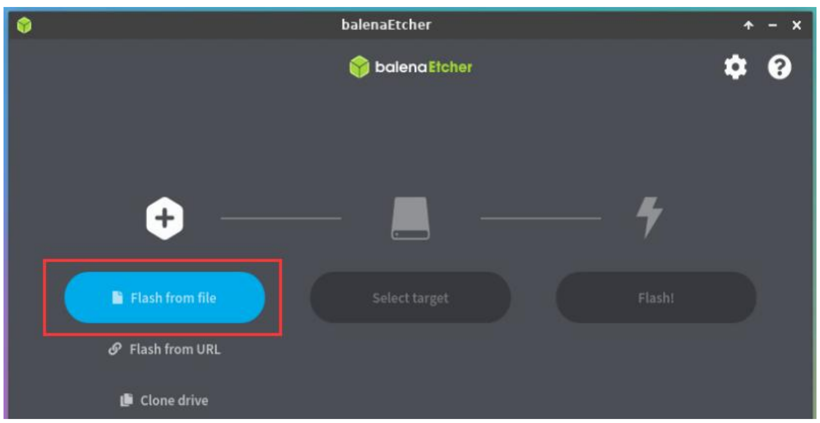

**If it is not pre-installed, for how to download and install the arm64 version of balenaEtcher, please refer to the instructions in the section on how to download and install the arm64 version of balenaEtcher.**

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

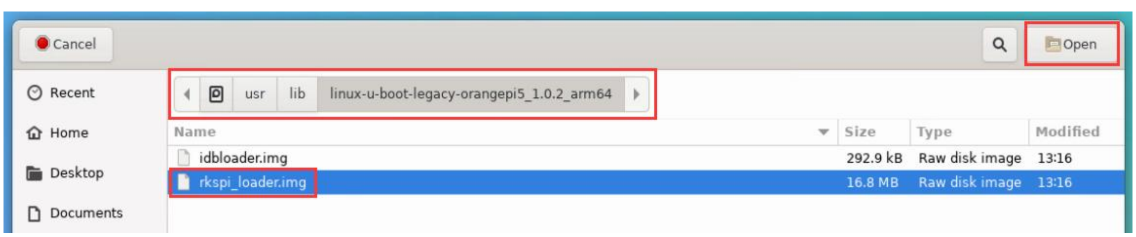



8) The method of using balenaEtcher to burn u-boot to the SPI Flash of the development board is as follows:

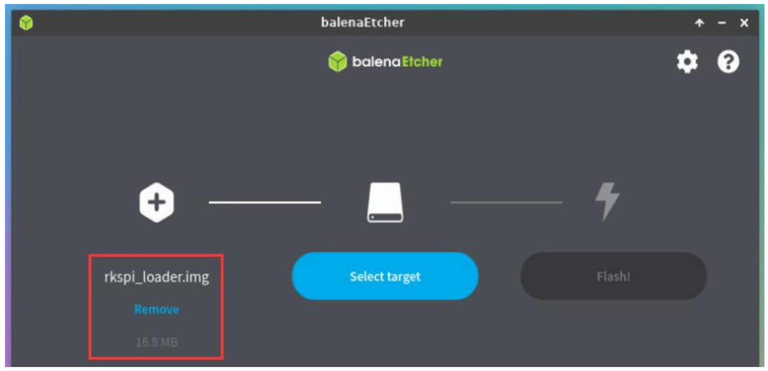
a. First click **Flash from file**



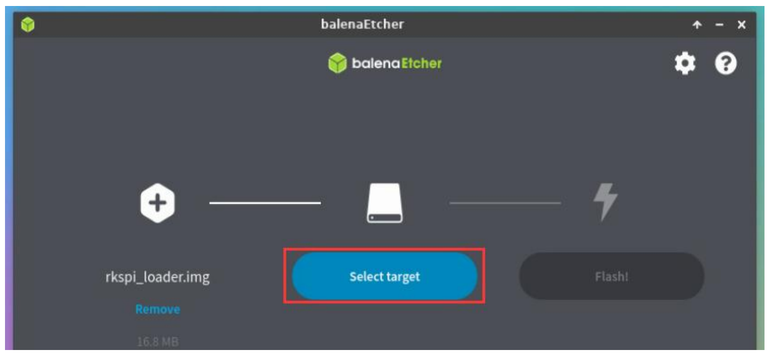

b. Then enter the **/usr/lib/linux-u-boot-legacy-orangepi5_1.x.x_arm64** directory, select **rkspi_loader.img**, and click **Open** to open



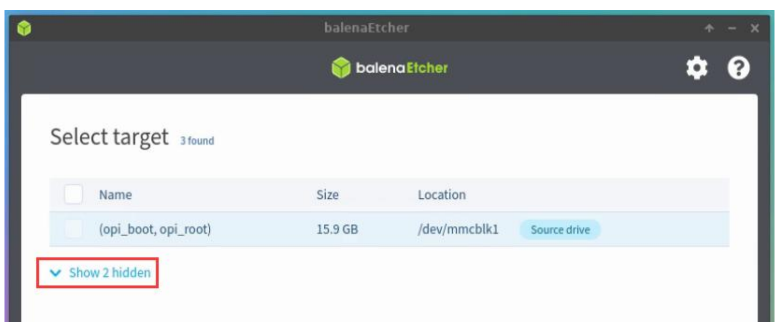

c. The interface after opening **rkspi_loader.img** is as follows:



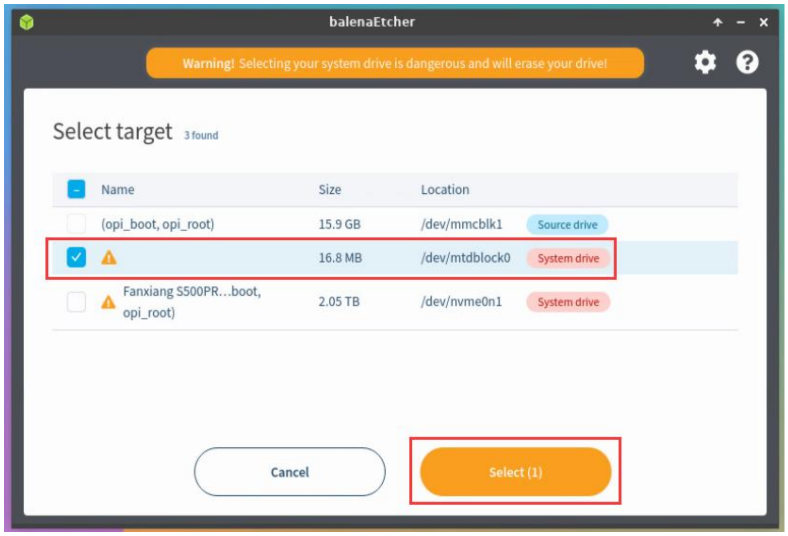

d. Then click **Select target**



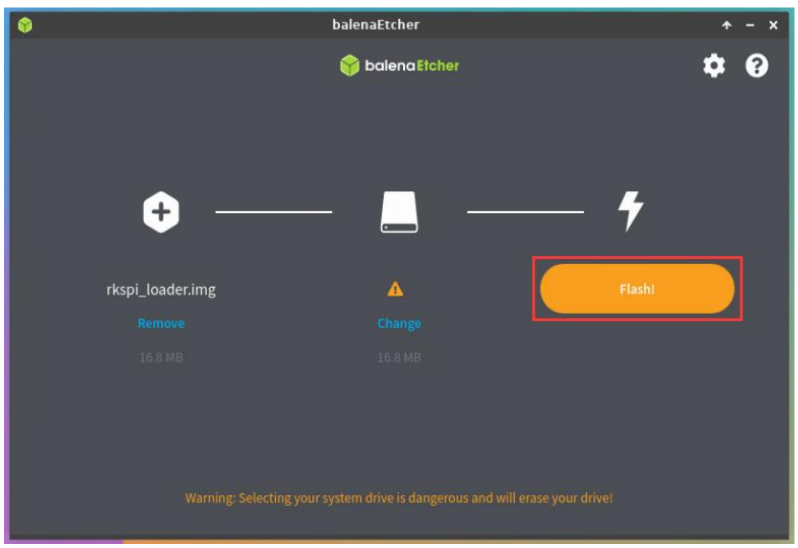

e. Then click **Show 2 hidden** to open more options for storage devices



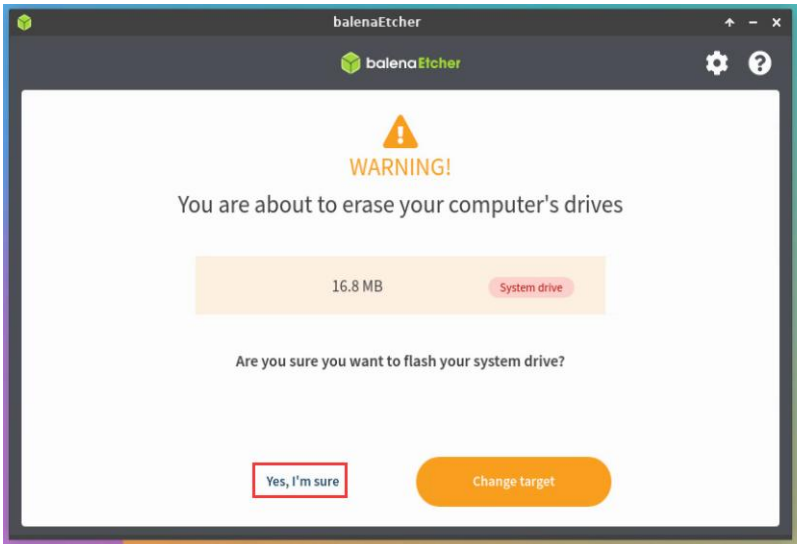

f. Then select the device name of SPI Flash **/dev/mtdblock0**, and click **Select**



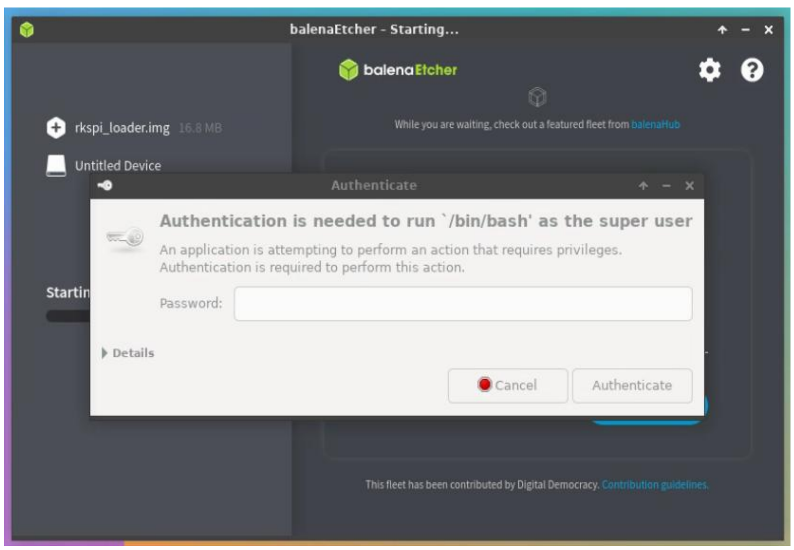

g. Then click **Flash**



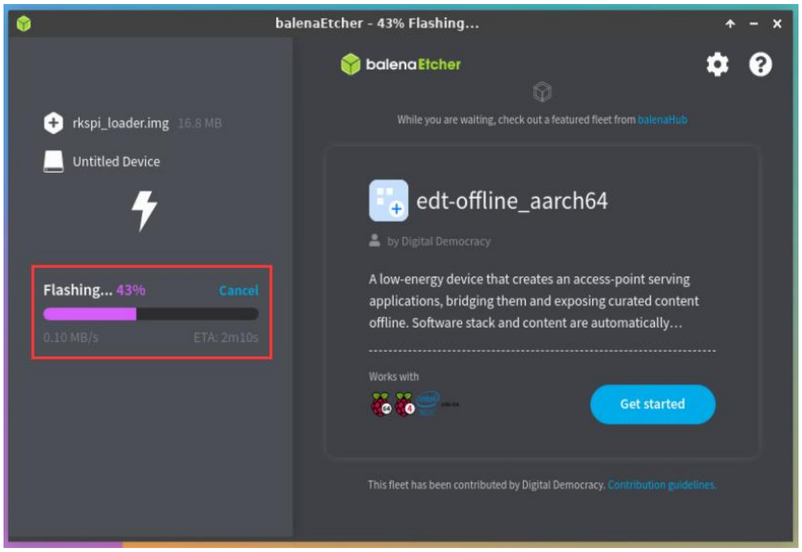

h. Then click **Yes, I'm sure**



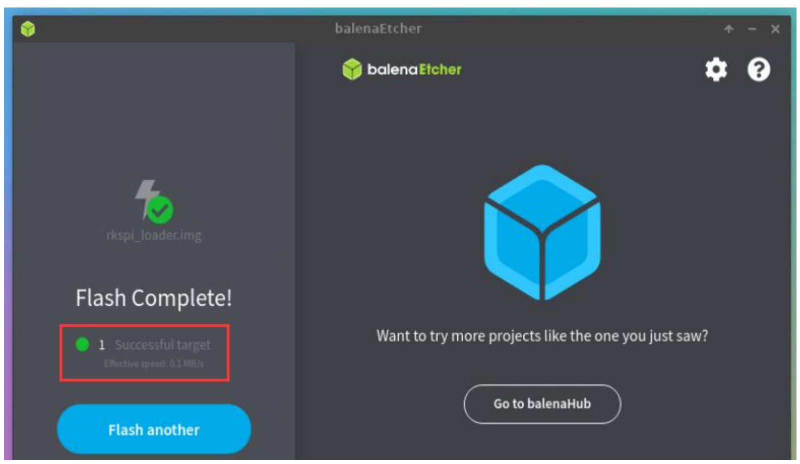

i. Then enter the password **orangepi** of the development board linux system, and it will start burning the u-boot image into the SPI Flash




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

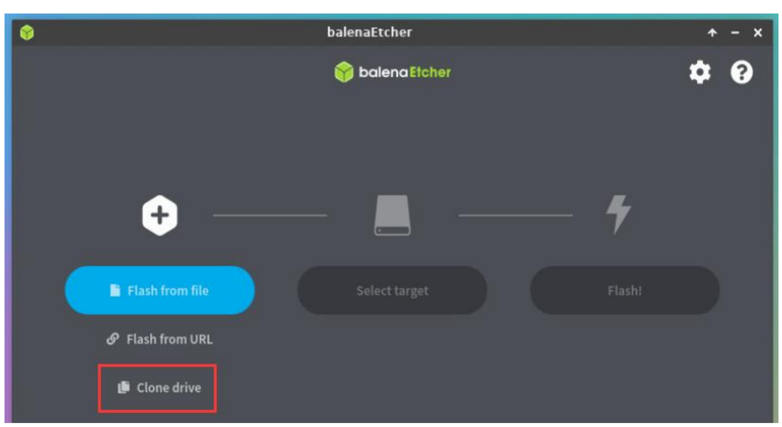



k. The display after burning is as follows:

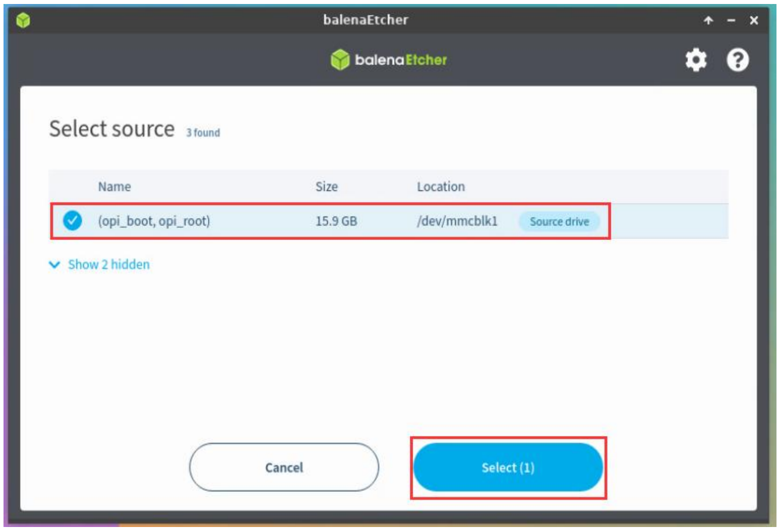



9) The method of burning the linux system in the TF card to the NVMe SSD (this method is equivalent to cloning the system in the TF card to the NVMe SSD)

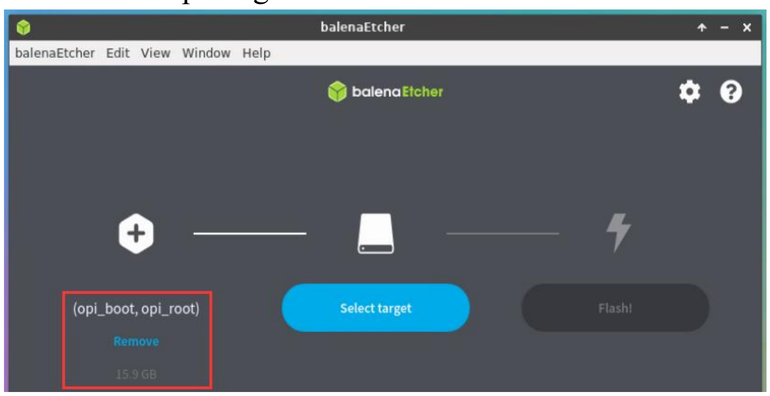
a. First click **Clone drive**



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



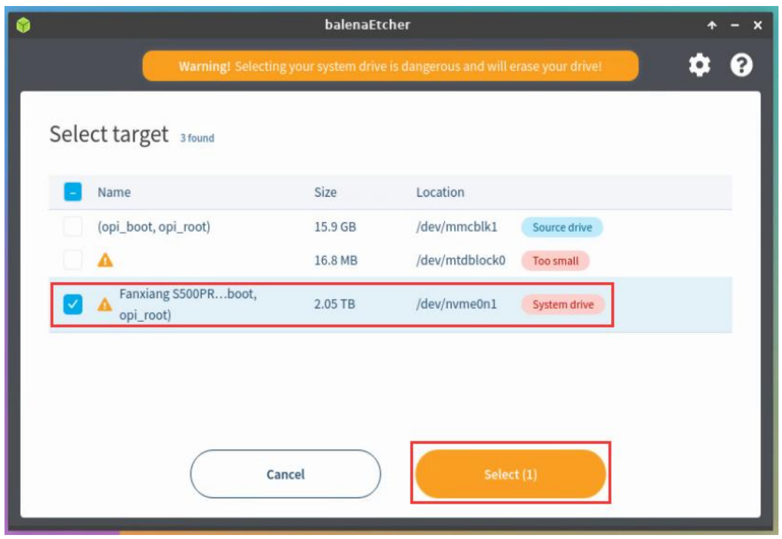

c. The interface after opening the TF card is as follows:



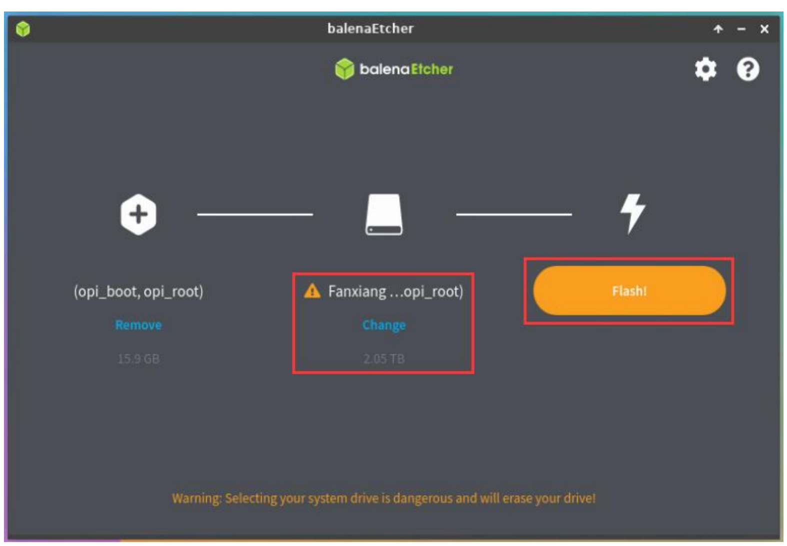

d. Then click **Select target**


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

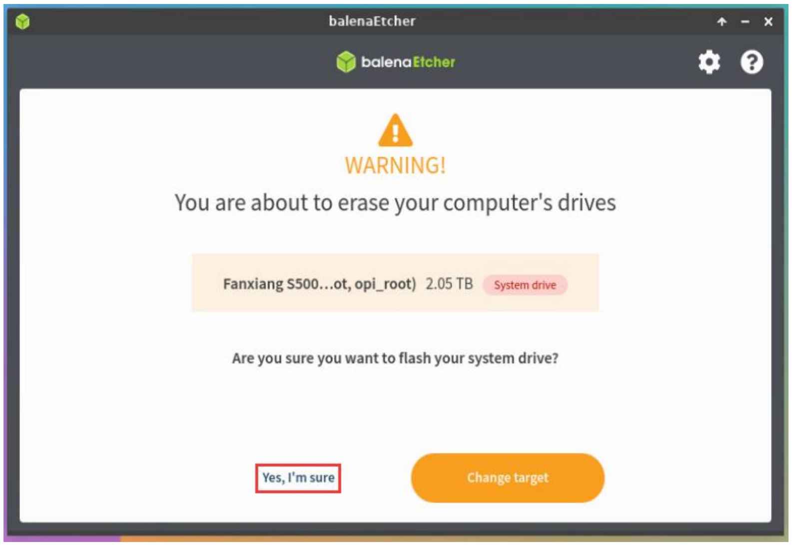

f. Then select the device name of the NVMe SSD **/dev/nvme0n1**, and click **Select**



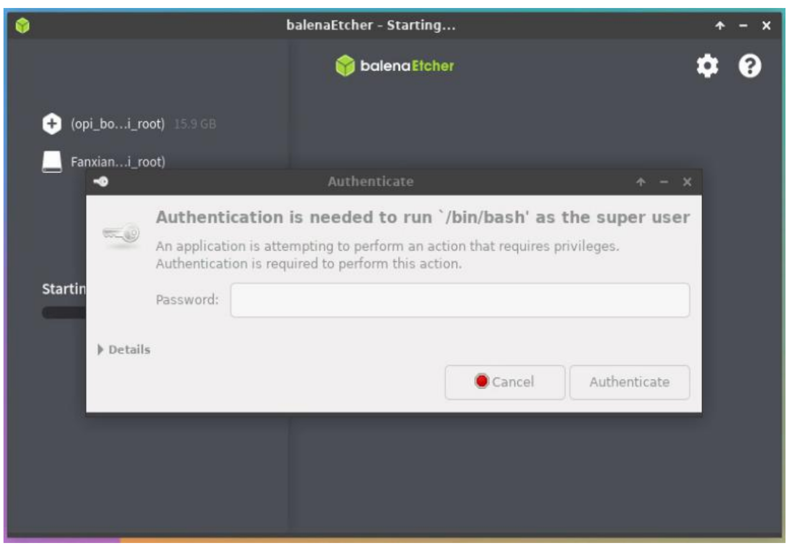

g. Then click **Flash**



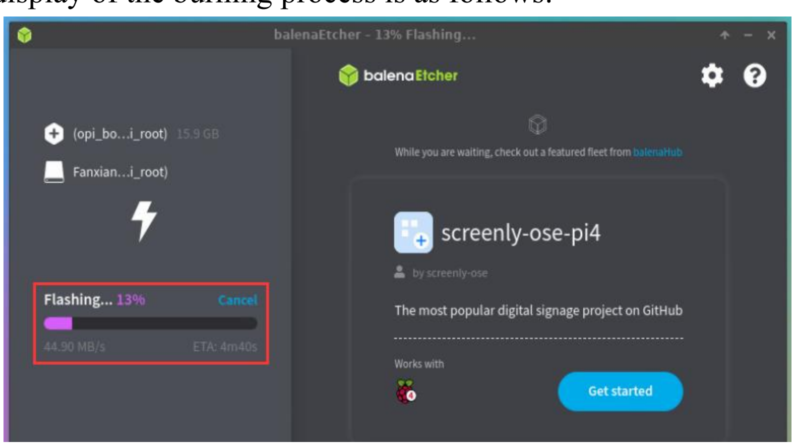

h. Then click **Yes, I'm sure**

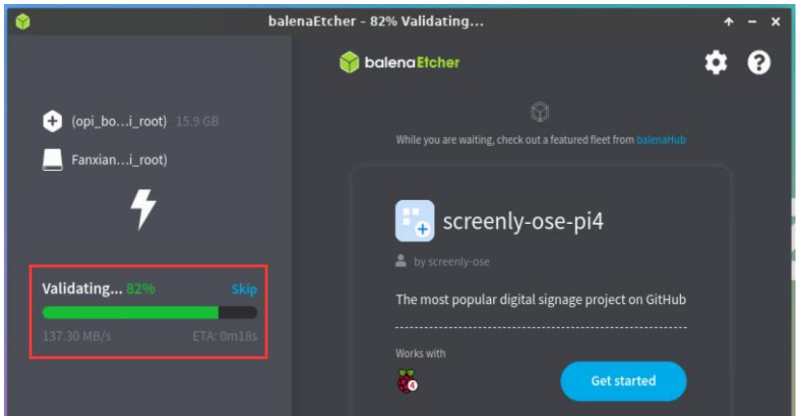



i. Then enter the password orangepi of the linux system on the development board, and it will start burning the linux image to the SSD

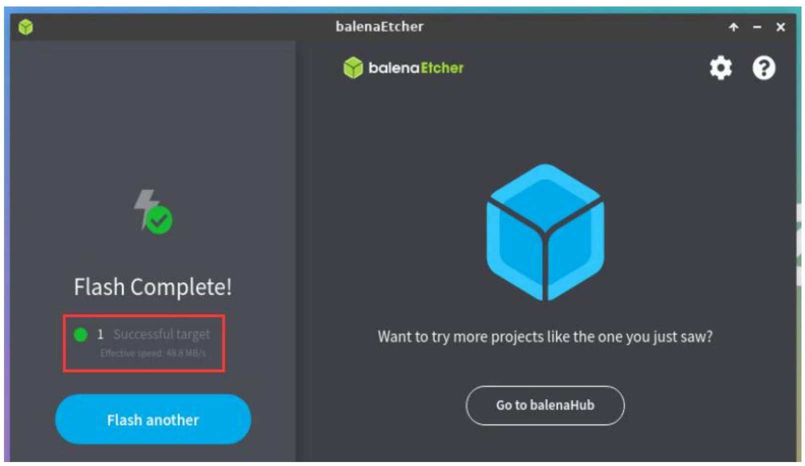



j. The display of the burning process is as follows:



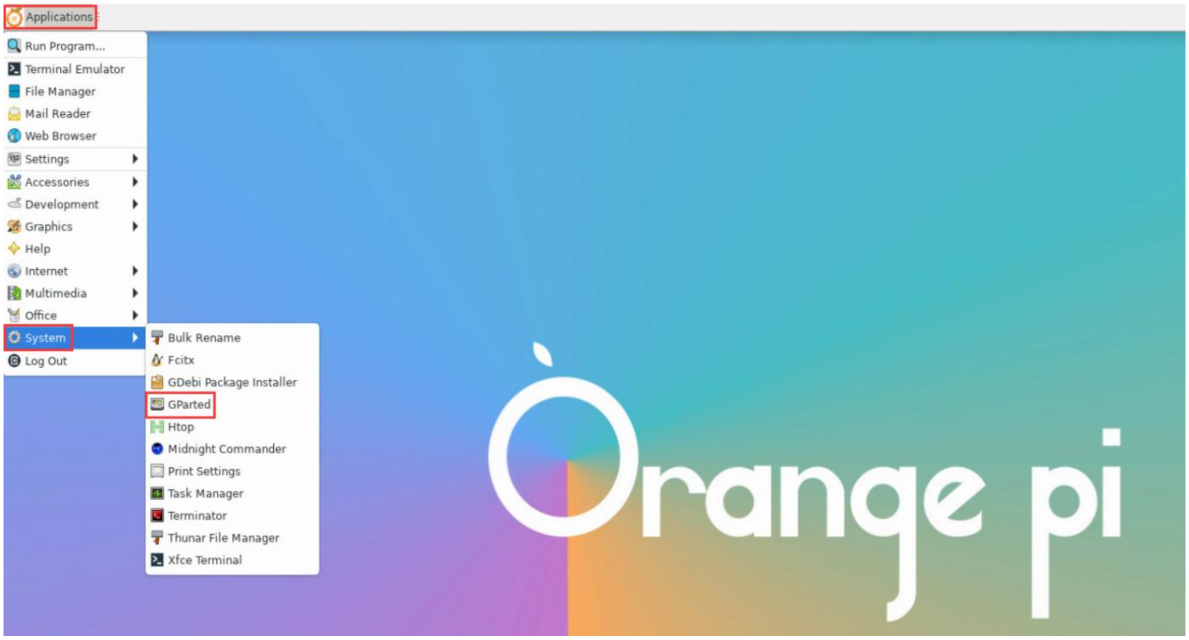



k. The display after burning is as follows:



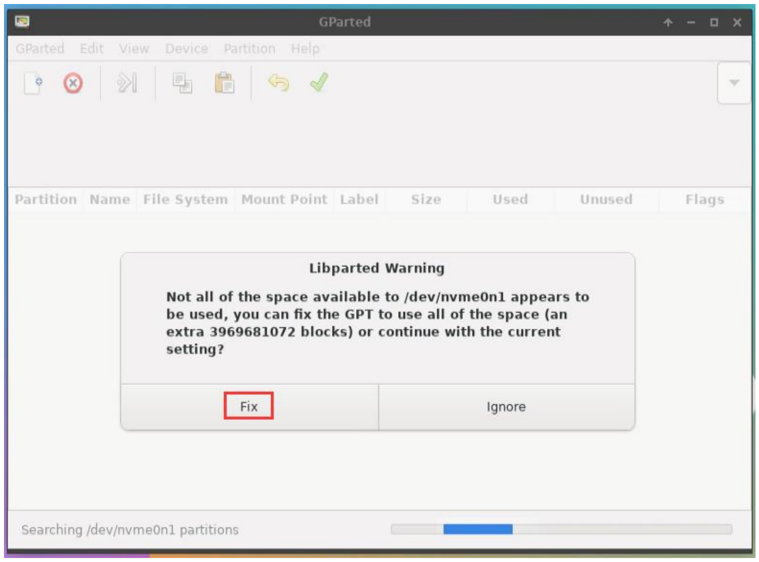

l. Then you need to expand the capacity of the rootfs partition in the NVMe SSD. The steps are as follows:

a) First open **GParted**

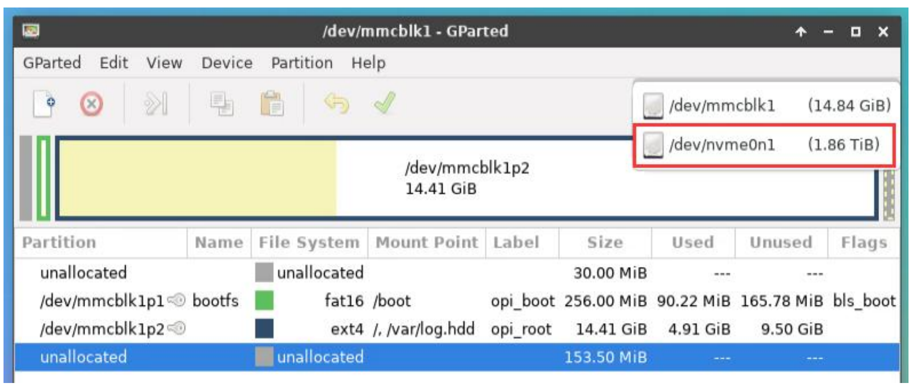



b) Then enter the password orangepi of the linux system, and click **Authenticate**

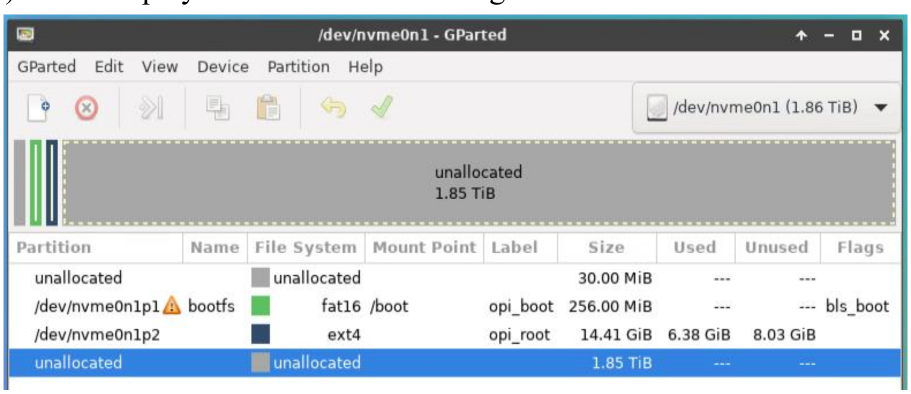

c) Then click **Fix**



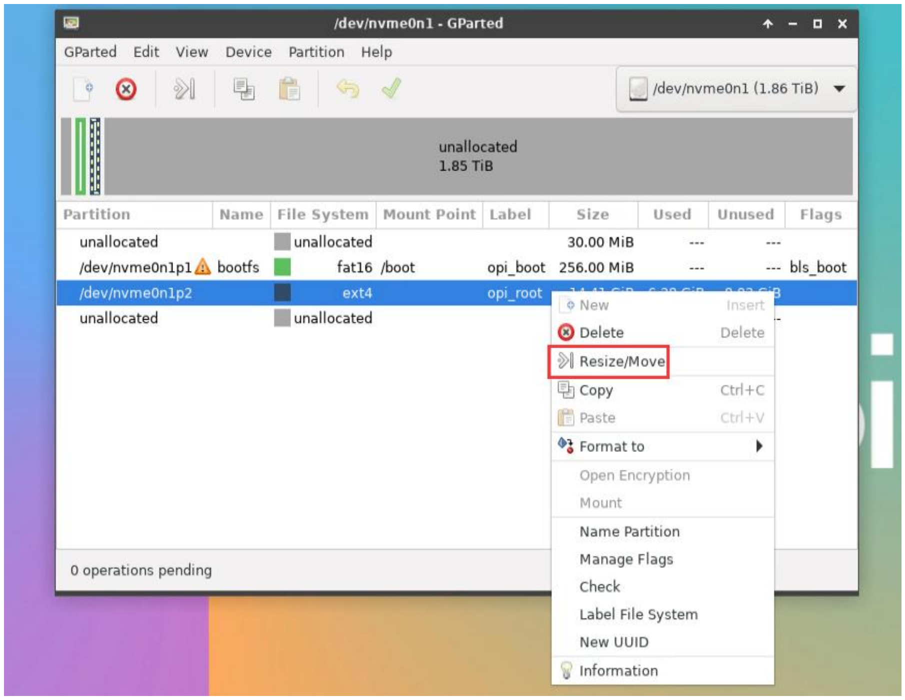

d) Then select NVMe SSD



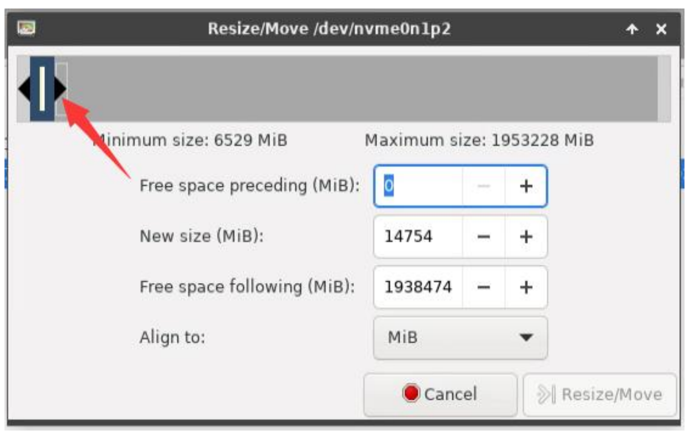

e) The display interface after selecting NVMe SSD is as follows:



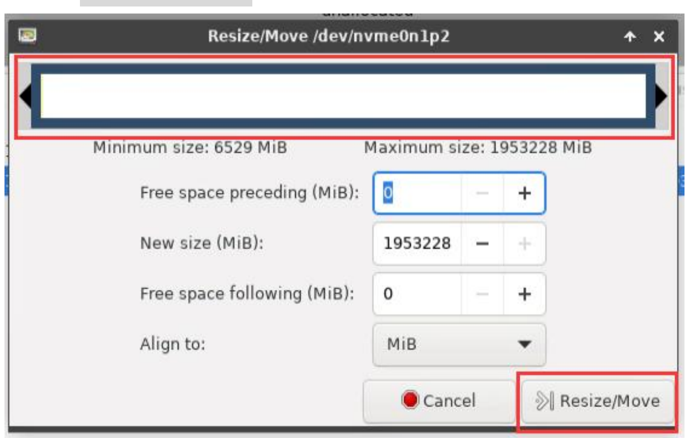

f) Then select the **/dev/nvme0n1p2** partition, click the right button again, and then select **Resize/Move**



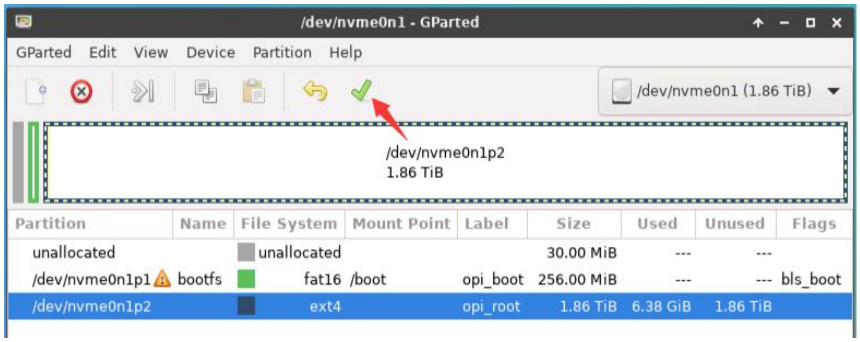

g) Then drag the capacity to the maximum at the position shown in the figure below



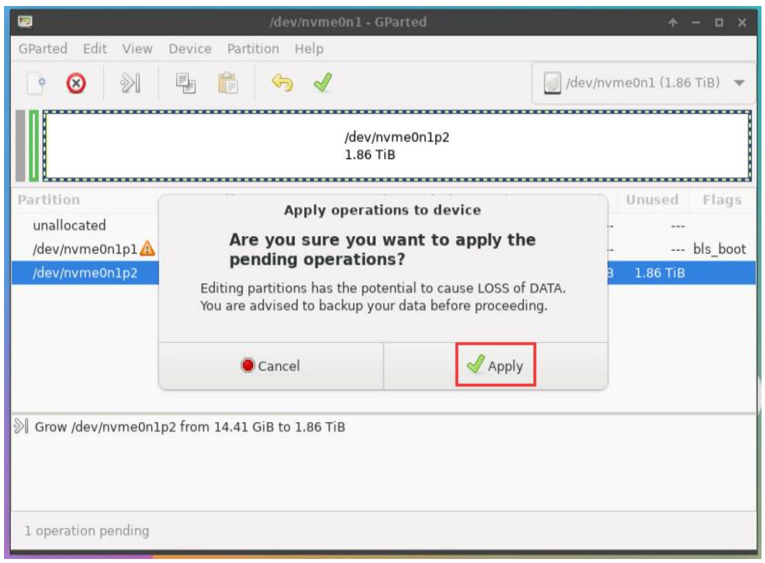

h) Then click **Resize/Move**



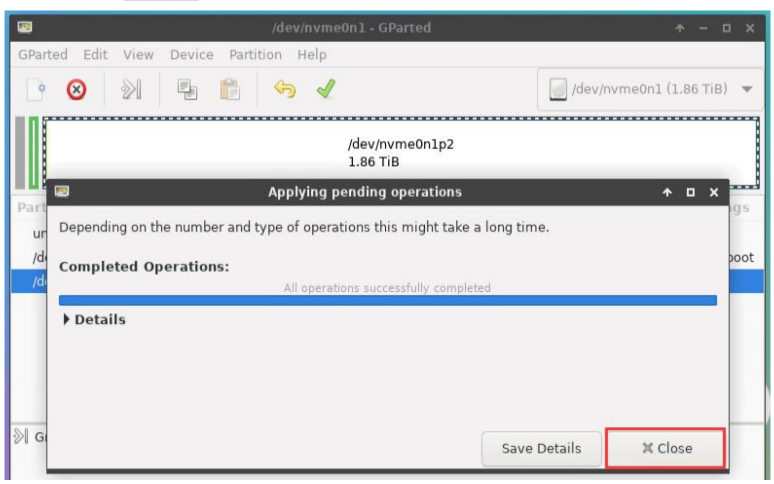
i) Then click the green √ in the picture below




j) Click **Apply** again




k) Then click **Close**

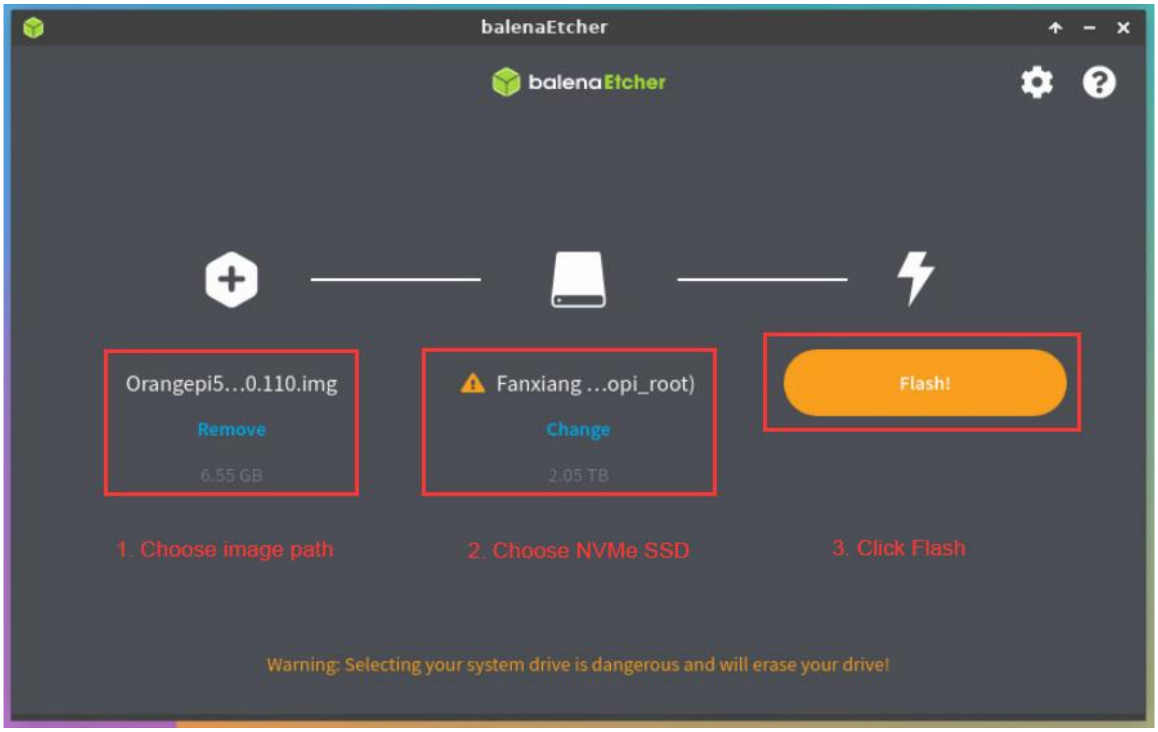



m. At this point, you can use the **sudo poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in SPIFlash+NVMe SSD will be started.

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. 6. How to write Linux image to SPI Flash+SATA SSD

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

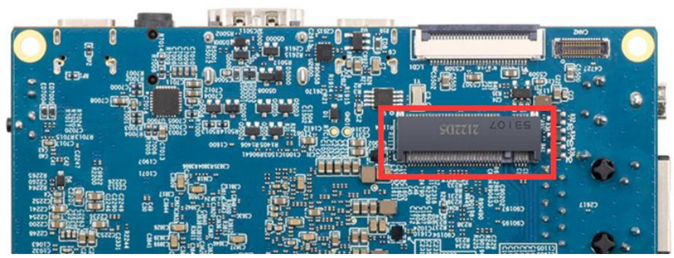
### 2. 6. 1. The method of using the dd command to burn

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

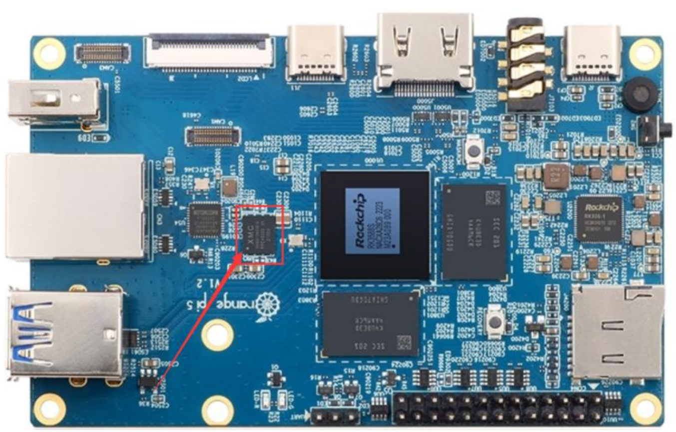
a. The M.2 2242 SSD is as follows


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


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



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



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

card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please refer to the instructions in the two sections of **the method of burning the Linux image to the TF card based on the Windows PC** and **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 dedicated to the sata ssd into the SPI Flash

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

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

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

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

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
Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.110.img
```

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

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

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

**3. If HDMI, LCD and other display screens are connected, you can open a command**

**line terminal on the desktop.**

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

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

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

sudo sync
```

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

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

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

```
sudo dd bs=1M if=Orangepi5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.110.img of=/dev/sda status=progress

sudo sync
```

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

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

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

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

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

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

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

c. Then uninstall the boot partition of the SATA SSD

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

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

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

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

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

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

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

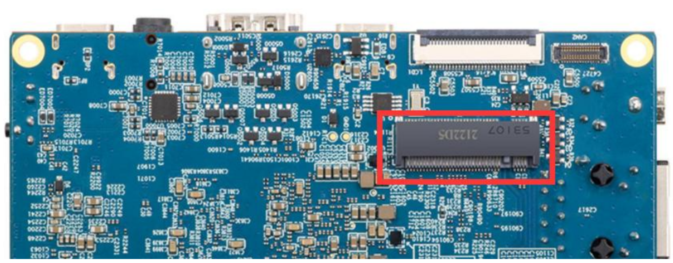
### 2.6.2. How to use balenaEtcher software to burn

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

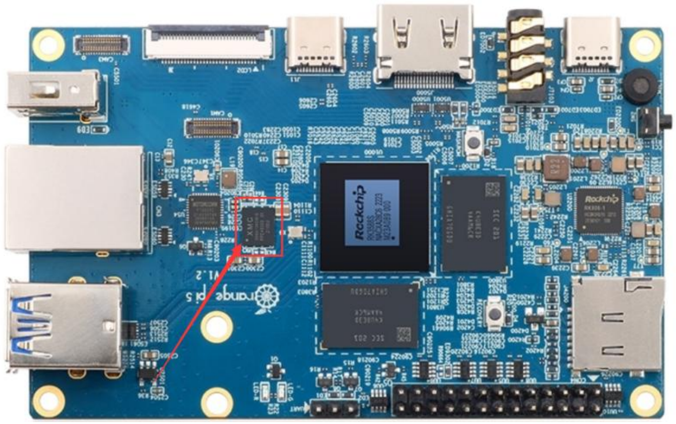
a. The M.2 2242 SSD is as follows


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


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



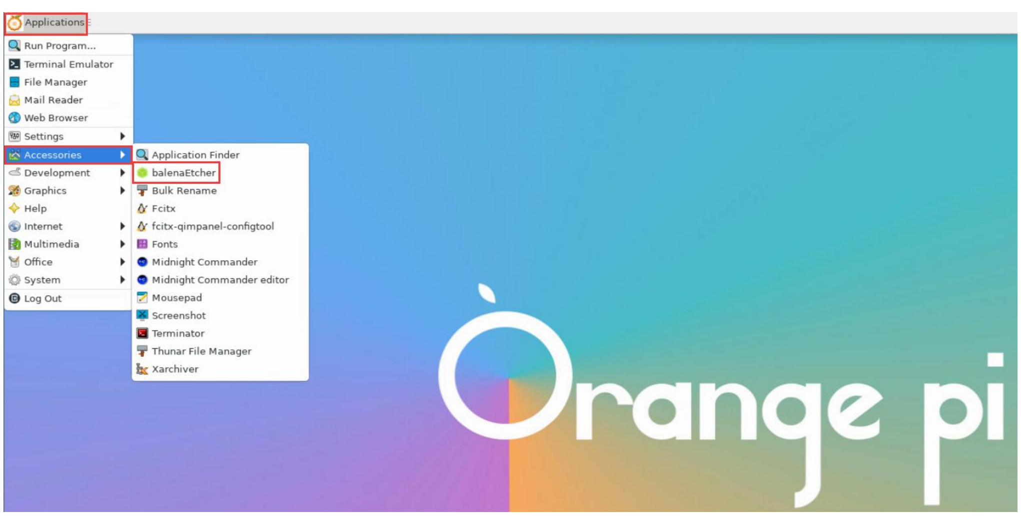
3) The position of the SPI Flash on the development board is shown in the figure below, no other settings are required before starting the programming



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

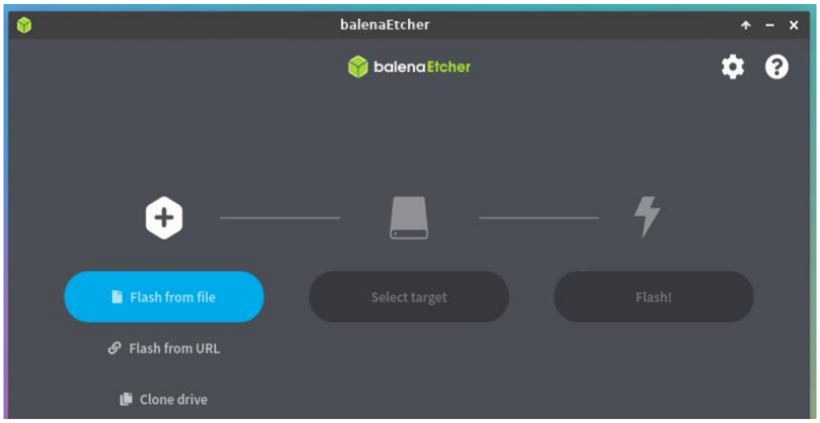
5) Then please refer to the instructions in the section of **the method of using SATA SSD** to open the sata ssd configuration to ensure that the system can recognize the ssd normally.

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



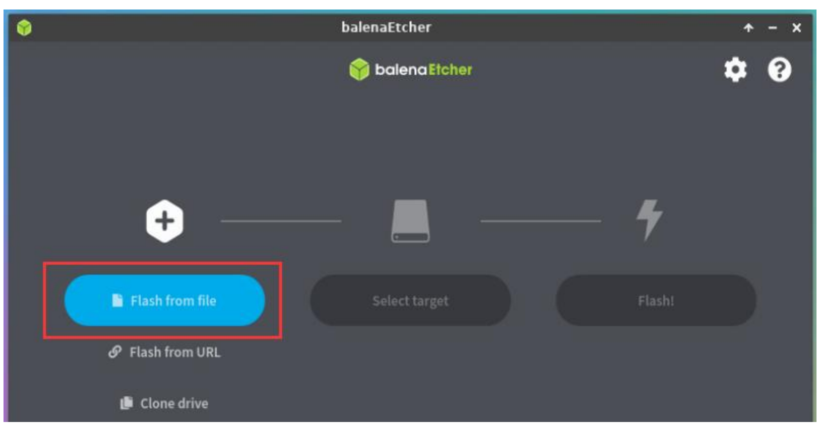
**If it is not pre-installed, for how to download and install the arm64 version of balenaEtcher, please refer to the instructions in the section on how to download and install the arm64 version of balenaEtcher.**

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

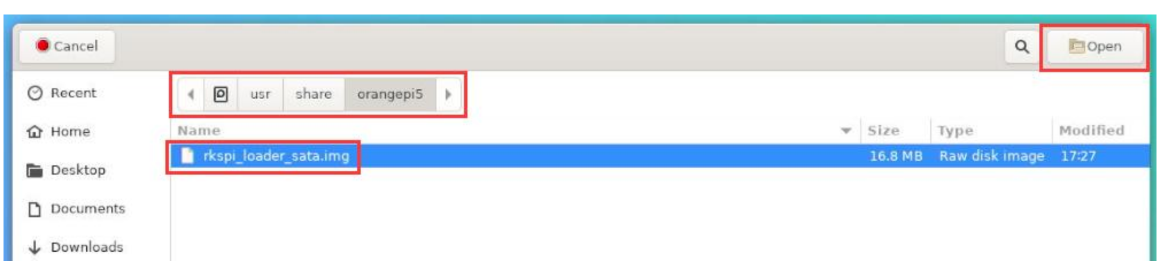



8) The method of using balenaEtcher to burn u-boot to the SPI Flash of the development board is as follows:

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



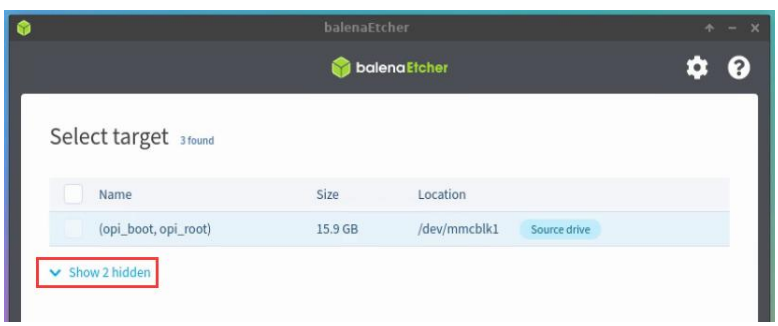

b. Then enter the **/usr/share/orangepi5/** directory, select **rkspi_loader_sata.img**, and click Open to open



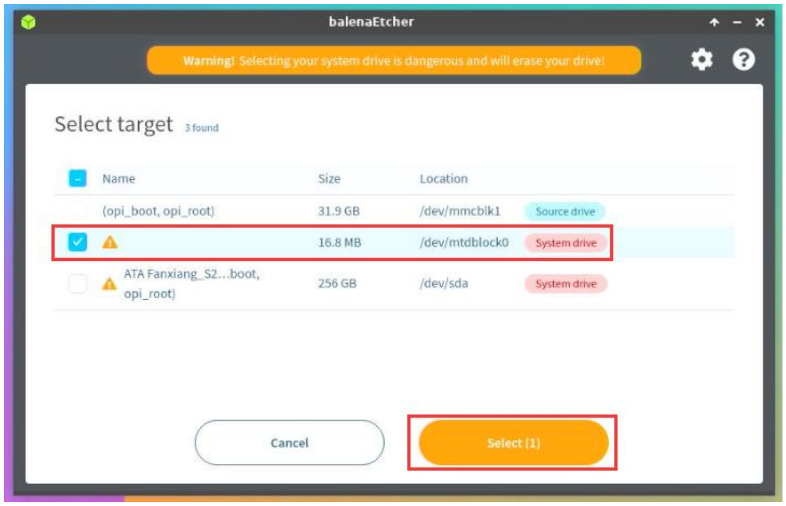
c. The interface after opening **rkspi_loader.img** is as follows:


d. Then click **Select target**

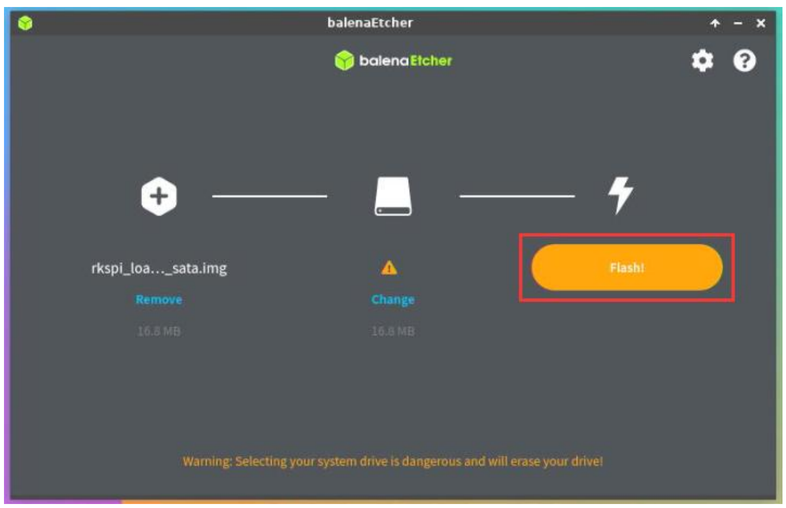

e. Then click **Show 2 hidden** to open more options for storage devices



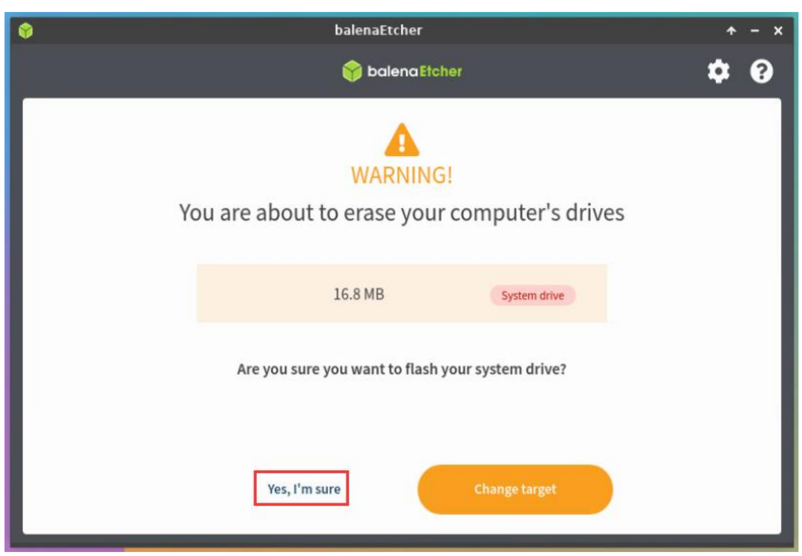

f. Then select the device name of SPI Flash **/dev/mtdblock0**, and click **Select**



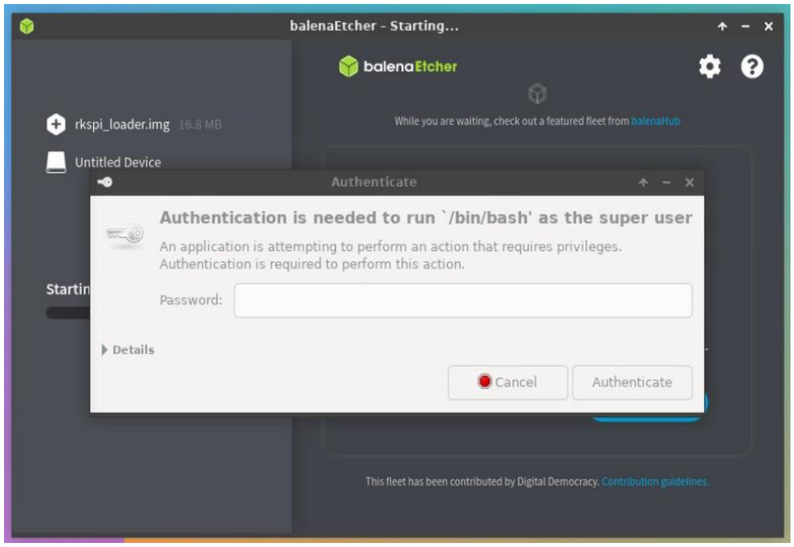

g. Then click **Flash**



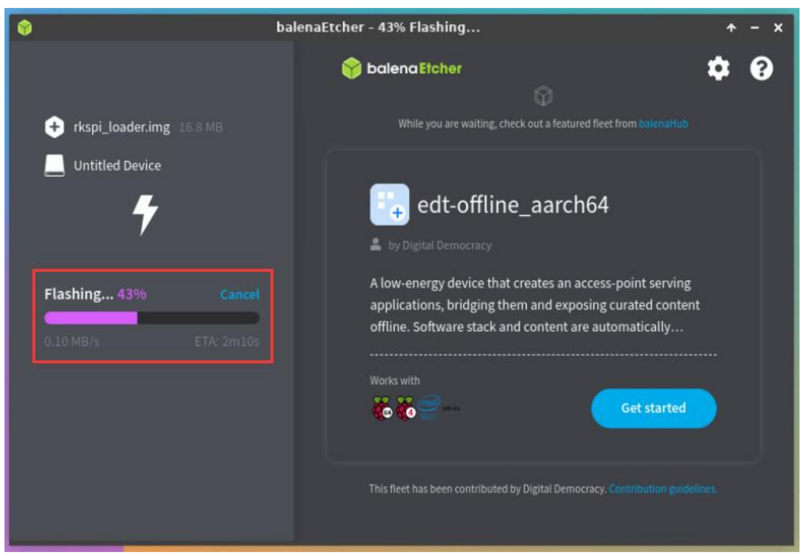

h. Then click **Yes, I'm sure**



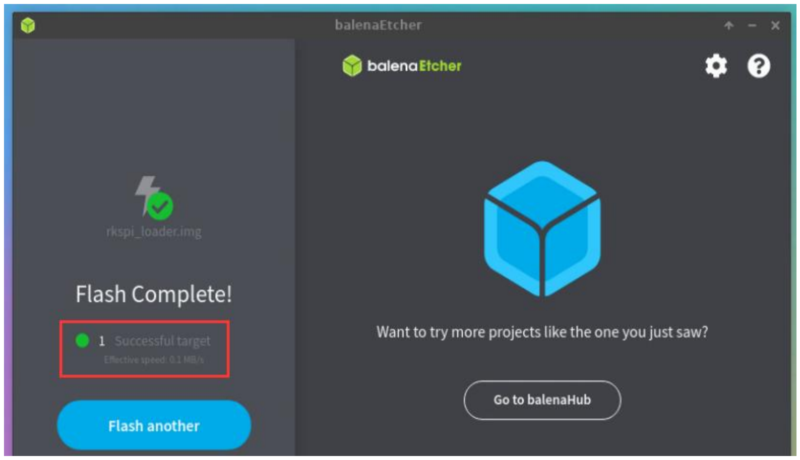
i. Then enter the password orangepi of the development board linux system, and it will start burning the u-boot image into the SPI Flash




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

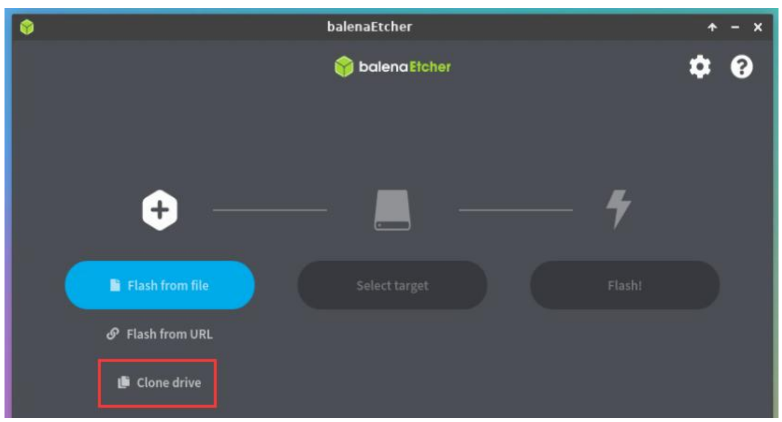



k. The display after burning is as follows:

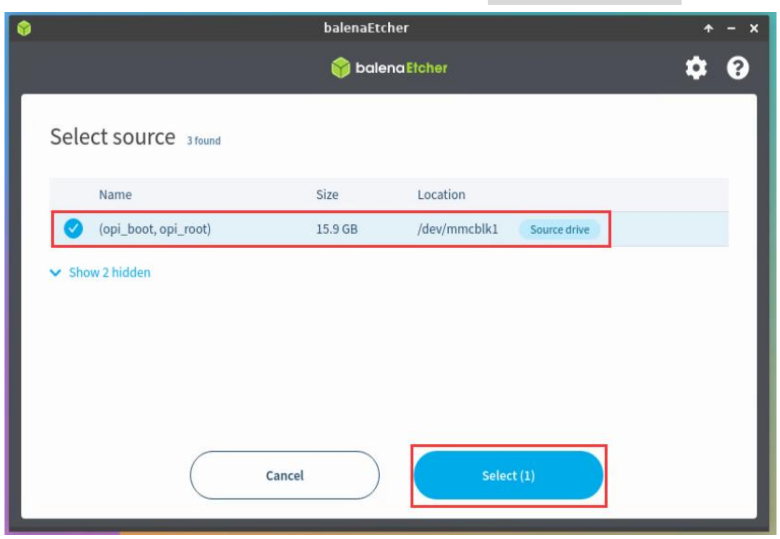



9) The method of burning the linux system in the TF card to the SSD (this method is equivalent to cloning the system in the TF card to the SSD)

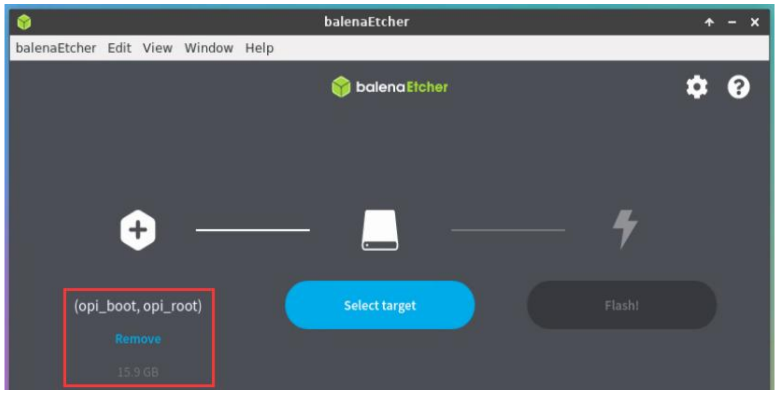
a. First click **Clone drive**



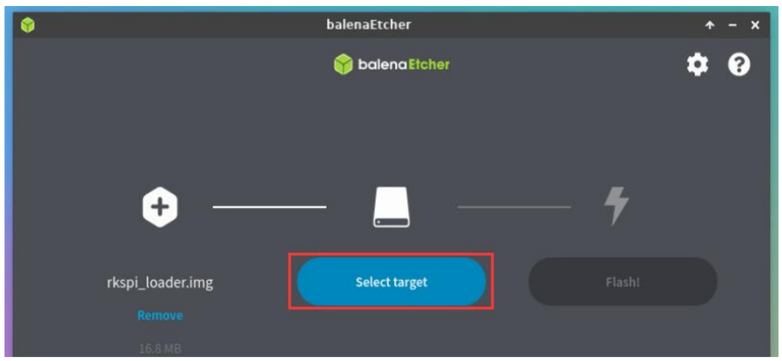

b. Then select the device name of the TF card **/dev/mmcblk1**



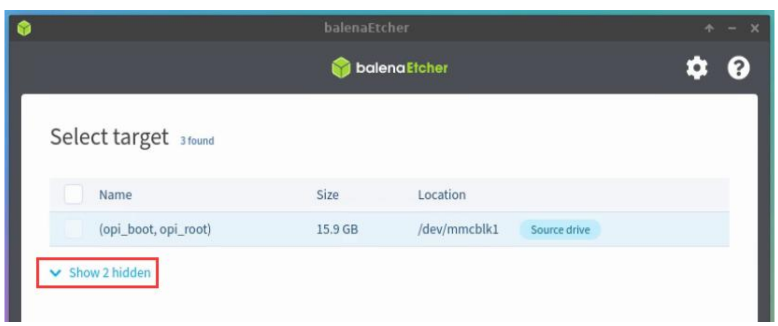

c. The interface after opening the TF card is as follows:



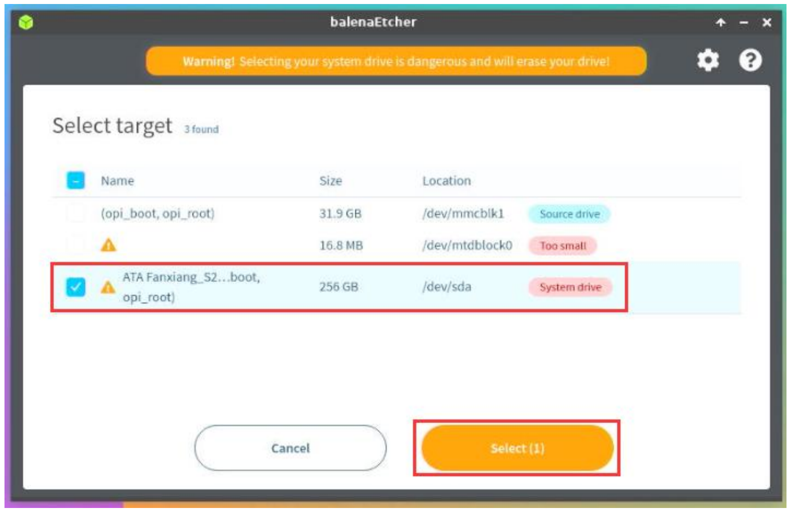

d. Then click **Select target**



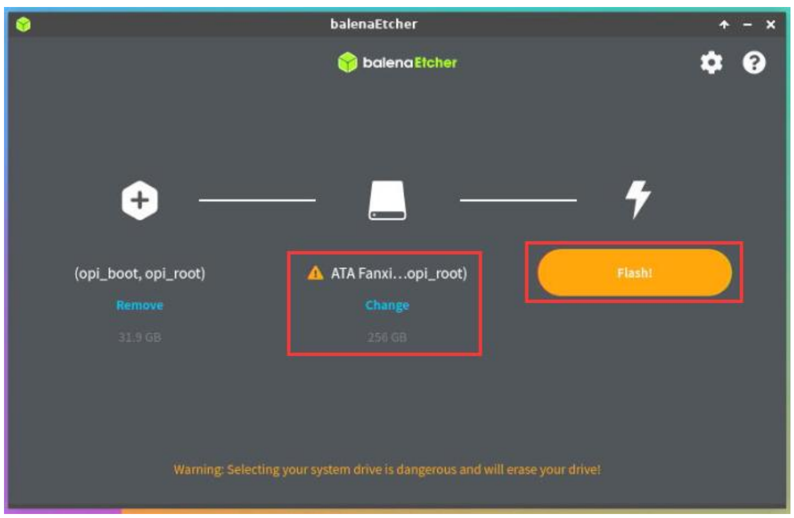

e. Then click **Show 2 hidden** to open more options for storage devices



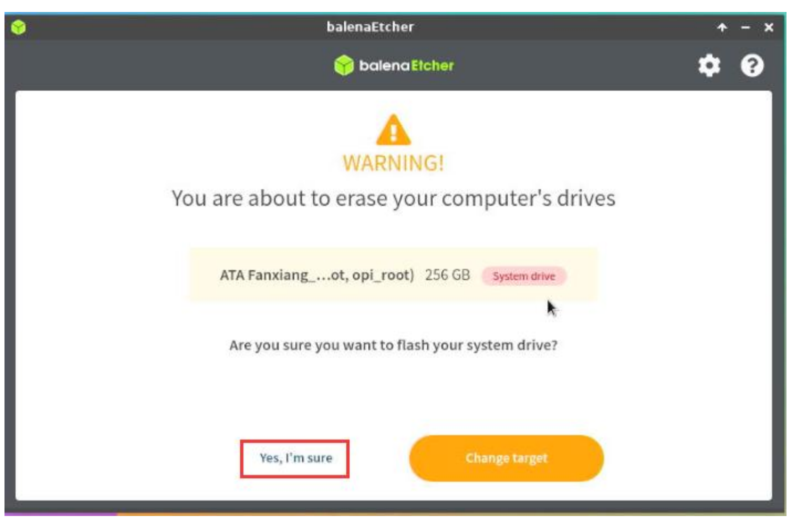

f. Then select the SSD device name **/dev/sda**, and click **Select**



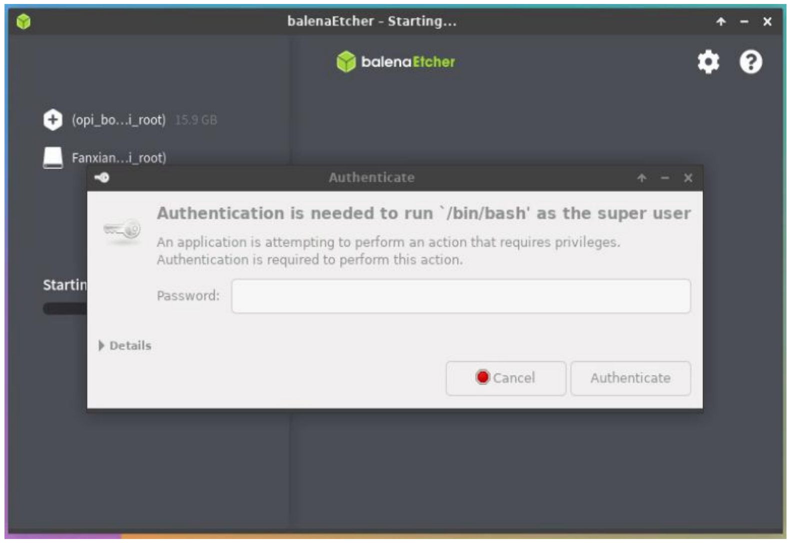

g. Then click **Flash**



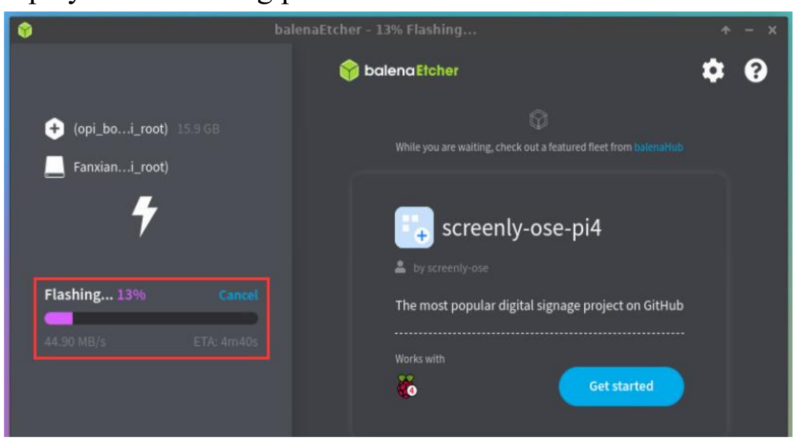

h. Then click **Yes, I'm sure**

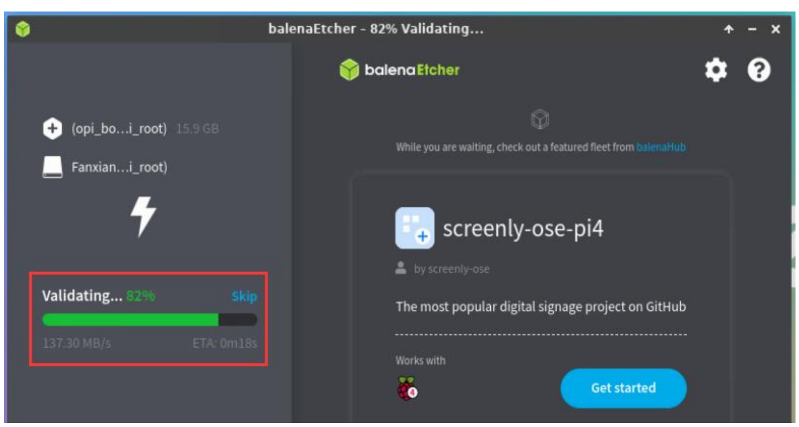



i. Then enter the password orangepi of the linux system on the development board, and it will start burning the linux image to the SSD

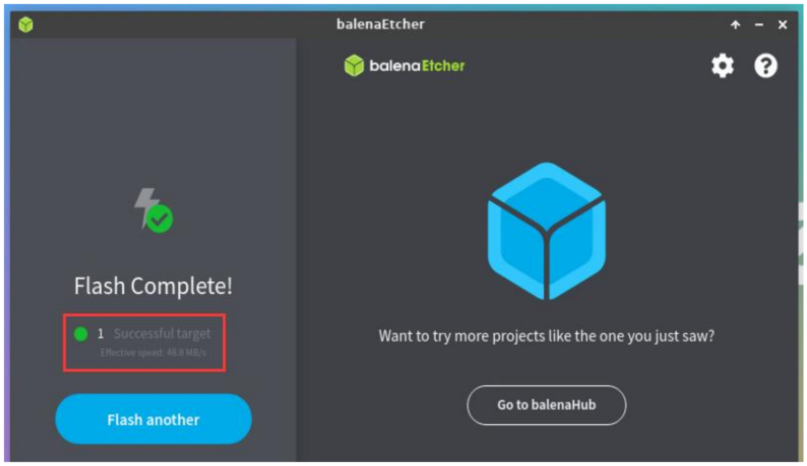



j. The display of the burning process is as follows:



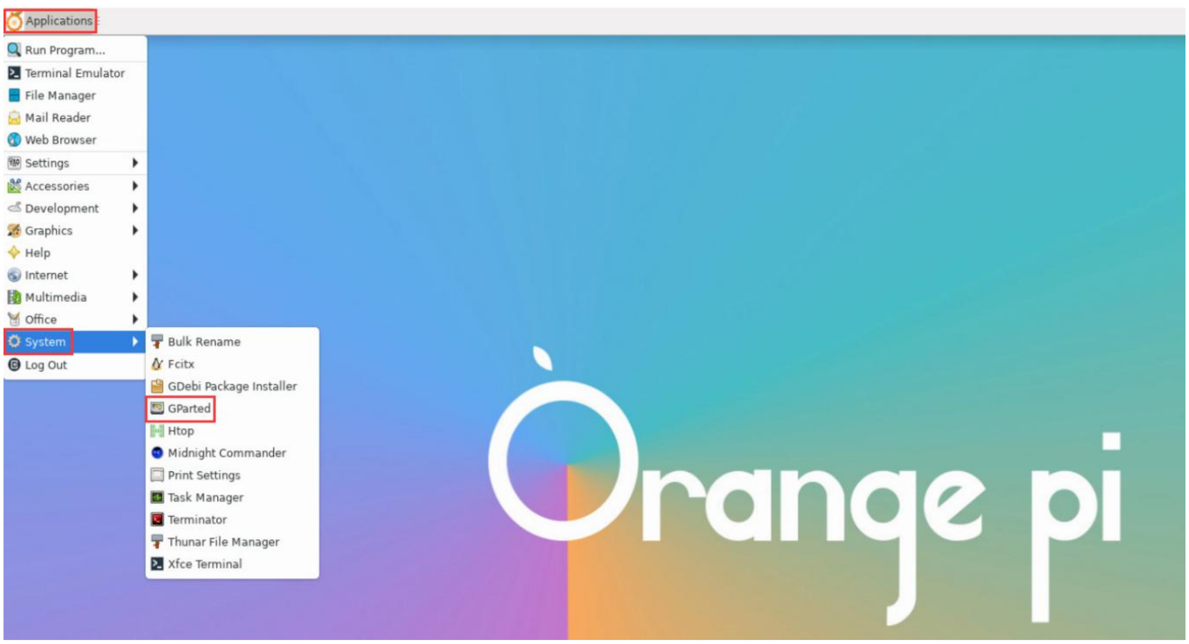




k. The display after burning is as follows:



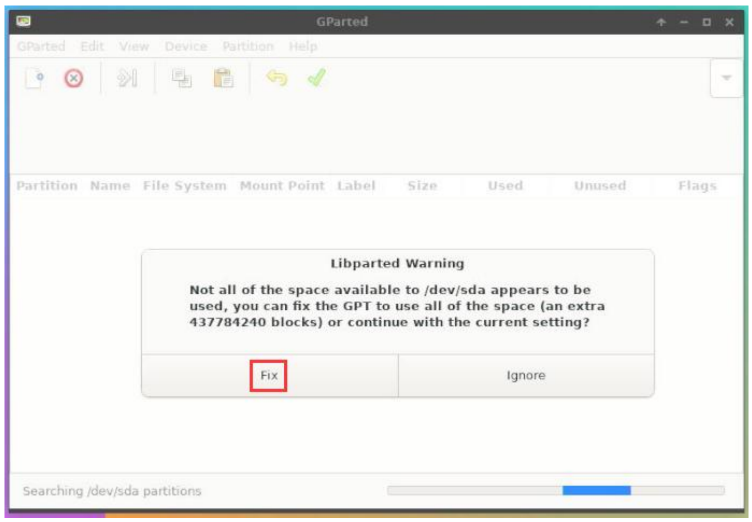

l. Then you need to expand the capacity of the rootfs partition in the SSD. The steps are as follows:

a) First open **GParted**

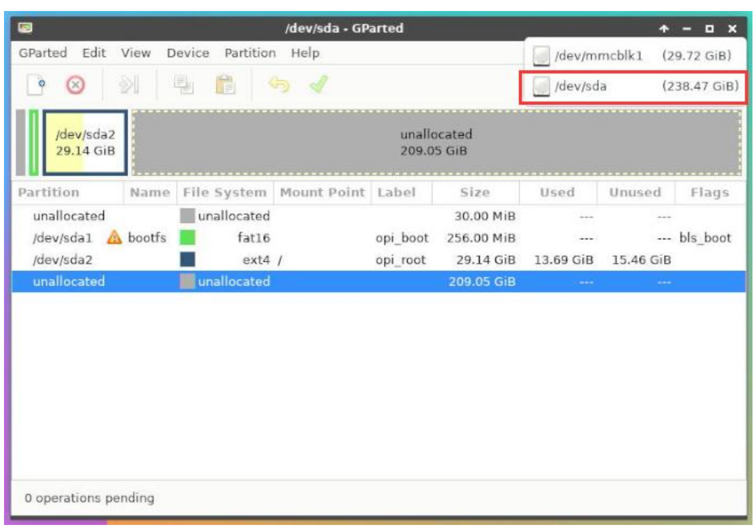



b) Then enter the password **orangepi** of the linux system, and click **Authenticate**

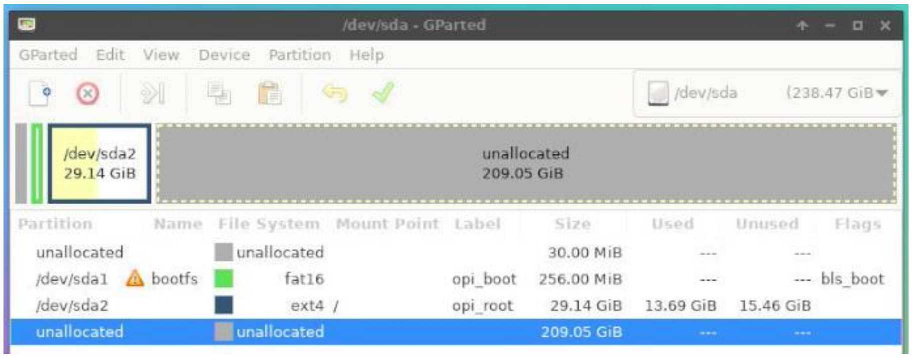

c) Then click **Fix**



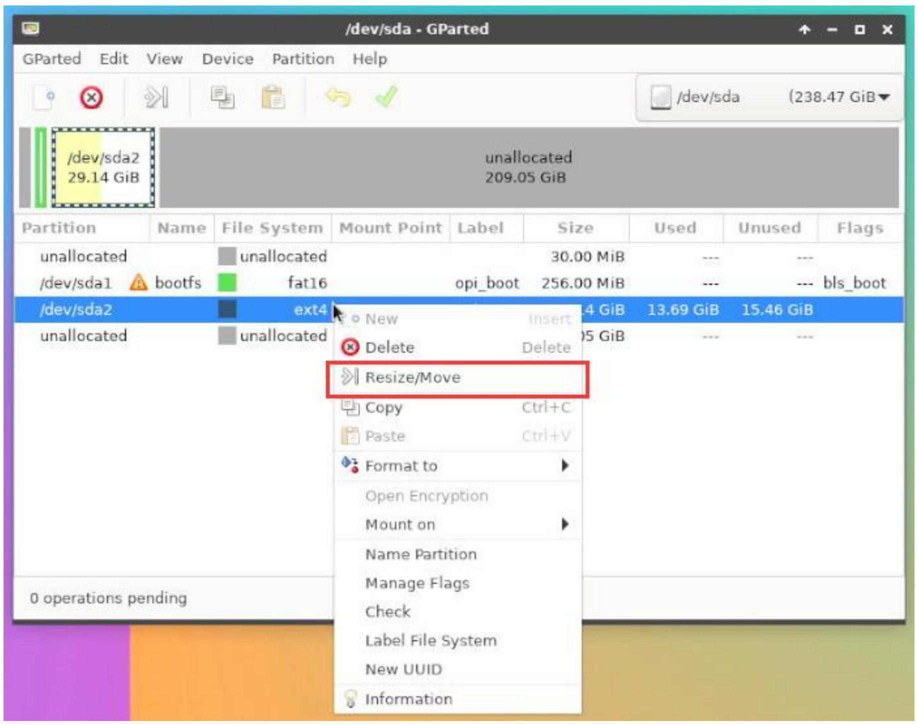

d) Then choose SSD



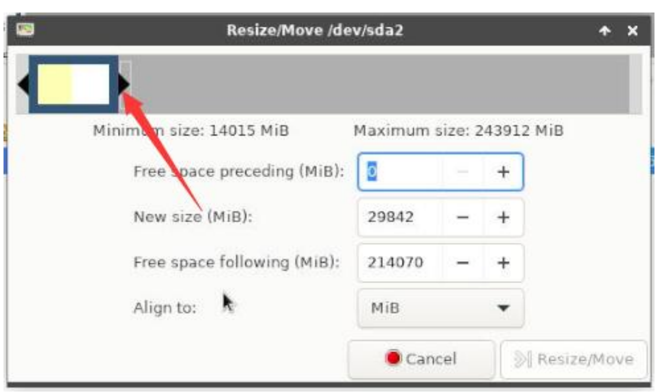

e) The display interface after selecting SSD is as follows:



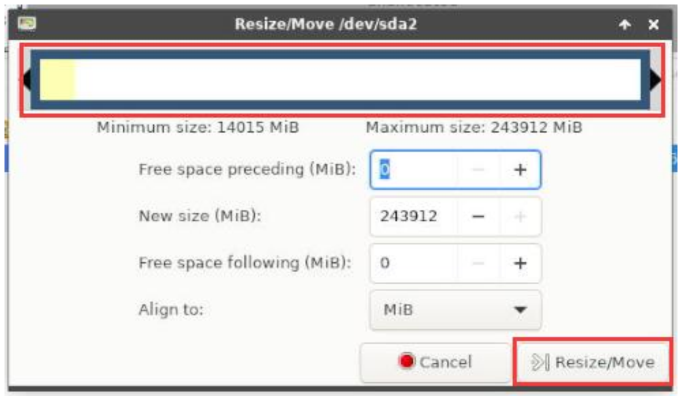
f) Then select the **/dev/sda2** partition, then right-click, and then select **Resize/Move**



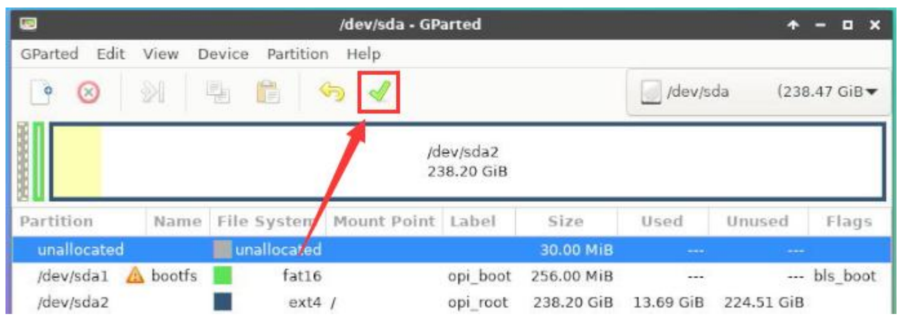
g) Then drag the capacity to the maximum at the position shown in the figure below



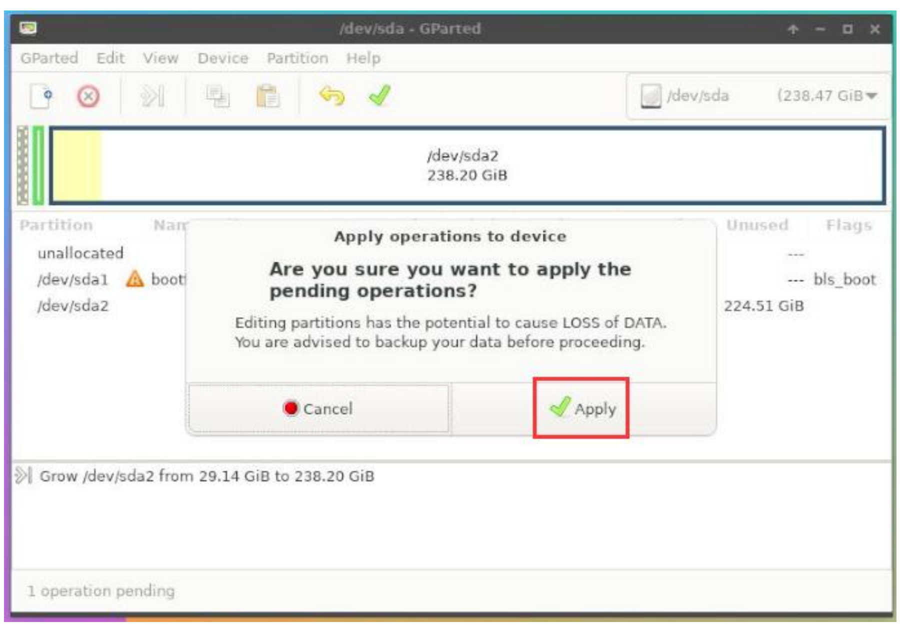
h) Then click **Resize/Move**



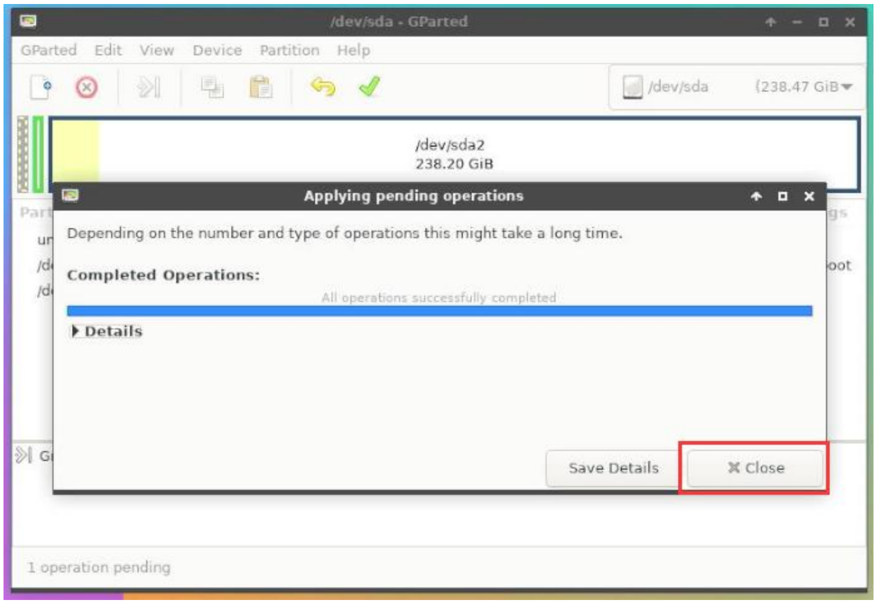

i) Then click the green √




j) Click **Apply**




k) Then click **Close**

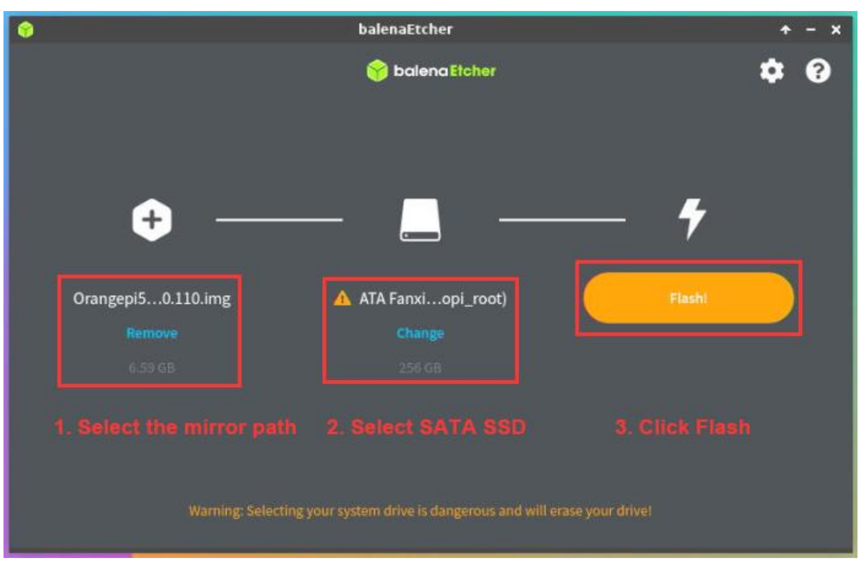



m. At this point, you can use the **sudo poweroff** command to shut down. Then please pull out the TF card, and then short press the power button to turn on, and then the linux system in SPIFlash+STAT SSD will be started.

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

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

b. Then use balenaEtcher to burn




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

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

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

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

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

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

c) Then unmount the boot partition of the SATA SSD

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

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

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

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

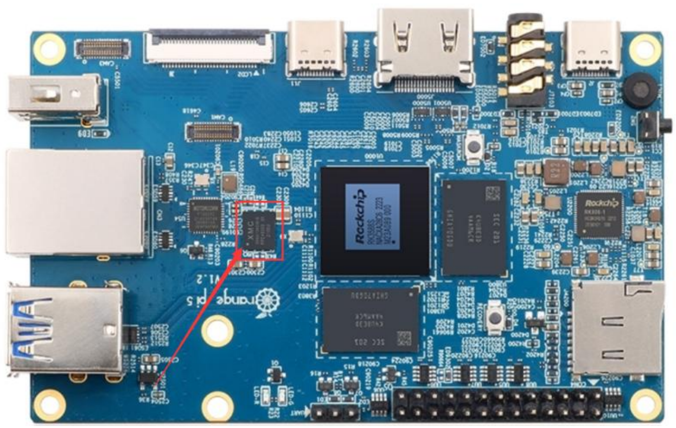
1) First, you need to prepare a USB storage device, such as a U disk

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

3) Then insert the USB storage device with the programmed Linux system into the USB3.0 interface of the development board. **Note, please do not insert the USB storage**

**device with the programmed system into other USB interfaces of the development board. Only the USB3.0 interface shown supports booting the Linux system**


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



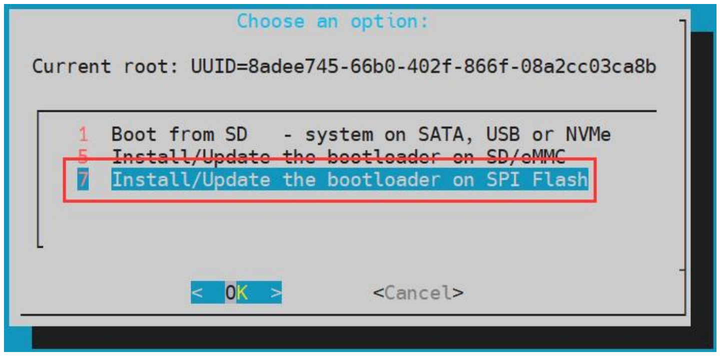
5) Burning the u-boot image to SPIFlash needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please refer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

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

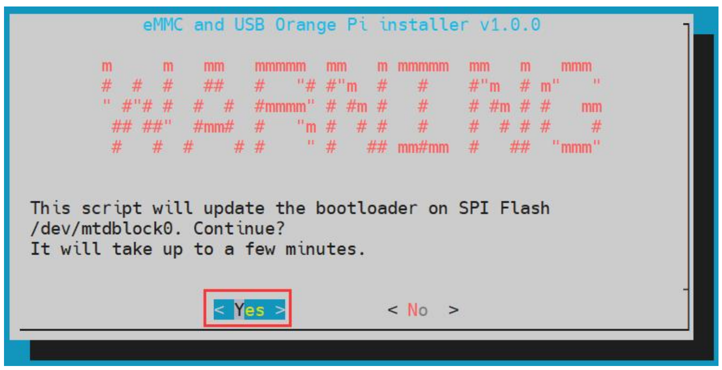
a. Run **nand-sata-install** first, **ordinary users remember to add sudo permission**

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

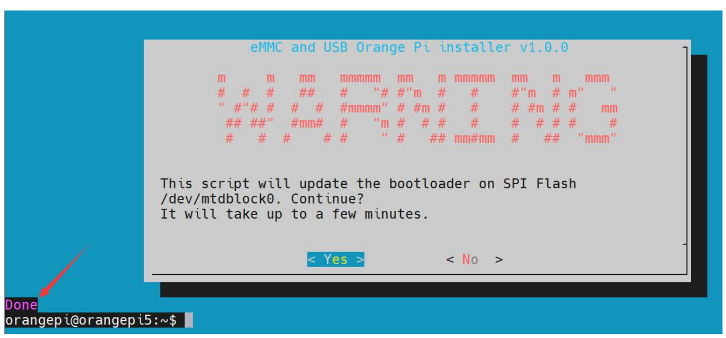
b. Then select **7 Install/Update ther bootloader on SPI Flash**




c. Then select **<Yes>**




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




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

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

```
orangepi@orangepi:~$ df -h
Filesystem            Size    Used Avail Use% Mounted on
```

```
udev            3.8G  8.0K  3.8G   1% /dev
tmpfs           769M  588K  769M   1% /run
/dev/sda2        15G  1.6G   13G  11% /
tmpfs           3.8G     0  3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/zram2      3.7G   60K  3.5G   1% /tmp
/dev/sda1       256M  111M  146M  44% /boot
/dev/zram1      194M  9.0M  171M   5% /var/log
tmpfs           769M     0  769M   0% /run/user/1000
```

## 2. 8. How to burn Android image to TF card

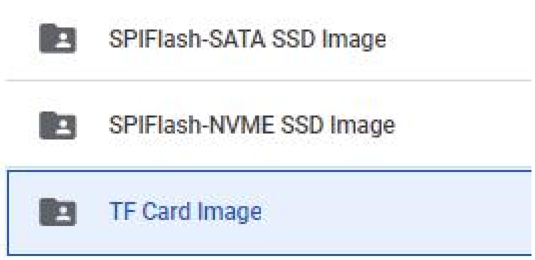
1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card of SanDisk and other brands

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 ensure that the version of the SDDiskTool tool is the latest v1.72**

4) Then download the Android12 image from the Orange Pi download page

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



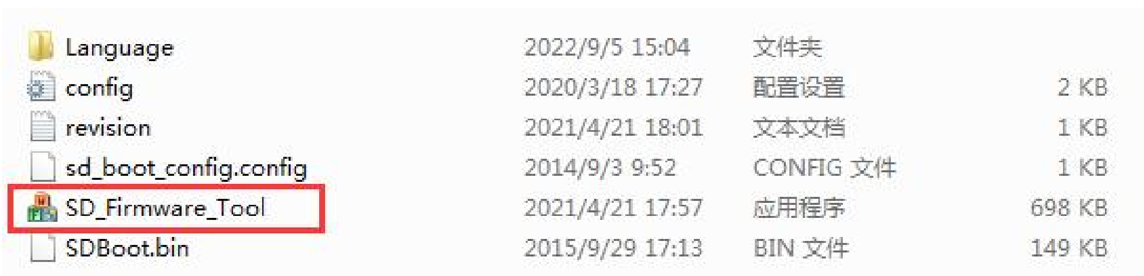

b. After entering the **TF card image** folder, you can see the following two images, the difference between them is:

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

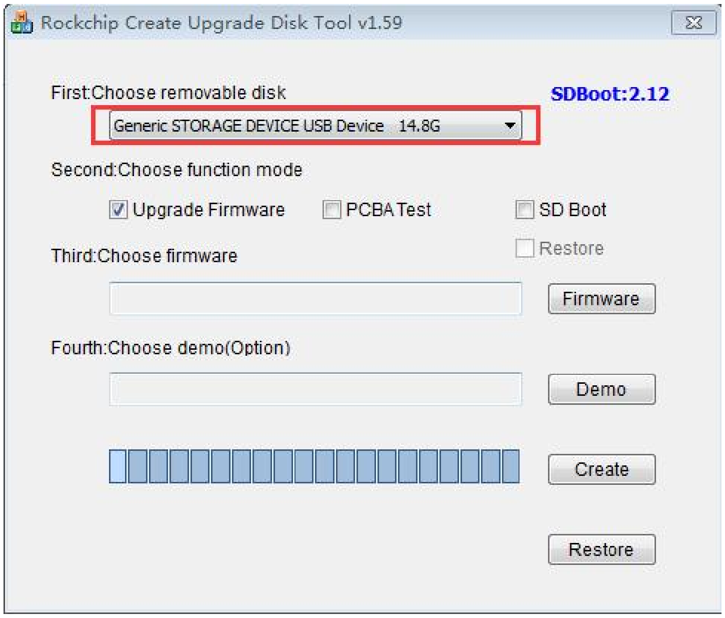
b) If you want to use LCD screen, please choose image with lcd


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

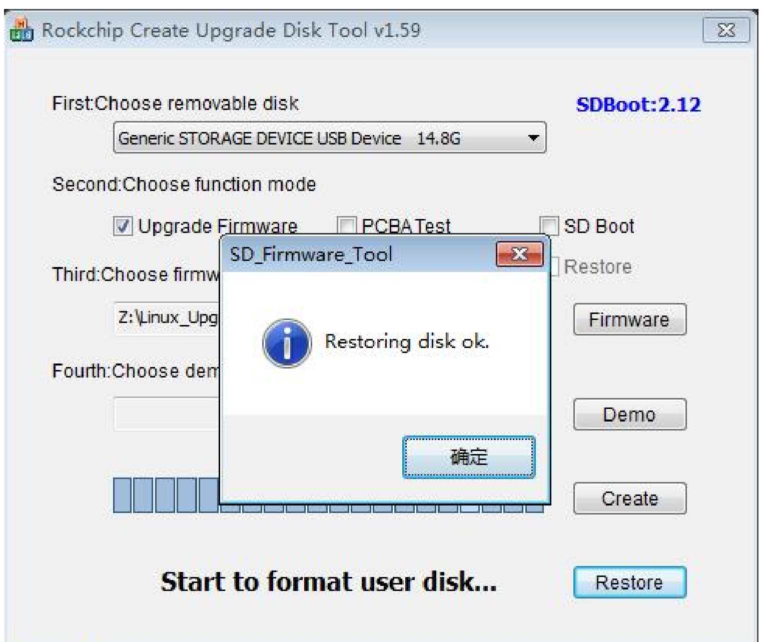
6) Then use decompression software to decompress **SDDiskTool_v1.72.zip**, this software does not need to be installed, just find **SD_Firmware_Tool.exe** in the decompressed folder and open it




7) After opening **SDDiskTool**, if the TF card is recognized normally, the inserted disk device will be displayed in the **"Select Removable Disk Device"** column. **Please make sure that the displayed disk device is consistent with the drive letter of the TF card you want to burn**, 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

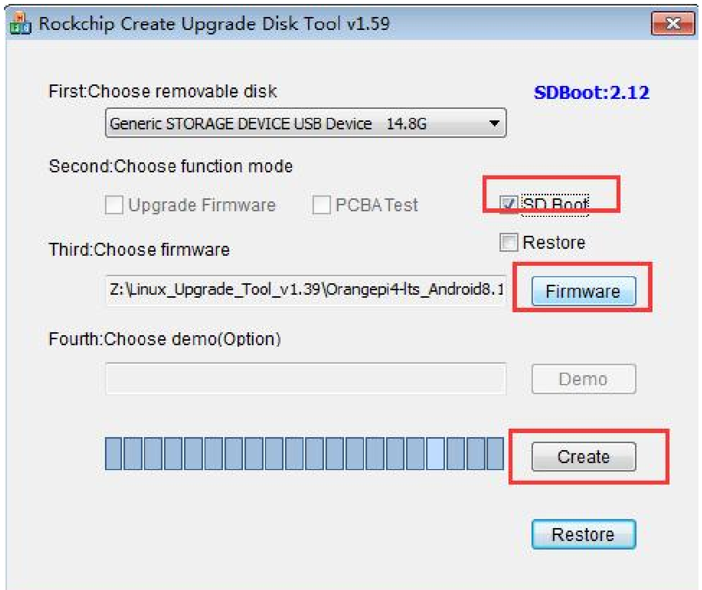



9) Then start writing the Android image to the TF card

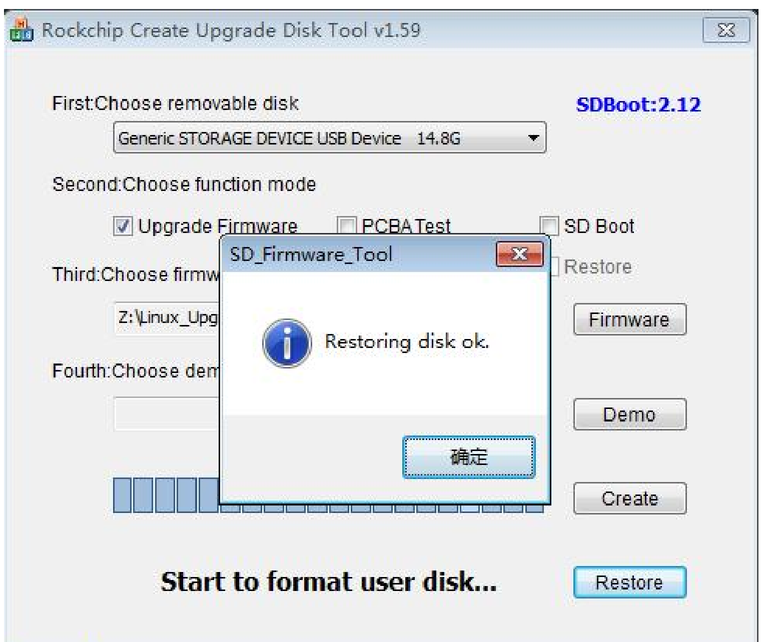
a. First check "**SD Boot**" in "**Select Function Mode**"

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

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




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. 9. How to burn Android image to SPI Flash+NVMe SSD

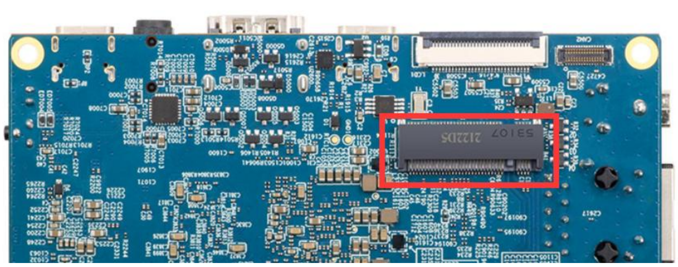
**Note that all the following operations are performed on a Windows computer.**

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

a. The M.2 2230 SSD is as follows

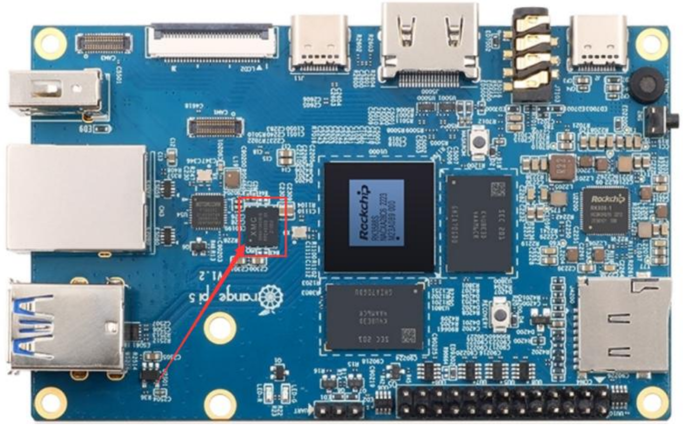

b. The M.2 2242 SSD is as follows


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

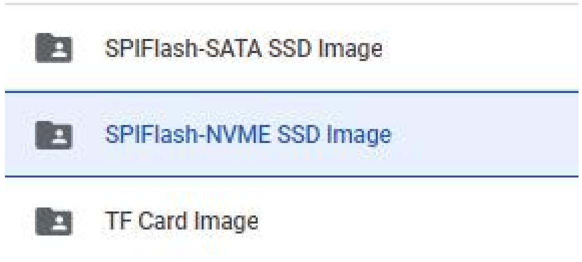


4) It is also necessary to prepare a data cable with a good quality Type-C interface

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

6) Then download the image of Android12

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




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

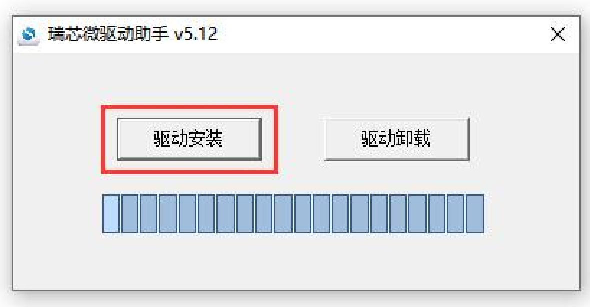
   a) The image without lcd is specially used for HDMI display and supports 8K display. If you do not use the LCD screen, please download the image without lcd

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

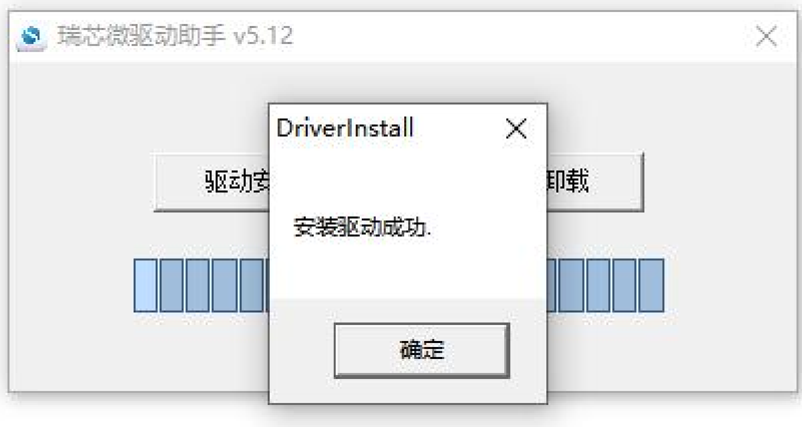

7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it


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

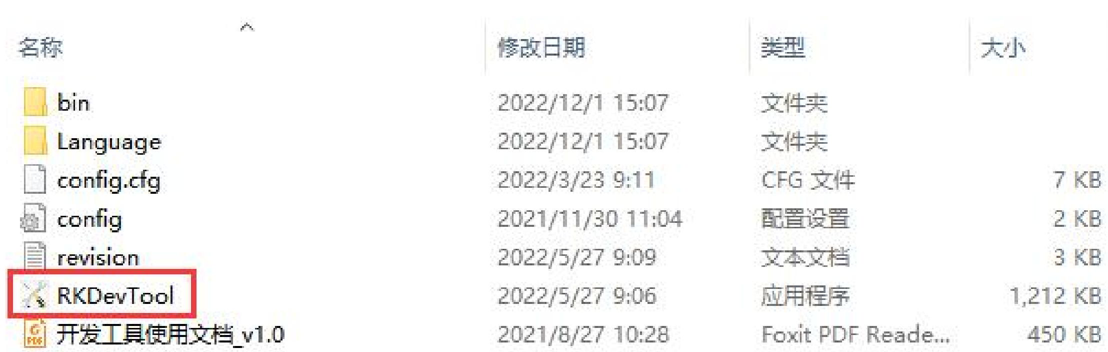
a. Click the "**Driver Install**" button



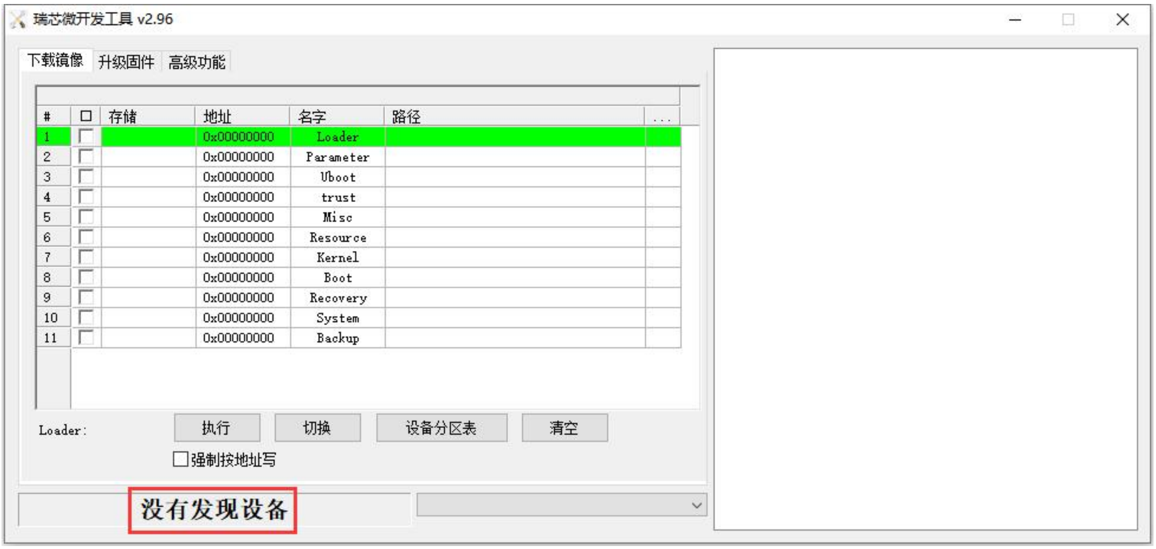

b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button.




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

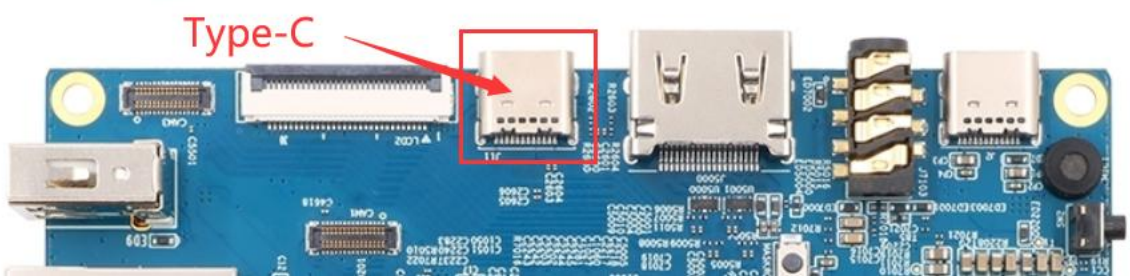



10) After opening the **RKDevTool** burning tool, because the computer has not been connected to the development board through the Type-C cable at this time, the lower left corner will prompt "**No device found**"



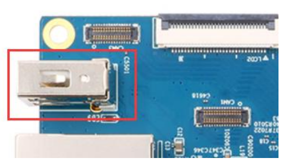

11) Then start burning the Android image to SPIFlash+NVMe SSD

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

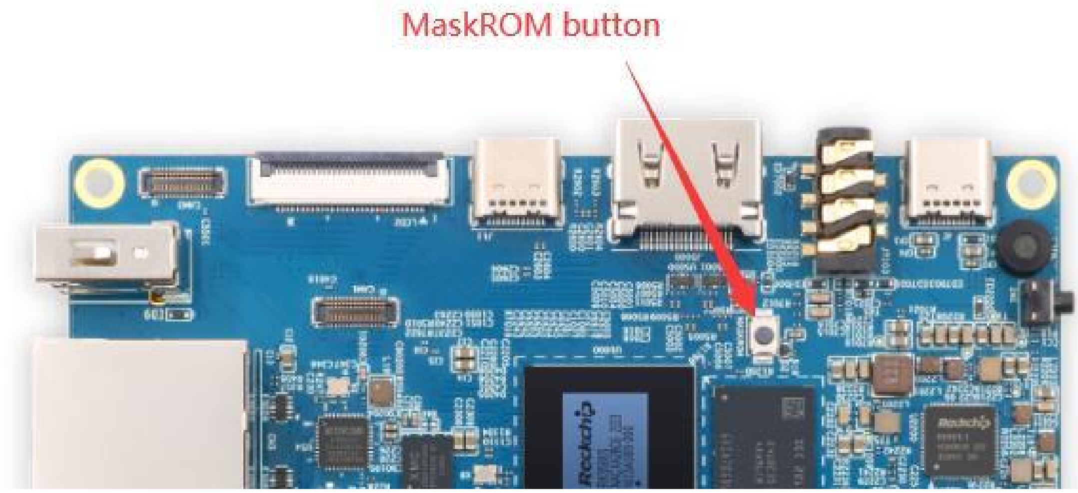



   b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

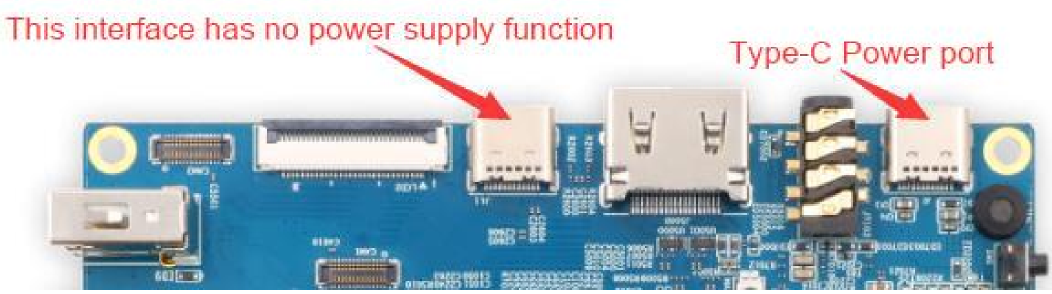
   c. Also need to ensure that the white USB2.0 interface in the position shown below is not plugged into a USB device



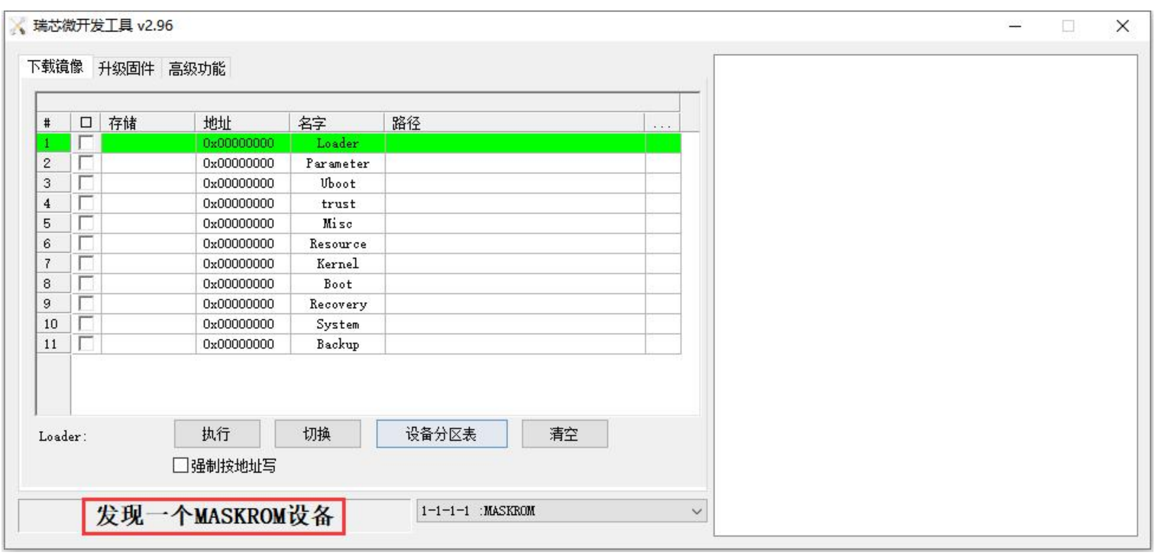
   d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:




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



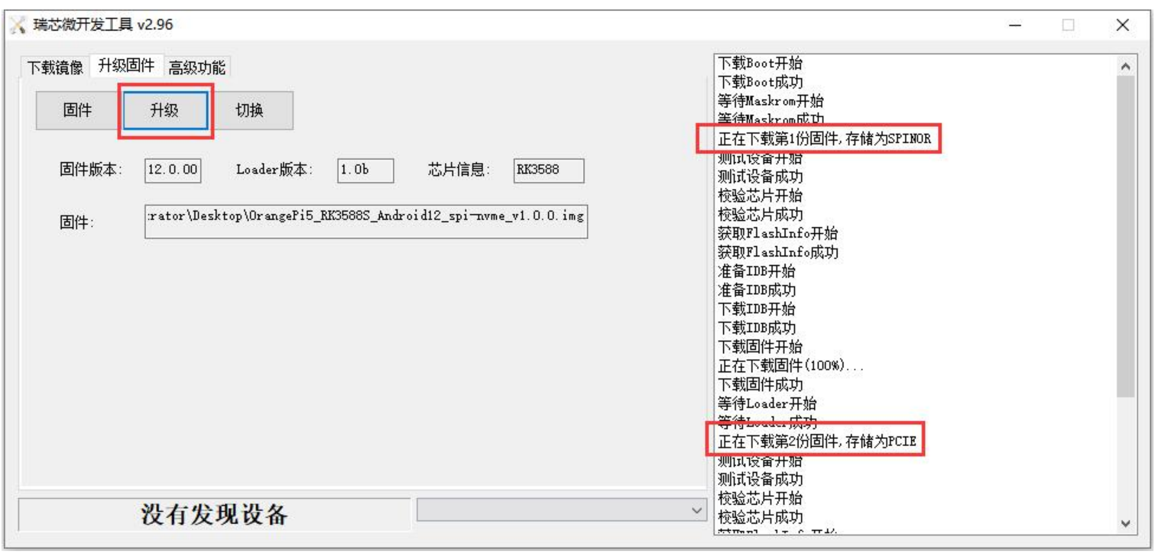

f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**




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


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


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




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

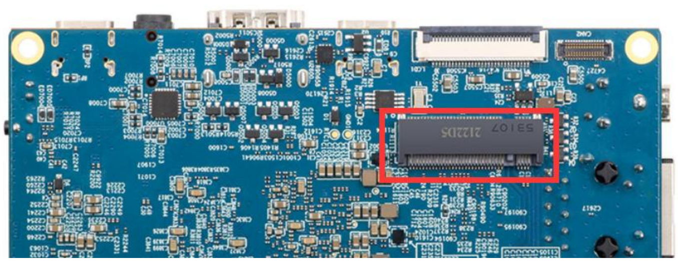
**Note that all the following operations are performed on a Windows computer.**

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

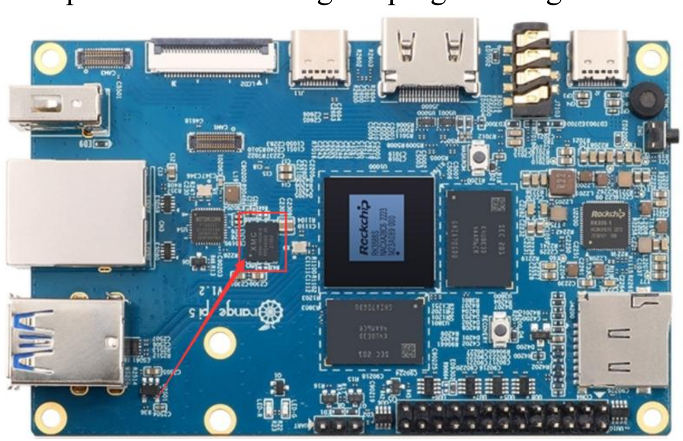
a. The M.2 2242 SSD is as follows


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


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



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

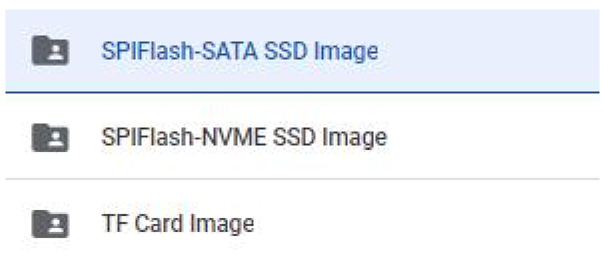


4) It is also necessary to prepare a data cable with a good quality Type-C interface

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

6) Then download the image of Android12

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




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

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

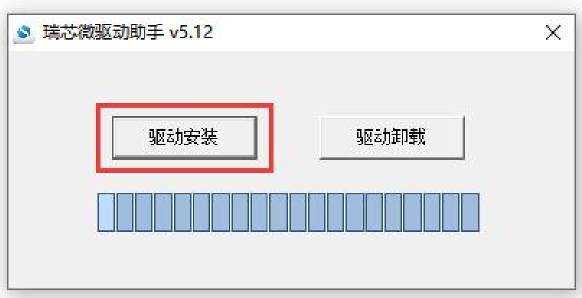
b) If you want to use LCD screen, please choose image with lcd


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

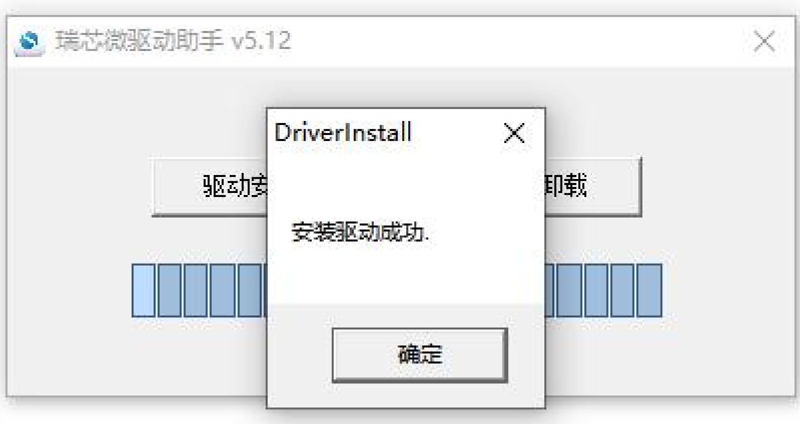
then find the **DriverInstall.exe** executable file in the decompressed folder and open it


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

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

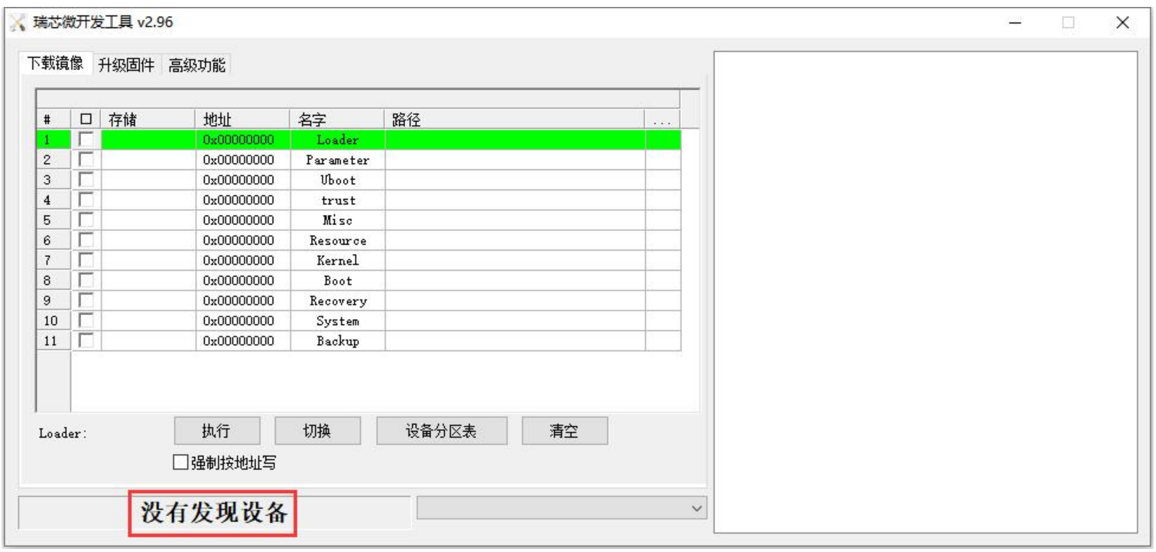



b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "OK" button.



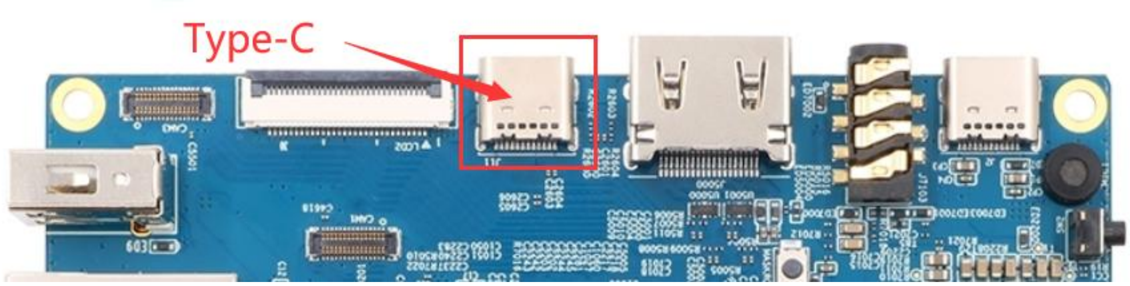

9) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it


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



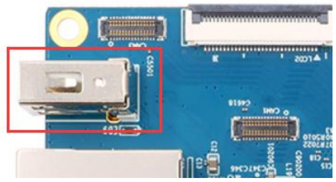

11) Then start burning the Android image to SPIFlash+NVMe SSD

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

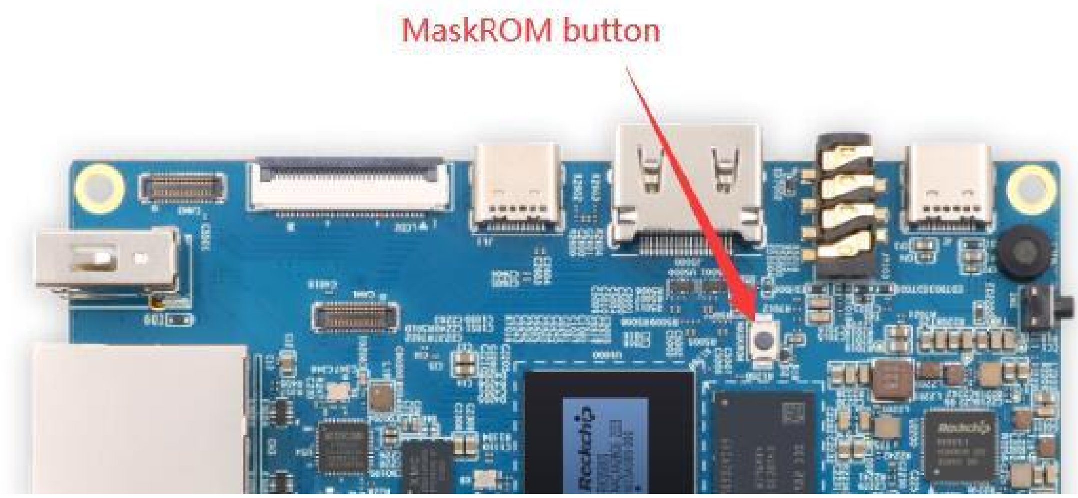



b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

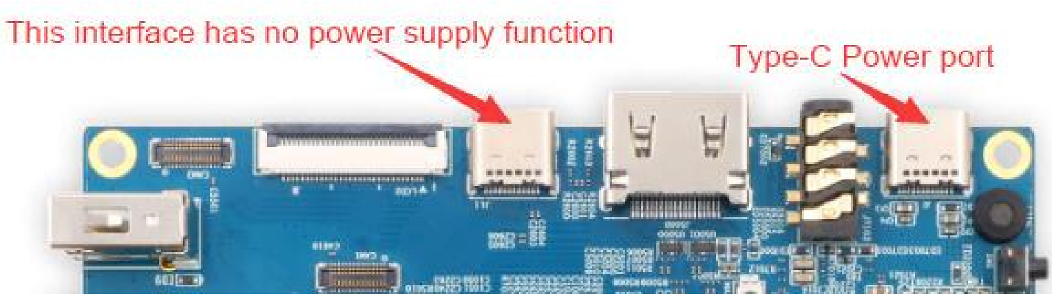
c. Also need to ensure that the white USB2.0 interface in the position shown below is not plugged into a USB device



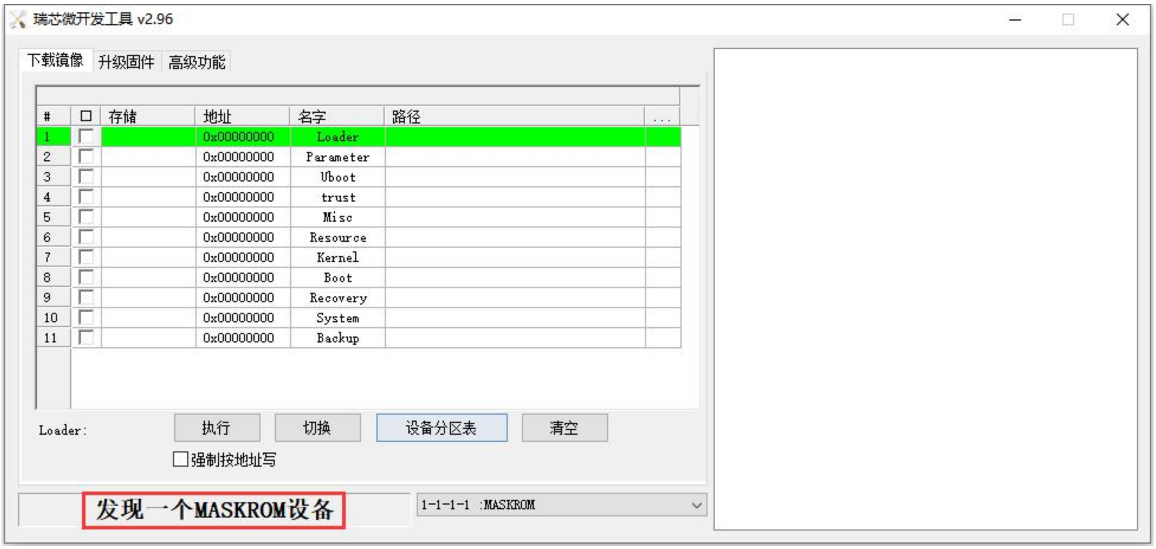
d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:




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



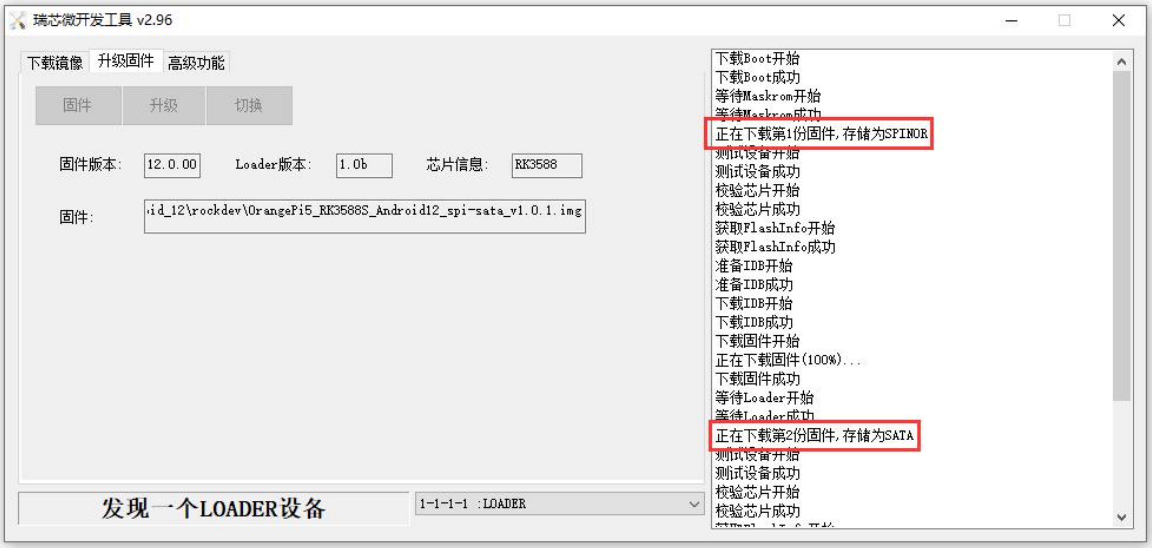

f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"




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


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


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




## 2.11. How to burn Orange Pi OS (Droid) image to TF card

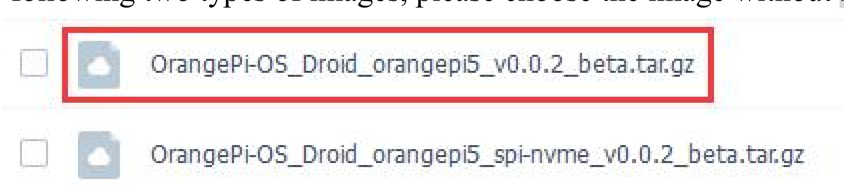
1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card of SanDisk and other brands

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 ensure that the version of the SDDiskTool tool is the latest v1.72**

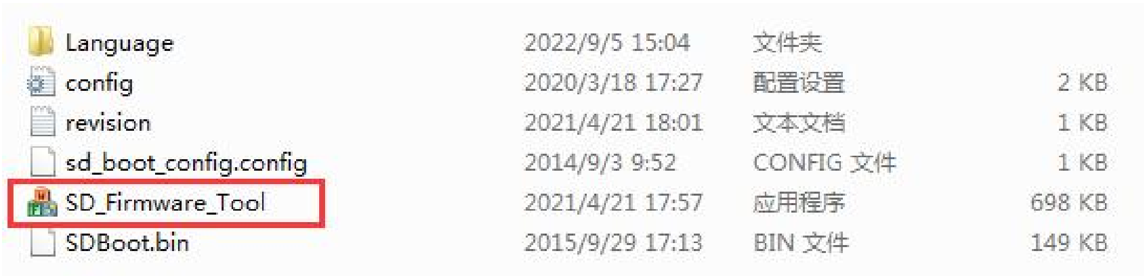
4) Then download the Orange Pi OS (Droid) image from the Orange Pi download page

a. After opening the download link of the Orange Pi OS (Droid) image, you can see the following two types of images, please choose the image without **spi-nvme**




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

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




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


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


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

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

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

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


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

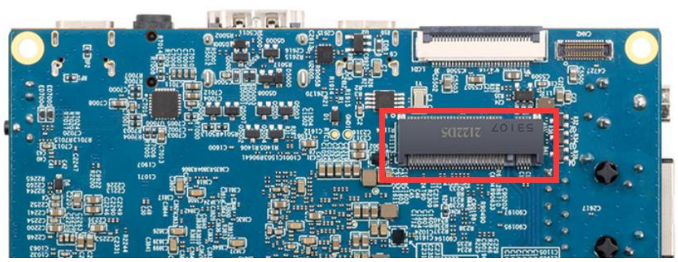

## 2. 12. Burn Orange Pi OS (Droid) image to SPIFlash+NVMe SDD

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

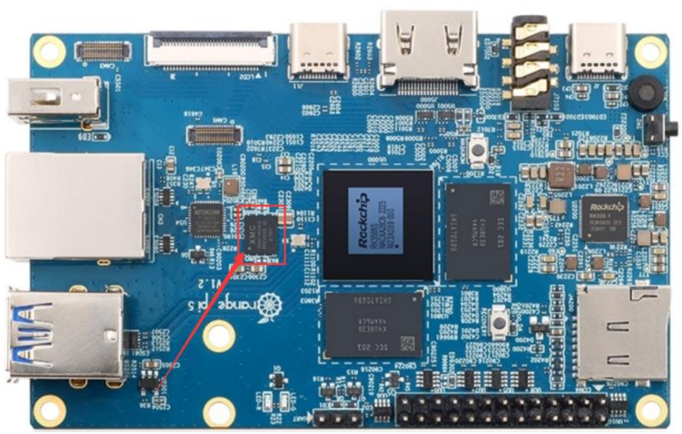
1) First, you need to prepare an NVMe SSD solid state drive
   a. The M.2 2230 SSD is as follows


   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

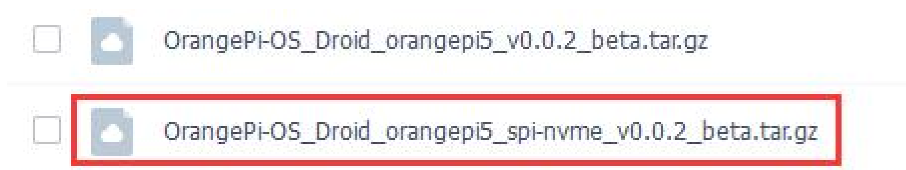


4) It is also necessary to prepare a data cable with a good quality Type-C interface

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

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

a. After opening the download link of the Orange Pi OS (Droid) image, you can see the following two types of images, please select the image with **spi-nvme** to download

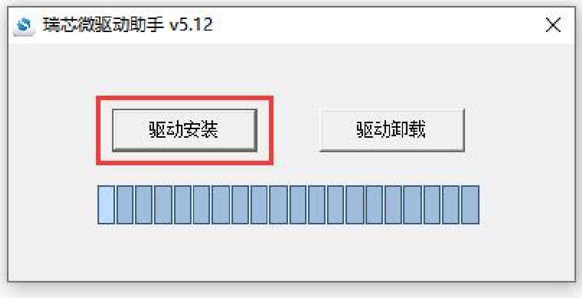



7) Then use the decompression software to decompress **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

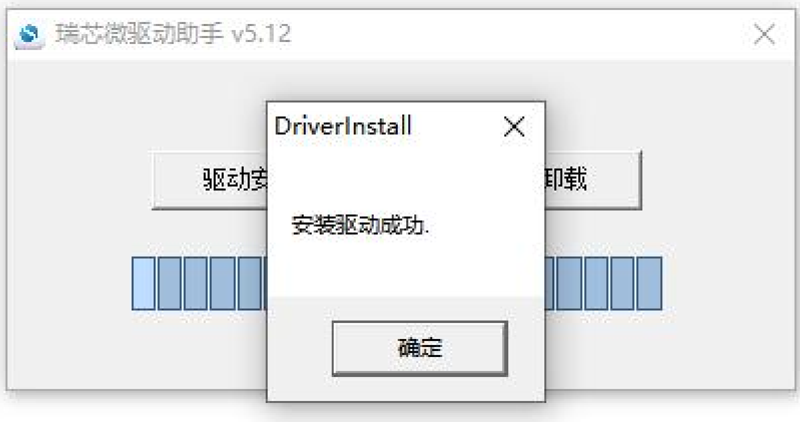
| 名称 | 修改日期 | 类型 | 大小 |
|---|---|---|---|
| ADBDriver | 2022/12/1 15:07 | 文件夹 | |
| bin | 2022/12/1 15:07 | 文件夹 | |
| Driver | 2022/12/1 15:07 | 文件夹 | |
| config | 2014/6/3 15:38 | 配置设置 | 1 KB |
| DriverInstall | 2022/2/28 14:11 | 应用程序 | 491 KB |
| Readme | 2018/1/31 17:44 | 文本文档 | 1 KB |
| revison |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

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

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

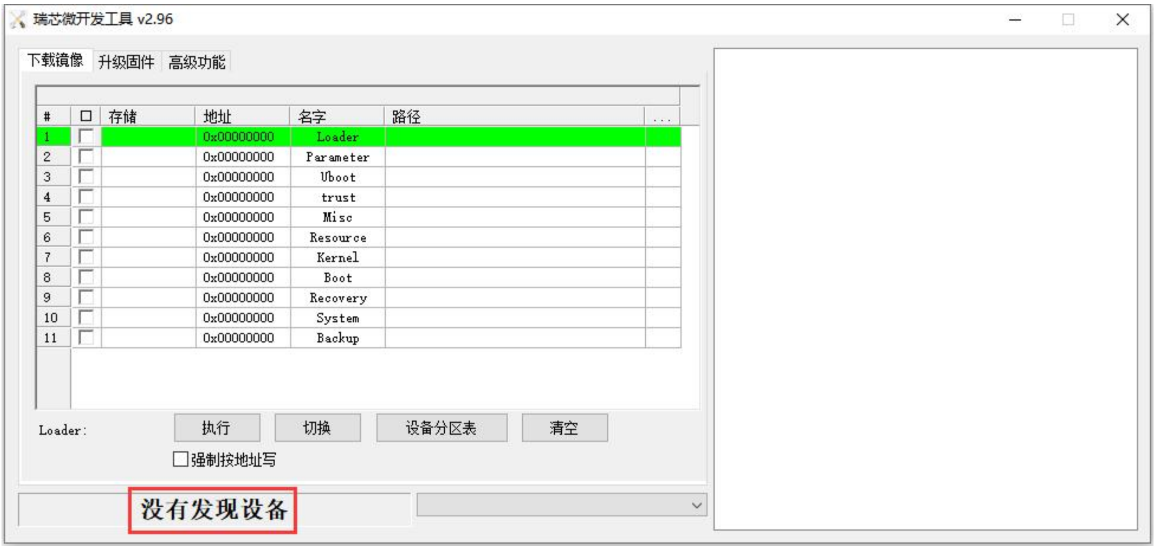



b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button.



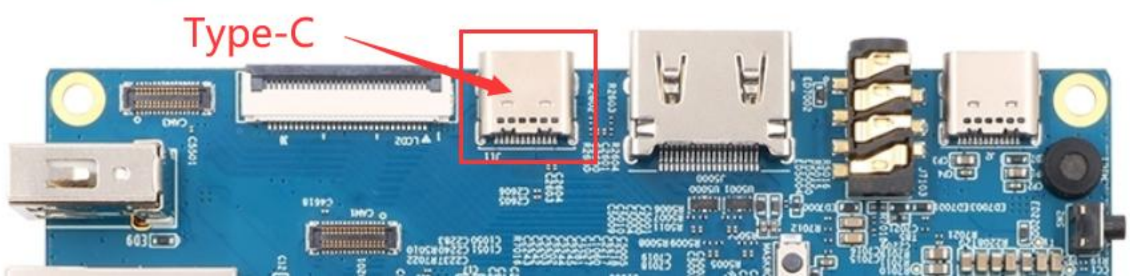

9) Then decompress **RKDevTool_Release_v2.96.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it


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



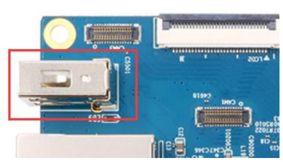

11) Then start burning the Orange Pi OS (Droid) image to SPIFlash+NVMe SSD

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

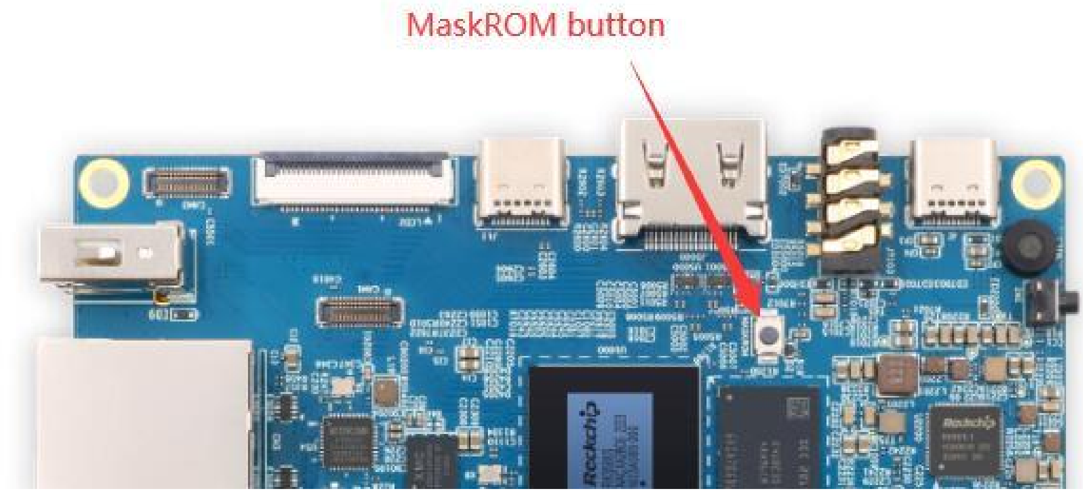



b. Make sure that the development board is not inserted into the TF card and not connected to the power supply

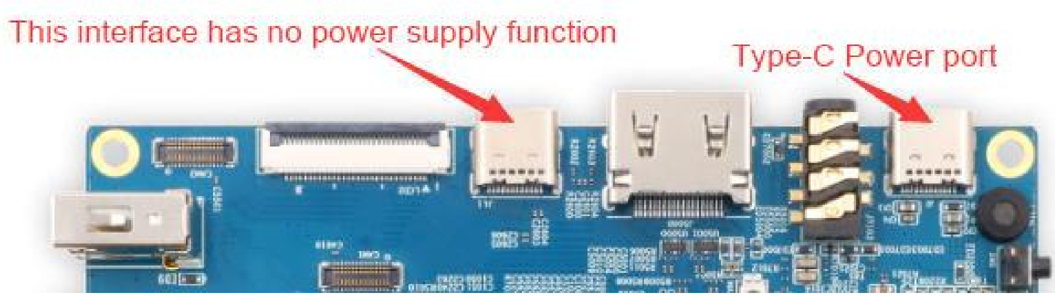
c. Also need to ensure that the white USB2.0 interface in the position shown below is not plugged into a USB device



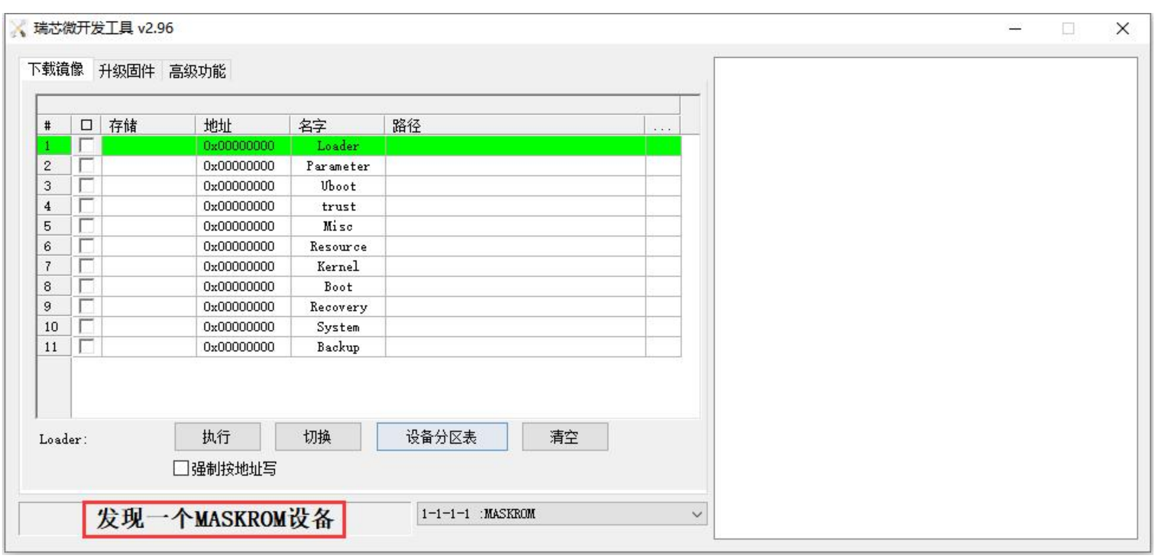
d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:




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



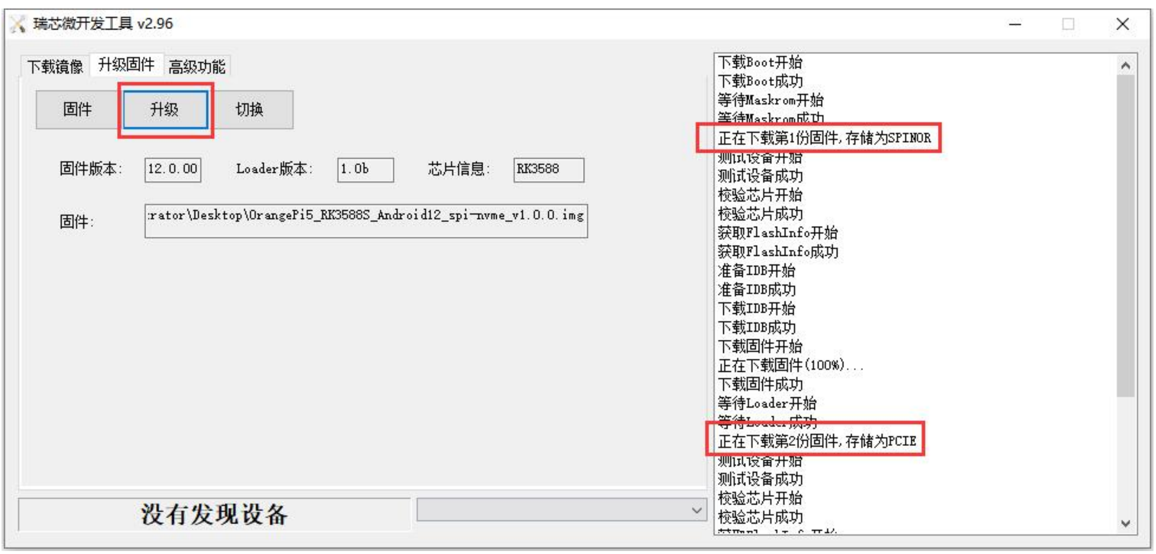

f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"




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


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


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




## 2. 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 buy an LCD screen, you can also use the LCD screen to display the system interface of the development board. If there is a Type-C to HDMI cable, the system interface of the development board can also be displayed through the Type-C interface.

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

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

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

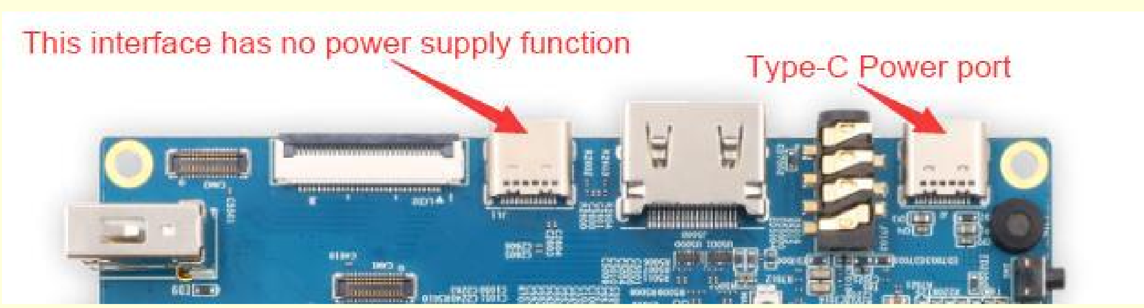
**Remember not to plug in a power adapter with a voltage output greater than 5V, as this will burn out the development board.**

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

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

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

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



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

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

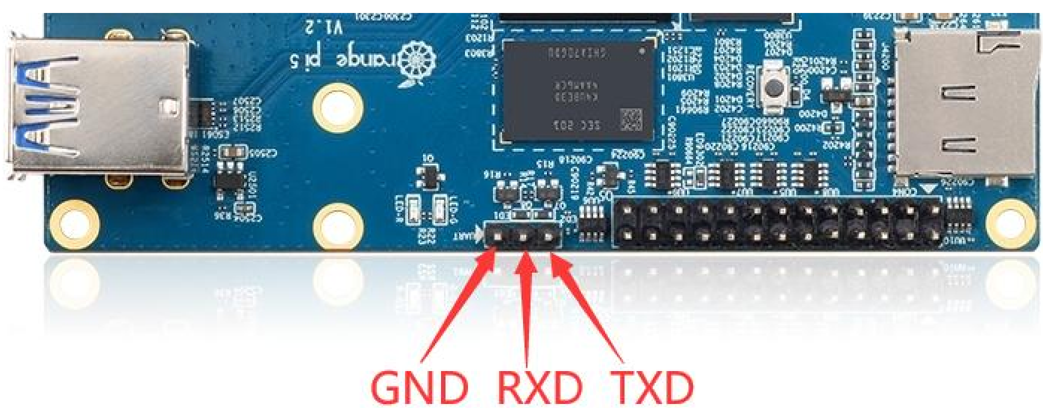
## 2. 14. How to use the debugging serial port

### 2. 14. 1. Connection instructions for the debug serial port

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

| |
|---|
| **For better compatibility, it is recommended to use the CH340 USB to TTL module instead of the CP2102 USB to TTL module.**<br>**Before purchasing a USB to TTL module, please confirm that the module supports a baud rate of 1500000.** |

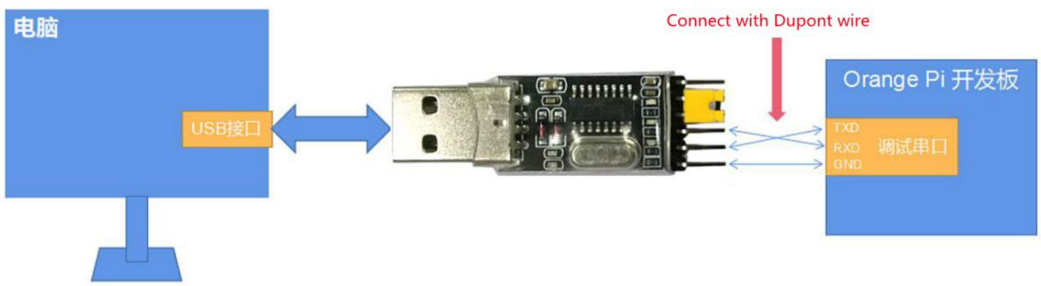

2) The corresponding relationship between GND, RXD and TXD pins of the debugging serial port of the development board is shown in the figure below



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

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

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




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

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

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

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

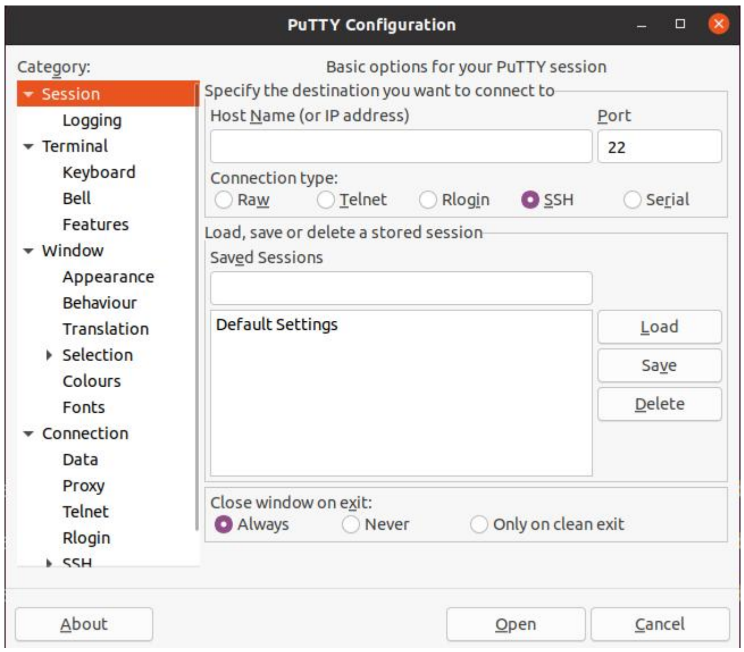
2) Then use the following command to install putty on Ubuntu PC

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

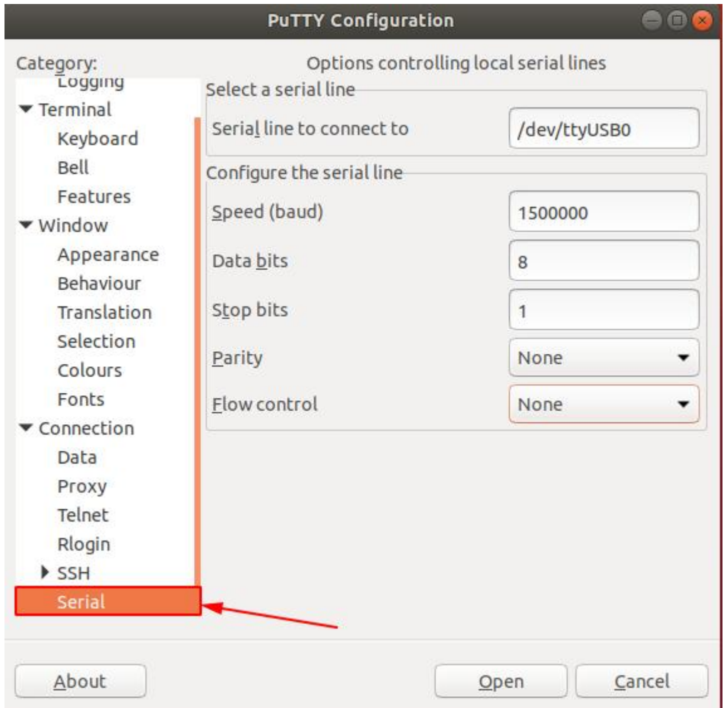
3) Then run putty, **remember to add sudo permission**

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

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




5) First select the setting interface of the serial port

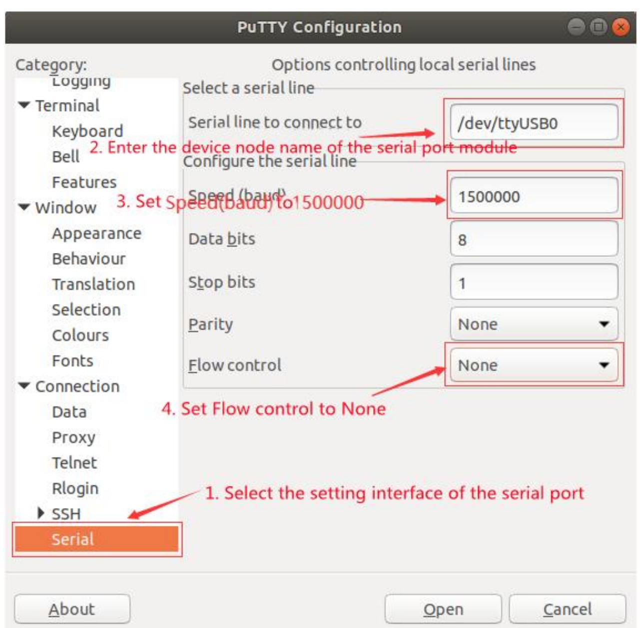



6) Then set the parameters of the serial port

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

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

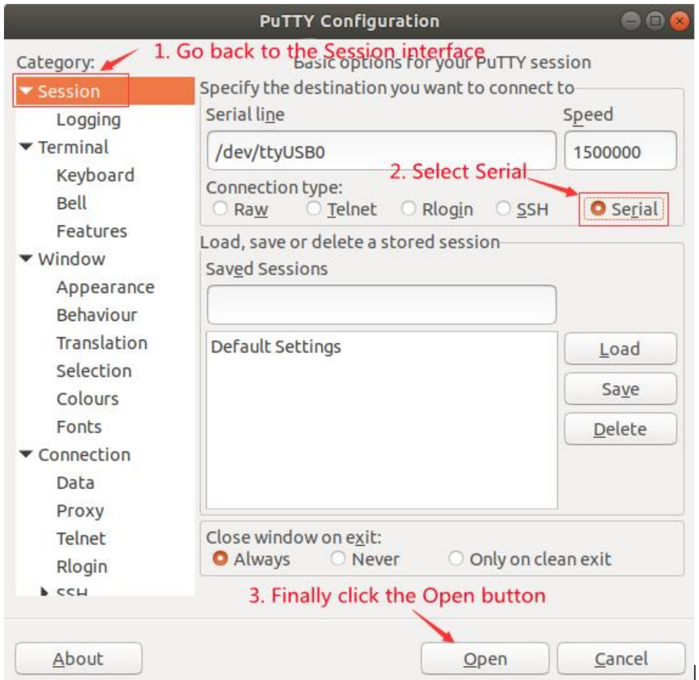
c. Set Flow control to None



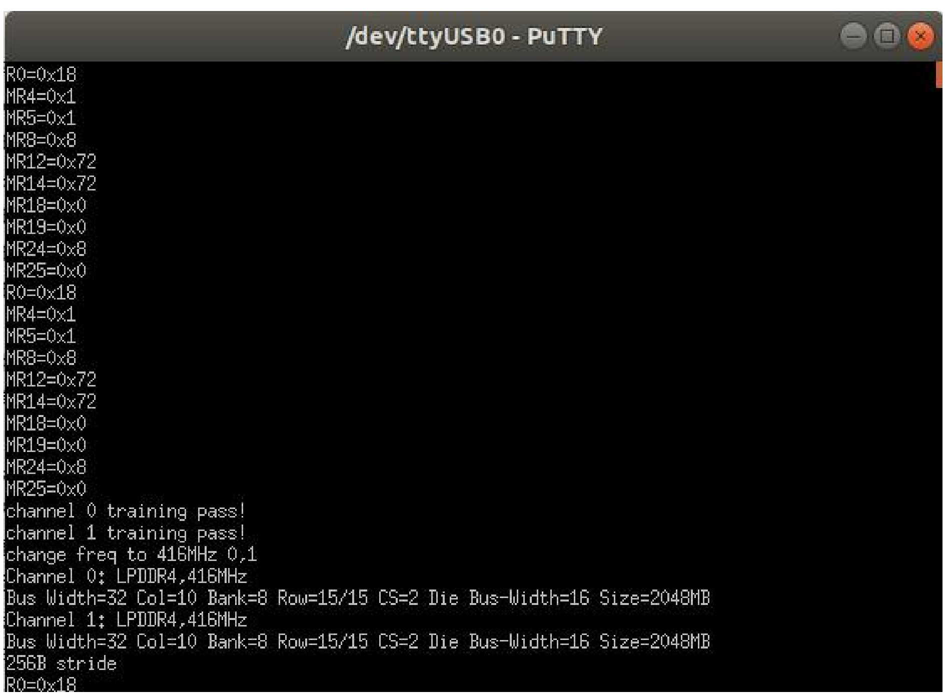
7) After setting the serial port setting interface, return to the Session interface

a. First select the **Connection type** as Serial

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




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



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

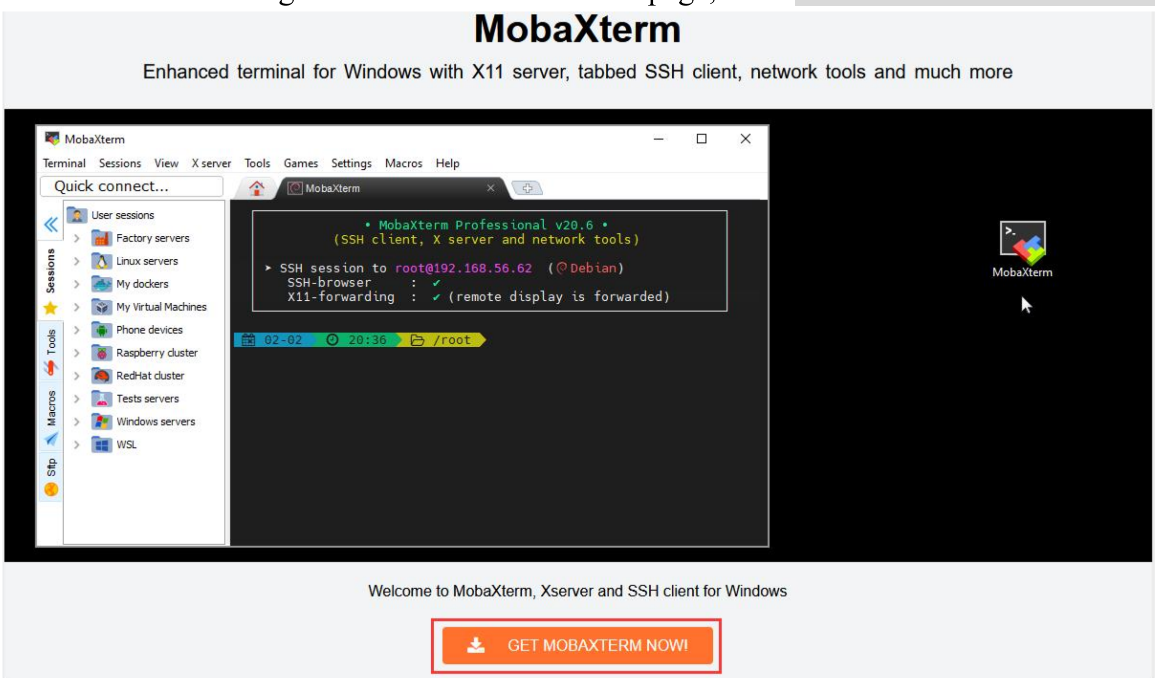
**There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without buying a serial number.**

1) Download MobaXterm

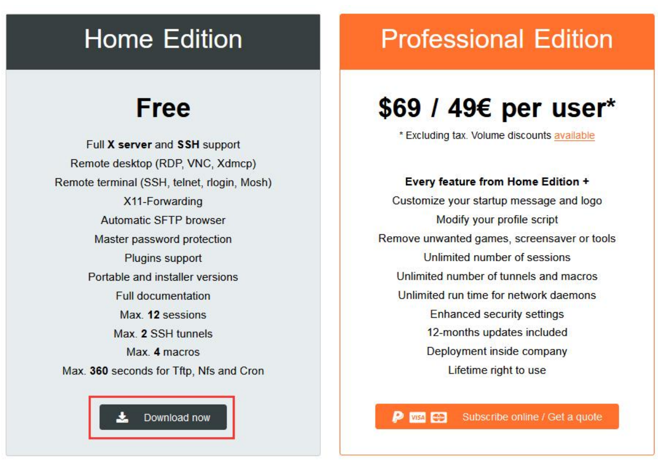
   a. Download MobaXterm website as follows

https://mobaxterm.mobatek.net

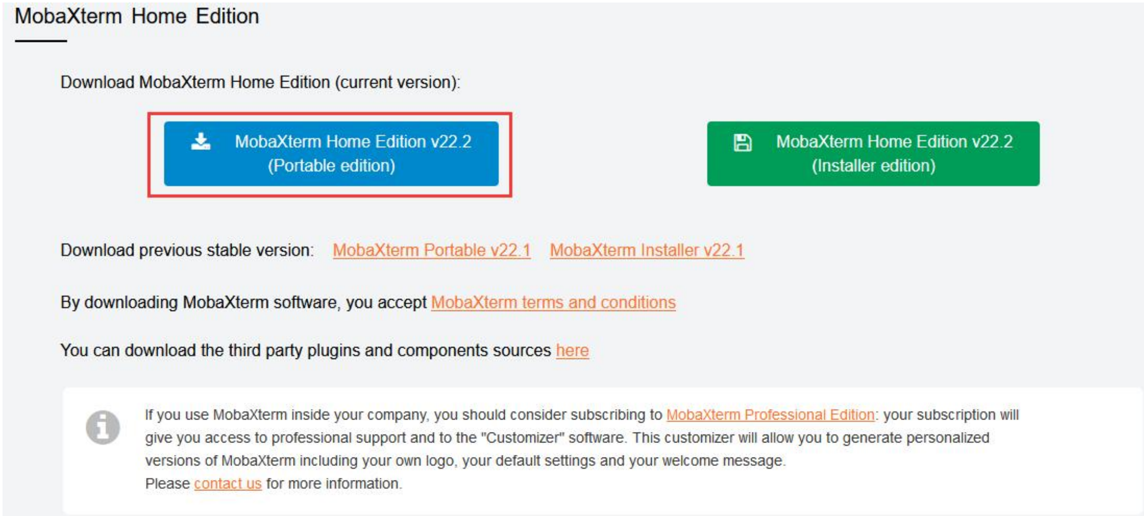
   b. After entering the MobaXterm download page, click **GET XOBATERM NOW!**




   c. Then choose to download the Home version



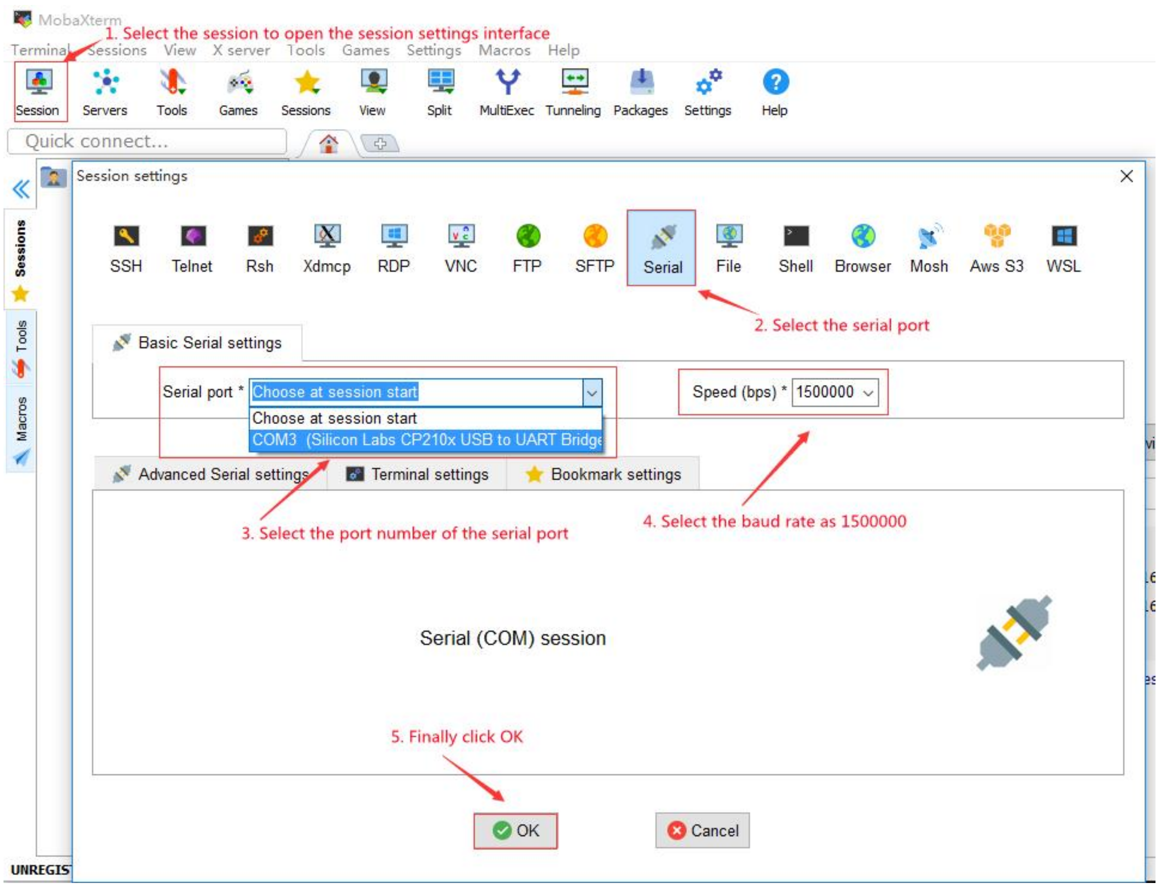
d. Then select Portable portable version, no need to install after downloading, just open it and use it



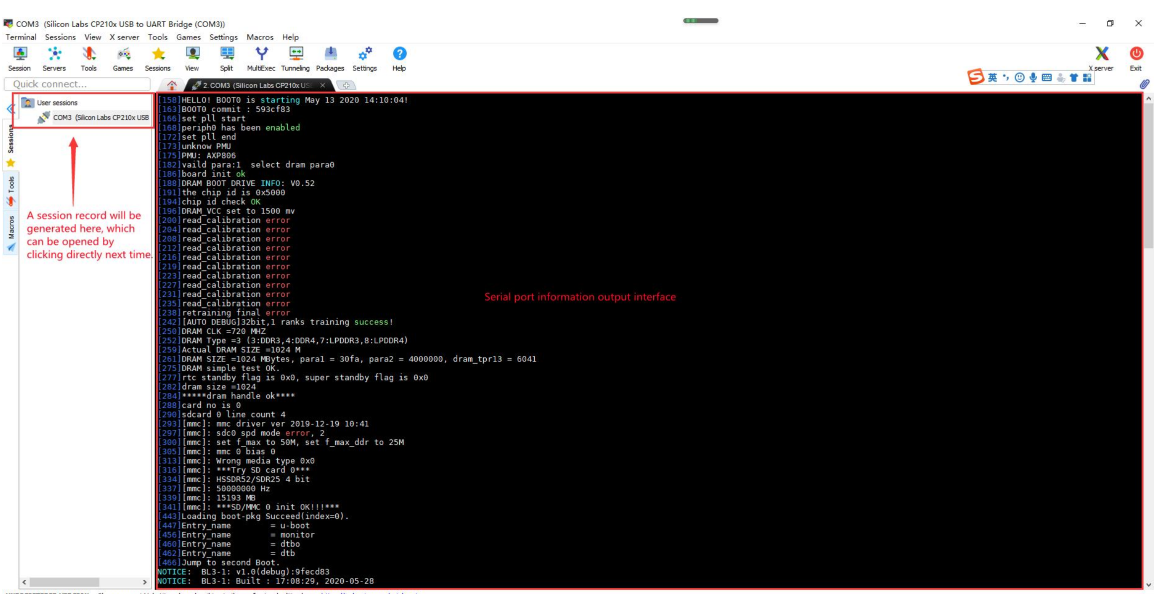

2) After downloading, use decompression software to decompress the downloaded compressed package, you can get the executable software of MobaXterm, and then double-click to open


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

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




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



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

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

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


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

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

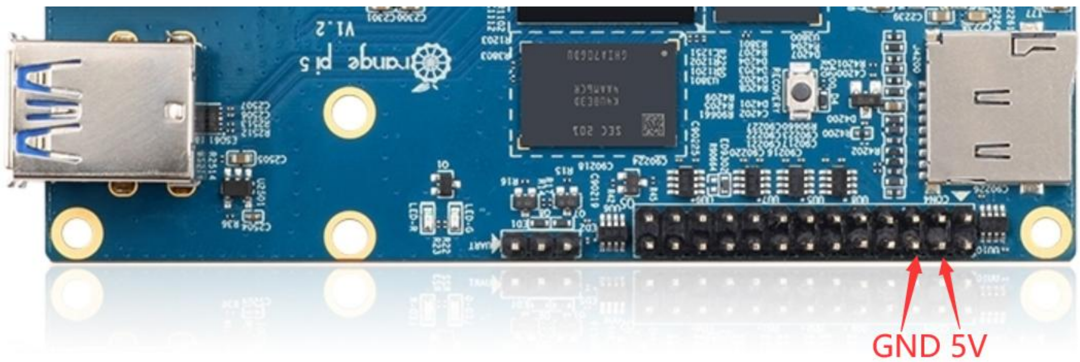
a. The USB A port of the power cord shown in the above picture needs to be plugged into the 5V/4A power adapter connector (**please do not plug into the**

**USB port of the computer for power supply**)

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

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

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




# 3. Linux system instructions

**Ubuntu images and Debian images are generally referred to as Linux images (they both use the Linux kernel), so when you see a Linux image or Linux system in the manual, it refers to a image or system like Ubuntu or Debian.**

**Many people will have doubts about whether they can use pure Ubuntu or pure Debian systems (pure here can be understood as systems downloaded from Ubuntu or Debian official websites). The answer is no, because Ubuntu and Debian do not provide an adapted system for the Orange Pi development board.**

**We can see from the official websites of Ubuntu and Debian that they both support the arm64 architecture (the SOC of the development board is the arm64 architecture), but please note that the support mentioned here refers only to the arm64 version of the software warehouse provided by Ubuntu or Debian (including Tens of thousands of software packages) or rootfs (these are the packages that Orange Pi uses when making Ubuntu or Debian systems). To make an Ubuntu or Debian system that can be used for a certain development board also needs to transplant U-boot and Linux kernel, etc., as well as repair the encountered bugs and optimize some functions, all of which are done by Orange Pi.**

**If Linux distributions such as CentOS, Kali, or OpenWRT are not ported by other developers or ported and adapted by themselves, they cannot be used on the development board of Orange Pi (hardware running these systems is no problem).**

**In addition, people often ask whether the system of other development boards can be used on the Orange Pi development board. The answer is no, because the chips and circuit connections used by different development boards are generally different. A system developed for a certain development board basically cannot be used on other development boards.**

## 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 |

## 3. 2. Linux system adaptation

| Features | Linux5.10 driver | Debian11 | Ubuntu20.04 | Ubuntu22.04 |
|---|---|---|---|---|
| USB2.0x2 | OK | OK | OK | OK |
| USB3.0x1 | OK | OK | OK | OK |
| USB Type-C 3.0 | OK | OK | OK | OK |
| DP display | OK | OK | OK | OK |
| M.2 NVMe SSD boot | OK | OK | OK | OK |
| M.2 SATA SSD boot | OK | OK | OK | OK |
| AP6275P-WIFI | OK | OK | OK | OK |
| AP6275P-Bluetooth | OK | OK | OK | OK |
| GPIO（26pin） | OK | OK | OK | OK |
| UART（26pin） | OK | OK | OK | OK |
| SPI（26pin） | OK | OK | OK | OK |
| I2C（26pin） | OK | OK | OK | OK |
| CAN（26pin） | OK | OK | OK | OK |
| PWM（26pin） | OK | OK | OK | OK |

| | | | | |
|---|---|---|---|---|
| **3pin debugging serial port** | **OK** | **OK** | **OK** | **OK** |
| **TF card start** | **OK** | **OK** | **OK** | **OK** |
| **HDMI video** | **OK** | **OK** | **OK** | **OK** |
| **HDMI audio** | **OK** | **OK** | **OK** | **OK** |
| **OV13850 camera** | **OK** | **OK** | **OK** | **NO** |
| **OV13855 camera** | **OK** | **OK** | **OK** | **NO** |
| **LCD1** | **OK** | **OK** | **OK** | **OK** |
| **LCD2** | **OK** | **OK** | **OK** | **OK** |
| **Gigabit Ethernet port** | **OK** | **OK** | **OK** | **OK** |
| **Network port status light** | **OK** | **OK** | **OK** | **OK** |
| **MIC** | **OK** | **OK** | **OK** | **OK** |
| **headphone playback** | **OK** | **OK** | **OK** | **OK** |
| **headphone recording** | **OK** | **OK** | **OK** | **OK** |
| **LED** | **OK** | **OK** | **OK** | **OK** |
| **GPU** | **OK** | **OK** | **OK** | **OK** |
| **NPU** | **OK** | **OK** | **OK** | **OK** |
| **VPU** | **OK** | **OK** | **OK** | **OK** |
| **switch button** | **OK** | **OK** | **OK** | **OK** |
| **watchdog test** | **OK** | **OK** | **OK** | **OK** |
| **Chromium hard solution video** | **OK** | **OK** | **OK** | **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 framed in the following box

```

```

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

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

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

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

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

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

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

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

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

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

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

```
console=serial
```

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

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

## 3. 4. Linux system login instructions

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

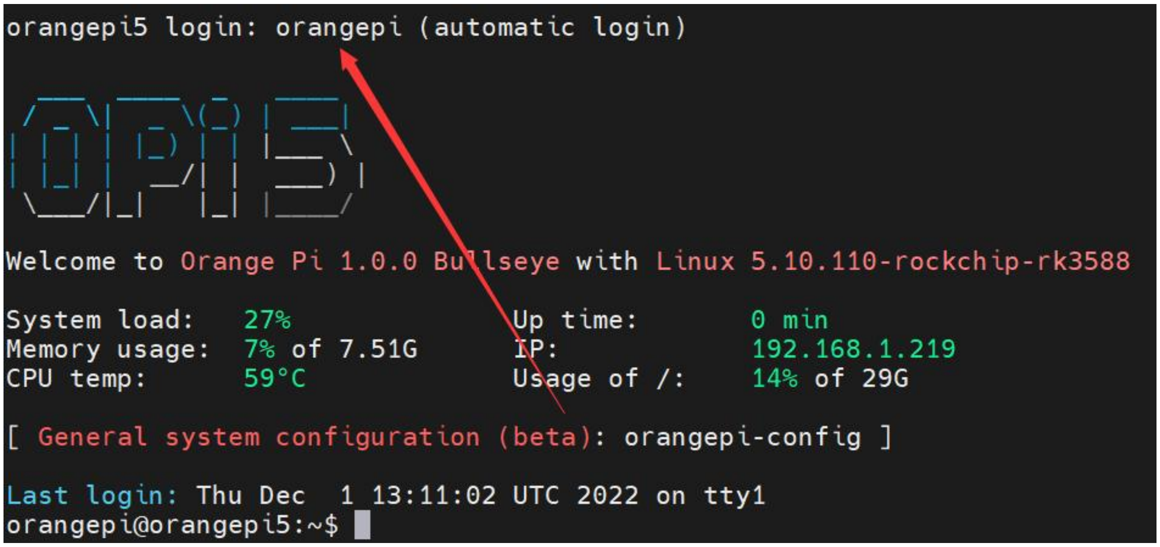
| account | password |
| --- | --- |
| root | orangepi |
| orangepi | orangepi |

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




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

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

3) Use the following command to disable automatic login terminal

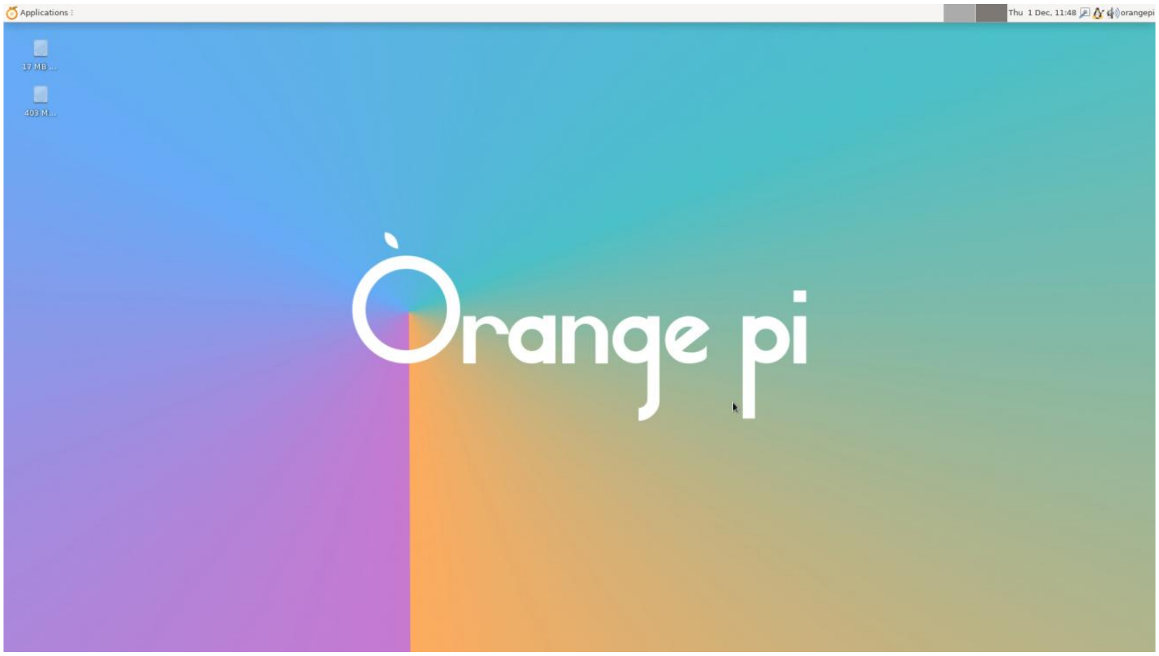
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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

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

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

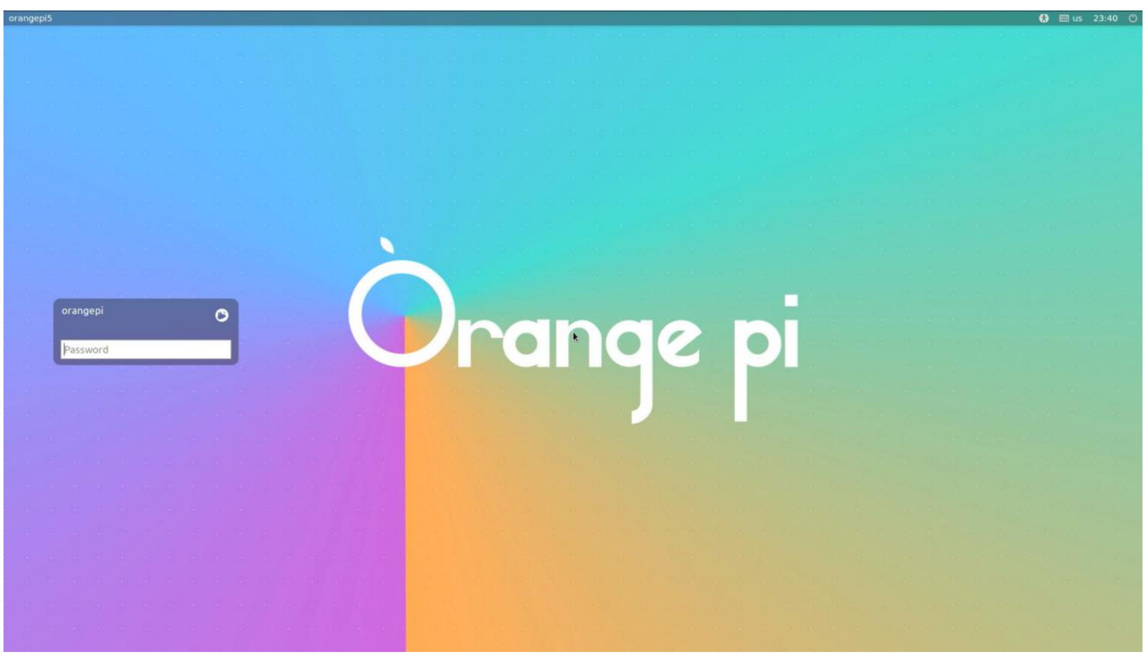
1) After the desktop version system is started, it will automatically log in to the desktop without entering a password




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

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

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



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

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

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

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


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

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

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

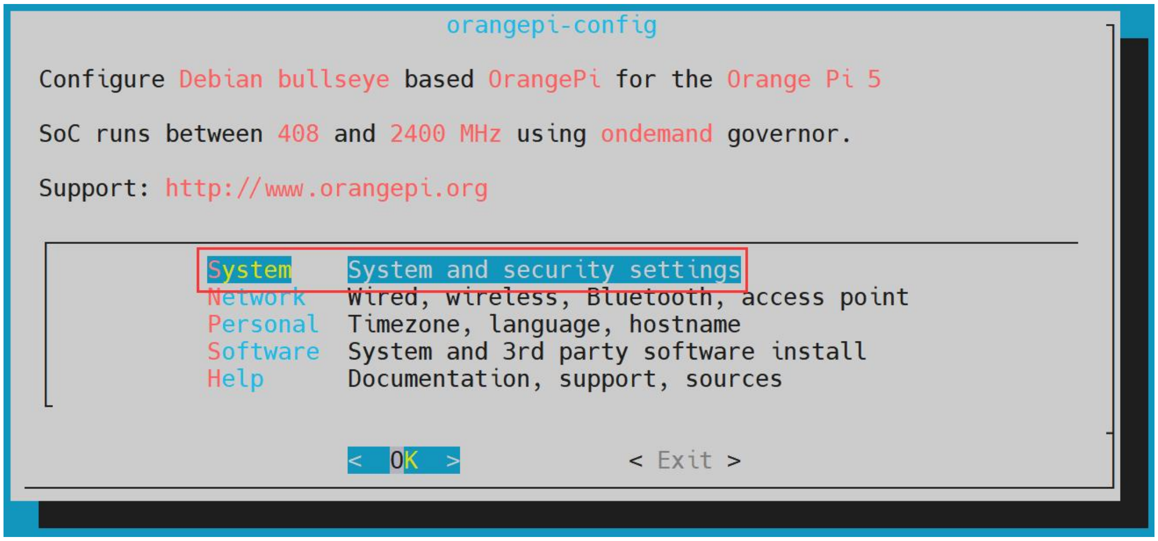
```
orangepi@orangepi:~$ sudo desktop_login.sh orangepi
```

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

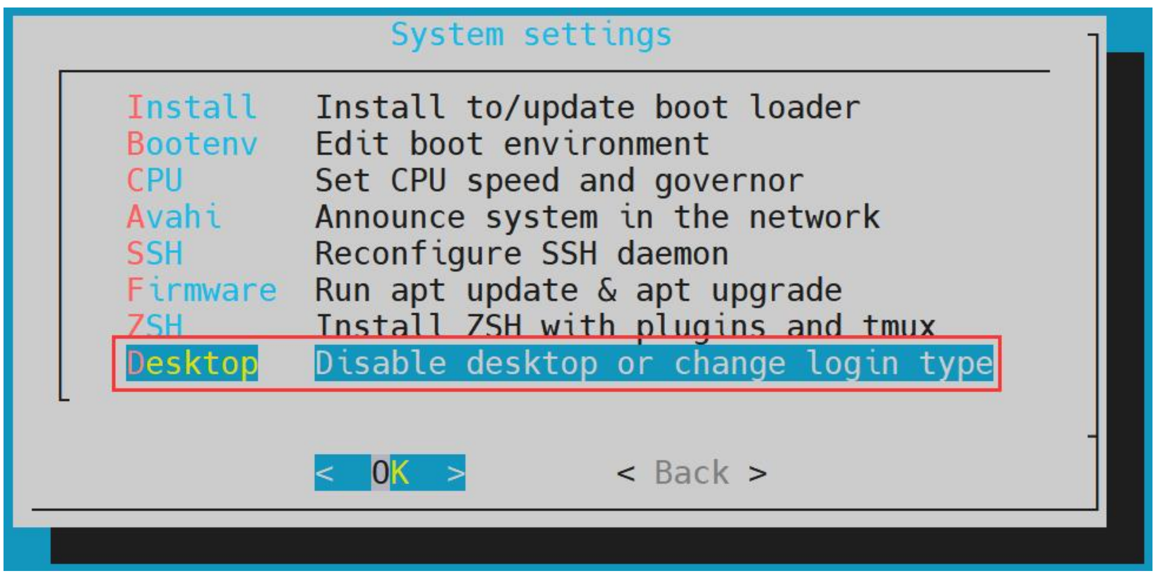
1) First enter the following command in the command line, **please remember to add sudo permission**

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

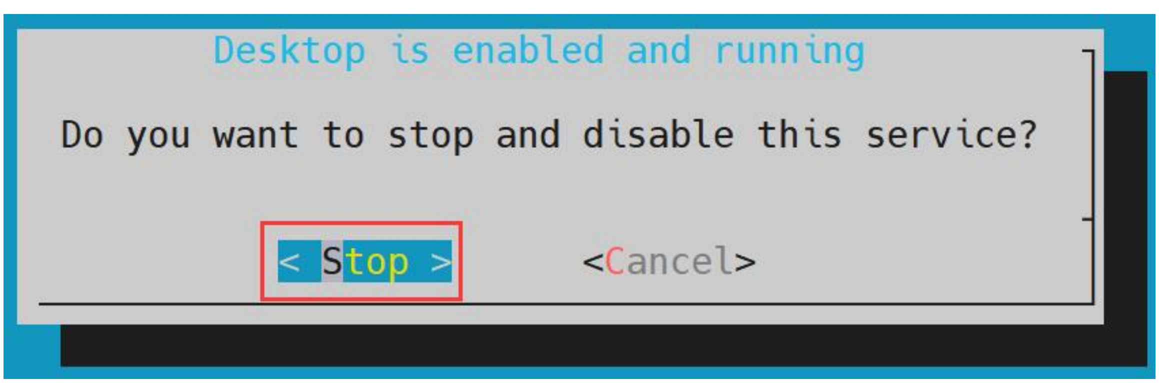
2) Then select **System**




3) Then select **Desktop**




4) Then select **<Stop>**



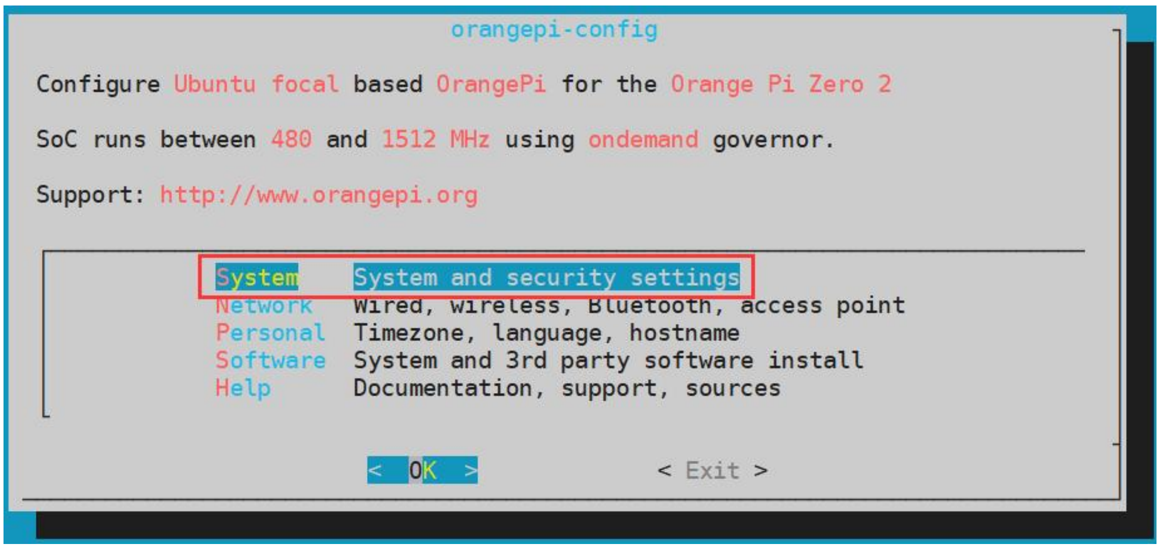
5) **Then restart the Linux system and you will find that the desktop will not be displayed**

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

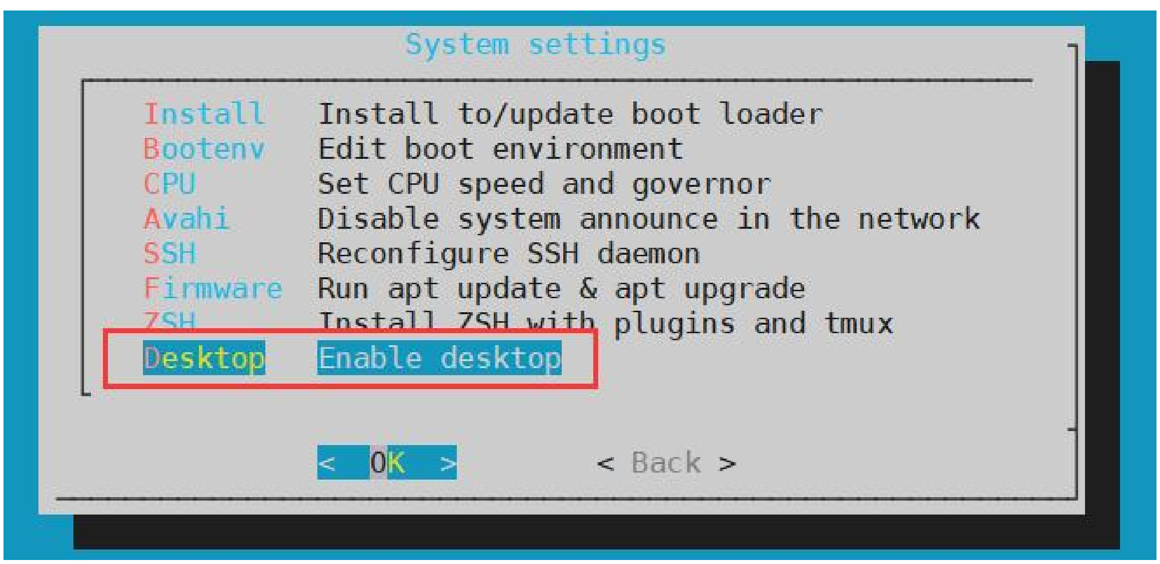
a. First enter the following command on the command line, **please remember to add sudo permission**

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

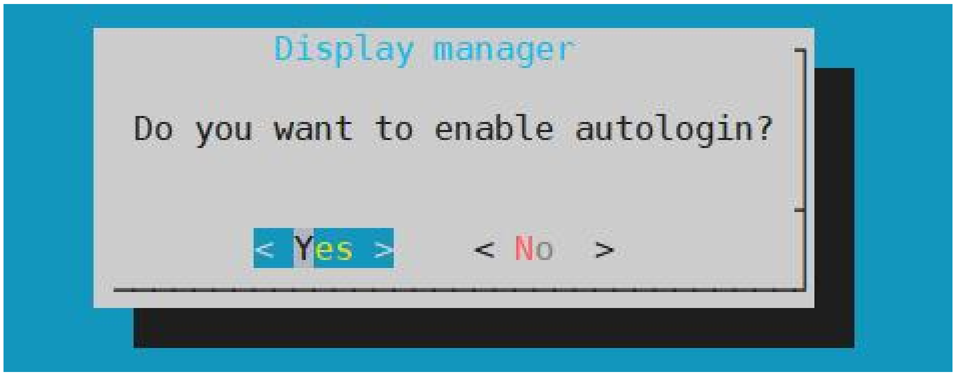
b. Then select **System**




c. Then select **Desktop Enable desktop**



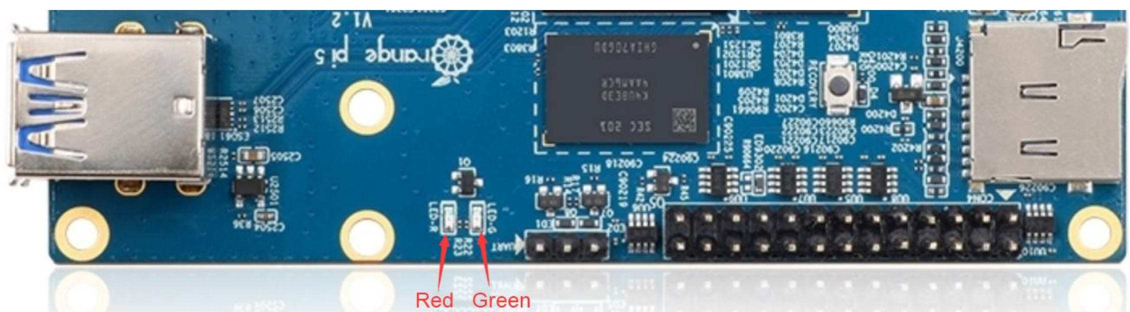

d. Then choose whether to automatically log in to the desktop, if you choose **<Yes>**, you will automatically log in to the desktop, if you choose **<No>**, the input interface for user and password will be displayed, and you need to enter the password to enter the desktop




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

## 3. 5. Onboard LED Light Test Instructions

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




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

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

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

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

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 1 > brightness
```

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

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

## 3. 6. Network connection test

### 3. 6. 1. Ethernet port test

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

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

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

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

**When using ifconfig to 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 monitor, then log in to the system and use the ip addr show eth0 command to view the IP address**

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

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

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

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

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

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

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

4) The command to test the network connectivity is as follows, the ping command can be

interrupted through the shortcut key of Ctrl+C

```
orangepi@orangepi:~$ ping www.baidu.com -I eth0
PING www.a.shifen.com (14.215.177.38) from 192.168.1.12 eth0: 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=56 time=6.74 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=56 time=6.80 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=56 time=6.26 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=56 time=7.27 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 6.260/6.770/7.275/0.373 ms
```

### 3. 6. 2. WIFI connection test

**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
   a. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the Linux system
   b. If the development board is connected to the HDMI display, you can log in to the linux system through the terminal displayed on the HDMI

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

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

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

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

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

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

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

3) 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
```

4) 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

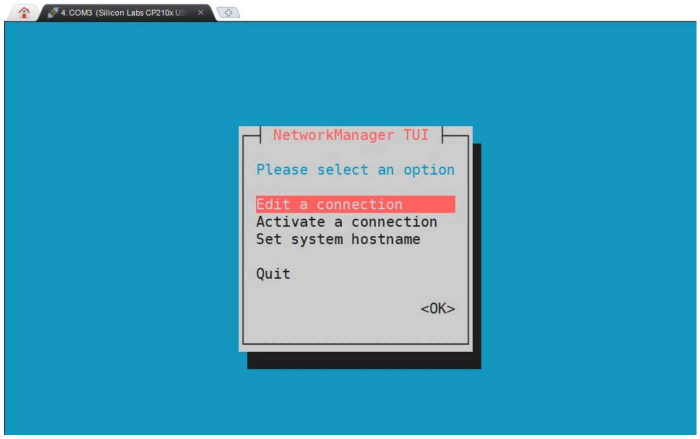
b. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the linux system (please use MobaXterm for the serial port software, and minicom cannot display the graphical interface)

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

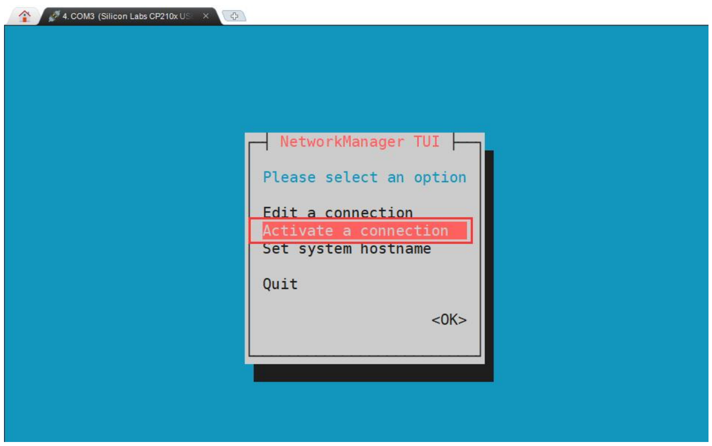
2) Then enter the nmtui command in the command line to open the wifi connection interface

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

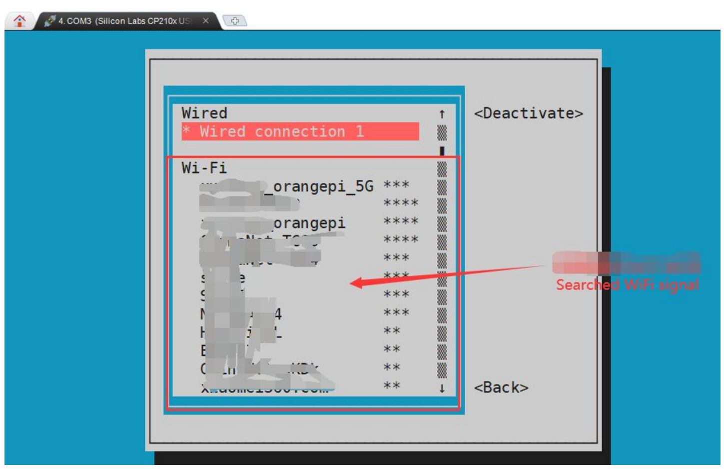
3) Enter the nmtui command to open the interface as shown below



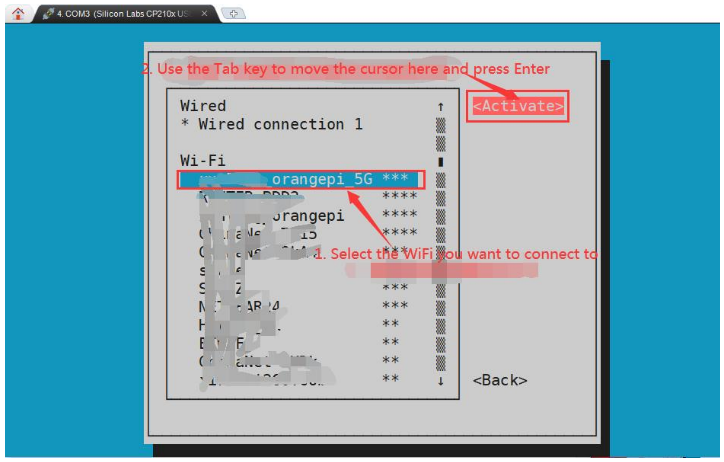

4) Select Activate a connect and press Enter



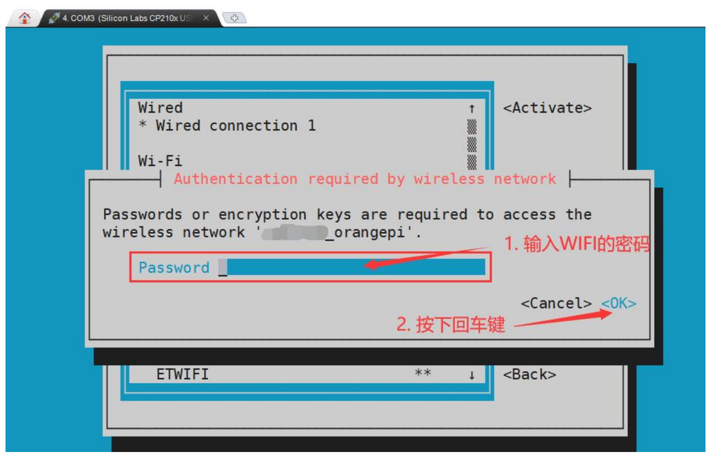

5) Then you can see all the searched WIFI hotspots



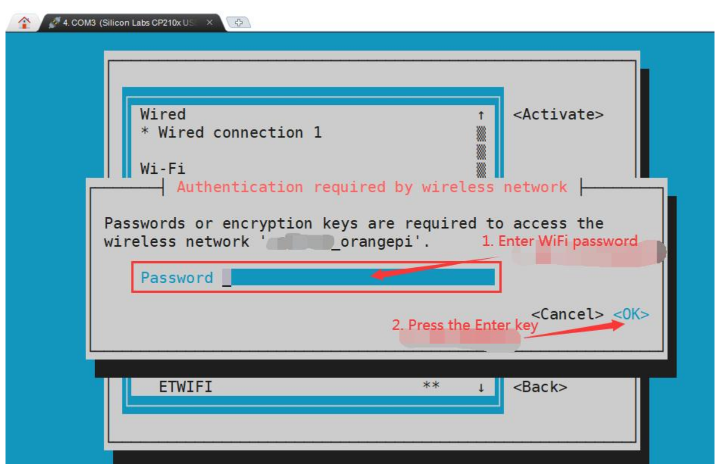

6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the cursor on Activate and press Enter




7) Then a dialog box for entering 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

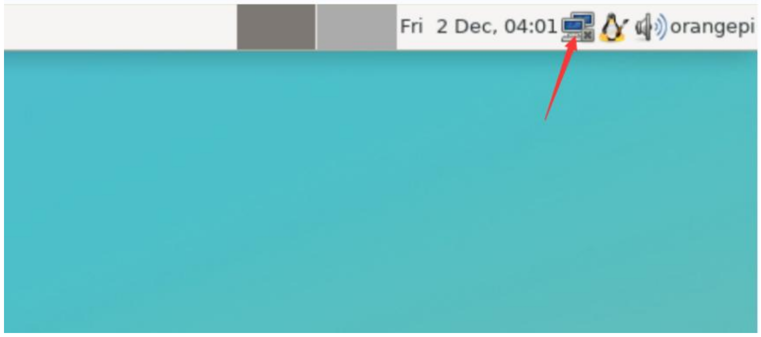
```
orangepi@orangepi:~$ ip addr show wlan0
11: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 24:8c:d3:aa:76:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
       valid_lft 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

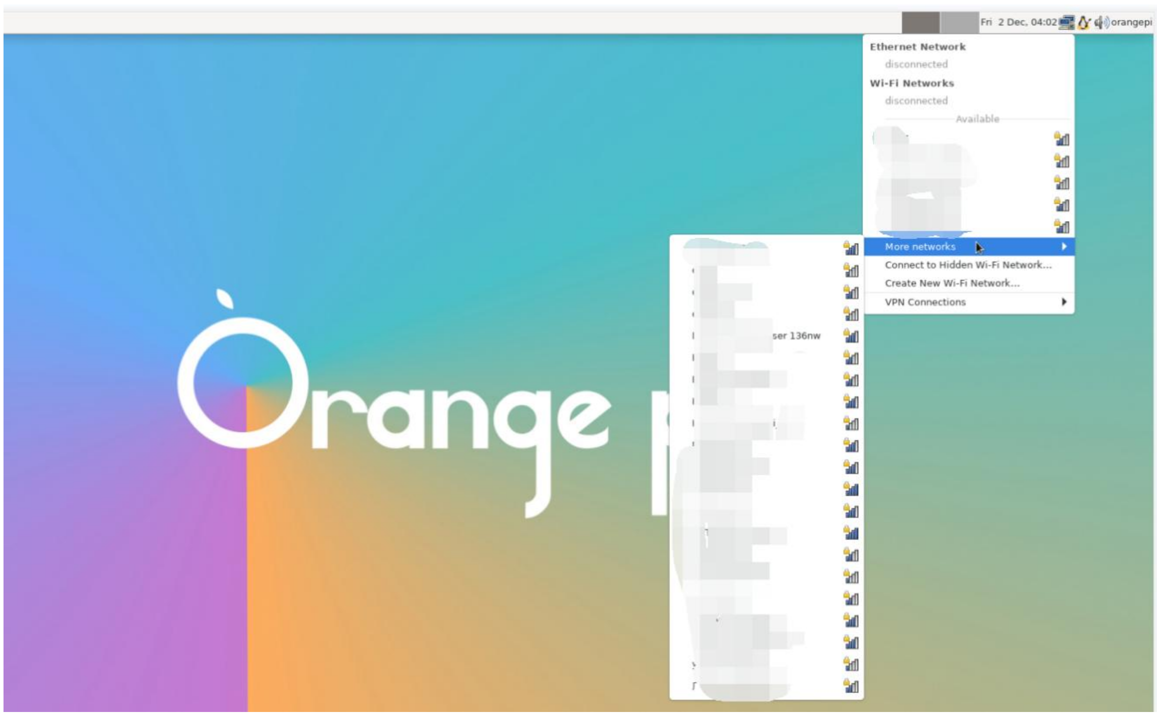
```
orangepi@orangepi:~$ ping www.orangepi.org -I wlan0
PING www.orangepi.org (182.92.236.130) from 192.168.1.49 wlan0: 56(84) bytes of data.
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=1 ttl=52 time=43.5 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=2 ttl=52 time=41.3 ms
64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

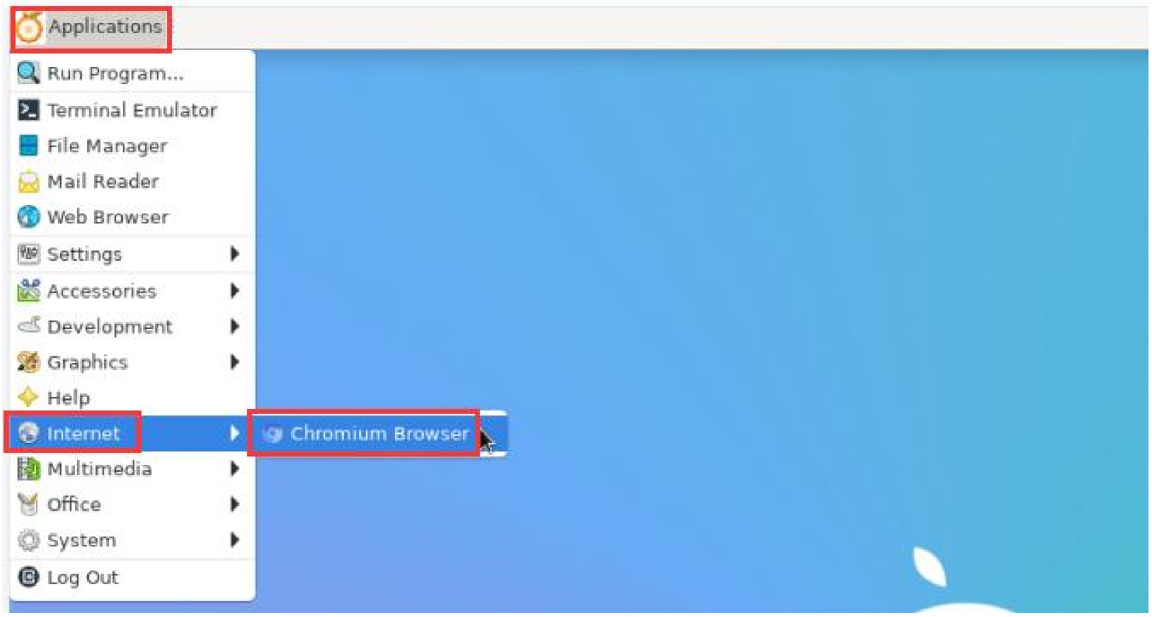



2) Click More networks in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to

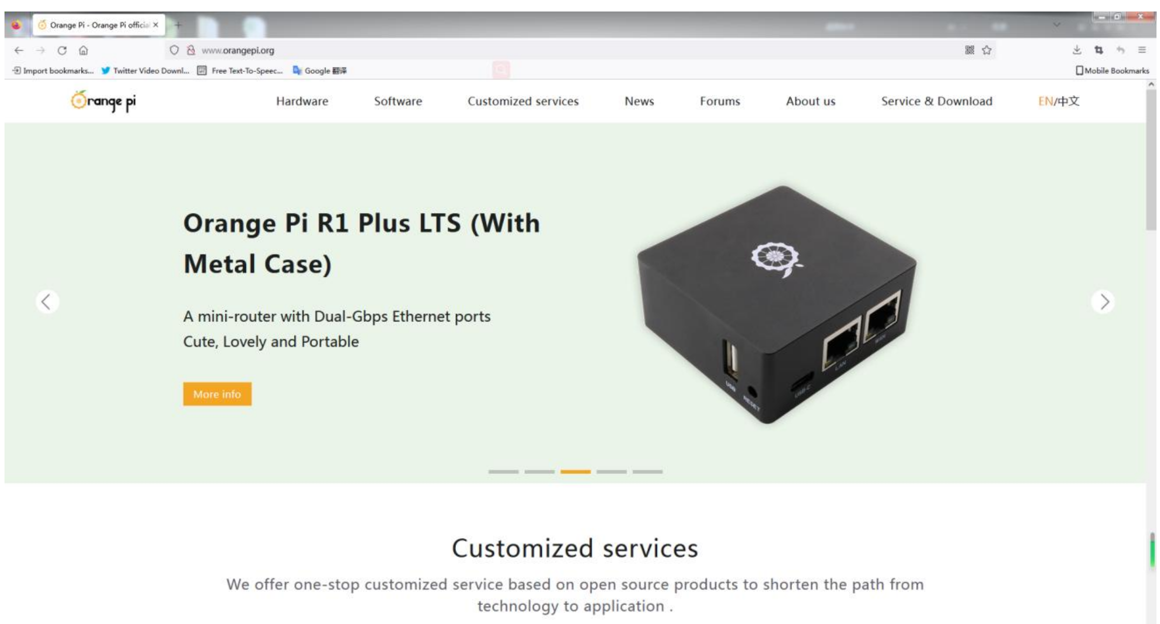



3) Then enter the password of the WIFI hotspot, and then click Connect to start connecting to WIFI

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

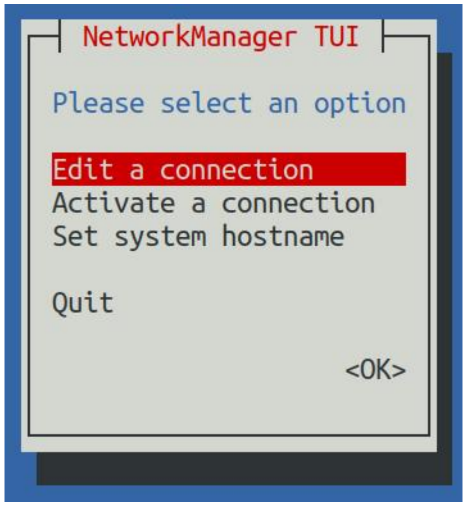
**Please do not set a static IP address by modifying the /etc/network/interfaces configuration file。**

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

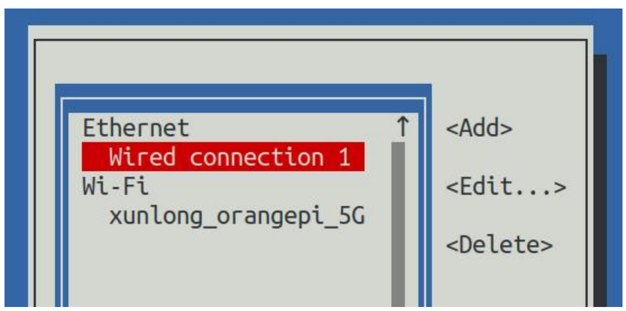
1) First run the nmtui command

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

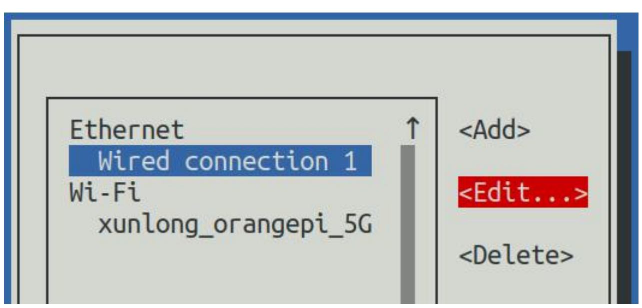
2) Then select Edit a connection and press Enter



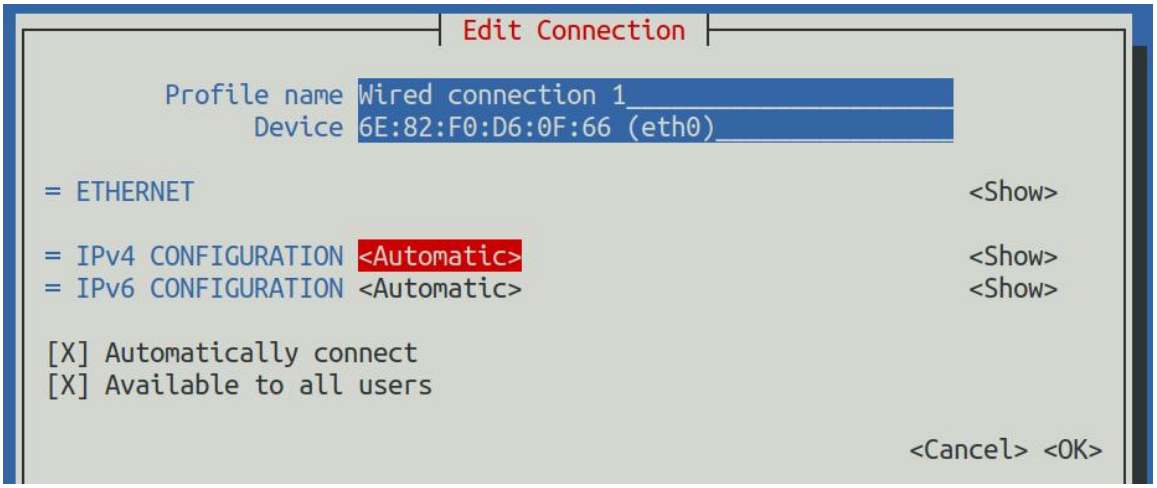

3) Then select the network interface that needs to set a static IP address, for example, to set the static IP address of the **Ethernet** interface, select **Wired connection 1**



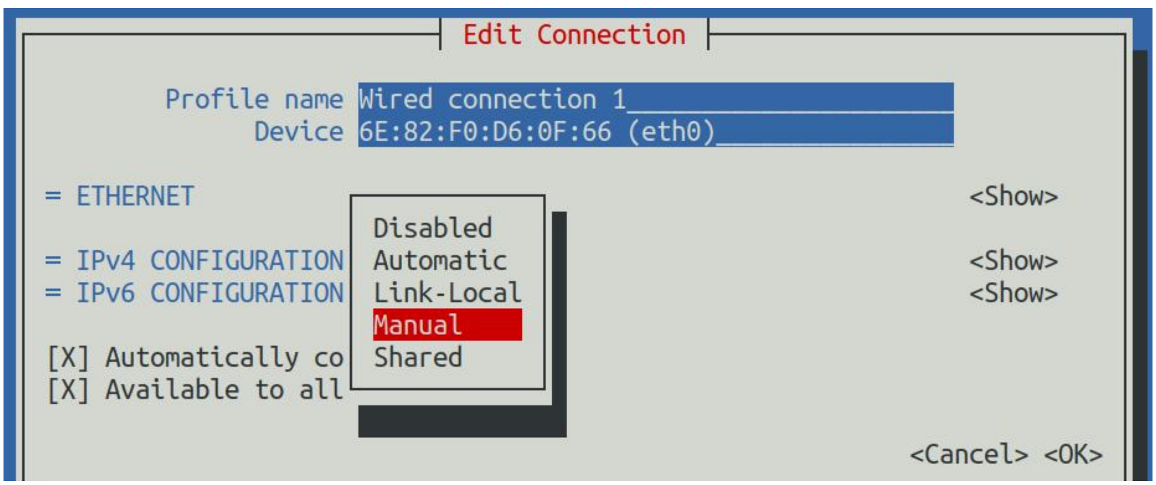

4) Then select **Edit** via the **Tab** key and press the Enter key



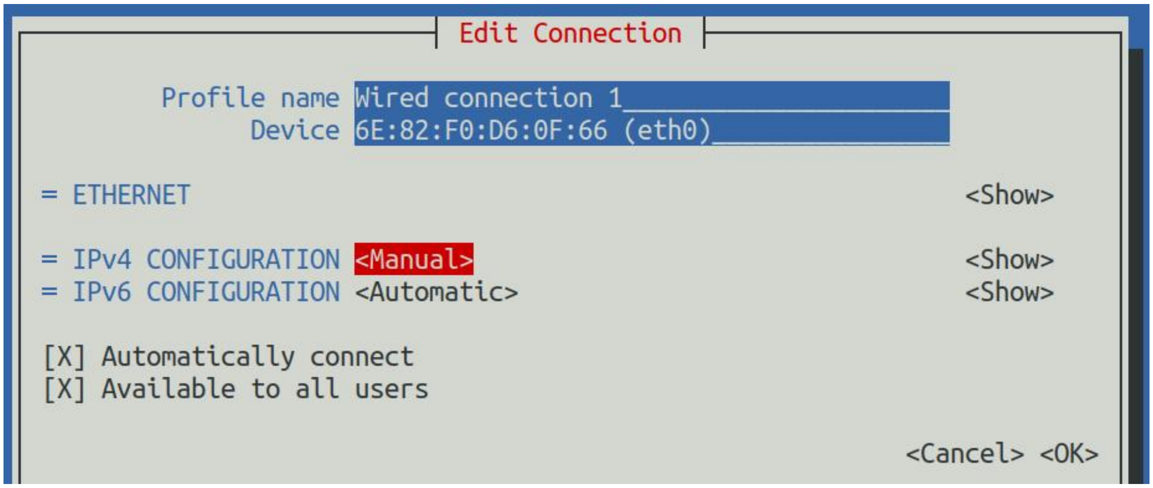
5) Then use the Tab key to move the cursor to the <Automatic> position shown in the figure below to configure IPv4



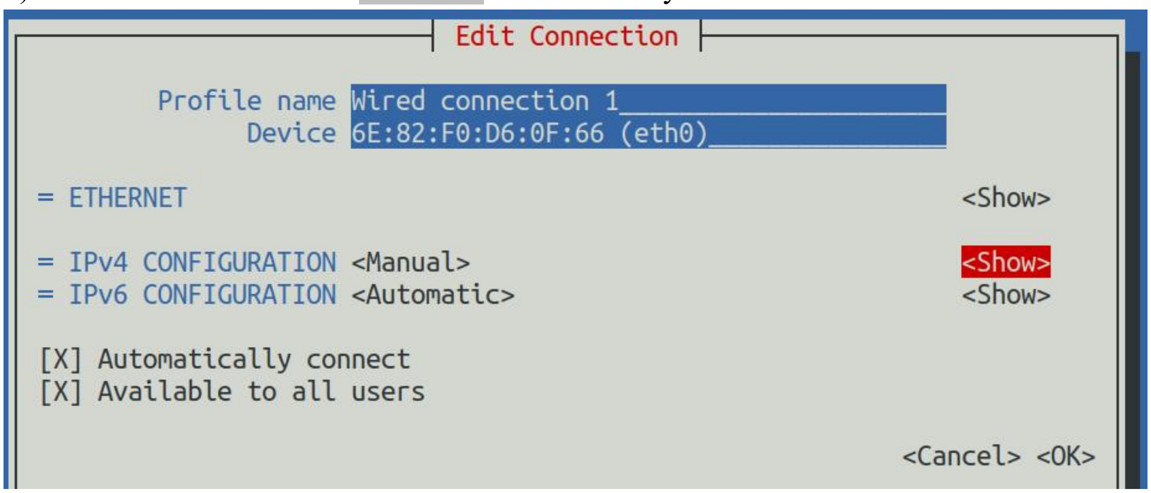

6) Then press Enter, select **Manual** through the up and down arrow keys, and press Enter to confirm



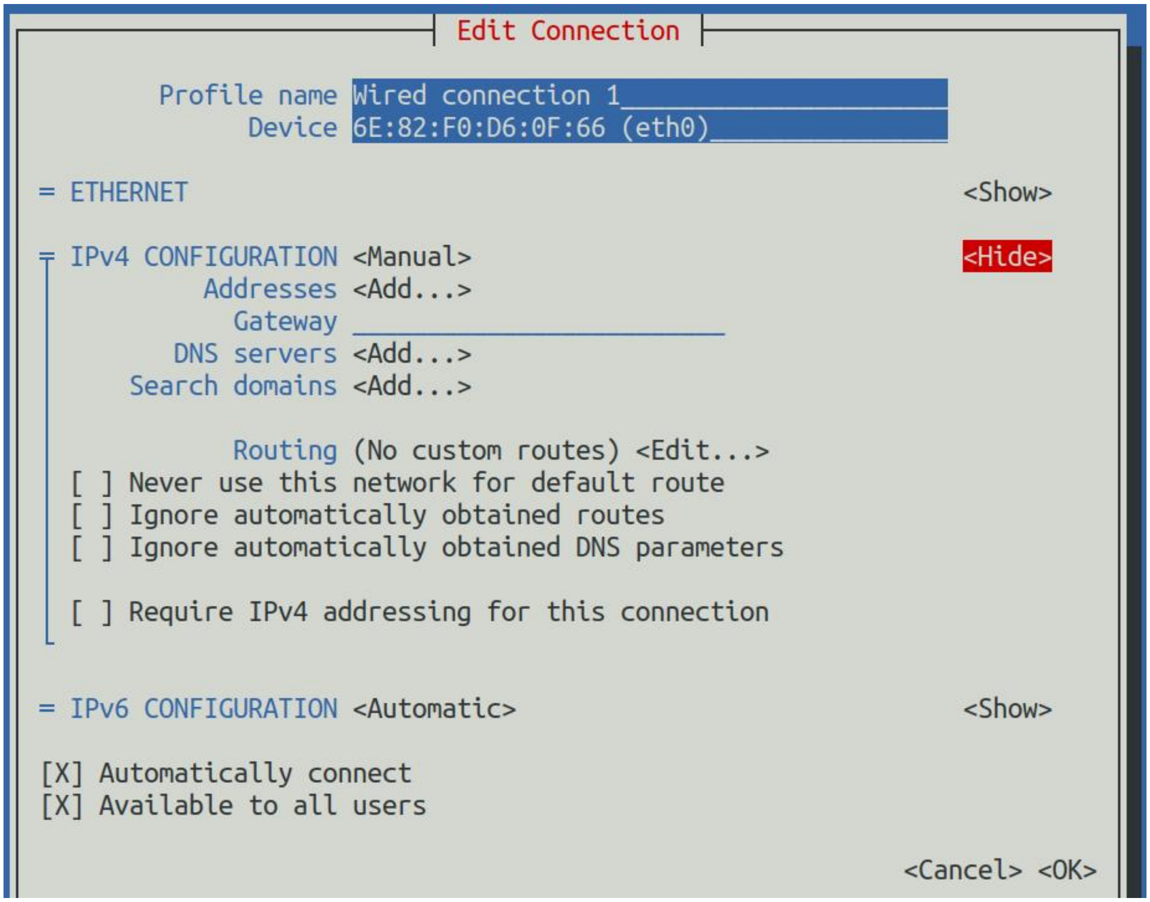

7) The display after selection is shown in the figure below



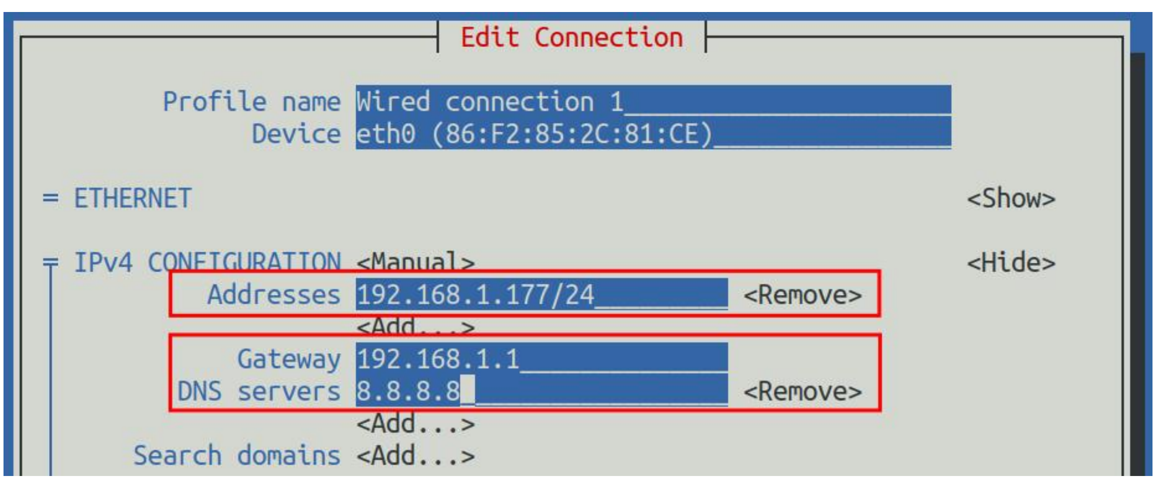

8) Then move the cursor to <Show> via the Tab key




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

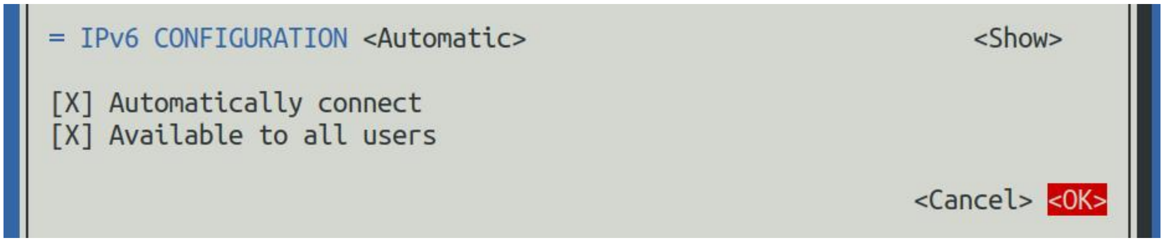



10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the position shown in the figure below (there are many other setting options in it, please explore by yourself)，**Please set it according to your specific needs, the value set in the figure below is just an example**

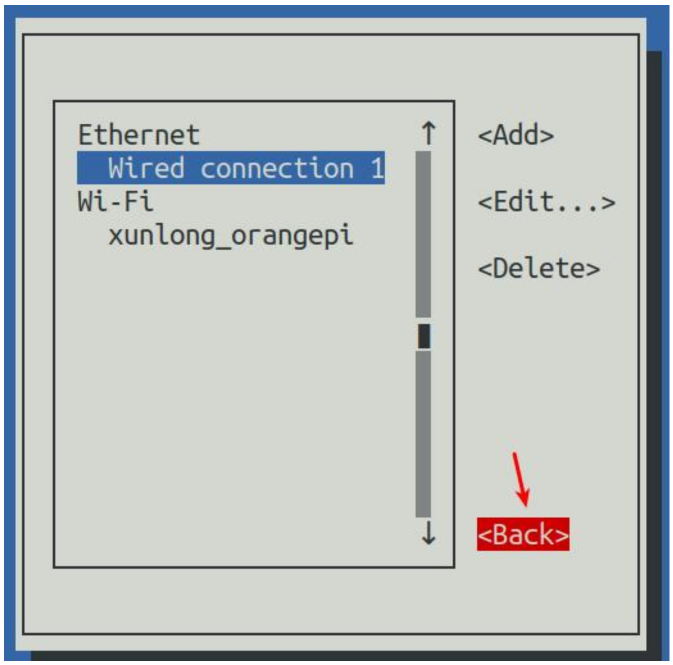



11) After setting, move the cursor to **<OK>** in the lower right corner, and press Enter to

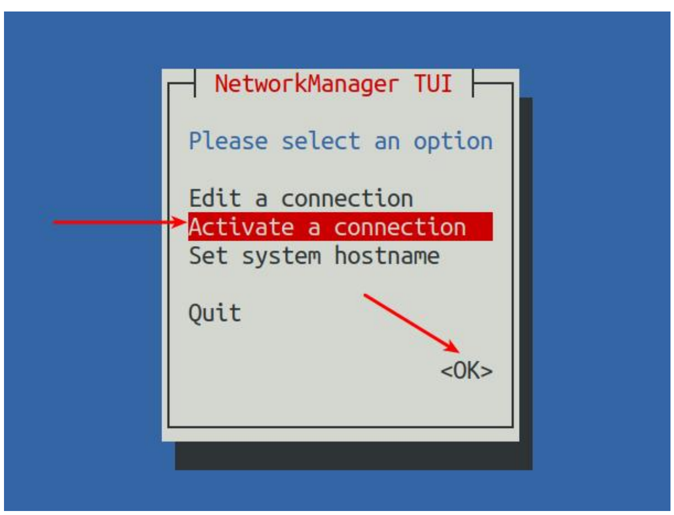
confirm



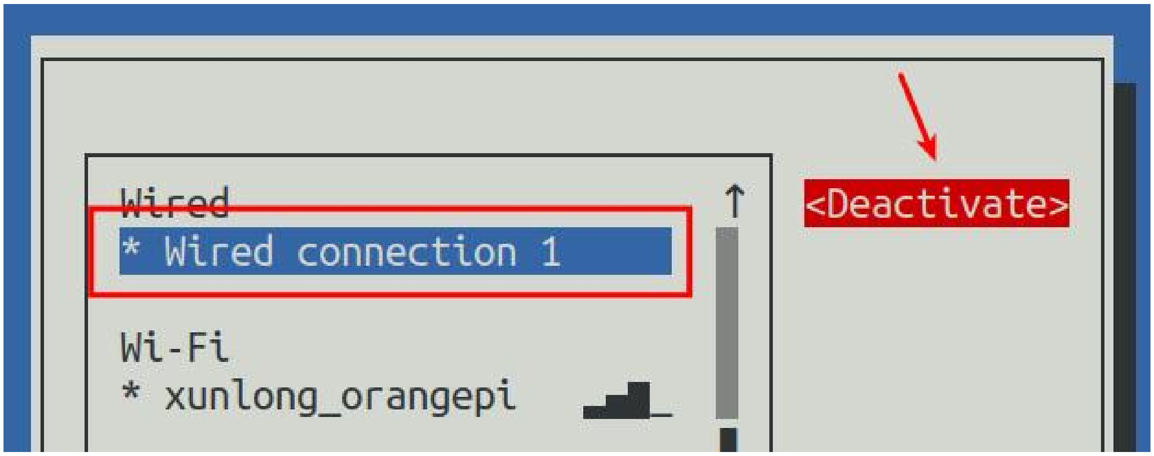

12) Then click **<Back>** to return to the previous selection interface



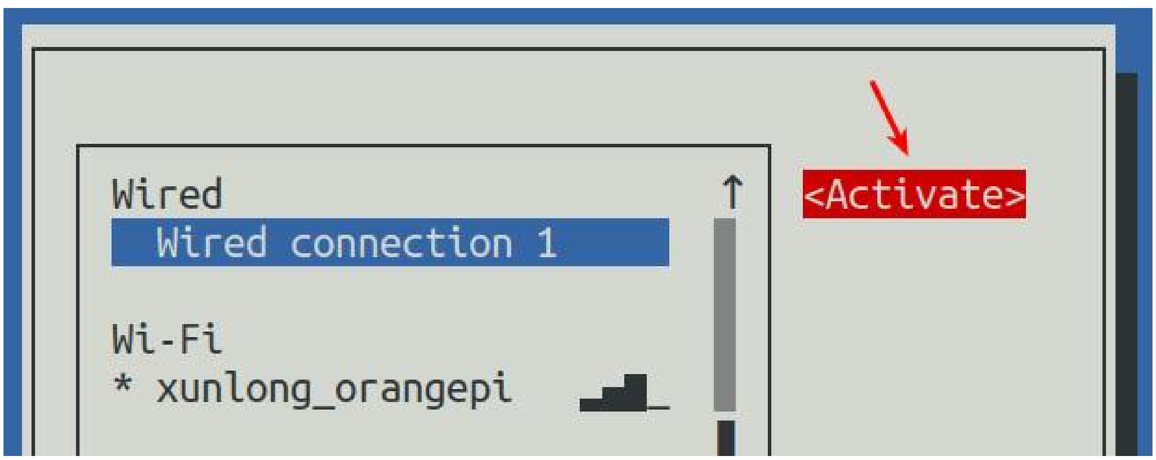

13) Then select **Activate a connection**, then move the cursor to **<OK>**, and finally click Enter



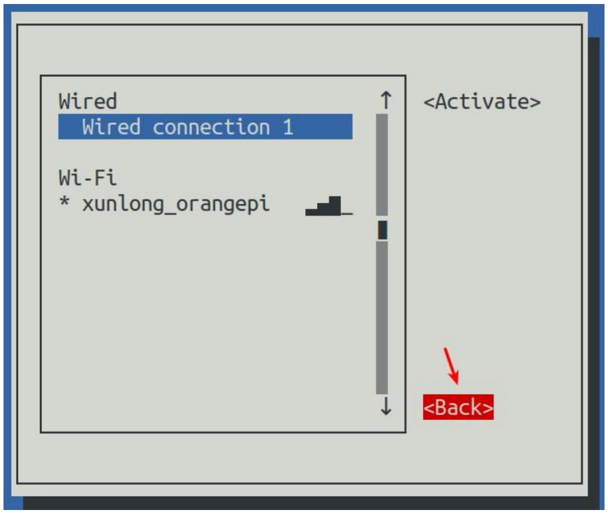
14) Then select the network interface that needs to be set, such as **Wired connection 1**, then move the cursor to **<Deactivate>**, and press Enter to disable **Wired connection 1**

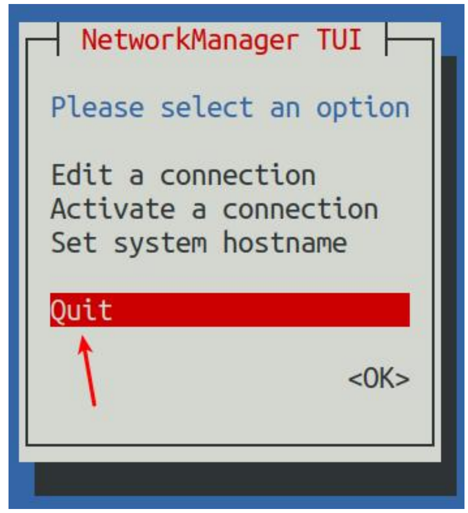



15) Then please do not move the cursor, and then press the Enter key to re-enable **Wired connection 1**, so that the static IP address set earlier will take effect




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

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

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

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

3) Then enter the following command, where

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

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

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

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

4) Then restart the linux system

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

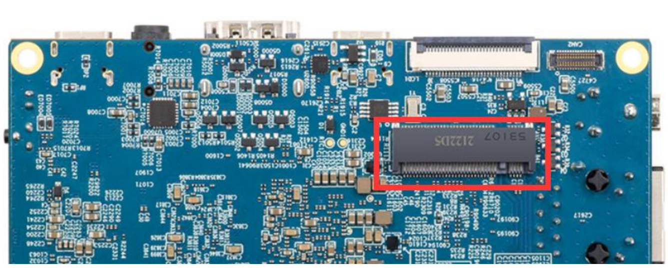
5) Then re-enter the linux system and use the **ip addr show eth0** command to see that the IP address has been set to the desired value

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

### 3. 6. 4. How to use AP6275P PCIe network card

1) First, you need to purchase an AP6275P PCIe network card as shown in the figure below


2) Then insert the AP6275P PCIe network card into the M.2 interface of the development board and fix it



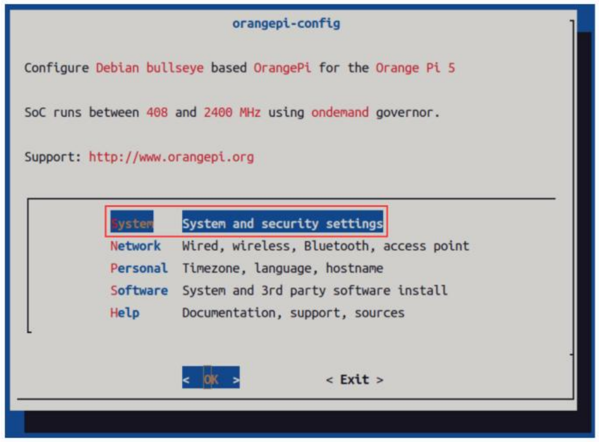
3) Then open the configuration of the AP6275P PCIe network card in the linux system,

the steps are as follows：

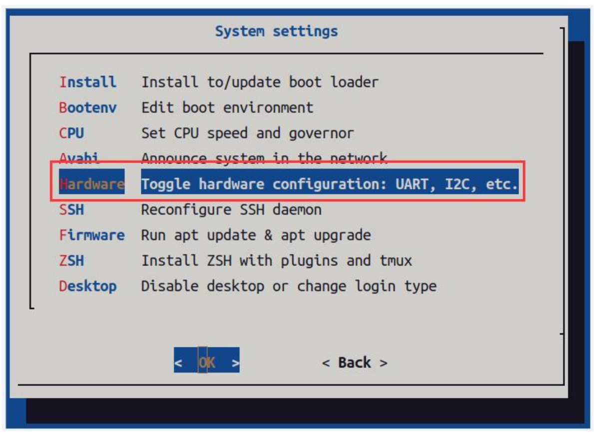
a. First run **orangepi-config**, normal users remember to add sudo permission

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

b. Then select **System**




c. Then select **Hardware**

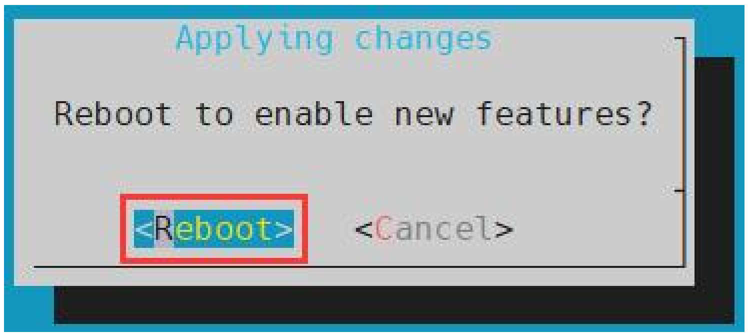



d. Then use the arrow keys on the keyboard to navigate to **wifi-ap6275p**, and then use the **space** to select


e. Then select <Save> to save

f. Then select **<Back>**


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




**The above settings will eventually add the configuration of overlays=wifi-ap6275p to /boot/orangepiEnv.txt. After setting, you can check it first. If this configuration does not exist, then there is a problem with the settings.**

**If you find it troublesome to use orangepi-config, you can also open /boot/orangepiEnv.txt, and then add the configuration of overlays=wifi-ap6275p.。**

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "ap6275p"**
**overlays=wifi-ap6275p**

4) If everything is normal after restarting the system, use the following command to see the device nodes of WIFI and Bluetooth

a. The command to view the WIFI device node is as follows：

orangepi@orangepi:~$ **ip addr show wlan0**
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state DORMANT group default qlen 1000
link/ether 70:f7:54:b8:b3:17 brd ff:ff:ff:ff:ff:ff

b. The command to view the Bluetooth device node is as follows：

orangepi@orangepi:~$ **hciconfig -a**
hci0: Type: Primary Bus: UART
BD Address: 82:CC:AE:62:CE:3E ACL MTU: 1021:8 SCO MTU: 64:1
UP RUNNING

```
RX bytes:958 acl:0 sco:0 events:73 errors:0
TX bytes:5544 acl:0 sco:0 commands:73 errors:0
Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
Link policy: RSWITCH SNIFF
Link mode: SLAVE ACCEPT
Name: 'orangepi5'
Class: 0x1c0000
Service Classes: Rendering, Capturing, Object Transfer
Device Class: Miscellaneous,
HCI Version: 5.1 (0xa)    Revision: 0x3f9
LMP Version: 5.1 (0xa)    Subversion: 0x1111
Manufacturer: Broadcom Corporation (15)
```

5) For the wifi connection and test method, **please refer to the section of WIFI connection test**, which will not be repeated here

6) For the test method of Bluetooth, **please refer to the section on Bluetooth usage**, so I won't go into details here

### 3. 6. 5. AP6275P PCIe NIC creates WIFI hotspot via create_ap

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

**https://github.com/oblique/create_ap**

**If you are using the latest image, the create_ap script has been pre-installed, and you can create a WIFI hotspot through the create_ap command. The basic command format of create_ap is as follows:**

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

*** options: You can use this parameter to specify the encryption method, the frequency band of the WIFI hotspot, the bandwidth mode, the network sharing**

**method, etc. You can get the options through create_ap -h**
*** wifi-interface：The name of the wireless network card**
*** interface-with-internet：The name of the network card that can be connected to the Internet, generally eth0**
*** access-point-name：hotspot name**
*** passphrase：hotspot password**

#### 3.6.5.1. create_ap method to create WIFI hotspot in NAT mode

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

```
orangepi@orangepi5:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi
```

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

```
orangepi@orangepi5:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi
Config dir: /tmp/create_ap.wlan0.conf.fPItFUJ2
PID: 3831
Network Manager found, set ap0 as unmanaged device... DONE
Creating a virtual WiFi interface... ap0 created.
Sharing Internet using method: nat
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.fPItFUJ2/hostapd_ctrl
ap0: interface state UNINITIALIZED->ENABLED
ap0: AP-ENABLED
```

3）Take out the mobile phone at this time, in the searched WIFI list, you can find the WIFI hotspot named **orangepi** created by the development board, and then click **orangepi** to connect to the hotspot, the password is the **orangepi** set above


4）After the connection is successful, the display is as shown in the figure below


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

```
orangepi@orangepi5:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.150  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::938f:8776:5783:afa2  prefixlen 64  scopeid 0x20<link>
        ether 4a:a0:c8:25:42:82  txqueuelen 1000  (Ethernet)
        RX packets 25370  bytes 2709590 (2.7 MB)
```

```
        RX errors 0    dropped 50    overruns 0    frame 0
        TX packets 3798    bytes 1519493 (1.5 MB)
        TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
        device interrupt 83
```

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


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

```
orangepi@orangepi5:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi -g 192.168.2.1
```

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



7）If the **--freq-band** parameter is not specified, the hotspot created by default is in the 2.4G frequency band. If you want to create a hotspot in the 5G frequency band, you can specify it through the **--freq-band 5** parameter. The specific command is as follows

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --freq-band 5
```

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

```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --hidden
```

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


### 3. 6. 5. 2. create_ap method to create WIFI hotspot in bridge mode

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi
```

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi
[sudo] password for orangepi:
Config dir: /tmp/create_ap.wlan0.conf.fg9U5Xgt
PID: 3141
Network Manager found, set ap0 as unmanaged device... DONE
Creating a virtual WiFi interface... ap0 created.
Sharing Internet using method: bridge
Create a bridge interface... br0 created.
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.fg9U5Xgt/hostapd_ctrl
ap0: interface state UNINITIALIZED->ENABLED
ap0: AP-ENABLED
```

3）Take out the mobile phone at this time, and you can find the WIFI hotspot named orangepi created by the development board in the searched WIFI list, and then you can click orangepi to connect to the hotspot, and the password is the orangepi set above


4）After the connection is successful, the display is as shown in the figure below


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

```
orangepi@orangepi:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.150  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::938f:8776:5783:afa2  prefixlen 64  scopeid 0x20<link>
```

```
        ether 4a:a0:c8:25:42:82  txqueuelen 1000  (Ethernet)
        RX packets 25370  bytes 2709590 (2.7 MB)
        RX errors 0  dropped 50  overruns 0  frame 0
        TX packets 3798  bytes 1519493 (1.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 83
```

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


6）If the --freq-band parameter is not specified, the hotspot created by default is in the 2.4G frequency band. If you want to create a hotspot in the 5G frequency band, you can specify the --freq-band 5 parameter. The specific command is as follows

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --freq-band 5
```

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

```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --hidden
```

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


## 3. 7. SSH remote login development board

**Linux systems enable ssh remote login by default and allow the root user to log in to the system. Before ssh login, 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

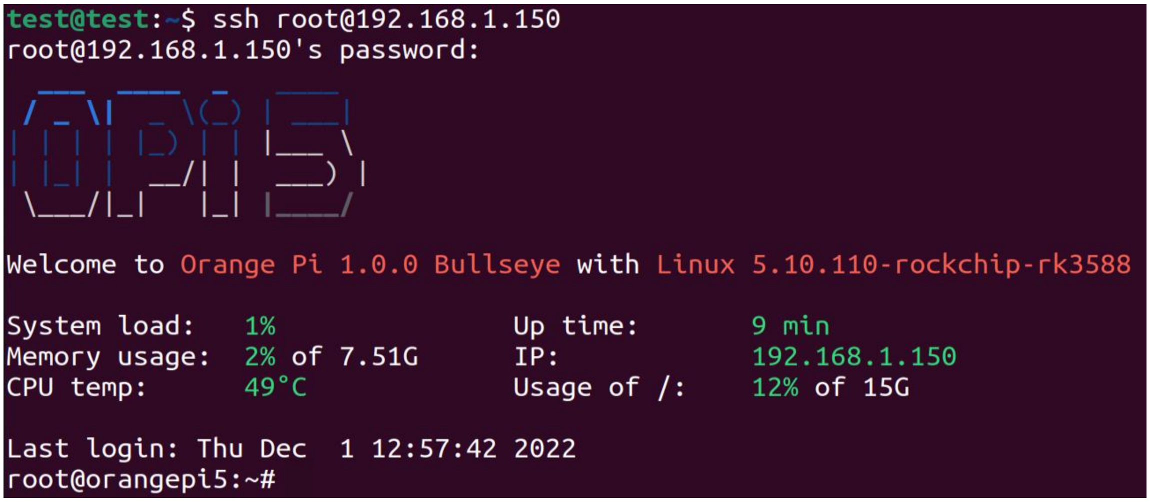
```
test@test:~$ ssh root@192.168.1.xxx            (Need to be replaced with the IP address of the development board)
root@192.168.1.xx's password:            （Enter the password here, the default password is orangepi）
```

**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 don't wonder if 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

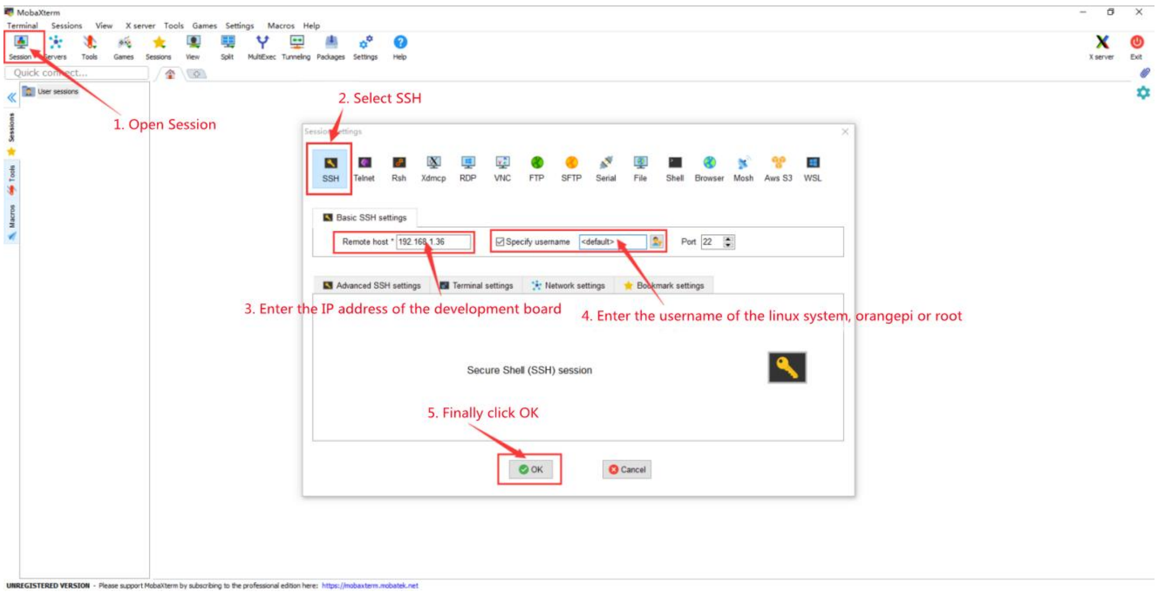
a. Open Session

b. Then select SSH in Session Setting

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

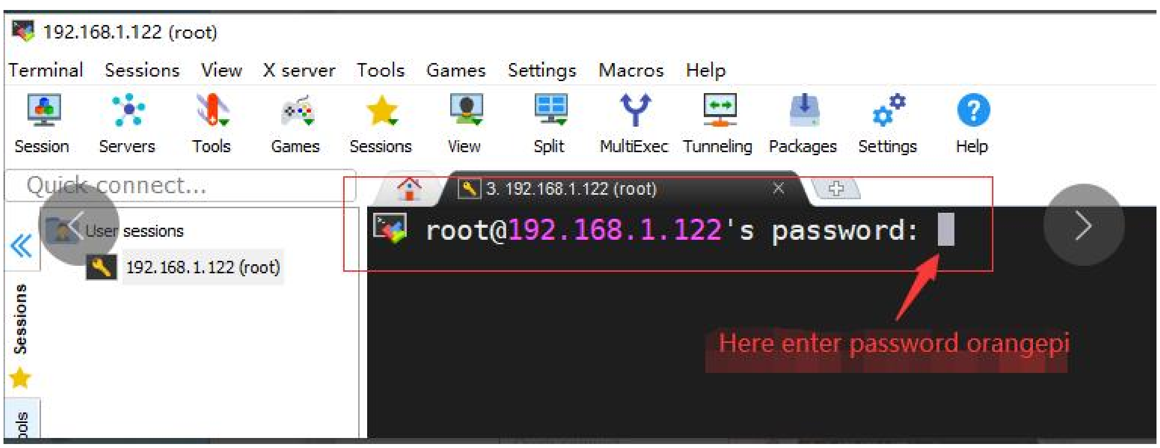
d. Then enter the user name root or orangepi of the linux system in Specify username

e. Finally click OK

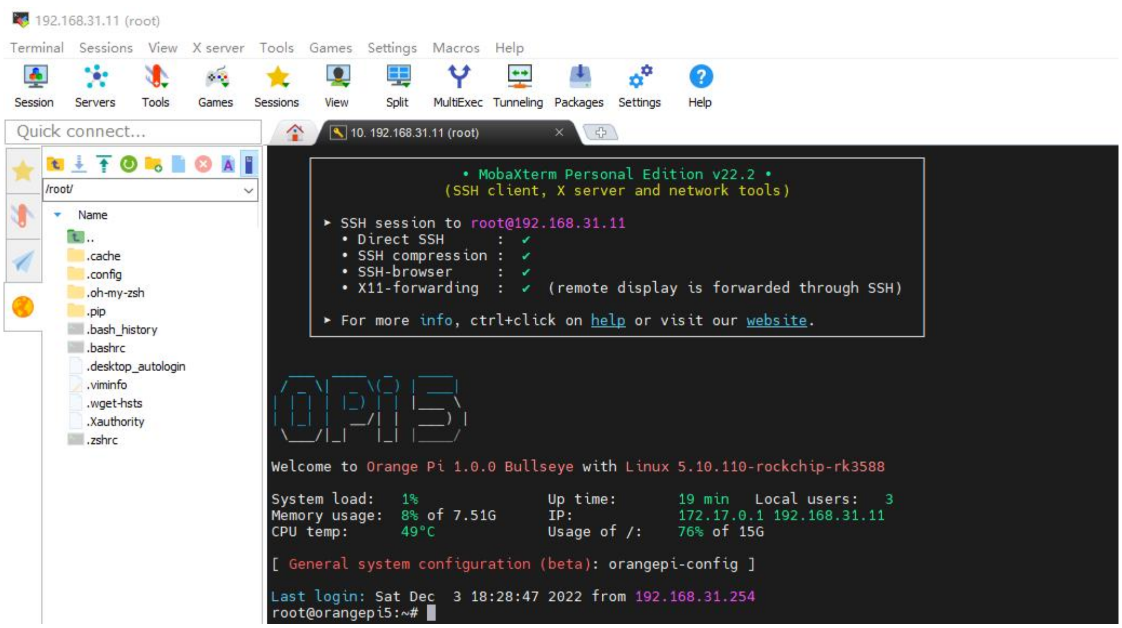



3) Then you will be prompted to enter a password. The default passwords for root and orangepi users are orangepi

**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 entering**




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




## 3. 8. How to use ADB？

### 3. 8. 1. How to use network adb

1) After the system starts, please confirm that adbd has been started

```
orangepi@orangepi:~$ ps -ax | grep "adbd"
    808 ?          Sl     0:00 /usr/bin/adbd
   3707 ttyFIQ0    S+     0:00 grep --color=auto adbd
```

2) Then check the IP address of the development board and write it down

3) Then install the adb tool on the Ubuntu PC

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

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

```
test@test:~$ adb connect 192.168.1.xx:5555      #Please replace the IP address with the IP address of the development board
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xx:5555
test@test:~$ adb devices
```

```
List of devices attached
192.168.1.xx:5555          device
```

5) Then use the following command to log in to the linux system of the development board

```
test@test:~$ adb shell
root@orangepi5:/#      <--- After seeing this prompt, it means that you have successfully logged in to the development board
```

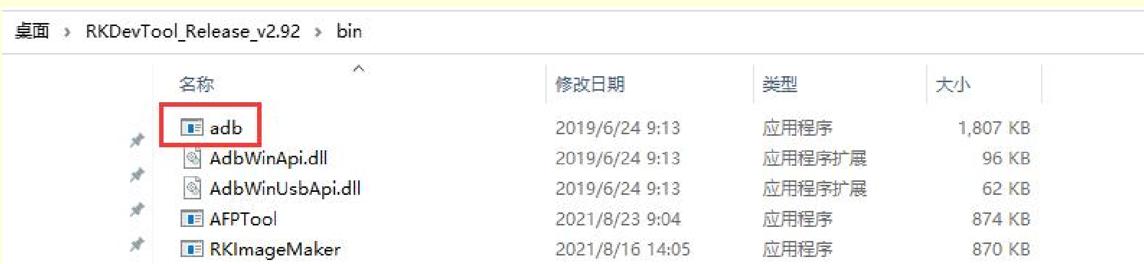
6) The command to upload files to the development board using adb is as follows

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

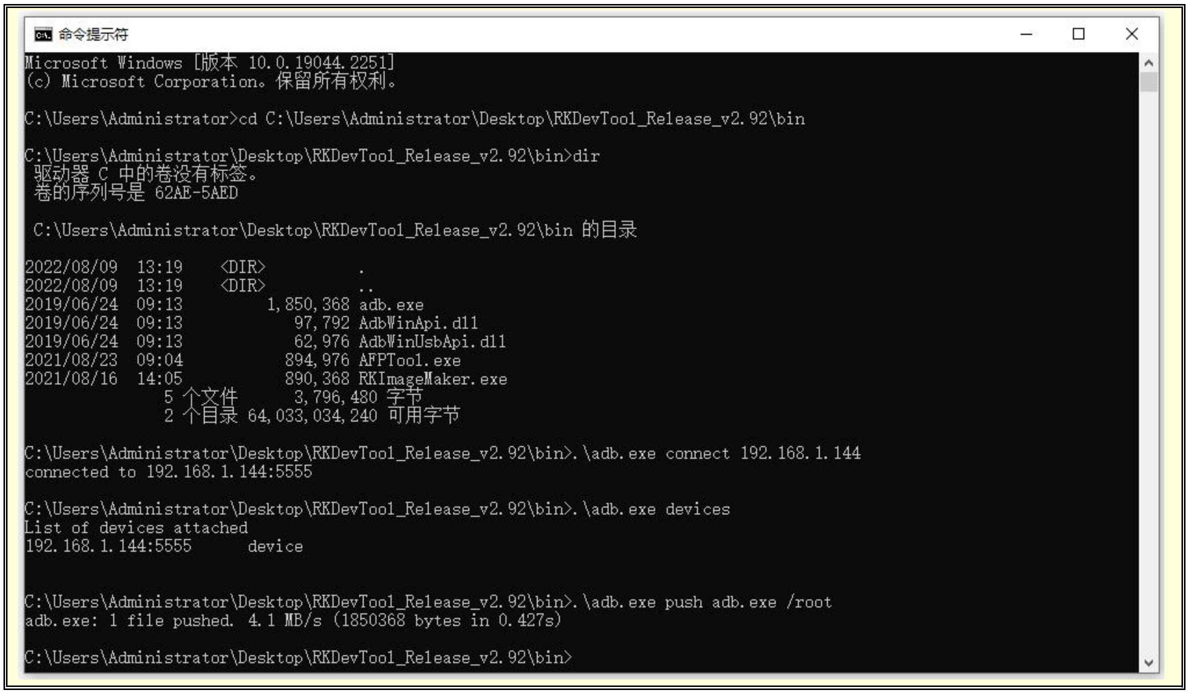
7) The command to restart the development board using adb is as follows

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

**If there is no adb tool in your Windows system, you can use the adb program in the RKDevTool software (this software is useful in the section on how to burn the Android image to SPIFlash+NVMe SDD).**



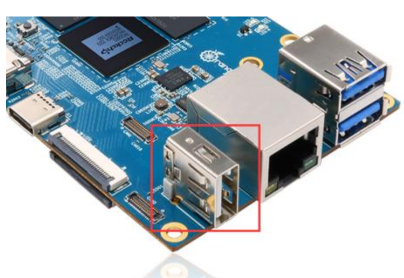

**An example of using adb in windows is shown below:**



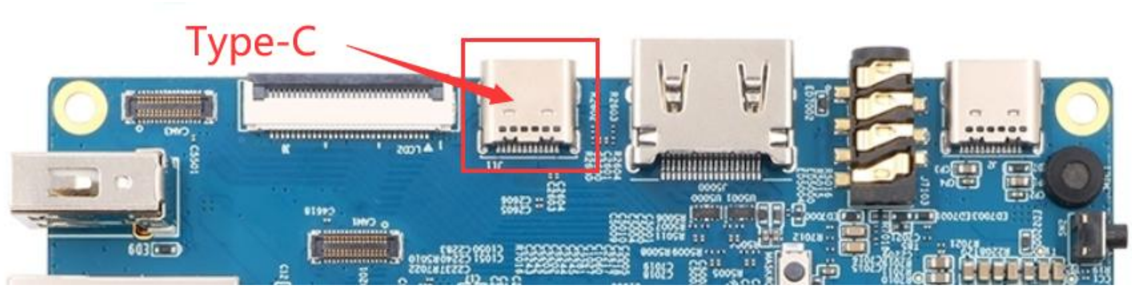

### 3.8.2. Use type-c data cable to connect to adb

1) First prepare a good quality Type-C data cable

2) Then please make sure that there is no USB device plugged into the USB interface below



3) Then connect the development board to the Ubuntu PC through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below:




4) Then run the following command to set the Type-C interface to device mode

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

If the set_device.sh script does not exist in the linux system, please use the following command directly:

```
orangepi@orangepi:~$ sudo bash -c "echo device > /sys/kernel/debug/usb/fc000000.usb/mode"
orangepi@orangepi:~$ sudo systemctl restart usbdevice
```

5) **Then please confirm that adbd has been started**

```
orangepi@orangepi:~$ ps -ax | grep "adbd"
    808 ?           Sl     0:00 /usr/bin/adbd
   3707 ttyFIQ0    S+     0:00 grep --color=auto adbd
```

6) Then install the adb tool on the Ubuntu PC

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

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

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

8) Then use the following command to log in to the linux system of the development board

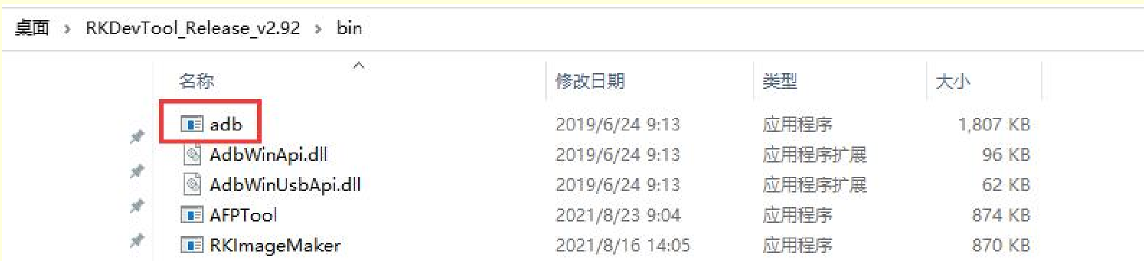
```
test@test:~$ adb shell
```

**root@orangepi5:/# <--- After seeing this prompt, it means that you have successfully logged in to the development board**

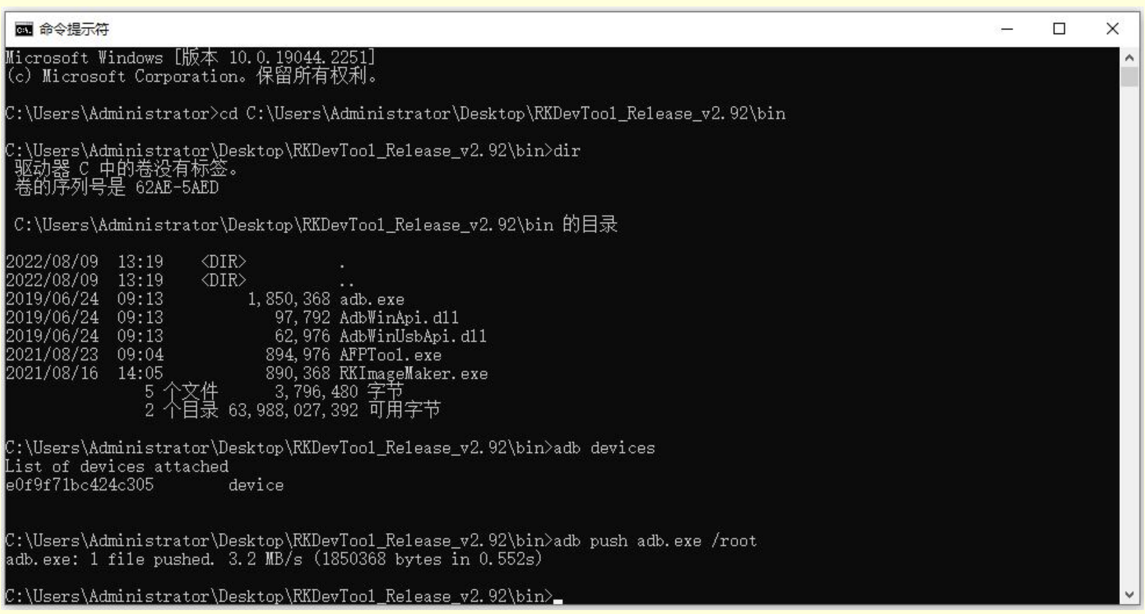
9) The command to upload files to the development board using adb is as follows

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

**If there is no adb tool in your Windows system, you can use the adb program in the RKDevTool software (this software is useful in the section on how to burn the Android image to SPIFlash+NVMe SDD).**




**An example of using adb in windows is shown below:**



## 3.9. The method of uploading files to the Linux system of the development board

### 3.9.1. The method of uploading files to the development board Linux system in Ubuntu PC

#### 3.9.1.1. How to upload files using the scp command

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

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

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

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

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

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

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

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

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

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

#### 3.9.1.2. How to upload files using filezilla

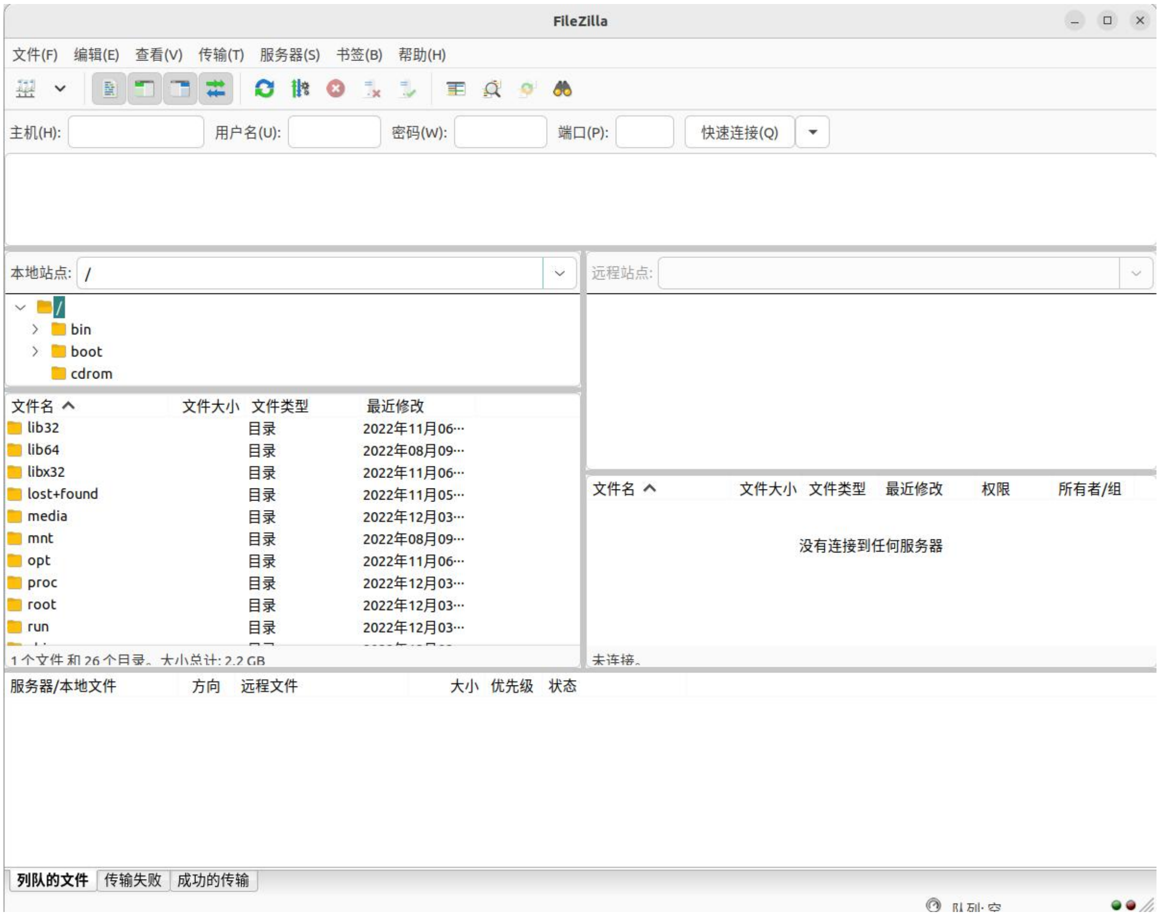
1) First install filezilla in Ubuntu PC

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

2) Then use the following command to open filezilla

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

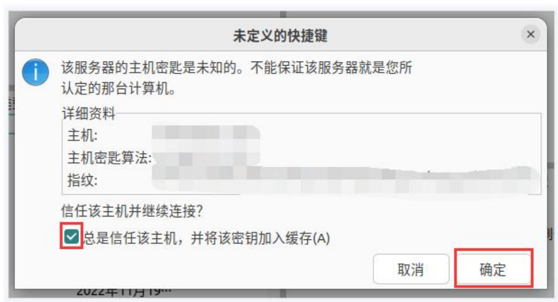
3) The interface after filezilla is opened is as follows, at this time, the display under the remote site on the right is empty



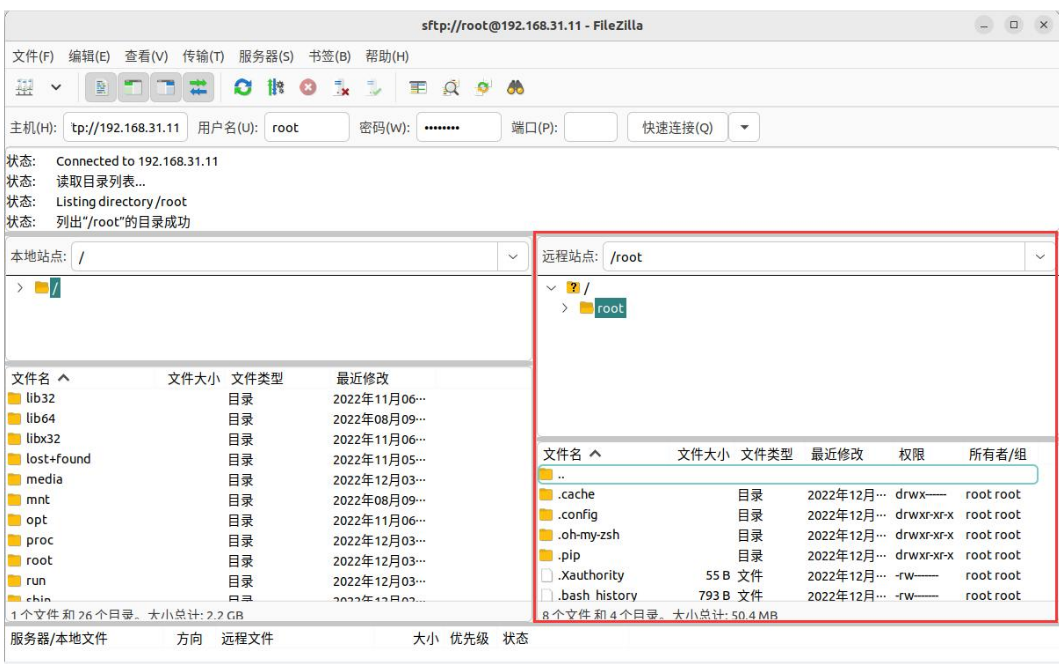

4) The method of connecting the development board is shown in the figure below


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


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

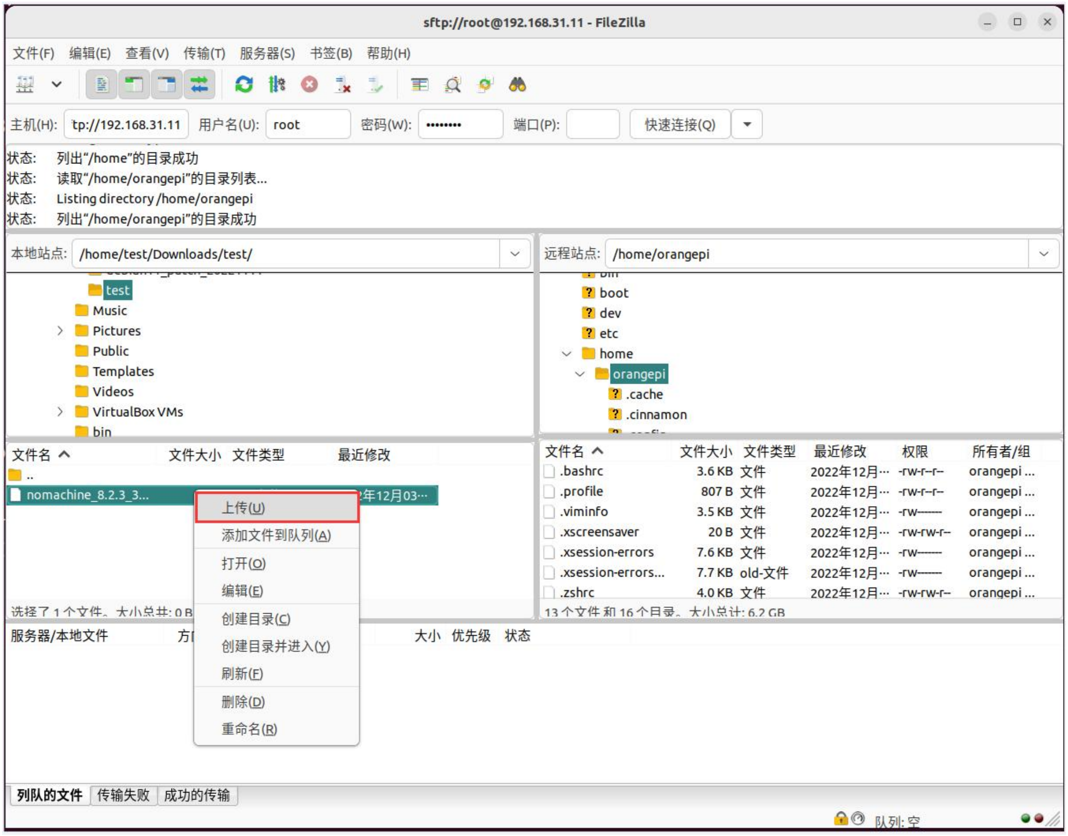



7) After the connection is successful, you can see the directory structure of the development board linux file system on the right side of the filezilla software




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

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



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

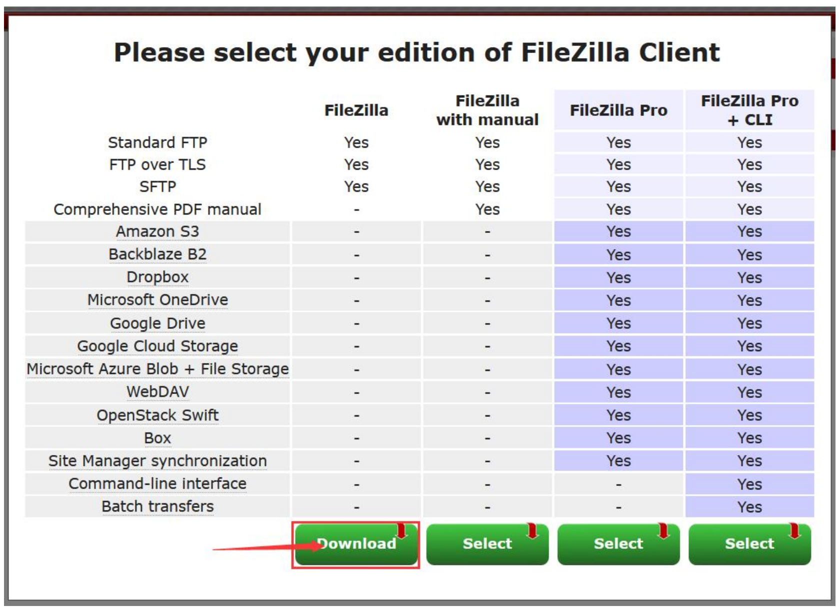
10) The method of uploading a folder is the same as that of uploading a file, so I won't go into details here

## 3. 9. 2. The method of uploading files to the Linux system of the development board in Windows PC

### 3. 9. 2. 1. How to upload files using filezilla

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

| |
|---|
| https://filezilla-project.org/download.php#close |

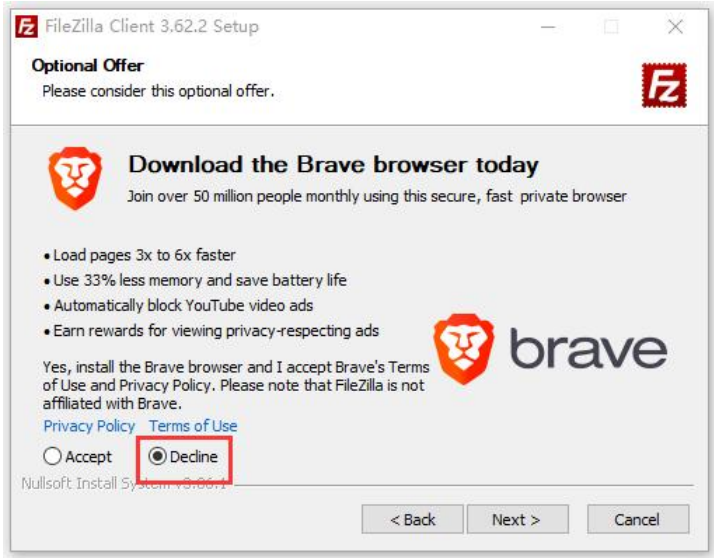
**Please select your edition of FileZilla Client**

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

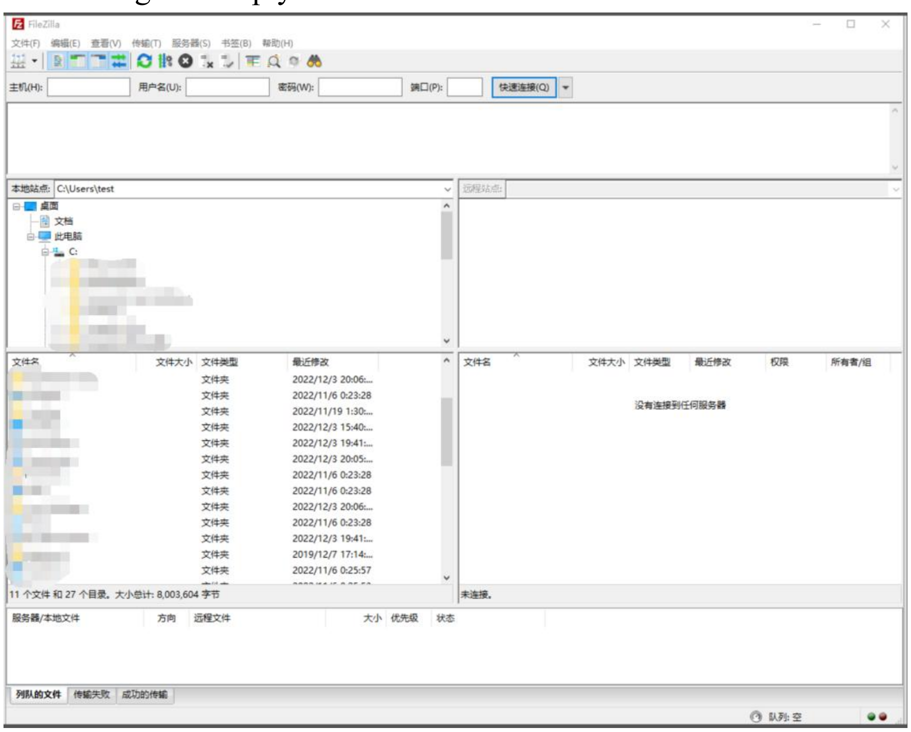
2) The downloaded installation package is as follows, and then double-click to install directly

**FileZilla_Server_1.5.1_win64-setup.exe**

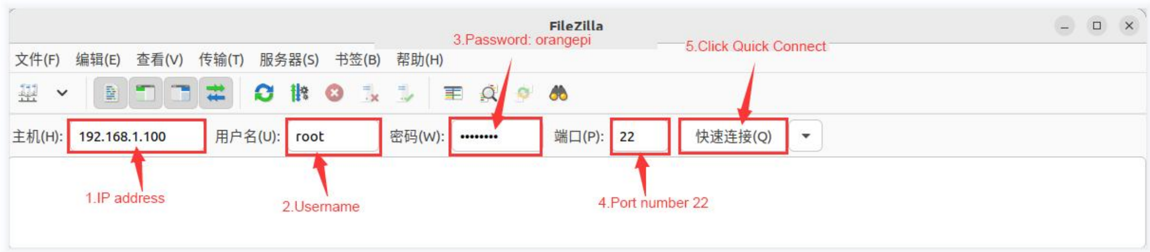
During the installation process, please select **Decline** on the following installation interface, and then select <**Next**>



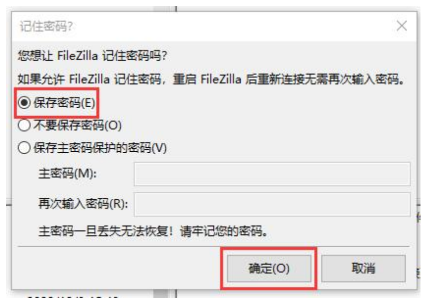

3) The interface after filezilla is opened is as follows, at this time, the display under the remote site on the right is empty



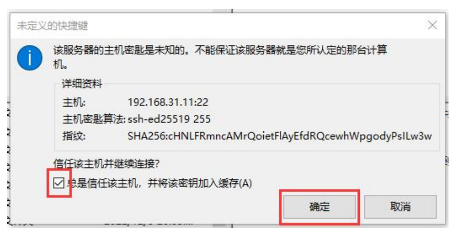
4) The method of connecting the development board is shown in the figure below：



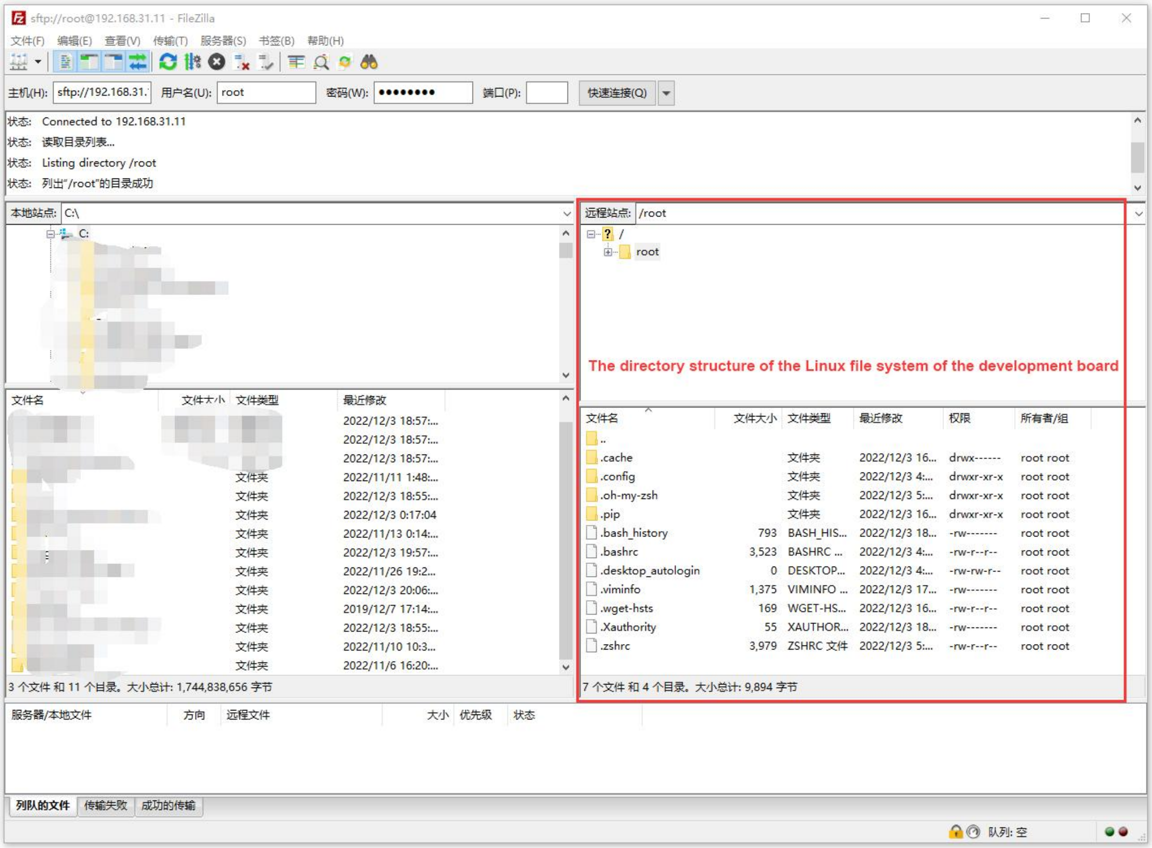

5) Then choose to **save the password**, and then click **OK**



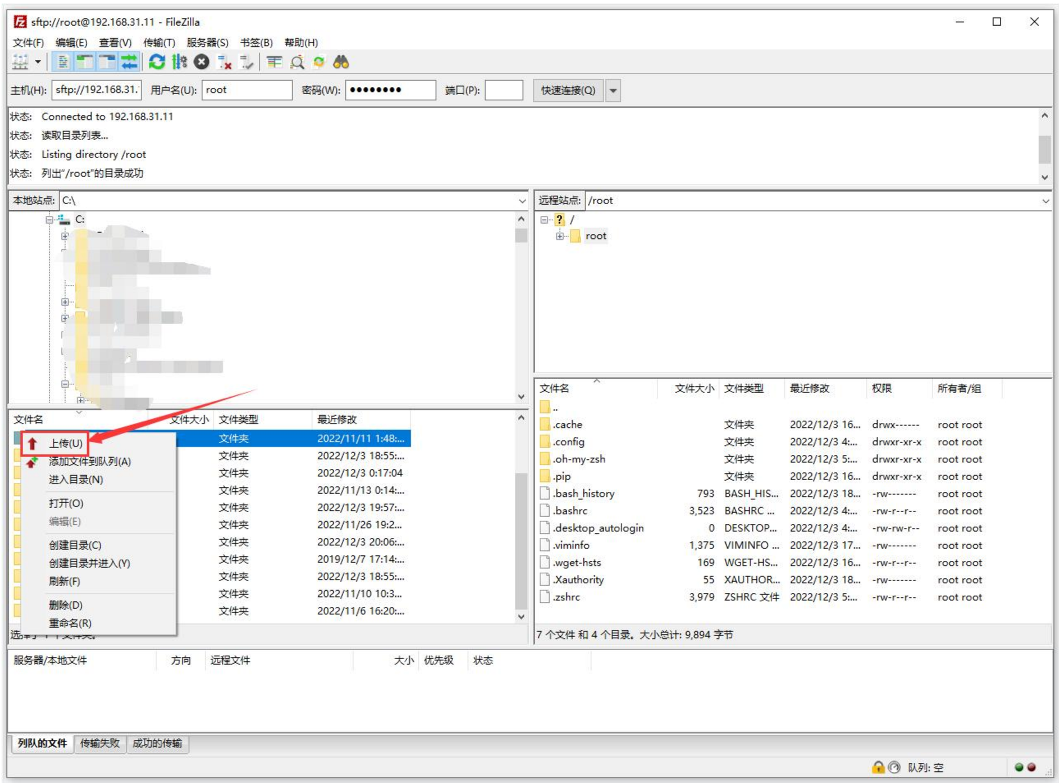

6) Then choose to **always trust this hos**t, and then click **OK**




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 Windows PC on the left side of the filezilla software, then click the right mouse button, and then click the upload option to start uploading the file to the development board bingo




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

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

## 3. 10. HDMI test

### 3. 10. 1. HDMI display test

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

2) After starting the linux system, if the HDMI monitor has image output, it means that the HDMI interface is working normally

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

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

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

### 3.10.2. HDMI to VGA display test

1) First, you need to prepare the following accessories

a. HDMI to VGA Converter


b. A VGA cable

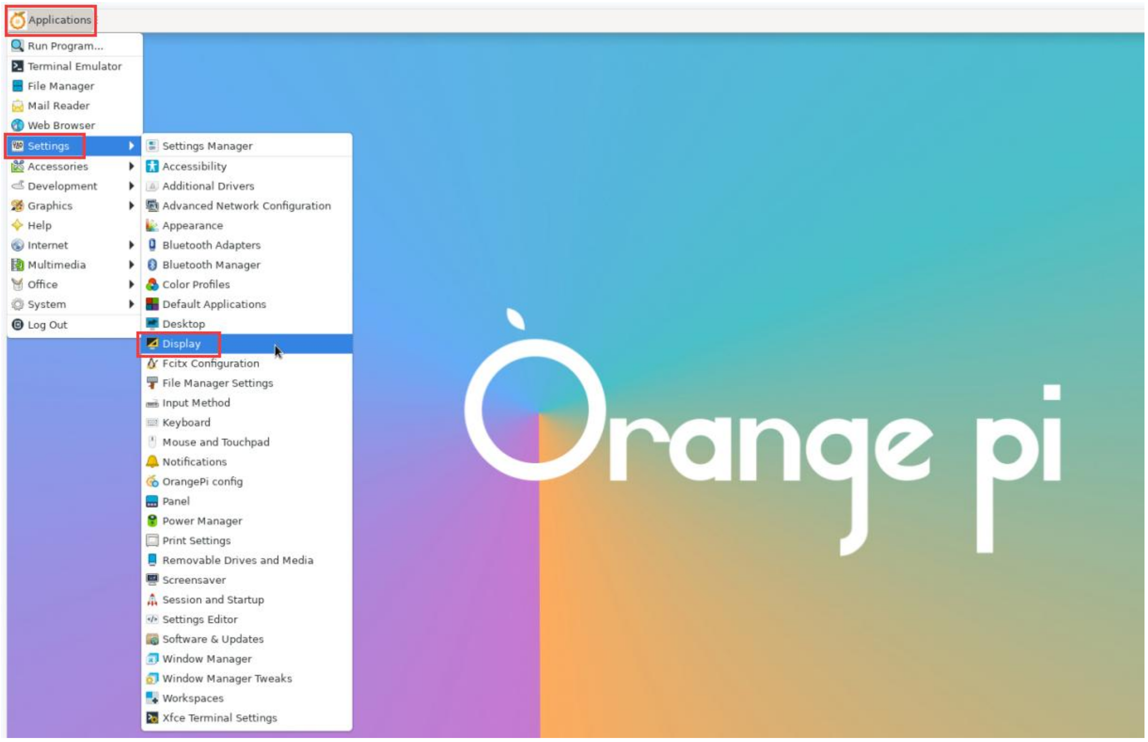
c. A monitor or TV that supports VGA interface

2) HDMI to VGA display test as shown below

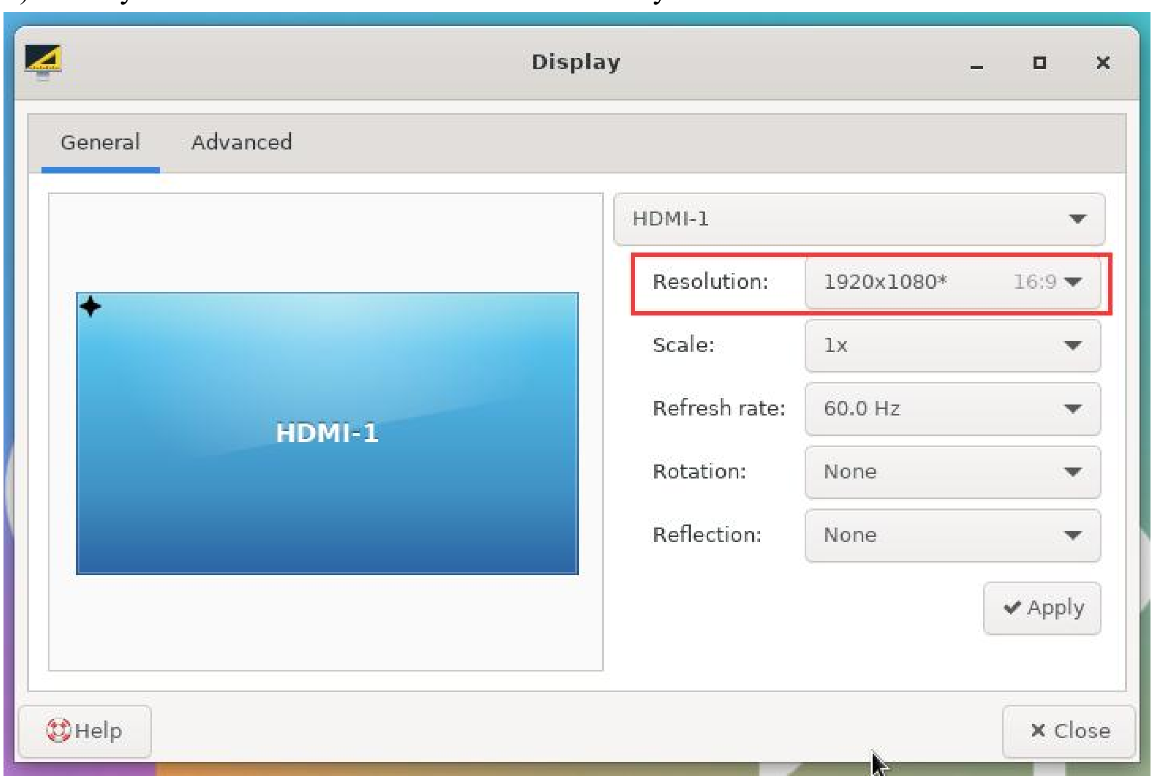

**When using HDMI to VGA display, the development board and the Linux system of the development board do not need to make any settings, only the HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI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

### 3. 10. 3. HDMI resolution setting method

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

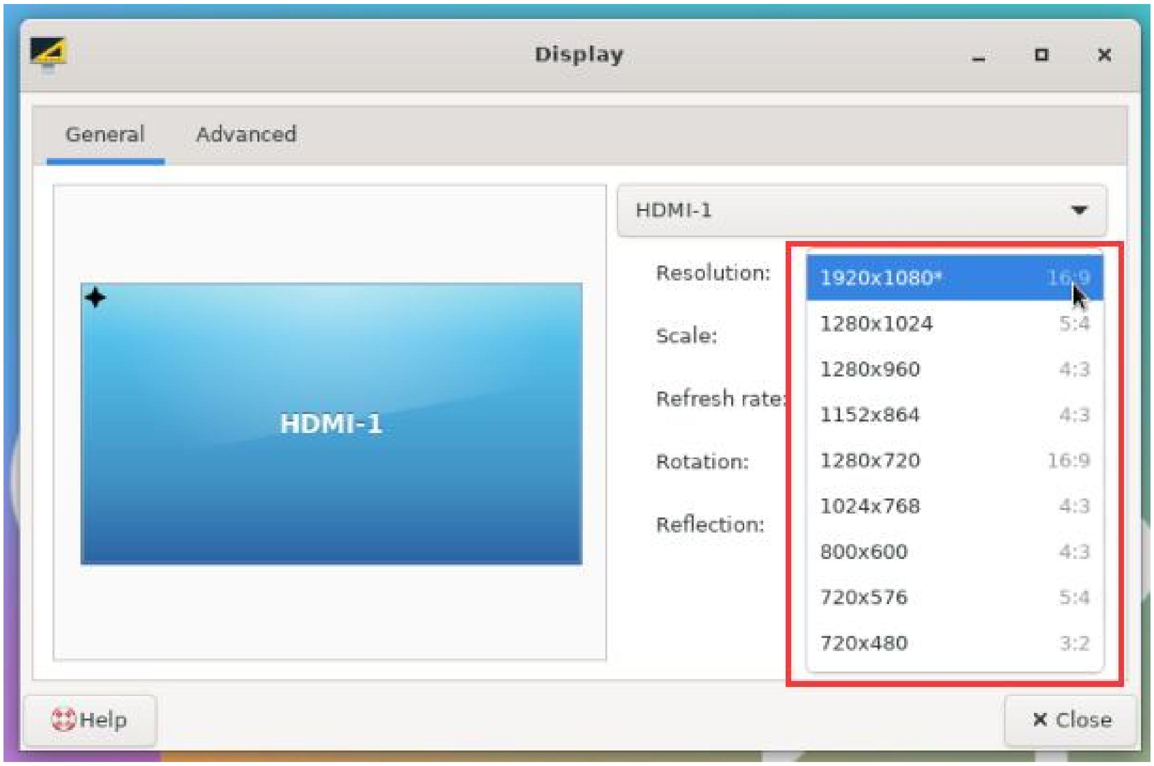


2) Then you can see the current resolution of the system

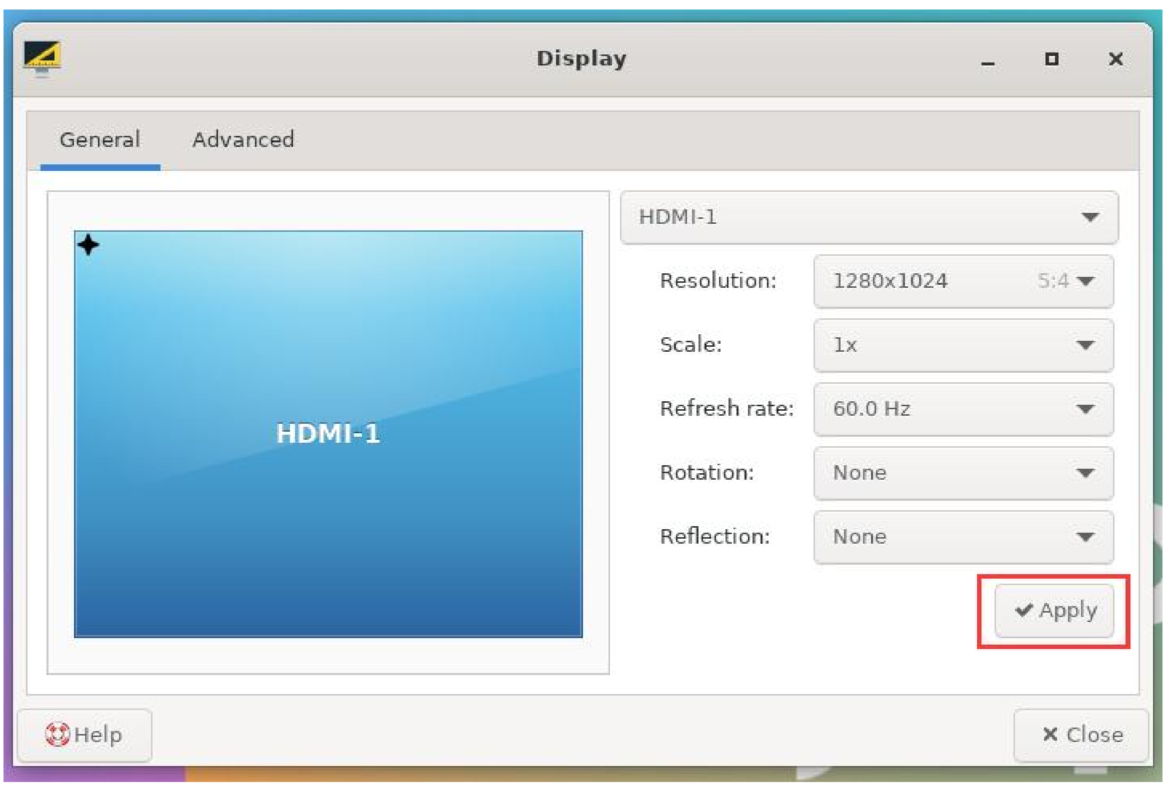



3) Click the drop-down box of Resolution to see all resolutions currently supported by

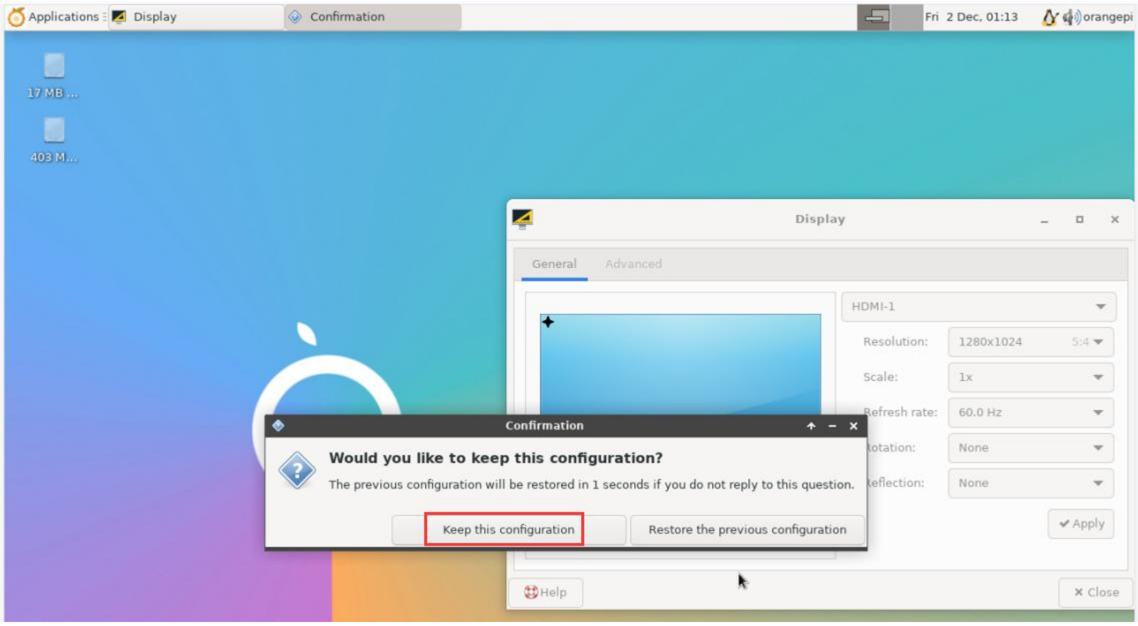
the display




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



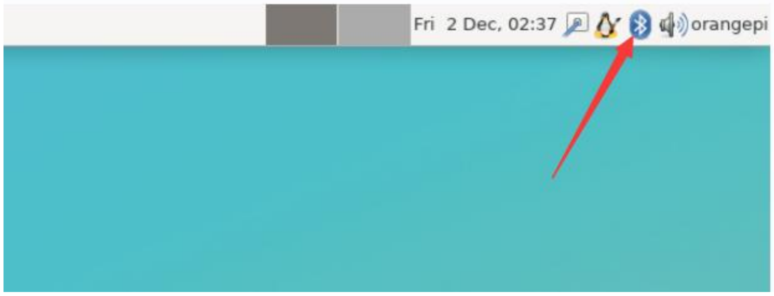
5) After the new resolution is set, select **Keep the configuration**



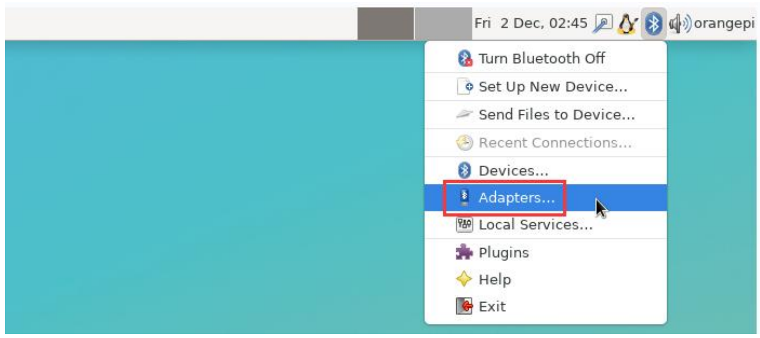

## 3. 11. How to use Bluetooth

### 3. 11. 1. Test method of desktop image

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

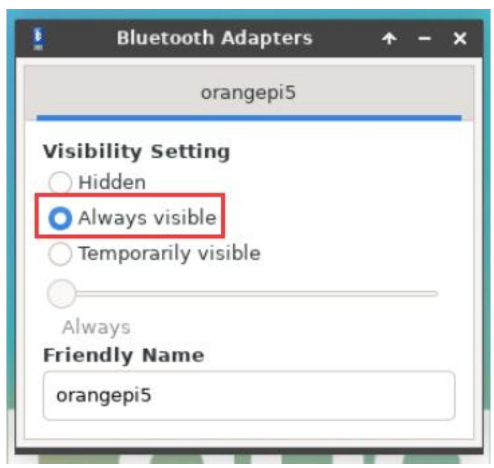



2) Then select the adapter

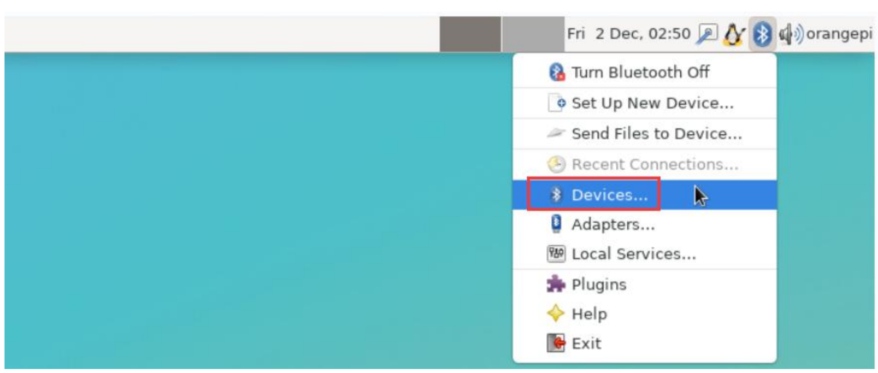



3) If there is a prompt on the following interface, please select **Yes**

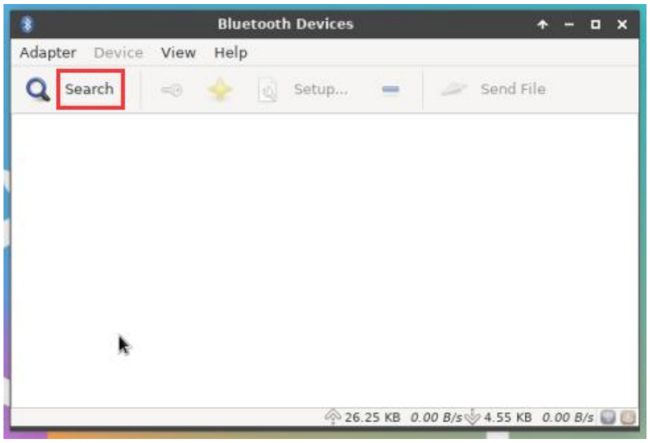

4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter setting interface, and then close it



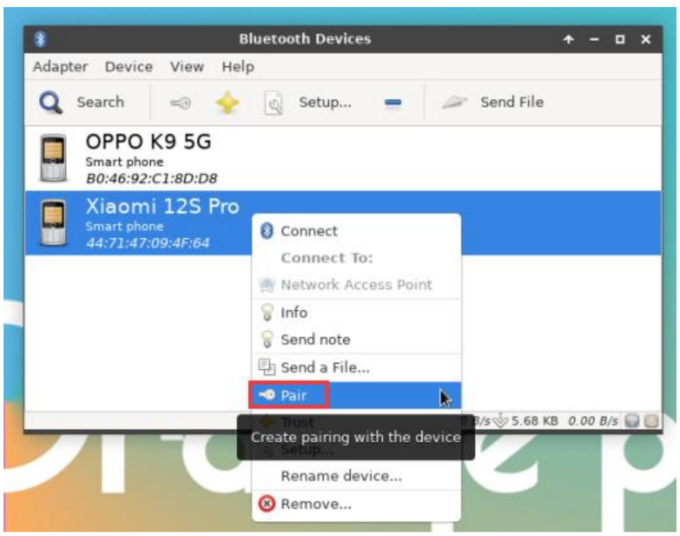

5) Then open the configuration interface of the Bluetooth device



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

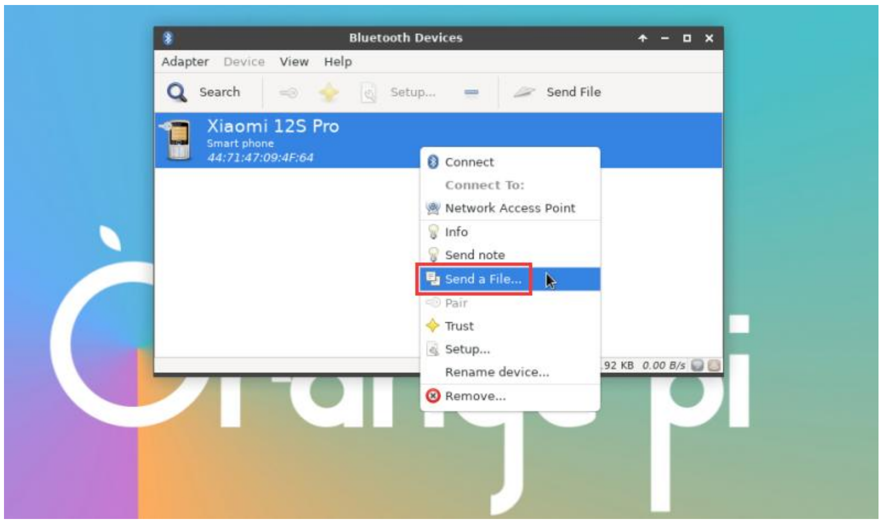



6) Then select the Bluetooth device you want to connect to, and then click the right button of the mouse to pop up the operation interface of the Bluetooth device, select Pair to start pairing, and the demonstration here is to pair with an Android phone

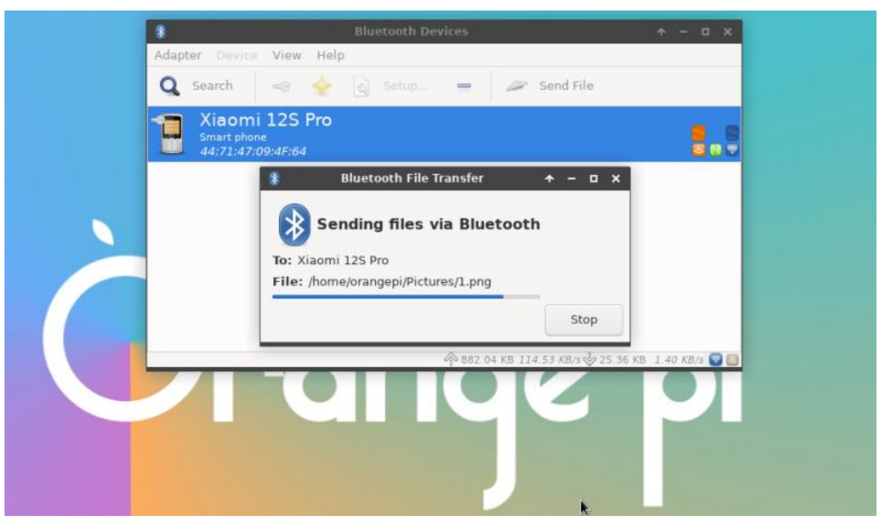



7) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop, just select Confirm to confirm, and the phone also needs to confirm at this time


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




9) The interface for sending pictures is as follows



## 3. 12. USB interface test

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

### 3. 12. 1. Connect USB mouse or keyboard to test

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

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

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

### 3. 12. 2. Connect USB storage device test

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

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

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

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

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

The following command can be used to mount the U disk in exfat format on the Linux system

```
orangepi@orangepi:~$ sudo apt-get install exfat-utils exfat-fuse
orangepi@orangepi:~$ sudo mount -t exfat /dev/sda1 /mnt/
```

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

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

### 3. 12. 3. USB wireless network card test

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

| serial number | model | |
|---|---|---|
| 1 | RTL8723BU<br>Support 2.4G WIFI+BT4.0 |  |
| 2 | RTL8811<br>Support 2.4G +5G WIFI |  |
| 3 | RTL8821CU<br>Support 2.4G +5G WIFI<br>Support BT 4.2 |  |

#### 3. 12. 3. 1. RTL8723BU test

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

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

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

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

```
orangepi@orangepi:~$ dmesg
......
[   83.438901] usb 2-1: new high-speed USB device number 2 using ehci-platform
[   83.588375] usb 2-1: New USB device found, idVendor=0bda, idProduct=b720, bcdDevice= 2.00
[   83.588403] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   83.588422] usb 2-1: Product: 802.11n WLAN Adapter
[   83.588443] usb 2-1: Manufacturer: Realtek
[   83.588460] usb 2-1: SerialNumber: 00e04c000001
[   83.601974] Bluetooth: hci0: RTL: examining hci_ver=06 hci_rev=000b lmp_ver=06 lmp_subver=8723
[   83.603894] Bluetooth: hci0: RTL: rom_version status=0 version=1
[   83.603920] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_fw.bin
[   83.610108] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_config.bin
[   83.611274] Bluetooth: hci0: RTL: cfg_sz 68, total sz 22564
[   83.658494] rtk_btusb: Realtek Bluetooth USB driver ver 3.1.6d45ddf.20220519-142432
[   83.658651] usbcore: registered new interface driver rtk_btusb
[   83.667124] usb 2-1: This Realtek USB WiFi dongle (0x0bda:0xb720) is untested!
[   83.667137] usb 2-1: Please report results to Jes.Sorensen@gmail.com
[   83.890140] usb 2-1: Vendor: Realtek
[   83.890153] usb 2-1: Product: 802.11n WLAN Adapter
[   83.890159] usb 2-1: rtl8723bu_parse_efuse: dumping efuse (0x200 bytes):
......
[   83.890412] usb 2-1: RTL8723BU rev E (SMIC) 1T1R, TX queues 3, WiFi=1, BT=1, GPS=0, HI PA=0
[   83.890417] usb 2-1: RTL8723BU MAC: 00:13:ef:f4:58:ae
[   83.890421] usb 2-1: rtl8xxxu: Loading firmware rtlwifi/rtl8723bu_nic.bin
[   83.895289] usb 2-1: Firmware revision 35.0 (signature 0x5301)
[   84.050893] Bluetooth: hci0: RTL: fw version 0x0e2f9f73
[   84.266905] Bluetooth: RFCOMM TTY layer initialized
[   84.266949] Bluetooth: RFCOMM socket layer initialized
[   84.266999] Bluetooth: RFCOMM ver 1.11
```

```
[    84.884270] usbcore: registered new interface driver rtl8xxxu
[    84.912046] rtl8xxxu 2-1:1.2 wlx0013eff458ae: renamed from wlan0
```

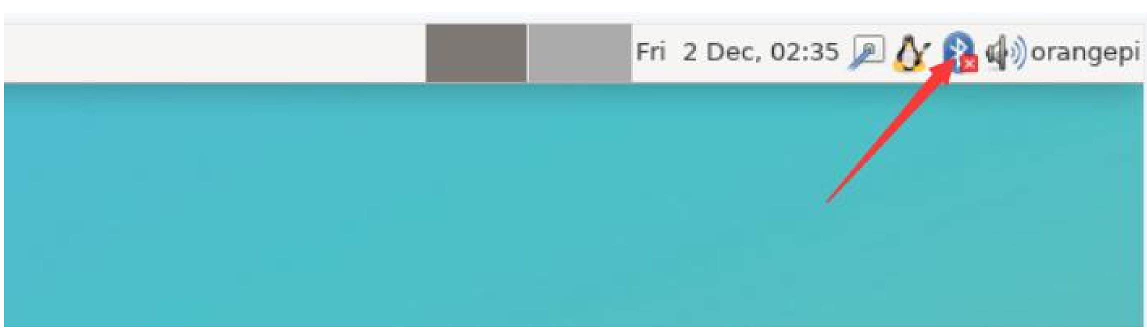
4) Then you can see the device node of RTL8723BU WIFI through the **sudo ifconfig** command. For the connection and test method of WIFI, please refer to **the section of WIFI connection test**, which will not be repeated here

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

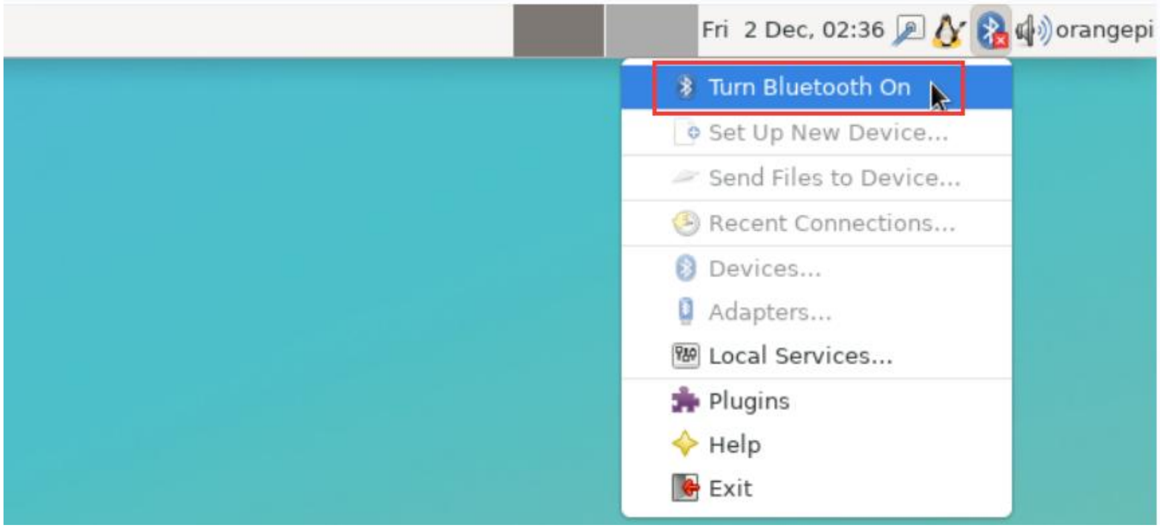
5) Then you can see the USB Bluetooth device through the **hciconfig** command

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

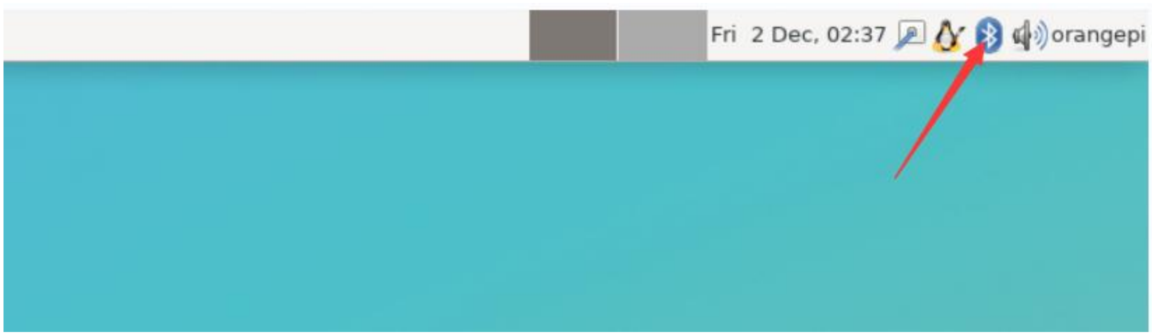
6) You can also see the bluetooth icon on the desktop. At this time, the bluetooth is not turned on, so a red **x** will be displayed




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




8) The display after turning on Bluetooth is as follows




9) For the test method of Bluetooth, please refer to **the section on Bluetooth usage**, so I won't go into details here

### 3. 12. 3. 2. RTL8811 test

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

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

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

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

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

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

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

### 3. 12. 3. 3. RTL8821CU test

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

2) Then use the lsusb command to see the device information of the rtl8821cu usb wifi module, please make sure that the USB module is not in Driver CDROM Mode

```
orangepi@orangepi:~$ lsusb | grep "Realtek"
Bus 002 Device 003: ID 0bda:c820 Realtek Semiconductor Corp. 802.11ac NIC
```

```
orangepi@orangepi:~$ lsusb | grep "Realtek"
Bus 002 Device 002: ID 0bda:1a2b Realtek Semiconductor Corp. RTL8188GU 802.11n
WLAN Adapter (Driver CDROM Mode)
```

**If the USB WIFI module seen by the lsusb command is in Driver CDROM Mode, please unplug the USB WIFI module again. If not, please manually execute the following command to switch to the next mode:**

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

3) The linux system will automatically load the rtl8821cu bluetooth and wifi related kernel modules, through the lsmod command, you can see that the following kernel modules have been automatically loaded

```
orangepi@orangepi:~$ lsmod
Module                  Size  Used by
8821cu               1839104  0
rtk_btusb              61440  0
```

4) Through the dmesg command, you can see the loading information of the rtl8821cu module

```
orangepi@orangepi:~$ dmesg
......
[   57.083693] usb 2-1: new high-speed USB device number 2 using ehci-platform
[   57.231888] usb 2-1: New USB device found, idVendor=0bda, idProduct=1a2b,
bcdDevice= 2.00
[   57.231916] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[   57.231937] usb 2-1: Product: DISK
[   57.231956] usb 2-1: Manufacturer: Realtek
[   57.242594] usb-storage 2-1:1.0: USB Mass Storage device detected
[   57.245674] scsi host0: usb-storage 2-1:1.0
[   58.069172] usb 2-1: USB disconnect, device number 2
[   58.440025] usb 2-1: new high-speed USB device number 3 using ehci-platform
[   58.587819] usb 2-1: New USB device found, idVendor=0bda, idProduct=c820,
bcdDevice= 2.00
[   58.587827] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
```

```
[    58.587833] usb 2-1: Product: 802.11ac NIC
[    58.587838] usb 2-1: Manufacturer: Realtek
[    58.587844] usb 2-1: SerialNumber: 123456
[    58.610463] rtk_btusb: Realtek Bluetooth USB driver ver 3.1.6d45ddf.20220519-142432
[    58.610656] usbcore: registered new interface driver rtk_btusb
[    58.634631] Bluetooth: hci0: RTL: examining hci_ver=08 hci_rev=000c lmp_ver=08 lmp_subver=8821
[    58.636729] Bluetooth: hci0: RTL: rom_version status=0 version=1
[    58.636740] Bluetooth: hci0: RTL: loading rtl_bt/rtl8821c_fw.bin
[    58.664190] Bluetooth: hci0: RTL: loading rtl_bt/rtl8821c_config.bin
[    58.664746] Bluetooth: hci0: RTL: cfg_sz 10, total sz 31990
[    59.122471] Bluetooth: hci0: RTL: fw version 0x829a7644
[    59.265513] usbcore: registered new interface driver rtl8821cu
[    59.280119] rtl8821cu 2-1:1.2 wlx90de80521825: renamed from wlan0
```

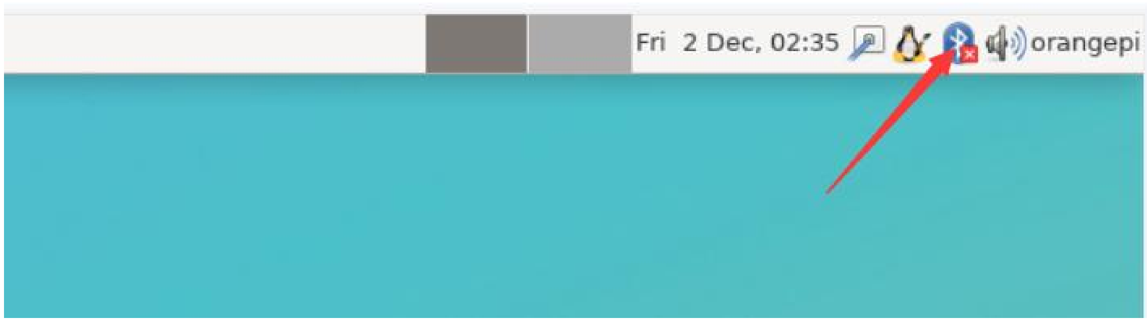
5) Then you can see the device node of rtl8821cu wifi through the **sudo ifconfig** command. For the wifi connection and test method, please refer to the section of **WIFI connection test**, which will not be repeated here

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

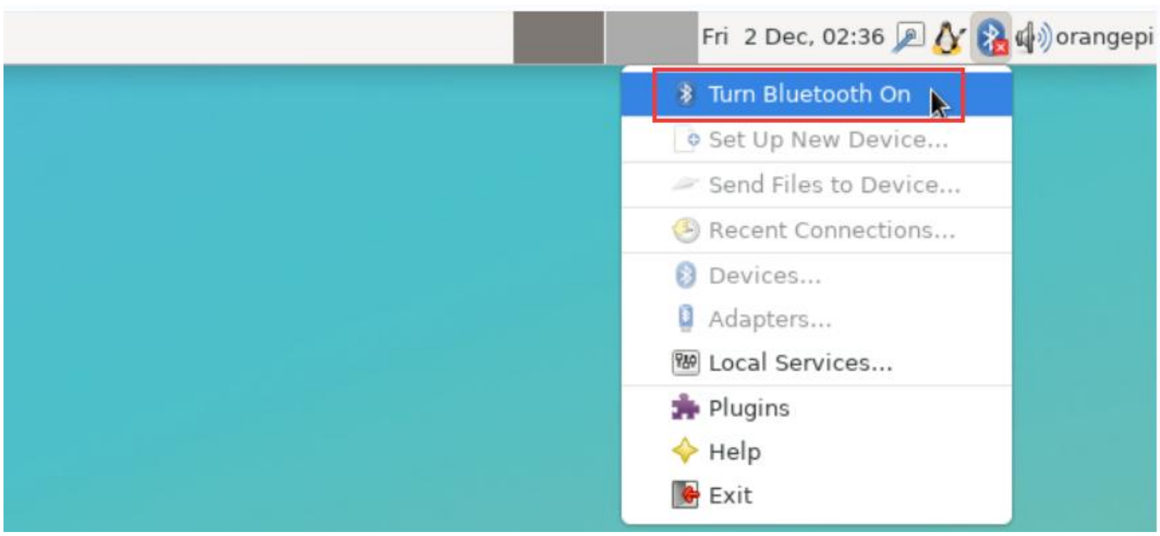
6) Then you can see the USB Bluetooth device through the **hciconfig** command

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

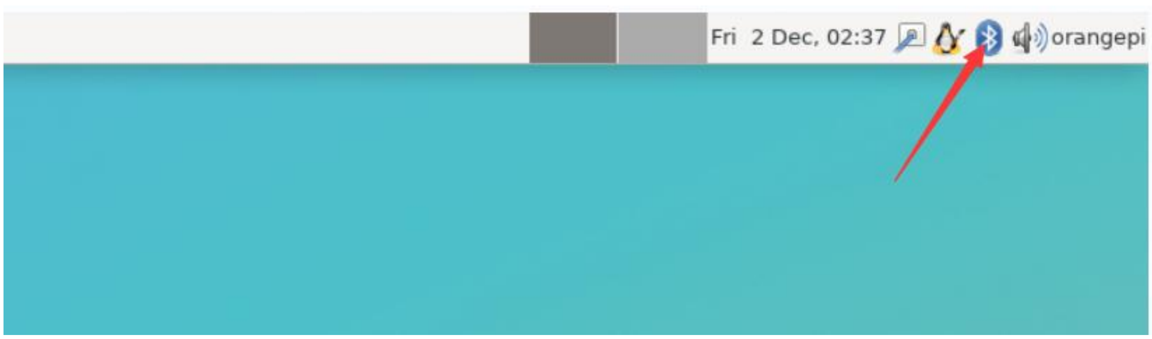
7) You can also see the bluetooth icon on the desktop. At this time, the bluetooth is not turned on, so a red **x** will be displayed




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




9) The display after turning on Bluetooth is as follows




10) For the test method of Bluetooth, please refer to **the section on Bluetooth usage**, so I won't go into details here

### 3. 12. 4. USB camera test

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

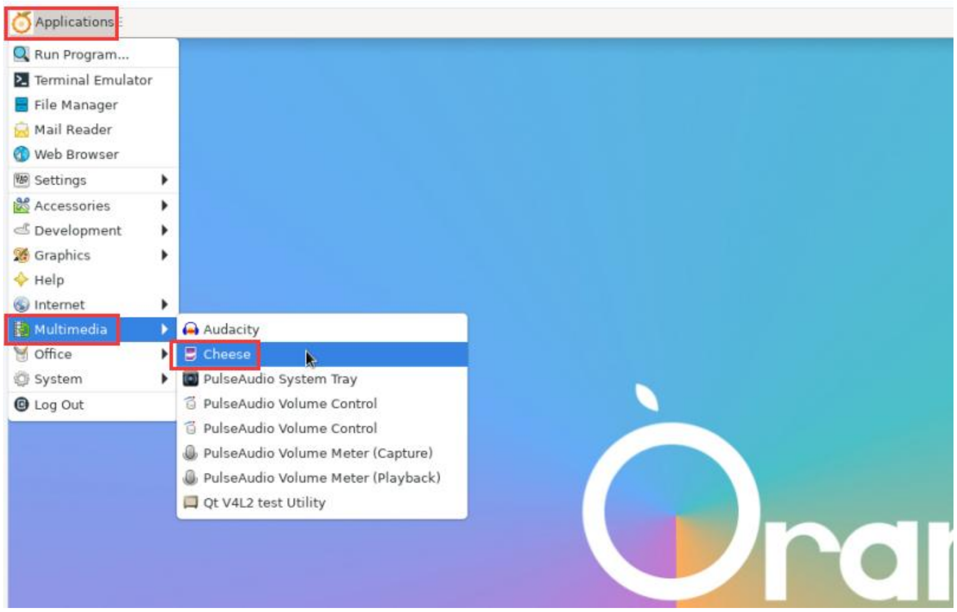

2) Through the v4l2-ctl command, you can see that the device node information of the USB camera is /dev/video0

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

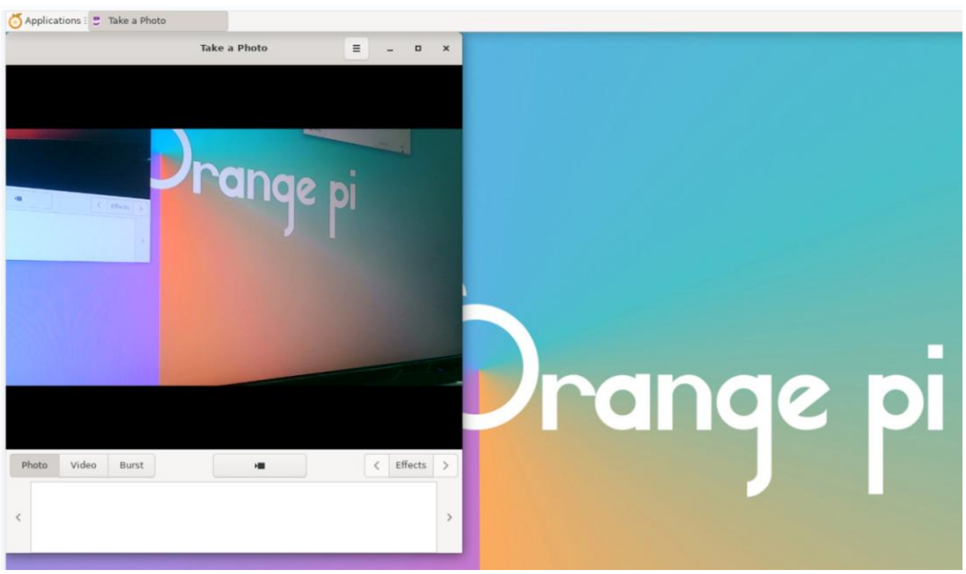
**Note that the l in v4l2 is a lowercase letter l, not the number 1.**

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

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




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




4) Method of using fswebcam to test USB camera

a. Install fswebcam

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

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

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

b) --no-banner is used to remove the watermark of the photo

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

d) The -S option is used to set the number of previous frames to skip

e)./image.jpg is used to set the name and path of the generated photo

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

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

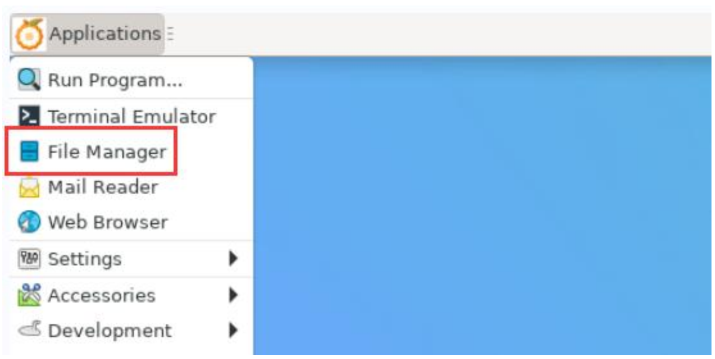
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test  （Modify the IP address and path according to the actual situation）
```

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

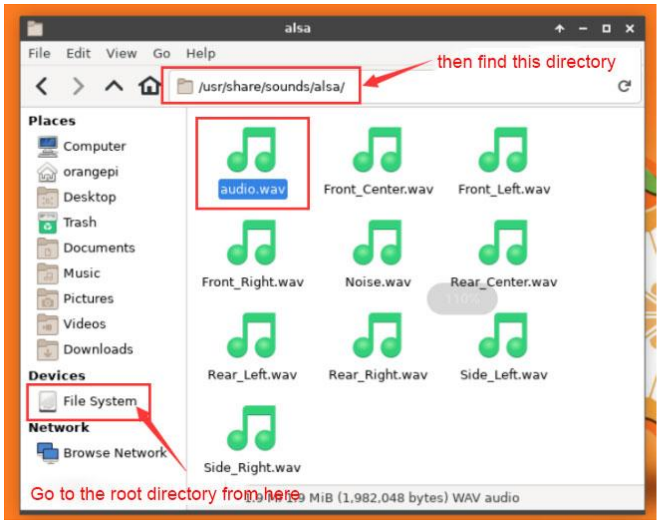
## 3. 13. Audio Test

### 3. 13. 1. Testing audio methods on desktop systems

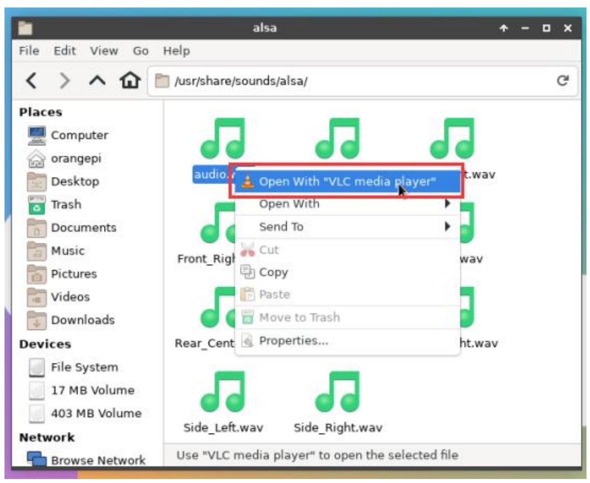
1) First open the file manager




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

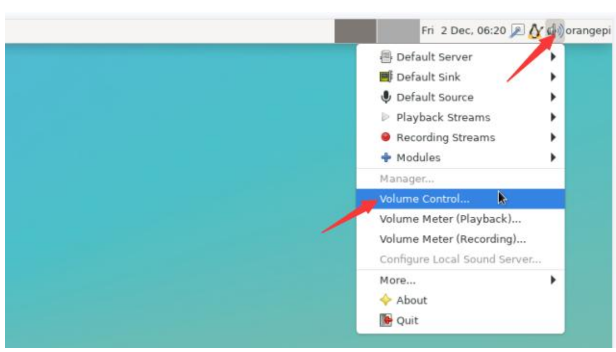



3) Then select the audio.wav file, right click and select open with vlc to start playing

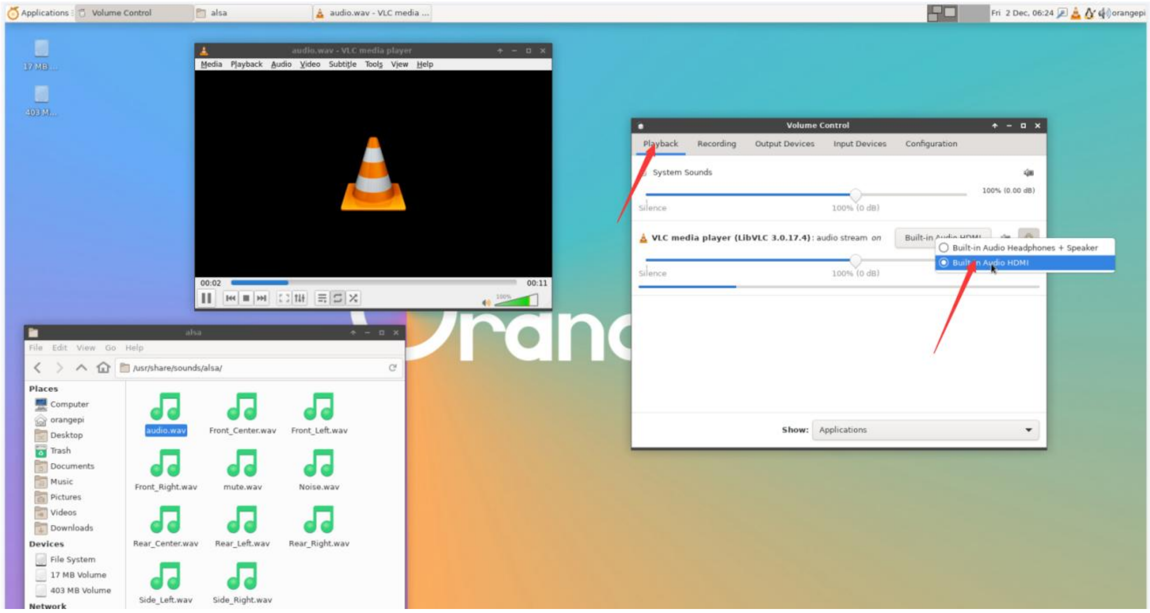


4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First open the volume control interface



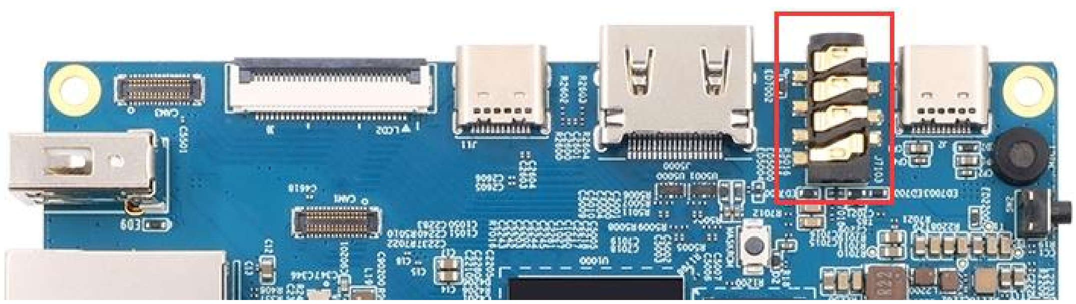
b. When playing audio, the audio device options that the playback software can use will be displayed in **Playback**, as shown in the figure below, where you can set which audio device to play to



### 3. 13. 2. The method of using commands to play audio

#### 3. 13. 2. 1. Headphone interface playback audio test

1) First insert the earphone into the earphone jack of the development board



2) Then you can use the aplay -l command to view the sound card devices supported by the linux system. From the output below, we can see that card 2 is the sound card device of es8388, that is, the sound card device of the headset

```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
```

```
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.6-0010-0 [dailink-multicodecs ES8323.6-0010-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3) Then use the aplay command to play the audio file that comes with the system. If the earphone can hear the sound, it means that the hardware can be used normally.

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

### 3. 13. 2. 2. HDMI audio playback test

1) First use the HDMI to HDMI cable to connect the Orange Pi development board to the TV (other HDMI monitors need to ensure that they can play audio)

2) Then check the serial number of the HDMI sound card. From the output below, you can know that the HDMI sound card is card 1

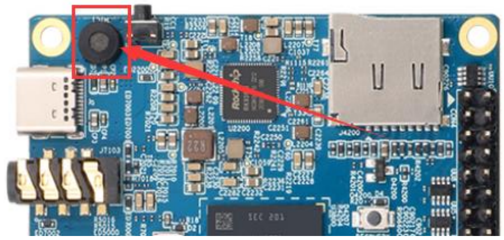
```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.6-0010-0 [dailink-multicodecs ES8323.6-0010-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

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

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

### 3. 13. 3. Method of using commands to test recording

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



2) Running the **test_record.sh main** command will record a piece of audio through the onboard MIC, and then play it to HDMI and headphones

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

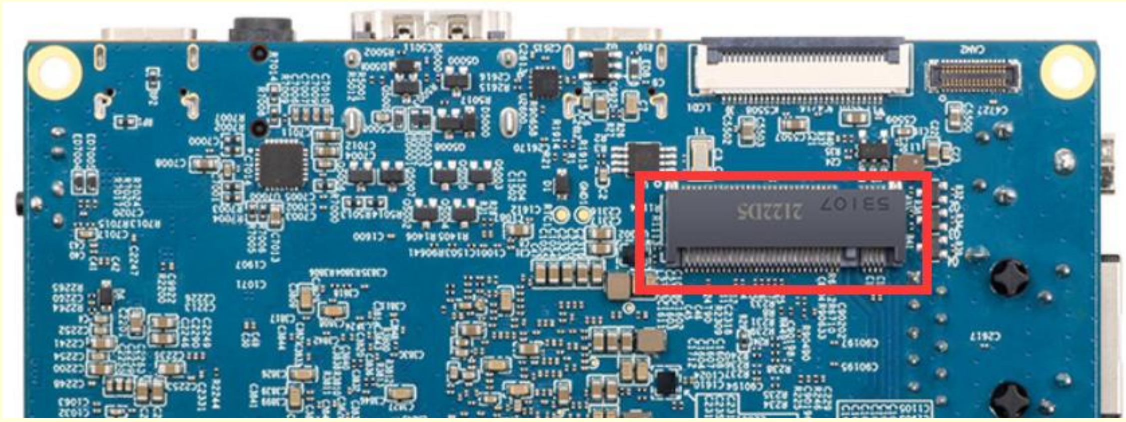
3) In addition to the onboard MIC, we can also record audio through headphones with MIC function. After inserting the headset with MIC function into the development board, run the **test_record.sh headset** command to record a piece of audio through the headset, and then play it to HDMI and the headset.

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

## 3. 14. How to use SATA SSD

**The m.2 interface shown in the figure below can use both nvme ssd and sata ssd. Since the pcie2.0 controller and the sata controller are optional, only one of them can be configured at the same time. The linux image released by Orange Pi opens the pcie configuration by default, so it can only recognize nvme ssd by default. If you**

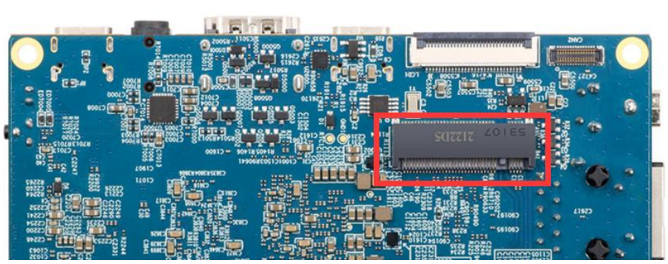
**want to use sata ssd, you need to open the corresponding configuration.**



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


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


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



3) There are two main usages of sata ssd：

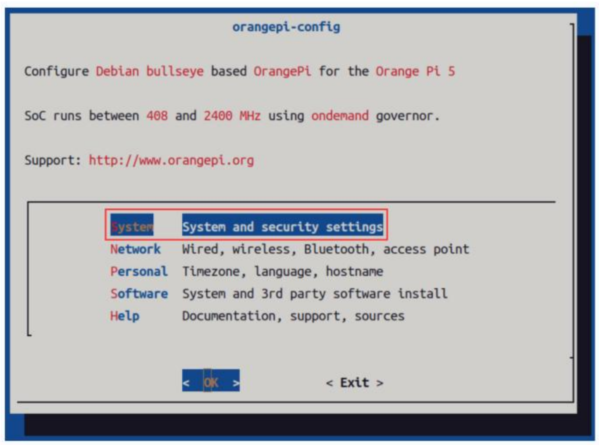
a. The linux system is in the TF card, and then insert the sata ssd as an external storage device. This section mainly explains this usage.

b. Burn the linux system into the sata ssd, and then start the linux system in the sata ssd. For this kind of usage, **please refer to the instructions in the section on the method of burning the Linux image to SPIFlash+SATA SDD**.

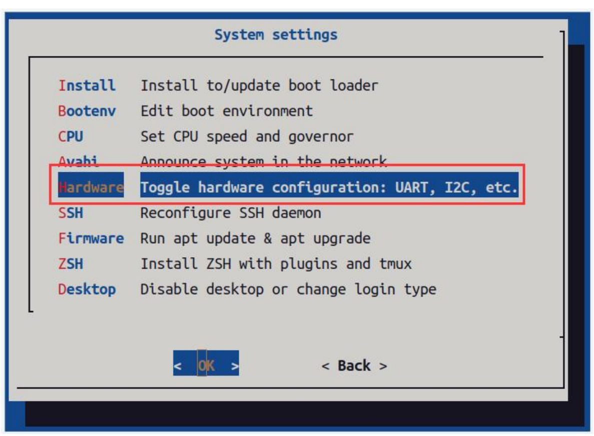
4) First run **orangepi-config**, normal users remember to add **sudo** permission

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

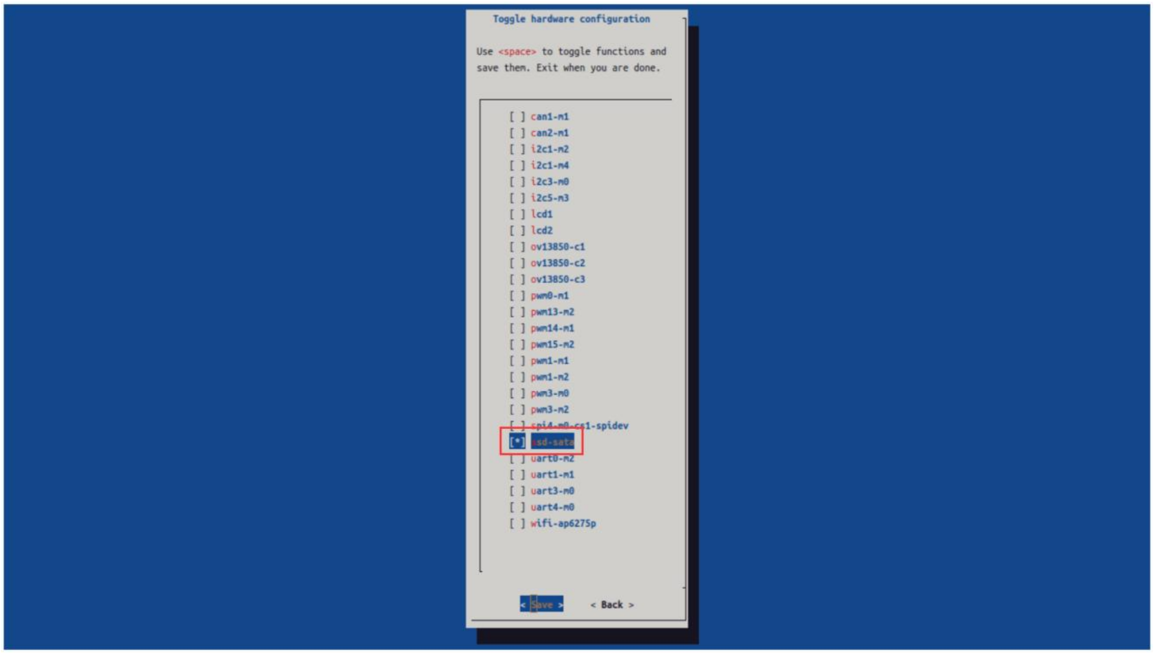
5) Then select **System**



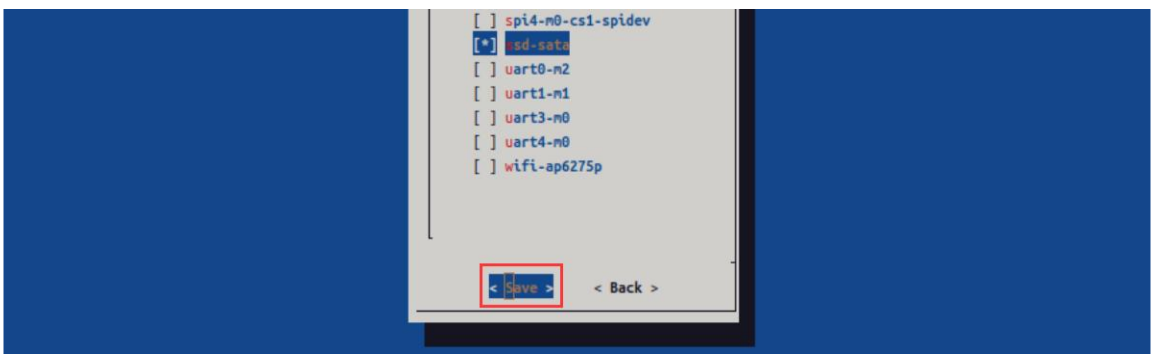

6) Then select **Hardware**



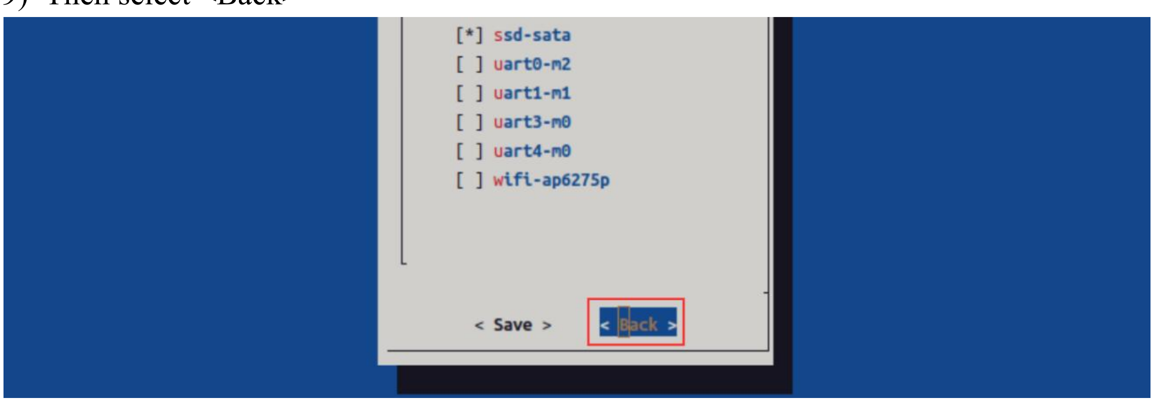

7) Then use the arrow keys of the keyboard to navigate to **ssd-sata**, and then use the space to select



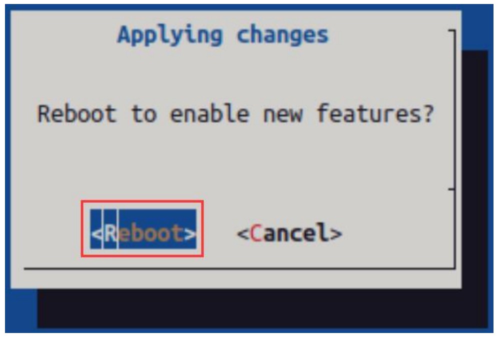

8) Then select <Save> to save




9) Then select <Back>




10) Then select <Reboot> to restart the system to make the configuration take effect




**The above settings will eventually add the configuration of overlays=ssd-sata to /boot/orangepiEnv.txt. After setting, you can check it first. If this configuration does not exist, then there is a problem with the settings.**

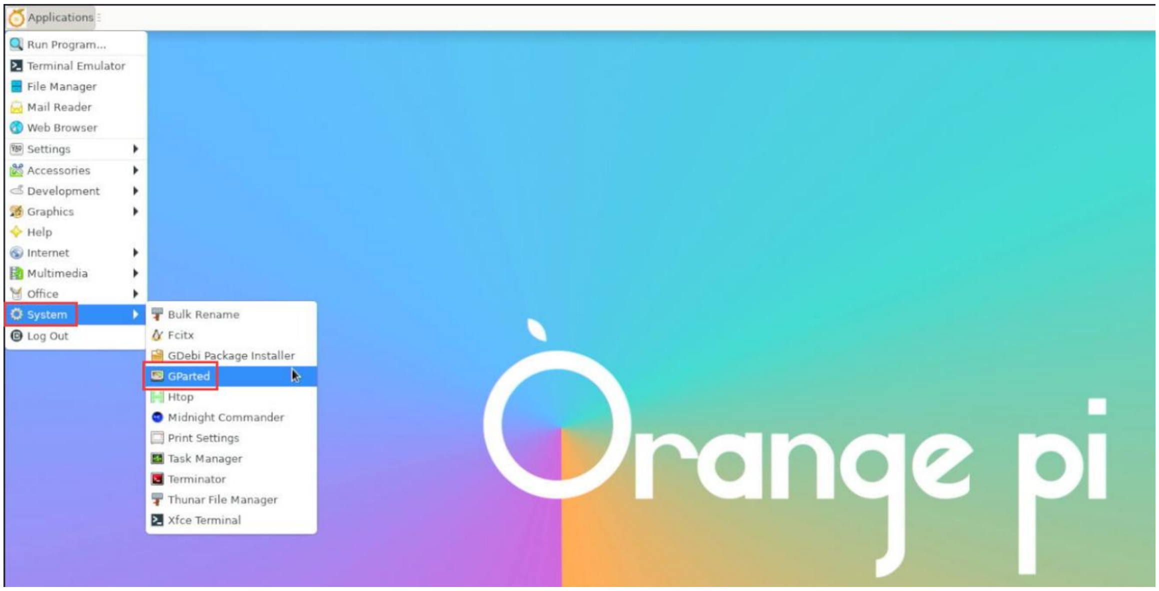
**If you find it troublesome to use orangepi-config, you can also open /boot/orangepiEnv.txt, and then add the configuration of overlays=ssd-sata.**

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "ssd"**
**overlays=ssd-sata**

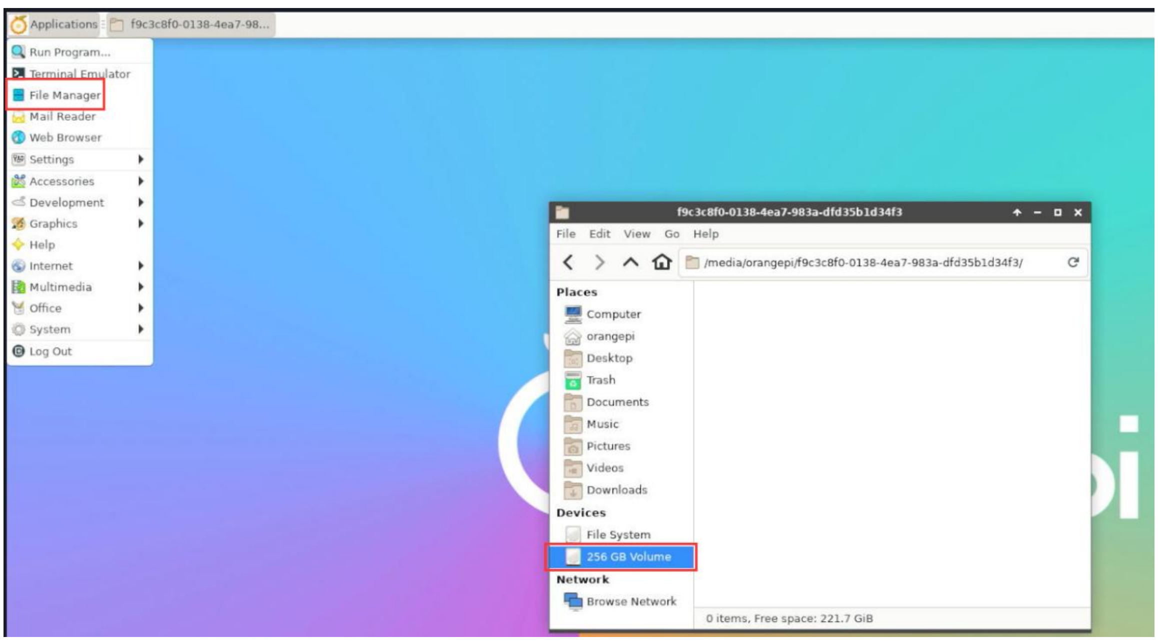
11) If everything is normal, after the system restarts, use the **sudo fdisk -l** command to see sata ssd information

```
orangepi@orangepi:~$ sudo fdisk -l
.......
Disk /dev/sda: 238.47 GiB, 256060514304 bytes, 500118192 sectors
Disk model: Fanxiang S201 25
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 43FFB292-340D-654C-8C30-6C64AEDAA0F4

Device       Start       End   Sectors   Size Type
/dev/sda1     2048 500117503 500115456 238.5G Linux filesystem
.......
```

12) Then use **GParted** to format or partition sata ssd




13) Then you can see the sata ssd device in the file management




14) In the server version system, you can use the mount command to mount the sata ssd to the required directory

```
orangepi@orangepi:~$ sudo mount /dev/sda1 /mnt
orangepi@orangepi:~$ df -h
Filesystem          Size   Used Avail Use% Mounted on
udev                 3.8G   8.0K   3.8G     1% /dev
```

```
tmpfs              769M  1.4M  768M   1% /run
/dev/mmcblk1p2      29G  5.9G   23G  21% /
tmpfs              3.8G     0  3.8G   0% /dev/shm
tmpfs              5.0M  4.0K  5.0M   1% /run/lock
tmpfs              3.8G   16K  3.8G   1% /tmp
/dev/mmcblk1p1     256M   90M  166M  36% /boot
/dev/zram1         194M   27M  154M  15% /var/log
tmpfs              769M   60K  769M   1% /run/user/1000
/dev/sda1          234G   28K  222G   1% /mnt
```

## 3. 15. Temperature sensor

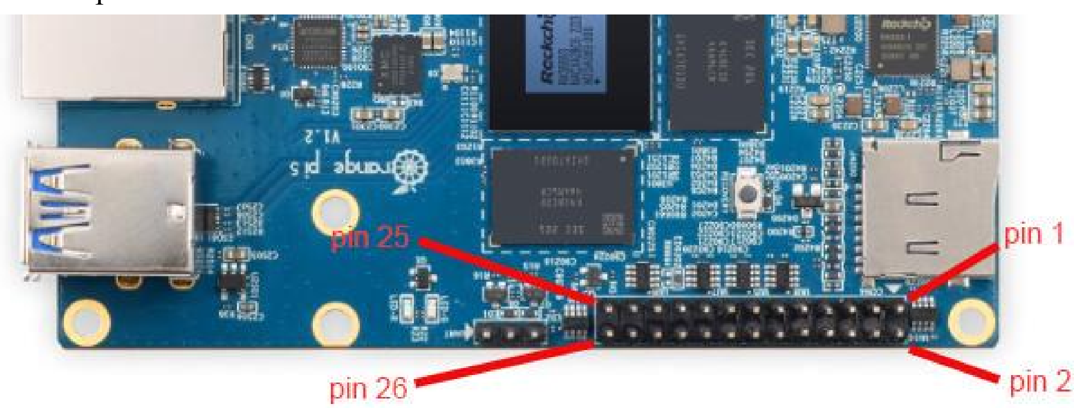
The command to view the system temperature sensor is:

```
orangepi@orangepi:~$ sensors
gpu_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

littlecore_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

bigcore0_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C

tcpm_source_psy_6_0022-i2c-6-22
Adapter: rk3x-i2c
in0:           0.00 V  (min =  +0.00 V, max =  +0.00 V)
curr1:         0.00 A  (max =  +0.00 A)

npu_thermal-virtual-0
Adapter: Virtual device
temp1:        +47.2°C
```

```
center_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.2°C

bigcore1_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.2°C

soc_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.2°C   (crit = +115.0°C)
```

## 3. 16. 26 Pin Interface Pin Description

1) Please refer to the figure below for the order of the 26 pin interface pins on the Orange Pi 5 development board




2) The functions of the 26 pin interface pins on the Orange Pi 5 development board are shown in the table below

a. The following is the complete pin diagram of 26pin

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 | 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3.3V | | 1 | 2 | | 5V | | | |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 | 4 | | 5V | | | |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 | 6 | | GND | | | |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 | 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| | | | GND | | 9 | 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 | 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 | 14 | | GND | | | |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 | 16 | 59 | GPIO1_D3 | UART4_RX_M0（feb70000） | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| | | | 3.3V | | 17 | 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| I2C3_SCL_M0 | UART3_TX_M0（feb60000） | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 | 20 | | GND | | | |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 | 22 | 92 | GPIO2_D4 | | | |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 | 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| | | | GND | | 25 | 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

b. The table below is the picture of the left half of the complete table above, so you can see it clearly

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 |
|---|---|---|---|---|---|
| | | | 3.3V | | 1 |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 |
| | | | GND | | 9 |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 |
| | | | 3.3V | | 17 |
| I2C3_SCL_M0 | UART3_TX_M0（feb60000） | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 |
| | | | GND | | 25 |

c. The table below is the picture of the right half of the complete table above, so you can see it clearly

| 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|
| 2 | | 5V | | | |
| 4 | | 5V | | | |
| 6 | | GND | | | |
| 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| 14 | | GND | | | |
| 16 | 59 | GPIO1_D3 | UART4_RX_M0（feb70000） | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| 20 | | GND | | | |
| 22 | 92 | GPIO2_D4 | | | |
| 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

**The pwm in the above table has marked the base address of the corresponding register, which is useful when checking which pwmchip in /sys/class/pwm/ corresponds to which pwm pin in the 26pin header.**

3) There are a total of 17 GPIO ports in the 26pin interface, and the voltage of all GPIO ports is **3.3v**

## 3. 17. How to install wiringOP

**Note that wiringOP has been pre-installed in the linux image released by Orange Pi. Unless the code of wiringOP is updated, there is no need to re-download, compile and install, just use it directly.**

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

**orangepi-build/external/cache/debs/arm64/wiringpi_2.46.deb**

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

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | ALT1   | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
root@orangepi5:~#
```

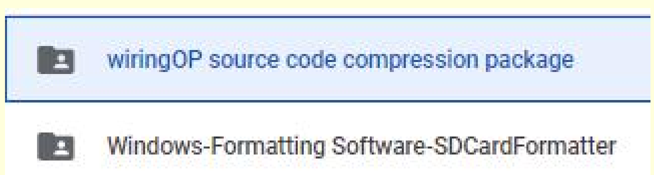
**wiringOP is currently mainly adapted to the functions of setting GPIO input and output, and setting high and low levels. Features like hardware PWM are not available.**

1) Download the code of wiringOP

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y git
orangepi@orangepi:~$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

**Note that Orange Pi 5 needs to download the code of the wiringOP next branch, please don't miss the -b next parameter.**

**If you have problems downloading the code from GitHub, you can download the source code compression package of wiringOP-OPI5.tar.gz from the official tool on the Orange Pi 5 data download page.**



2) Compile and install wiringOP

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

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

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

3) Test the output of the gpio readall command as follows

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | ALT1   | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
root@orangepi5:~#
```

## 3. 18. 26pin interface GPIO, I2C, UART, SPI and PWM test

**Note, if you need to set overlays to open multiple configurations at the same time, please use spaces to separate them and write them on one line as follows.**

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

**overlays=i2c1-m2 lcd1 ov13850-c1 pwm13-m2 spi4-m0-cs1-spidev uart0-m2**

### 3. 18. 1. 26pin GPIO port test

**The linux system released by Orange Pi has a pre-installed blink_all_gpio program, which will set all 17 GPIO ports in the 26pin to switch between high and low levels continuously.**

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

**The way to run the blink_all_gpio program is as follows:**

```
orangepi@orangepi5:~$ sudo blink_all_gpio         #Remember to add sudo permission
[sudo] password for orangepi:                     #A password is required here
```

1) The following takes pin No. 7—the corresponding GPIO is GPIO1_C6—the corresponding wPi number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
```

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

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

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

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

Use gpio readall to see that the value (V) of pin 7 has changed to 0

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |    OUT | 0 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
```

4) Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

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

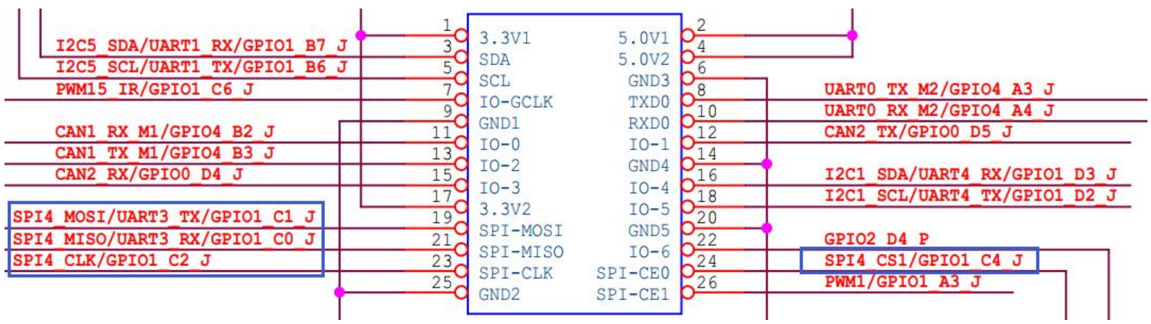
Use gpio readall to see that the value (V) of pin 7 has changed to 1

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |    OUT | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
```

5) The setting method of other pins is similar, just modify the serial number of wPi to the corresponding serial number of the pin

### 3. 18. 2. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the spi available for Orange Pi 5 is spi4




**In the Linux system, the spi4 in the 26pin is closed by default, and it needs to be opened manually to use it.**

**Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open spi4.**

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**
**overlays=spi4-m0-cs1-spidev**

2) First check whether there is a spidev4.1 device node in the linux system. If it exists, it means that SPI4 has been set up and can be used directly

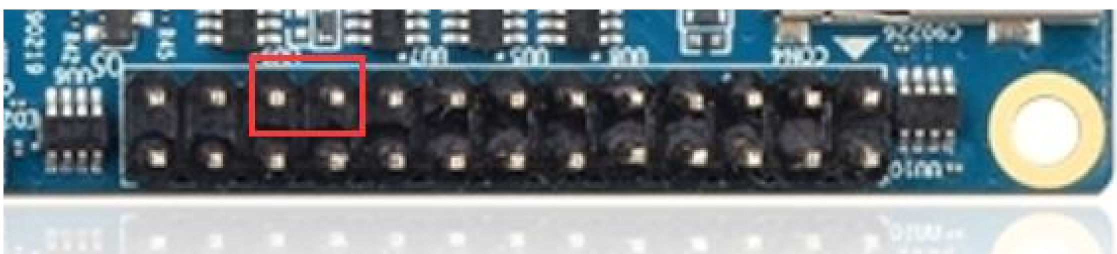
```
orangepi@orangepi:~$ ls /dev/spidev4.1
/dev/spidev4.1
```

**Note that /dev/spidev4.0 cannot be used, please use /dev/spidev4.1, don't make a mistake.**

3) Do not short-circuit the mosi and miso pins of SPI4, the output result of running spidev_test is as follows, you can see that the data of TX and RX are inconsistent

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒..................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF  | ................................
```

4) Then short-circuit the two pins of mosi (the 19th pin in the 26pin interface) and miso (the 21st pin in the 26pin interface) of SPI4, and then run the output of spidev_test as follows, you can see the sending and receiving same data



```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev4.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒..................▒.
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒..................▒.
```

### 3. 18. 3. 26pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi 5 is i2c1, i2c3 and i2c5, a total of three sets of i2c buses

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 | 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3.3V | | 1 | 2 | | 5V | | | |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 | 4 | | 5V | | | |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 | 6 | | GND | | | |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 | 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| | | | GND | | 9 | 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 | 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 | 14 | | GND | | | |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 | 16 | 59 | GPIO1_D3 | UART4_RX_M0 | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| | | | 3.3V | | 17 | 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| I2C3_SCL_M0 | UART3_TX_M0 | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 | 20 | | GND | | | |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 | 22 | 92 | GPIO2_D4 | | | |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 | 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| | | | GND | | 25 | 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

**As can be seen from the above table, i2c1 can be derived from pins 12 and 15 of the 26pin (i2c1_m2), or from pins 16 and 18 of the 26pin (i2c1_m4), please follow your own needs Just select a group. Please don't think that these are two different sets of i2c buses.**

**In the linux system, the i2c in the 26pin is turned off by default, and it needs to be turned on manually before it can be used.**

**Add the following configuration in red font to /boot/orangepiEnv.txt, and then restart the Linux system to open i2c1, i2c3 and i2c5 at the same time. If you only need to open one, then just fill in one.**

**Select the settings for i2c1_m2 as shown below:**

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

**overlays=i2c1-m2 i2c3-m0 i2c5-m3**

**Select the settings for i2c1_m4 as shown below:**

orangepi@orangepi:~$ **sudo vim /boot/orangepiEnv.txt**

**overlays=i2c1-m4 i2c3-m0 i2c5-m3**

2) After starting the linux system, first confirm that there is an i2c device node under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-*
/dev/i2c-0   /dev/i2c-10   /dev/i2c-3   /dev/i2c-6   /dev/i2c-9
/dev/i2c-1   /dev/i2c-2    /dev/i2c-5   /dev/i2c-7
```

3) Then connect an i2c device to the i2c pin of the 26pin connector

| | i2c1-m2 | i2c1-m4 | i2c3-m0 | i2c5-m3 |
|---|---|---|---|---|
| Sda pin | Corresponding to pin 12 | Corresponding to pin 16 | Corresponding to pin 21 | Corresponding to pin 3 |
| Sck pin | Corresponding to pin 15 | Corresponding to pin 18 | Corresponding to pin 19 | Corresponding to pin 5 |
| Vcc pin | Corresponding to pin 1 | Corresponding to pin 1 | Corresponding to pin 1 | Corresponding to pin 1 |
| Gnd pin | Corresponding to pin 6 | Corresponding to pin 6 | Corresponding to pin 6 | Corresponding to pin 6 |

4) Then use the i2cdetect -y command, if the address of the connected i2c device can be detected, it means that i2c can be used normally

```
orangepi@orangepi:~$ sudo i2cdetect -y 1        #i2c1's command
orangepi@orangepi:~$ sudo i2cdetect -y 3        #i2c3 command
orangepi@orangepi:~$ sudo i2cdetect -y 5        #i2c5 command
```

```
orangepi@orangepi5:~$ sudo i2cdetect -y 5
[sudo] password for orangepi:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
orangepi@orangepi5:~$
```

### 3. 18. 4. 26pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5 are uart0, uart1, uart3 and uart4, a total of four sets of uart buses

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 | 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3.3V | | 1 | 2 | | 5V | | | |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 | 4 | | 5V | | | |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 | 6 | | GND | | | |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 | 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| | | | GND | | 9 | 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 | 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 | 14 | | GND | | | |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 | 16 | 59 | GPIO1_D3 | UART4_RX_M0（feb70000） | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| | | | 3.3V | | 17 | 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| I2C3_SCL_M0 | UART3_TX_M0（feb60000） | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 | 20 | | GND | | | |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 | 22 | 92 | GPIO2_D4 | | | |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 | 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| | | | GND | | 25 | 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

**In the Linux system, the uart in the 26pin is closed by default, and it needs to be opened manually before it can be used.**

**Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open uart0, uart1, uart3 and uart4 at the same time. If you only need to open one, then fill in one.**

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=uart0-m2    uart1-m1    uart3-m0    uart4-m0
```

2) After entering the linux system, first confirm whether there is a device node corresponding to uart under /dev

```
orangepi@orangepi:~$ ls /dev/ttyS*
/dev/ttyS0    /dev/ttyS1    /dev/ttyS3    /dev/ttyS4    /dev/ttyS9
```

3) Then start to test the uart interface, first use the DuPont line to short the rx and tx of the uart interface to be tested

| | uart0 | uart1 | uart3 | uart4 |
|---|---|---|---|---|
| Tx pin | Corresponding to pin 8 | Corresponding to pin 5 | Corresponding to pin 19 | Corresponding to pin 18 |
| Rx pin | Corresponding to pin 10 | Corresponding to pin 3 | Corresponding to pin 21 | Corresponding to pin 16 |


4) Use the gpio serial command to test the loopback function of the serial port as shown below. If you can see the following print, it means that the serial port communication is normal

a. Test UART0

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS0
[sudo] password for orangepi:    #enter password here
```

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

b. Test UART1

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS1
[sudo] password for orangepi:    #enter password here

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

c. Test UART3

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS3
[sudo] password for orangepi:    #enter password here

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

d. Test UART4

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttyS4
[sudo] password for orangepi:    #enter password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
```

```
Out:     5:    ->     5^C
```

### 3. 18. 5. PWM test method

1) As can be seen from the table below, the available pwm for Orange Pi 5 includes pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15, a total of six pwm

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 | 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3.3V | | 1 | 2 | | 5V | | | |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 | 4 | | 5V | | | |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 | 6 | | GND | | | |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 | 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| | | | GND | | 9 | 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 | 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 | 14 | | GND | | | |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 | 16 | 59 | GPIO1_D3 | UART4_RX_M0（feb70000） | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| | | | 3.3V | | 17 | 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| I2C3_SCL_M0 | UART3_TX_M0（feb60000） | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 | 20 | | GND | | | |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 | 22 | 92 | GPIO2_D4 | | | |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 | 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| | | | GND | | 25 | 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

**As can be seen from the table above:**

**pwm1 can be derived from pin 16 of 26pin (pwm1_m1), or from pin 26 of 26pin (pwm1_m2)**

**pwm3 can be derived from pin 15 of 26pin (pwm3_m0), or from pin 23 of 26pin (pwm3_m2)**

**Please select the corresponding pin according to your needs. Please don't think that these are two different pwm buses.**

**In the linux system, the pwm in the 26pin is closed by default, and it needs to be opened manually to use it.**

**Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open pwm0, pwm13, pwm14 and pwm15 at the same time. If you only need to open one, then fill in one.**

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=pwm0-m1 pwm13-m2 pwm14-m1 pwm15-m2
```

**Select the settings of pwm1_m1 as follows, please do not open pwm1-m1 and pwm1-m2 at the same time:**

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=pwm1-m1
```

**The settings to select pwm1_m2 are as follows:**

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
```

**overlays=pwm1-m2**

**Select the settings of pwm3_m0 as follows, please do not open pwm3-m0 and pwm3-m2 at the same time:**

**orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt**

**overlays=pwm3-m0**

**The settings to select pwm3_m2 are as follows:**

**orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt**

**overlays=pwm3-m2**

2) When a pwm is turned on, there will be an extra pwmchipX in /sys/class/pwm/ (X is a specific number), for example, after turning on pwm15, check the pwmchipX under /sys/class/pwm/ two became three

```
orangepi@orangepi:~$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

3) Which pwmchip above corresponds to pwm15, let's check the output of the ls /sys/class/pwm/ -l command first, as shown below:

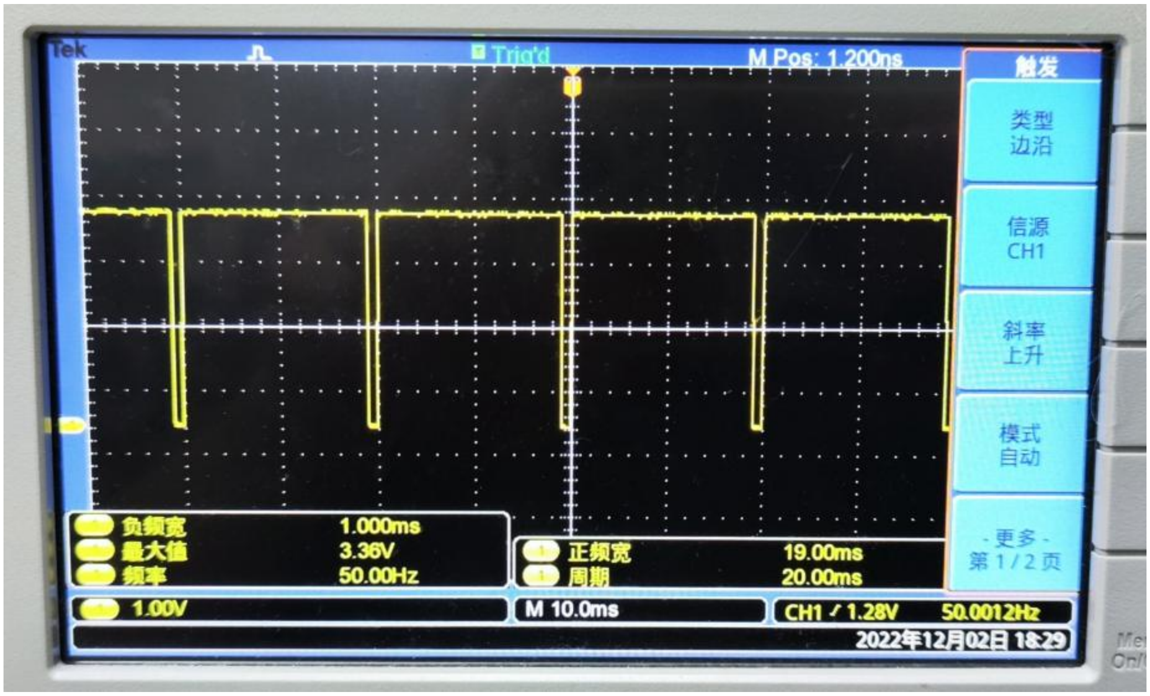
```
orangepi@orangepi5:~$ ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Dec  2 10:20 pwmchip0 -> ../../devices/platform/fd8b0020.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Dec  2 10:20 pwmchip1 -> ../../devices/platform/febd0020.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Dec  2 10:20 pwmchip2 -> ../../devices/platform/febf0030.pwm/pwm/pwmchip2
orangepi@orangepi5:~$
```

4) Then it can be known from the table below that the base address of the pwm15 register is febf0030, and then look at the output of the ls /sys/class/pwm/ -l command, you can see that pwmchip2 is linked to febf0030.pwm, so pwm15 corresponds to pwmchip as pwmchip2

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 | 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3.3V | | 1 | 2 | | 5V | | | |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 | 4 | | 5V | | | |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 | 6 | | GND | | | |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 | 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| | | | GND | | 9 | 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 | 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 | 14 | | GND | | | |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 | 16 | 59 | GPIO1_D3 | UART4_RX_M0（feb70000） | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| | | | 3.3V | | 17 | 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| I2C3_SCL_M0 | UART3_TX_M0（feb60000） | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 | 20 | | GND | | | |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 | 22 | 92 | GPIO2_D4 | | | |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 | 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| | | | GND | | 25 | 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

5) Then use the following command to make pwm15 output a 50Hz square wave (please switch to the root user first, and then execute the following command)

```
root@orangepi:~# echo 0 > /sys/class/pwm/pwmchip2/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip2/pwm0/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip2/pwm0/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip2/pwm0/enable
```




6) For the pwm15 demonstrated above, other pwm test methods are similar.

## 3. 19. How to install and use wiringOP-Python

**wiringOP-Python is the Python language version of wiringOP, which is used to operate hardware resources such as GPIO, I2C, SPI and UART of the development board in Python programs.**

**In addition, please note that all the following commands are operated under the root user.**

### 3. 19. 1. How to install wiringOP-Python

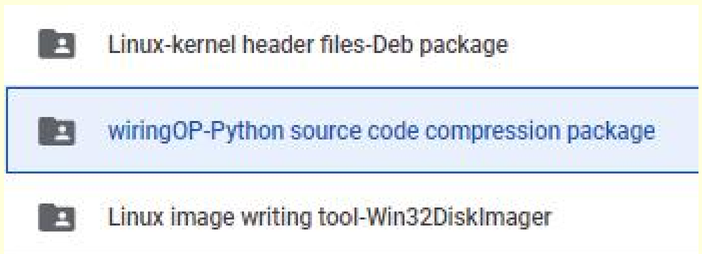
1) First install the dependency package

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python

**Note that the following git clone --recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on wiringOP. Please make sure that the download process does not report errors due to network problems.**

**If the code cannot be downloaded, please go to the official tool to download the source code compression package.**




```
root@orangepi:~# git clone --recursive https://github.com/orangepi-xunlong/wiringOP-Python -b next

Cloning into 'wiringOP-Python'...
remote: Enumerating objects: 602, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 602 (delta 20), reused 26 (delta 12), pack-reused 562
Receiving objects: 100% (602/602), 309.30 KiB | 1.23 MiB/s, done.
Resolving deltas: 100% (349/349), done.
Submodule 'wiringOP' (https://github.com/orangepi-xunlong/wiringOP.git) registered for path 'wiringOP'
Cloning into '/root/test/wiringOP-Python/wiringOP'...
remote: Enumerating objects: 654, done.
remote: Counting objects: 100% (273/273), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 654 (delta 244), reused 245 (delta 238), pack-reused 381
Receiving objects: 100% (654/654), 360.54 KiB | 1.73 MiB/s, done.
Resolving deltas: 100% (424/424), done.
Submodule path 'wiringOP': checked out '85f1331cd8fda668115461ec1c06cb342057eb03'
```

3) Then use the following command to compile wiringOP-Python and install it into the Linux system of the development board

```
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# python3 generate-bindings.py > bindings.i
root@orangepi:~/wiringOP-Python# sudo python3 setup.py install
```

4) Then enter the following command, if there is help information output, it means that wiringOP-Python is installed successfully, press the q key to exit the help information interface

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi;    help(wiringpi)"
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.
```

5) The steps to test whether wiringOP-Python is successfully installed under the python command line are as follows：

a. First use the python3 command to enter the command line mode of python3

```
root@orangepi:~# python3
```

b. Then import the python module of wiringpi

```
>>> import wiringpi;
```

c. Finally, enter the following command to view the help information of wiringOP-Python, and press the q key to exit the help information interface

```
>>> help(wiringpi)
Help on module wiringpi:

NAME
    wiringpi

DESCRIPTION
```

```
    # This file was automatically generated by SWIG (http://www.swig.org).
    # Version 4.0.2
    #
    # Do not make changes to this file unless you know what you are doing--modify
    # the SWIG interface file instead.

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

    class GPIO(builtins.object)
     |   GPIO(pinmode=0)
     |


>>>
```

### 3. 19. 2. 26pin GPIO port test

**wiringOP-Python is the same as wiringOP, you can also determine which GPIO pin to operate by specifying the wPi number, because there is no command to check the wPi number in wiringOP-Python, so you can only check the wPi number of the board through the gpio command in wiringOP. Correspondence between pins.**

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | IN     | SCL.1    | 10  | 58   |
 |   49 |  11 | SPI4_TXD |     IN | 1 | 19 || 20 |   |        | GND      |     |      |
 |   48 |  12 | SPI4_RXD |     IN | 1 | 21 || 22 | 1 | IN     | GPIO2_D4 | 13  | 92   |
 |   50 |  14 | SPI4_CLK |     IN | 1 | 23 || 24 | 1 | ALT1   | SPI4_CS1 | 15  | 52   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | PWM1     | 16  | 35   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
root@orangepi5:~#
```

1) The following takes pin No. 7—the corresponding GPIO is GPIO1_C6—the corresponding wPi number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepi5:~# gpio readall
 +------+-----+----------+--------+---+   OPI5   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   54 |   2 |    PWM15 |     IN | 1 |  7 || 8  | 0 | ALT6   | RXD.0    | 3   | 131  |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | TXD.0    | 4   | 132  |
 |  138 |   5 |  CAN1_RX |     IN | 1 | 11 || 12 | 1 | IN     | CAN2_TX  | 6   | 29   |
 |  139 |   7 |  CAN1_TX |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   28 |   8 |  CAN2_RX |     IN | 1 | 15 || 16 | 1 | IN     | SDA.1    | 9   | 59   |
```

2) The steps to test directly with the command are as follows:

   a. First set the GPIO port to the output mode, where the first parameter of the pinMode function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
from wiringpi import GPIO; wiringpi.wiringPiSetup() ; \
wiringpi.pinMode(2, GPIO.OUTPUT) ;  "
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.LOW)"
```

   c. Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.HIGH)"
```

3) The steps to test in the command line of python3 are as follows:

   a. First use the python3 command to enter the command line mode of python3

```
root@orangepi:~# python3
```

   b. Then import the python module of wiringpi

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

   c. Then set the GPIO port to output mode, where the first parameter of the pinMode function is the serial number of the wPi corresponding to the pin, and the second parameter is the GPIO mode

```
>>> wiringpi.wiringPiSetup()
0
>>> wiringpi.pinMode(2, GPIO.OUTPUT)
```

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

```
>>> wiringpi.digitalWrite(2, GPIO.LOW)
```

e. Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

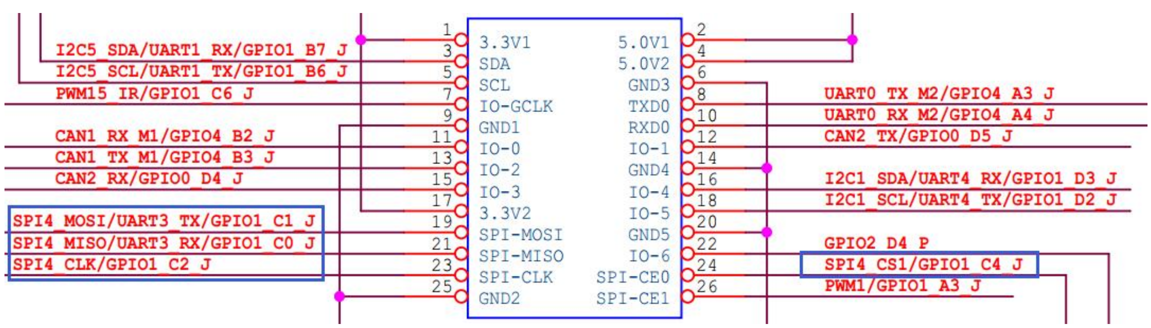
```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

4) The method of wiringOP-Python to set GPIO high and low levels in python code can refer to the blink.py test program in the examples below. The blink.py test program will set the voltage of all GPIO ports in the 26 pins of the development board to change continuously.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# ls blink.py
blink.py
root@orangepi:~/wiringOP-Python/examples# python3 blink.py
```

### 3. 19. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the spi available for Orange Pi 5 is spi4




**In the Linux system, the spi4 in the 26pin is closed by default, and it needs to be opened manually to use it.**

**Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open spi4.**

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=spi4-m0-cs1-spidev
```

2) First check whether there is a spidev4.1 device node in the linux system. If it exists, it means that SPI4 has been set up and can be used directly

```
orangepi@orangepi:~$ ls /dev/spidev4.1
/dev/spidev4.1
```

**Note that /dev/spidev4.0 cannot be used, please use /dev/spidev4.1, don't make a mistake.**

3) Then you can use the spidev_test.py program in the examples to test the loopback function of the SPI. The spidev_test.py program needs to specify the following two parameters:

a.--channel: Specify the channel number of SPI

b.--port: specify the port number of SPI

4) Do not short-circuit the mosi and miso pins of SPI4, the output result of running spidev_test.py is as follows, you can see that the data of TX and RX are inconsistent

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 4 --port 1
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev4.1
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@..........|
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF    |................|
```

5) Then use the Dupont wire to short-circuit the two pins of txd (pin 19 in the 26pin

interface) and rxd (pin 21 in the 26pin interface) of SPI1 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 SPI4 loopback test is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 1 --port 0
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev4.1
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  |......@..........|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  |......@..........|
```

### 3. 19. 4. 26pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi 5 is i2c1, i2c3 and i2c5, a total of three sets of i2c buses

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 | 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3.3V | | 1 | 2 | | 5V | | | |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 | 4 | | 5V | | | |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 | 6 | | GND | | | |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 | 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| | | | GND | | 9 | 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 | 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 | 14 | | GND | | | |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 | 16 | 59 | GPIO1_D3 | UART4_RX_M0 | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| | | | 3.3V | | 17 | 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| I2C3_SCL_M0 | UART3_TX_M0 | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 | 20 | | GND | | | |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 | 22 | 92 | GPIO2_D4 | | | |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 | 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| | | | GND | | 25 | 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

**As can be seen from the above table, i2c1 can be derived from pins 12 and 15 of the 26pin (i2c1_m2), or from pins 16 and 18 of the 26pin (i2c1_m4), please follow your own needs Just select a group. Please don't think that these are two different sets of i2c buses.**

**In the linux system, the i2c in the 26pin is turned off by default, and it needs to be turned on manually before it can be used.**

**Add the following configuration in red font to /boot/orangepiEnv.txt, and then restart the Linux system to open i2c1, i2c3 and i2c5 at the same time. If you only need to open one, then just fill in one.**

**Select the settings for i2c1_m2 as shown below:**

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=i2c1-m2    i2c3-m0    i2c5-m3

Select the settings for i2c1_m4 as shown below:
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=i2c1-m4    i2c3-m0    i2c5-m3
```

2) After starting the linux system, first confirm that there is an i2c device node under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-*
/dev/i2c-0    /dev/i2c-10    /dev/i2c-3    /dev/i2c-6    /dev/i2c-9
/dev/i2c-1    /dev/i2c-2     /dev/i2c-5    /dev/i2c-7
```

3) Then connect an i2c device to the i2c pin of the 26pin connector, here we take the ds1307 RTC module as an example

| | i2c1-m2 | i2c1-m4 | i2c3-m0 | i2c5-m3 |
|---|---|---|---|---|
| Sda pin | Corresponding to pin 12 | Corresponding to pin 16 | Corresponding to pin 21 | Corresponding to pin 3 |
| Sck pin | Corresponding to pin 15 | Corresponding to pin 18 | Corresponding to pin 19 | Corresponding to pin 5 |
| Vcc pin | Corresponding to pin 1 | Corresponding to pin 1 | Corresponding to pin 1 | Corresponding to pin 1 |
| Gnd pin | Corresponding to pin 6 | Corresponding to pin 6 | Corresponding to pin 6 | Corresponding to pin 6 |


4) Then use the i2cdetect -y command, if the address of the connected i2c device can be detected, it means that i2c can be used normally

```
orangepi@orangepi:~$ sudo i2cdetect -y 1        #i2c1's command
orangepi@orangepi:~$ sudo i2cdetect -y 3        #i2c3 command
```

```
orangepi@orangepi:~$ sudo i2cdetect -y 5        #i2c5 command
```

```
orangepi@orangepi5:~$ sudo i2cdetect -y 5
[sudo] password for orangepi:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
orangepi@orangepi5:~$
```

5) Then you can run the ds1307.py test program in the examples to read the RTC time

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device \
"/dev/i2c-5"
Thu 2023-01-05    14:57:55
Thu 2023-01-05    14:57:56
Thu 2023-01-05    14:57:57
^C
exit
```

### 3. 19. 5. 5.26pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5 are uart0, uart1, uart3 and uart4, a total of four sets of uart buses

| 复用功能 | 复用功能 | 复用功能 | GPIO | GPIO序号 | 引脚序号 | 引脚序号 | GPIO序号 | GPIO | 复用功能 | 复用功能 | 复用功能 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3.3V | | 1 | 2 | | 5V | | | |
| PWM13_M2（febf0010） | UART1_RX_M1（feb40000） | I2C5_SDA_M3 | GPIO1_B7 | 47 | 3 | 4 | | 5V | | | |
| | UART1_TX_M1 | I2C5_SCL_M3 | GPIO1_B6 | 46 | 5 | 6 | | GND | | | |
| | | PWM15_IR_M2（febf0030） | GPIO1_C6 | 54 | 7 | 8 | 131 | GPIO4_A3 | UART0_TX_M2（fd890000） | | |
| | | | GND | | 9 | 10 | 132 | GPIO4_A4 | UART0_RX_M2 | | |
| | PWM14_M1（febf0020） | CAN1_RX_M1 | GPIO4_B2 | 138 | 11 | 12 | 29 | GPIO0_D5 | CAN2_TX_M1 | I2C1_SDA_M2 | |
| | | CAN1_TX_M1 | GPIO4_B3 | 139 | 13 | 14 | | GND | | | |
| PWM3_IR_M0（fd8b0030） | I2C1_SCL_M2 | CAN2_RX_M1 | GPIO0_D4 | 28 | 15 | 16 | 59 | GPIO1_D3 | UART4_RX_M0（feb70000） | I2C1_SDA_M4 | PWM1_M1（fd8b0010） |
| | | | 3.3V | | 17 | 18 | 58 | GPIO1_D2 | UART4_TX_M0 | I2C1_SCL_M4 | PWM0_M1（fd8b0000） |
| I2C3_SCL_M0 | UART3_TX_M0（feb60000） | SPI4_MOSI_M0 | GPIO1_C1 | 49 | 19 | 20 | | GND | | | |
| I2C3_SDA_M0 | UART3_RX_M0 | SPI4_MISO_M0 | GPIO1_C0 | 48 | 21 | 22 | 92 | GPIO2_D4 | | | |
| | PWM3_IR_M2（fd8b0030） | SPI4_CLK_M0 | GPIO1_C2 | 50 | 23 | 24 | 52 | GPIO1_C4 | SPI4_CS1_M0 | | |
| | | | GND | | 25 | 26 | 35 | GPIO1_A3 | PWM1_M2（fd8b0010） | | |

**In the Linux system, the uart in the 26pin is closed by default, and it needs to be opened manually before it can be used.**

**Add the following red font configuration in /boot/orangepiEnv.txt, and then restart the Linux system to open uart0, uart1, uart3 and uart4 at the same time. If you only**

**need to open one, then fill in one.**

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=uart0-m2   uart1-m1   uart3-m0   uart4-m0
```

2) After entering the linux system, first confirm whether there is a device node corresponding to uart under /dev

```
orangepi@orangepi:~$ ls /dev/ttyS*
/dev/ttyS0   /dev/ttyS1   /dev/ttyS3   /dev/ttyS4   /dev/ttyS9
```

3) Then start to test the uart interface, first use the DuPont line to short the rx and tx of the uart interface to be tested

| | uart0 | uart1 | uart3 | uart4 |
|---|---|---|---|---|
| Tx pin | Corresponding to pin 8 | Corresponding to pin 5 | Corresponding to pin 19 | Corresponding to pin 18 |
| Rx pin | Corresponding to pin 10 | Corresponding to pin 3 | Corresponding to pin 21 | Corresponding to pin 16 |


4) Use the serialTest.py program in the examples to test the loopback function of the serial port as shown below. If you can see the following print, it means that the serial port communication is normal

a. Test UART0

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS0"

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

```
exit
```

b. Test UART1

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS1"

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

c. Test UART3

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS3"

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

d. Test UART4

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device \
"/dev/ttyS4"

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

## 3. 20. Hardware watchdog test

The watchdog_test program is pre-installed in the linux system released by Orange Pi, which can be tested directly.

The method of running 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. 21. Check the serial number of the RK3588S chip

The commands of the RK3588S chip serial number are shown below. The serial number of each chip is different, so you can use the serial number to distinguish multiple development boards.

```
orangepi@orangepi:~$ cat_serial.sh
Serial              : 1404a7682e86830c
```

## 3. 22. Method of installing docker

1) The linux image provided by Orange PI has been pre -installed with Docker, but the Docker service is not opened by default

2) Use **enable_docker.sh** script to enable the docker service, and then you can start using the docker command, and the docker service will be automatically activated next time the system starts the system

```
orangepi@orangepi:~$ enable_docker.sh
```

3) Then you can use the following command to test the docker.If you can run hello-worm to indicate that docker can use it normally

```
orangepi@orangepi:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
256ab8fe8778: Pull complete
Digest:
sha256:7f0a9f93b4aa3022c3a4c147a449ef11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
.....
```

## 3. 23. The method of downloading and installing ARM64 version Balenaetcher

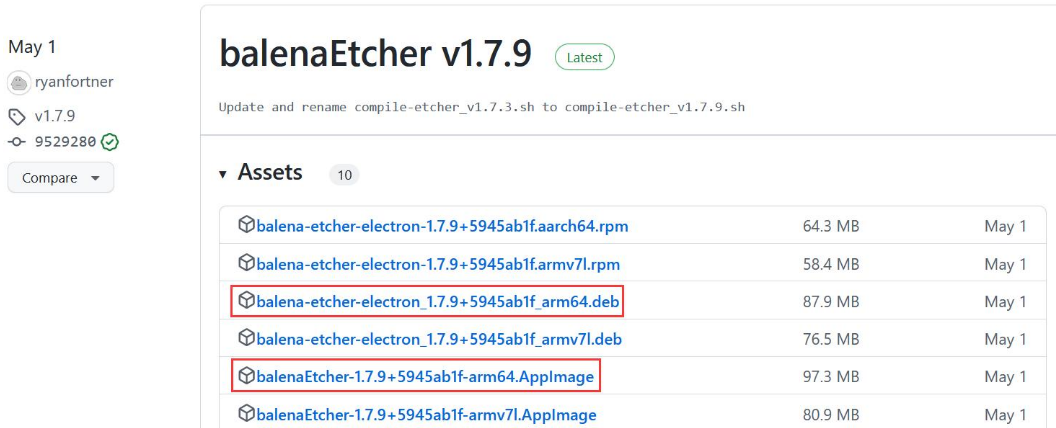
1) The download address of Balenaetcher ARM64 is as below

a. The download address of the .deb installation package is shown below, you need to install it to use

**https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balena-etcher-electron_1.7.9+5945ab1f_arm64.deb**

b. The download address of the Appimage version without installation is shown below：

**https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balenaEtcher-1.7.9+5945ab1f-arm64.AppImage**

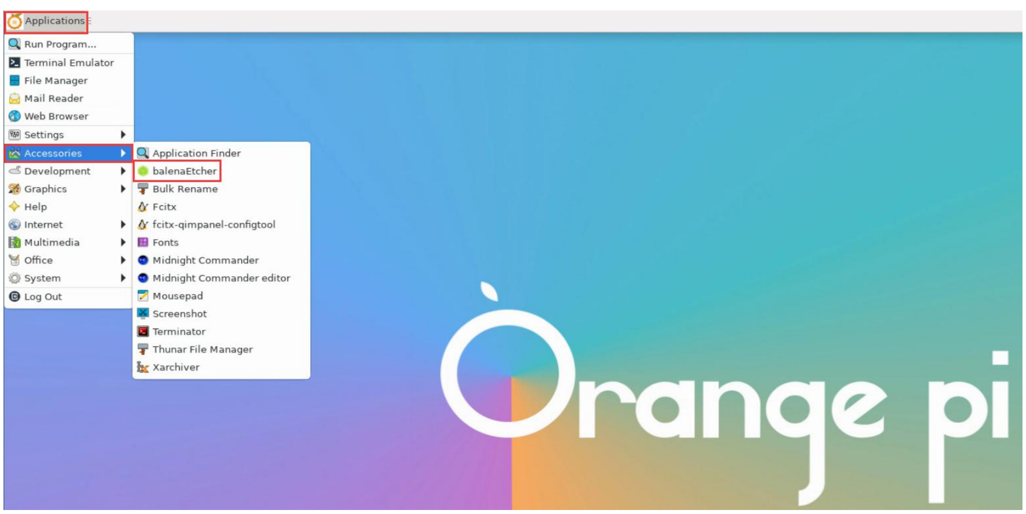



2) How to install and use the deb version of Balenaetcher：

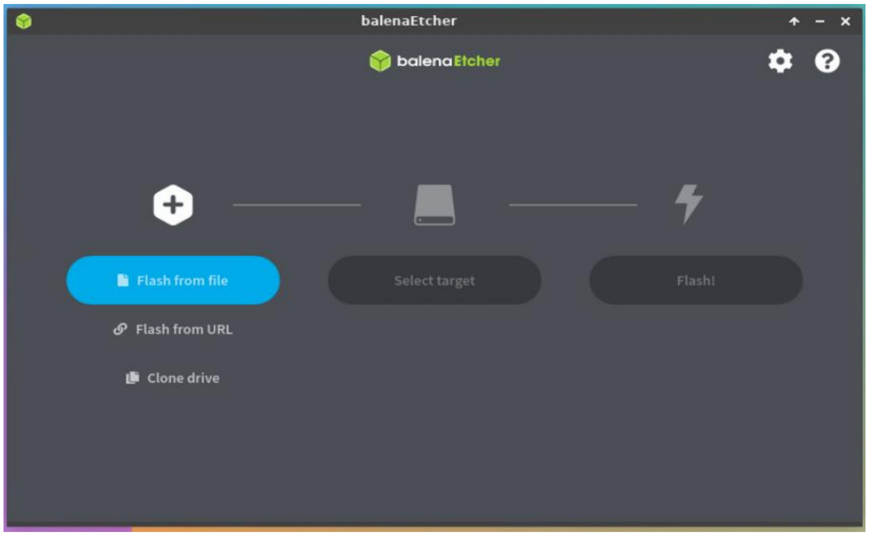
a. The deb version of Balenaetcher installation command is shown below:

```
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 the balenaetcher is opened is shown below：

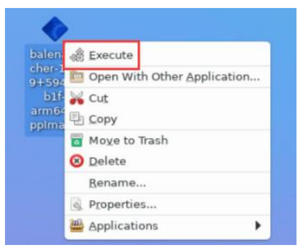



3) How to use the AppImage version of balenaetcher:

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. 24. The installation method of the Pagoda Linux panel

**Pagoda Linux panel is a server management software that improves operation and maintenance efficiency. It supports more than 100 server management functions such as one -click LAMP/LNMP/Cluster/Monitoring/Website/FTP/Database/Java (excerpted from the official website of the pagoda)**

1) The order of compatibility recommendation of the pagoda Linux system is

```
Debian11 > Ubuntu 22.04
```

2) Then enter the following command in the Linux system to start the installation of the pagoda

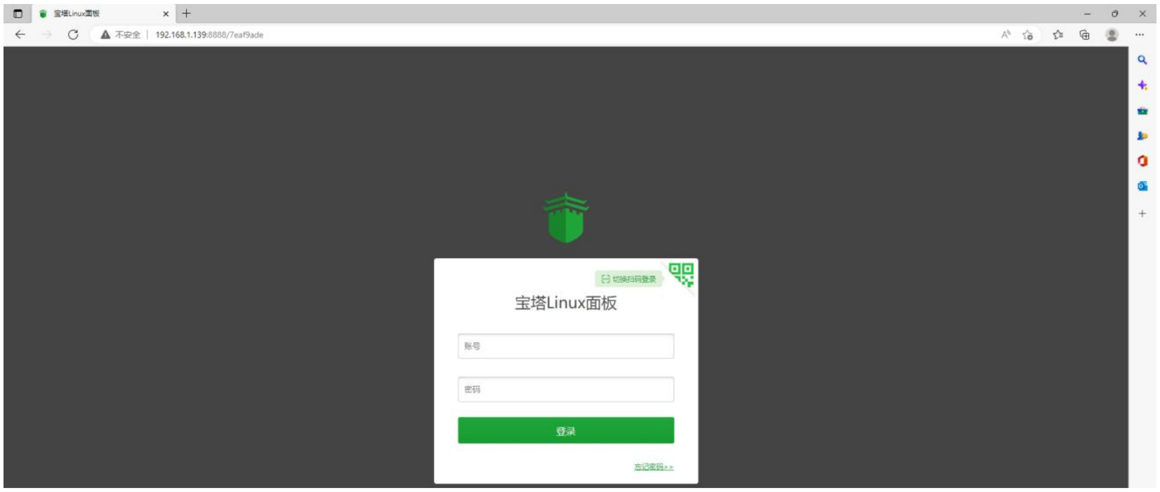
```
orangepi@orangepi:~$ sudo install_bt_panel.sh
```

3) Then the pagoda installation program reminds whether to install **Bt-Panel** to**/www** folder, and enter y at this time

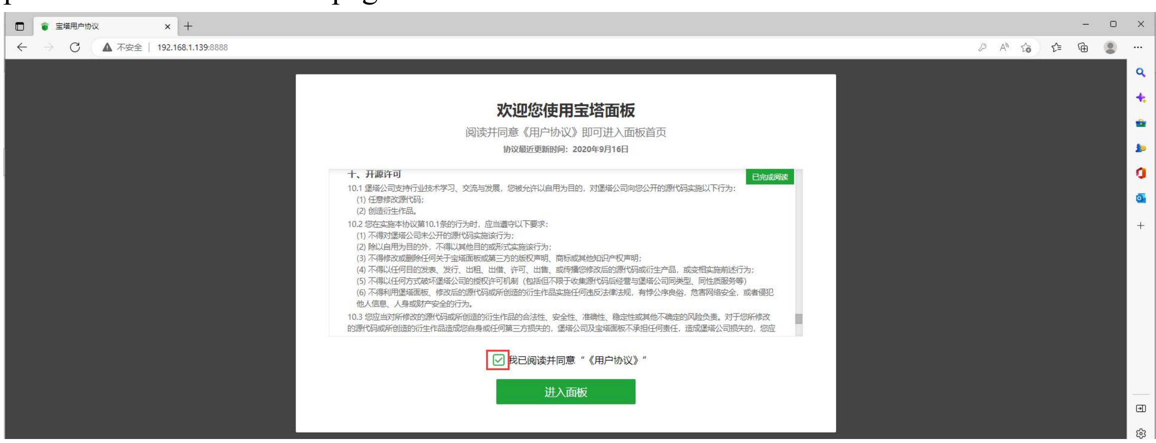
```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| 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
```

4) Then what to do is wait patiently. When you see the printing information below the terminal output, it means that the pagoda has been installed. The entire installation process takes about 12 minutes.There may be some differences according to the difference in network speed

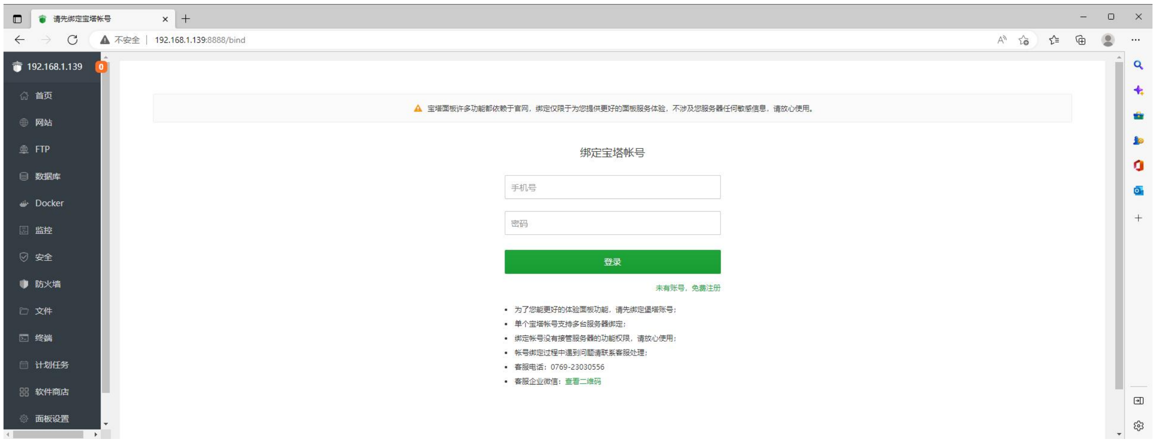

5) Enter the **panel address** displayed above in the browser to open the login interface of the pagoda Linux panel, and then enter the **username** and **password** displayed in the corresponding position.



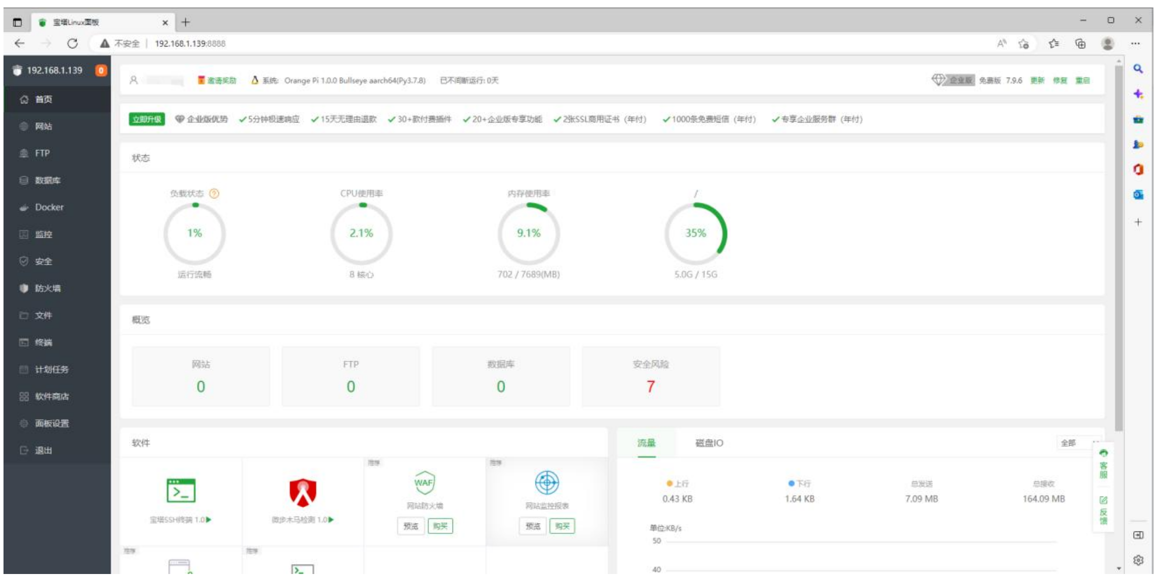

6) After successfully logging in to the pagoda, the following welcome interface will pop up. First of all, please take the intermediate user notice to read to the bottom, and then you can choose "I have agreed and read the" User Agreement ", and then click" Enter the panel " You can enter the pagoda




7) After entering the pagoda, you will first prompt the account that needs to be bound to the official website of the pagoda. If there is no account number, you can go to the official website of the pagoda (**https://www.bt.cn**) to register a account

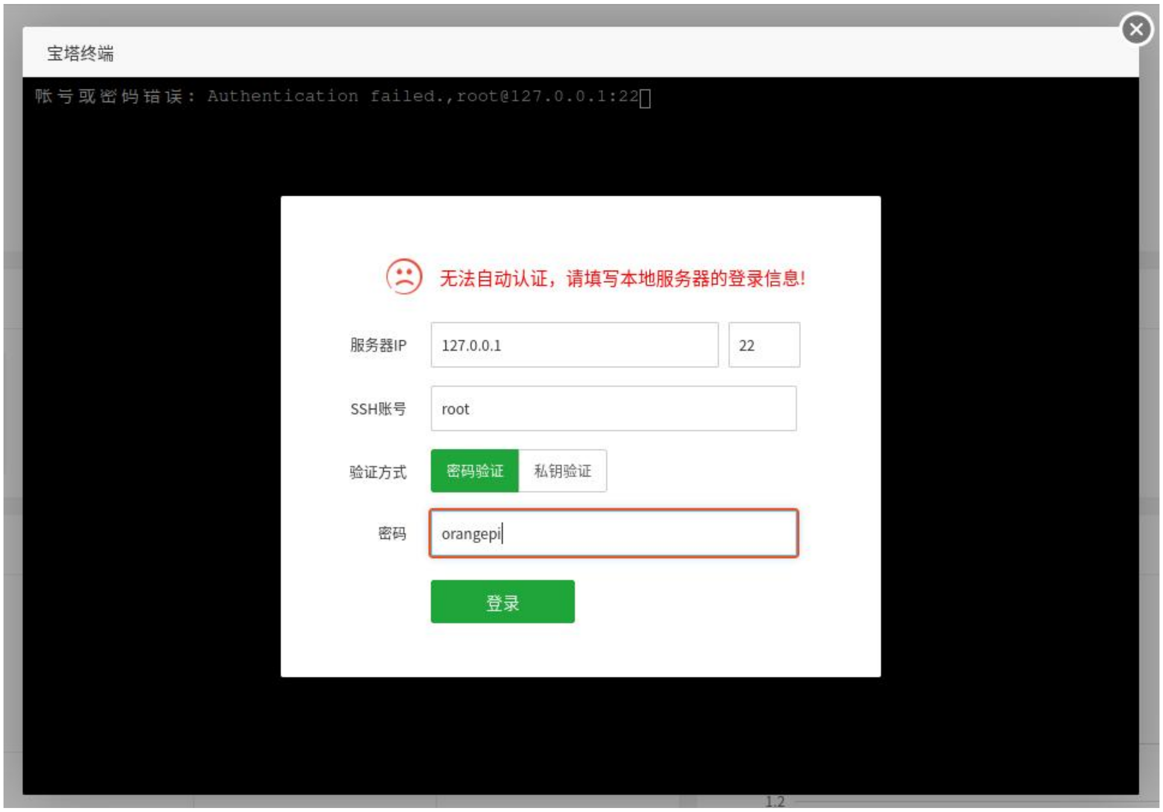



8) The final display interface is shown in the figure below. You can intuitively see some status information of the development board Linux system, such as load status, CPU usage, memory usage, and storage space usage.

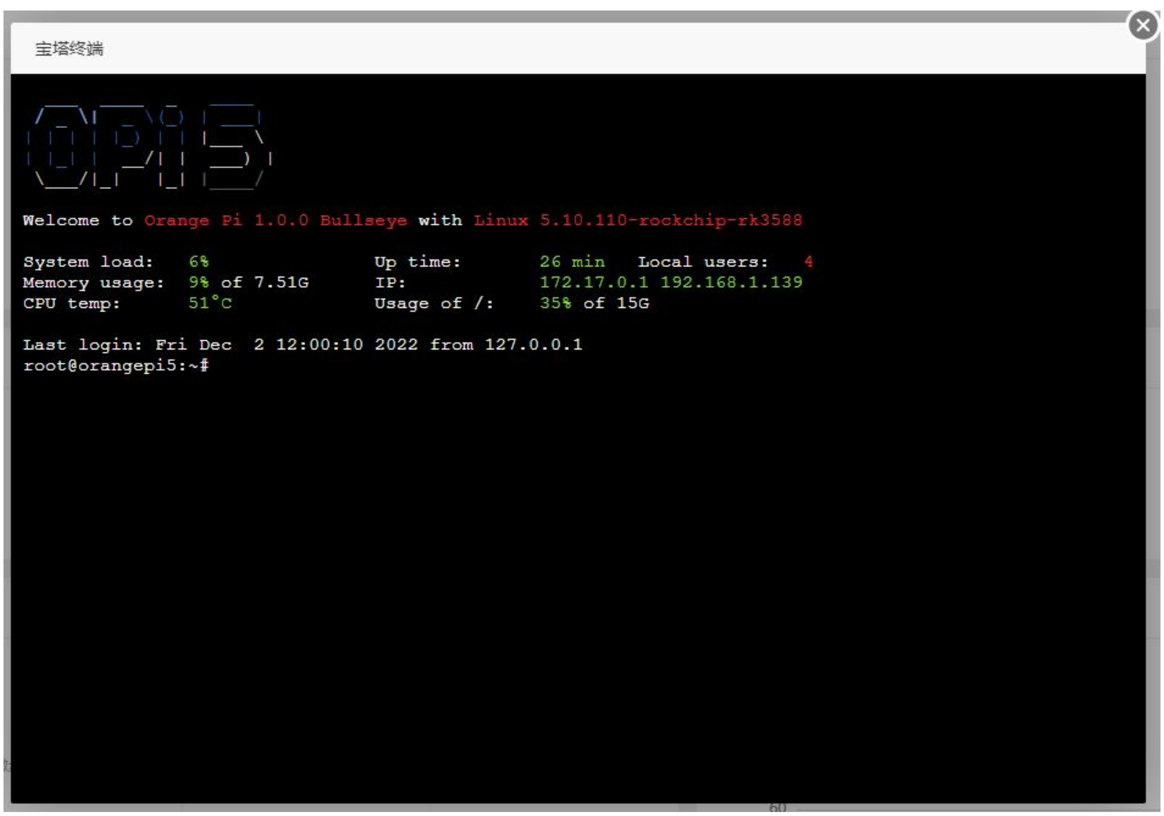



9) Test the SSH terminal login of the pagoda

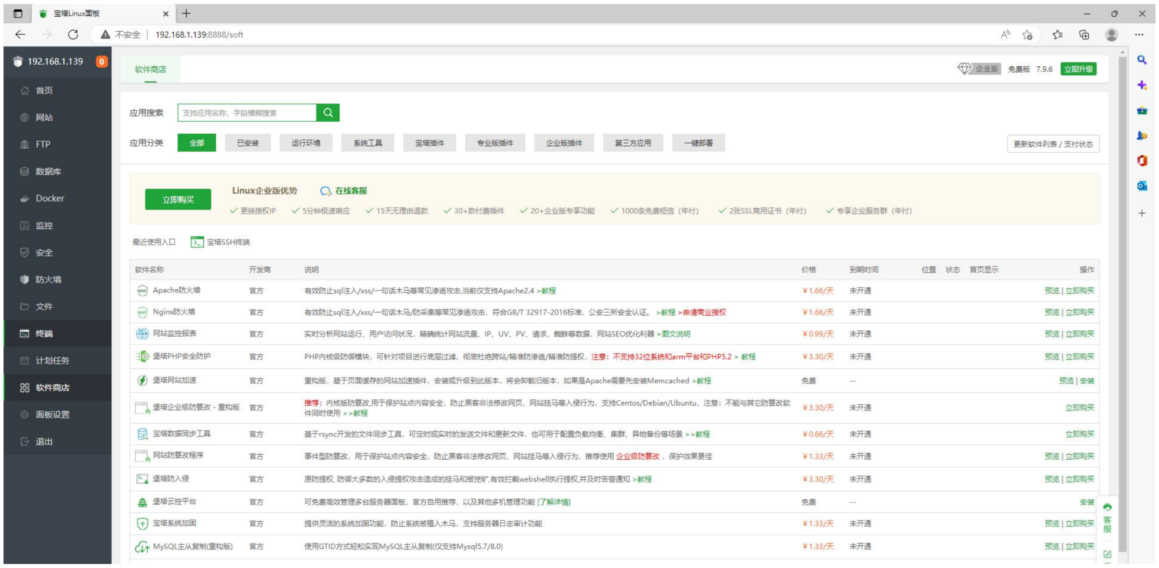
a. After opening the SSH terminal of the pagoda, you will first prompt the password of the development board system. At this time, enter **orangepi** in the password box (the default password, if you have modification, please fill in the modified one).




b. The display after successful login is shown in the figure below



10) Software such as Apache, MySQL, and PHP can be installed in the software store of the pagoda, and various applications can be deployed in one click. Please explore it yourself. It won't demonstrate one by one here



11) Pagoda command line tool test

```
orangepi@orangepi5:~$ sudo bt
[sudo] password for orangepi:
===============宝塔面板命令行==================
(1) 重启面板服务           (8) 改面板端口
(2) 停止面板服务           (9) 清除面板缓存
(3) 启动面板服务           (10) 清除登录限制
(4) 重载面板服务
(5) 修改面板密码           (12) 取消域名绑定限制
(6) 修改面板用户名         (13) 取消IP访问限制
(7) 强制修改MySQL密码      (14) 查看面板默认信息
(22) 显示面板错误日志      (15) 清理系统垃圾
(23) 关闭BasicAuth认证     (16) 修复面板(检查错误并更新面板文件到最新版)
(24) 关闭动态口令认证          (17) 设置日志切割是否压缩
(25) 设置是否保存文件历史副本  (18) 设置是否自动备份面板
(0) 取消                   (29) 取消访问设备验证
===============================================
请输入命令编号：14
===============================================
正在执行(14)...
===============================================
curl: (28) Resolving timed out after 10000 milliseconds
==================================================================
BT-Panel default info!
==================================================================
外网面板地址: http://:8888/7eaf9ade
内网面板地址: http://192.168.1.139:8888/7eaf9ade
*以下仅为初始默认账户密码，若无法登录请执行bt命令重置账户/密码登录
username: nslvetif
password: ********
If you cannot access the panel,
release the following panel port [8888] in the security group
若无法访问面板，请检查防火墙/安全组是否有放行面板[8888]端口
==================================================================
orangepi@orangepi5:~$
```

12) For more functions of the pagoda, please 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** |

## 3.25. Set the Chinese environment and install the 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.**

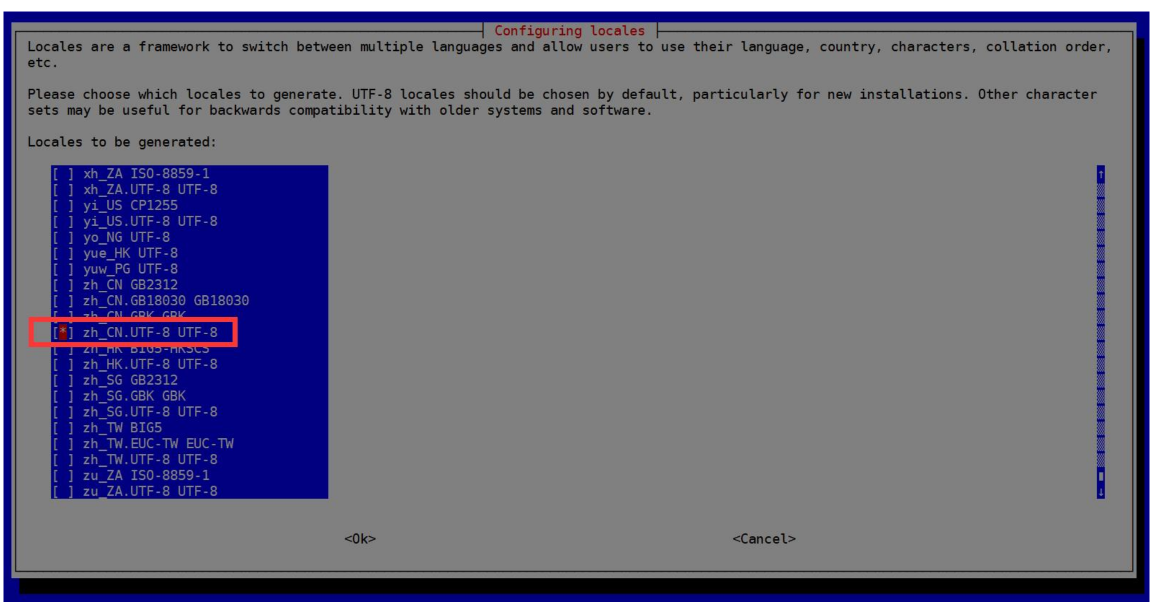
### 3.25.1. Debian 11 system installation method

1) First set the default **Locale** as Chinese

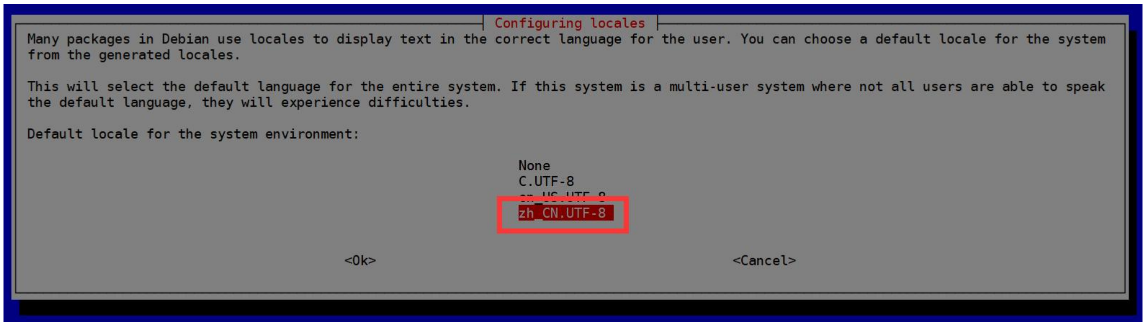
a. Enter the command below to start configured **Locale**

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
```

b. Then select **zh_CN.UTF-8 UTF-8** in the pop-up interface (through the upper and lower direction keys on the keyboard to move up and down, select it through the space key, and finally move the cursor to **<ok>** through the TAB key, then return back Just car)



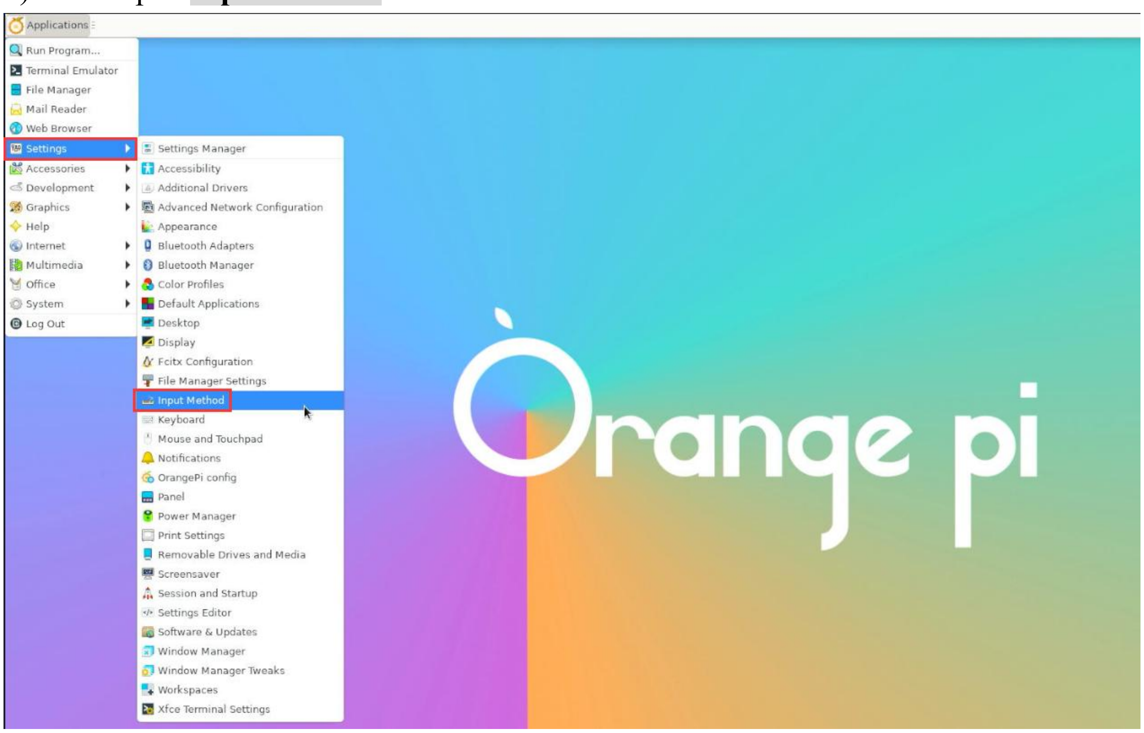

c. Then set the default **locale** to **zh_CN.UTF-8**



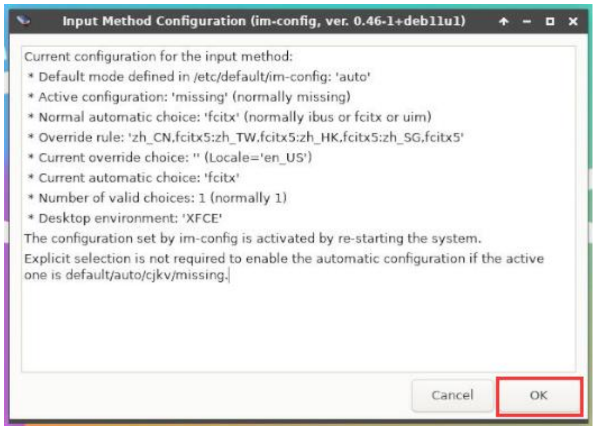

d. After exiting the interface, the **Locale** settings will start. The output displayed by the command line is shown below

```
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.
```

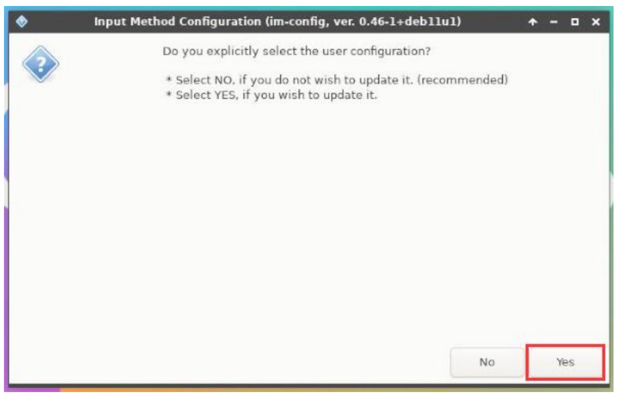
2) Then open **Input Method**



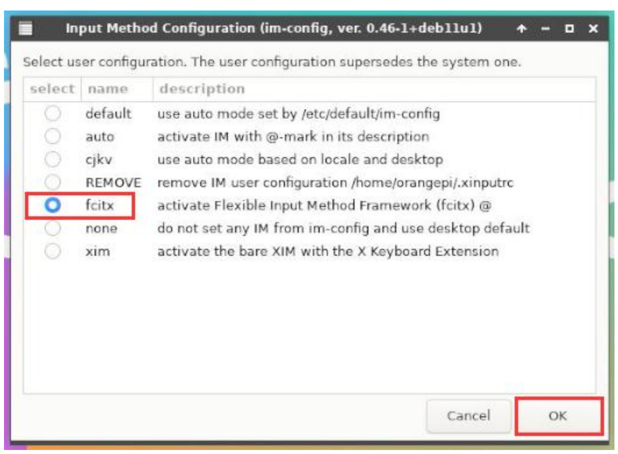

3) Then choose **OK**




4) Then choose **Yes**



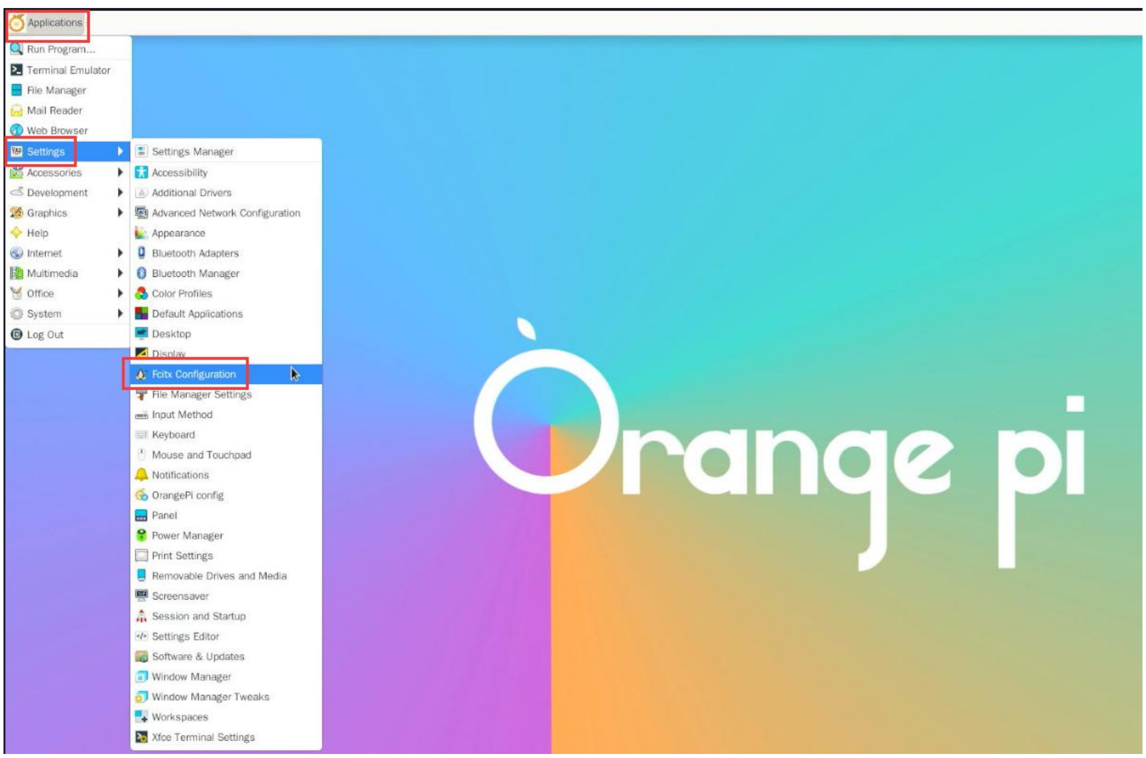

5) Then choose **fcitx**



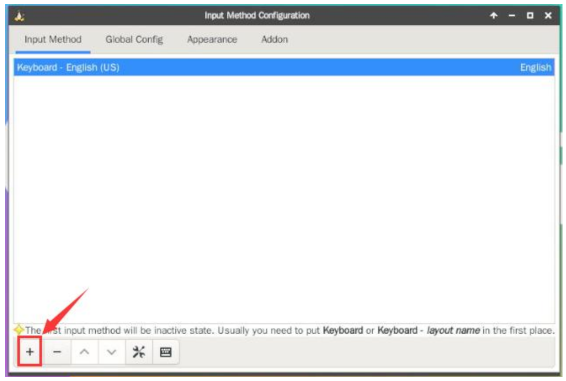

6) Then choose **OK**


7) **Then restart the Linux system to make the configuration take effect**

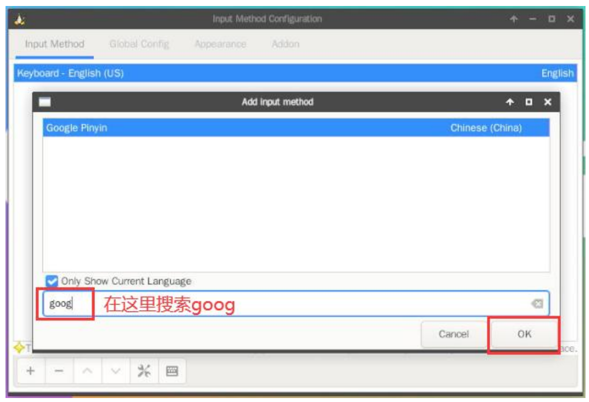
8) Then open **Fcitx configuration**



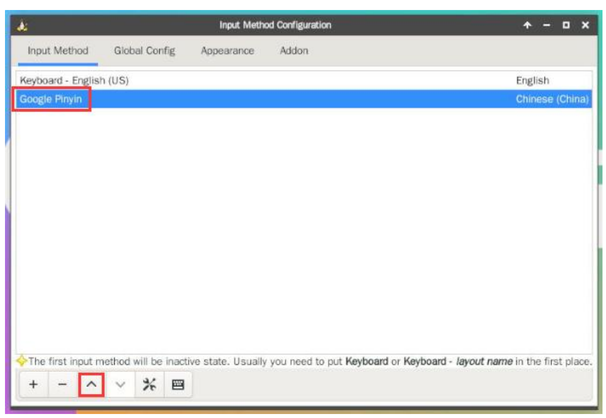

9) Then click the + of the position shown in the figure below

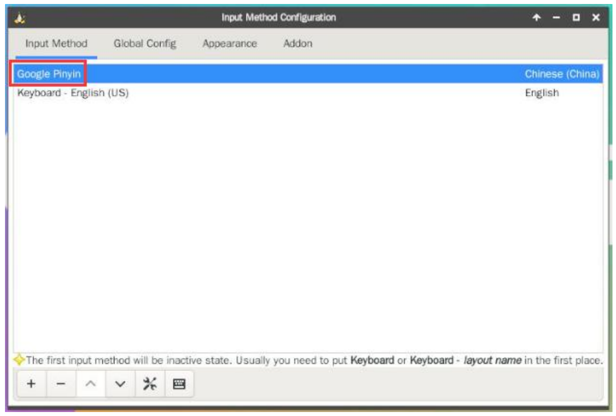



10) Then search **Google Pinyin** and click **OK**

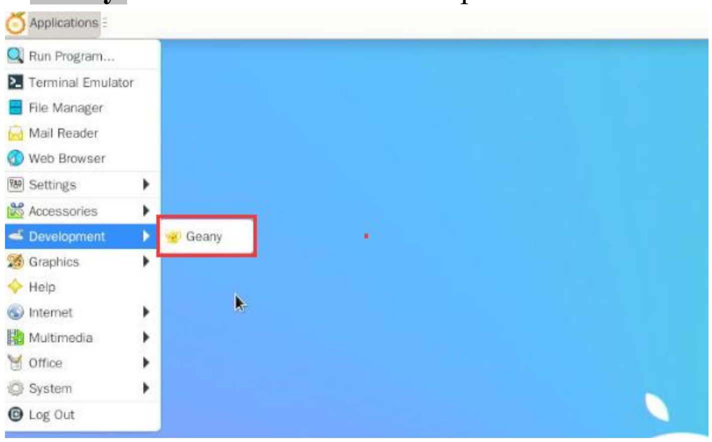



11) Then put **Google Pinyin** to the forefront

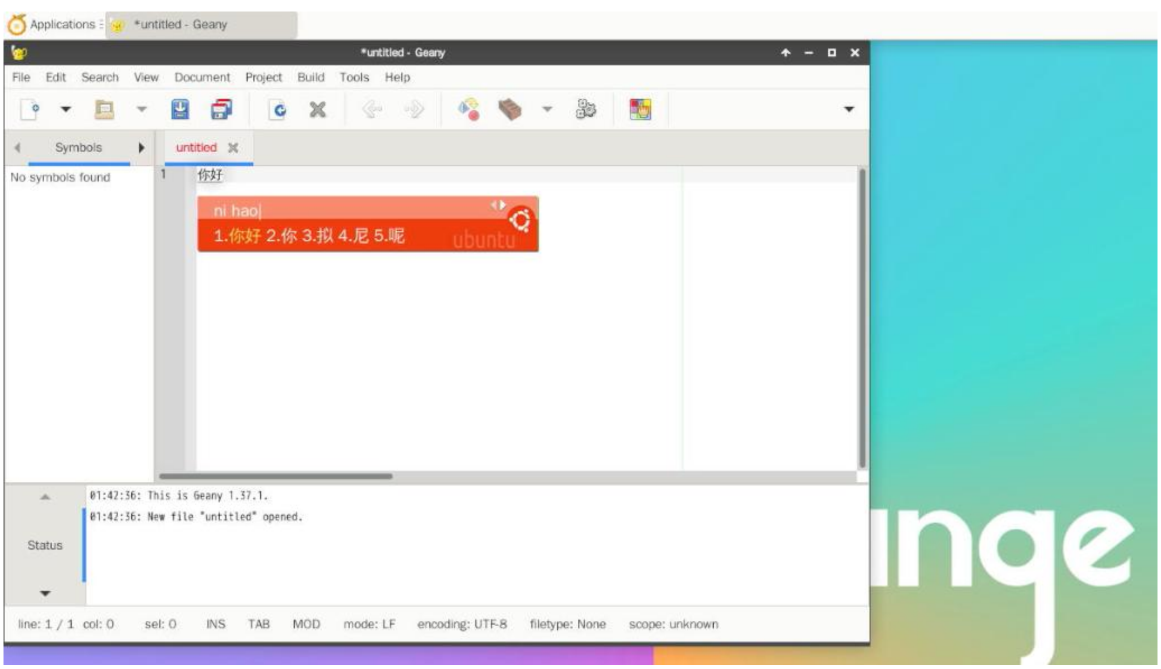





12) Then open the **Geany** editor and Test Chinese input method




13) Chinese input method test as shown below

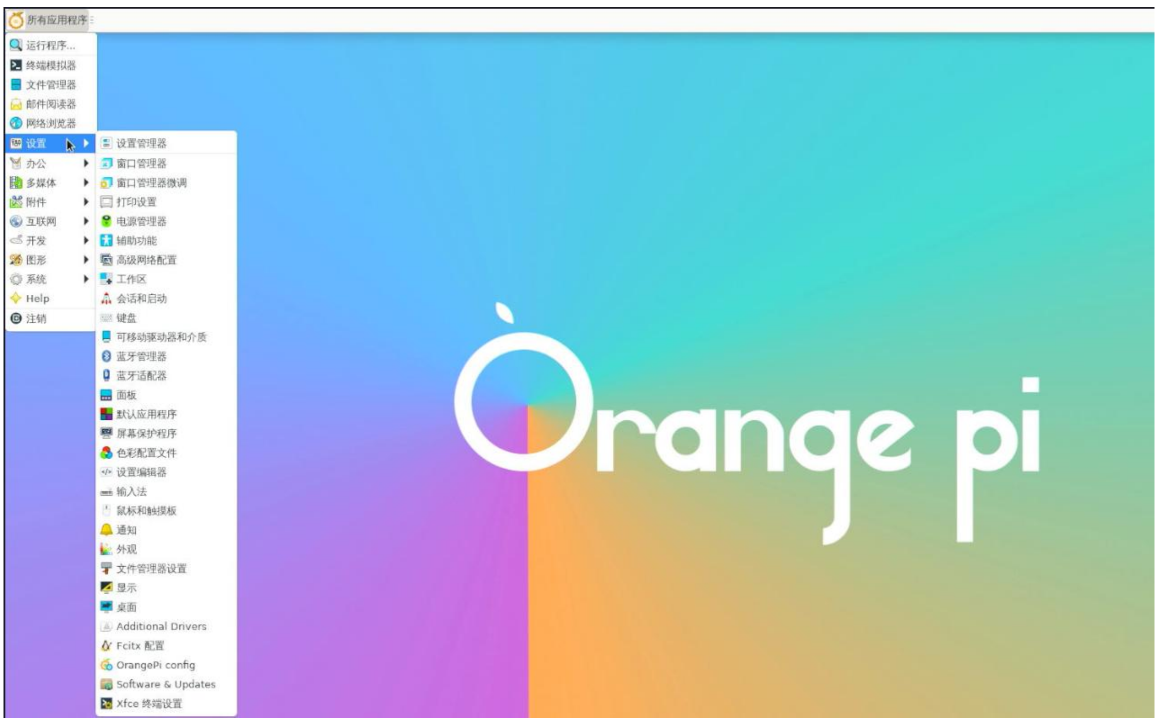


14) You can switch between Chinese and English input methods through **CTRL+Space** shortcut

15) If the entire system is required as Chinese, the variables in **/etc/default/local** can be set to zh_CN.UTF-8

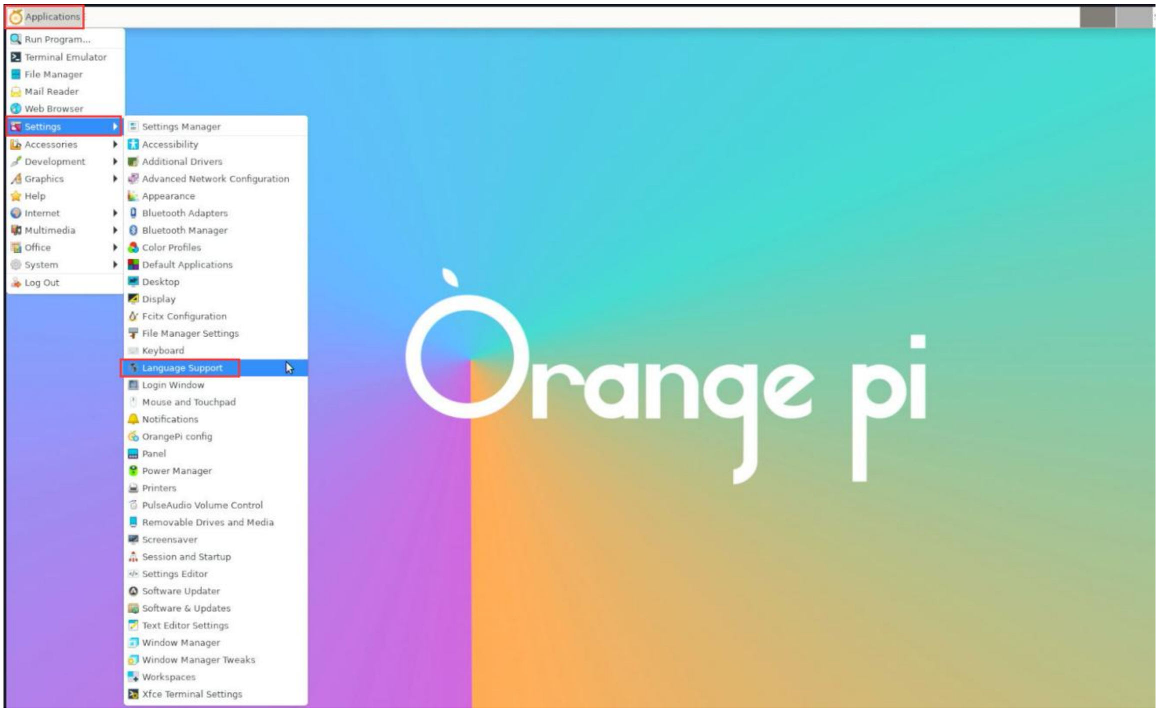
```
orangepi@orangepi:~$ sudo vim /etc/default/locale
#   File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** to see the system display as Chinese

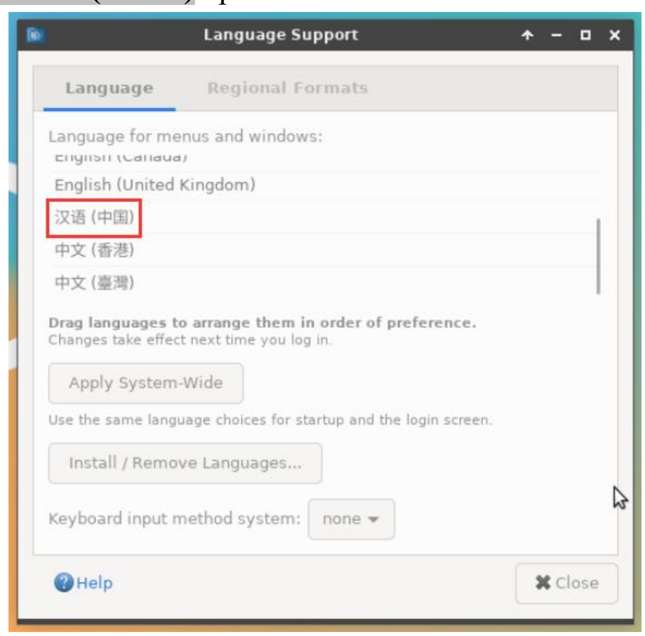



### 3. 25. 2. Ubuntu 20.04 installation method

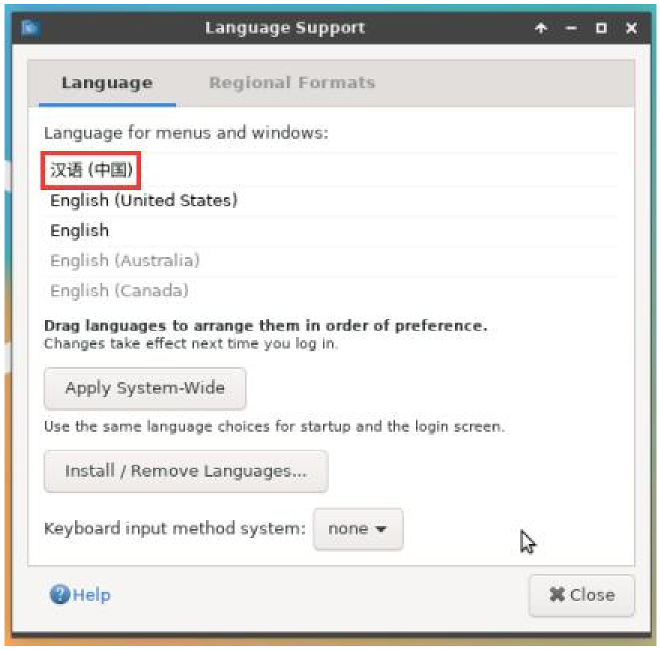
1) First open **Language Support**




2) Then find the **Chinese (China)** option

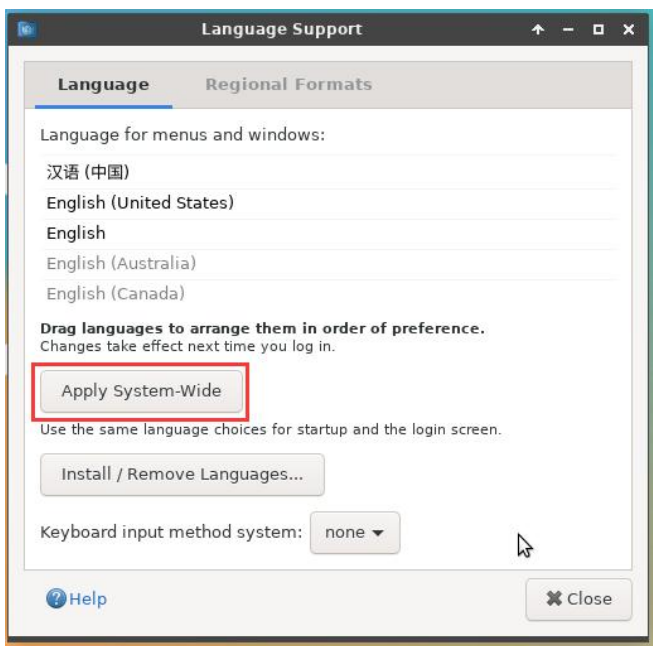



3) Then please use the mouse button to select **Chinese (China)** and hold it down, and then drag it up to the beginning.The display after dragging is shown in the figure below




**Note that this step is not easy to drag, please try more patiently.**

4) Then select **Apply System-Wide** to apply Chinese settings to the entire system

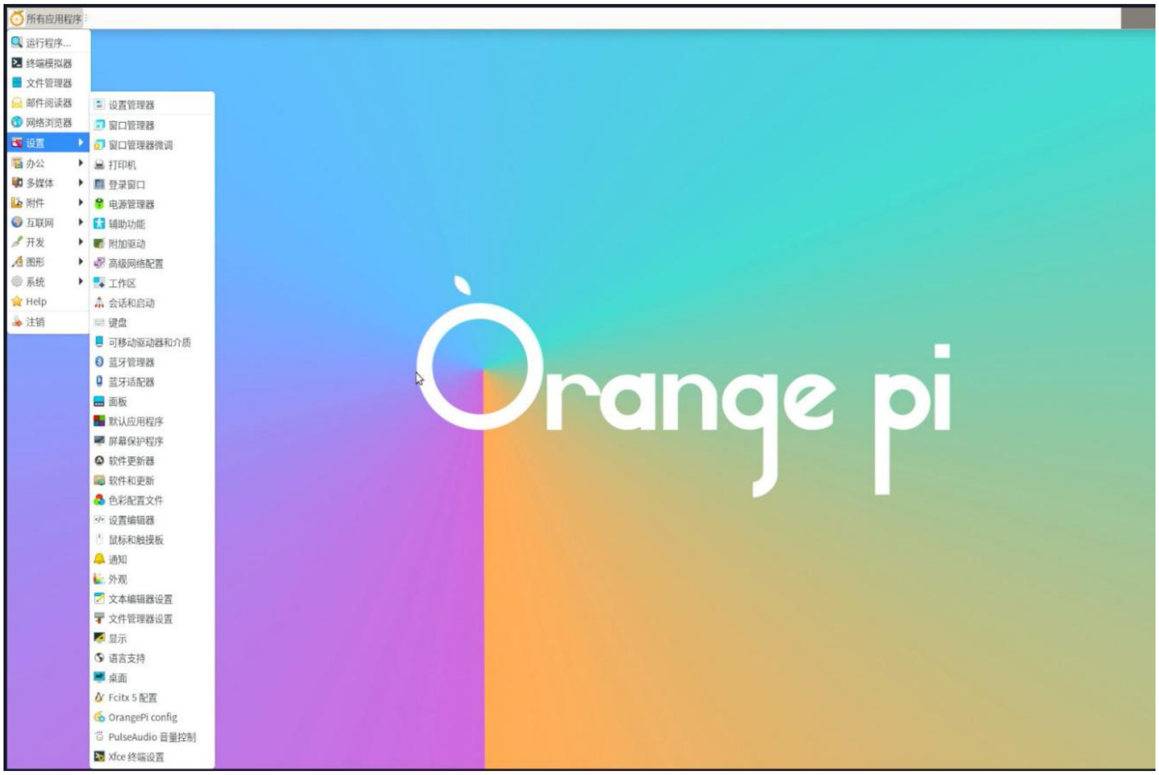



5) Then set the **Keyboard input method system** to **fcitx**

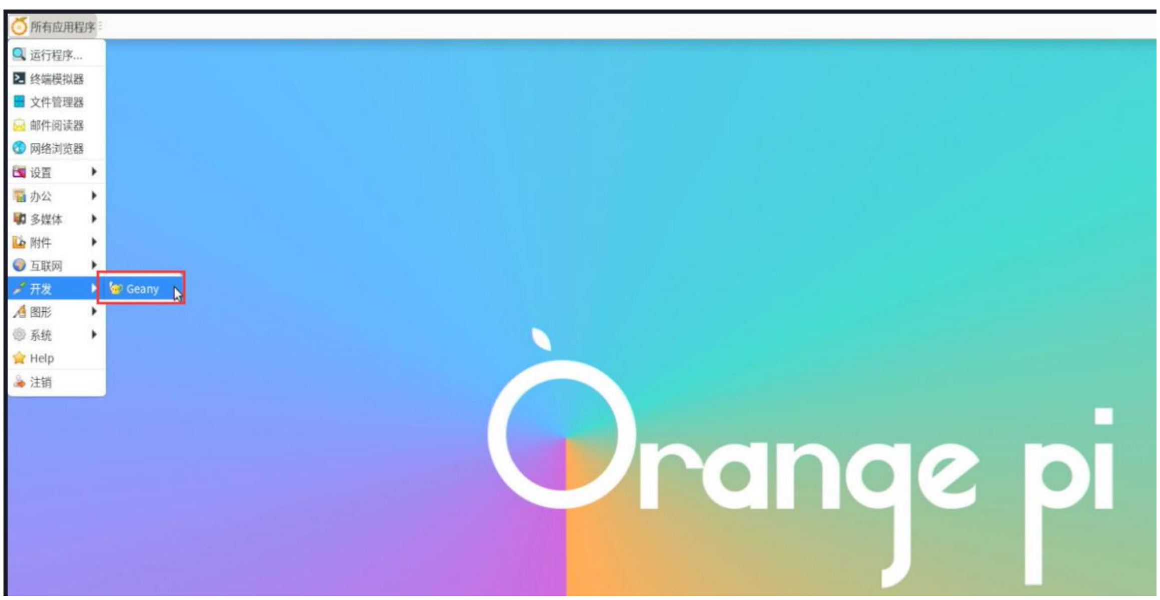

6) **Then restart the Linux system to make the configuration take effect**

7) After re -entering the system, please choose **not to ask me again** at the interface below, and then determine whether the standard folder should be updated to Chinese according to your preferences

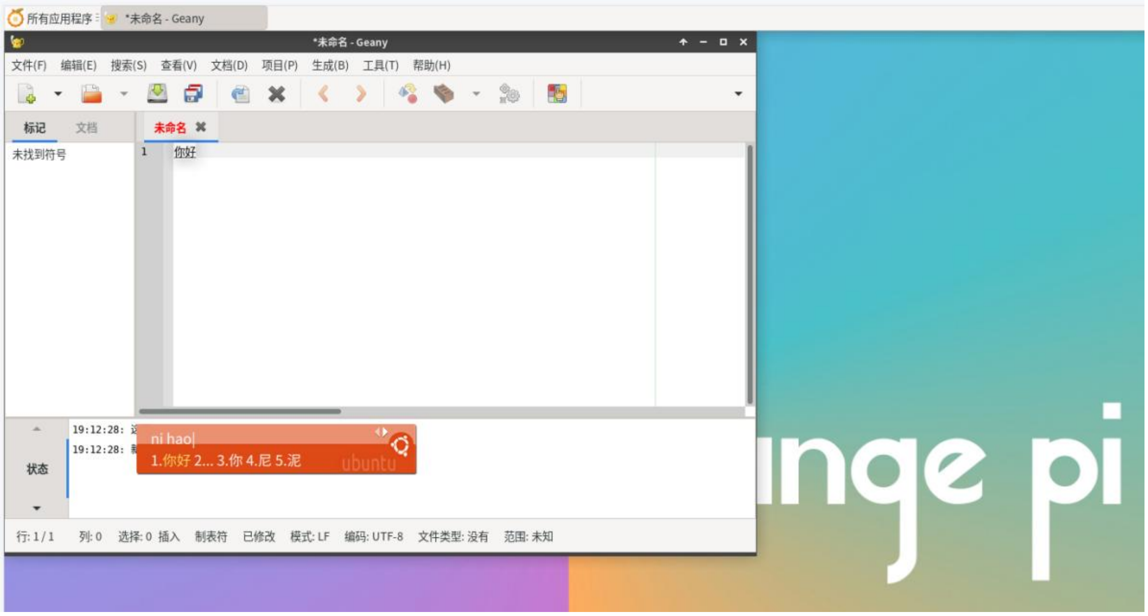

8) Then you can see that the desktop is displayed as Chinese




9) Then we can open **Geany** to test the Chinese input method, The way to open is shown in the figure below

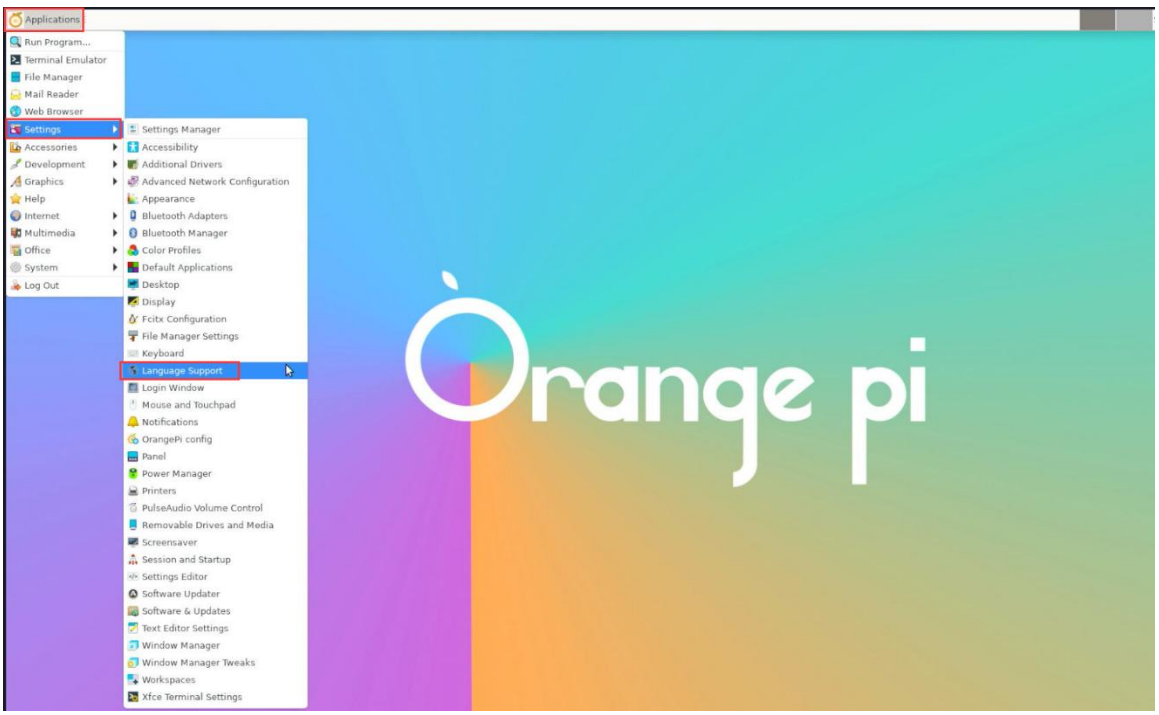



10) After opening **Geany**, the default is an English input method. We can switch to Chinese input method through the **CTRL+Space** shortcut keys, and then we can enter Chinese



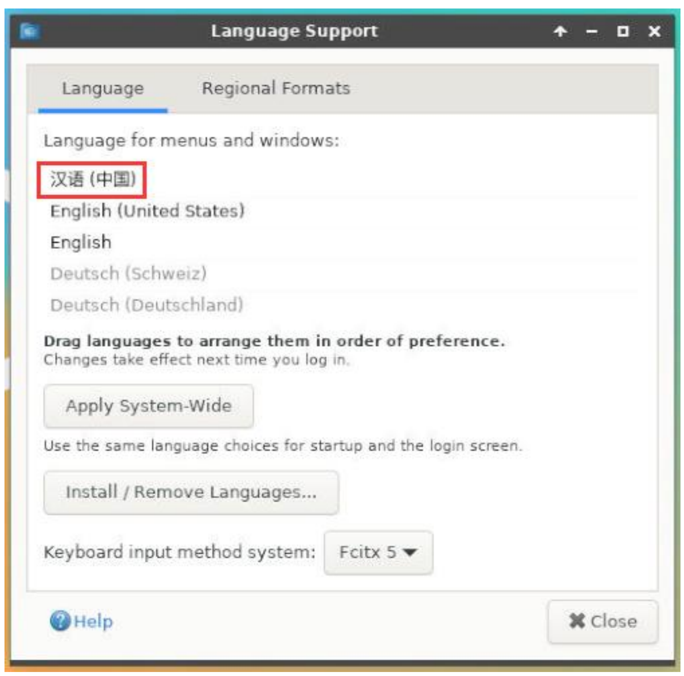

### 3. 25. 3. ubuntu 22.04 installation method

1) First open **Language Support**



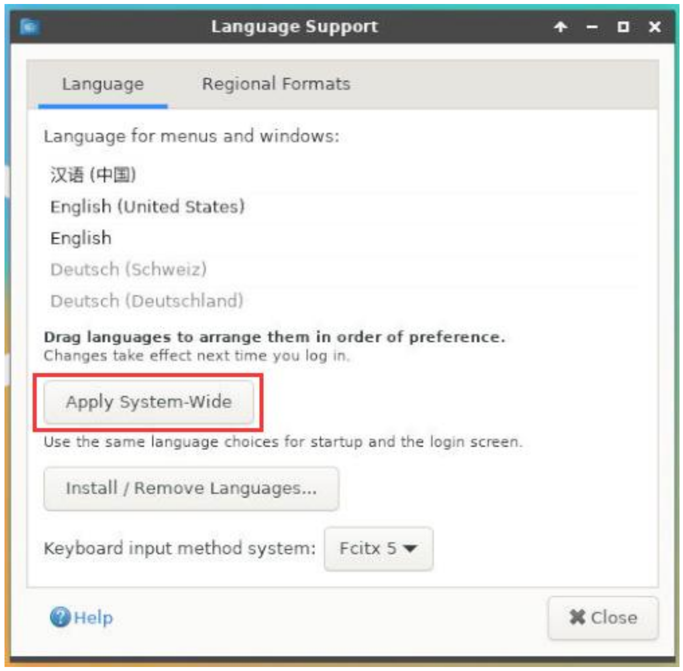

2) Then find the **Chinese (China)** option


3) Then please use the 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.**

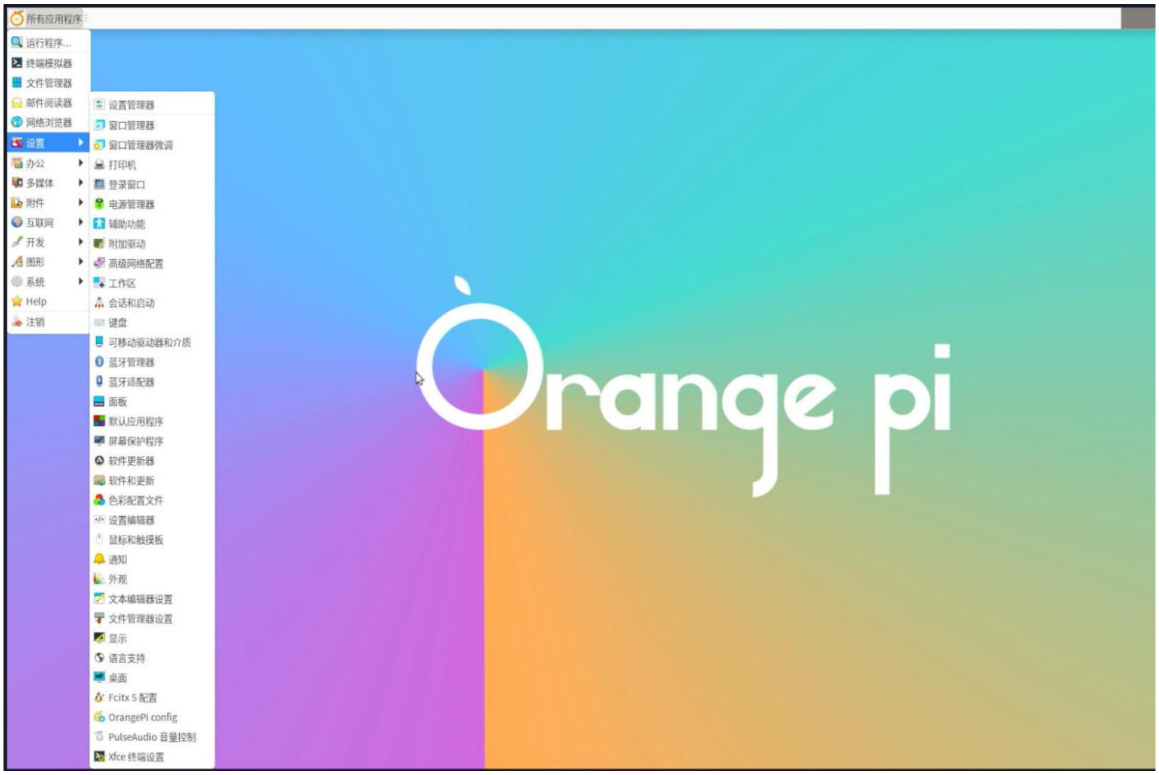
4) Then select **Apply System-Wide** to apply Chinese settings to the entire system



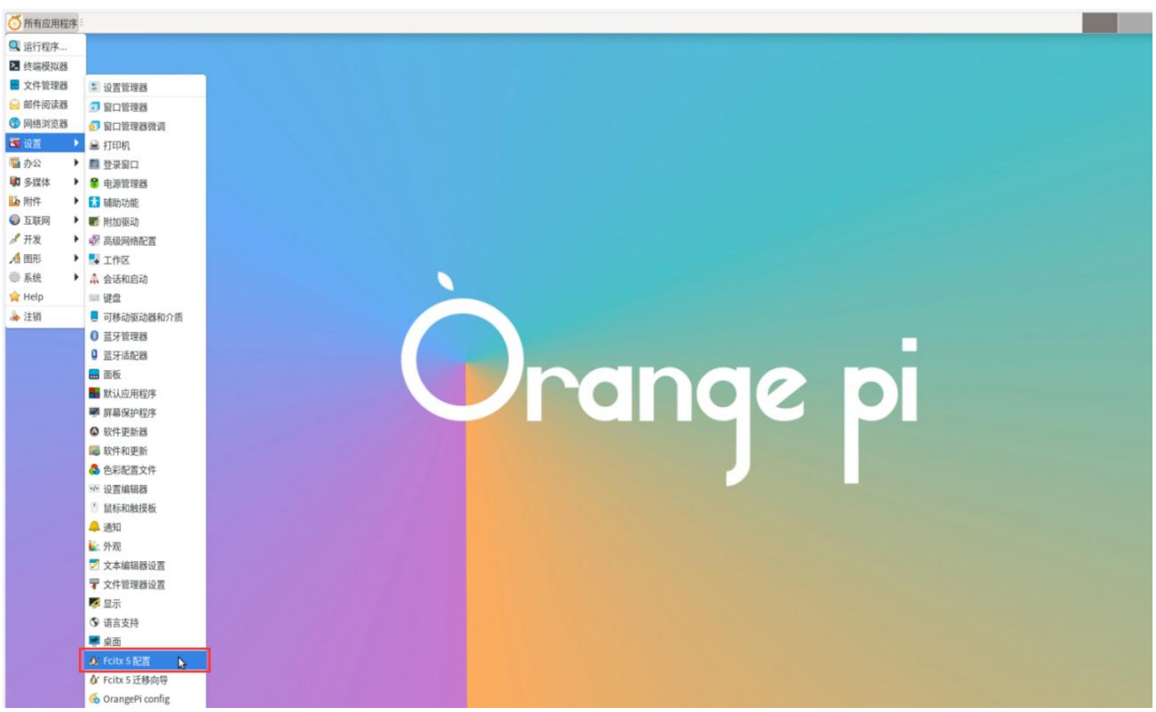

5) **Then restart the linux system to make the configuration effective**

6) After re -entering the system, please choose **not to ask me again** at the interface below, and then determine whether the standard folder should be updated according to your preferences

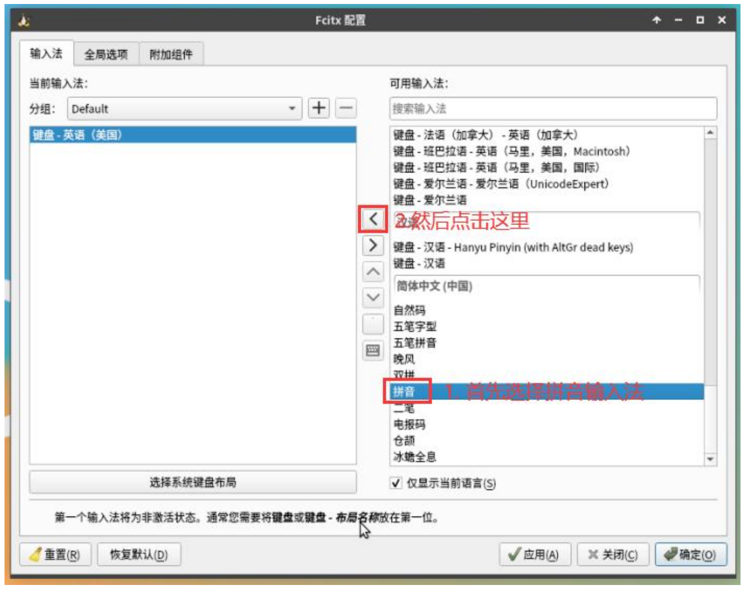

7) Then you can see that the desktop is displayed as Chinese



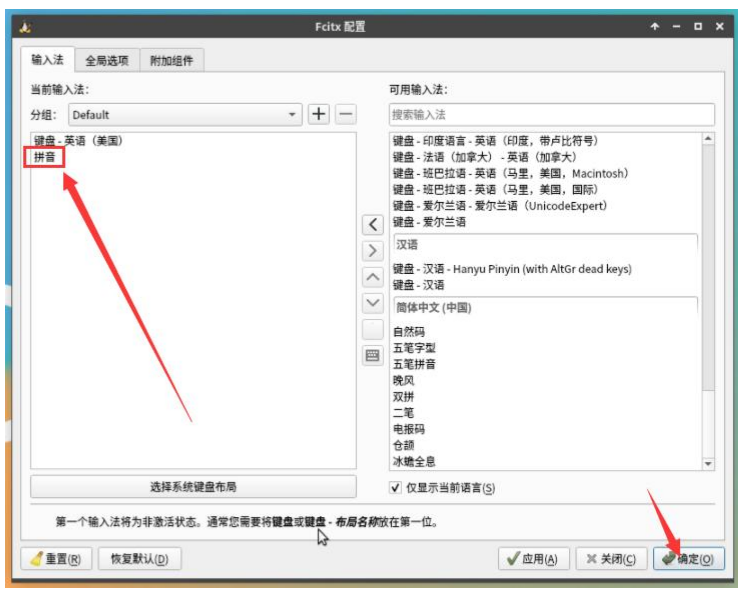

8) Then open the Fcitx5 configuration program



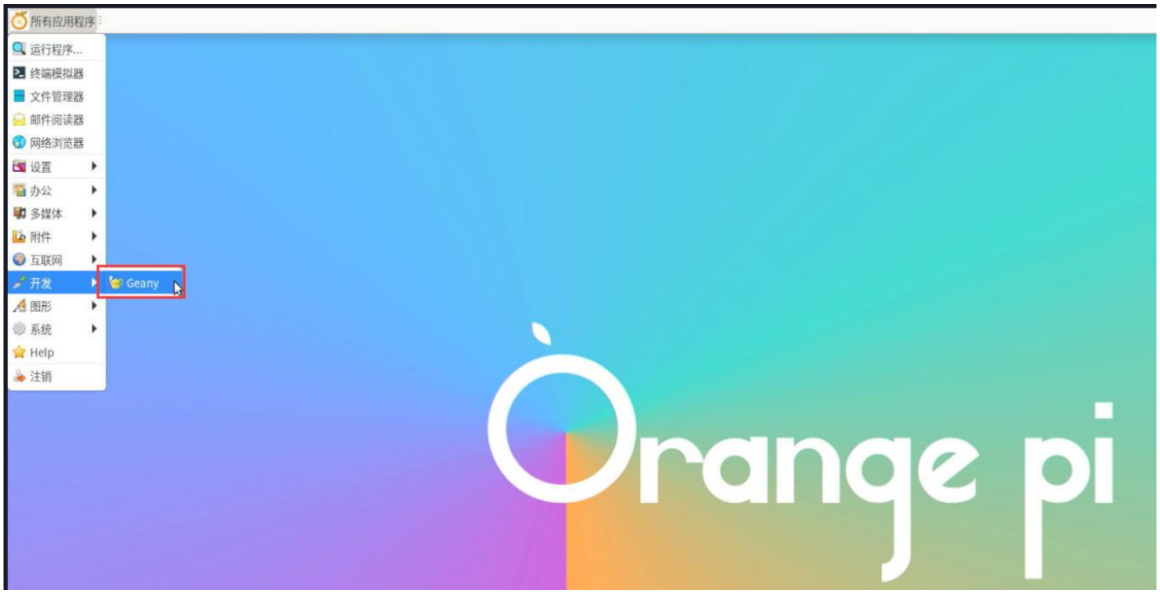

9) Then choose to use Pinyin input method



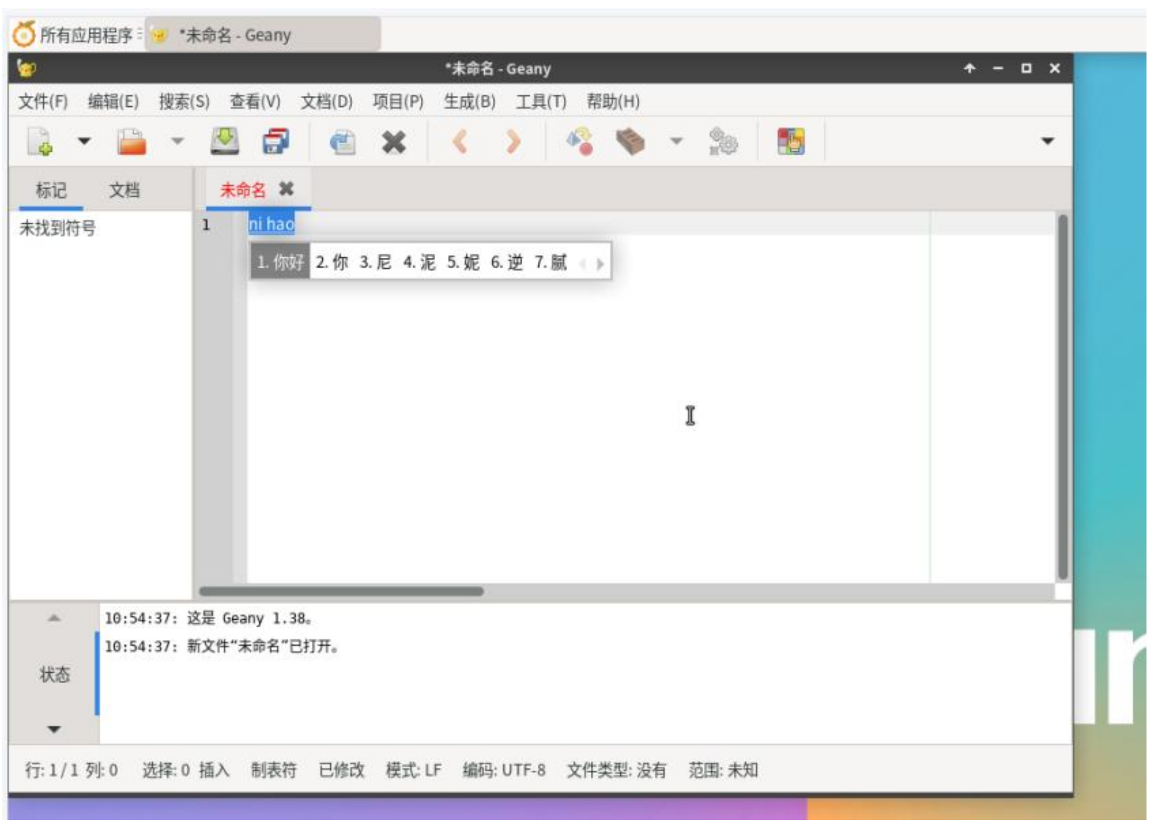

10) The interface after selecting is shown below, then click OK




11) Then we can open **Geany** to test the Chinese input method, the opening method is shown in the figure below




12) After opening **Geany**, the default is an English input method. We can switch to Chinese input method through the **CTRL+Space** shortcut keys, and then we can enter Chinese



## 3. 26. How to remotely log in to the Linux system desktop

### 3. 26. 1. Use NoMachine remote login

**Make sure the Ubuntu or Debian system installed on the development board is a desktop version. In addition, nomachine also provides detailed documents. It is strongly recommended to read this document to be familiar with the use of NoMachine. The document link is shown below:**

**https://knowledgebase.nomachine.com/DT10R00166**

**Nomachine supports Windows, Mac, Linux, iOS, and Android platforms, so we can remotely log in to control Orange PI development boards through Nomachine on a variety of devices. The following demonstrates the Linux system desktop of the Orange PI development board through Nomachine in Windows. For installation methods for other platforms, please refer to the official documentation of Nomachine.**

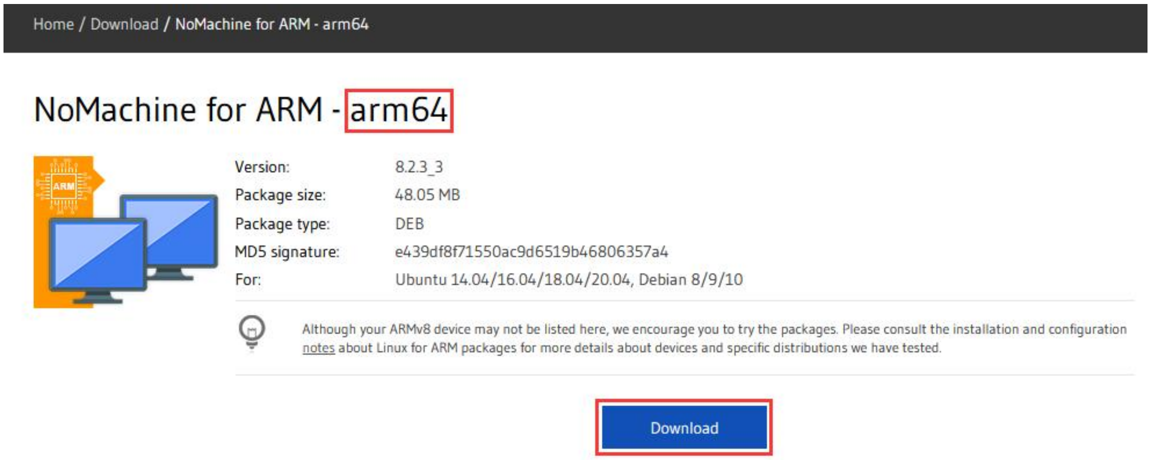
**Before operation, please ensure that the Windwos computer and the development board are in the same local area network, and the Ubuntu or Debian system that can log in to the development board normally.**

1) First download the installation package of Nomachine software Linux **arm64** deb, and then install it in the Linux system of the development board

a. Since RK3588S is a SOC of the ARMv8 architecture, the system we use is Ubuntu or Debian, so you need to download **NoMachine for ARM ARMv8 DEB** installation package. The download link is shown below:

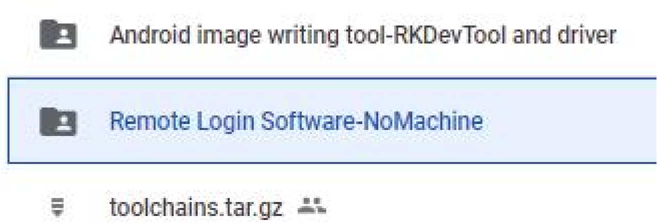
**Note that this download link may change, please recognize the deb package of the ARMV8/ARM64 version.**

```
https://downloads.nomachine.com/download/?id=116&distro=ARM
```

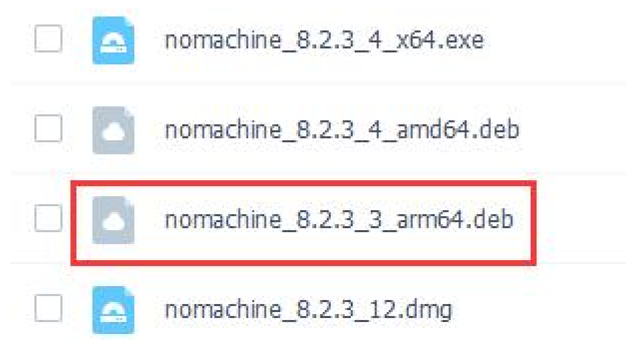



b. In addition, you can also download the installation package of **NoMachine** in the official tool


First enter the **remote login software-Nomachin**e folder




Then download the ARM64 version of the deb installation package



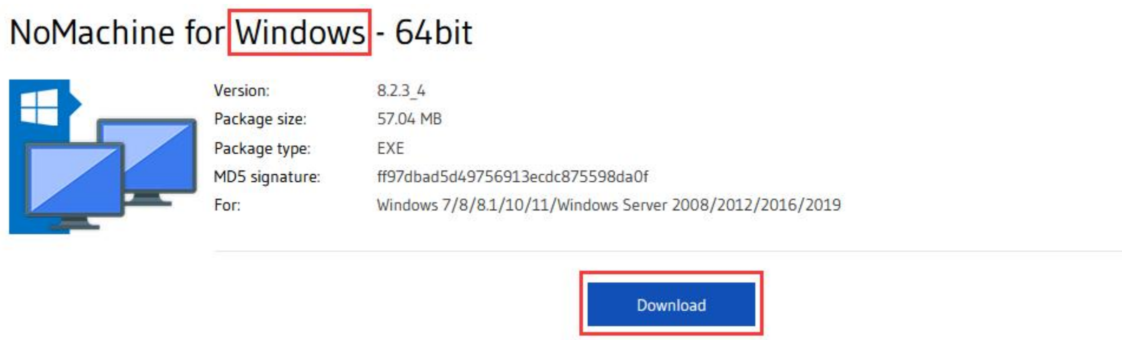

c. Then upload the downloaded **nomachine_8.2.3_3_arm64.deb** to the Linux system of the development board

d. Then use the following command to install **NoMachine** in the Linux system in the development board

```
orangepi@orangepi:~$ sudo dpkg -i nomachine_8.2.3_3_arm64_arm64.deb
```

2) Then download the NoMachine software Windows version of the installation package, the download address is shown below

**https://downloads.nomachine.com/download/?id=8**

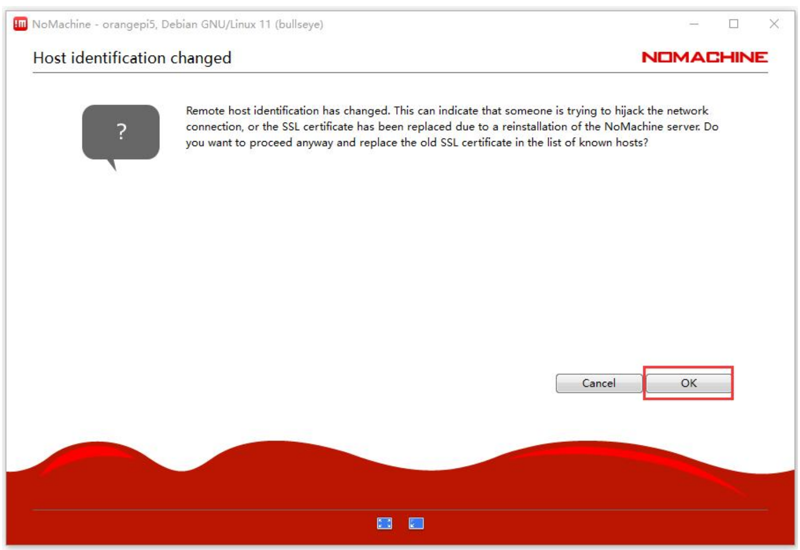



3) Then install nomachine in Windows, **please restart the computer after installation**

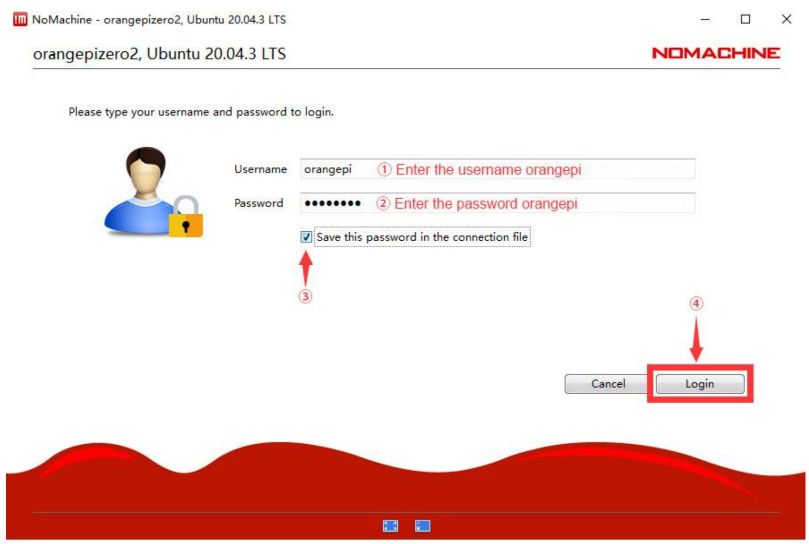
4) Then open **NoMachine** in Window


5) After Nomachine starts, it will automatically scan the other equipment installed in the local area network with Nomachine. After entering the main interface of Nomachine, you can see that the development board is already in the connected device list, and then click the location shown in the red box below in the figure below. You can start logging in to the linux system desktop of the development board


6) Then click **OK**



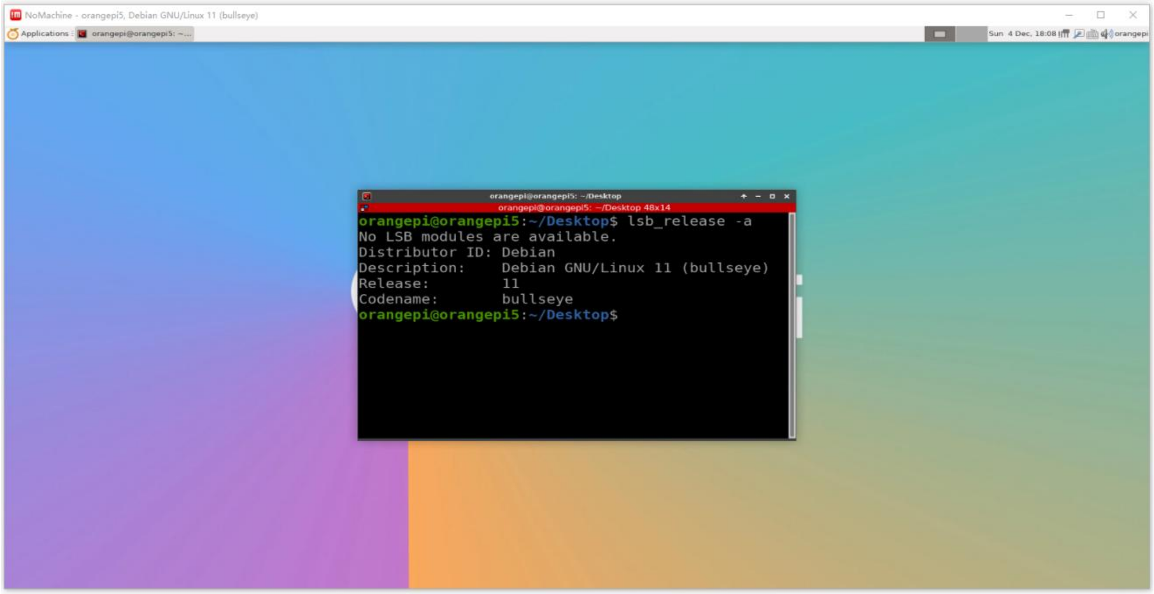

7) Then enter the user name and password of the linux system in the corresponding position in the figure below, and then click **login** to start logging in



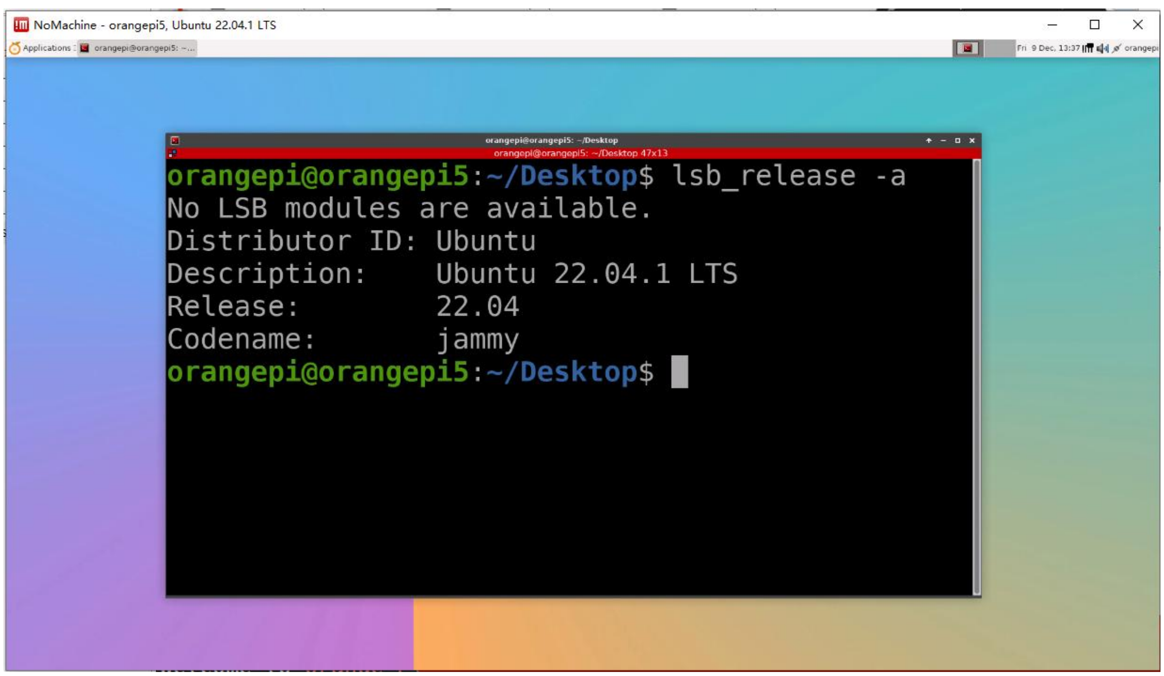

8) Then click OK in the next interface


9) Finally, you can see the desktop of the development board Linux system

a. Debian11




b. Ubuntu22.04




### 3. 26. 2. Use VNC remote login

**Before operation, please ensure that the Windwos computer and the development board are in the same local area network, and the Ubuntu or Debian system that can log in to the development board normally.**

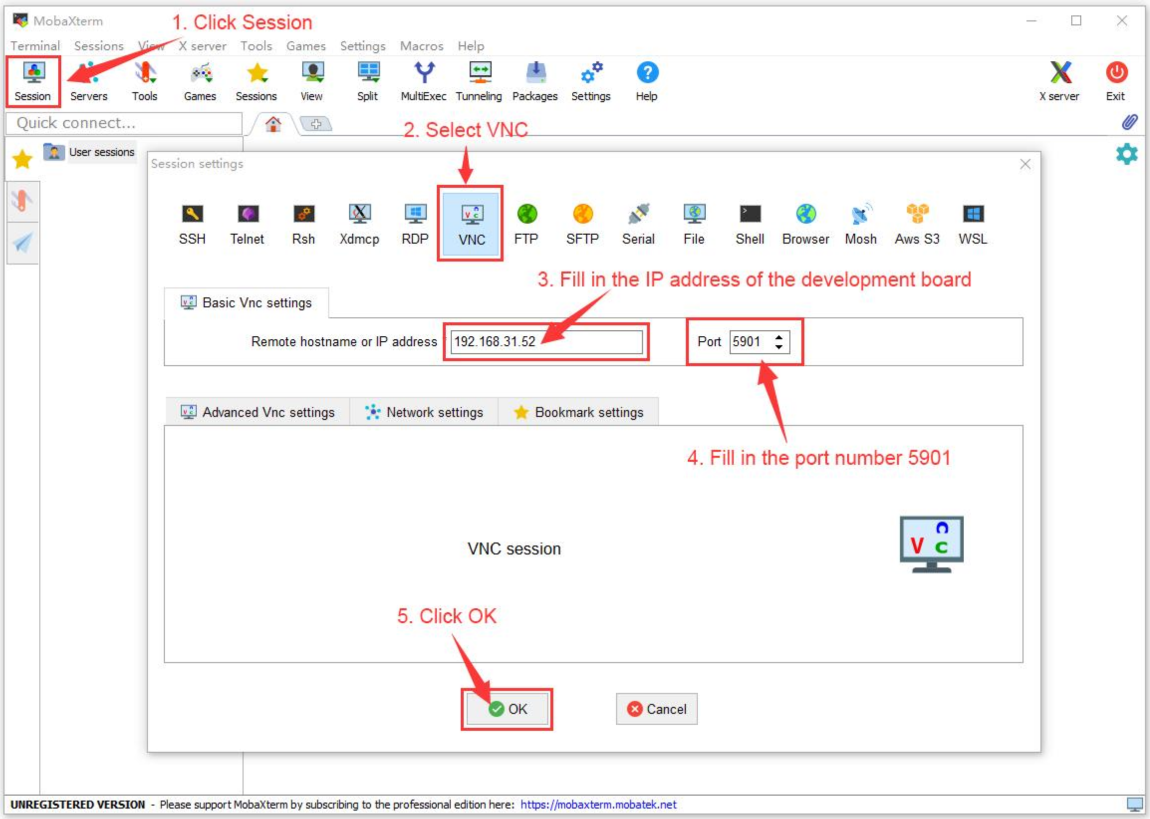
**Ubuntu20.04 tests many problems with VNC, please do not use this method.**

1) First run the **set_vnc.sh** script settings, **remember to add SUDO permissions**

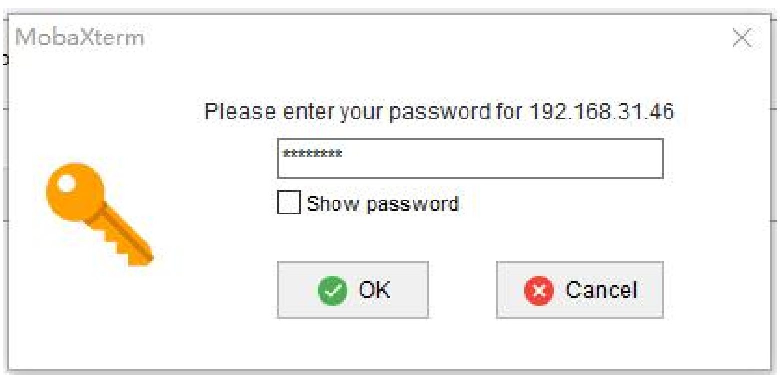
```
orangepi@orangepi:~$ sudo set_vnc.sh
You will require a password to access your desktops.

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 orangepi5:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi5:1.log

Killing Xtightvnc process ID 3047

New 'X' desktop is orangepi5:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepi5:1.log
```

2) The steps to connect the development board Linux system desktop using MobaxTerm software are shown below:

a. First click session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm

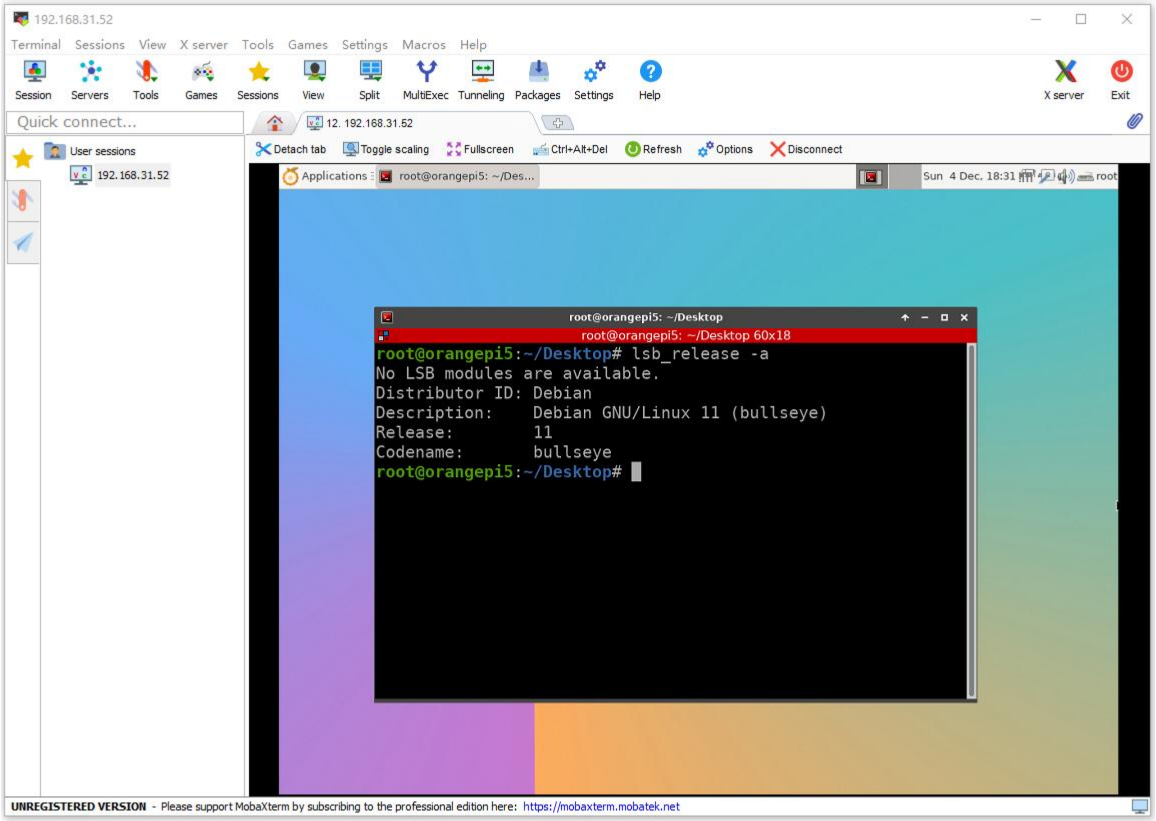



b. Then enter the password of the VNC set before

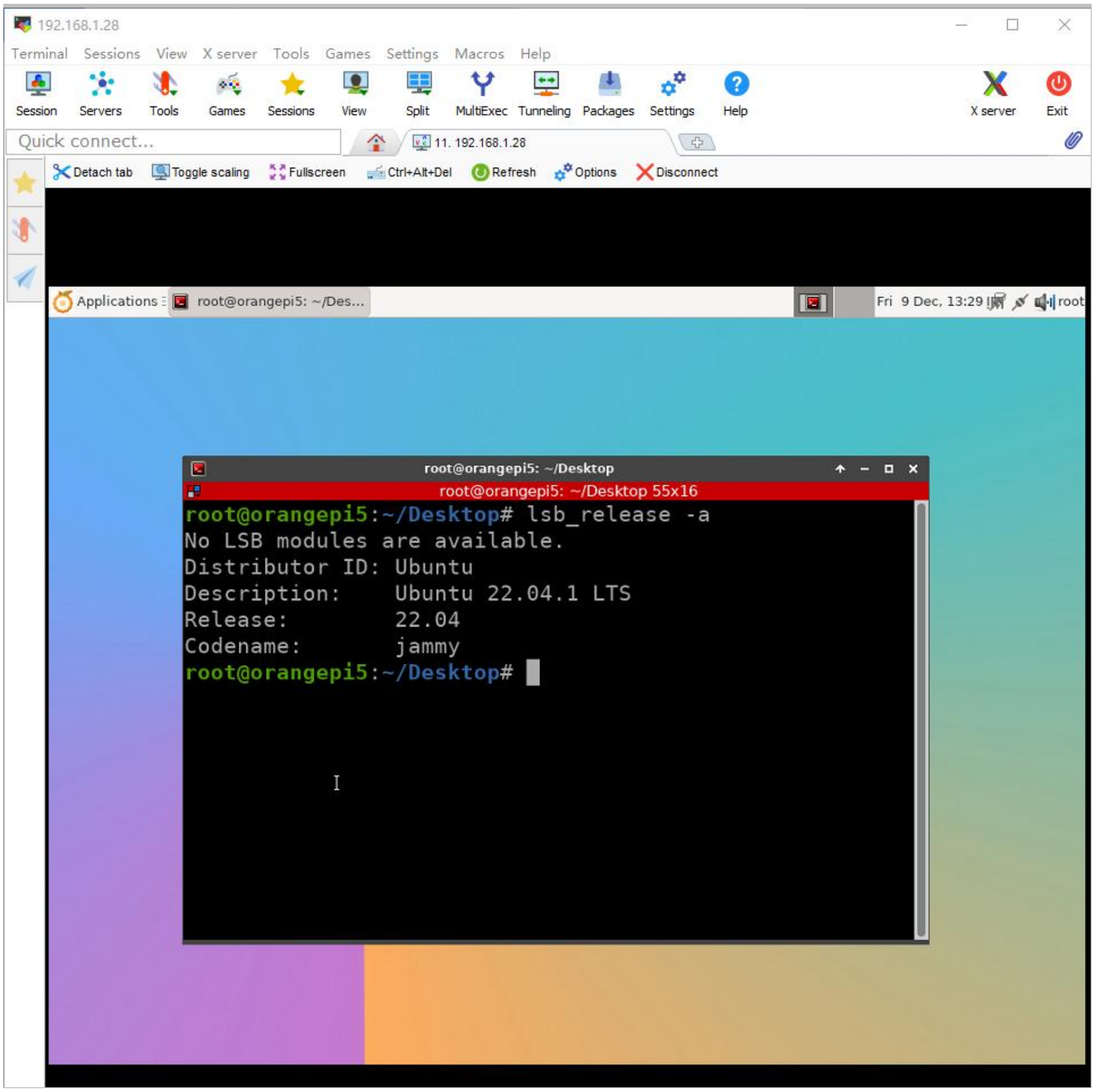



c. The interface after the login is successfully displayed as shown in the figure below, and then you can remotely operate the desktop of the linux system remotely.

a) Debian11 login shows the following shown

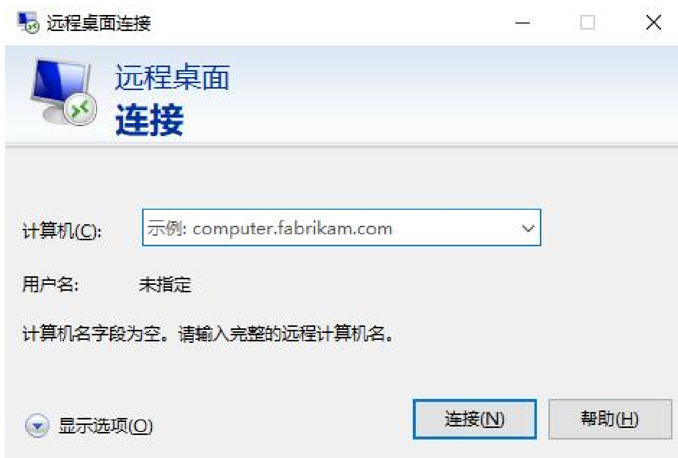



b) Ubuntu22.04 Login shows as shown below




3) The steps to log in to the development board Linux system desktop using the **remote desktop connection** of Windows are

a. First open the **remote desktop connection** that comes with Windows




b. Then enter the IP address of the development board

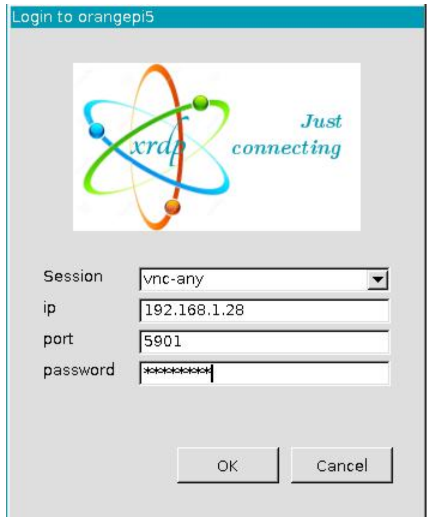

c. Then set the connection information according to the description below

a) **Session：** Need to choose vnc-any

b) **ip：** You can enter 127.0.0.0 or the IP address of the development board

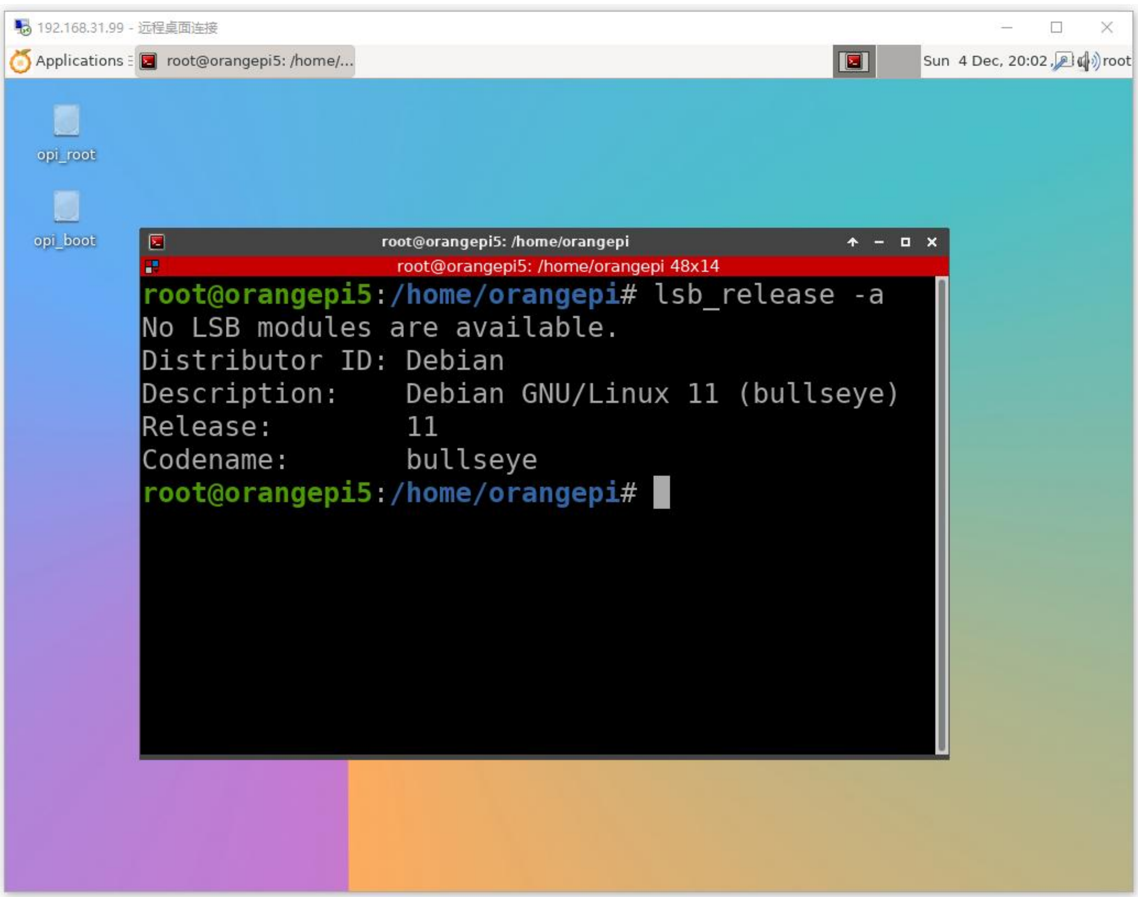
c) **port：** Generally 5901

d) **password：** You need to enter the password of VNC




d. The display of successfully logging in to the development board Linux system desktop is shown in the figure below

a) Debian11 login shows the following shown




**b) Ubuntu22.04 is currently unavailable, please do not use this method**

## 3. 27. Some programming language test supported by the linux system

### 3. 27. 1. Debian Bullseye system

1) Debian Bullseye is installed with the GCC compilation tool chain by default, which can directly compile the C language program in the linux system of the development board

a. The version of gcc is shown below

```
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. **hello_world.c** program that writes c language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

2) Debian Bullseye defaults to install Python3

a. Python specific version is shown below

```
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. The **hello_world.py** program that writes python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The results of running **hello_world.py** are shown below

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bullseye's compilation tool and operating environment without Java 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 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. 27. 2. Ubuntu Focal system

1) Ubuntu Focal has a gcc compilation tool chain by default, which can directly compile the C language program in the linux system of the development board

a. The version of gcc is shown below

```
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. Python3 specific version is shown below

```
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. The **hello_world.py** program that writes python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The results of running **hello_world.py** are shown below

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Ubuntu Focal defaults to compile tools and operating environments that are not installed in Java

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. 27. 3. Ubuntu Jammy system

4) Ubuntu Jammy has the gcc compilation tool chain by default, which can compile the C language program directly in the linux system of the development board

a. The version of gcc is shown below

```
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. The **hello_World.c** program that writes c language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
```

```
            printf("Hello World!\n");

            return 0;
}
```

c. Then compile and run **hello_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c
orangepi@orangepi:~$ ./hello_world
Hello World!
```

5) Ubuntu Jammy is installed with Python3 by default

a. Python3 specific version is shown below

```
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. The **hello_world.py** program that writes Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

c. The results of running **hello_world.py** are shown below

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

6) Ubuntu Jammy defaults to compile tools and operating environment without 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 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. 28. QT installation method

1) Use the following scripts to install QT5 and QT Creator

```
orangepi@orangepi:~$ install_qt.sh
```

2) After installation, it will automatically print the QT version number

a. Ubuntu20.04's own Qt version is **5.12.8**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/aarch64-linux-gnu
```

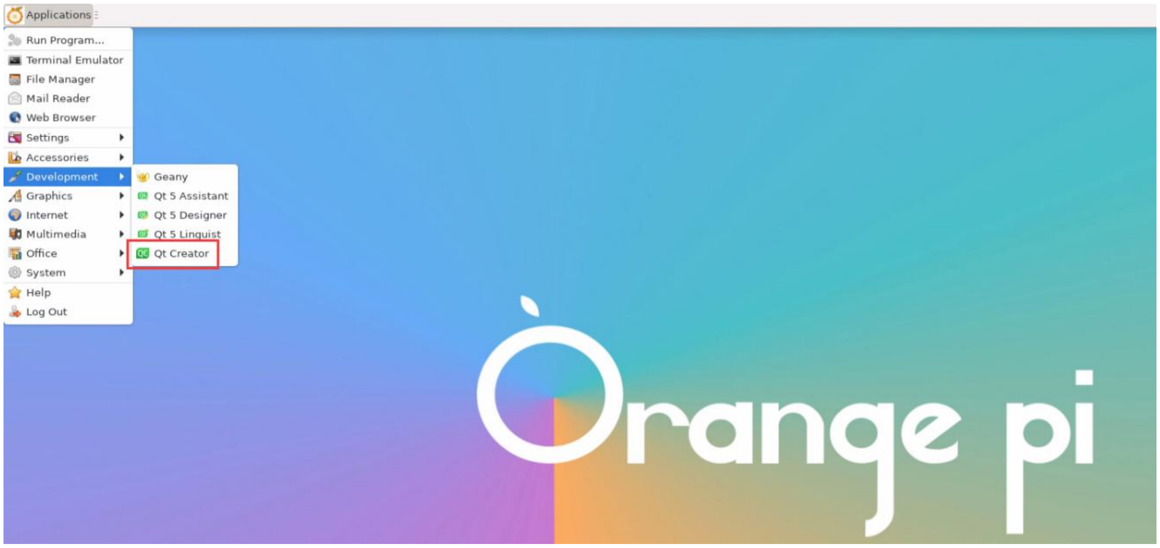
b. Ubuntu20.04's own Qt version is **5.15.3**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.3 in /usr/lib/aarch64-linux-gnu
```

c. The QT version comes with Debian11 is **5.15.2**

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

3) Then you can see the lax icon of QT Creator in **Applications**



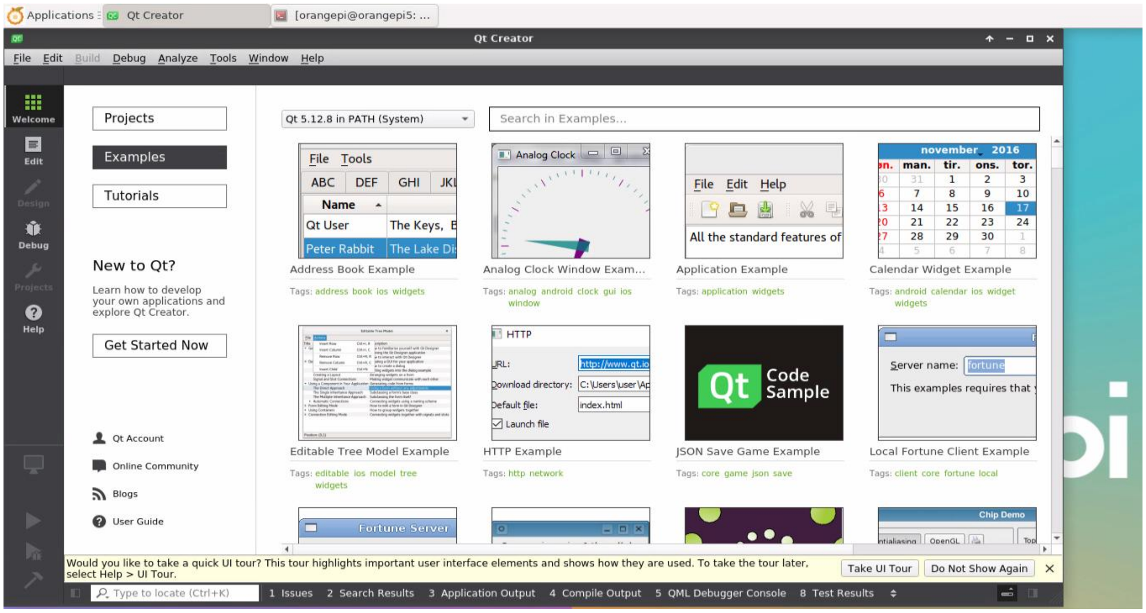

You can also use the following command to open QT Creator

```
orangepi@orangepi:~$ qtcreator
```

**During the startup process of QT and QT applications, if the error below is prompted, please ignore it directly. This error will not affect the operation of the application**

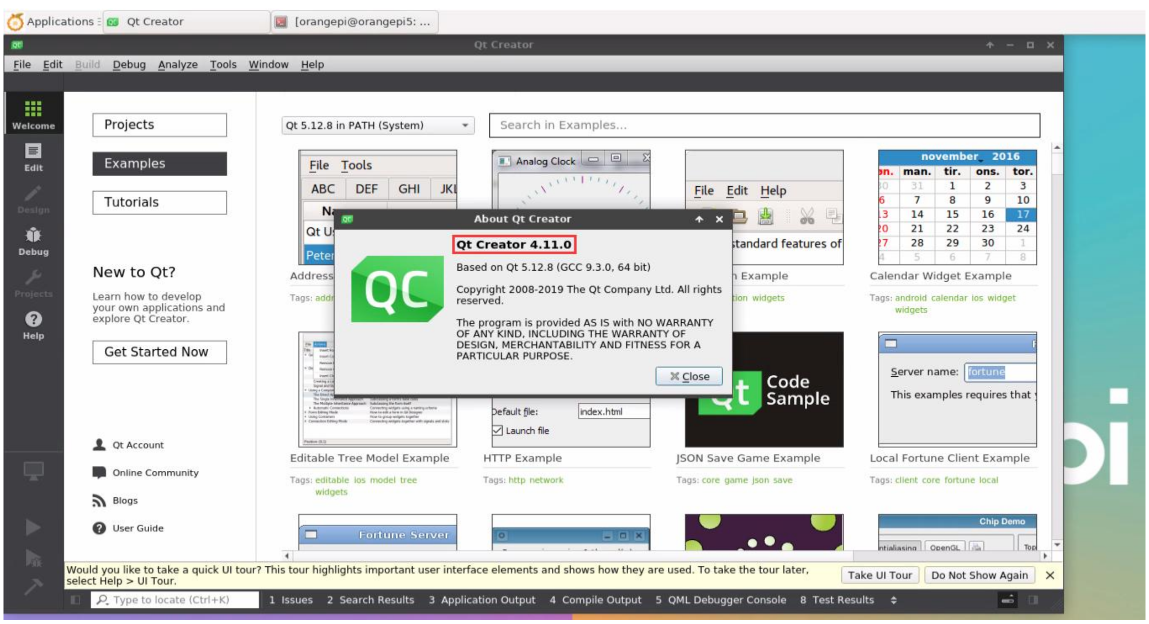
**libGL error: failed to create dri screen**
**libGL error: failed to load driver: rockchip**
**libGL error: failed to create dri screen**
**libGL error: failed to load driver: rockchip**

4) The interface after the QT Creator is opened is shown below

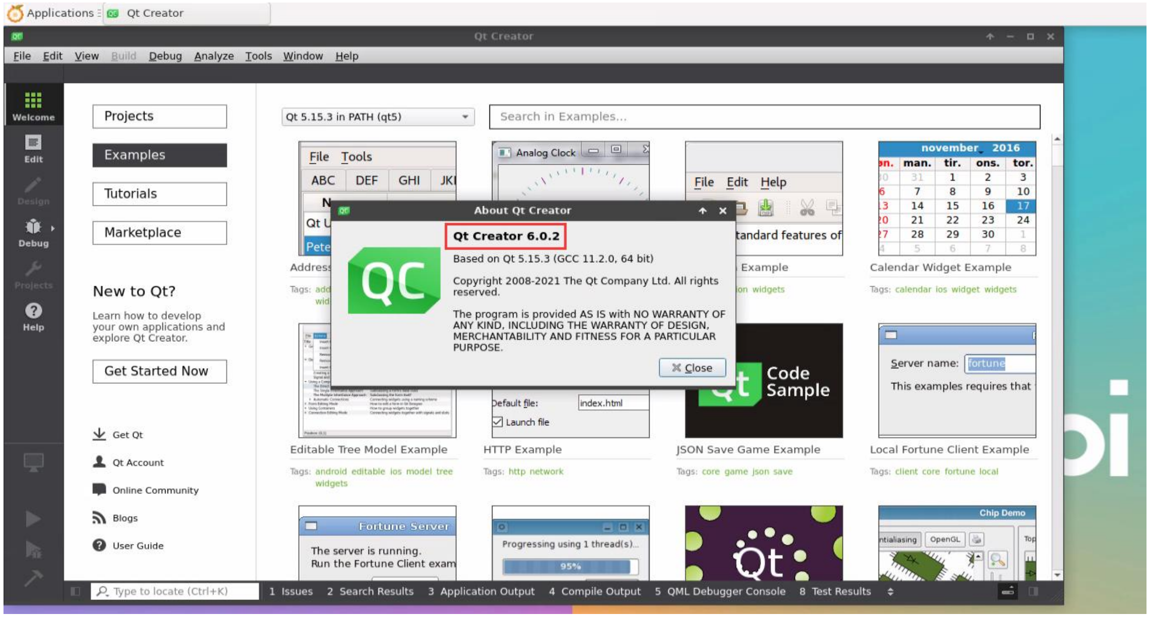



5) The version of QT Creator is shown below

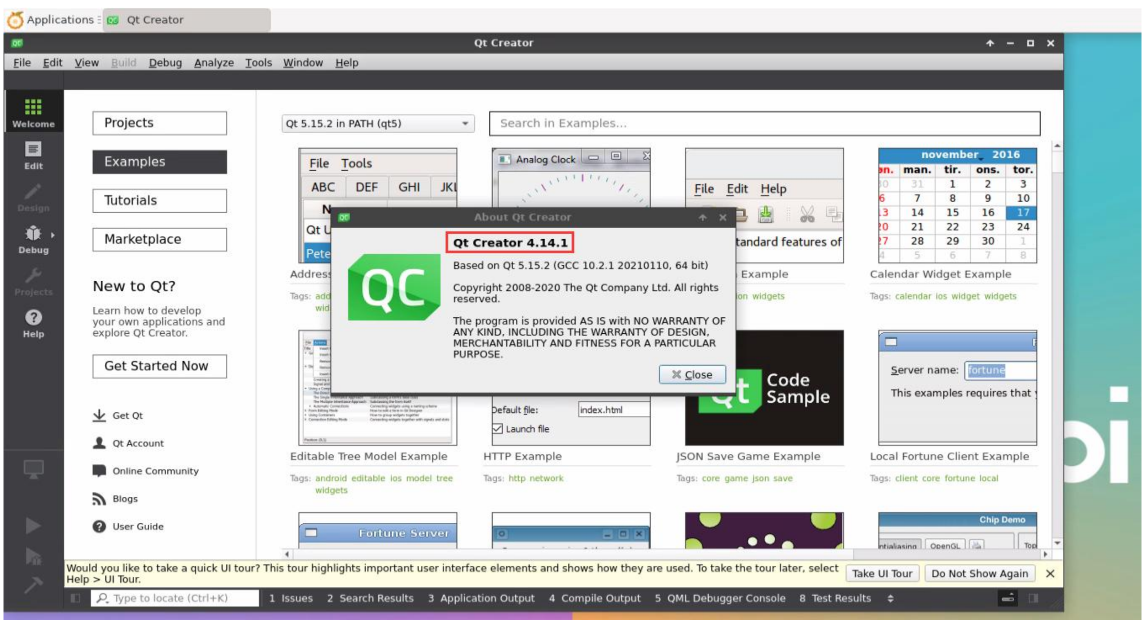
a. The default version of QT Creator in **Ubuntu20.04** is shown below




b. The default version of QT Creator in **Ubuntu22.04** is shown below

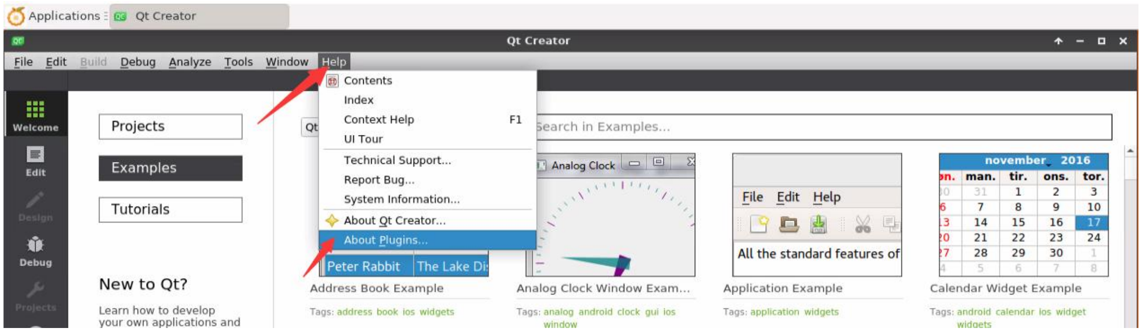



c. The default version of QT Creator in **Debian11** is shown below

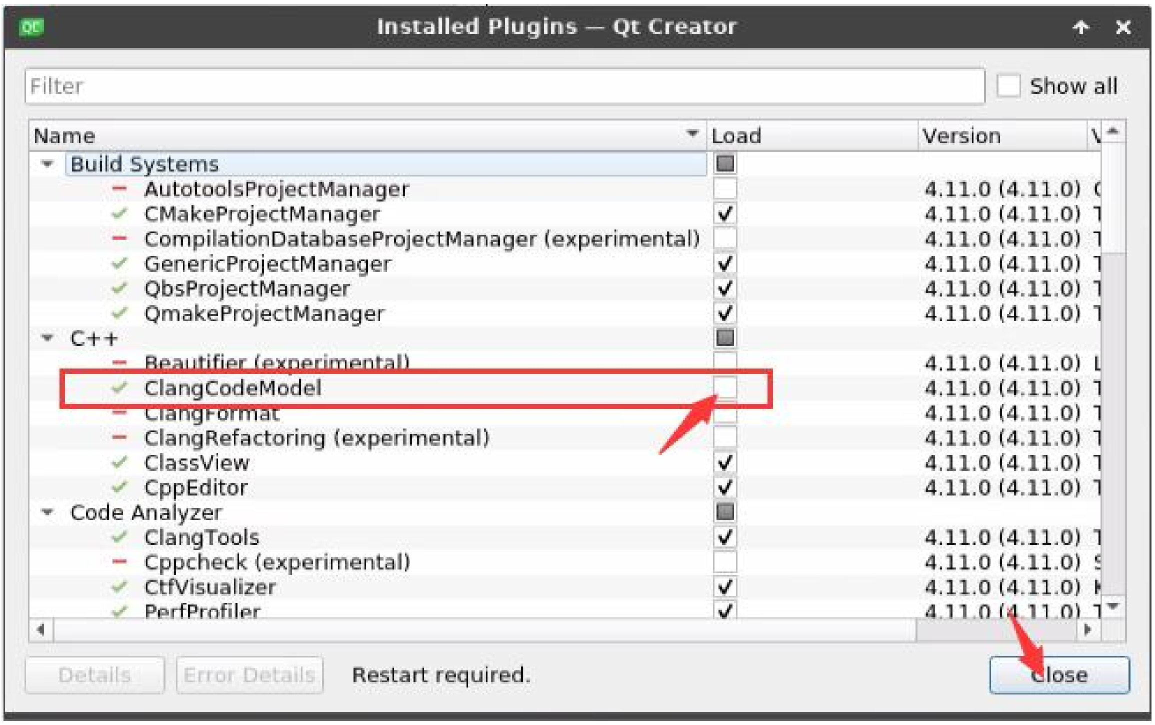



6) Then set QT

a. First open **Help**->**About Plugins...**

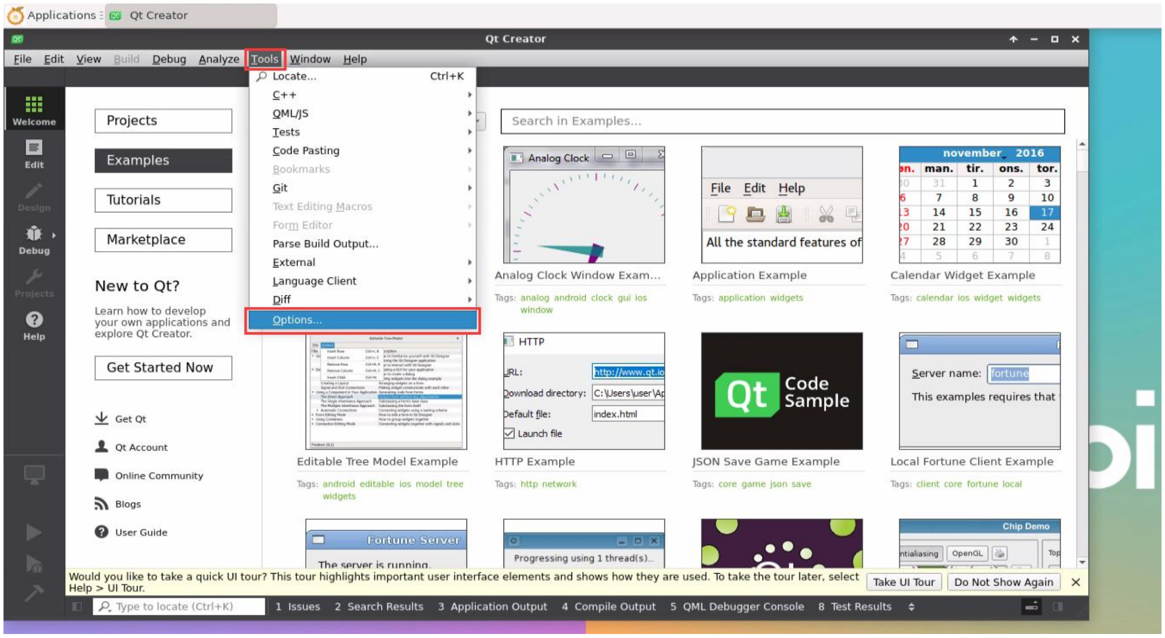


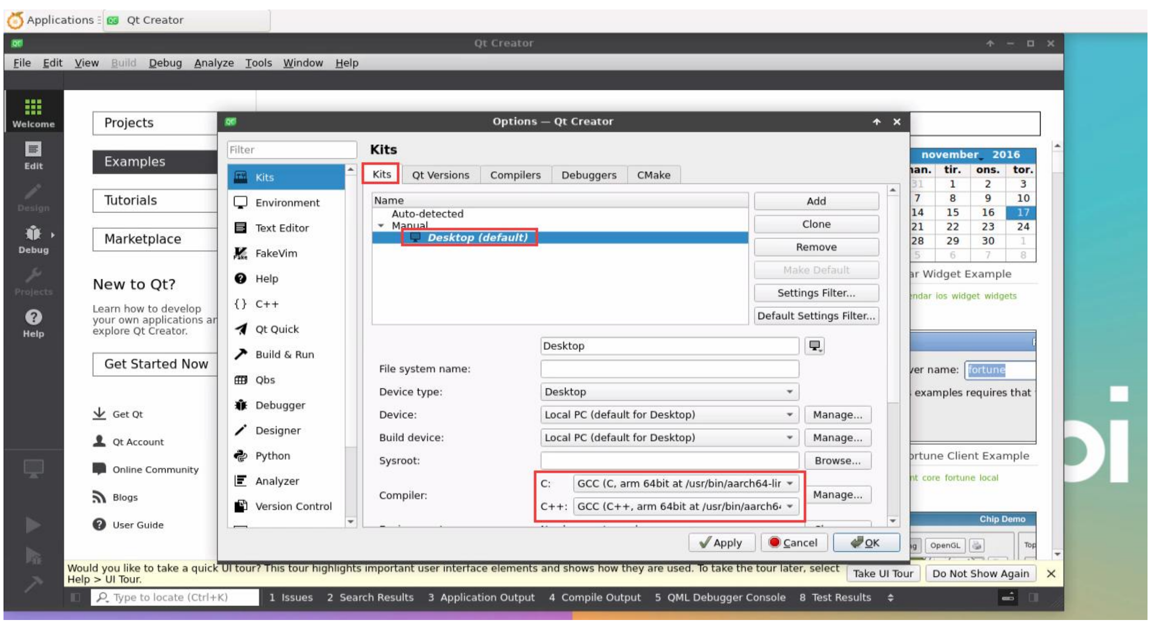
b. Then remove the hook of **ClangCodemodel**



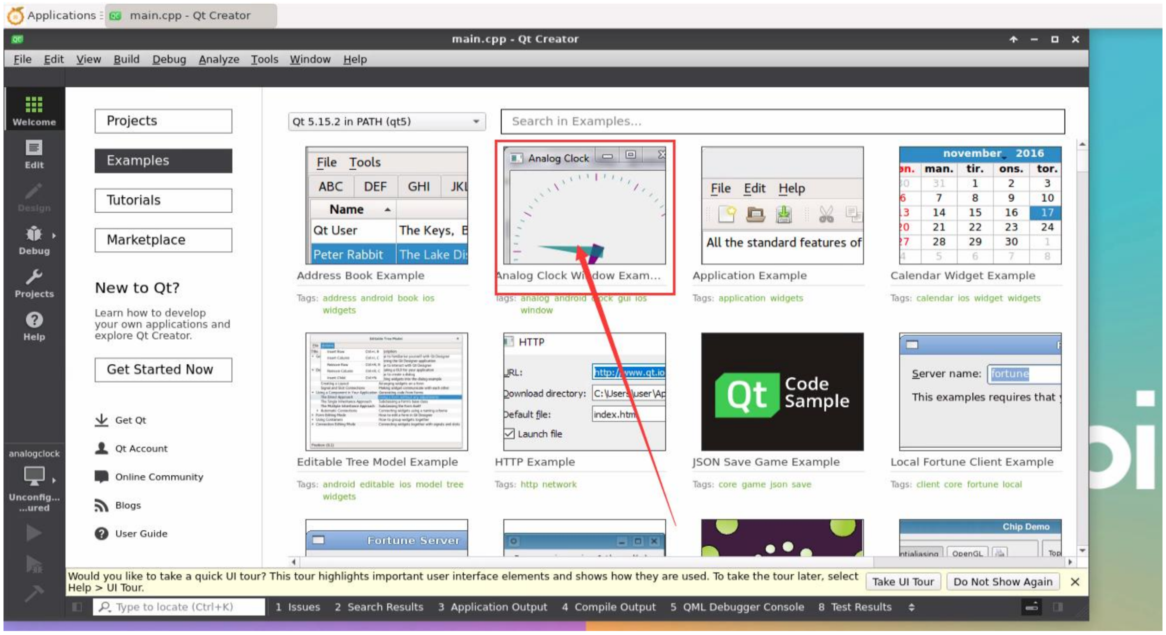

c. **After setting, you need to restart Qt Creator**

d. Then make sure the GCC compiler used by QT Creator, if defaults to Clang, please modify it to GCC

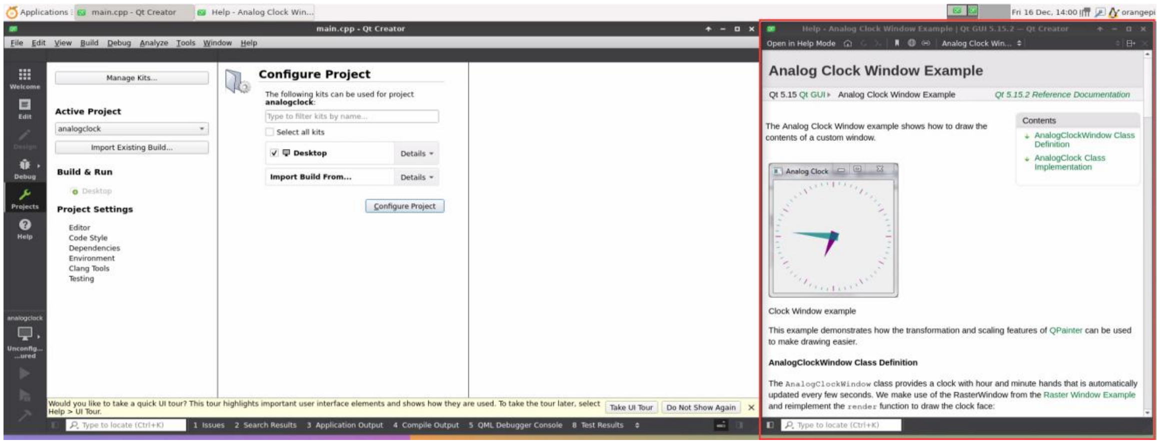


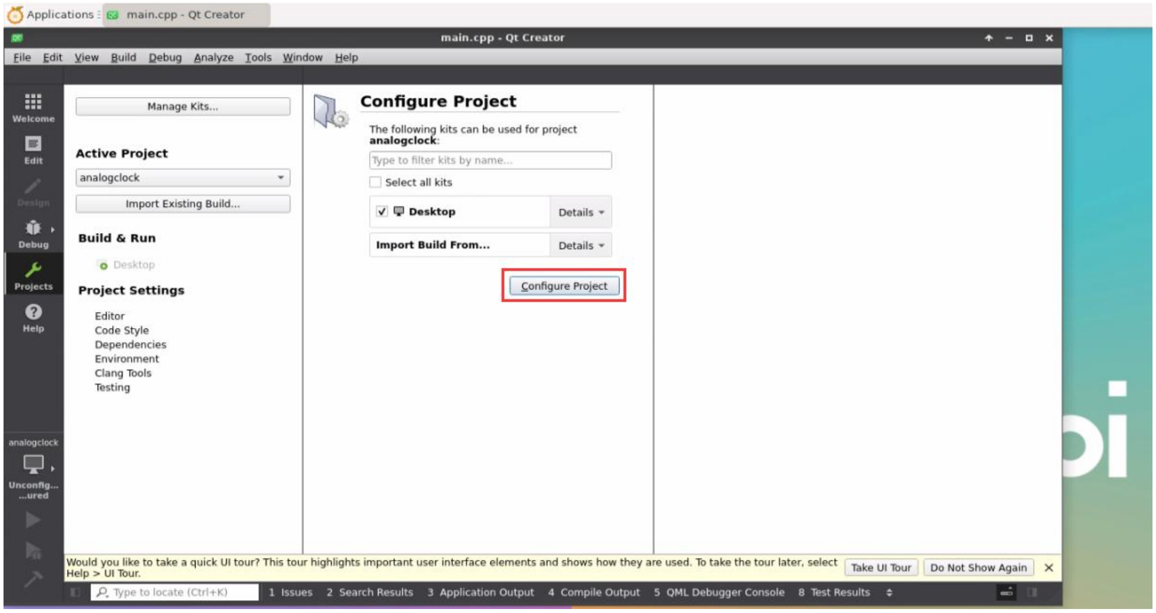

7) Then you can open a sample code



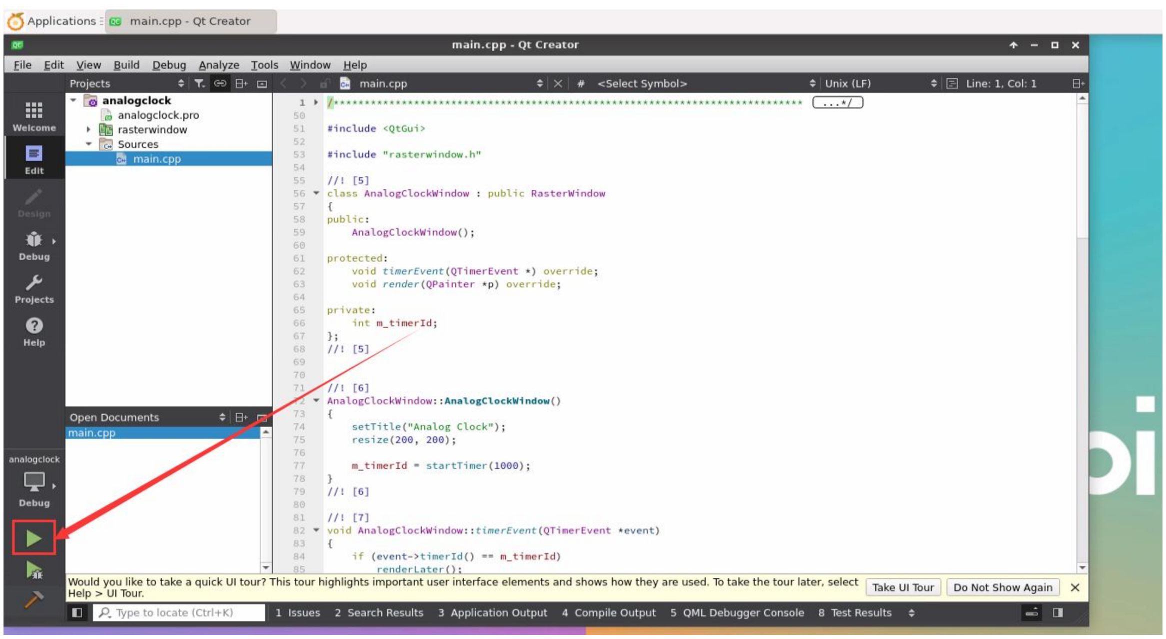
8) After clicking the example code, you will automatically open the corresponding explanation document. You can carefully look at the instructions for usage



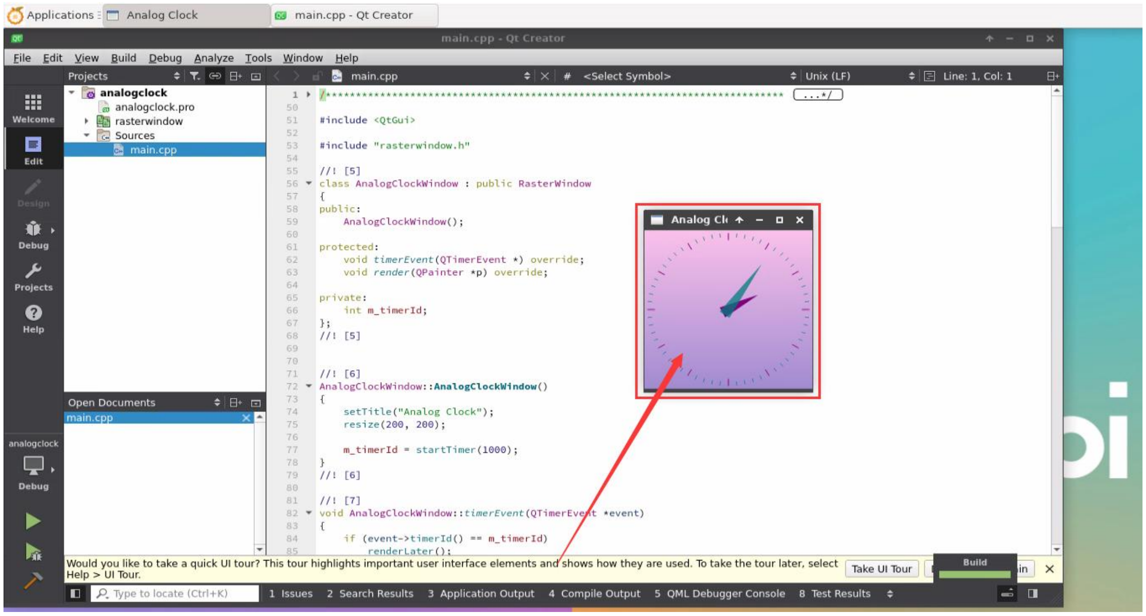
9) Then click **Configure Project**




10) Then click the sample code in the green triangle in the lower left corner




11) After waiting for a while, the interface shown in the figure below will pop up. At this time, it means that QT can compile and run normally




12) Reference materials

https://wiki.qt.io/Install_Qt_5_on_Ubuntu
https://download.qt.io/archive/qtcreator
https://download.qt.io/archive/qt

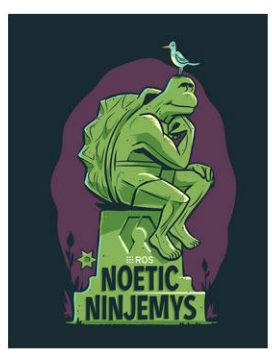
## 3. 29. ROS Installation Method

### 3. 29. 1. Ubuntu20.0 to install ROS 1 noetic

1) The current active version of ROS 1 is shown below. 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) Ros 1 **Noetic Ninjemys** official installation document links are shown below:

**http://wiki.ros.org/noetic/Installation/Ubuntu**

3) ROS **Noetic Ninjemys** official installation document Ubuntu recommended Ubuntu20.04, so please make sure that the system used in the development board is **Ubuntu20.04 desktop version system**

**http://wiki.ros.org/noetic/Installation**


4) Then use the following script to install ros1

```
orangepi@orangepi5:~$ install_ros.sh ros1
```

5) Before using the ROS tool, you need to initialize rosdep first, and then install some system dependencies and core components in some ROS when compiling the source code

**Note that the following commands need to ensure that the development board**

**can access github normally, otherwise an error will be reported due to network problems.**

**Install_ros.sh script will try to modify /etc/hosts and run the following commands automatically. However, this method cannot guarantee that you can access github normally. If install_ros.sh has prompting the following errors after installing ros1, please think other methods to allow the linux system of the development board to access github normally, and then manually run the following manually**

**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**

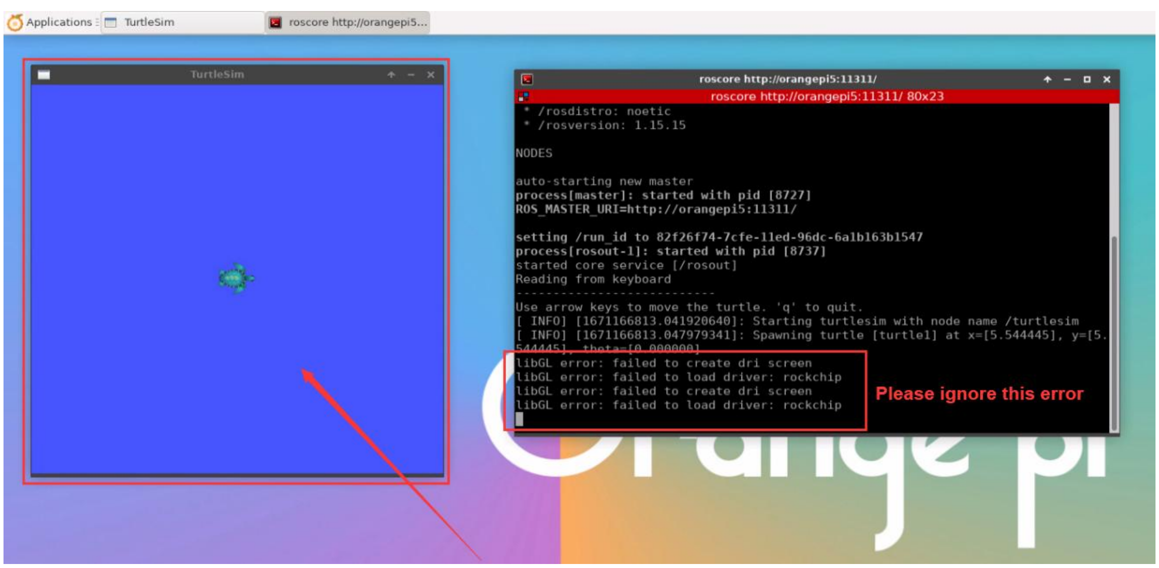
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

        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
```

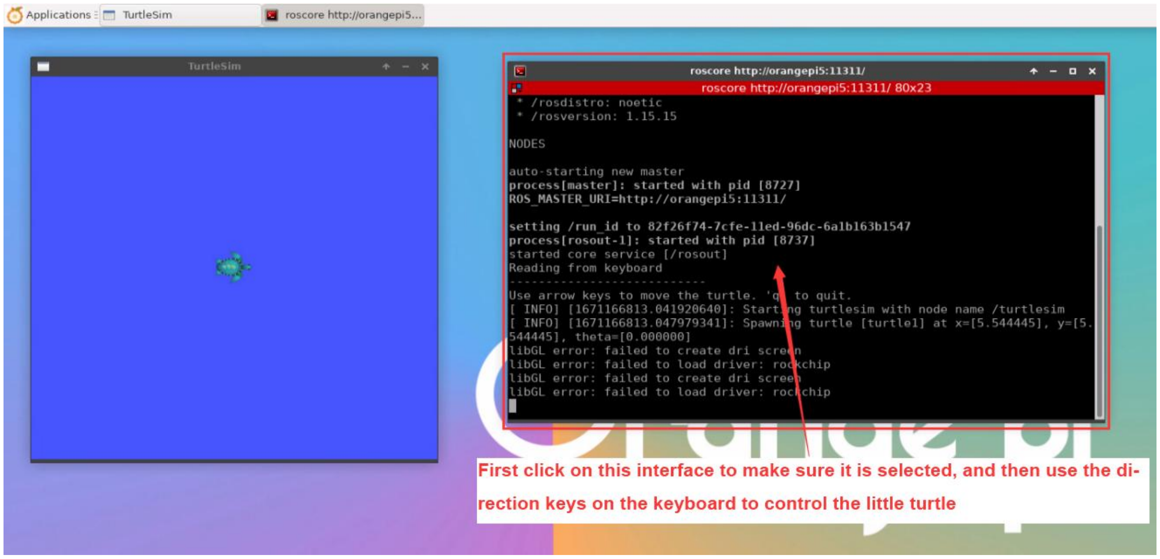
6) Then open a command line window on the **desktop**, and then use the **test_ros.sh** script to start a small turtle routine to test whether the ROS can use normally

```
orangepi@orangepi:~$ test_ros.sh
```

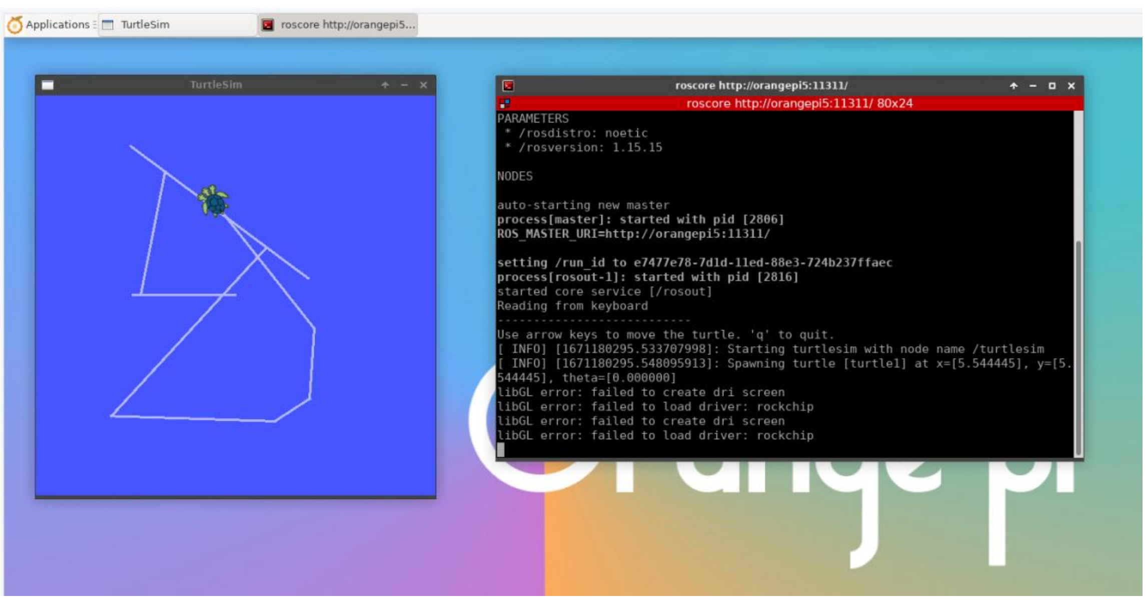
7) After running the **test_ros.sh** script, a small turtle shown in the figure below will pop up




8) Then please keep the terminal window just open at the top




9) Press the direction button on the keyboard at this time to control the small turtles up, down, left and right




## 3. 29. 2. Ubuntu20.04 to install ROS 2 Galactic

1) The current active version of ROS 2 is shown below. The recommended version is **Galactic Geochelone**

**Active ROS 2 distributions**

**Recommended** **Development**


| 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) ROS 2 **Galactic Geochelone**'s official installation document link is shown below:

```
docs.ros.org/en/galactic/Installation.html
http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html
```

3) ROS 2 **Galactic Geochelone**'s official installation document Ubuntu Linux is recommended to use Ubuntu 20.04, so please make sure that the system used in the development board is **Ubuntu20.04 desktop version system**. There are several ways to install ROS 2. The following demonstrates the Ros 2 **Galactic Geochelone** by **Debian packages**

4) You can install ros2 with **install_ros.sh** script

```
orangepi@orangepi:~$ install_ros.sh ros2
```

5) **install_ros.sh** script will automatically run the **ros2 -h** command after ros2 is installed. If you can see the following printing, it means that ros2 installation is complete

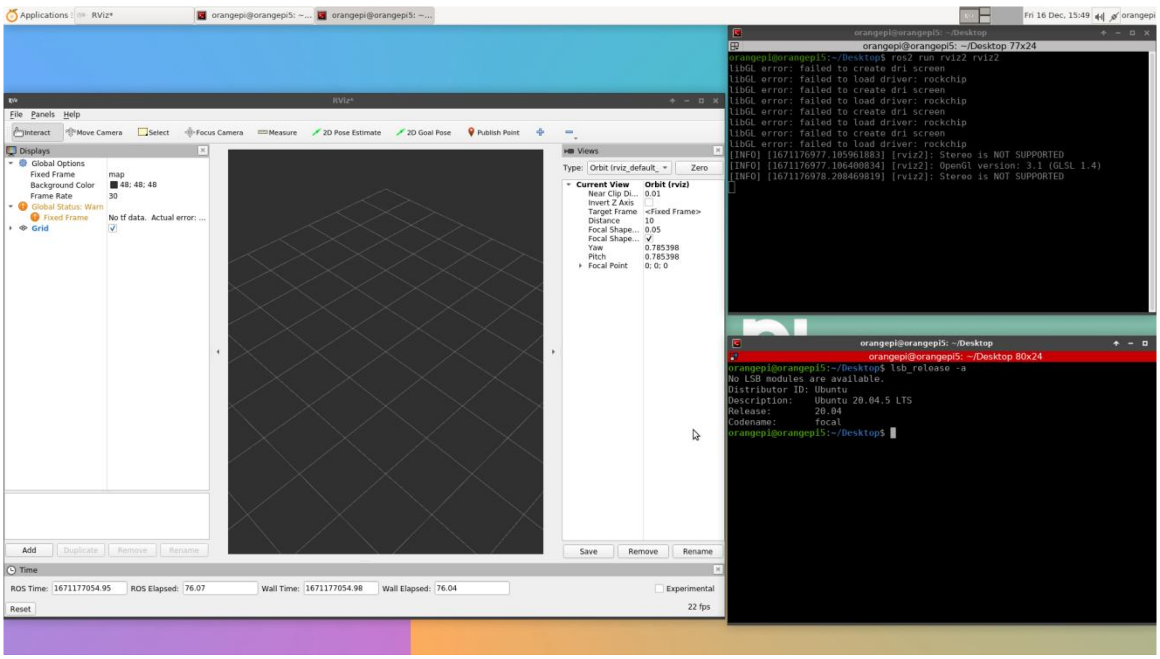
```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action     Various action related sub-commands
  bag        Various rosbag related sub-commands
  component  Various component related sub-commands
  daemon     Various daemon related sub-commands
  doctor     Check ROS setup and other potential issues
  interface  Show information about ROS interfaces
  launch     Run a launch file
  lifecycle  Various lifecycle related sub-commands
  multicast  Various multicast related sub-commands
  node       Various node related sub-commands
  param      Various param related sub-commands
  pkg        Various package related sub-commands
  run        Run a package specific executable
  security   Various security related sub-commands
  service    Various service related sub-commands
  topic      Various topic related sub-commands
  wtf        Use `wtf` as alias to `doctor`

  Call `ros2 <command> -h` for more detailed usage.
```

6) Then you can use **test_ros.sh** script to test whether the ROS 2 is successfully installed. If you can see the printing below, it means that ROS 2 can run normally

```
orangepi@orangepi5:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```




8) How to use ROS, please refer to the document of ROS 2

http://docs.ros.org/en/galactic/Tutorials.html

## 3. 29. 3. Ubuntu22.04 The method of installing ROS 2 Humble

1) You can install ros2 with **install_ros.sh** script

```
orangepi@orangepi:~$ install_ros.sh ros2
```

2) **install_ros.sh** script will automatically run the **ros2 -h** command after ros2 is installed. If you can see the printing below

```
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.
```

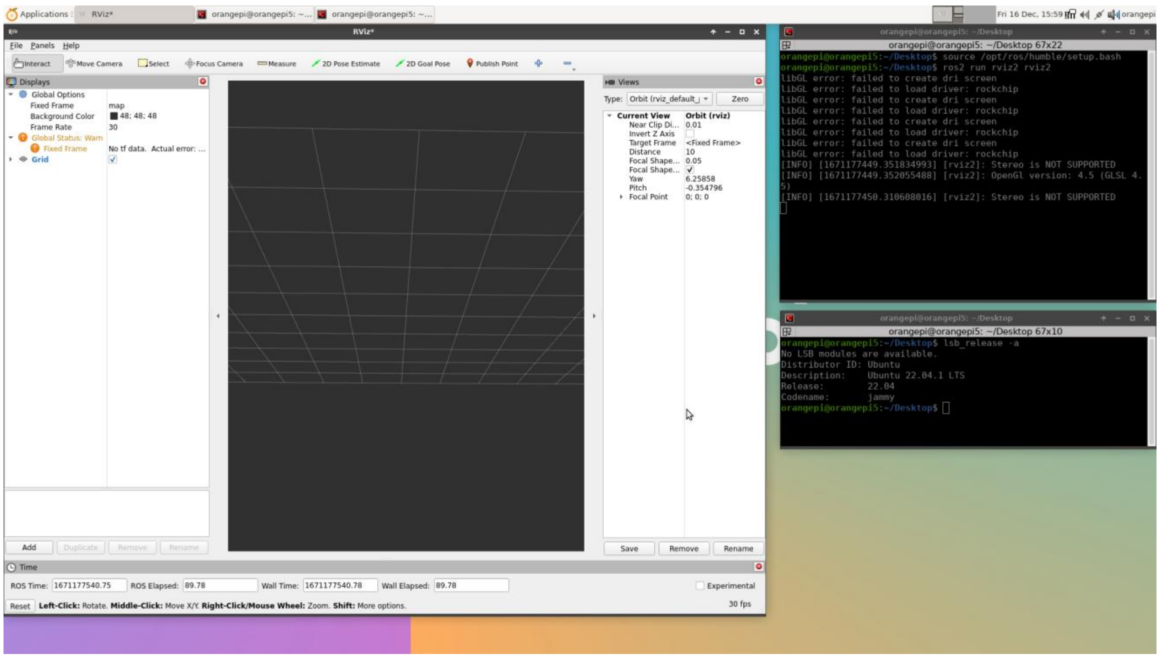
3) Then you can use **test_ros.sh** script to test whether the ROS 2 is successfully installed. If you can see the printing below, it means that ROS 2 can run normally

```
orangepi@orangepi5:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
```

```
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/humble/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



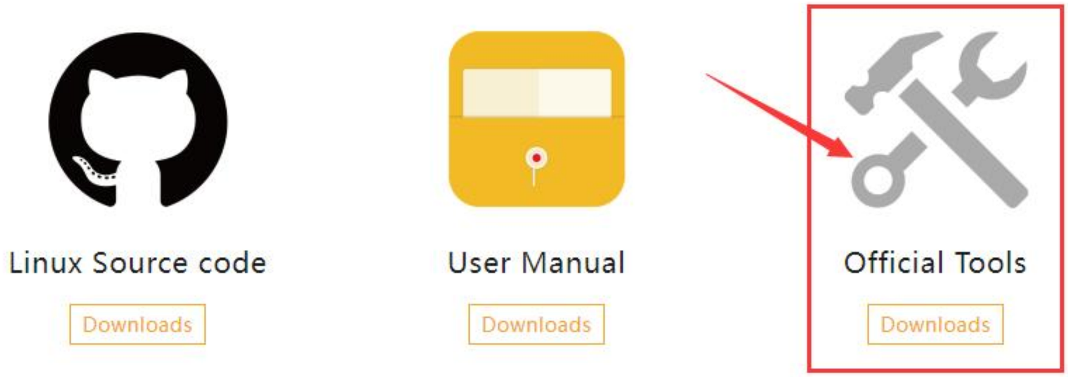

5) Reference document

**http://docs.ros.org/en/humble/index.html**
**http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html**

## 3. 30. The method of installing the kernel header file

1) There are two ways to obtain the kernel header file:

   a. Method 1: Download from the **official tools** of the development board data download page.




b. Method two: Compile the kernel source code using orangepi-build will automatically generate the deb package of the kernel header file. For specific methods, please refer to **4.4. Compile the Linux kernel** one instructions.

2) Then upload the kernel header file deb to the Linux system of the development board. The upload method can refer to the method of **uploading files to the development board Linux system**.instructions chapter

3) Then use the following command to install the kernel header file deb package

**The name of the kernel file deb package needs to be replaced with the actual name, please do not copy it.**

```
orangepi@orangepi:~$ sudo dpkg -i linux-headers-legacy-rockchip-rk3588_1.x.x_arm64.deb
```

4) After installation, you can see the folder where the kernel header file is located under the **/usr/src**

```
orangepi@orangepi:~$ ls /usr/src
linux-headers-5.10.110-rockchip-rk3588
```

5) Then you can write a **hello** kernel module test under the kernel header file

a. First write the code of the hello kernel module, as shown below:

```
orangepi@orangepi:~$ vim hello.c
#include <linux/init.h>
#include <linux/module.h>

static int hello_init(void)
{
```

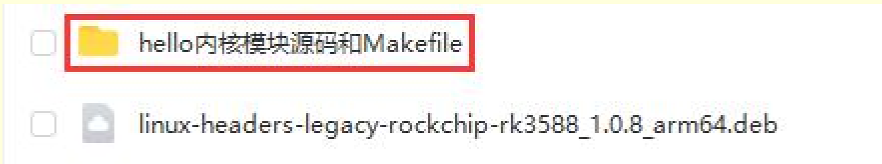
```
        printk("Hello Orange Pi -- init\n");

        return 0;
}
static void hello_exit(void)
{
        printk("Hello Orange Pi -- exit\n");

        return;
}

module_init(hello_init);
module_exit(hello_exit);

MODULE_LICENSE("GPL");
```

b. Then write a makefile file that compiles the Hello kernel module, as shown below:

```
orangepi@orangepi:~$ vim Makefile
ifneq ($(KERNELRELEASE),)
obj-m:=hello.o
else
KDIR :=/lib/modules/$(shell uname -r)/build
PWD   :=$(shell pwd)
all:
	make -C $(KDIR) M=$(PWD) modules
clean:
	rm -f *.ko *.o *.mod.o *.mod *.symvers *.cmd   *.mod.c *.order
endif
```

c. Then use the make command to compile the hello kernel module, and the output of the compilation process is shown below：

**If you compile the code you copy here, if you have any problems, go to the official tool to download the source code test.**




```
orangepi@orangepi:~$ make
make -C /lib/modules/5.10.110-rockchip-rk3588/build M=/home/orangepi modules
make[1]: Entering directory '/usr/src/linux-headers-5.10.110-rockchip-rk3588'
  CC [M]  /home/orangepi/hello.o
  MODPOST /home/orangepi/Module.symvers
  CC [M]  /home/orangepi/hello.mod.o
  LD [M]  /home/orangepi/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.10.110-rockchip-rk3588'
```

d. After the compilation, the **hello.ko** kernel module will be generated

```
orangepi@orangepi:~$ ls *.ko
hello.ko
```

e. Use the **insmod** command to insert the **hello.ko** kernel module into the kernel

```
orangepi@orangepi:~$ sudo insmod hello.ko
```

f. Then use the **demsg** command to view the output of the **hello.ko** kernel module. If you can see the output instructions below, the **hello.ko** kernel module is loaded correctly

```
orangepi@orangepi:~$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
```

g. Use the **rmmod** command to uninstall the **hello.ko** kernel module

```
orangepi@orangepi:~$ sudo rmmod hello
orangepi@orangepi:~$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
[ 3173.800892] Hello Orange Pi -- exit
```

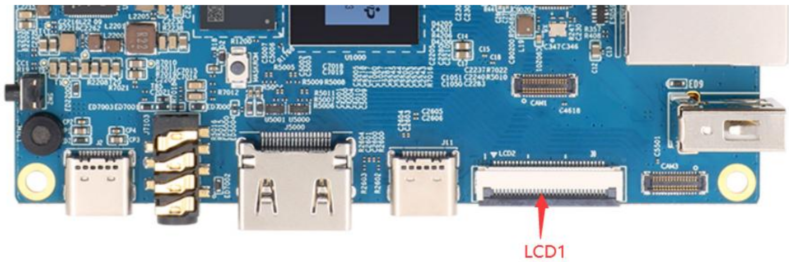
## 3. 31. How to use the 10.1-inch MIPI LCD screen

### 3. 31. 1. Open the 10.1 -inch MIPI LCD screen configuration method

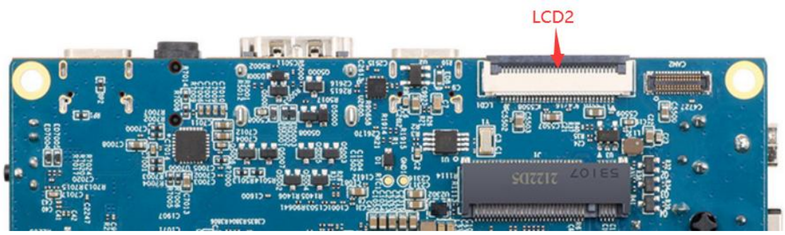
1) The Linux image defaults to the configuration of the mipi lcd screen by default. If you need to use the MIPI LCD screen, you need to open it manually.

2) There are two interfaces of the mipi lcd screen on the development board, we define：

a. The position of the lcd1 interface is:




b. The position of the lcd2 interface is:

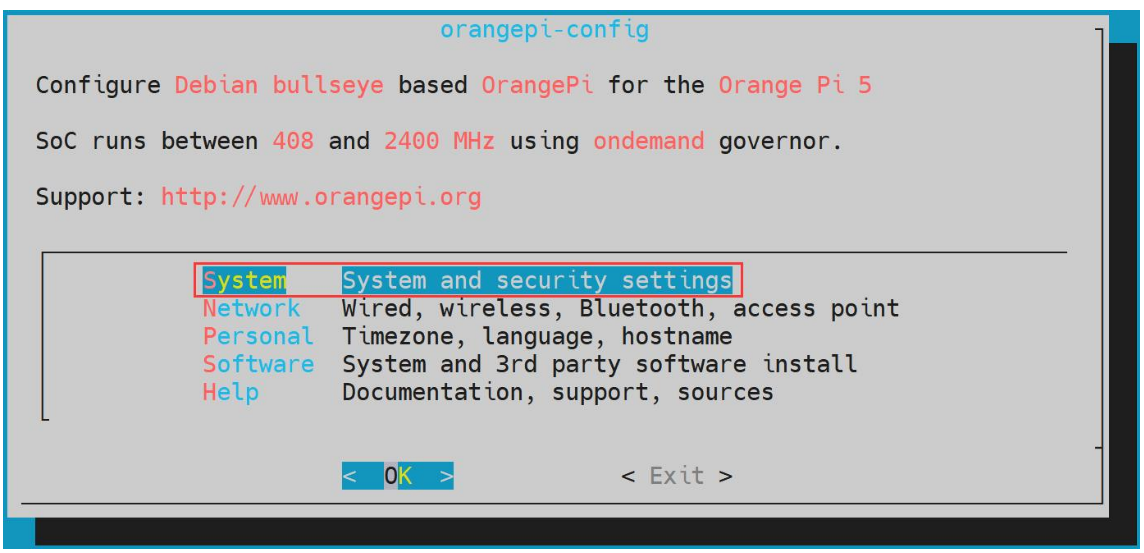



3) The steps of opening the mipi lcd configuration are shown below:

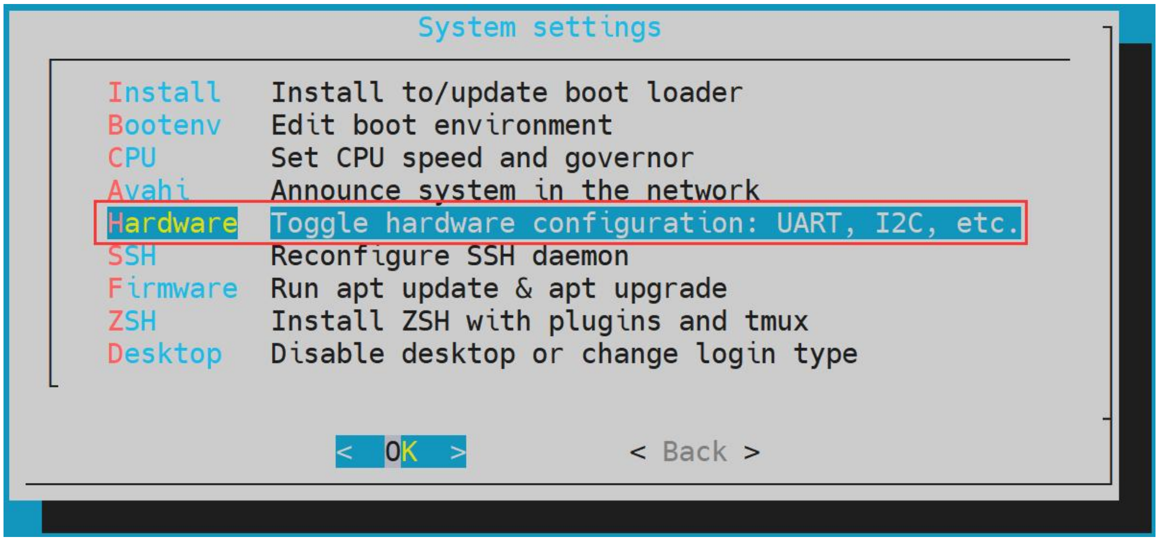
a. First run the **orangepi-config**, ordinary users remember to add **sudo** permissions

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

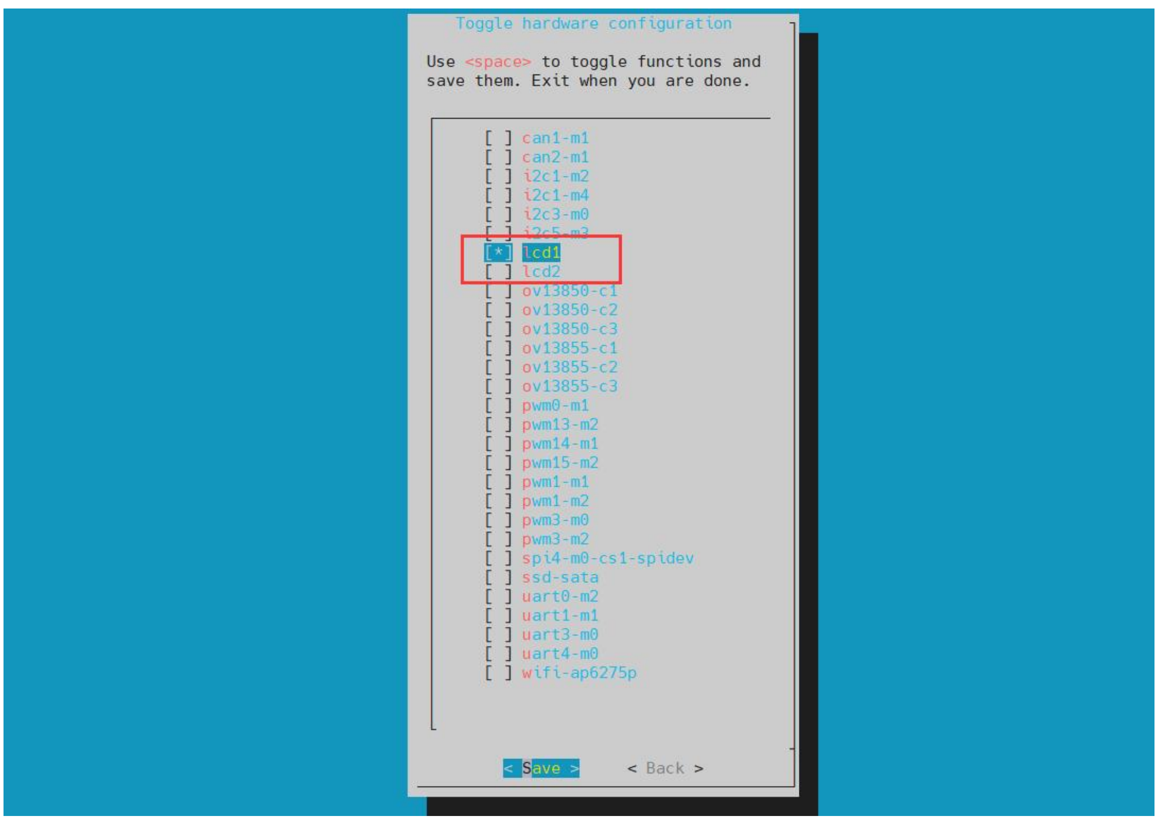
b. Then choose **System**




c. Then choose **Hardware**

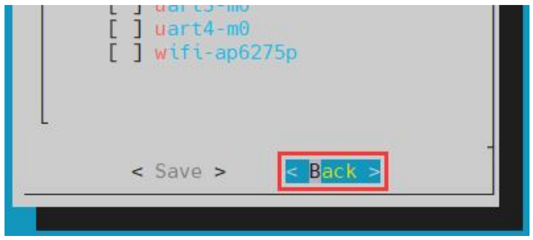



d. Then use the direction keys of the keyboard to position lcd1 or lcd2 (to open which one if you want to use it, and two screens can be opened at the same time), then use the **space** to select

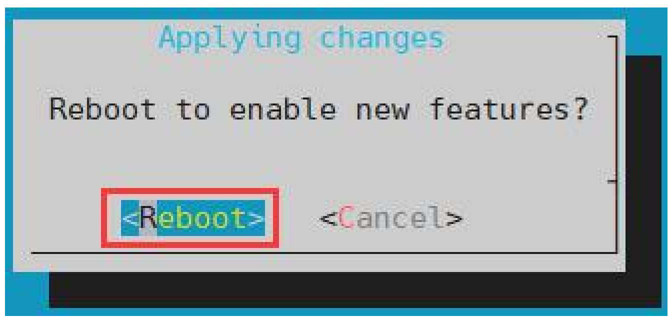



e. Then select **<Save>** to Save


f. Then choose **<Back>**




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




**The above settings will eventually add overlays=lcd1 or overlays=lcd2 or overlays=lcd1 lcd2 in /boot/orangepiepienv.txt. After setting, you can check it first. If this configuration does not exist, then there is a problem with settings.**

**If you feel that using orangepi-config is more troublesome, you can also use vim editors to open the /boot/orangepienv.txt, and then add overlays=lcd1 or overlays=lcd2 or overlays=lcd1 lcd2.This configuration is also possible.**

orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep "lcd"**
**overlays=lcd1 #Example configuration**

4) After starting, you can see the display of the lcd screen as shown below (default vertical screen):

### 3.31.2. The method of the server version of the image rotation display direction

1) Add **extraargs = fbcon = rotate** to the **/Boot/orangepienv.txt**: You can set the direction of the Linux system displayed in the direction of the rotating direction.Among them, **fbcon = rotate:** The following numbers can be set to be set to：

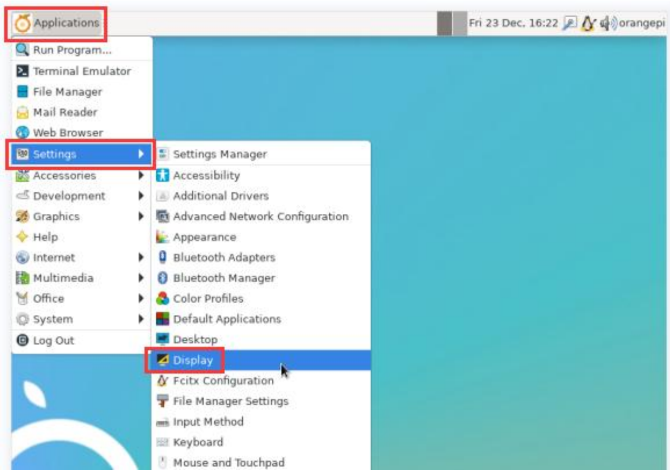
a. 0: Normal screen (default vertical screen)
b. 1: Turn 90 degrees clock
c. 2: Flip 180 degrees
d. 3: Turn to 270 degrees clock

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=lcd1
extraargs=fbcon=rotate:3
```

2) Then **restart** the Linux system to see the direction of the lcd screen displayed has been spinned

### 3.31.3. The method of rotating and touching the desktop image

1) First open **Display** settings in the Linux system



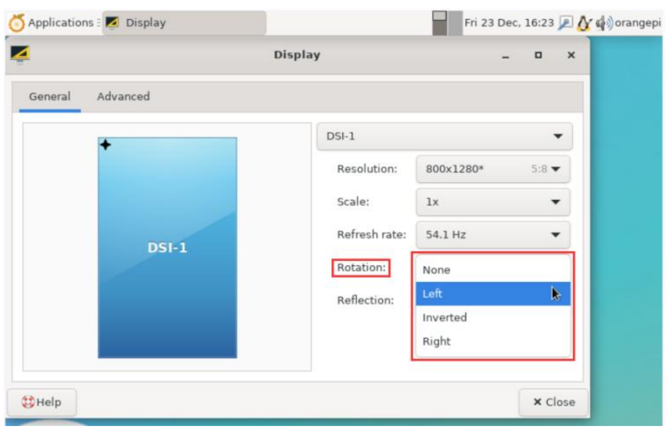

2) Then select the direction you want to rotate in the **Rotation**

a. **None**: Not rotate

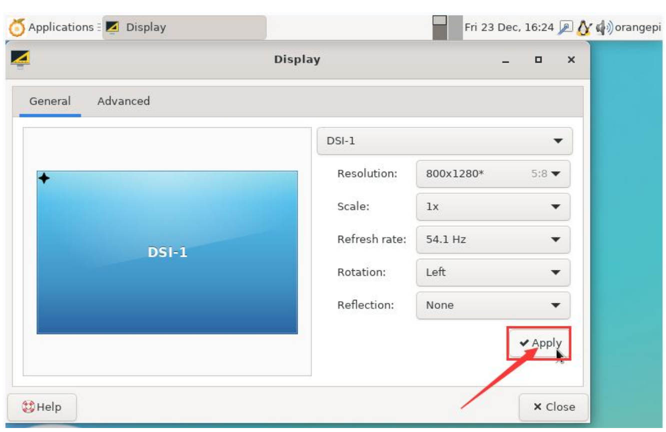
b. **Left**: Rotate 90 degrees to the left

c. **Inverted**: Flipting up and down, equivalent to rotating 180 degrees

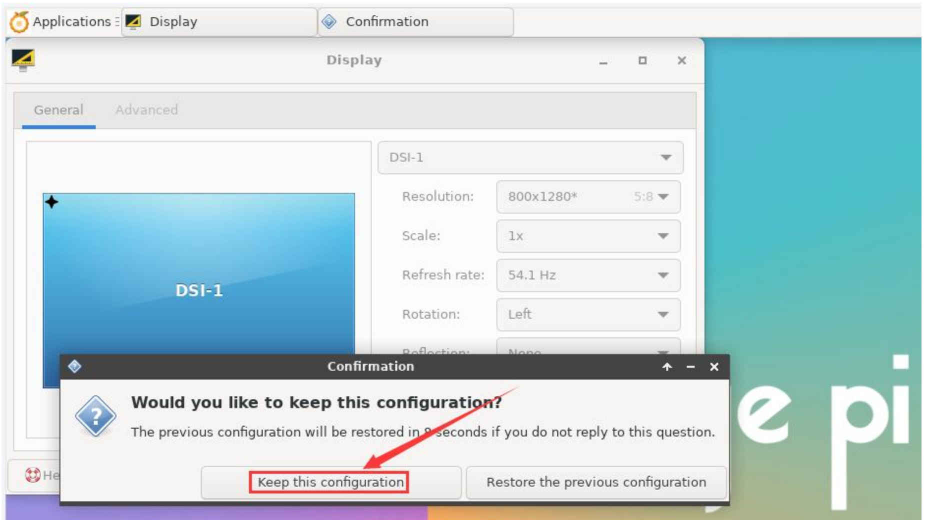
d. **Right**: Rotate 90 degrees to the right




3) Then click **Apply**



4) Then choose **Keep this configuration**




5) At this time, the screen display has been rotated, and then the **Display** program is turned off.

6) The above steps will only select the display direction, and it will not rotate the direction of touch. Use **set_lcd_rotate.sh** script to rotate the direction of touch. After this script is set, it will be automatically restarted, and then you can test whether the touch can be used normally.

a. **None**: Not rotate

```
orangepi@orangepi:~$ set_lcd_rotate.sh none
```

b. **Left**: Rotate 90 degrees to the left

```
orangepi@orangepi:~$ set_lcd_rotate.sh left
```

c. **Inverted**: Flipting up and down, equivalent to rotating 180 degrees

```
orangepi@orangepi:~$ set_lcd_rotate.sh inverted
```

d. **Right**: Rotate 90 degrees to the right

```
orangepi@orangepi:~$ set_lcd_rotate.sh right
```

**set_lcd_rotate.sh script mainly does four things:**

1. **Rotate the direction displayed by framebuffer**
2. **The direction of rotation touch**
3. **Turn off logo**

**4. Restart the system**

**The direction of the rotation touch is achieved by adding Option "TransformFormationMatrix""x x x x x x x x x" in /usr/share/X11/xorg.conf.d/40-libinput.conf . The "x x x x x x x x x" is different in different directions**

7) Touch rotation reference materials

```
https://wiki.ubuntu.com/X/InputCoordinateTransformation
```

## 3. 32. Instructions for Turn off the logo use instructions

1) Open the LOGO only displayed in the desktop version of the system

2) logoSet up **bootlogo** variable to **false** in **/boot/orangepienv.txt** to turn off the switch to the logo

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=false
```

3) Set the **bootlogo** variable to **true** in the **/boot/orangepiEnv.txt** turn the turn -off logo

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
```

4) The position of the logo picture in the LINUX system is

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

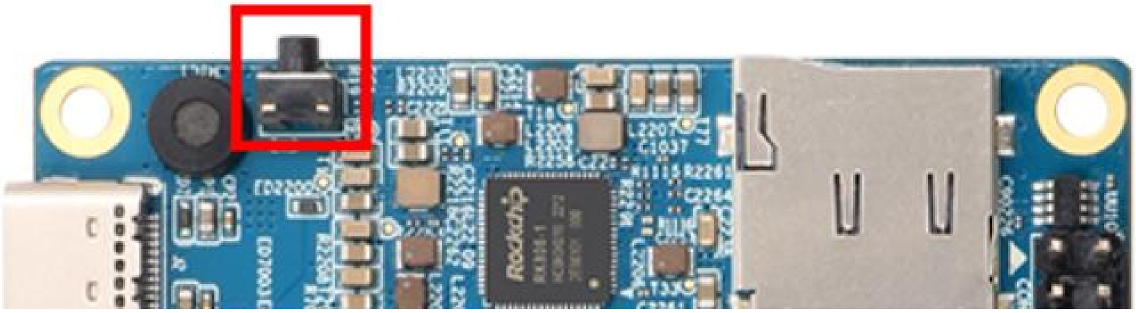
## 3. 33. The method of shutting down and restarting the development board

1) In the process of running the linux system, if the Type-C power supply is directly out of power, it may cause the file system to lose certain data or damage. Therefore, please use the **poweroff** command to turn off the linux system of the development board before

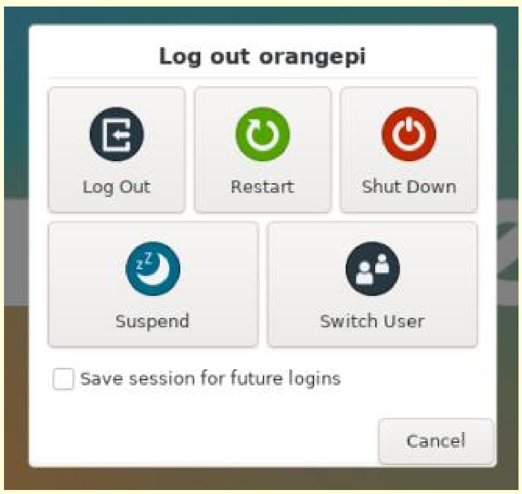
power off, and then then again Unplug the power supply

```
orangepi@orangepi:~$ sudo poweroff
```

2) In addition, the development board is equipped with a switch button, and you can also press the switch button on the development board to turn off。

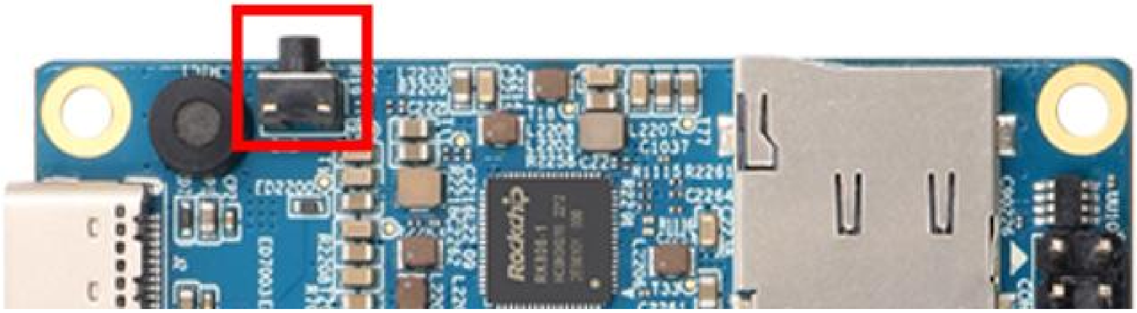


**Note that the linux desktop system will pop up the confirmation box after pressing the buttons. You need to click the Shut Down option before turning off.**




**Also, please note that the Suspend function shown in the figure above is unavailable.**

3) After shutting down, press the switch button on the development board to turn on.



4) Restart the command of the Linux system

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

# 4. Linux SDK——orangepi-build instructions for use

## 4. 1. Compile system requirements

**We can compile the linux image of the development board in the x64 computer, or compile the linux image of the development board in the Ubuntu22.04 system of the development board. Please choose one by one according to your preference.**

**If you use orangepi-build to compile the Linux image in the Ubuntu22.04 system of the development board to compile the Linux image, please do heat dissipation, especially when SSD startup. If the heat dissipation is not done well, it is prone to error in the file system running.**

### 4. 1. 1. Use the development board Ubuntu22.04 system to compile

1) Linux SDK, **orangepi-build**, supports the upper operation of the development board's **Ubuntu 22.04** (other systems have not been tested), so before downloading Orangepi-Build, first make sure that the Ubuntu version installed on the development board is Ubuntu 22.04. The command of the Ubuntu version installed on the development board is shown below. If the Release field is not **22.04**, it means that the 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 code such as kernel and U-Boot is stored on GitHub, it is very important to ensure that the development board can download the code from GitHub normally when compiling image.**

### 4. 1. 2. Use X64's Ubuntu22.04 computer to compile

1) Linux SDK, **orangepi-build**, supports running on a computer with **Ubuntu 22.04**, so before downloading Orangepi-Build, first make sure that the Ubuntu version of your computer installed is Ubuntu 22.04. Check the command of the Ubuntu version installed by the computer as shown below. If the release field is not **22.04**, it means that the 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 a Windows system, there is no computer with Ubuntu 22.04, you can consider using **VirtualBox** or **VMware** to install a Ubuntu 22.04 virtual machine in the Windows system. But please note that Orange-Build is compiled on the WSL virtual machine. Because Orangepi-BUILD has not been tested in the WSL virtual

machine, it cannot be ensured that it can use Orangepi-Build in WSL normally.

3) Ubuntu 22.04 **amd64** version installation image download address is:

**https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso**

or

**https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.1-desktop-amd64.iso**

4) After installing Ubuntu 22.04 in the computer or virtual machine, please set up the software source of Ubuntu 22.04 as a Tsinghua source first, otherwise it is easy to make mistakes due to network reasons when installing the software later.

a. The method of replacing Tsinghua source refer to the instructions of this webpage

**https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/**

b. Note that the Ubuntu version needs to be switched to 22.04


c. The content of the **/etc/apt/sources.list** file that needs to be replaced is

```
test@test:~$ sudo mv /etc/apt/sources.list /etc/apt/sources.list.bak
test@test:~$ sudo vim /etc/apt/sources.list
# The source code image is annotated by default to improve the speed of APT Update. If necessary, you can cancel the annotation by yourself
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
```

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# Pre -release software sources, it is not recommended to enable
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

d. After replacement, you need to update the package information and make sure there is no error

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

**e. In addition, because the source code such as kernel and U-Boot is stored on GitHub, please make sure that the computer can download the code normally when compiling images, which is very important.**

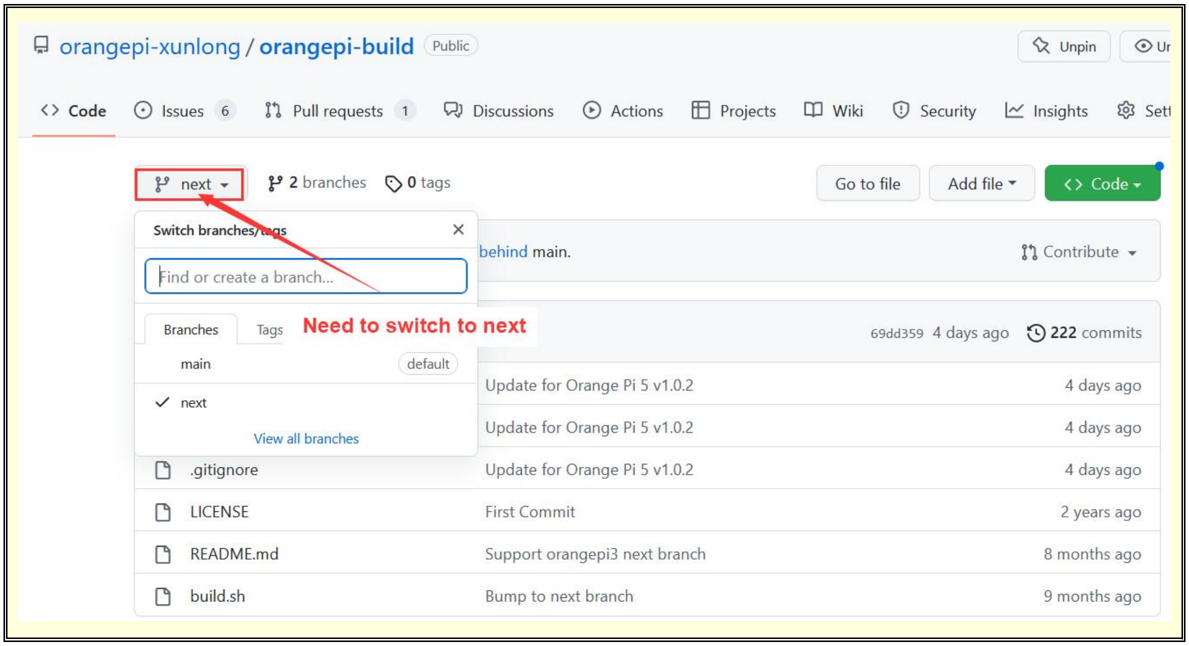
## 4. 2. Get the source code of linux sdk

### 4. 2. 1. Download orangepi-build from github

1) Linux sdk actually refers to the Orange-Build code. Orangepi-Build is modified based on the Armbian Build compilation system. OrangePi-Build can compile multiple version of Linux images. First download the code-build code, and the command is shown below:

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y git
test@test:~$ git clone https://github.com/orangepi-xunlong/orangepi-build.git -b next
```

**Note that the Orange Pi 5 development board needs to download the next branch source code of orangepi-build. The Git Clone command above needs to specify the branch of orangepi-build source code to next.**




**Download orangepi-build code through the git clone command is the user name and password that does not need to enter the GitHub account (the same is the same for downloading other code in this manual). Names and passwords are usually input errors in the address input of OrangePi-Build repository behind Git Clone. Please check the command whether there is any error in the command, instead of thinking that we have forgotten the username and password of the github account here.**

2) The u-boot and linux kernel versions currently used in the development board are shown below

| Branch | u-boot version | Linux Kernel version |
|---|---|---|
| legacy | u-boot 2017.09 | linux5.10 |

**The branches mentioned here are not the same thing as orangepi-build source code, please do not confuse. This branch is mainly used to distinguish different kernel source code.**

**At present, the linux5.10 bsp kernel provided by RK is defined as the legacy branch. If the main line is supported in the future, a Current branch will be added.**

3) orangepi-build will contain the following files and folders after downloading

a. **build.sh**: Compile the startup script

b. **external**: Including configuration files, specific scripts, and source code of some programs, etc.

c. **LICENSE**: GPL 2 License file

d. **README.md**: orangepi-build description file

e. **scripts**: General script compiled Linux image

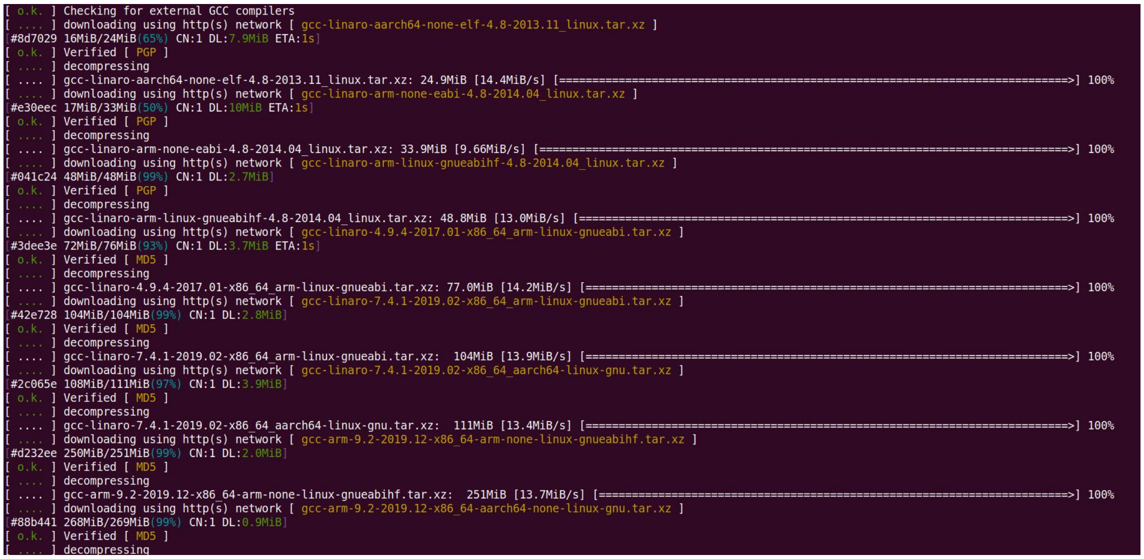
```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

**If the orangepi-build code downloaded from github, you may find that after downloading, you may find that orangepi-build does not include the source code of the U-Boot and Linux kernels, nor does it compile the U-Boot and Linux kernel needs to be used. Chain, this is normal, because these things are stored in other separate GitHub warehouses or some servers (the address will be described in detail below). orangepi-build will specify the address of the U-Boot, Linux kernel and cross compilation tool chain in the script and configuration file. When running Orange-Build, when it is found that there are no these things in the local area, it will automatically download the corresponding places.**

### 4. 2. 2. Download the cross compilation tool chain

**Only by using orangepi-build to compile images in the x64 computer, the cross compile tool chain is downloaded. The Linux image compiled to compile the development board in the development board's Ubuntu22.04 will not download the cross compilation tool chain. At this time, orangepi-build/toolchains will be an empty folder.**

1) orangepi-build will automatically download the cross-compile tool chain in the **toolchains** folder when running for the first time. After running Orangepi-Build's build.sh script every time, it will check whether the cross compile tool chain in **toolchains** exists If there is no existence, it will start downloading again. If it exists, it will be used directly and will not be downloaded repeatedly.




2) The image website of the cross-compilation tool chain in China is the open source software image station of Tsinghua University

**https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/**

3) After the download of **toolchains** is downloaded, it will contain multiple versions of cross compilation tool chain. The development board will only use two of them.

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

4) The cross compilation tool chain used by compiling the linux kernel source code is

a. linux5.10

```
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

5) The cross compilation tool chain used by compiling u-boot source code is

a. v2017.09

```
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
```

### 4. 2. 3. orangepi-build complete directory structure description

1) orangepi-build warehouse does not include Linux kernel, u-boot source code, and cross compilation tool chain after downloading.

a. The git warehouse stored in the Linux kernel source code is shown below:

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk3588
```

b. The git warehouse stored in the u-boot source code is shown below:

```
https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2017.09-rk3588https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2018.05-sun50iw9
```

**If you are not familiar with orangepi-build, it is unclear to compile the detailed process of compiling the Linux kernel and U-Boot, please do not download the above Linux kernel and U-Boot source code to compile operations, because orangepi-build compile scripts and configuration files Some adjustments and optimizations of U-Boot and Linux will be made. If orangepi-build is not used to compile U-Boot and Linux, it may encounter problems that fail or cannot start.**

2) orangepi-build will download the cross compile tool chain, u-boot, and Linux kernel source code when running the first run. After successfully compiling the Linux image, there are files and folders that can be seen in orangepi-build.

a. **build.sh**: Compile the startup script
b. **external**: Including the configuration file, a specific function script, and the source code of some programs need to be used.The rootfs compression packet that has been cached during the compile image is also stored in externa
c. **kernel**:The source code stored in the Linux kernel, which is **orange-pi-5.10-rk3588** folder is stored in the RK3588/RK3588S series of the kernel source code of the legacy branch. Please do not modify the name of the file folder of the kernel source. The kernel source code will be downloaded again when the compilation system is running
d. **LICENSE**: GPL 2 License file
e. **README.md**: orangepi-build description file
f. **output**: Store the compiled u-boot, Linux and other deb bags, compile logs, and compile-generated images and other files
g. **scripts**: General script compiled Linux image

h. **toolchains**: Staying cross compilation tool chain

i. **u-boot**: The source code stored in u-boot, which is called **v2017.09-rk3588**, is stored in the RK3588/RK3588S series of the u-boot source code of the legacy branch. Modified, the U-Boot source code will be downloaded again when the compilation system is running

j. **userpatches**: Store the configuration file needed to be used in the compilation script

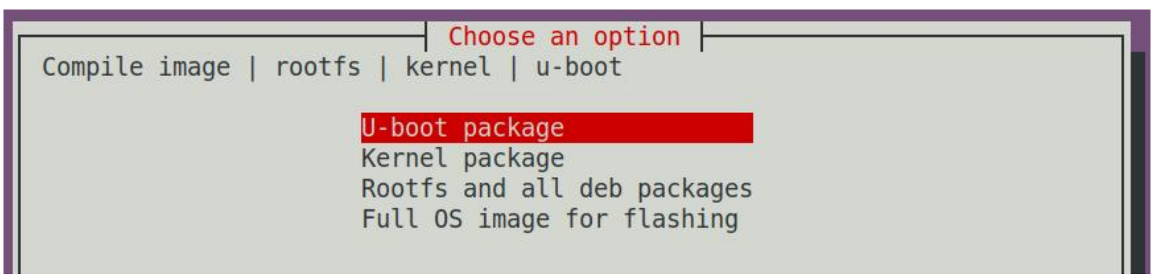
```
test@test:~/orangepi-build$ ls
build.sh    external    kernel    LICENSE    output    README.md    scripts
toolchains   u-boot   userpatches
```

## 4. 3. Compile u-boot

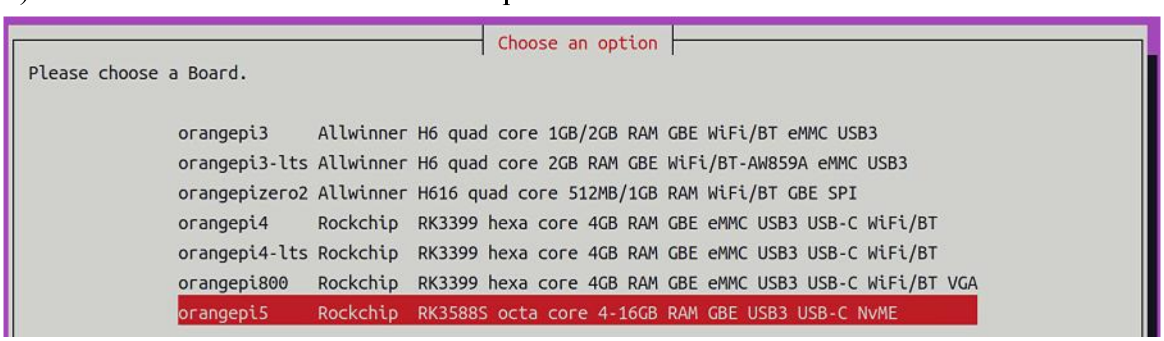
1) Run the build.sh script, remember to add Sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package**, then press Enter




3) Then select the model of the development board




4) Then I will start to compile u-boot, and the part of the information prompted during compilation will be as follows

a. u-boot Source code version

```
[ o.k. ] Compiling u-boot [ v2017.09 ]
```

b. The version of the cross compile tool chain

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 7.4.1 ]
```

c. The path of the u-boot deb package generated by compilation

```
[ o.k. ] Target directory [ orangepi-build/output/debs/u-boot ]
```

d. The package name of the compiled u-boot deb package

```
[ o.k. ] File name [ linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb ]
```

e. Time to compile

```
[ o.k. ] Runtime [ 1 min ]
```

f. Repeat the command of the u-boot. Use the following commands to select the u-boot directly through the graphic interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi5 BRANCH=legacy
BUILD_OPT=u-boot KERNEL_CONFIGURE=no ]
```

5) View compiled u-boot deb package

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb
```

6) The file contained in the deb package generated by the generated u-boot is shown below

a. Use the following commands to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ $ dpkg -x \
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb .      (Pay attention to the last
one".")
test@test:~/orangepi_build/output/debs/u-boot$ ls
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb    usr
```

b. The file after decompression is shown below

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-legacy-orangepi5_1.0.2_arm64
    │   ├── idbloader.img
    │   ├── rkspi_loader.img
```

```
│   └── u-boot.itb
└── u-boot
    ├── LICENSE
    ├── orangepi_5_defconfig
    └── platform_install.sh

3 directories, 6 files
```

7) When orangepi-bulid compile system compile u-boot source code, the source code of u-boot is first synchronized with the u-boot source code of the github server. Therefore, if you want to modify the source code of the u-boot, you need to close the download and update function of the source code.（**You need to compile the u-boot once to close this feature, otherwise you will not be prompted to find the source code of the u-boot. If it is downloaded from the source code compression package from Google Drive, there is no such problem because the source code of U-Boot is all the source code of u-boot. It has been cached.**），Otherwise, the modifications will be restored, and the method is as follows.：

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

8) When debugging the u-boot code, you can use the following method to update U-Boot in the Linux image for testing.

a. Upload the compiled U-Boot DEB package to the Linux system of the development board.

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the installed u-boot deb package

```
root@orangepi:~# apt purge -y linux-u-boot-orangepi5-legacy
```

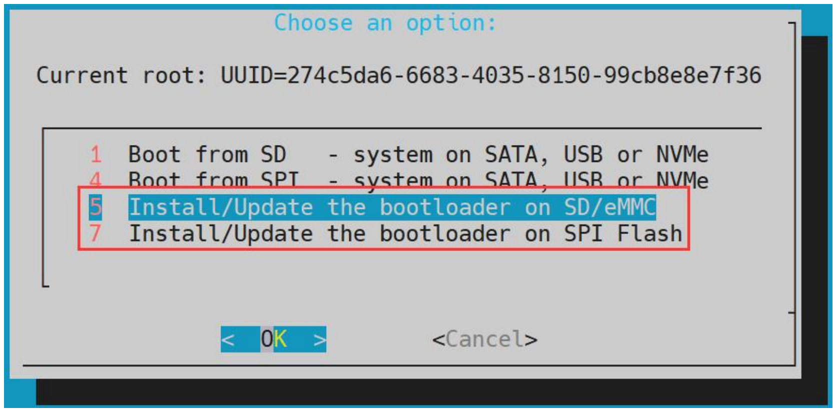
c. Install the new u-boot deb package just uploaded.

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi5_1.0.2_arm64.deb
```

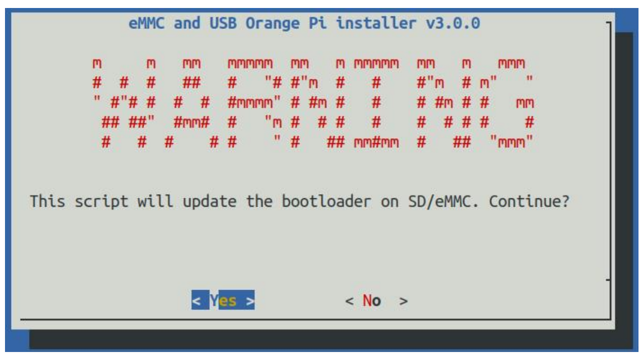
d. Then run nand-sata-installl script

```
root@orangepi:~# nand-sata-install
```

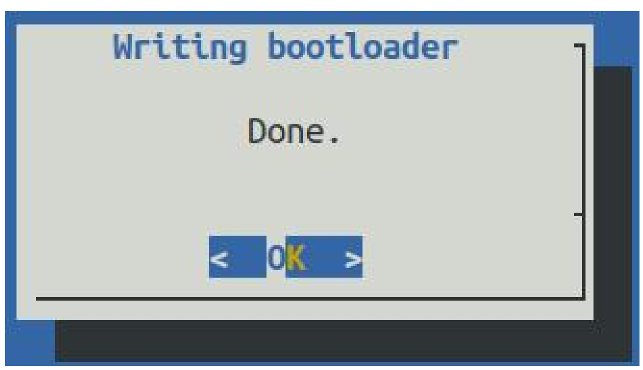
e. Then select 5 **5 Install/Update the bootloader on SD/eMM** to update the U-Boot or **7 Install/Update the bootloader on SPI Flash** in the TF card to update the U-Boot in SPI Flash




f. After pressing the Enter key, a warring will pop up first




g. Press the carbage key to update the U-Boot, and the following information will be displayed after the update.




h. Then you can restart the development board to test whether the modification of the U-Boot has taken effect

9) Other useful information

a. u-boot 2017.09 Source code, the Defconfig configuration file used in the development board is

orangepi-build/u-boot/v2017.09-rk3588/configs/orangepi_5_defconfig

b. u-boot 2017.09 Source code, the development board uses a dts file.

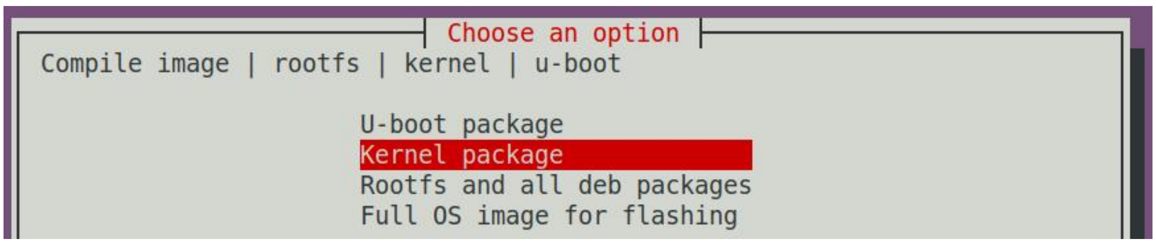
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3588s-orangepi-5.dts

## Compile the linux kernel

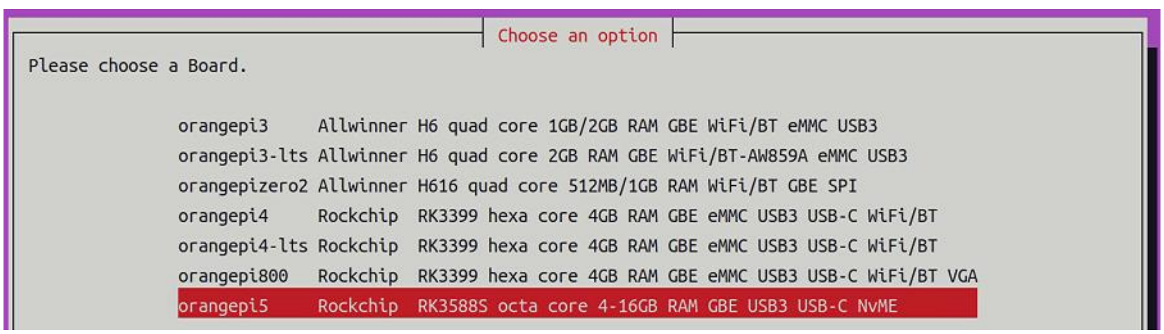
1) Run the build.sh script, remember to add Sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

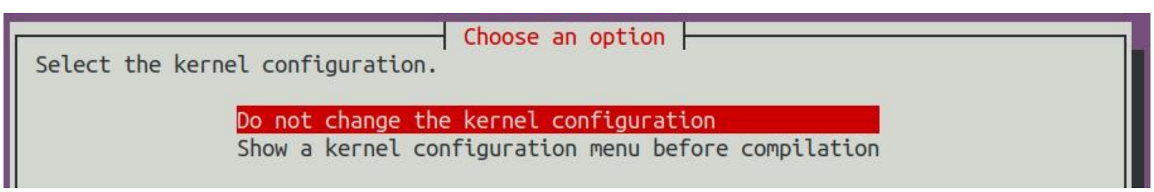
2) Select **Kernel package** and press Enter.




3) Then select the model of the development board

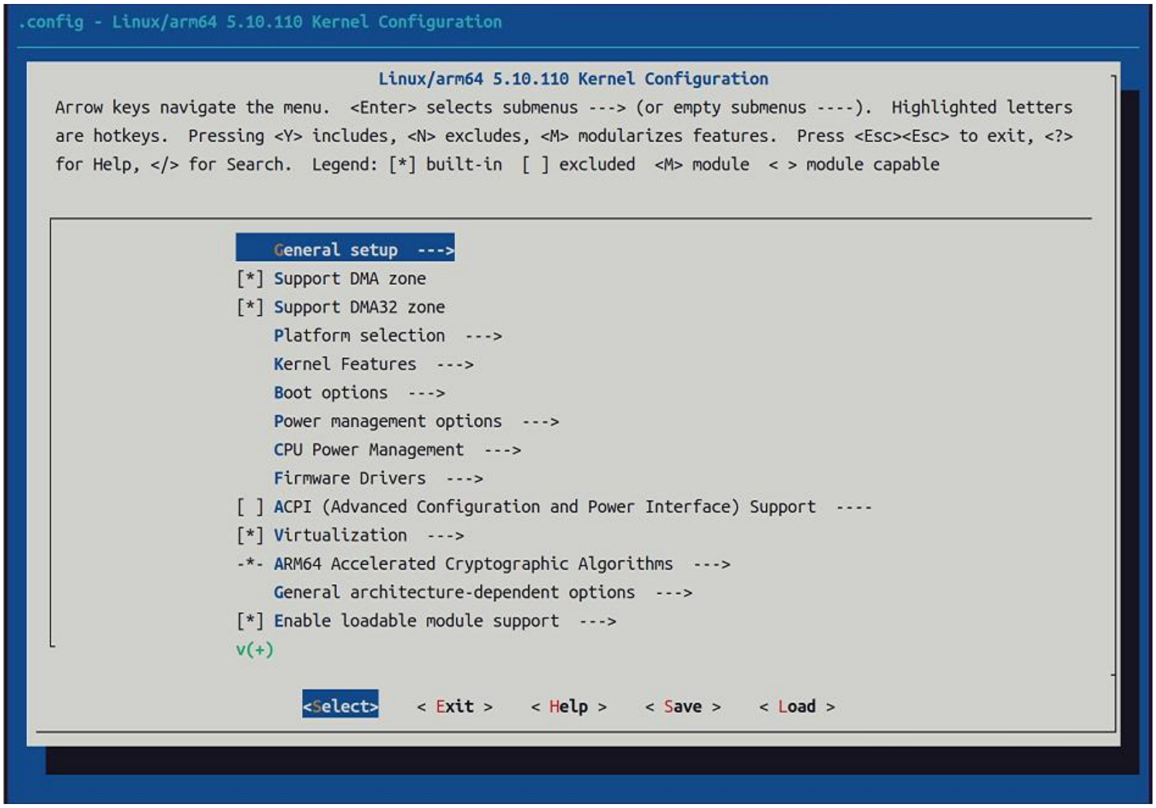



4) Then it will be prompted whether the kernel configuration interface needs to be displayed. If the kernel configuration is not required, select the first one. If you need to modify the kernel configuration, select the second one




5) If the step 4 is selected to display the kernel configuration menu (the second option), the interface of the kernel configuration opened through the **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration. However, after exiting,

the kernel source code will be compiled.




a. If you do not need to modify the configuration option of the kernel, when running the build.sh script,Passing the **KERNEL_CONFIGURE=no** to temporarily shield the configuration interface of the pop -up kernel.

```
test@test:~/orangepi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

b. You can also set **orangepi-build/userpatches/config-default.conf** configuration file. **KERNEL_CONFIGURE=no**, which can permanently disable this function.

c. When compiling the kernel, if the error is prompted, this is because the terminal interface of the Ubuntu PC is too small, causing the interface of the **make menuconfig** to be displayed.Please maximize the terminal of Ubuntu PC, and then re -run the build.sh script.

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

6) Part of the information prompted when compiling the kernel source code is as follows

a. Linux kernel source version

[ o.k. ] Compiling current kernel [ **5.10.110** ]

b. The version of the cross compilation tool chain used

[ o.k. ] Compiler version [ **aarch64-none-linux-gnu-gcc 9.2.1** ]

c. The configuration file used by the core and the path it stored

[ o.k. ] Using kernel config file [ **config/kernel/linux-rockchip-rk3588-legacy.config** ]

d. The path of the compiled kernel-related deb package

[ o.k. ] Target directory [ **orangepi-build/output/debs/** ]

e. Compiled and generated kernel images deb bag name

[ o.k. ] File name [ **linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb** ]

f. The time of compilation and use

[ o.k. ] Runtime [ **5 min** ]

g. Finally, the compilation command of the kernel selected repeatedly will be displayed. The following commands are not selected by the graphic interface, and the kernel source code can be directly compiled.

[ o.k. ] Repeat Build Options [ **sudo ./build.sh BOARD=orangepi5 BRANCH=legacy BUILD_OPT=kernel KERNEL_CONFIGURE=no** ]

7) View compiled core -related deb packages.

a. **linux-dtb-legacy-rockchip-rk3588_1.0.2_arm64.deb** containing kernels dtb files
b. **linux-headers-legacy-rockchip-rk3588_1.0.2_arm64.deb** Contains kernel head files
c. **linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-legacy-rockchip-rk3588_1.0.2_arm64.deb
output/debs/linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb
output/debs/linux-headers-legacy-rockchip-rk3588_1.0.2_arm64.deb
```

8) The file contained in the deb package generated by the generated linux-image is shown below

a. Use the following commands to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi_build/output/debs$ mkdir test
test@test:~/orangepi_build/output/debs$ cp \
linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb test/
test@test:~/orangepi_build/output/debs$ cd test
test@test:~/orangepi_build/output/debs/test$ dpkg -x \
linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb .
test@test:~/orangepi_build/output/debs/test$ ls
boot   etc   lib   linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb   usr
```

b. The decompressive file is shown below.

```
test@test:~/orangepi-build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.110-rockchip-rk3588
│   ├── System.map-5.10.110-rockchip-rk3588
│   └── vmlinuz-5.10.110-rockchip-rk3588
├── etc
│   └── kernel
├── lib
│   └── modules
├── linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb
└── usr
    ├── lib
    └── share
```

9) orangepi-bulid compile system compiles Linux kernel source code first will synchronize the Linux kernel source code with the Linux kernel source code of the

GitHub server. If you want to modify the source code of the Linux kernel, you need to close the update function of the source code (**need to compile it completely once. This function can be closed after the Linux kernel source code, otherwise it will be prompted that the source code of the Linux kernel cannot be found. If it is downloaded from the source code compressed from the Google Drive, there is no such problem, because the source code of Linux has been cached**), otherwise it will be done. The modification will be restored, the method is as follows:

Set the IGNORE_UPDATES variable in **userpatches/config-default.conf** to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If you modify the kernel, you can use the following method to update the kernel and kernel module of the development board Linux system

a. Upload the compiled deb package in Linux kernel to be uploaded to the Linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board, uninstall the installed Linux kernel deb bag

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk3588
```

c. Install the deb package in the new Linux kernel that just uploaded up

```
root@orangepi:~# dpkg -i linux-image-legacy-rockchip-rk3588_1.0.2_arm64.deb
```

d. Then restart the development board, and then check whether the kernel -related modification has taken effect

```
root@orangepi:~# reboot
```

10) Other useful information

a. The kernel configuration file storage position is shown below. Please do not go to the kernel source code to find the kernel configuration file used in the development board.

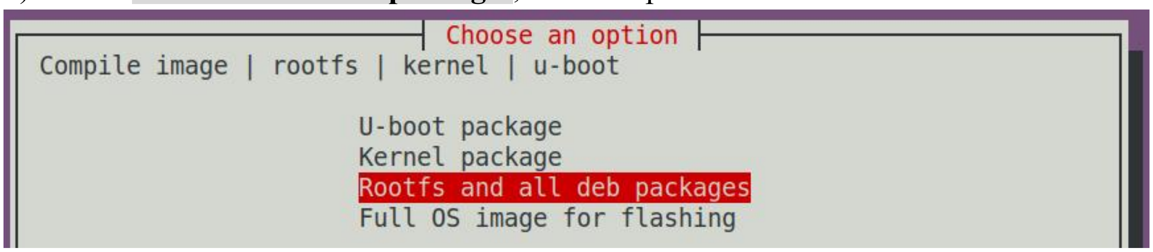
```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-legacy.config
The location of the dts file used in the development board is
orangepi-build/kernel/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/rk3588s
-orangepi-5.dts
```

## 4. 4. Compile rootfs

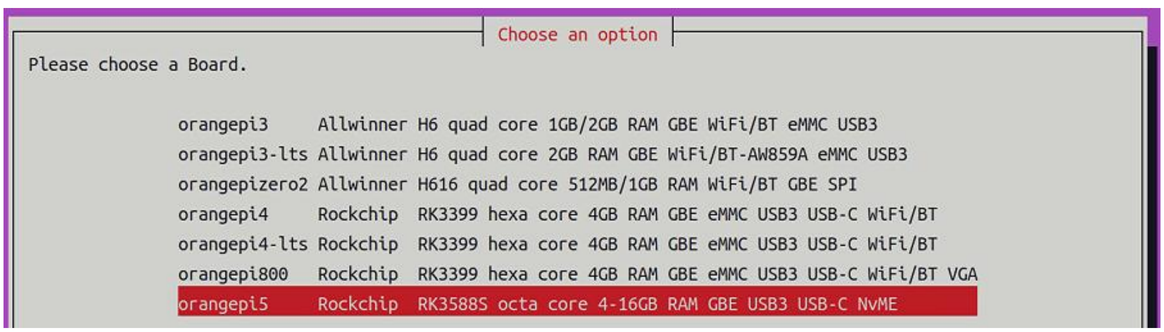
1) Run the build.sh script, remember to add Sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

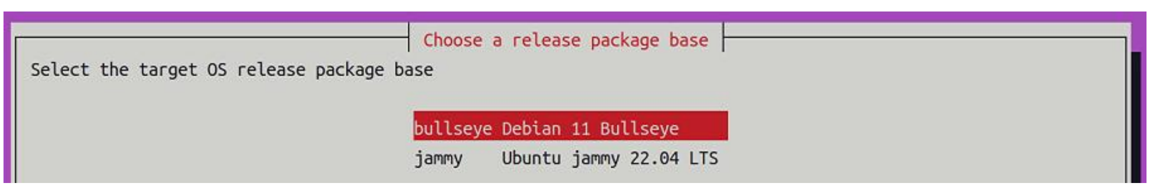
2) Select **Rootfs and all deb packages**, and then press Enter



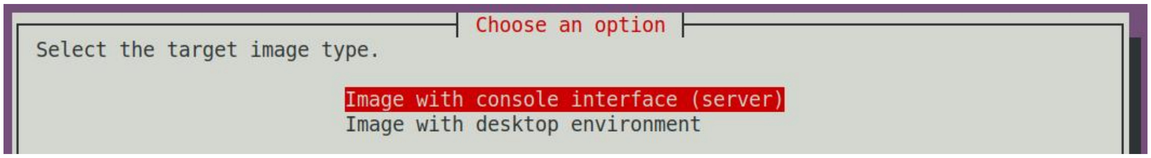

3) Then select the model of the development board



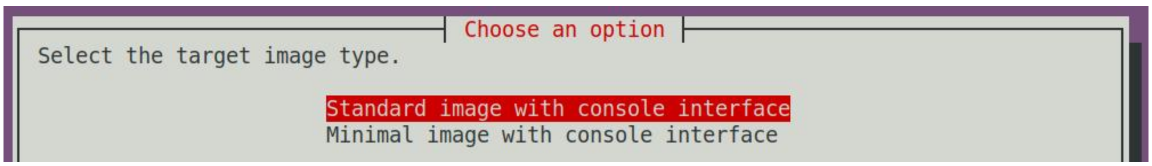

4) Then select the type of rootfs



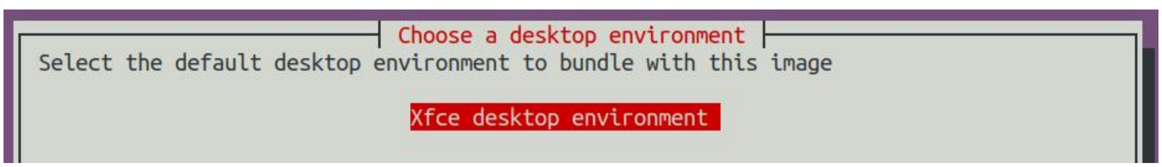

5) Then select the type of the image
   a. **Image with console interface (server)**Indicates the image of the server version, the volume is relatively small
   b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

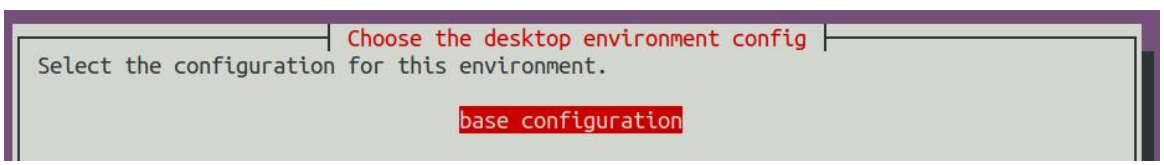



6) If it is a image that compiles the server version, you can also choose to compile the Standard version or the minimal version. The pre -installed software pre -installed software will be much less than the Standard version. (**No special needs, please do not choose the minimal version, because many things are not pre -installed by default, some functions may not be used**)



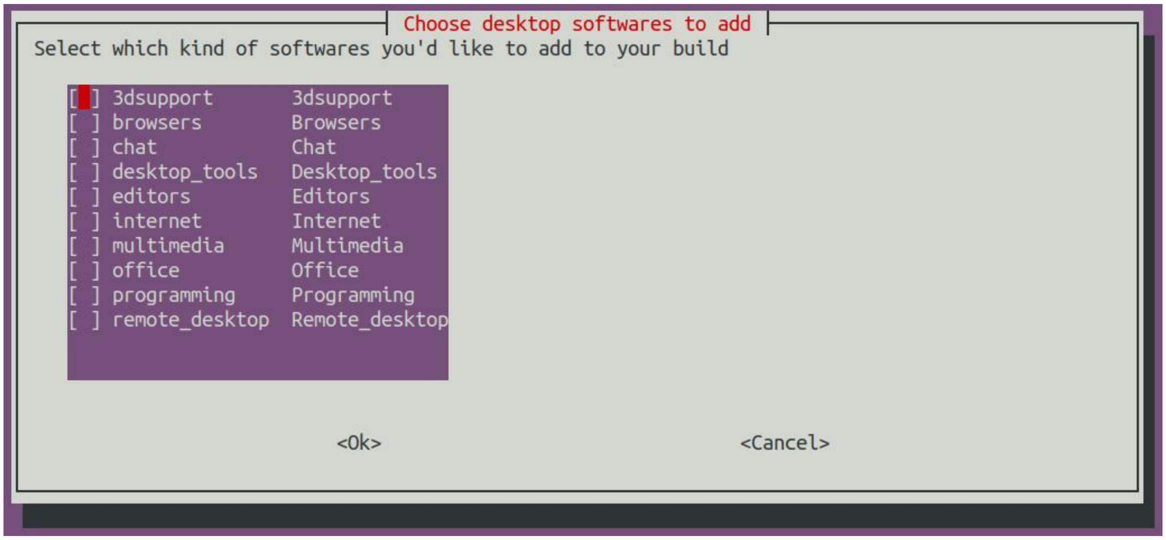

7) If the image of the desktop version also needs to choose the type of desktop environment, but currently only supports XFCE, so press the Enter key directly.




**Note that although Ubuntu Jammy has opened the Gnome's compilation option, it is not carefully tested. At present, please do not choose the Gnome desktop environment.**

Then you can choose an additional software package you need to install. Press the Enter key here directly.




8) Then you will start compiling rootfs, and part of the information prompted during compilation is shown below

a. rootfs type

[ 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 compiled rootfs compressed package

[ o.k. ] File name [ **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** ]

d. Time to compile

[ o.k. ] Runtime [ **13 min** ]

9) View compiled rootf compression package

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**It is a compressed package for rootfs. The meaning of each field of the name is

a) **Jammy** Represents the type of rootfs's Linux release version

b) **Xfce** Indicates that rootfs is the type of desktop version, if it is **cli**, it means the server version type

c) **arm64** Indicates the architecture of rootfs

d) **f930ff6ebbac1a72108a2e100762b18f** is a MD5 hash value generated by all the package names installed by rootfs. As long as the list of software packets installed by rootfs is not modified, this value will not change. The compilation script will be judged by this MD5 hash value to judge Do you need re -compilation of rootfs

b. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list** List the package name of all software packages installed by Rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.current
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list
```

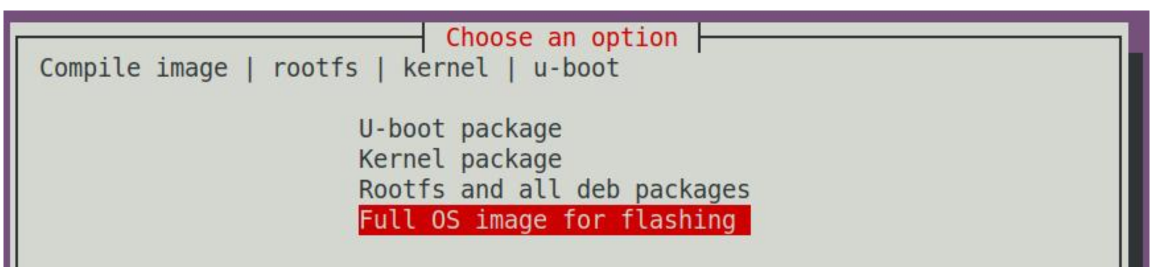
10) If the required rootfs already exists under **external/cache/rootfs**, then compile rootfs again will skip the compilation process directly, and will not start compiling again. When compiling the image, you will also go to the **external/cache/rootfs** to find out if there is a cache rootfs. If you have it, you can use it directly, so that you can save a lot of download compilation time.

## 4. 5. Compile linux images

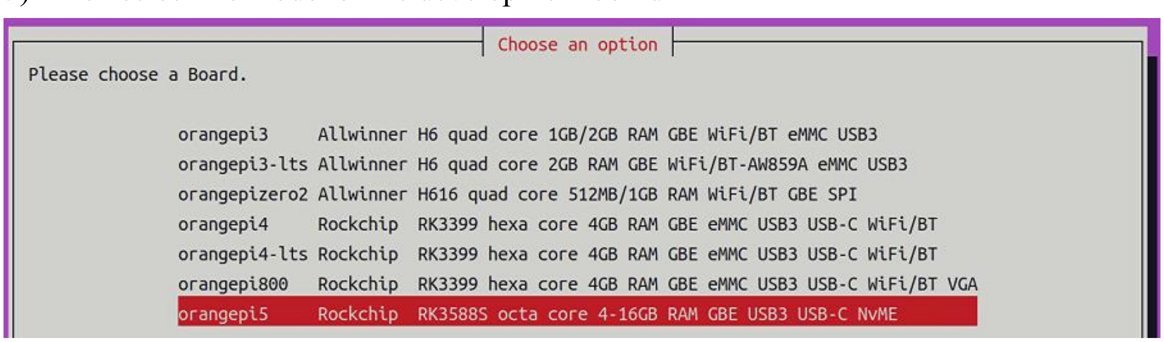
1) Run the build.sh script, remember to add Sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

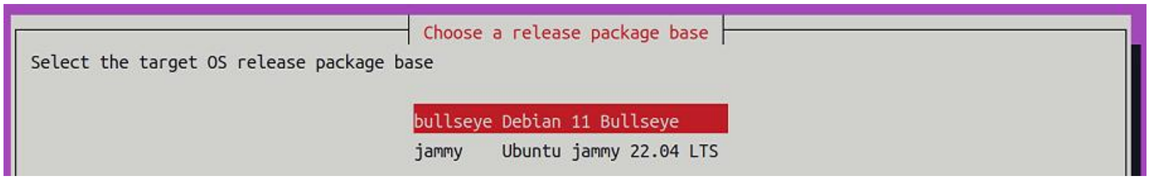
2) Select **Full OS image for flashing**, and then press Enter




3) Then select the model of the development board



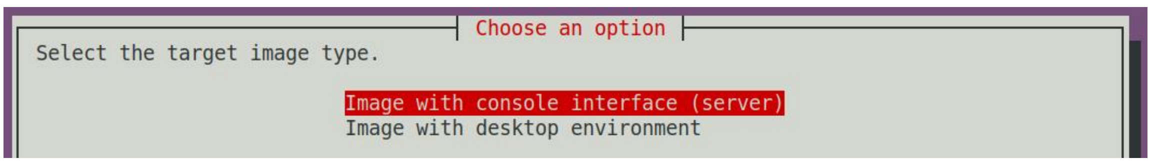
4) Then select the type of rootfs



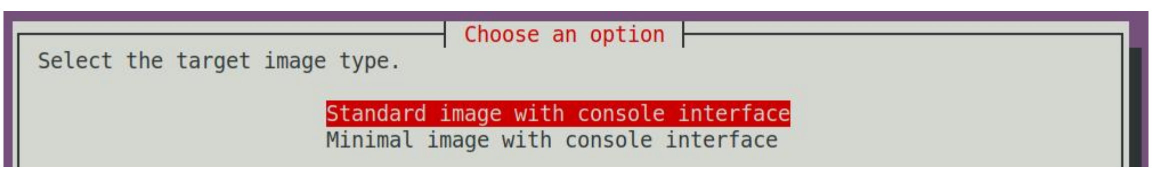

5) Then select the type of the image

a. **Image with console interface (server)** Indicates the image of the server version, the volume is relatively small

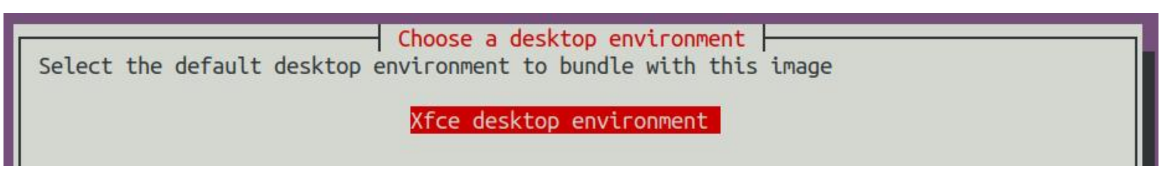
b. **Image with desktop environment** Indicates a image with a desktop, which is relatively large

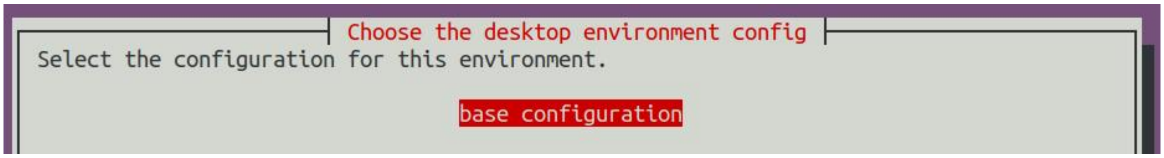



6) If it is a image that compiles the server version, you can also choose to compile the Standard version or the minimal version. The pre -installed software pre -installed software will be much less than the Standard version. (**No special needs, please do not choose the minimal version, because many things are not pre -installed by default, and some functions may not be used**).



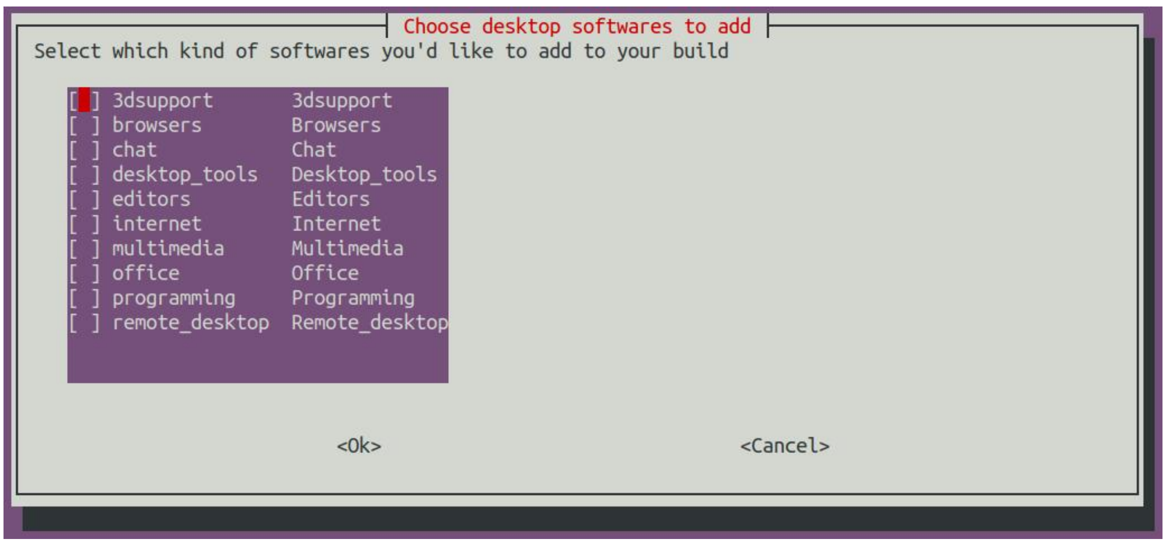

7) If the image of the desktop version also needs to choose the type of the desktop environment, but currently only supports XFCE, so you can go directly to the car






**Note that although Ubuntu Jammy has opened the Gnome's compilation option, it is not carefully tested. At present, please do not choose the Gnome desktop environment.**

Then you can choose an additional software package you need to install. Press the Enter key here directly.




8) Then you will start to compile Linux image, and the general process of compilation will be as follows

a. Initialize the compilation environment of Ubuntu PC, install the software package required for the compilation process.
b. Download the source code of the U-Boot and Linux kernel (if it has cached, only the code is updated)
c. Compile the U-Boot source code and generate the deb package of U-Boot
d. Compile Linux source code and generate Linux -related deb packages
e. Make Linux Firmware deb package
f. Make deb package of Orange-Config tools
g. Make a board -level Deb package.

h. If it is compiled the desktop version of the image, it will also make a desktop -related Deb package.
i. Check whether rootfs has cached. If there is no cache, re -make rootfs. If it has cached, it will be used directly.
j. Install the deb package to the rootfs that generated before.
k. Make some specific settings for different development boards and different types of images, such as pre -installed software packages, modifying system configurations, etc.
l. Then make a image file and format the partition, the default type is ext4
m. Copy the configured rootfs into the partition of the image
n. Then update initramfs
o. Finally, the u-boot bin file is written into the image through the dd command

9) The following information will be prompted after compiling the image

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/orangepi5_1.0.2_debian_bullseye_linux5.10.110_xfce_desktop/orangepi5_1.0.2_debian_bullseye_linux5.10.110_xfce_desktop.img ]
```

b. Time to compile.

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the command of the Compile image. Use the following command without the need to choose through the graphic interface. You can directly start the Compile image

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepi5 BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

# 5. Instructions for the use of theandroid 12 system

## 5. 1. How to use the use of wireless network card

1) At present, the USB wireless network card model that is adapted to the image is shown below:

| Chip model | Function | VID&PID | Adaptation |
|---|---|---|---|
| RTL8821CU | 2.4G +5G WIFI+BT 4.2 | 0bda:c820 | Only support wifi, Bluetooth needs to be adapted |
| RTL8723BU | 2.4G WIFI+BT4.0 | 0bda:b720 | Only support wifi, Bluetooth needs to be adapted |
| RTL8811CU | 2.4G +5G WIFI | 0bda:c811 | Only WIFI function, supported |

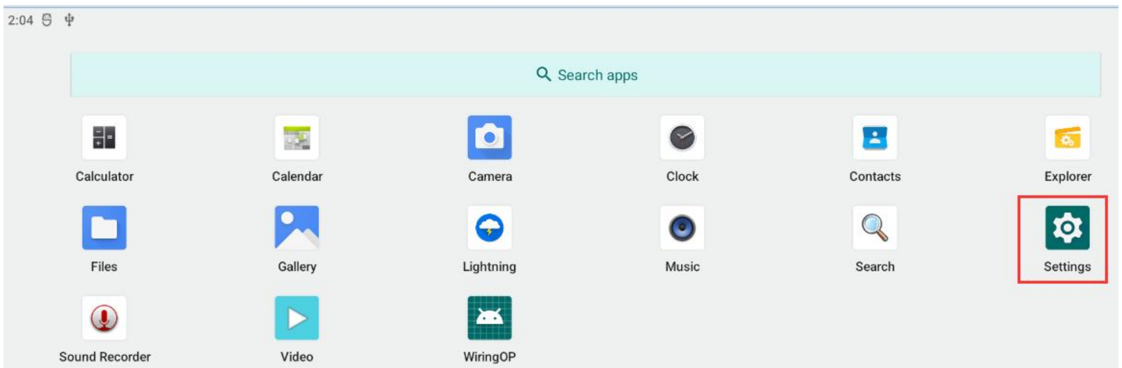
2) The picture of the above three USB wireless network cards is shown below:

a. The picture of the RTL8821CU USB wireless network card module is shown below:

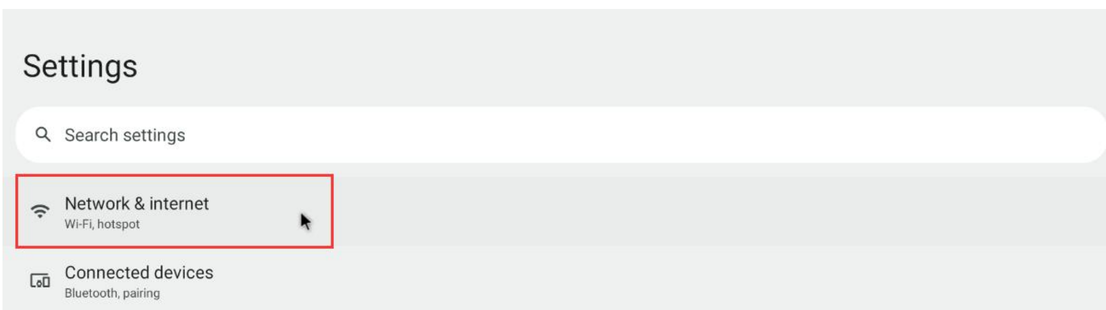

b. The pictures of the RTL8723BU USB wireless network card module are shown below:


c. The picture of the RTL8811CU USB wireless network card module is shown below:

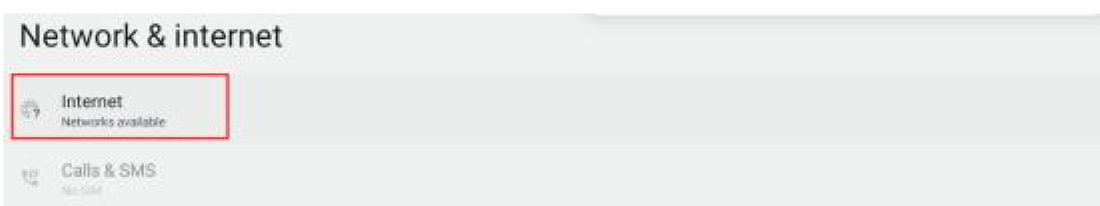

3) The test methods of the USB wireless network cards of the above 3 models are the same. First, insert the USB network card into the USB interface of the development board, and then enter **Setting**



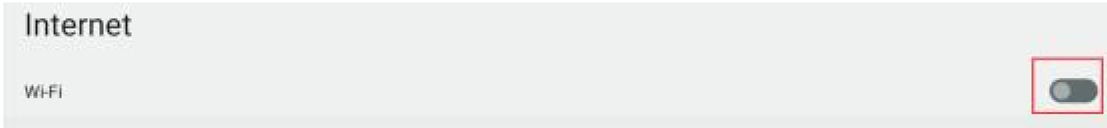

4) Then choose **Network & internet**



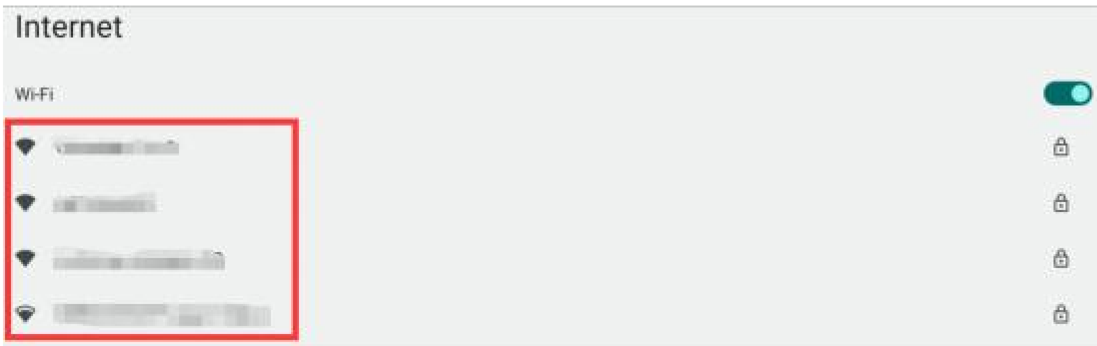

5) Then choose **Internet**



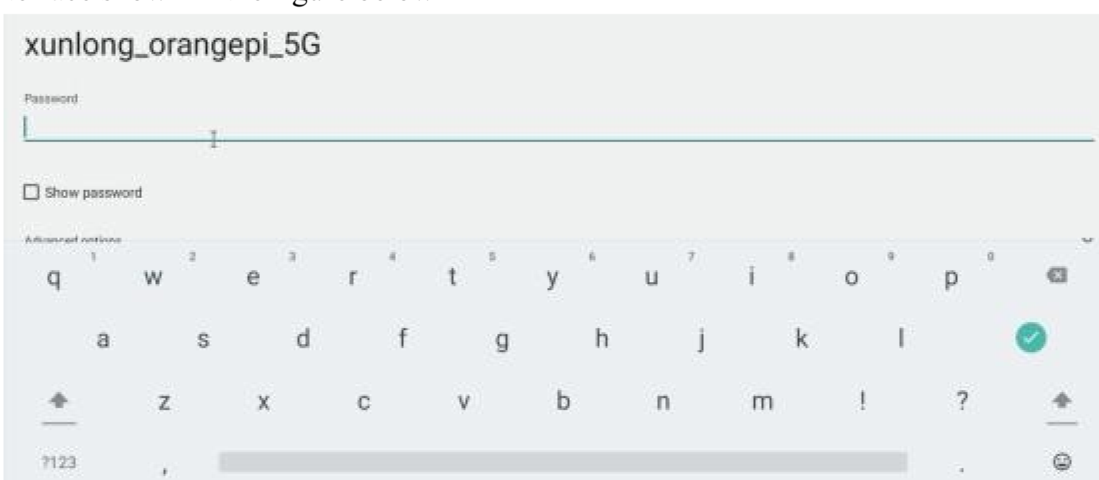

6) Then turn on the **Wi-Fi** switch



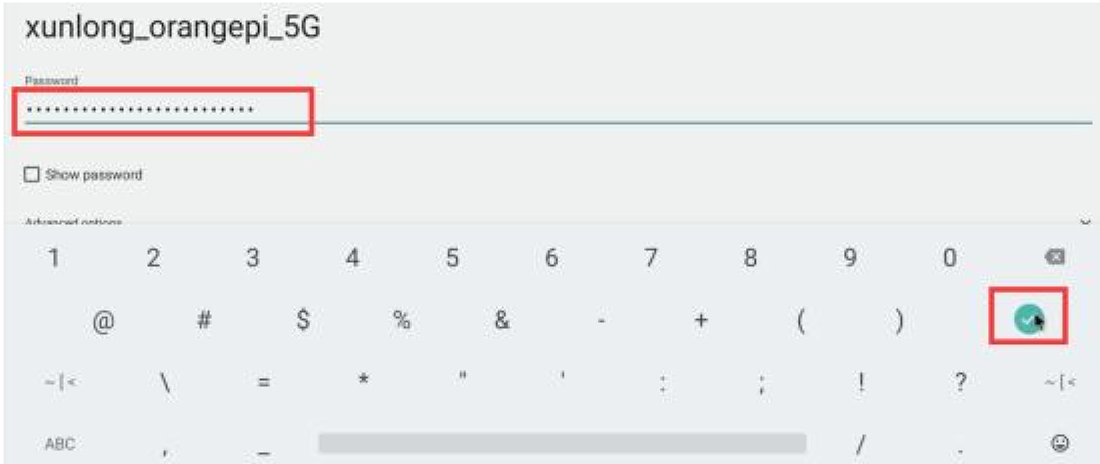

7) If everything is normal after opening the **Wi-Fi**, you can scan to the nearby Wi-Fi hotspot



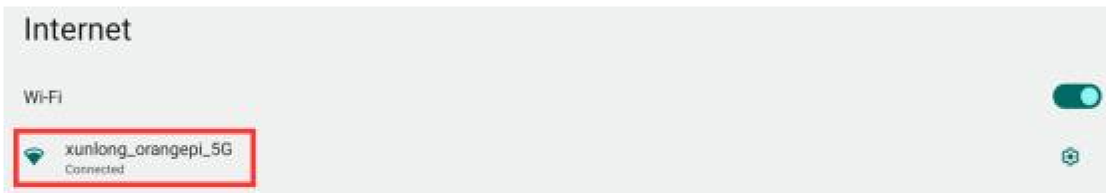

8) Then select the Wi-Fi you want to connect, and you will pop up the password input interface shown in the figure below




9) Then use the keyboard to enter the corresponding password of the wi-fi, and then use the mouse to click the Enter button in the virtual keyboard to start connecting Wi-Fi



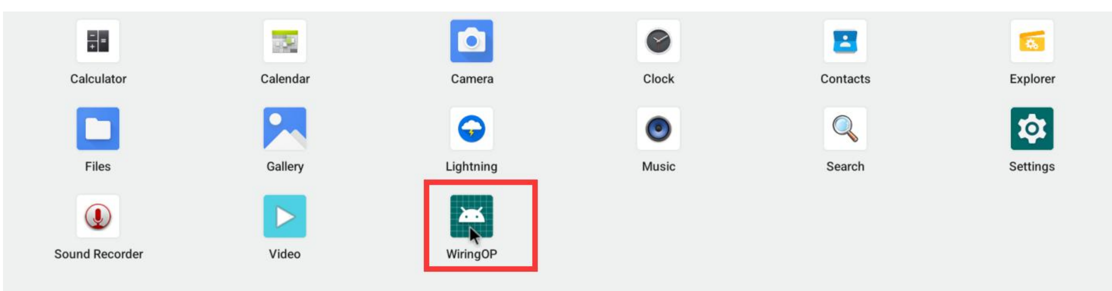

10) The display after the Wi-Fi connection is successful as shown in the figure below：



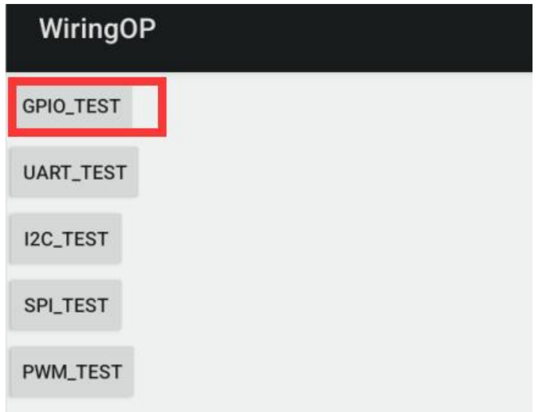

## 5. 2. 26Pin interface GPIO, UART, SPI and PWM test

### 5. 2. 1. 26pin GPIO port test

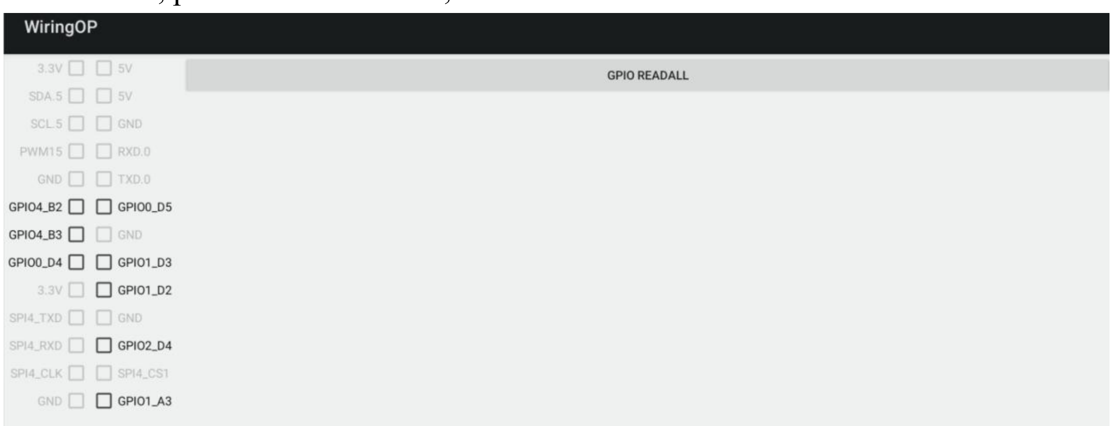
1) First click the Wiringop icon to open the Wiringop App



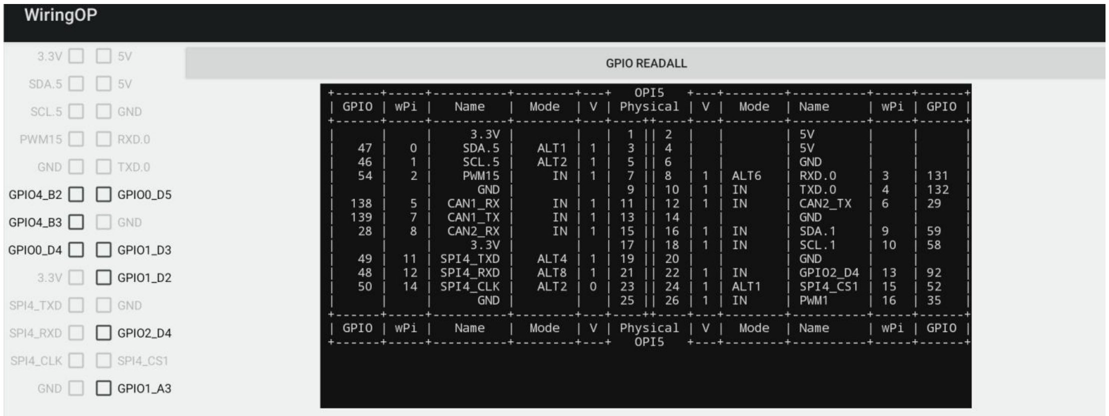

2) The main interface of the Wiringop app is displayed as shown in the figure below, and then click the **GPIO_TEST** button to open the GPIO test interface




3) The GPIO test interface is shown in the figure below. The two rows of the **CheckBox** button on the left and the 26PIN pin are one -to -one relationship. When checking the **CheckBox** button, the corresponding pin will be set to **OUT** mode, the pin level settings will be set. For high levels, when the check -up is canceled, the pin level is set to a low level; when clicking the **GPIO READALL** button on the right, you can get the WPI, GPIO mode, pin level information, etc.




4) Then click the **GPIO READALL** button, and the output information is shown in the figure below

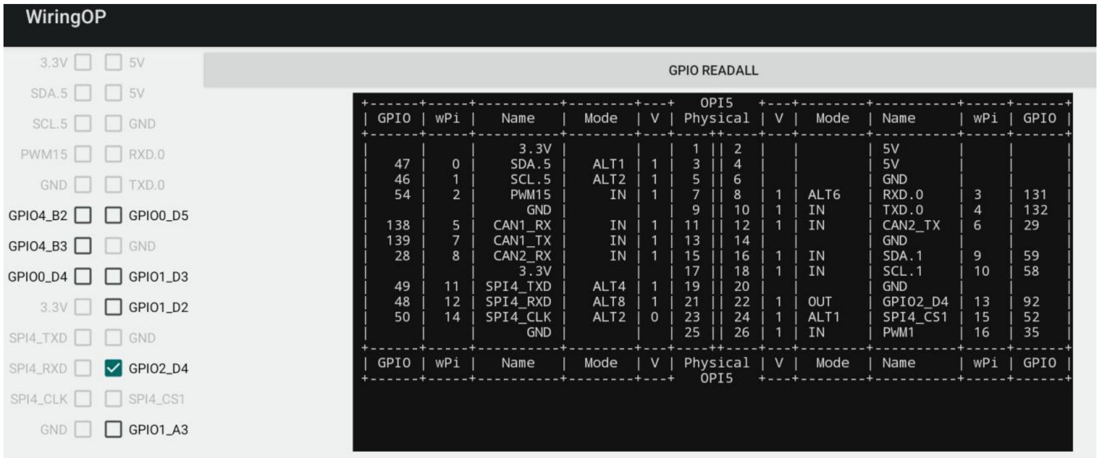



| GPIO | wPi | Name | Mode | V | Physical | | V | Mode | Name | wPi | GPIO |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3.3V | | | 1 | 2 | | | 5V | | |
| 47 | 0 | SDA.5 | ALT1 | 1 | 3 | 4 | | | 5V | | |
| 46 | 1 | SCL.5 | ALT2 | 1 | 5 | 6 | | | GND | | |
| 54 | 2 | PWM15 | IN | 1 | 7 | 8 | 1 | ALT6 | RXD.0 | 3 | 131 |
| | | GND | | | 9 | 10 | 1 | IN | TXD.0 | 4 | 132 |
| 138 | 5 | CAN1_RX | IN | 1 | 11 | 12 | 1 | IN | CAN2_TX | 6 | 29 |
| 139 | 7 | CAN1_TX | IN | 1 | 13 | 14 | | | GND | | |
| 28 | 8 | CAN2_RX | IN | 1 | 15 | 16 | 1 | IN | SDA.1 | 9 | 59 |
| | | 3.3V | | | 17 | 18 | 1 | IN | SCL.1 | 10 | 58 |
| 49 | 11 | SPI4_TXD | ALT4 | 1 | 19 | 20 | | | GND | | |
| 48 | 12 | SPI4_RXD | ALT8 | 1 | 21 | 22 | 1 | IN | GPIO2_D4 | 13 | 92 |
| 50 | 14 | SPI4_CLK | ALT2 | 0 | 23 | 24 | 1 | ALT1 | SPI4_CS1 | 15 | 52 |
| | | GND | | | 25 | 26 | 1 | IN | PWM1 | 16 | 35 |
| GPIO | wPi | Name | Mode | V | Physical | | V | Mode | Name | wPi | GPIO |

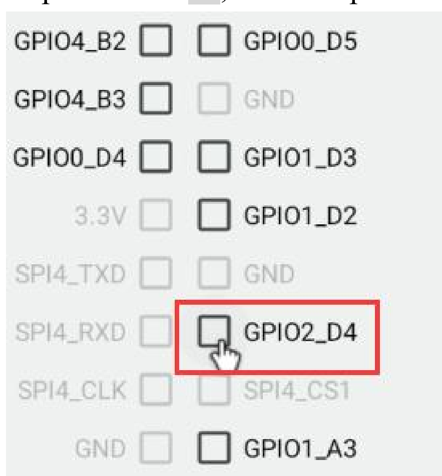
OPI5

5) Taking the high and low level of the **GPIO2_D4** as an example, click the **CheckBox** button in the figure below. When the button is selected, the **GPIO2_D4** is set to a high level. After setting, you can use the value of the voltage of the pins by the multimeter. If it is **3.3v**, Explain that setting high -electricity is successful

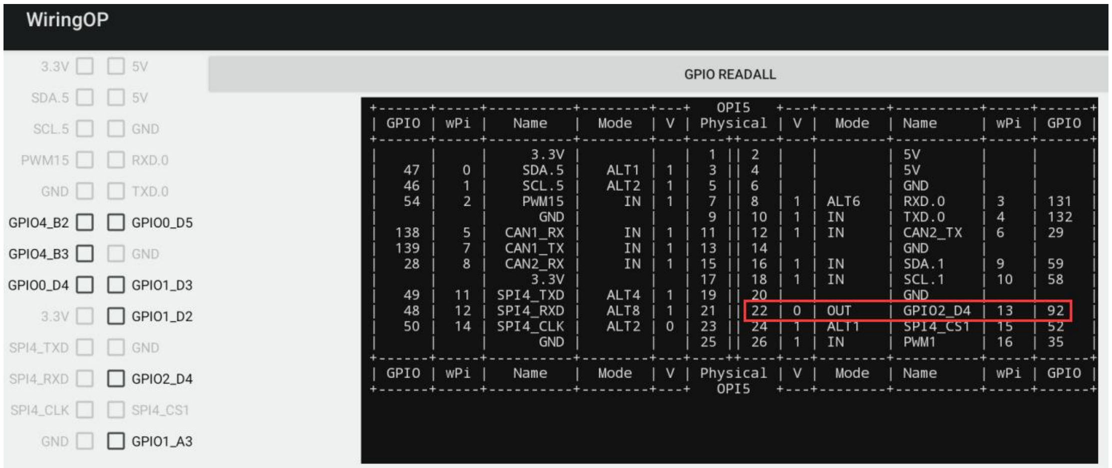

6) Then click the **GPIO READALL** button to see that the pins mode of the current **GPIO2_D4** is **OUT**, and the pin level is high level




7) Click the**CheckBox** button in the figure below to cancel the check status. The **GPIO2_D4** pin is set to a low level. After setting, you can use the value of the voltage of the multimeter to measure the pins.If it is **0v**, the low -power flat is set.



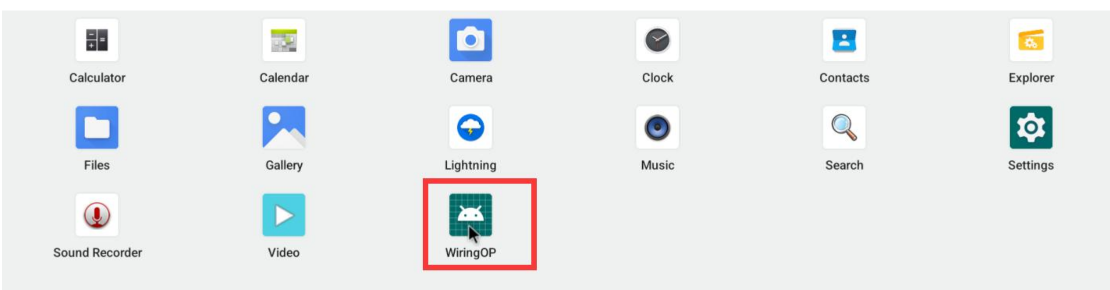

8) Then click the **GPIO READALL** button to see that the pins mode of the current GPIO2_D4 is OUT, and the pin level is low.




## 5. 2. 2. 26pin UART test

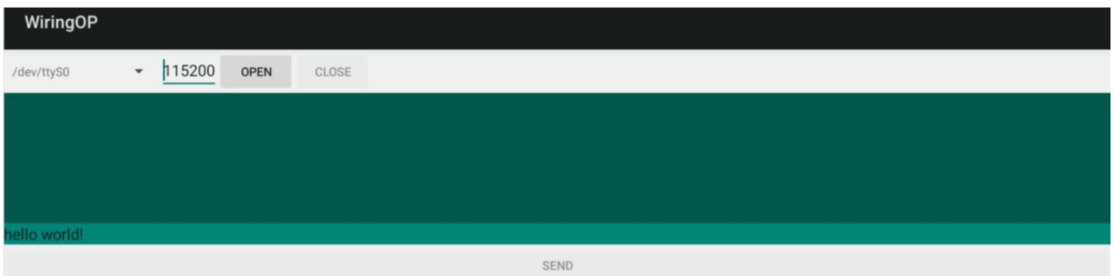
1) In Android default, the UART0 serial port is only opened. The position of UART0 at 26pin is shown in the figure below. The corresponding device node is/dev/ttys0


2) First click the WiringOP icon to open the Wiringop App

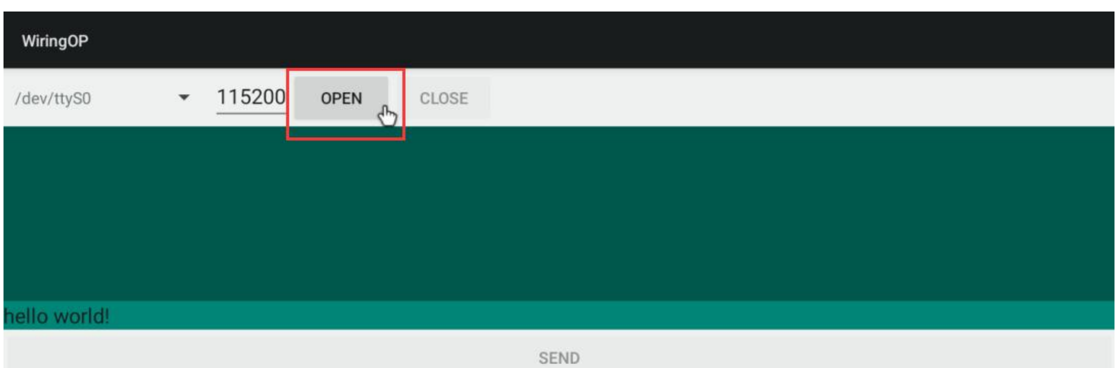



3) The main interface of the WiringOP APP is displayed as shown in the figure below, and then click the **UART_TEST** button to open the UART test interface


4) The serial test interface of the APP is shown in the figure below

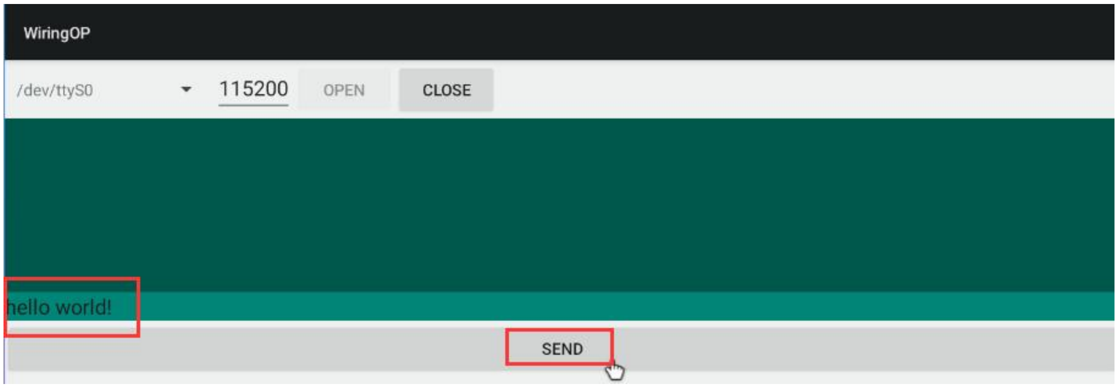



5) Then enter the baud rate you want to set in the editing box, and then click the **OPEN** button to open the **/dev/ttyS0** node. After successful, the **OPEN** button becomes an optional state.The **CLOSE** button and the **SEND** button become an optional state

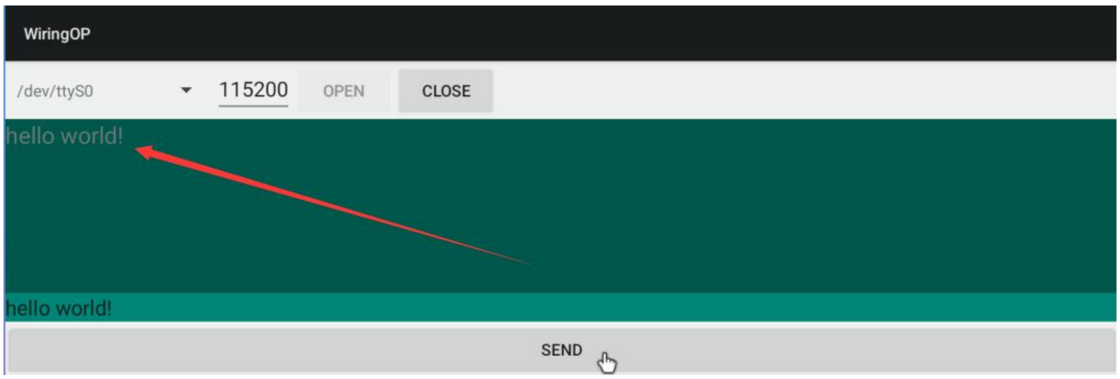



6) Then use the DuPont line to shorte the RXD and TXD pin of uart0

7) Then you can enter a section of characters in the editing box below, click the **SEND** button to start sending



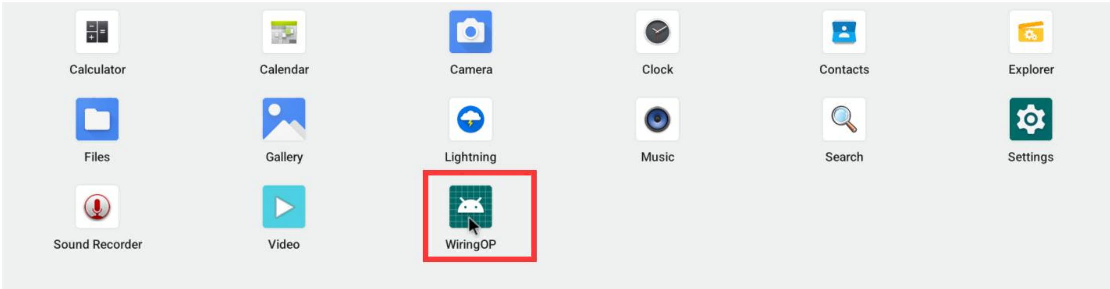

8) If everything is normal, the receiving string will be displayed in the receiving box




## 5. 2. 3. 26pin's PWM test

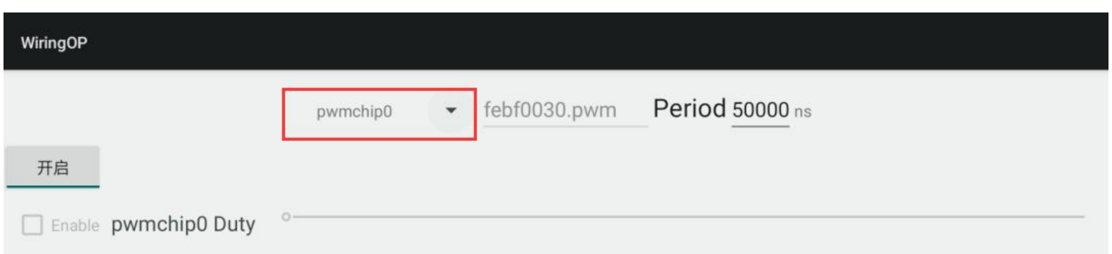
1) Android only opened **PWM15** by default. The corresponding pins are at the position of 26Pin.


2) First click theWiringOP icon to open the Wiringop App

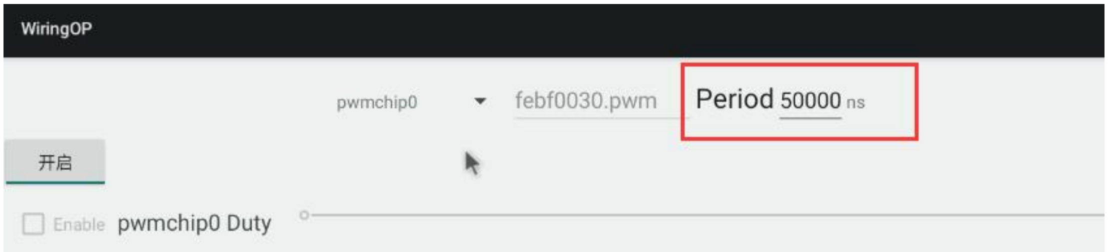



3) Then click the **PWM_TEST** button to enter the PWM test interface at the main interface of WiringOP

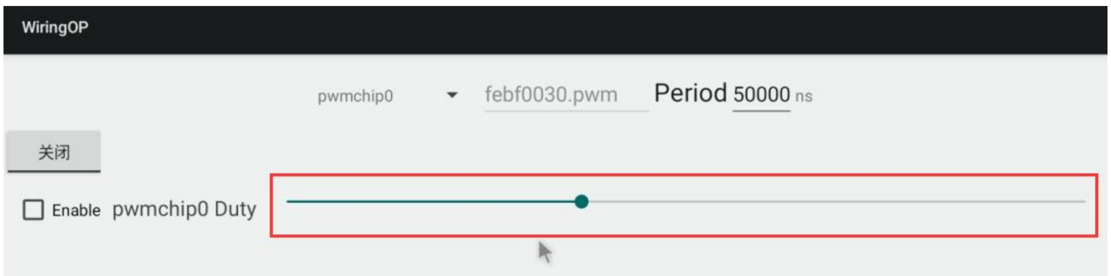

4) The corresponding address corresponding to the PWM15 is **febf0030**. The right side of PWMCHIP0 is exactly the **febf0030.pwm**. If the displayed base address is wrong, please click the drop -down option to select other PWMCHIP until the **febf0030** is displayed on the right.

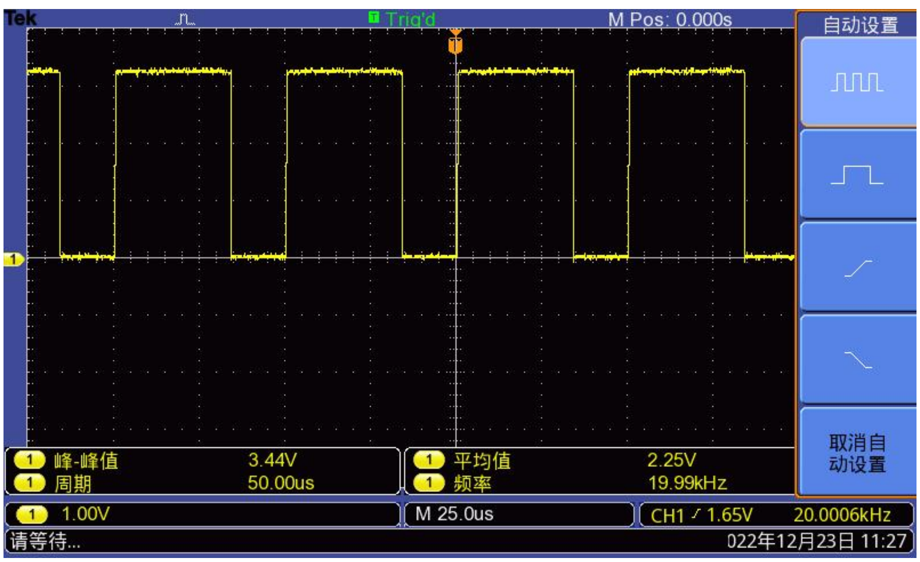



5) Then confirm the PWM cycle. The default configuration is **50000ns**, and the PWM frequency is **20KHz**. You can modify it by yourself.Click on the button to export **PWM15**



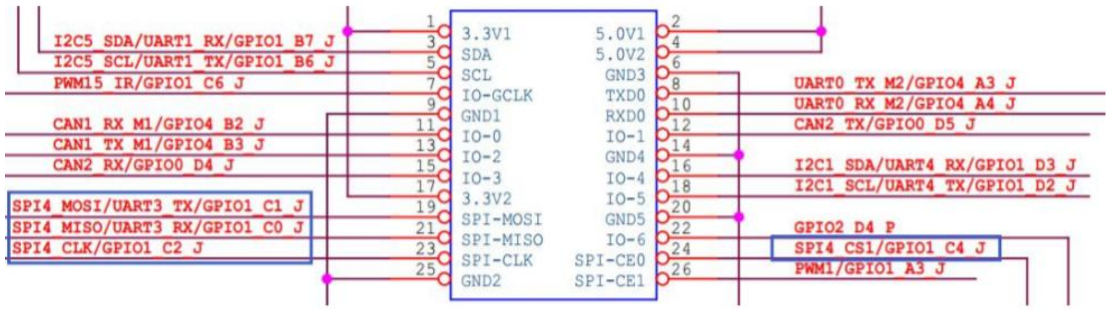

6) Then drag the drag below to change the PWM duty ratio, and then check the enable to output PWM



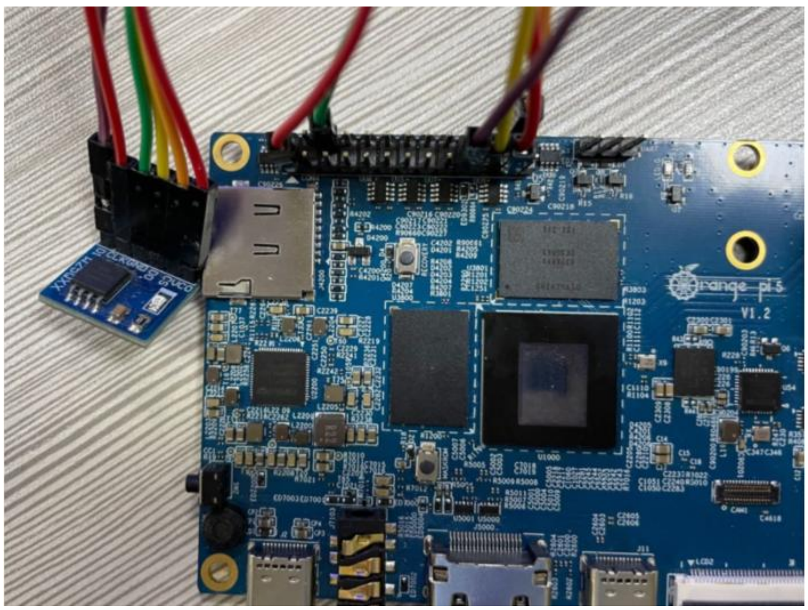




## 5. 2. 4. 26pin's SPI test

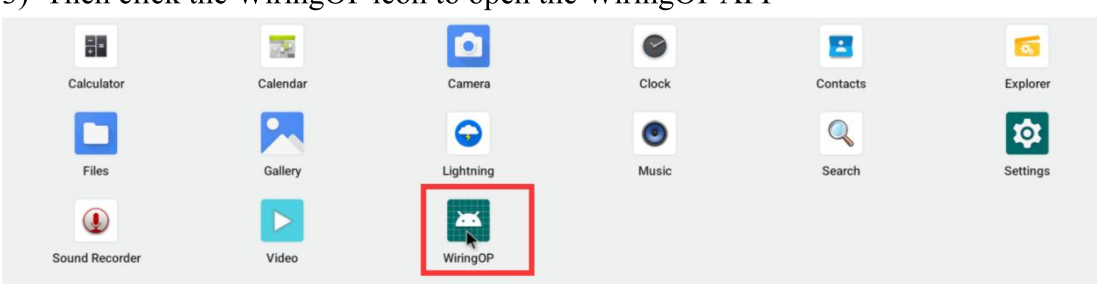
1) From the schematic diagram of the 26pin interface, the SPI available for Orange Pi 5 is spi4



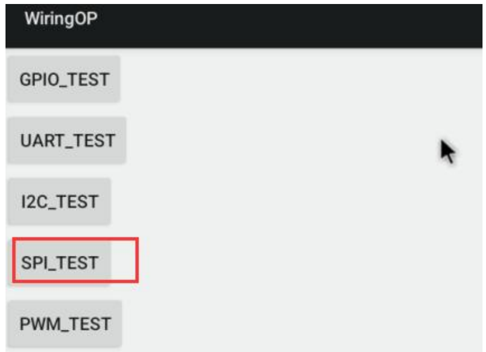

2) Here is the w25q64 module to test the SPI interface, and first access the w25q64 device at the SPI4 interface.



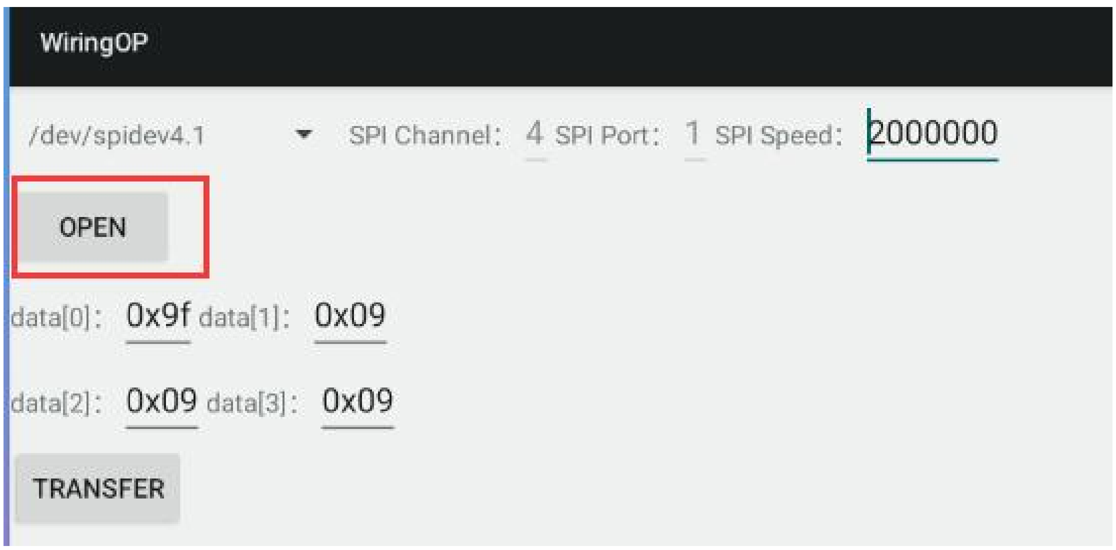
3) Then click the WiringOP icon to open the WiringOP APP



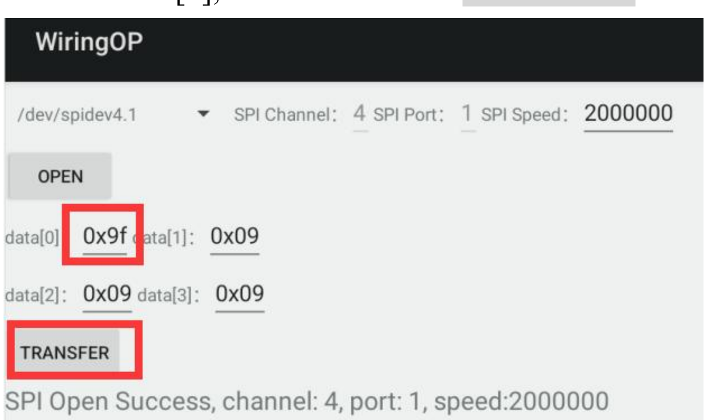

4) The main interface of the WiringOP APP shows as shown in the figure below, click the SPI_TEST button to open the SPI test interface




5) Then click the **OPEN** button to initialize SPI




6) Then fill in 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) The last app will display the ID information read


8) The MANUFACTURER ID of the w25q64 module is EFh, the Device ID is 4017h, and the value read above is corresponding (H represents hexadecimal)

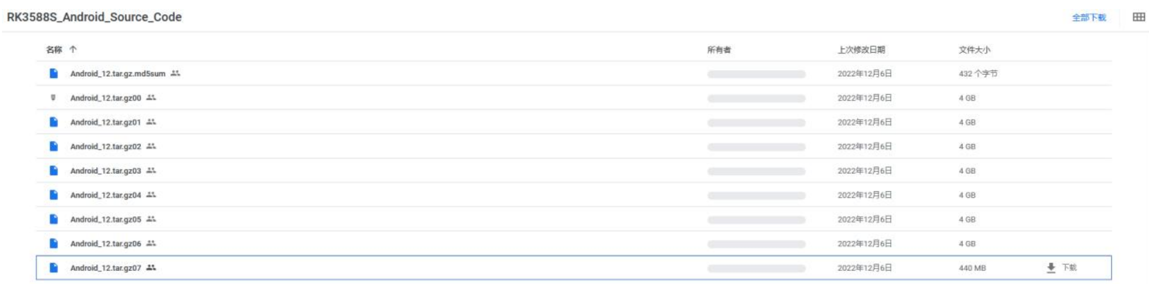
| **MANUFACTURER ID** | **(MF7 - MF0)** | |
|---|---|---|
| Winbond Serial Flash | EFh | |
| | | |
| **Device ID** | **(ID7 - ID0)** | **(ID15 - ID0)** |
| **Instruction** | **ABh, 90h, 92h, 94h** | **9Fh** |
| W25Q64FV (SPI) | 16h | 4017h |
| W25Q64FV (QPI) | 16h | 6017h |

# 6. The compilation method of the Android 12 source code

## 6. 1. Download the source code of Android 12

1) First download the sub -roll compression package of Android 12 source code from Google Drive or Google Web Disk

a. Google network disk




2) After downloading the sub -roll compression package of Android 12 source code, please check the MD5 check and whether it is correct. If it is not correct, please download the source code again.

```
test@test:~$ md5sum -c Android_12.tar.gz.md5sum
Android_12.tar.gz00: Sure
Android_12.tar.gz01: Sure
Android_12.tar.gz02: Sure
Android_12.tar.gz03: Sure
Android_12.tar.gz04: Sure
Android_12.tar.gz05: Sure
Android_12.tar.gz06: Sure
Android_12.tar.gz07: Sure
```

3) Then you need to merge multiple compression files into one, and then decompress it.

```
test@test:~$ cat Android_12.tar.gz0* > Android_12.tar.gz
test@test:~$ tar -xvf Android_12.tar.gz
```

## 6. 2. Compile the source code of Android 12

1) First install the software package required to compile the Android12 source code.

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y git gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip
test@test:~$ sudo apt-get install -y u-boot-tools
```

2) In the source code, there are build.sh compile scripts, and the compile parameters are as follows.

a. **-U**: Compile uboot
b. **-K**: Compile kernel
c. **-A**: Compile android
d. **-u**: Package generates update.img and update_spi_nvme.img
e. **-o**: Compile OTA bag
f. **-d**: Specify kernel dts

3) Compile Uboot, Kernel, Android and pack it into update.img

```
test@test:~$ cd Android_12
test@test:~/ Android_12$ source build/envsetup.sh
test@test:~/ Android_12$ lunch rk3588s_s-userdebug
test@test:~/ Android_12$ ./build.sh -AUKu
```

4) After the compilation is completed, the following information will be printed

```
********rkImageMaker ver 2.1********
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
Making update.img OK.
```

```
Make update image ok!
```

5) The final image file will be placed in the **rockdev/Image-rk3588s_s** directory. Among them, update.img is a TF card startup image, and update_spi_nvme.img is NVMe SSD boot image.

```
test@test:~/Android_12$ cd rockdev/Image-rk3588s_s
test@test:~/Android_12/rockdev/Image-rk3588s_s $ ls update*
update.img update_spi_nvme.img
```