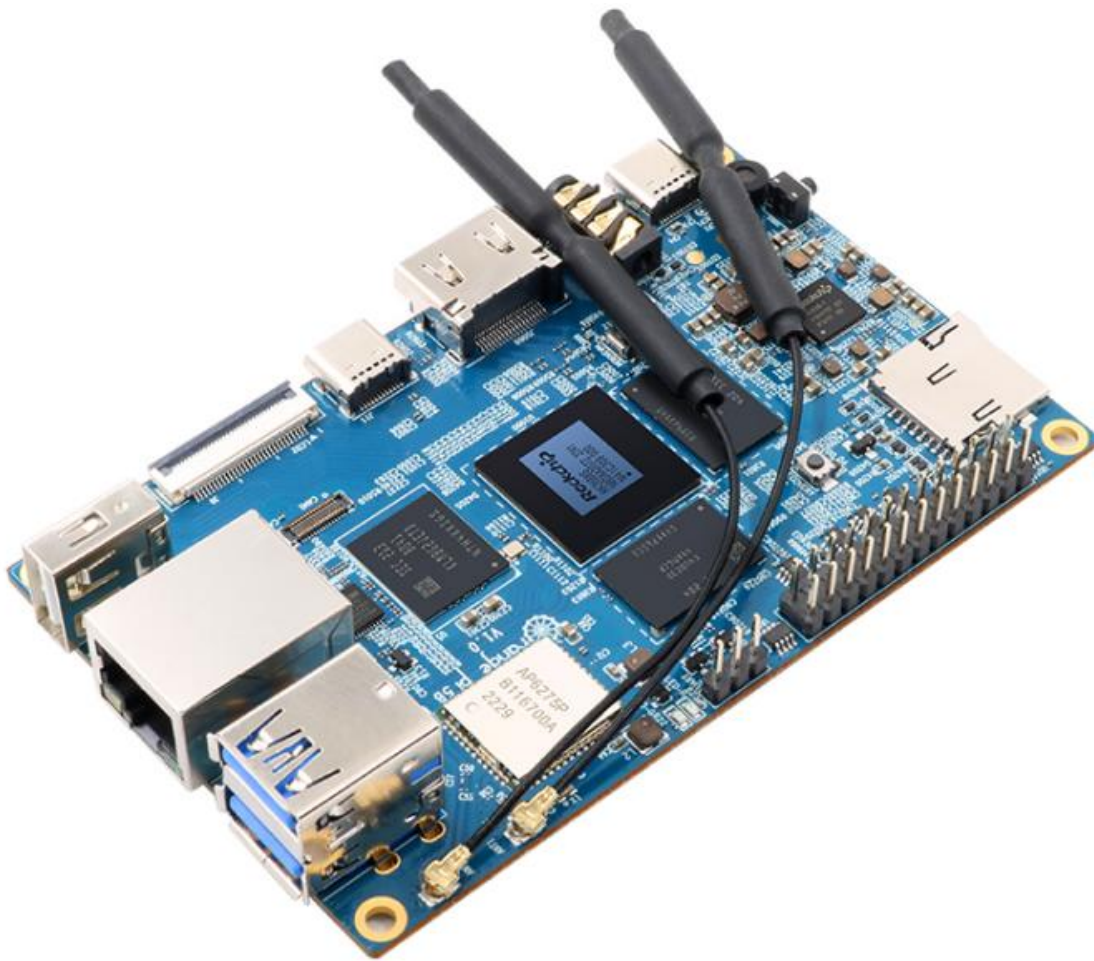


# Orange Pi 5B

## User Manual



# Contents

1. Basic features of Orange Pi 5B .....	1
1. 1. What is Orange Pi 5B .....	1
1. 2. Purpose of Orange Pi 5B .....	1
1. 3. Hardware features of Orange Pi 5B .....	2
1. 4. Top view and bottom view of Orange Pi 5B .....	4
1. 5. Interface details of Orange Pi 5B .....	6
2. Introduction to the use of the development board .....	8
2. 1. Prepare the required accessories .....	8
2. 2. Download the image of the development board and related materials .....	13
2. 3. How to burn Linux image to TF card .....	15
2. 3. 1. Method of burning Linux image to TF card based on Windows PC .....	15
2. 3. 2. How to burn Linux image using balenaEtcher .....	15
2. 3. 3. Method of burning Linux image to TF card based on Ubuntu PC .....	31
2. 4. How to burn Android image to TF card .....	35
2. 5. How to burn Orange Pi OS (Droid) image to TF card .....	38
2. 6. How to burn Linux image into eMMC .....	41
2. 6. 1. How to use RKDevTool to burn Linux image to eMMC .....	41
2. 6. 2. After burning the linux image into the eMMC, the linux system will start automatically. ....	50
2. 7. How to burn Android image into eMMC .....	52
2. 7. 1. How to burn Android image into eMMC via Type-C cable .....	52
2. 7. 2. How to burn Android 12 image into eMMC via TF card .....	58
2. 8. How to burn Orange Pi OS (Droid) image into eMMC .....	62
2. 8. 1. Burn Orange Pi OS (Droid) image to eMMC via Type-C cable .....	62
2. 8. 2. Burn Orange Pi OS (Droid) image to eMMC via TF card .....	67
2. 9. How to burn Orange Pi OS (OH) image to TF card .....	71



2. 10.	How to burn Orange Pi OS (OH) image into eMMC.....	74
2. 10. 1.	Burn Orange Pi OS (OH) image to eMMC via Type-C cable .....	74
2. 11.	Start the Orange Pi development board.....	79
2. 12.	How to use the debugging serial port.....	80
2. 12. 1.	Connection instruction of debugging serial port.....	80
2. 12. 2.	How to use the debugging serial port on the Ubuntu platform .....	82
2. 12. 3.	How to use the debugging serial port on Windows platform .....	85
2. 13.	Instructions for using the 5v pin in the 26pin interface of the development board to supply power .....	89
3.	Ubuntu/Debian Server and Xfce desktop system usage instructions .....	90
3. 1.	Supported Linux image types and kernel versions .....	91
3. 2.	Linux system adaptation .....	91
3. 2. 1.	Linux5.10 system adaptation situation .....	91
3. 2. 2.	Linux6.1 system adaptation situation .....	92
3. 3.	The format of linux commands in this manual .....	93
3. 4.	Linux system login instructions .....	95
3. 4. 1.	Linux system default login account and password .....	95
3. 4. 2.	How to set automatic terminal login in linux system .....	95
3. 4. 3.	Instructions for automatic login of Linux desktop version system .....	96
3. 4. 4.	The setting method of root user automatic login in Linux desktop version system .....	98
3. 4. 5.	The method of disabling the desktop in the Linux desktop version system .....	98
3. 5.	Onboard LED Light Test Instructions .....	101
3. 6.	Network connection test .....	102
3. 6. 1.	Ethernet port test .....	102
3. 6. 2.	WIFI connection test .....	104
3. 6. 3.	How to set a static IP address .....	111
3. 6. 4.	AP6275P PCIe NIC creates WIFI hotspot via create_ap .....	120
3. 7.	SSH remote login development board .....	129
3. 7. 1.	SSH remote login development board under Ubuntu .....	129



3. 7. 2. SSH remote login development board under Windows .....	130
3. 8. How to use ADB .....	132
3. 8. 1. How to use network adb .....	132
3. 8. 2. Use type-c data cable to connect to adb .....	134
3. 9. The method of uploading files to the Linux system of the development board .....	137
3. 9. 1. The method of uploading files to the development board Linux system in Ubuntu PC .....	137
3. 9. 2. The method of uploading files to the Linux system of the development board in Windows PC .....	140
3. 10. HDMI test .....	145
3. 10. 1. HDMI display test .....	145
3. 10. 2. HDMI to VGA display test .....	146
3. 10. 3. HDMI resolution setting method .....	147
3. 11. How to use Bluetooth .....	150
3. 11. 1. Test method of desktop image .....	150
3. 12. USB interface test .....	154
3. 12. 1. Connect USB mouse or keyboard to test .....	154
3. 12. 2. Connect USB storage device test .....	154
3. 12. 3. USB camera test .....	155
3. 13. Audio Test .....	157
3. 13. 1. Testing audio methods on desktop systems .....	157
3. 13. 2. The method of using commands to play audio .....	159
3. 13. 3. Method of using commands to test recording .....	161
3. 14. Temperature sensor .....	161
3. 15. 26 Pin Interface Pin Description .....	162
3. 16. How to install wiringOP .....	164
3. 17. 26pin interface GPIO, I2C, UART, SPI and PWM test .....	166
3. 17. 1. 26pin GPIO port test .....	166
3. 17. 2. 26pin SPI test .....	168
3. 17. 3. 26pin I2C test .....	169
3. 17. 4. 26pin UART test .....	171





3. 17. 5.	How to test PWM using /sys/class/pwm .....	173
3. 17. 6.	CAN test method .....	176
3. 18.	How to use wiringOP hardware PWM .....	183
3. 18. 1.	How to set PWM using wiringOP' s gpio command .....	184
3. 18. 2.	How to use the PWM test program .....	189
3. 19.	How to install and use wiringOP-Python .....	190
3. 19. 1.	How to install wiringOP-Python .....	191
3. 19. 2.	26pin GPIO port test .....	193
3. 19. 3.	26pin SPI test .....	195
3. 19. 4.	26pin I2C test .....	197
3. 19. 5.	26pin UART test .....	199
3. 20.	Hardware watch the door dog test .....	201
3. 21.	View the serial number of the RK3588S chip .....	202
3. 22.	Method of installing docker .....	202
3. 23.	Download the method of downloading and installing ARM64 version of Balenaetcher203	
3. 24.	The installation method of the BT linux panel .....	205
3. 25.	Set the Chinese environment and install Chinese input method .....	211
3. 25. 1.	Debian System installation method .....	211
3. 25. 2.	UBuntu 20.04 system installation method .....	217
3. 25. 3.	ubuntu 22.04 installation method .....	222
3. 26.	Remote login Linux system desktop method .....	228
3. 26. 1.	Use nomachine remote login .....	228
3. 26. 2.	Use VNC remote login .....	233
3. 27.	Some programming language test supported by the linux system .....	236
3. 27. 1.	Debian Bullseye System .....	236
3. 27. 2.	Debian Bookworm system .....	238
3. 27. 3.	Ubuntu Focal system .....	239
3. 27. 4.	Ubuntu Jammy system .....	241
3. 28.	QT installation method .....	243
3. 29.	ROS installation method .....	251



3. 29. 1. Ubuntu20.04 to install ROS 1 Noetic .....	251
3. 29. 2. Ubuntu20.04 to install ROS 2 Galactic .....	255
3. 29. 3. Ubuntu22.04 The method of installing ROS 2 Humble .....	258
3. 30. How to install kernel header files .....	260
3. 31. How to use 10.1 inch MIPI LCD screen .....	263
3. 31. 1. 10.1 -inch MIPI screen assembly method .....	263
3. 31. 2. Open 10.1 -inch MIPI LCD screen configuration method .....	266
3. 31. 3. The server version of the image rotation display direction method ..	270
3. 31. 4. The desktop image rotation display and touch direction method .....	271
3. 32. Instructions for opening the logo use .....	273
3. 33. OV13850 and OV13855 MIPI test methods for testing .....	274
3. 34. How to use the ZFS file system .....	280
3. 34. 1. How to install ZFS .....	281
3. 34. 2. How to create a ZFS pool .....	282
3. 34. 3. Test the data deduplication function of ZFS .....	284
3. 34. 4. Test the data compression function of ZFS .....	284
3. 35. How to install and use CasaOS .....	285
3. 35. 1. CasaOS installation method .....	285
3. 35. 2. How to use CasaOS .....	286
3. 36. Methods of using NPU .....	295
3. 36. 1. Preparation tools .....	295
3. 36. 2. Install RKNN-Toolkit2 on Ubuntu PC .....	295
3. 36. 3. Use RKNN-Toolkit2 for model conversion and model inference .....	296
3. 36. 4. Call the C interface to deploy the RKNN model and run it on the development board .....	302
3. 37. RK3588 How to use Baidu Feipian .....	305
3. 37. 1. Ubuntu PC environment construction .....	305
3. 37. 2. Board environment construction .....	309
3. 37. 3. Example of deploying model using FastDeploy .....	313
3. 38. The method of shutting down and restarting the development board .....	317
4. Ubuntu22.04 Gnome Wayland Desktop System Instructions .....	319
4. 1. Ubuntu22.04 Gnome Desktop System Adaptation .....	319



4. 2.	How to confirm that the window system currently is wayland.....	320
4. 3.	How to switch the default audio device .....	322
4. 4.	How to test GPU .....	323
4. 5.	How to play Chromium browser hardware solution video .....	324
4. 6.	How to play Kodi hardware solution video .....	326
4. 7.	How to install ROS2 Humble on Ubuntu22.04 Gnome .....	337
4. 8.	How to set Chinese environment and install Chinese input method .....	339
5.	Orange Pi OS Arch System Instructions .....	345
5. 1.	Orange Pi OS Arch system adaptation status .....	345
5. 2.	How to use AP6275P PCIe WIFI6+ Bluetooth module .....	346
5. 3.	How to use the 10.1-inch MIPI LCD screen .....	353
5. 3. 1.	10.1-inch MIPI screen assembly method .....	353
5. 3. 2.	How to open the 10.1-inch MIPI LCD screen configuration .....	356
5. 3. 3.	Methods of rotating display and touch direction .....	358
5. 4.	Testing methods for OV13850 and OV13855 MIPI cameras .....	361
5. 5.	How to install wiringOP .....	366
5. 6.	26pin interface GPIO, I2C, UART, SPI, CAN and PWM testing .....	367
5. 6. 1.	26pin GPIO port test .....	368
5. 6. 2.	26pin GPIO port pull-down resistor setting method .....	369
5. 6. 3.	26pin SPI test .....	370
5. 6. 4.	26pin I2C test .....	372
5. 6. 5.	26pin UART test .....	374
5. 6. 6.	PWM test method .....	376
5. 6. 7.	CAN test method .....	379
6.	Linux SDK——Orange Pi Build Instruction .....	382
6. 1.	Compile system requirements .....	382
6. 1. 1.	Use the development board Ubuntu 22.04 system to compile .....	382
6. 1. 2.	Use X64's Ubuntu 22.04 computer to compile .....	382
6. 2.	Get the source code of linux sdk .....	385



6. 2. 1. Download orangepi-build from github .....	385
6. 2. 2. Download the cross compilation tool chain .....	386
6. 2. 3. orangepi-build complete directory structure description .....	388
6. 3. Compile u-boot .....	389
6. 4. Compile the linux kernel .....	393
6. 5. Compile rootfs .....	399
6. 6. Compile linux images .....	402
7. Linux Development Manual .....	406
7. 1. How to compile the kernel source code separately in the linux system of the development board .....	406
8. How to use the Android 12 system. ....	409
8. 1. The Android versions supported .....	409
8. 2. Adaptation of Android functions .....	409
8. 3. WIFI connection test method .....	410
8. 4. How to use Wi-Fi hotspot .....	412
8. 5. How to test Bluetooth .....	415
8. 6. How to use 10.1 inch MIPI screen .....	418
8. 7. How to test OV13850 and OV13855 MIPI camera .....	420
8. 8. 26 Pin interface GPIO, UART, SPI and PWM test .....	426
8. 8. 1. 26pin GPIO port .....	426
8. 8. 2. 26pin UART test .....	431
8. 8. 3. 26pin SPI test .....	434
8. 8. 4. 26 pin PWM test .....	437
8. 9. How to use ADB .....	439
8. 9. 1. Use a data cable to connect to adb debugging .....	439
8. 9. 2. adb debug using network connection .....	440
8. 10. 2.4G USB remote control tested by Android Box .....	441
8. 11. How to use HDMI CEC function in Android Box system .....	442
9. The compilation method of the Android 12 source code .....	444



9. 1. Download the source code of Android 12 .....	444
9. 2. Compile the source code of Android 12 .....	445
10. Appendix .....	446
10. 1. User Manual Update History .....	446
10. 2. Image Update History .....	447



# 1. Basic features of Orange Pi 5B

## 1.1. What is Orange Pi 5B

Orange Pi 5B 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, with 4GB/8GB/16GB (LPDDR4/4x) memory and 32GB/64GB/128GB/256GB onboard eMMC, with up to 8K display processing capability.

Orange Pi 5B introduces quite a lot of interfaces, including HDMI output, Type-C, WIFI6, Bluetooth, 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 5B 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 5B

We can use it to achieve:

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

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




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

Introduction to hardware features	
CPU	<ul style="list-style-type: none"><li>• Rockchip RK3588S (8nm LP process)</li><li>• 8-core 64-bit processor</li><li>• 4-core Cortex-A76 and 4-core Cortex-A55 core architecture</li><li>• 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</li></ul>
GPU	<ul style="list-style-type: none"><li>• Integrated ARM Mali-G610</li><li>• OpenGL ES1.1/2.0/3.2, OpenCL 2.2 and Vulkan 1.2</li></ul>
NPU	<ul style="list-style-type: none"><li>• Built-in AI accelerator NPU with a computing power of up to 6 Tops</li><li>• Support INT4/INT8/INT16 mixed operation</li></ul>
video output	<ul style="list-style-type: none"><li>• HDMI 2.1, up to 8K @60Hz</li><li>• DP1.4 (DisplayPort)</li><li>• 2 * MIPI D-PHY TX 4Lane</li></ul>
Memory	4GB/8GB/16GB (LPDDR4/4x)
Camera	<ul style="list-style-type: none"><li>• 1 * MIPI CSI 4Lane</li><li>• 2 * MIPI D-PHY RX 4Lane</li></ul>
PMU	RK806-1
onboard storage	<ul style="list-style-type: none"><li>• MicroSD (TF) Card Slot</li><li>• 32/64/128/256 GB eMMC</li></ul>
ethernet	10/100/1000Mbps ethernet (YT8531C )
WIFI+BT	Onboard WI-FI6+BT 5.0 module (AP6275P), supports BLE
audio	<ul style="list-style-type: none"><li>• 3.5mm headphone jack audio in/out</li><li>• Onboard MIC input</li><li>• HDMI output</li></ul>
USB interface	<ul style="list-style-type: none"><li>1 * USB3.0 interface</li><li>2 * USB2.0 interface (one of which is shared with Type-C interface)</li><li>1 * USB3.0 Type-C port</li></ul>
26pin extension header	Used to expand UART, PWM, I2C, SPI, CAN and GPIO



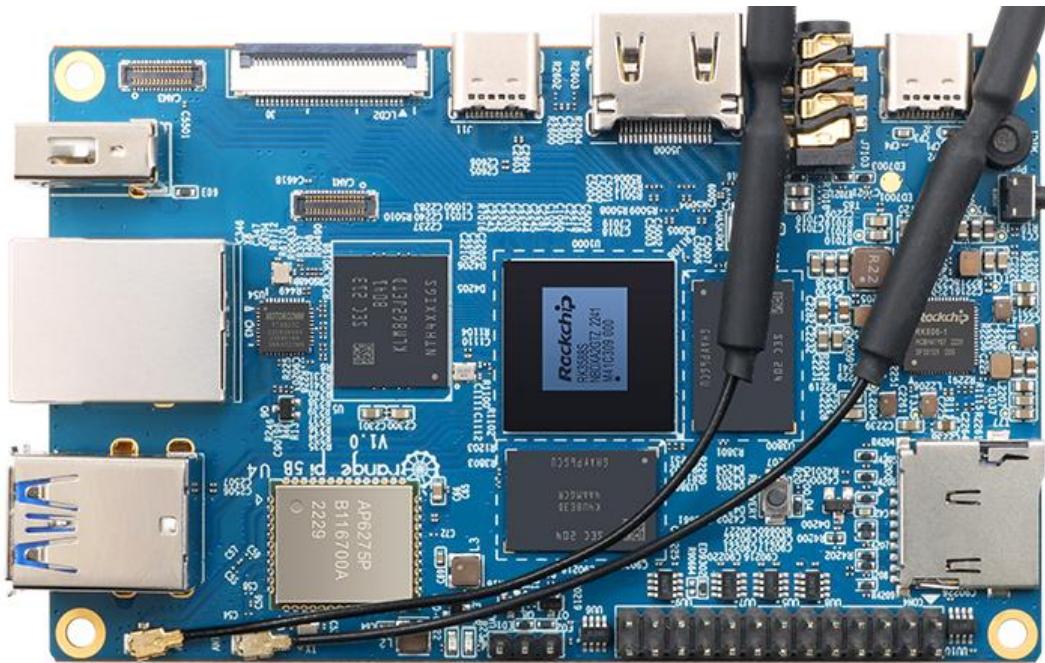


	interfaces
Debug serial port	3pin debugging serial port
LED	Power light and status light
button	1 * MaskROM key, 1 * RECOVERY, 1 * switch key
Power supply	Type-C interface power supply 5V/4A;
Supported OS	Operating systems such as Orange Pi OS (Droid), Android12.1, Debian11, Debian12, Ubuntu20.04 and Ubuntu22.04
<b>Introduction of Appearance Specifications</b>	
Product Size	100mm*62mm
weight	46g
 range Pi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd.	

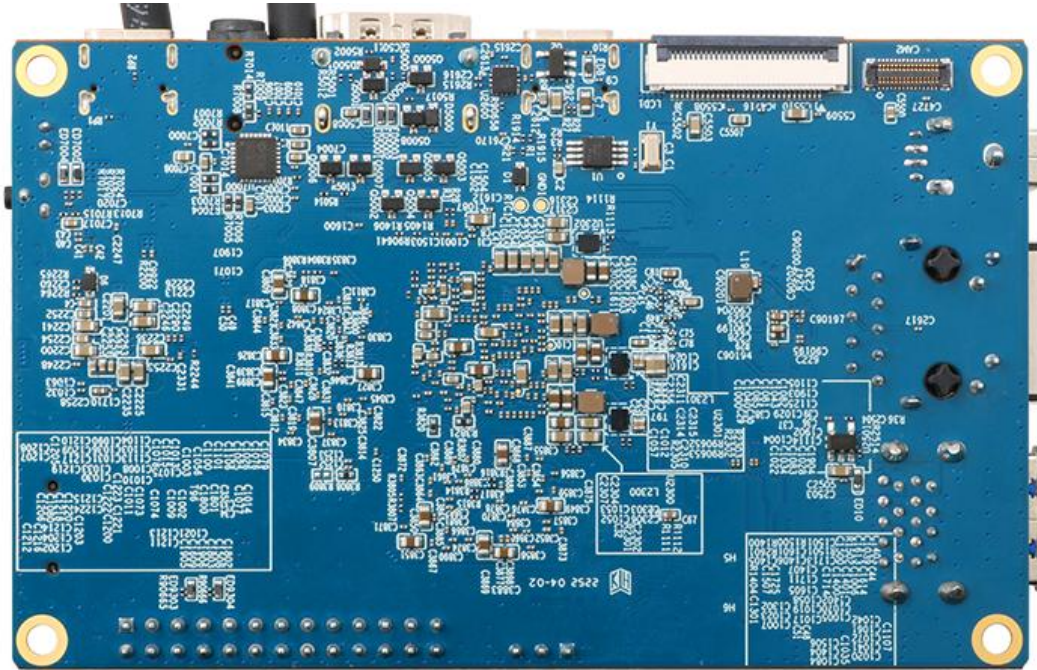


## 1. 4. Top view and bottom view of Orange Pi 5B

Top view:

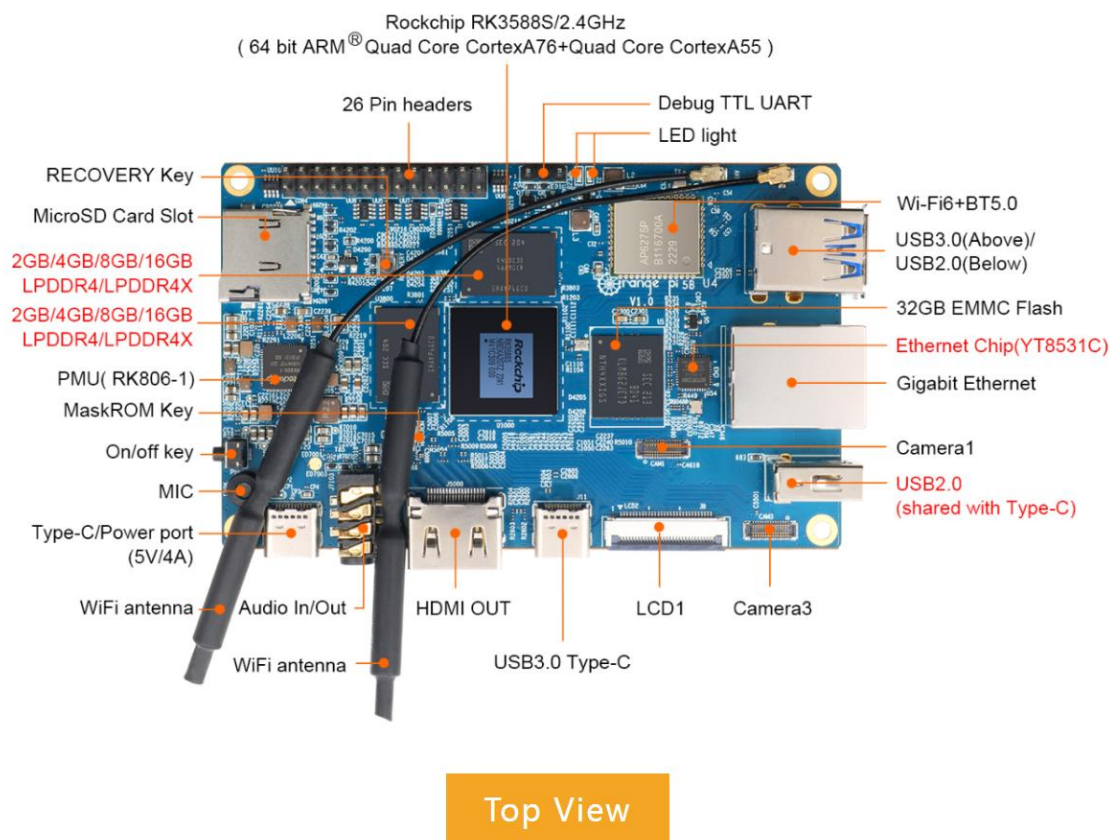


Bottom view:

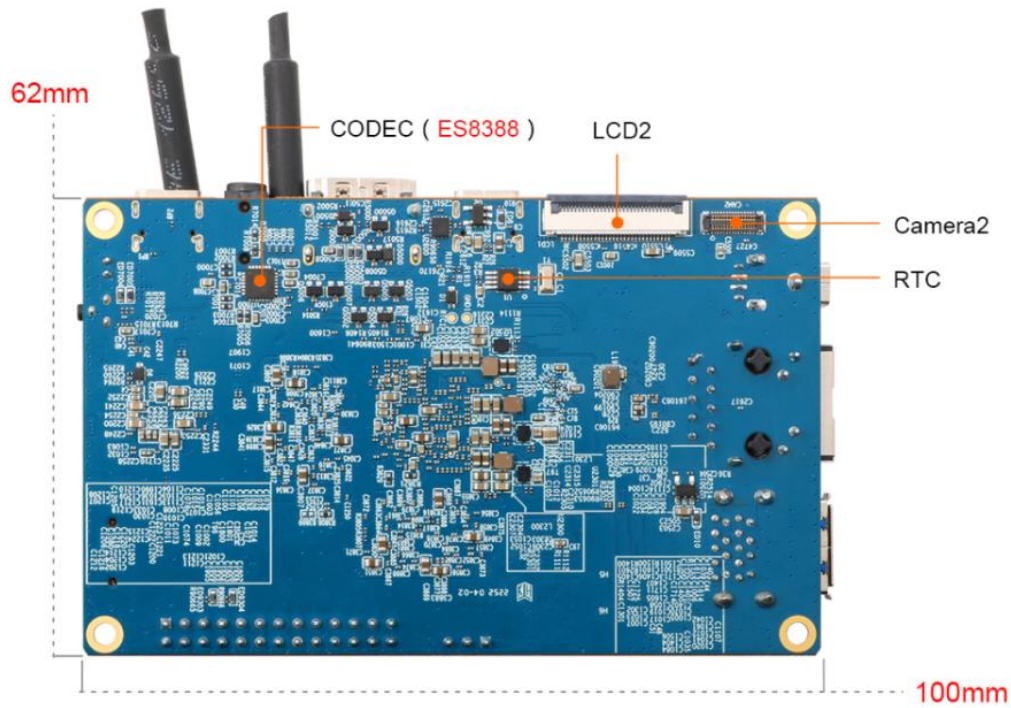




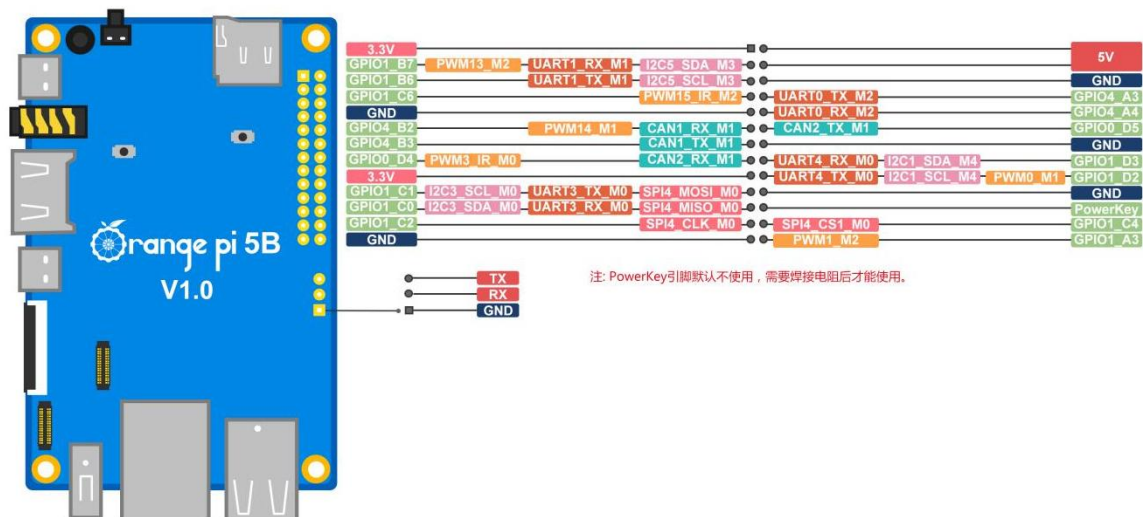
## 1. 5. Interface details of Orange Pi 5B







### Bottom View



The diameter of the four positioning holes is 3.0mm.

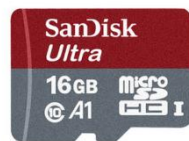


## 2. Introduction to the use of the development board

### 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)

SanDisk 闪迪



- 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 (this screen includes the adapter board and OPi5Plus/OPi5)





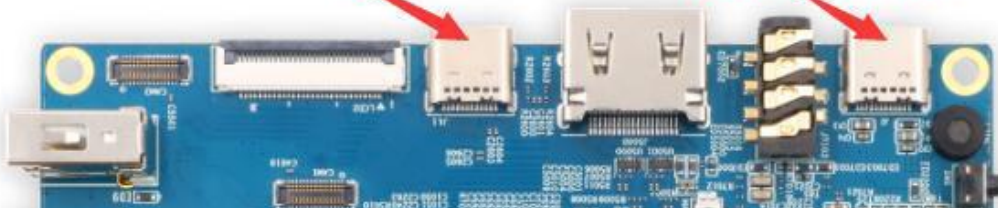
8) Power adapter, Orange Pi 5B is recommended to use 5V/4A Type-C power supply for power supply



**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.**

**This interface has no power supply function**

**Type-C Power port**



**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

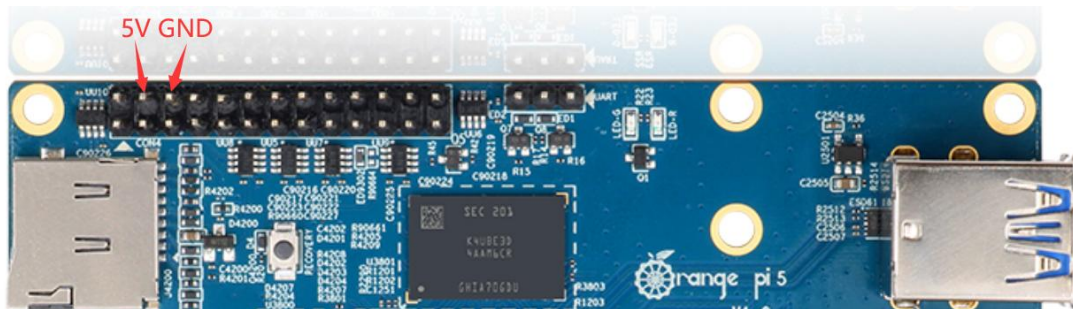


10) USB camera



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).**

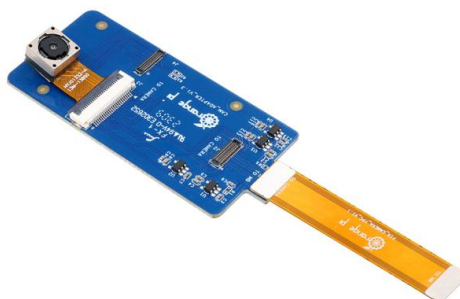


12) 100M or 1000M network cable, used to connect the development board to the Internet

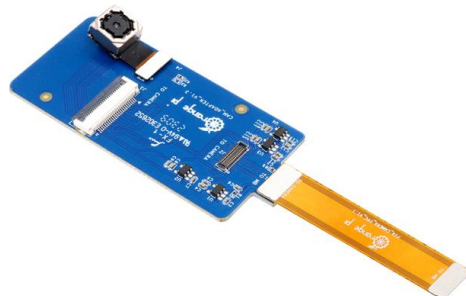
13) Data cable of Type-C interface, used for burning image, using ADB and other functions



14) OV13850 camera with 13 million MIPI interface



15) OV13855 camera with 13 million MIPI interface



## 16) Matching shell



17) **3.3V** USB to TTL module and DuPont line, when using serial port debugging function, need USB to TTL module and DuPont line to connect the development board and computer



## 18) Personal computer with Ubuntu and Windows operating systems installed

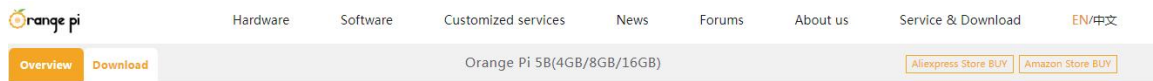
1	Ubuntu22.04 PC	Optional, used to compile Linux source code
2	Windows PC	For burning Android and Linux images

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

1) The website for downloading the English version is:



<http://www.orange-pi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-pi-5.html>



## Orange Pi 5B (4GB/8GB/16GB)



### Official Resources



User Manual

[Downloads](#)



Schematic

[Downloads](#)



Mechanical

[Downloads](#)



Official Tools

[Downloads](#)

### Official Images



Orange Pi OS(Arch)

[Downloads](#)



Orange Pi OS(Droid)

[Downloads](#)



Ubuntu Image

[Downloads](#)



Debian Image

[Downloads](#)



Android Image

[Downloads](#)



Android Source Code

[Downloads](#)



Linux Source code

[Downloads](#)

2) The information mainly includes

- Android source code:** Save on Google Drive
- Linux source code:** saved on Github
- User Manual and Schematic:** Save on Google Drive
- Official tools:** It mainly includes the software that needs to be used in the



process of using the development board

- e. **Android image:** Save on Google Drive
- f. **Ubuntu image:** Save on Google Drive
- g. **Debian image:** Save on Google Drive
- h. **Orange Pi OS image:** Save on Google Drive

## 2.3. How to burn Linux image to TF card

### 2.3.1. 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.**

### 2.3.2. How to burn Linux image using balenaEtcher

1) First prepare a TF card with a capacity of 16GB or larger. The transmission speed of the TF card must be **class10** or above. It is recommended to use TF cards from SanDisk and other brands.

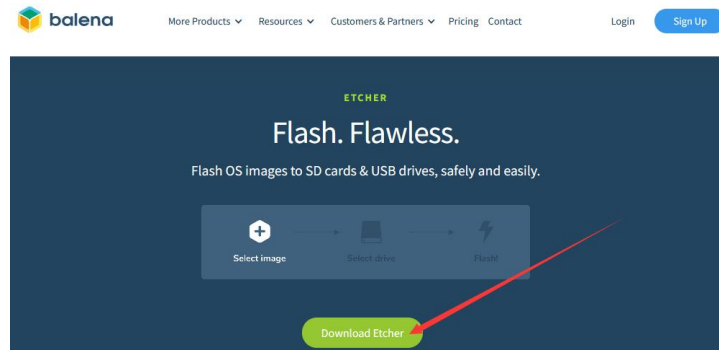
2) Then use the card reader to insert the TF card into the computer

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

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

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

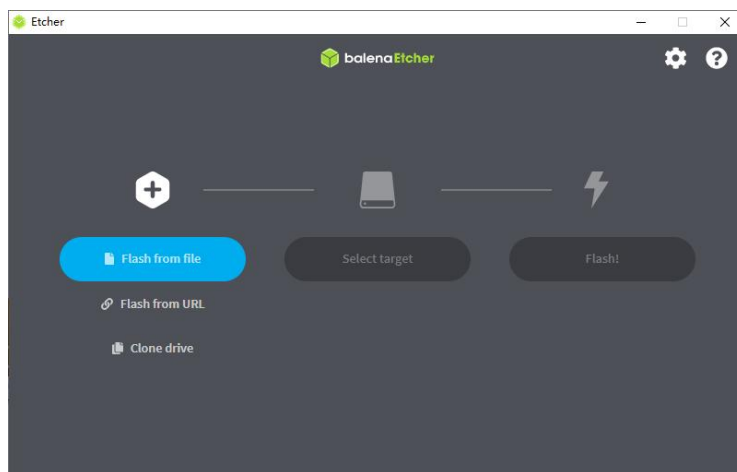
5) After entering the balenaEtcher download page, click the green download button to jump to the software download location.



6) Then you can choose to download the Portable version of balenaEtcher software. The Portable version does not need to be installed. You can use it by double-clicking to open it.

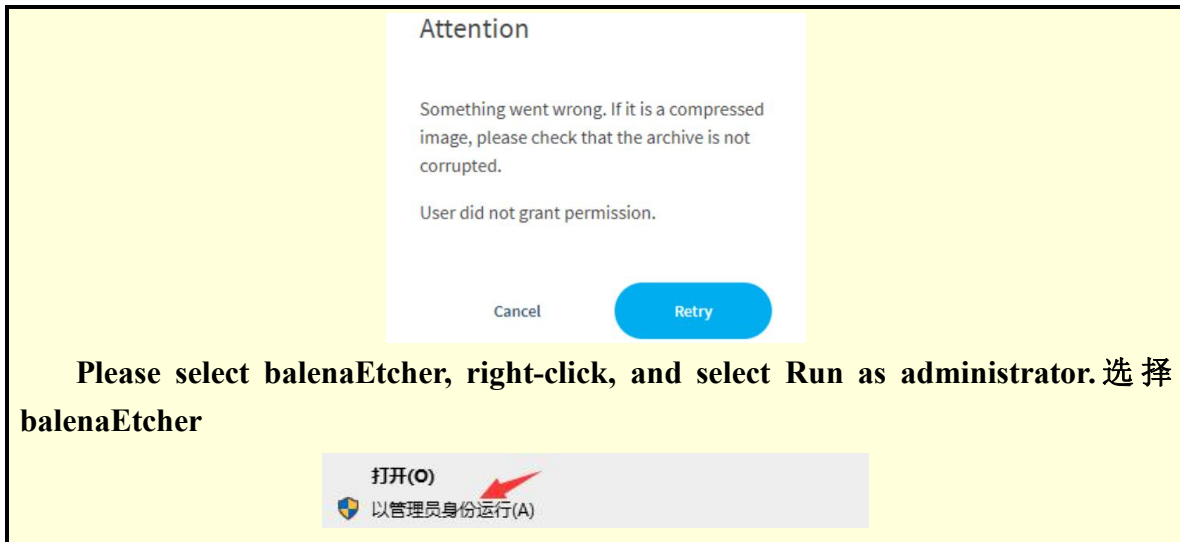


7) If you downloaded the version of balenaEtcher that needs to be installed, please install it first and then use it. If you downloaded the Portable version of balenaEtcher, just double-click to open it. The balenaEtcher interface after opening is as shown below:



**If the following error is prompted when opening balenaEtcher:**

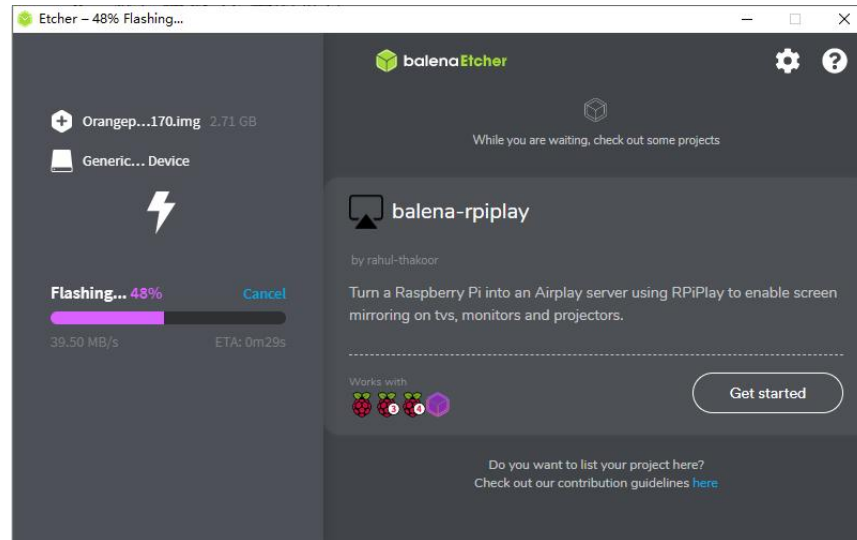




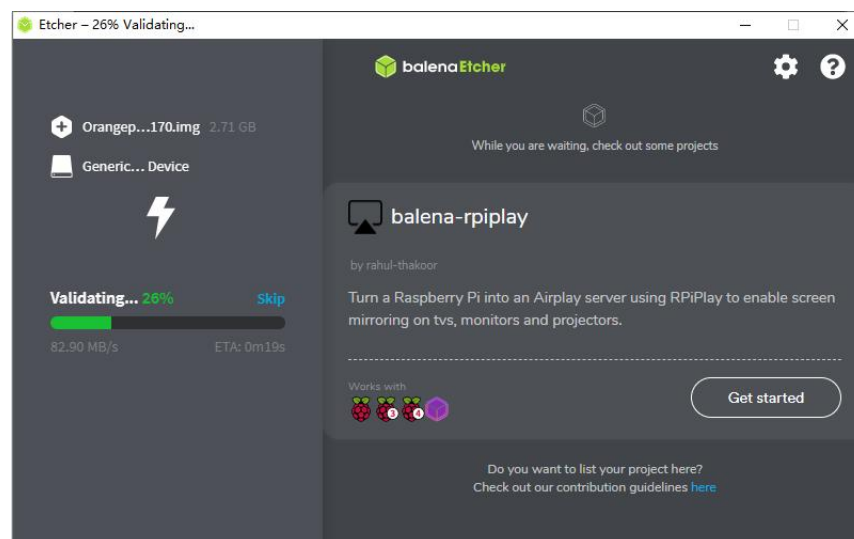
- 8) The specific steps to use balenaEtcher to burn a Linux image are as follows
- First select the path of the Linux image file to be burned.
  - Then select the drive letter of the TF card
  - Finally click Flash and it will start burning the Linux image to the TF card.



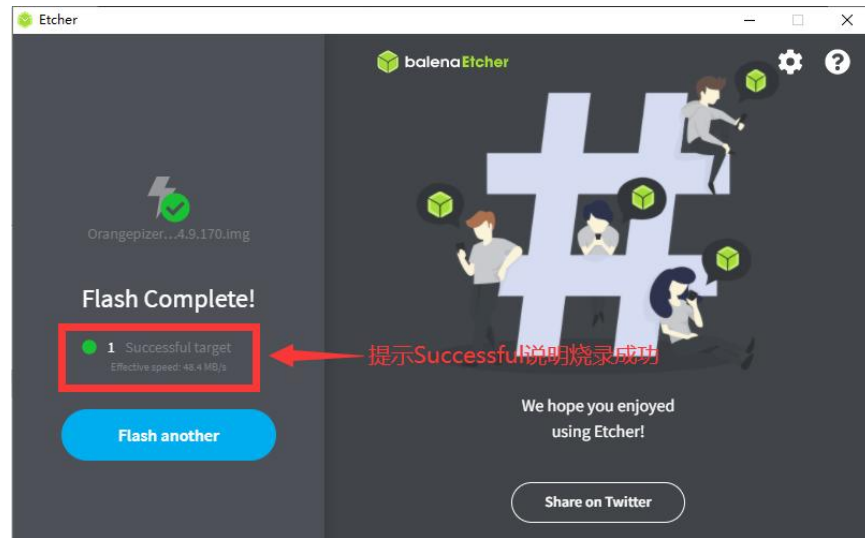
- 9) The interface displayed during the process of burning the Linux image by balenaEtcher is as shown in the figure below. In addition, the progress bar displays purple to indicate that the Linux image is being burned into the TF card.



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



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



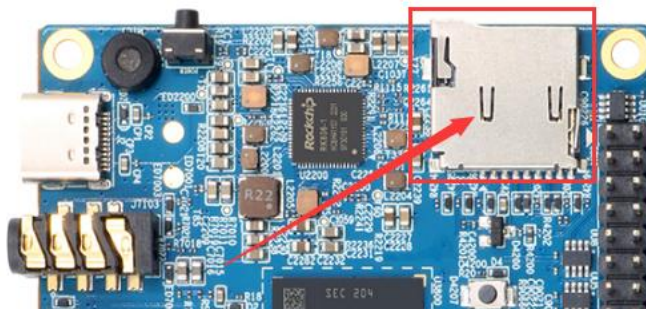
## 2. 3. 2. 1. How to use RKDevTool to burn Linux image to TF card

1) First, you need to prepare a data cable with a good quality Type-C interface



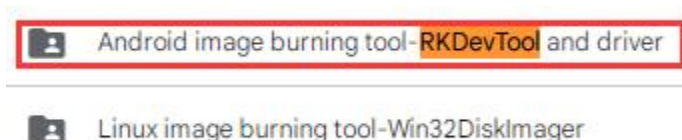
2) You also need to prepare a 16GB or larger capacity TF card. The transmission speed of the TF card must be class 10 or above. It is recommended to use a TF card of SanDisk and other brands

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

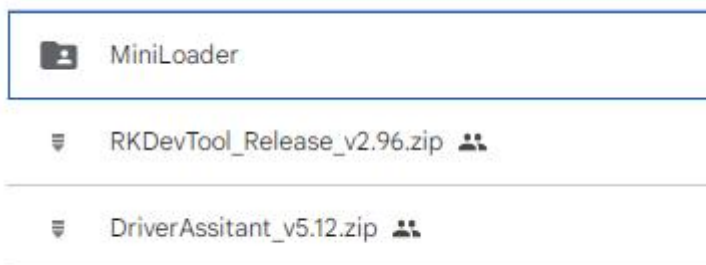


4) Then download the Rockchip microdriver **DriverAssitant\_v5.12.zip** and **MiniLoader** and the burning tool **RKDevTool\_Release\_v3.15.zip** from the [Orange Pi data download page](#)

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



- b. Then download all the files below



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

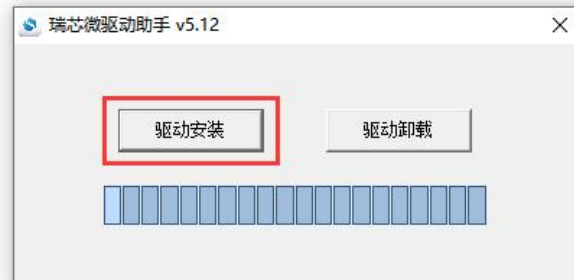
5) Then download the Linux operating system image file compression package that you want to burn from [the Orange Pi data download page](#), and then use the decompression software to decompress it. Among the decompressed files, the file ending with ".img" is the image file of the operating system ,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

6) Then use the decompression software to decompress **DriverAssitant\_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

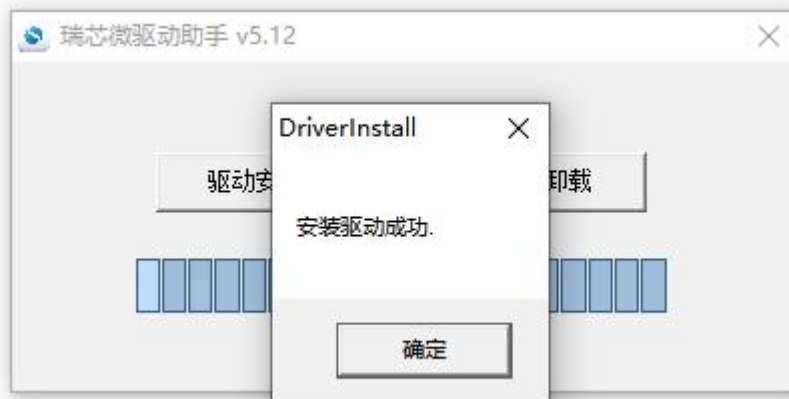
名称	修改日期	类型	大小
ADBDriver	2022/12/1 15:07	文件夹	
bin	2022/12/1 15:07	文件夹	
Driver	2022/12/1 15:07	文件夹	
config	2014/6/3 15:38	配置设置	1 KB
<b>DriverInstall</b>	2022/2/28 14:11	应用程序	491 KB
Readme	2018/1/31 17:44	文本文档	1 KB
revision	2022/2/28 14:14	文本文档	1 KB

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

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



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

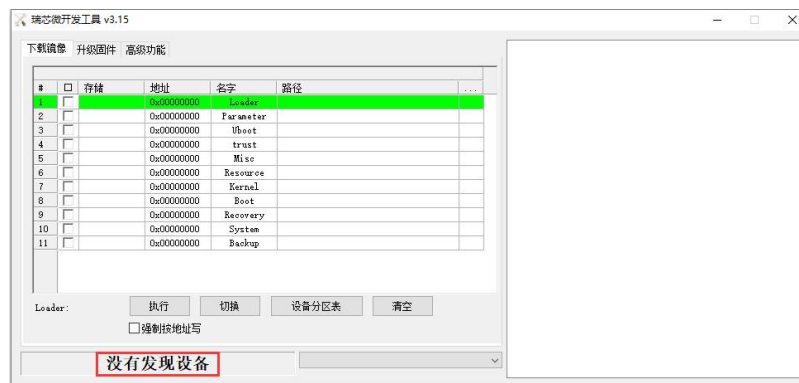


8) Then decompress **RKDevTool\_Release\_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it



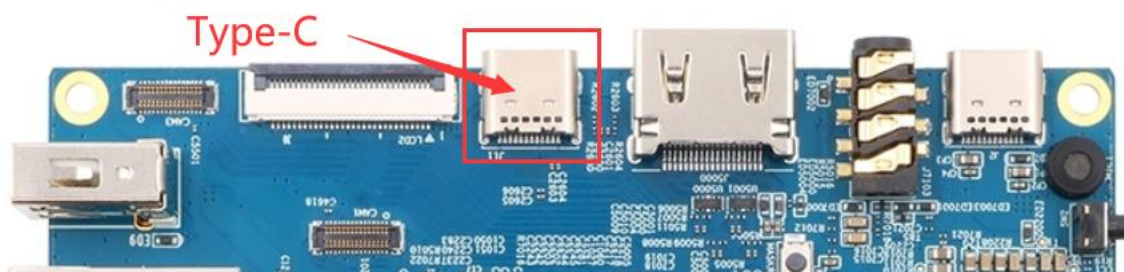
名称	修改日期	类型	大小
bin	2022/12/1 15:07	文件夹	
Language	2022/12/1 15:07	文件夹	
config.cfg	2022/3/23 9:11	CFG 文件	7 KB
config	2021/11/30 11:04	配置设置	2 KB
revision	2022/5/27 9:09	文本文档	3 KB
<b>RKDevTool</b>	2022/5/27 9:06	应用程序	1,212 KB
开发工具使用文档_v1.0	2021/8/27 10:28	Foxit PDF Reade...	450 KB

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



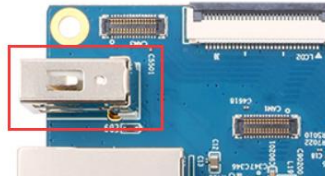
10) Then start burning the Linux image into eMMC

- 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 the development board is 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:

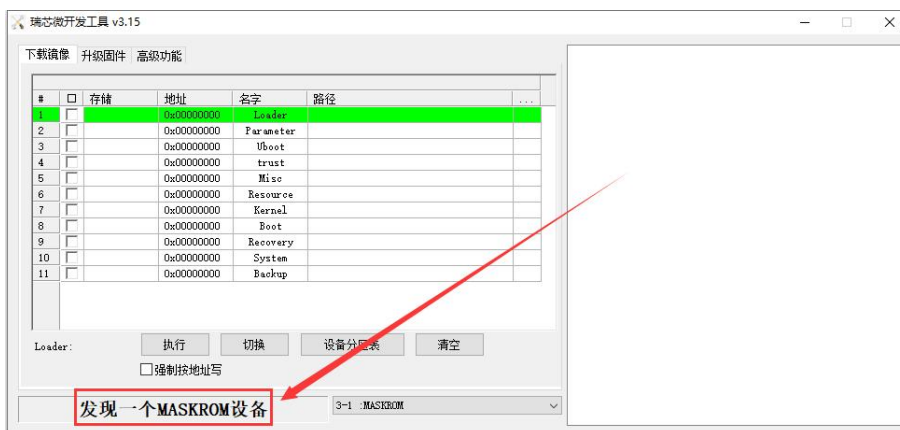


- e. Then connect the power supply of the Type-C interface to the development board and power on

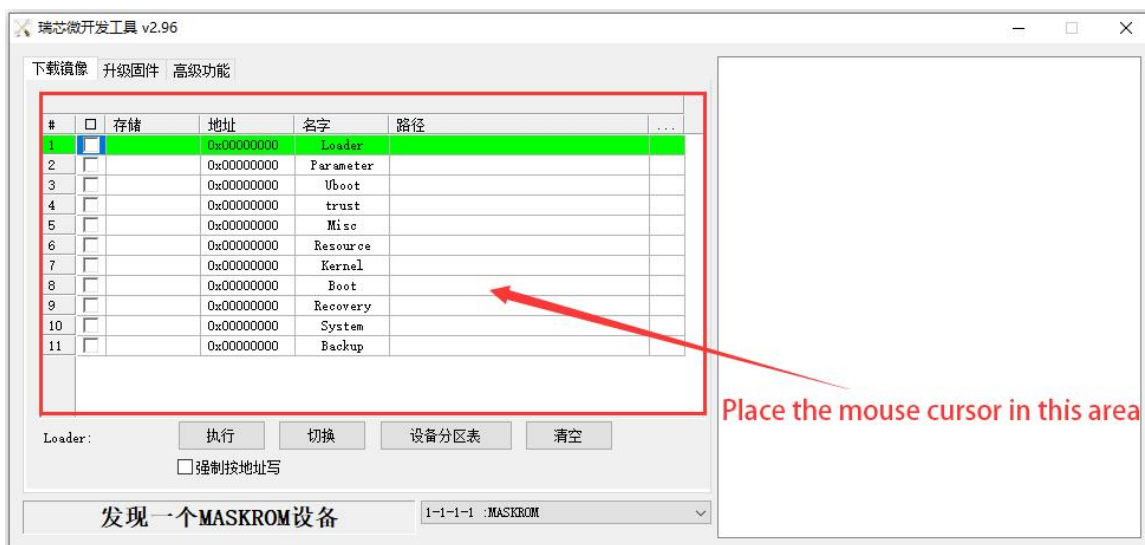


- f. If the previous steps are successful, the development board will enter the MASKROM mode at this time, and the interface of the burning tool will prompt "**found a MASKROM device**"

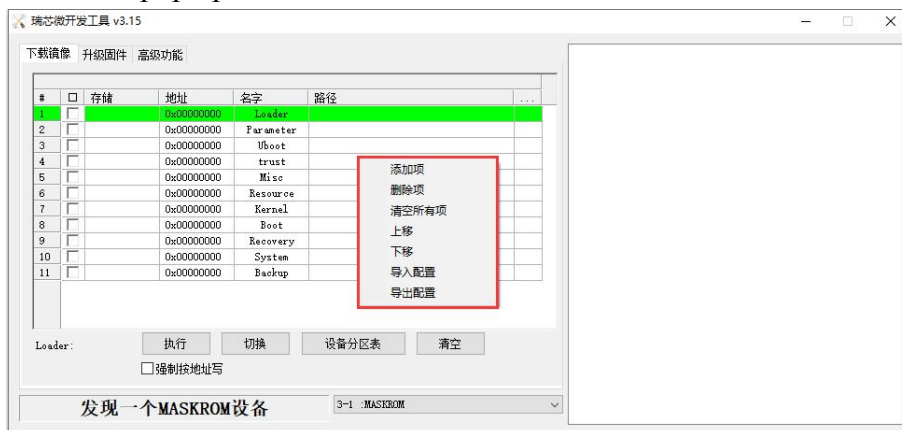




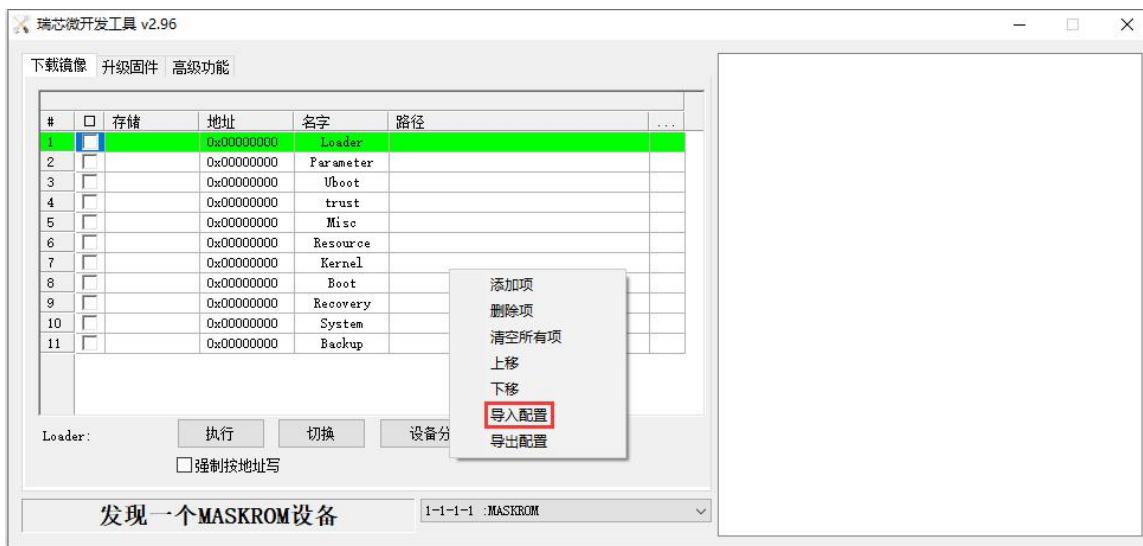
g. Then place the mouse cursor in the area below



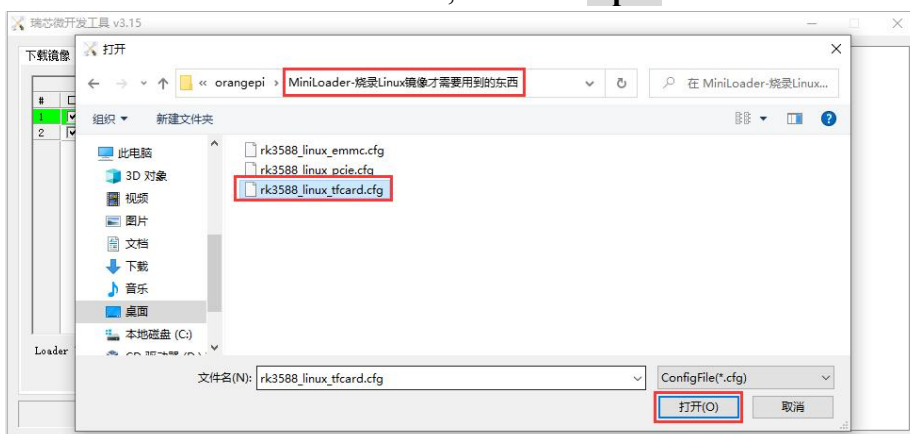
h. Then click the right mouse button and the selection interface shown in the figure below will pop up



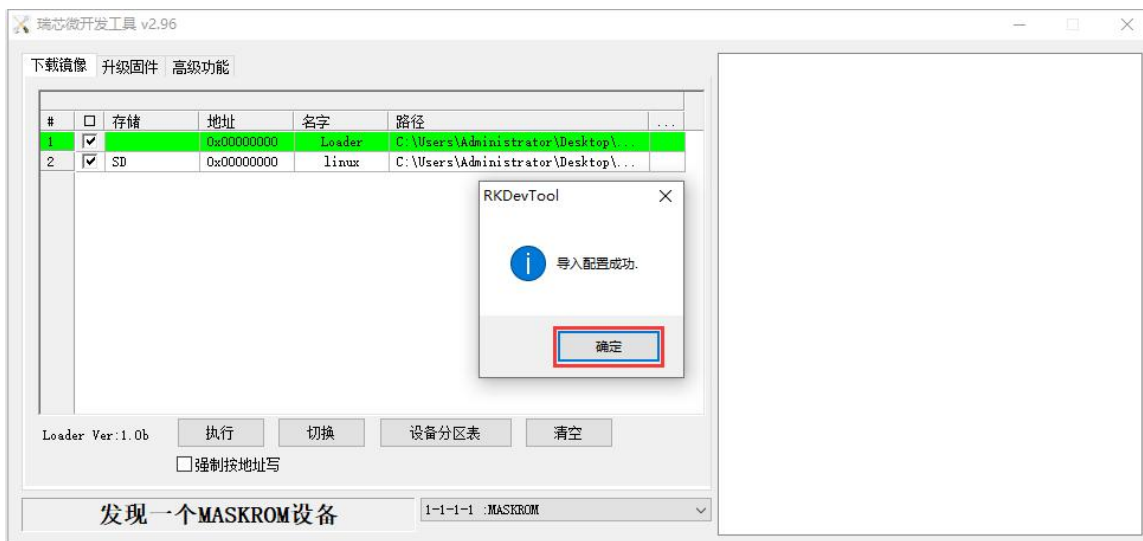
i. Then select the **import configuration** option



- j. Then select the **rk3588\_linux\_tfc card.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and click **Open**

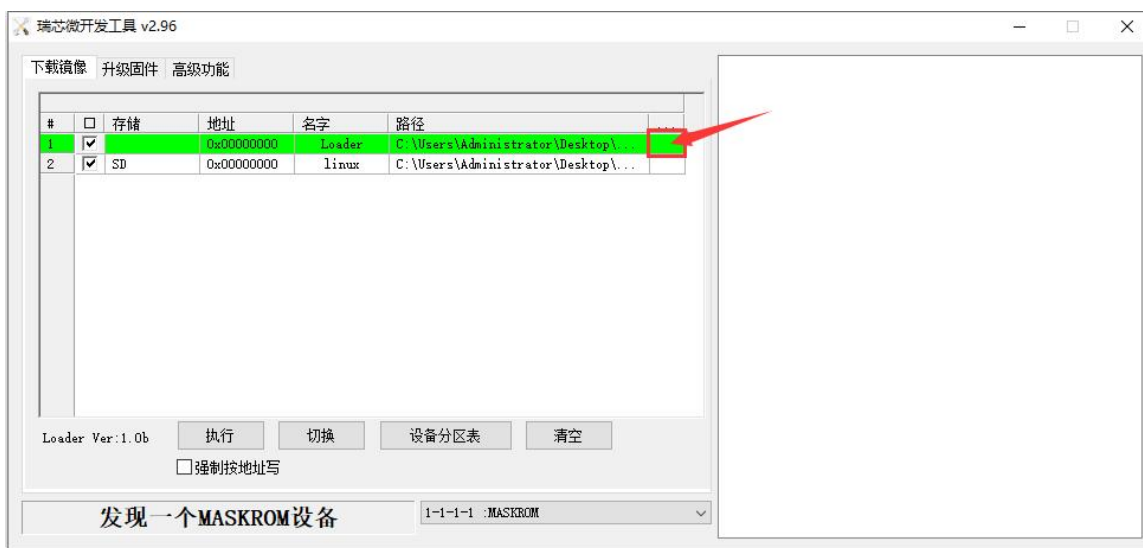


- k. Then click **OK**

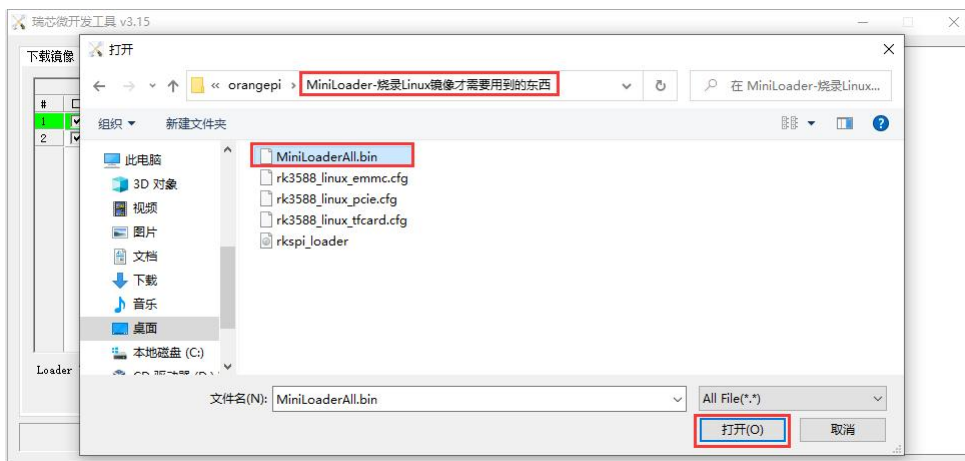




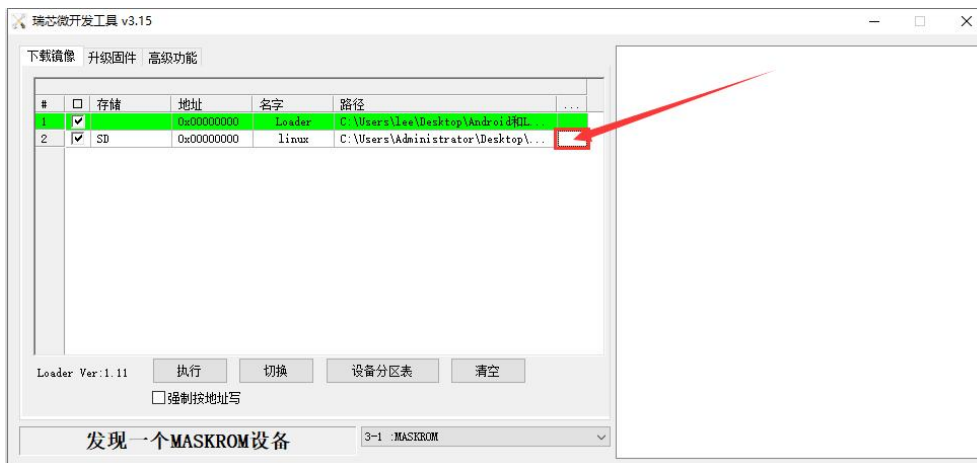
1. Then click the position shown in the figure below



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



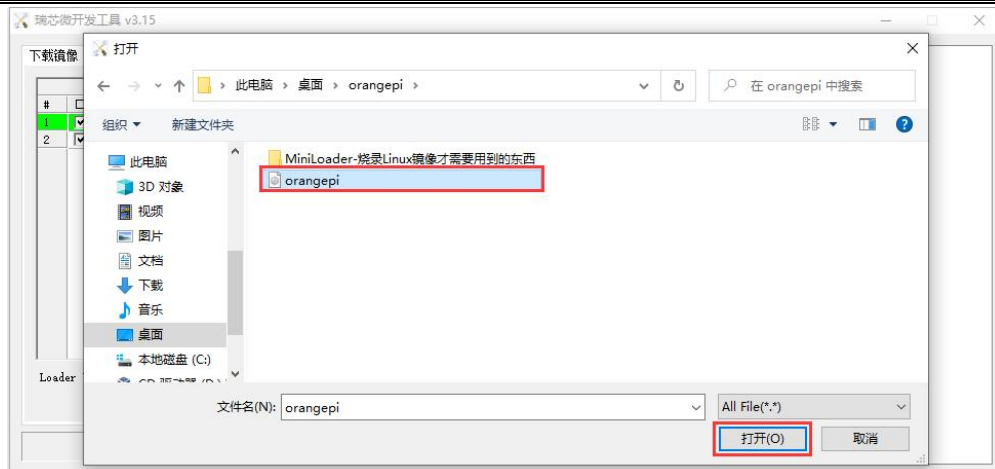
n. Then click the position shown in the figure below



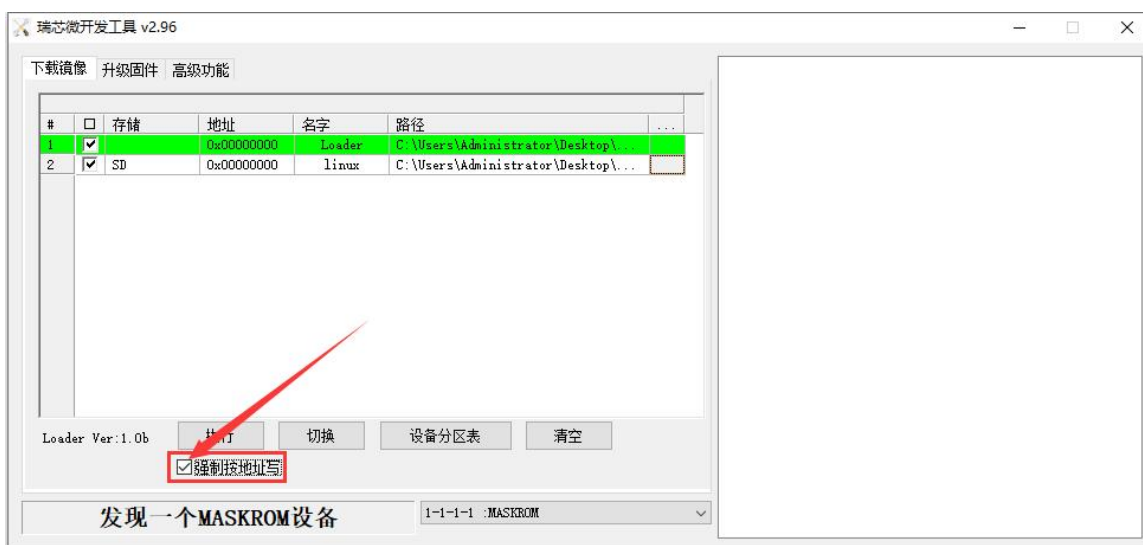


- o. Then select the path of the Linux image you want to burn, and then click **Open**

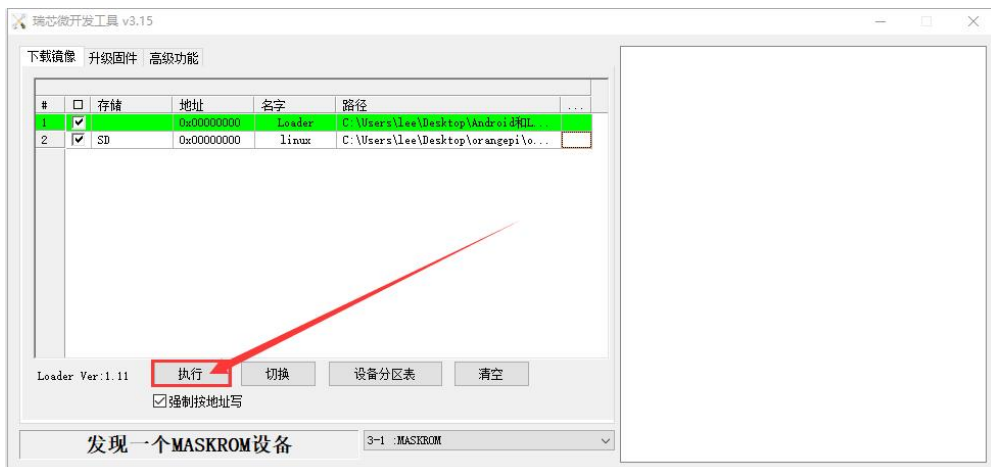
**Before burning the image, it is recommended to rename the Linux image to be burned to `orangepi.img` or another shorter name, so that you can see the percentage of the burning progress when burning the image.**



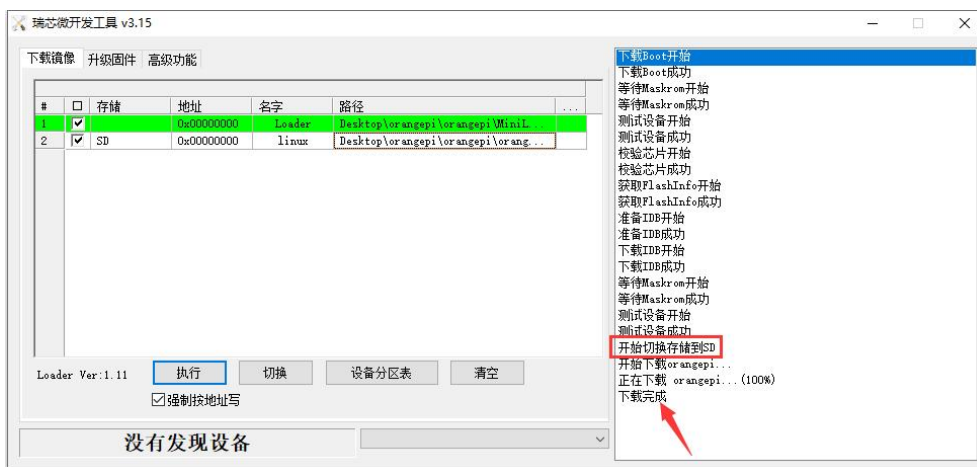
- p. Then please check the option to **force writing by address**



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



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



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

## 2. 3. 2. 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

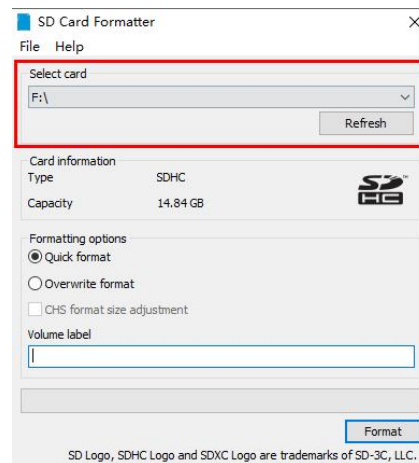
3) Then format the TF card

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

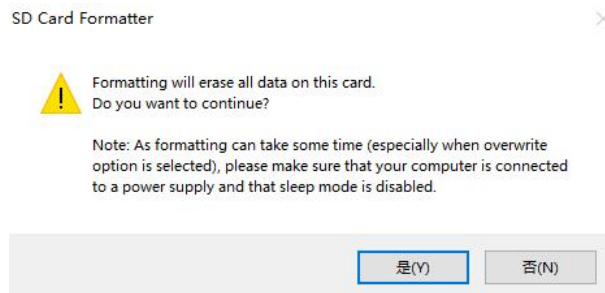


[https://www.sdcard.org/downloads/formatter/eula\\_windows/SDCardFormatterv5\\_WinEN.zip](https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip)

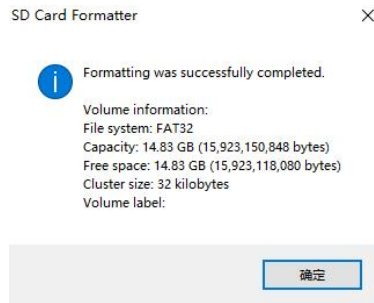
- b. After downloading, unzip and install directly, and then open the software
- c. If only a TF card is inserted into the computer, the drive letter of the TF card will be displayed in the "Select card" column. If multiple USB storage devices are inserted into the computer, you can select the corresponding 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

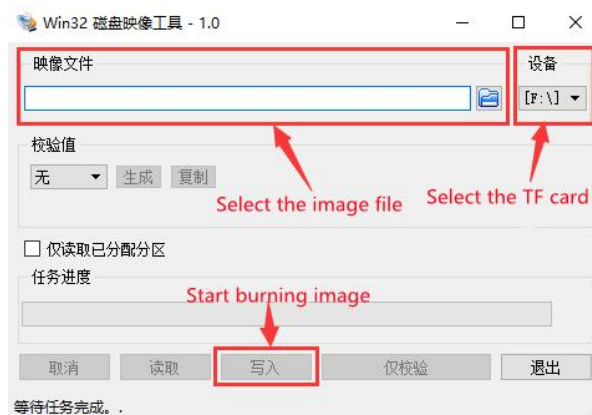
<http://sourceforge.net/projects/win32diskimager/files/Archive/>

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. 3. 3. 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.**

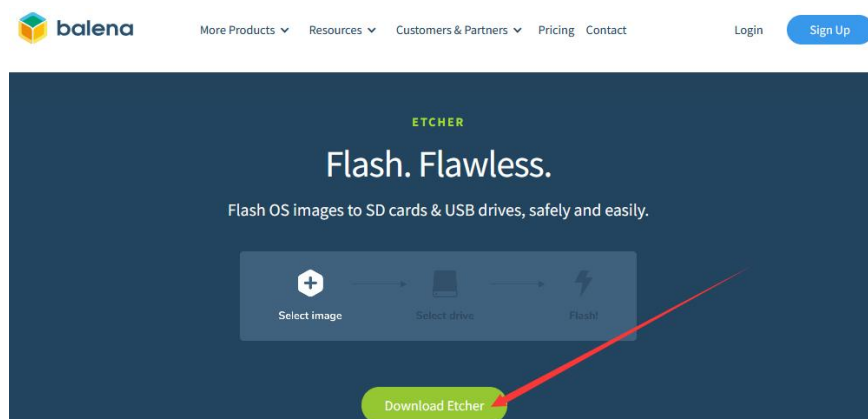
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brands

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, click the green download button to jump to the software download place.



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



DOWNLOAD

## Download Etcher

ASSET	OS	ARCH	
ETCHER FOR WINDOWS (X86 X64) (INSTALLER)	WINDOWS	X86 X64	<a href="#">Download</a>
ETCHER FOR WINDOWS (X86 X64) (PORTABLE)	WINDOWS	X86 X64	<a href="#">Download</a>
ETCHER FOR WINDOWS (LEGACY 32 BIT) (X86 X64) (PORTABLE)	WINDOWS	X86 X64	<a href="#">Download</a>
ETCHER FOR MACOS	MACOS	X64	<a href="#">Download</a>
ETCHER FOR LINUX X64 (64-BIT) (APPIMAGE)	LINUX	X64	<a href="#">Download</a>
ETCHER FOR LINUX (LEGACY 32 BIT) (APPIMAGE)	LINUX	X86	<a href="#">Download</a>

Looking for [Debian \(.deb\) packages](#) or [Red Hat \(.rpm\) packages](#)? [OSS hosting by cloudsmith](#)

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

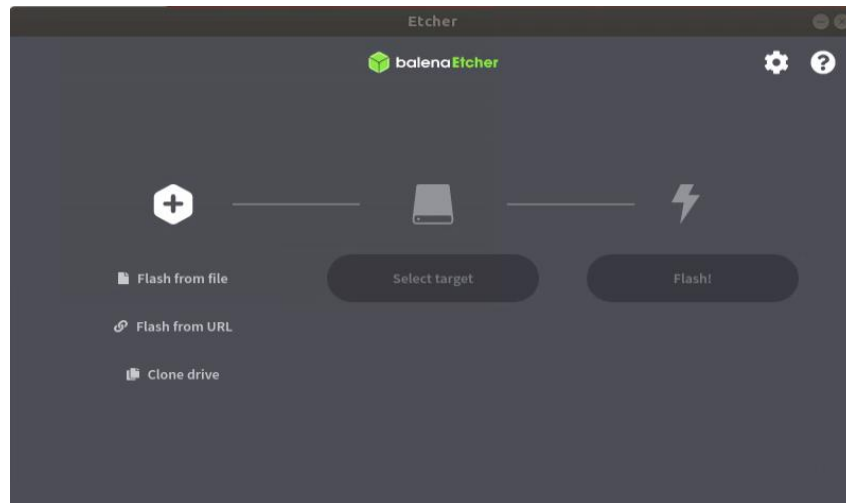
The decompression command for the compressed package ending in 7z is as follows

```
test@test:~$ 7z x orangepi5b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
test@test:~$ ls orangepi5b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.*
orangepi5b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
orangepi5b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.sha      #checksum and
file
orangepi5b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img    #image file
```

7) After decompressing the image, you can first use the `sha256sum -c *.sha` command to calculate whether the checksum is correct. If the 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
orangepi5b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img: OK
```

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



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

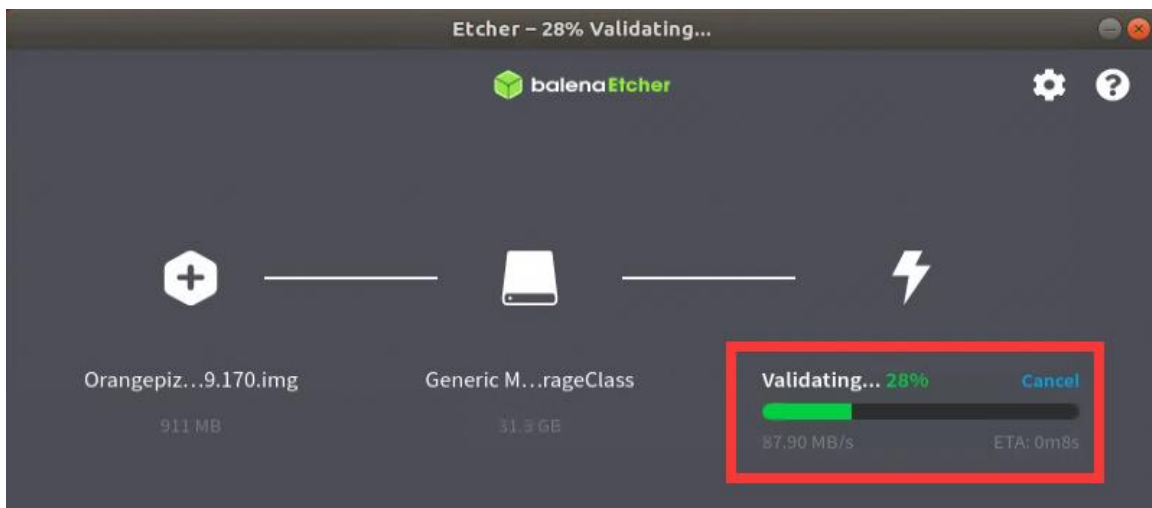
- a. First select the path of the Linux image file to be burned
- b. Then select the drive letter of the TF card
- c. Finally, click Flash to start burning the Linux image to the TF card



10) The interface displayed 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



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



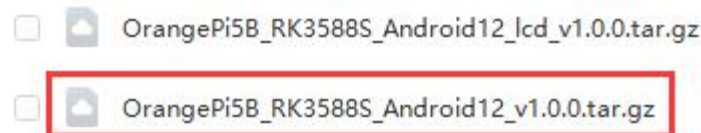
13) After successful burning, the display interface of balenaEtcher is shown in the figure below. If a green indicator icon is displayed, it means that the image burning is successful. At this time, you can exit balenaEtcher, and then pull out the TF card and insert it into the TF card slot of the development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



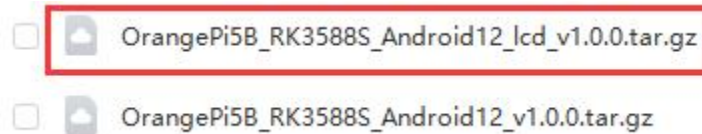
## 2. 4. How to burn Android image to TF card

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

- 1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use 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 Android image from [Orange Pi's download page](#). After opening the download link of the Android image, you can see the following two types of Android images. The difference between them is:
  - a. The image without lcd is specially used for HDMI display, if you don't use LCD screen, please download the image without lcd



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



c. The image with box is the image of TV box type



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

Language	2022/9/5 15:04	文件夹	
config	2020/3/18 17:27	配置设置	2 KB
revision	2021/4/21 18:01	文本文档	1 KB
sd_boot_config.config	2014/9/3 9:52	CONFIG 文件	1 KB
<b>SD_Firmware_Tool</b>	2021/4/21 17:57	应用程序	698 KB
SDBoot.bin	2015/9/29 17:13	BIN 文件	149 KB

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 be sure to confirm 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

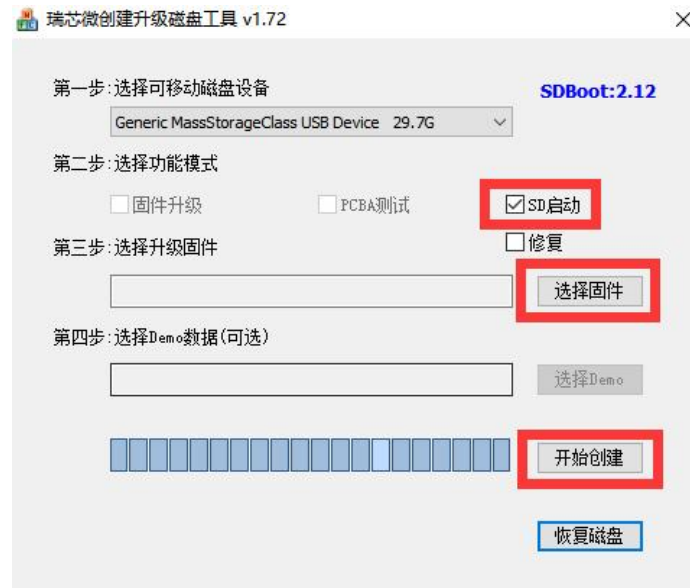


8) After confirming the drive letter, you can format the TF card first, click **the restore disk** button in **SDDiskTool**, or use the **SD Card Formatter** mentioned above to format the TF card



- 9) Then start writing the Android image to the TF card
- First check "SD Boot" in "Select Function Mode"
  - Then select the path of the Android image in the "Select to upgrade firmware" column
  - 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. 5. How to burn Orange Pi OS (Droid) image to TF card

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

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



- 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

Language	2022/9/5 15:04	文件夹	
config	2020/3/18 17:27	配置设置	2 KB
revision	2021/4/21 18:01	文本文档	1 KB
sd_boot_config.config	2014/9/3 9:52	CONFIG 文件	1 KB
<b>SD_Firmware_Tool</b>	2021/4/21 17:57	应用程序	698 KB
SDBoot.bin	2015/9/29 17:13	BIN 文件	149 KB

- 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 be sure to confirm 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
- First check "SD Boot" in "Select Function Mode"
  - Then select the path of the Orange Pi OS (Droid) image in the "Select to upgrade firmware" column
  - 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



## 2.6. How to burn Linux image into eMMC

### 2.6.1. How to use RKDevTool to burn Linux image to eMMC

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

**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 data cable with a good quality Type-C interface

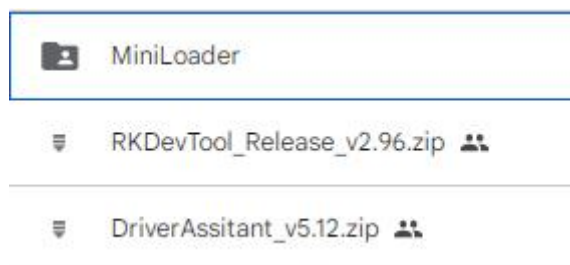


2) Then download Rockchip **DriverAssitant\_v5.12.zip** and **MiniLoader** and the burning tool **RKDevTool\_Release\_v3.15.zip** from [the Orange Pi data download page](#)

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



- b. Then download all the files below



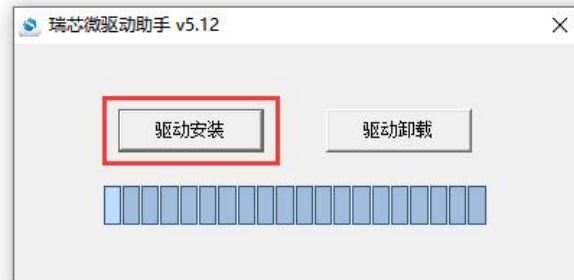
3) Then download the Linux operating system image file compression package that you want to burn from [the Orange Pi data download page](#), and then use the decompression software to decompress it. Among the decompressed files, the file ending with ".img" is the image file of the operating system, the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

4) Then use the decompression software to decompress **DriverAssitant\_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

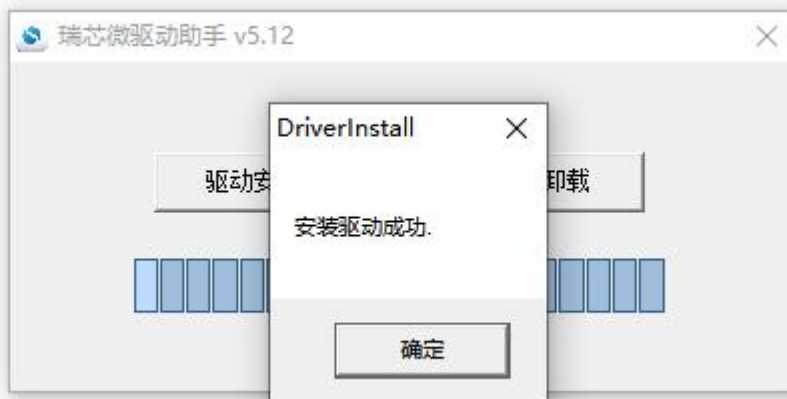
名称	修改日期	类型	大小
ADBDriver	2022/12/1 15:07	文件夹	
bin	2022/12/1 15:07	文件夹	
Driver	2022/12/1 15:07	文件夹	
config	2014/6/3 15:38	配置设置	1 KB
<b>DriverInstall</b>	2022/2/28 14:11	应用程序	491 KB
Readme	2018/1/31 17:44	文本文档	1 KB
revision	2022/2/28 14:14	文本文档	1 KB

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

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



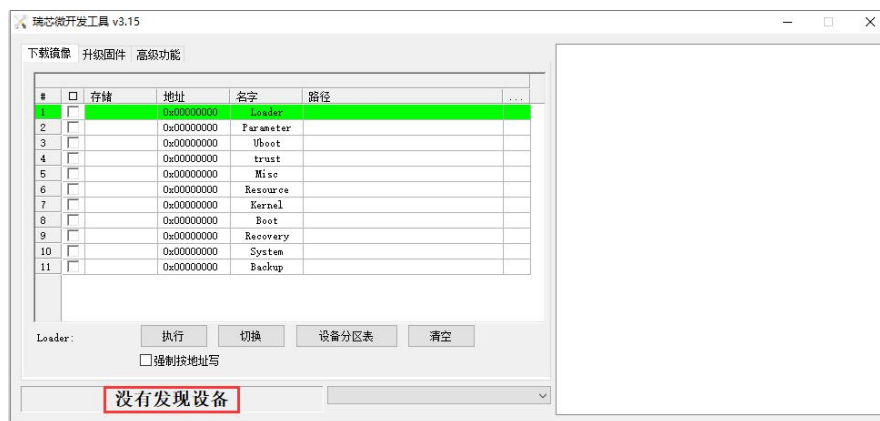
- b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button.



6) Then decompress **RKDevTool\_Release\_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

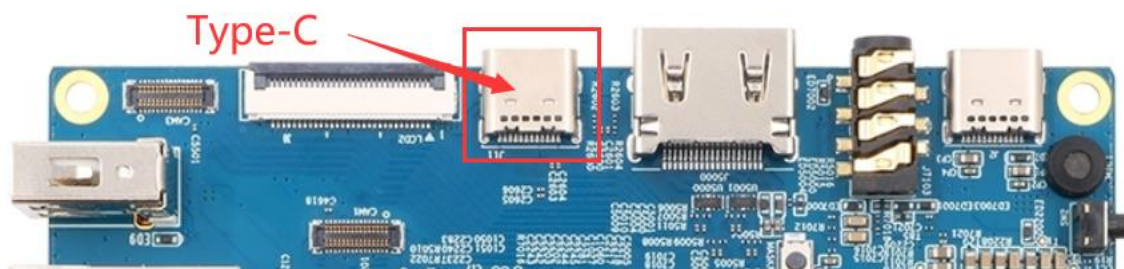
名称	修改日期	类型	大小
bin	2022/12/1 15:07	文件夹	
Language	2022/12/1 15:07	文件夹	
config.cfg	2022/3/23 9:11	CFG 文件	7 KB
config	2021/11/30 11:04	配置设置	2 KB
revision	2022/5/27 9:09	文本文档	3 KB
<b>RKDevTool</b>	2022/5/27 9:06	应用程序	1,212 KB
开发工具使用文档_v1.0	2021/8/27 10:28	Foxit PDF Reade...	450 KB

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



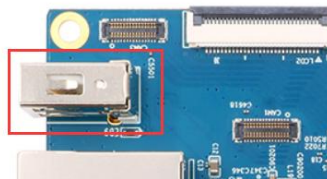
8) Then start burning the Linux image into eMMC

- 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





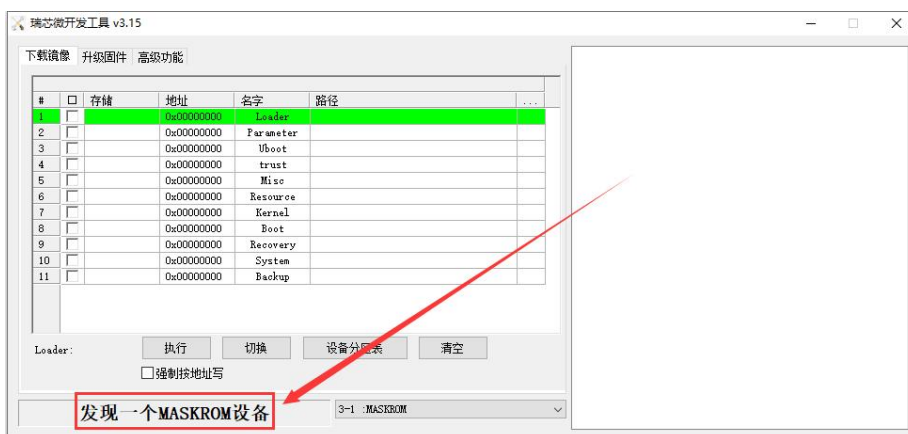
- d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:



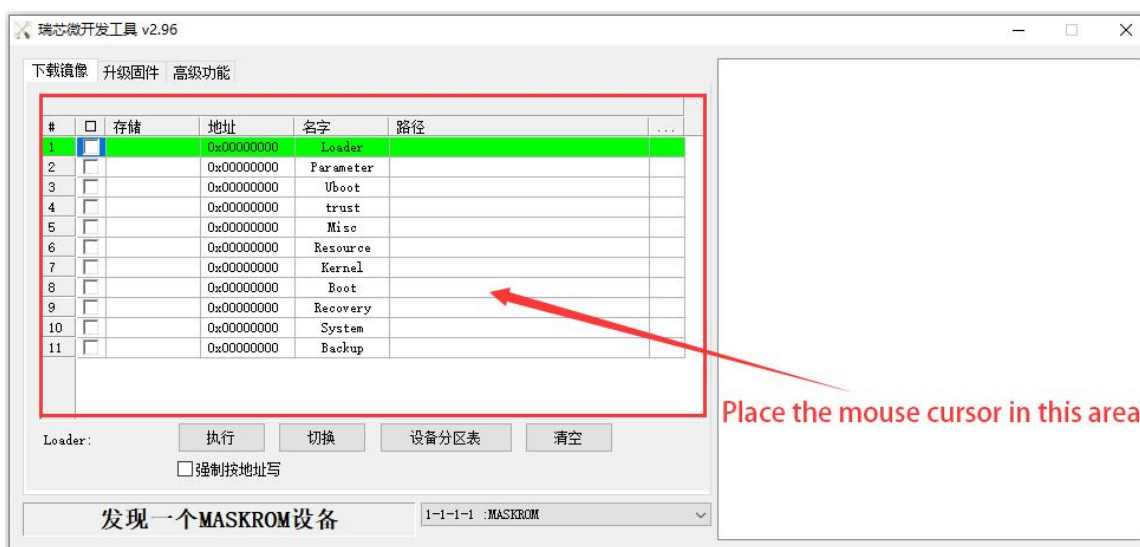
- e. Then connect the power supply of the Type-C interface to the development board and power on



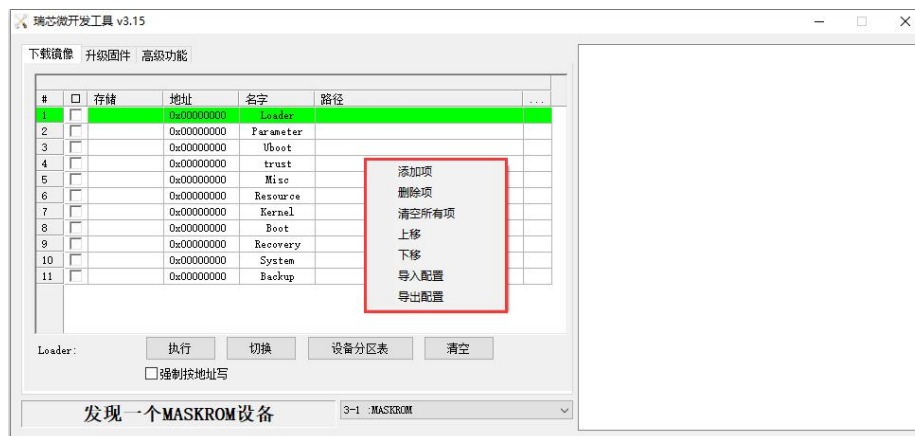
- f. If the previous steps are successful, the development board will enter the **MASKROM** mode at this time, and the interface of the burning tool will prompt "found a MASKROM device"



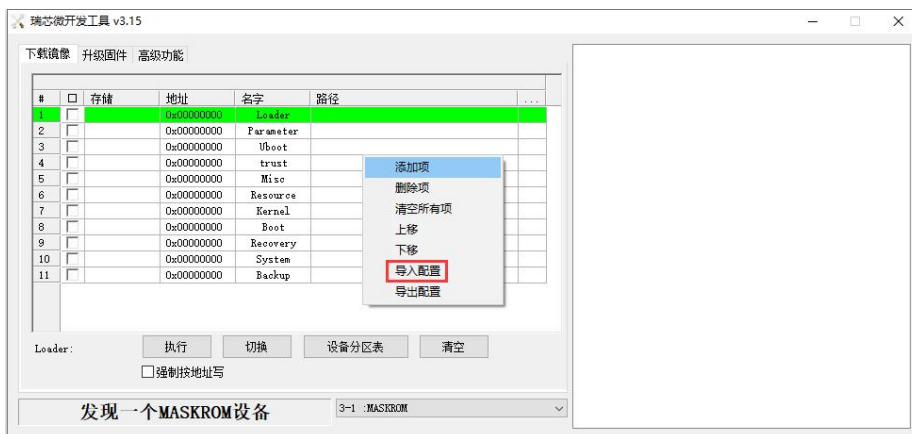
g. Then place the mouse cursor in the area below



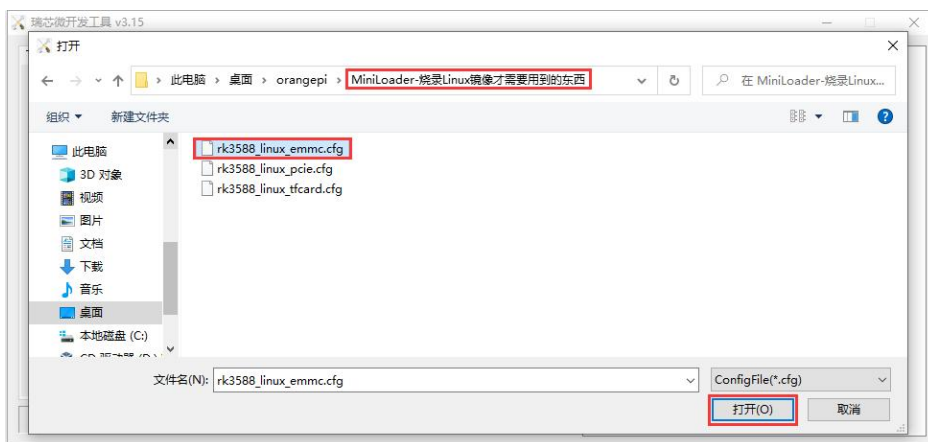
h. Then click the right mouse button and the selection interface shown in the figure below will pop up



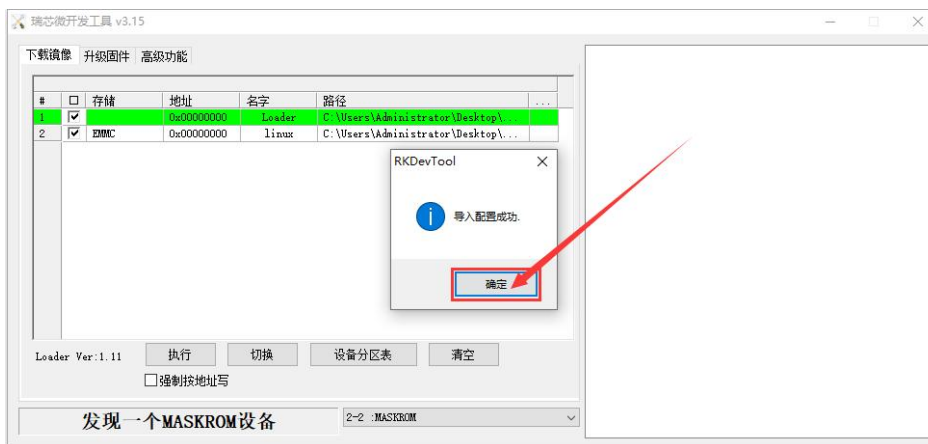
i. Then select **the import configuration** option



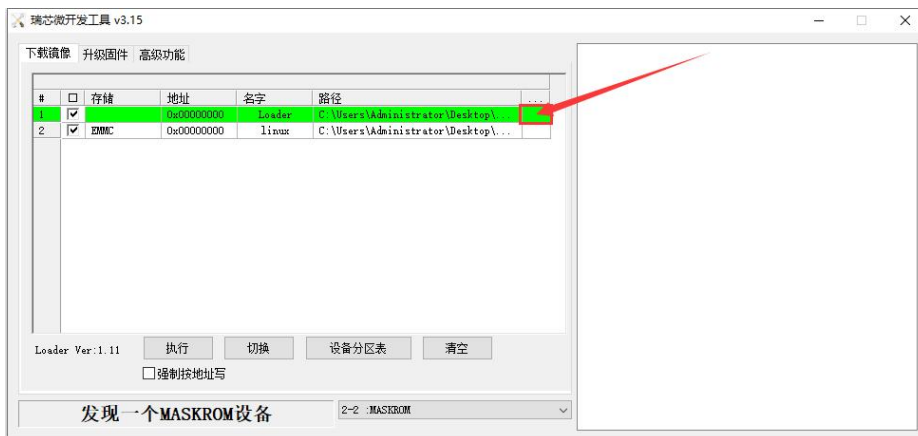
- j. Then select the **rk3588\_linux\_emmc.cfg** configuration file in the **MiniLoader** folder downloaded earlier, and click **Open**



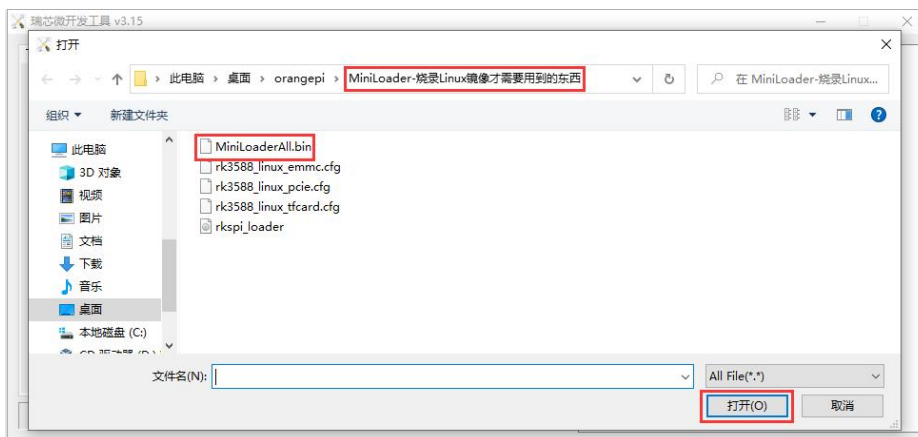
- k. Then click **OK**



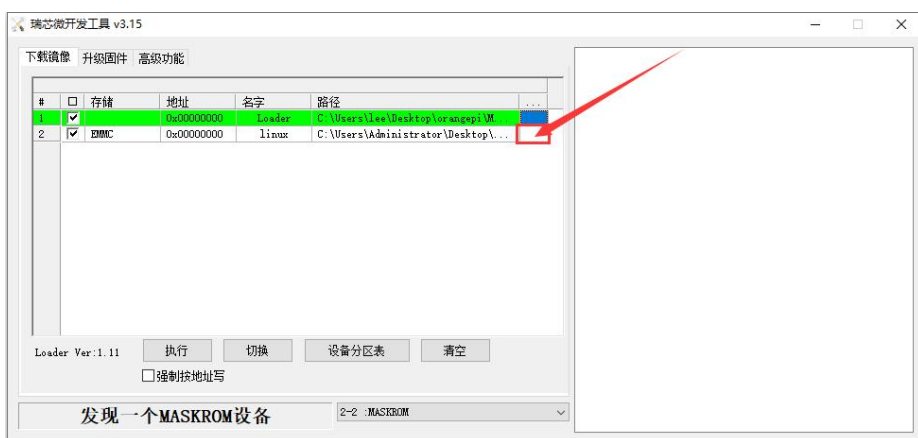
- l. Then click the position shown in the figure below



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



- n. Then click the position shown in the figure below

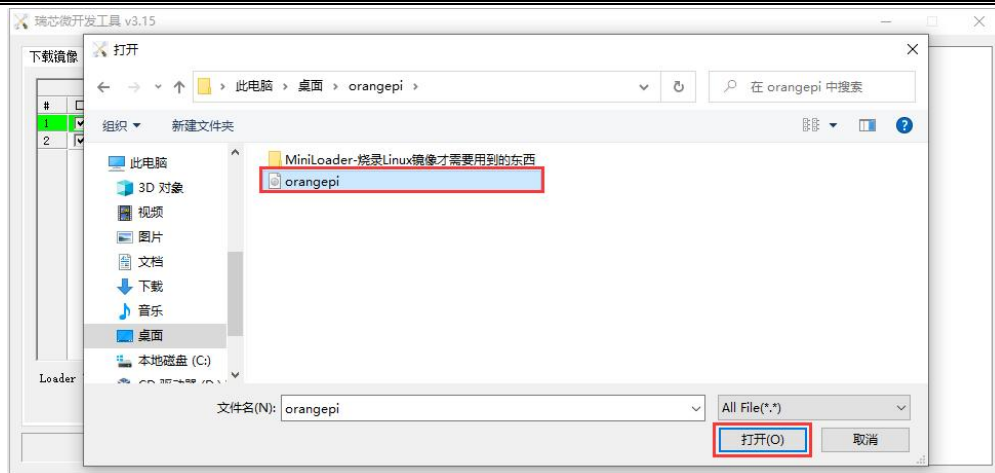


- o. Then select the path of the Linux image you want to burn, and then click **Open**

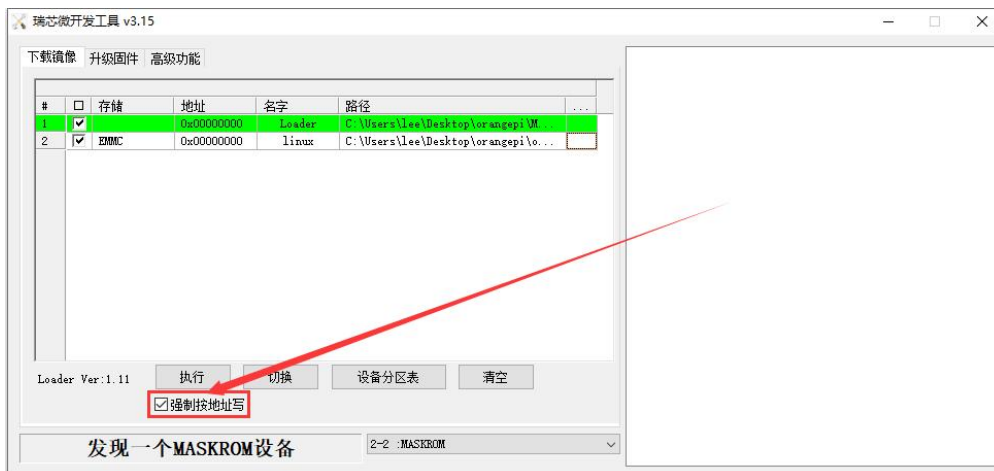
**Before burning the image, it is recommended to rename the Linux image to be**



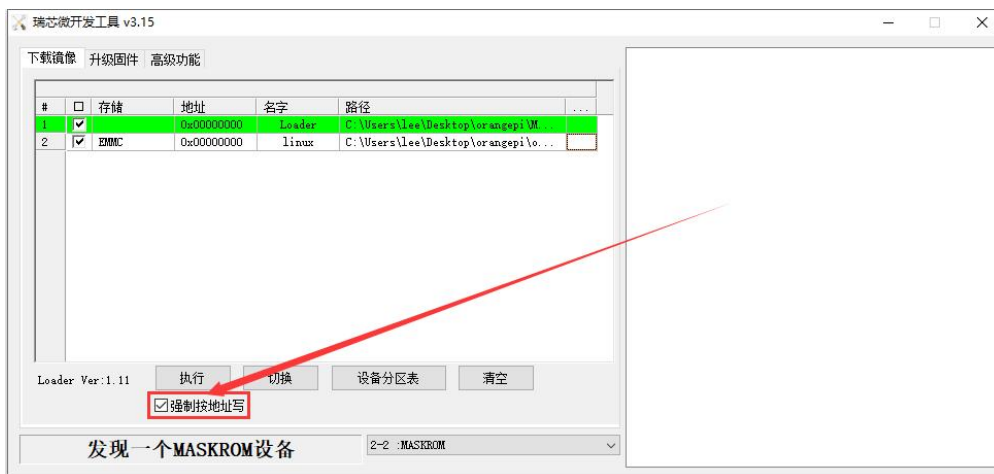
burned to orangepi.img or another shorter name, so that you can see the percentage of the burning progress when burning the image.



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

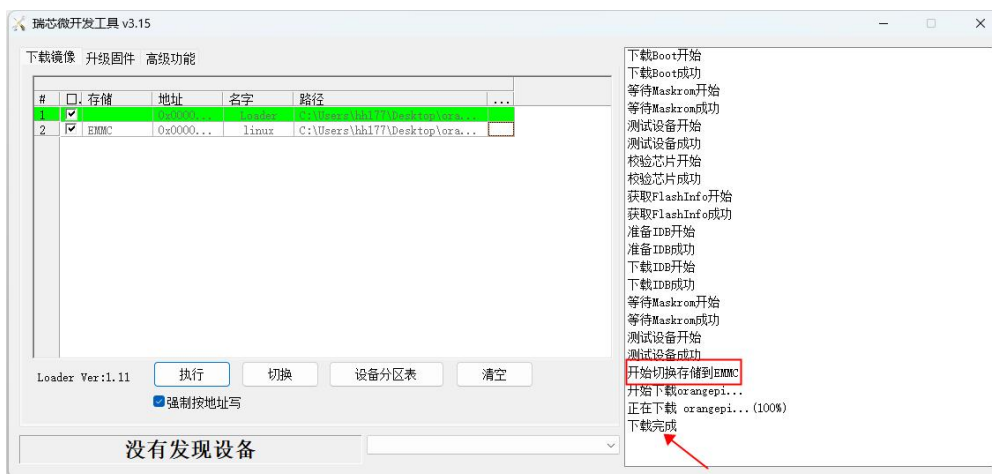


q. Click the Execute button again to start burning the linux image to the eMMC of the development board





r. After the r.linux image is burnt, the display log is shown in the figure below



s. After burning the linux image into the eMMC, the linux system will start automatically.

## 2.6.2. After burning the linux image into the eMMC, the linux system will start automatically.

**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) Using the dd command to burn the linux image to eMMC needs to be completed with the help of a TF card, so first you need to burn the linux image to the TF card, and then use the TF card to start the development board to enter the linux system. For the method of burning the Linux image to the TF card, please refer to the description of [the method of burning the Linux image to the TF card](#).

2) After using the TF card to start the linux system, we first upload the decompressed 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](#).

3) 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
Orangepi5b_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

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

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

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

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

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

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

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

- a. In the following command, the **if=** parameter is followed by the full path where the linux image is stored + the name of the Linux image (such as **the name of /home/orangepi/Desktop/Linux image**). Because we have entered the path of the linux image above, we only need to fill in the name of the Linux image.
- b. Please do not copy the linux image name in the following command, but replace it with the actual image name (because the version number of the image may be updated).

```
sudo dd bs=1M if=Orangepi5b_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/mmcblk1 status=progress
sudo sync
```





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

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

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

8) After starting the system in eMMC, use the **df -h** command to see the actual hard disk capacity

```
orange@orange:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           769M  9.5M  760M   2% /run
/dev/mmcblk1p2  29G   5.4G   23G  20% /
tmpfs           3.8G   0   3.8G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.8G  12K  3.8G   1% /tmp
/dev/mmcblk1p1  256M  116M  141M  46% /boot
/dev/zram1       188M  2.8M  171M   2% /var/log
tmpfs           769M   80K  769M   1% /run/user/1000
```

## 2.7. How to burn Android image into eMMC

### 2.7.1. How to burn Android image into eMMC via Type-C cable

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

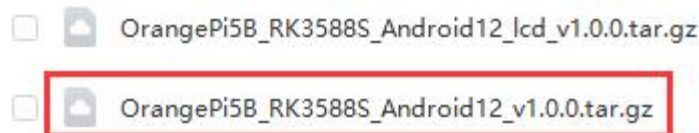
1) First, you need to prepare a data cable with a good quality Type-C interface



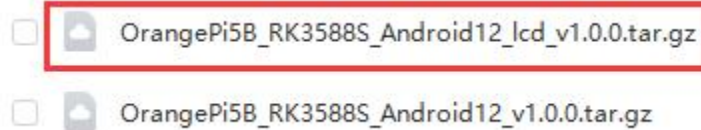
2) Then download the Rockchip driver **DriverAssitant\_v5.12.zip** and the burning tool **RKDevTool\_Release\_v3.15.zip** from the Orange Pi data download page

3) Then download the Android image from [Orange Pi's download page](#). After opening the download link of the Android image, you can see the following two types of Android images. The difference between them is:

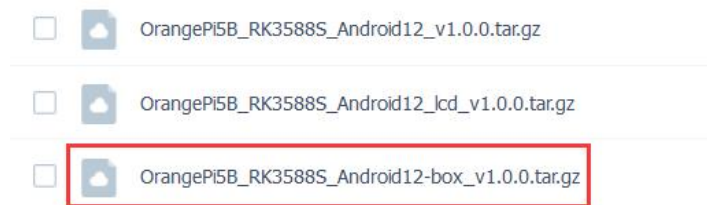
- a. The image without lcd is specially used for HDMI display, if you don't use LCD screen, please download the image without lcd



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



- c. The image with box is the image of TV box type



4) 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



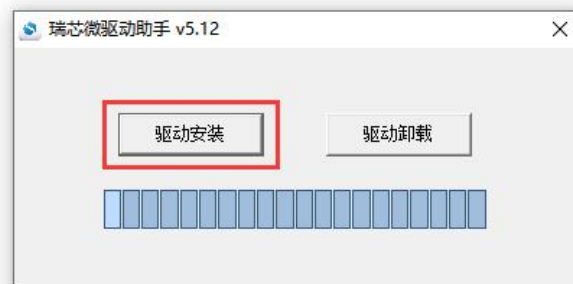
the Android image file, and the size is more than 1GB

5) Then use the decompression software to decompress **DriverAssitant\_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

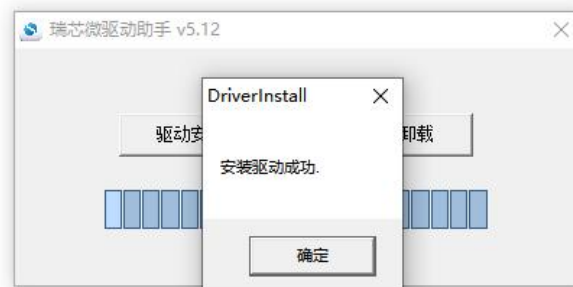
名称	修改日期	类型	大小
ADBDriver	2022/12/1 15:07	文件夹	
bin	2022/12/1 15:07	文件夹	
Driver	2022/12/1 15:07	文件夹	
config	2014/6/3 15:38	配置设置	1 KB
<b>DriverInstall</b>	2022/2/28 14:11	应用程序	491 KB
Readme	2018/1/31 17:44	文本文档	1 KB
revision	2022/2/28 14:14	文本文档	1 KB

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

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



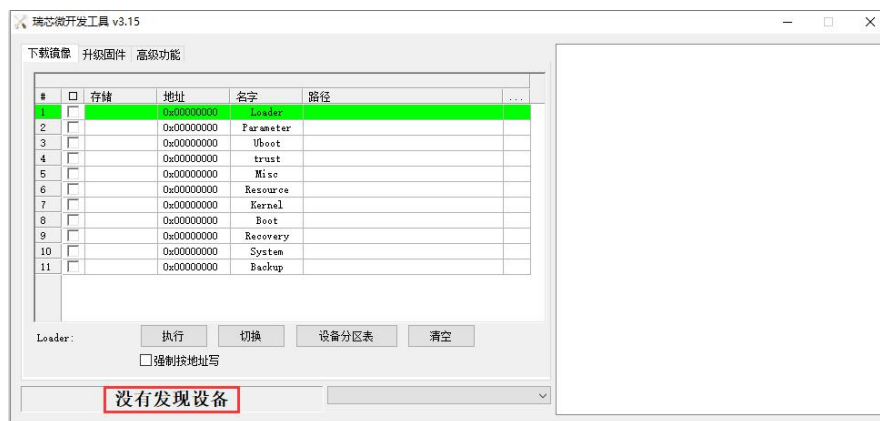
b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button.



7) Then decompress **RKDevTool\_Release\_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

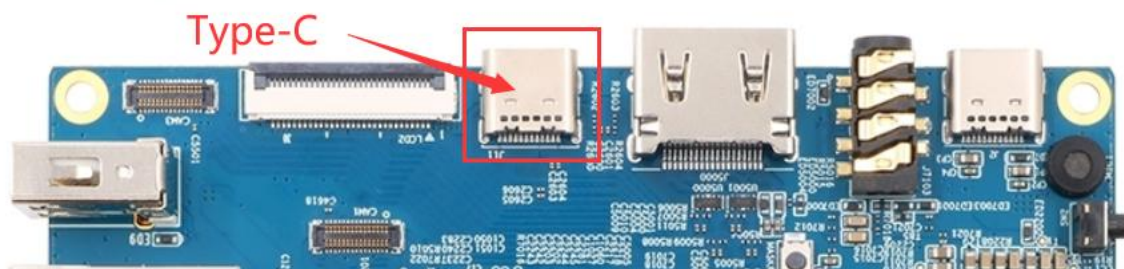
名称	修改日期	类型	大小
bin	2022/12/1 15:07	文件夹	
Language	2022/12/1 15:07	文件夹	
config.cfg	2022/3/23 9:11	CFG 文件	7 KB
config	2021/11/30 11:04	配置设置	2 KB
revision	2022/5/27 9:09	文本文档	3 KB
<b>RKDevTool</b>	2022/5/27 9:06	应用程序	1,212 KB
开发工具使用文档_v1.0	2021/8/27 10:28	Foxit PDF Reade...	450 KB

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

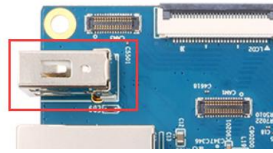


9) Then start burning the Android image into eMMC

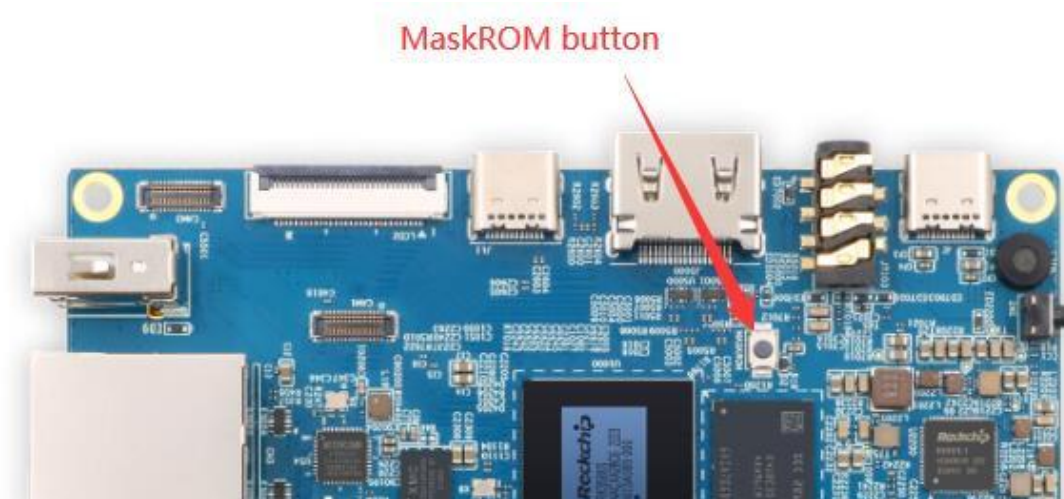
- a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below



- 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



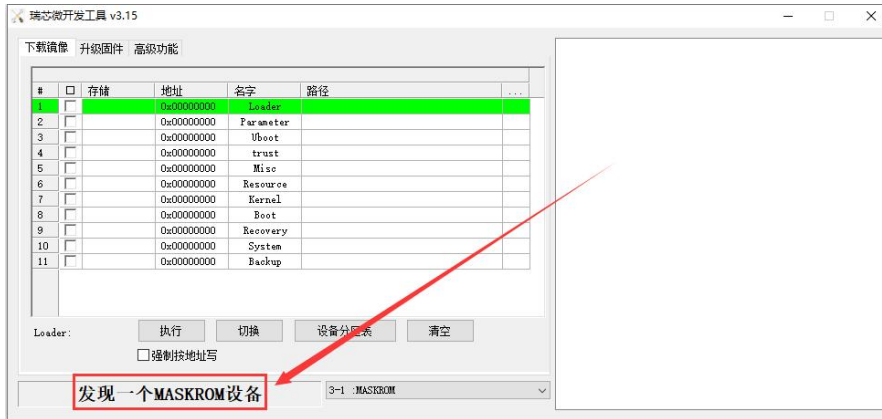
- d. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is shown in the figure below:



- 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

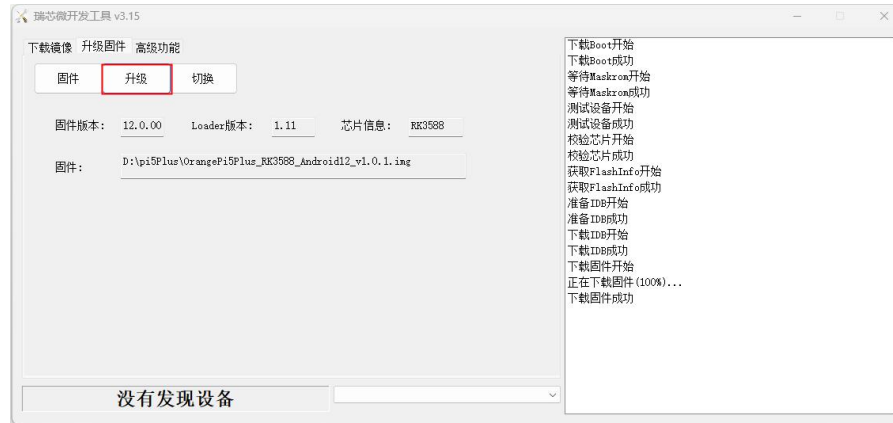


h. Then click the "**Firmware**" button to select the path of the Android image that needs to be burned



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





### 2.7.2. How to burn Android 12 image into eMMC via TF card

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

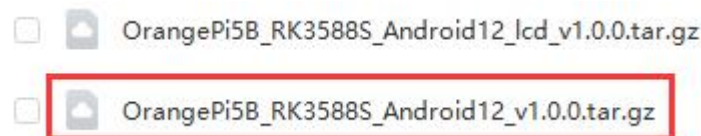
1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card of SanDisk and other brands

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

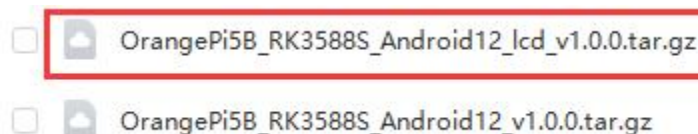
3) Then download the SDDiskTool programming tool from [the Orange Pi data download page](#), **please ensure that the version of the SDDiskTool tool is the latest v1.72**

4) Then download the Android image from [Orange Pi's download page](#). After opening the download link of the Android image, you can see the following two types of Android images. The difference between them is:

- a. The image without lcd is specially used for HDMI display, if you don't use LCD screen, please download the image without lcd



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





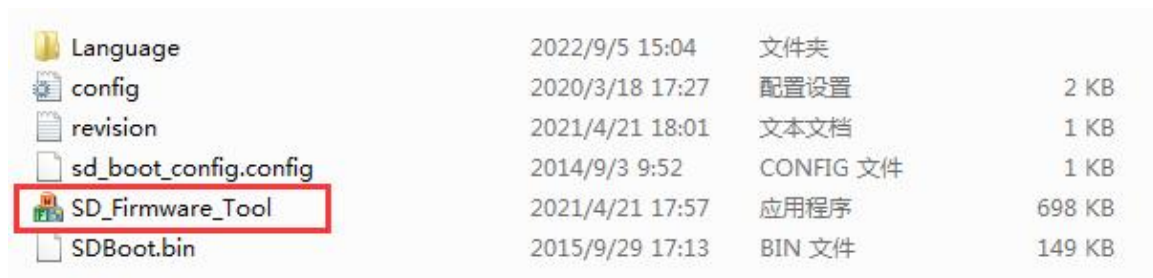


c. The image with box is the image of TV box type



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



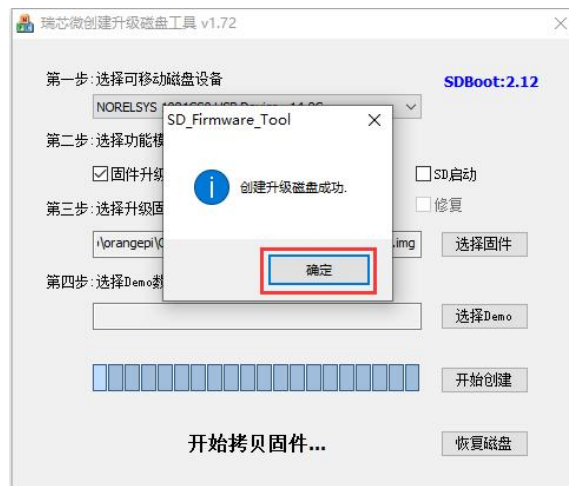
8) After confirming the drive letter, you can format the TF card first, click **the restore disk** button in SDDiskTool, or use the **SD Card Formatter** mentioned above to format the TF card



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

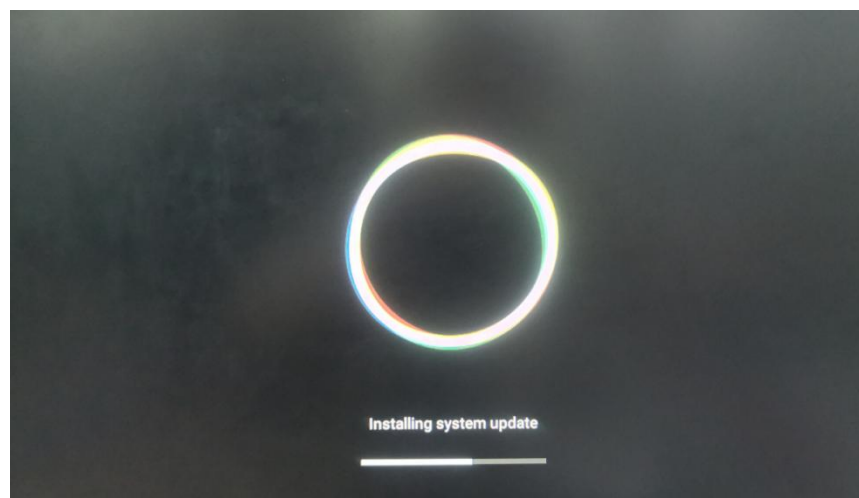


10) After the burning is completed, the display is as shown in the figure below, and then you can exit SDDiskTool



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

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



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



```

vbmeta writing...
RKA_File_Download entry.name=vbmeta
RKA_File_Download entry.name=vbmeta DONE!
boot writing...
RKA_File_Download entry.name=boot
RKA_File_Download entry.name=boot DONE!
recovery writing...
RKA_File_Download entry.name=recovery
RKA_File_Download entry.name=recovery DONE!
baseparameter writing...
RKA_File_Download entry.name=baseparameter
RKA_File_Download entry.name=baseparameter DONE!
super writing...
RKA_SparseFile_Download entry.name=super
INFO:Start to download super.offset=0x1da000,size=3263168512
INFO:ErasePartition super.offset=0x1da000,size=3263168512, part_size=0x614000
INFO:RKA_SparseFile_Download-->total_chunks=3889
RKA_SparseFile_Download entry.name=super DONE!
parameter checking...
uboot checking...
RKA_File_Check entry.name=uboot
RKA_File_Check entry.name=uboot DONE!
misc checking...
RKA_File_Check entry.name=misc
RKA_File_Check entry.name=misc DONE!
dtbo checking...
RKA_File_Check entry.name=dtbo
RKA_File_Check entry.name=dtbo DONE!
vbmeta checking...
RKA_File_Check entry.name=vbmeta
RKA_File_Check entry.name=vbmeta DONE!
boot checking...
RKA_File_Check entry.name=boot
RKA_File_Check entry.name=boot DONE!
recovery checking...
RKA_File_Check entry.name=recovery
RKA_File_Check entry.name=recovery DONE!
baseparameter checking...
RKA_File_Check entry.name=baseparameter
RKA_File_Check entry.name=baseparameter DONE!
super checking...
RKA_SparseFile_Check entry.name=super
INFO:Start to check super.offset=0x1da000,size=164u
RKA_SparseFile_Check entry.name=super Done!
Finish to upgrade firmware.
SD upgrade ok.
prksdboot->do_rk_mode_update Successful!
Doing Actions succeeded.please remove the sdcard.....

```

## 2.8. How to burn Orange Pi OS (Droid) image into eMMC

### 2.8.1. Burn Orange Pi OS (Droid) image to eMMC via Type-C cable

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

- 1) First, you need to prepare a data cable with a good quality Type-C interface





2) Then download the Rockchip driver **DriverAssitant\_v5.12.zip** and the burning tool **RKDevTool\_Release\_v3.15.zip** from [the Orange Pi data download page](#)

3) Then download the Orange Pi OS (Droid) image from [the Orange Pi download page](#)



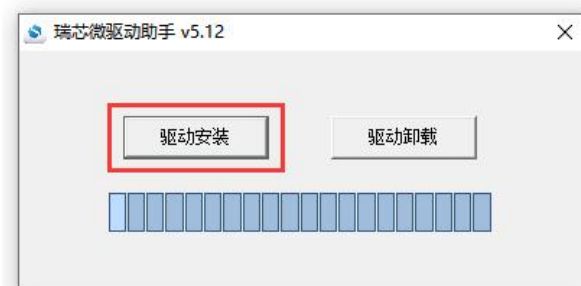
4) Then use the decompression software to decompress the compressed package of the downloaded Orange Pi OS (Droid) image. Among the decompressed files, the file ending with ".img" is the Orange Pi OS (Droid) image file, and the size is more than 1GB

5) Then use the decompression software to decompress **DriverAssitant\_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the decompressed folder and open it

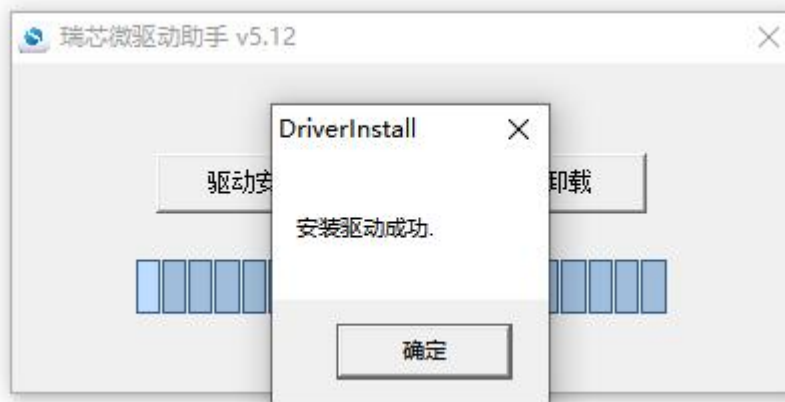
名称	修改日期	类型	大小
ADBDriver	2022/12/1 15:07	文件夹	
bin	2022/12/1 15:07	文件夹	
Driver	2022/12/1 15:07	文件夹	
config	2014/6/3 15:38	配置设置	1 KB
<b>DriverInstall</b>	2022/2/28 14:11	应用程序	491 KB
Readme	2018/1/31 17:44	文本文档	1 KB
revison	2022/2/28 14:14	文本文档	1 KB

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

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



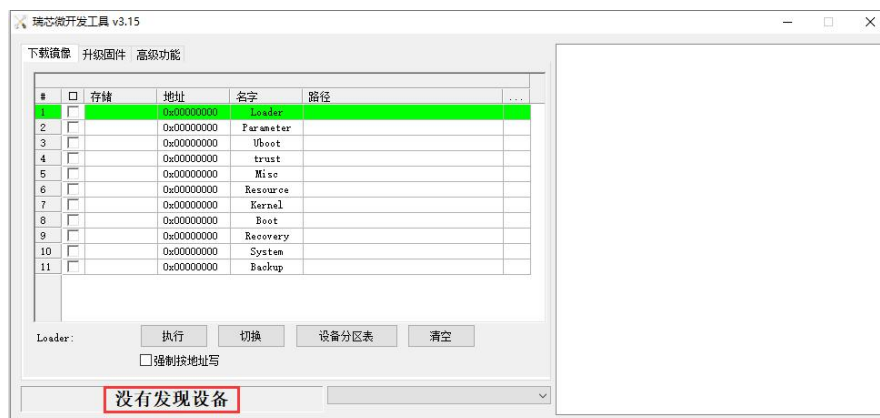
b. After waiting for a period of time, a pop-up window will prompt "**The driver is installed successfully**", and then click the "**OK**" button.



7) Then decompress **RKDevTool\_Release\_v3.15.zip**, this software does not need to be installed, just find **RKDevTool** in the decompressed folder and open it

名称	修改日期	类型	大小
bin	2022/12/1 15:07	文件夹	
Language	2022/12/1 15:07	文件夹	
config.cfg	2022/3/23 9:11	CFG 文件	7 KB
config	2021/11/30 11:04	配置设置	2 KB
revision	2022/5/27 9:09	文本文档	3 KB
<b>RKDevTool</b>	2022/5/27 9:06	应用程序	1,212 KB
开发工具使用文档_v1.0	2021/8/27 10:28	Foxit PDF Reade...	450 KB

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

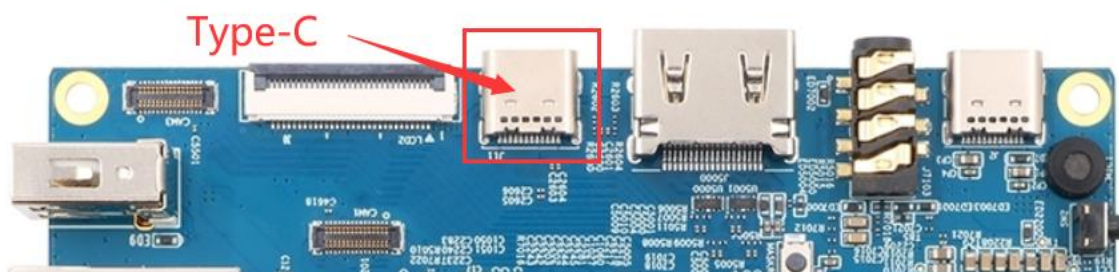


9) Then start burning the Orange Pi OS (Droid) image into eMMC

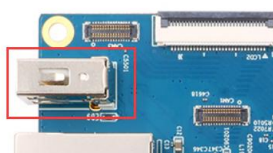




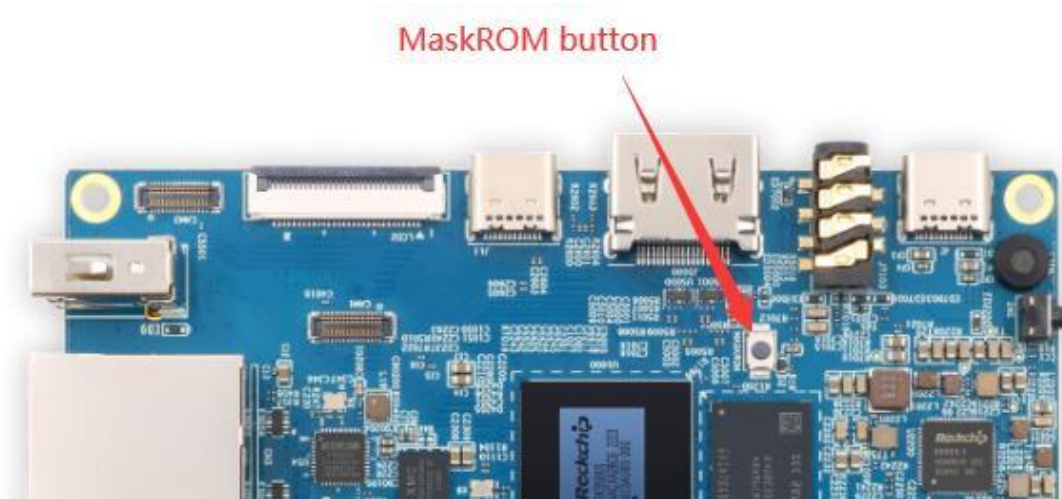
- a. First, connect the development board to the Windows computer through the Type-C data cable. The position of the Type-C interface on the development board is shown in the figure below



- 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:

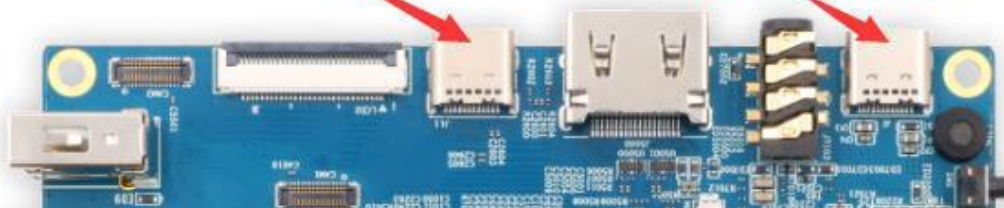


- e. Then connect the power supply of the Type-C interface to the development board and power on

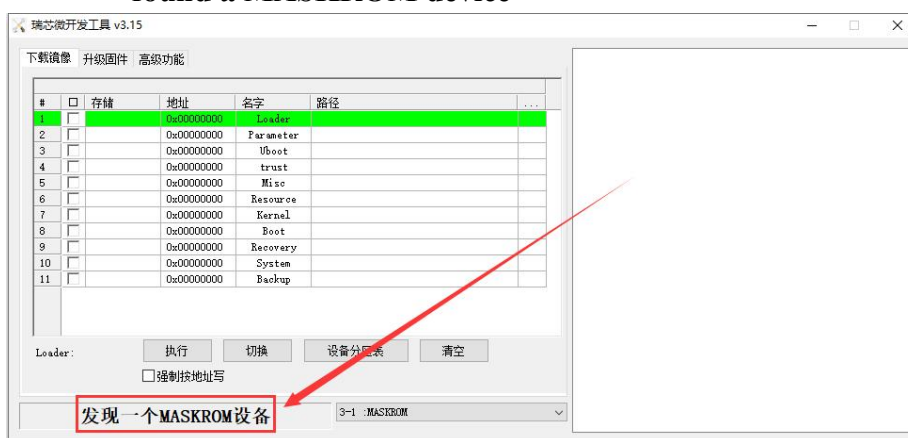


This interface has no power supply function

Type-C Power port



- 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 path of the Orange Pi OS (Droid) image that needs to be burned



- i. Finally, click the "Upgrade" button to start burning, and the log during the burning process is shown in the figure below. After burning, the Orange Pi OS (Droid) system will start automatically.



### 2. 8. 2. Burn Orange Pi OS (Droid) image to eMMC via TF card

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

- 1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use 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](#)



Opios-droid-aarch64-opi5b-23.04-linux5.10.110.tar.gz

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

Language	2022/9/5 15:04	文件夹	
config	2020/3/18 17:27	配置设置	2 KB
revision	2021/4/21 18:01	文本文档	1 KB
sd_boot_config.config	2014/9/3 9:52	CONFIG 文件	1 KB
<b>SD_Firmware_Tool</b>	2021/4/21 17:57	应用程序	698 KB
SDBoot.bin	2015/9/29 17:13	BIN 文件	149 KB

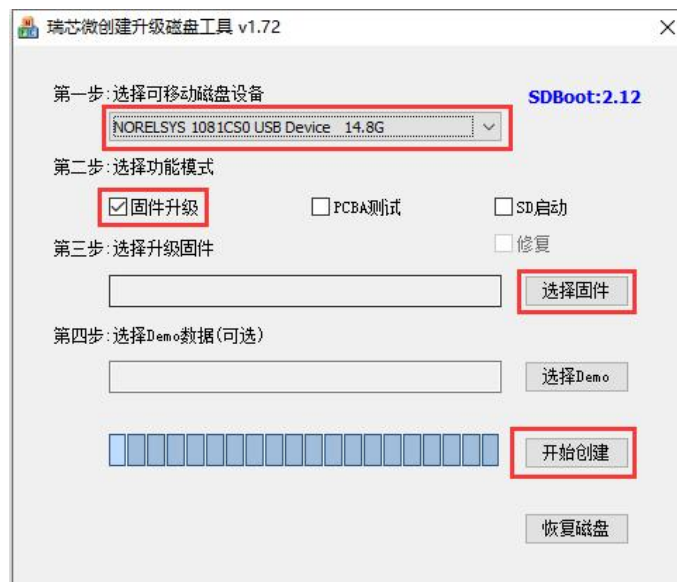
7) After opening **SDDiskTool**, if the TF card is recognized normally, the inserted disk device will be displayed in the "Select Removable Disk Device" column. **Please make sure that the displayed disk device is consistent with the drive letter of the TF card you want to burn**, if there is no display,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



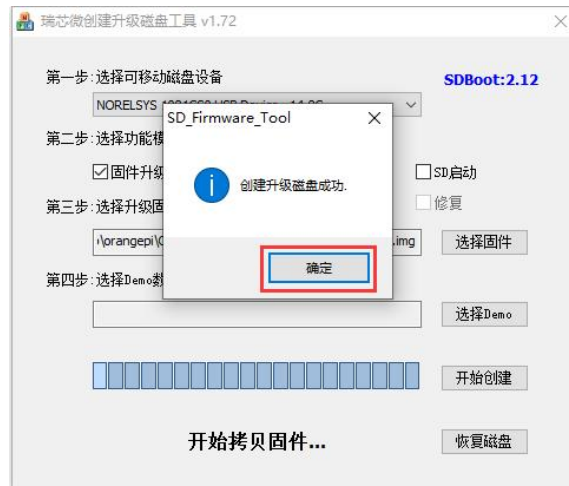
8) After confirming the drive letter, you can format the TF card first, click the **restore disk** button in **SDDiskTool**, or use the **SD Card Formatter** mentioned above to format the TF card



- 9) Then start to write the Orange Pi OS (Droid) image to the TF card
- First confirm that the displayed drive letter is the drive letter corresponding to the TF card under "**Select Removable Disk Device**"
  - Then select "**Firmware Upgrade**" in "**Select Function Mode**"
  - c. Then select the path of the Orange Pi OS (Droid) firmware in the "**Select to upgrade firmware**" column
  - d. Finally, click the "**Start Create**" button to start burning

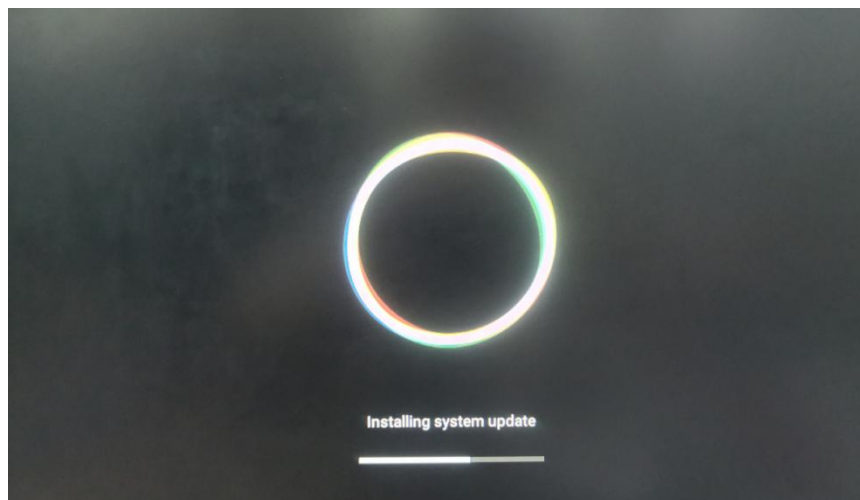


- 10) After the burning is completed, the display is as shown in the figure below, and then you can exit SDDiskTool



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

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



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



```

vbmeta writing...
RKA_File_Download entry.name=vbmeta
RKA_File_Download entry.name=vbmeta DONE!
boot writing...
RKA_File_Download entry.name=boot
RKA_File_Download entry.name=boot DONE!
recovery writing...
RKA_File_Download entry.name=recovery
RKA_File_Download entry.name=recovery DONE!
baseparameter writing...
RKA_File_Download entry.name=baseparameter
RKA_File_Download entry.name=baseparameter DONE!
super writing...
RKA_SparseFile_Download entry.name=super
INFO:Start to download super.offset=0x1da000,size=3263168512
INFO:ErasePartition super.offset=0x1da000,size=3263168512, part_size=0x614000
INFO:RKA_SparseFile_Download-->total_chunks=3889
RKA_SparseFile_Download entry.name=super DONE!
parameter checking...
uboot checking...
RKA_File_Check entry.name=uboot
RKA_File_Check entry.name=uboot DONE!
misc checking...
RKA_File_Check entry.name=misc
RKA_File_Check entry.name=misc DONE!
dtbo checking...
RKA_File_Check entry.name=dtbo
RKA_File_Check entry.name=dtbo DONE!
vbmeta checking...
RKA_File_Check entry.name=vbmeta
RKA_File_Check entry.name=vbmeta DONE!
boot checking...
RKA_File_Check entry.name=boot
RKA_File_Check entry.name=boot DONE!
recovery checking...
RKA_File_Check entry.name=recovery
RKA_File_Check entry.name=recovery DONE!
baseparameter checking...
RKA_File_Check entry.name=baseparameter
RKA_File_Check entry.name=baseparameter DONE!
super checking...
RKA_SparseFile_Check entry.name=super
INFO:Start to check super.offset=0x1da000,size=164u
RKA_SparseFile_Check entry.name=super DONE!
Finish to upgrade firmware.
SD upgrade ok.
prksdboot->do_rk_mode_update Successful!
Doing Actions succeeded.please remove the sdcard.....

```

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

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

- 1) First prepare a TF card with 8GB or larger capacity. The transmission speed of the TF card must be class10 or above. It is recommended to use TF cards from SanDisk and other brands.
- 2) Then use the card reader to insert the TF card into the computer
- 3) Then download the SDDiskTool burning tool from the [Orange Pi download page](#).  
**Please ensure that the version of the SDDiskTool tool is the latest v1.72**
- 4) Then download the image of Orange Pi OS (OH) from the [Orange Pi download page](#)



Opios-oh-4.0-release-aarch64-opi5b-24.1-linux5.10.tar.gz.gz





5) Then use decompression software to decompress the compressed package of the downloaded Orange Pi OS (OH) image. In the decompressed file, the file ending with ".img" is the Orange Pi OS (OH) image file, which is about 3GB in size.

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.

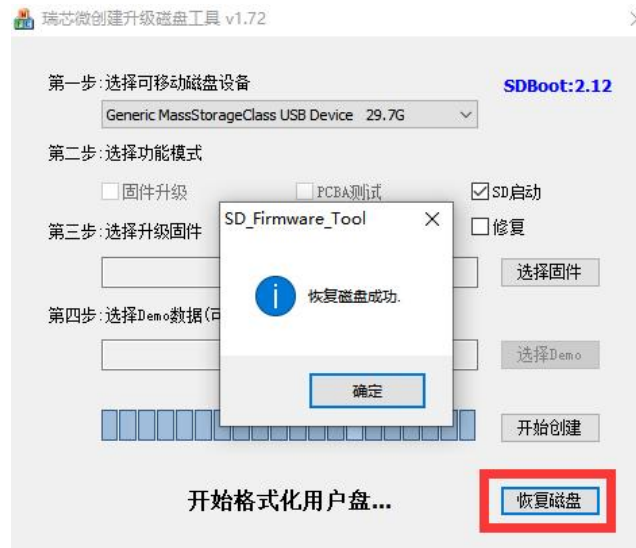
Language	2022/9/5 15:04	文件夹	
config	2020/3/18 17:27	配置设置	2 KB
revision	2021/4/21 18:01	文本文档	1 KB
sd_boot_config.config	2014/9/3 9:52	CONFIG 文件	1 KB
<b>SD_Firmware_Tool</b>	2021/4/21 17:57	应用程序	698 KB
SDBoot.bin	2015/9/29 17:13	BIN 文件	149 KB

7) 打 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, try unplugging the TF card. 开 SDDiskTool

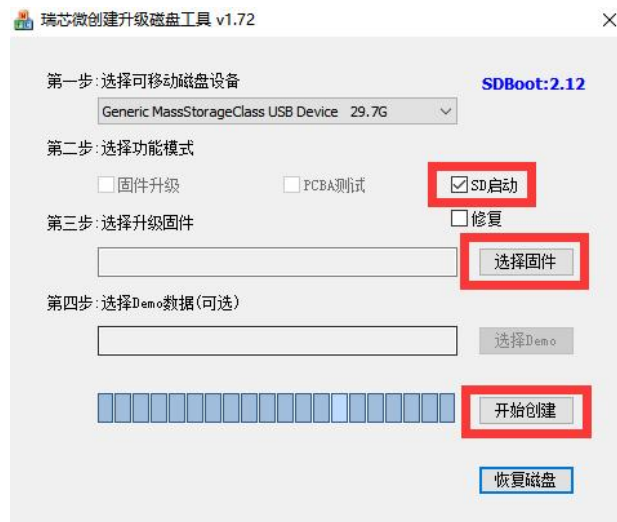


8) After confirming the drive letter, you can format the TF card first and click the **Recover Disk** button in **SDDiskTool**. You can also use the **SD Card Formatter** mentioned earlier to format the TF card.





- 9) Then start writing the Orange Pi OS (OH) image to the TF card
- First check "SD boot" in "Select function mode"
  - Then select the path to the Orange Pi OS (OH) image in the "Select Upgrade Firmware" column
  - Finally, click the "Start Creating" button to start burning the Orange Pi OS (OH) 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. 10. How to burn Orange Pi OS (OH) image into eMMC

### 2. 10. 1. Burn Orange Pi OS (OH) image to eMMC via Type-C cable

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

- 1) First you need to prepare a good quality Type-C interface data cable



- 2) Then download the Rockchip microdriver **DriverAssitant\_v5.12.zip** and the burning tool **RKDevTool\_Release\_v3.15.zip** from the [Orange Pi data download page](#)
- 3) Then download the image of Orange Pi OS (OH) from the [Orange Pi download page](#)



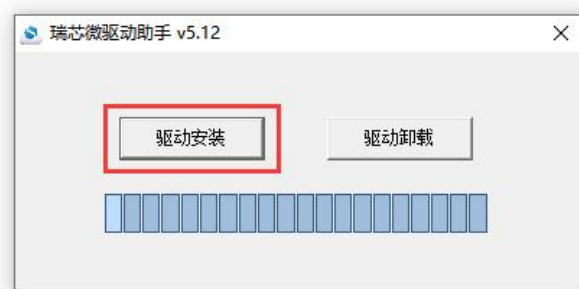
4) Then use decompression software to decompress the compressed package of the downloaded Orange Pi OS (OH) image. In the decompressed file, the file ending with ".img" is the Orange Pi OS (OH) image file, which is about 3GB in size.

5) Then use decompression software to decompress **DriverAssitant\_v5.12.zip**, then find the **DriverInstall.exe** executable file in the decompressed folder and open it.

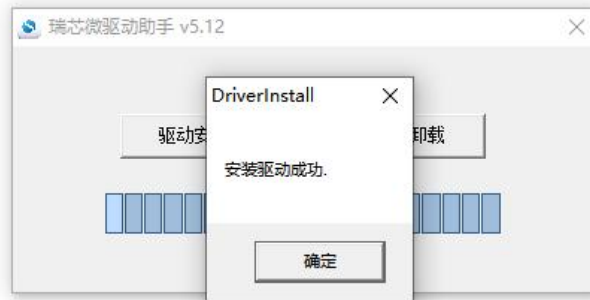
名称	修改日期	类型	大小
ADBDriver	2022/12/1 15:07	文件夹	
bin	2022/12/1 15:07	文件夹	
Driver	2022/12/1 15:07	文件夹	
config	2014/6/3 15:38	配置设置	1 KB
<b>DriverInstall</b>	2022/2/28 14:11	应用程序	491 KB
Readme	2018/1/31 17:44	文本文档	1 KB
revison	2022/2/28 14:14	文本文档	1 KB

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

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



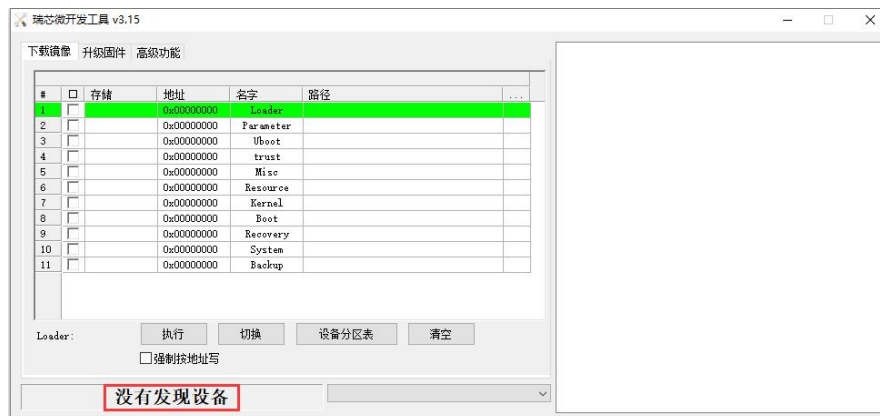
b. After waiting for a period of time, a window will pop up prompting "**Driver installation successful**", then click the "**OK**" button.



7) Then unzip **RKDevTool\_Release\_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

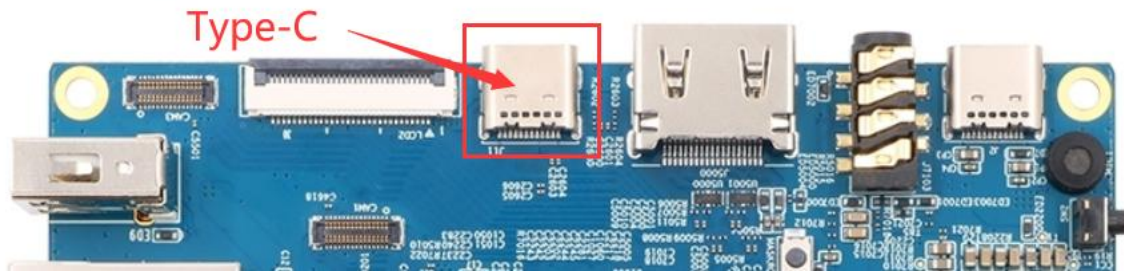
名称	修改日期	类型	大小
bin	2022/12/1 15:07	文件夹	
Language	2022/12/1 15:07	文件夹	
config.cfg	2022/3/23 9:11	CFG 文件	7 KB
config	2021/11/30 11:04	配置设置	2 KB
revision	2022/5/27 9:09	文本文档	3 KB
<b>RKDevTool</b>	2022/5/27 9:06	应用程序	1,212 KB
开发工具使用文档_v1.0	2021/8/27 10:28	Foxit PDF Reade...	450 KB

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

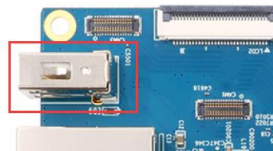


9) Then start burning the Orange Pi OS (OH) image into the eMMC

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



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



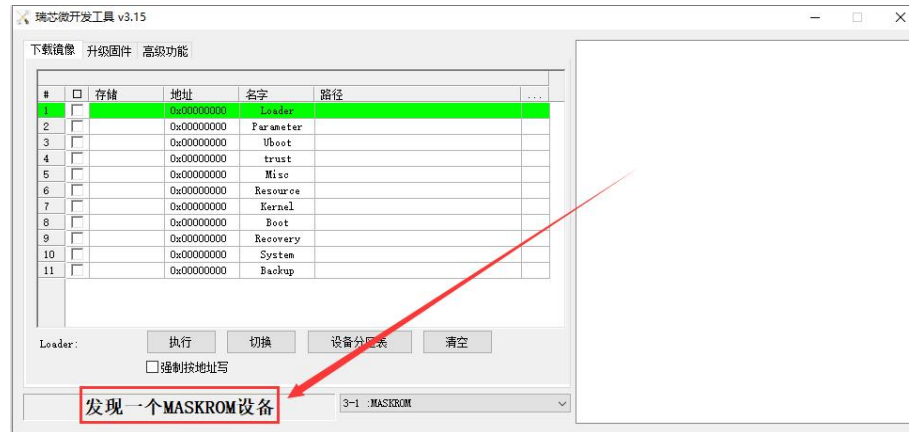
- c. Then press and hold the MaskROM button on the development board. The position of the MaskROM button on the development board is as shown in the figure below:



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



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



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

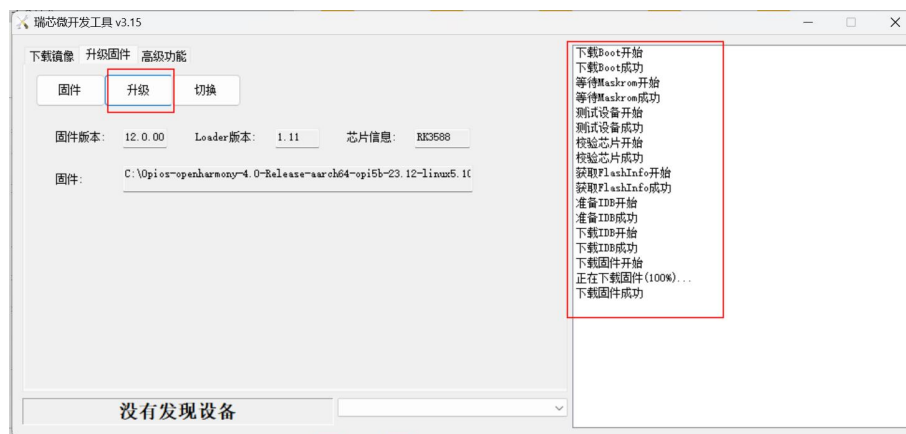


- g. Then click the "Firmware" button to select the path to the Orange Pi OS (OH) image that needs to be burned.



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





## 2. 11. Start the Orange Pi development board

- 1) The development board has an on-board eMMC, and the Orange Pi OS (Droid) image is burned by default. You can directly use the image in the eMMC to start and fully function test after you get the development board.
- 2) If you need to use the linux image, you can insert the TF card with the linux image burned into the TF card slot of the Orange Pi development board.
- 3) 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.
- 4) Connect a USB mouse and keyboard to control the Orange Pi development board.
- 5) The development board has an Ethernet port, which can be plugged into a network cable for Internet access.
- 6) 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.**

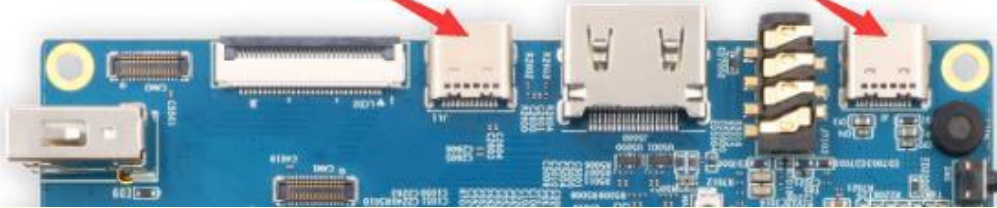
**The Type-C power port does not support PD negotiation.**

**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.**

This interface has no power supply function

Type-C Power port



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

8) If you want to view the output information of the system through the debugging serial port, please use the serial 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. 12. How to use the debugging serial port

### 2. 12. 1. Connection instruction of debugging serial port

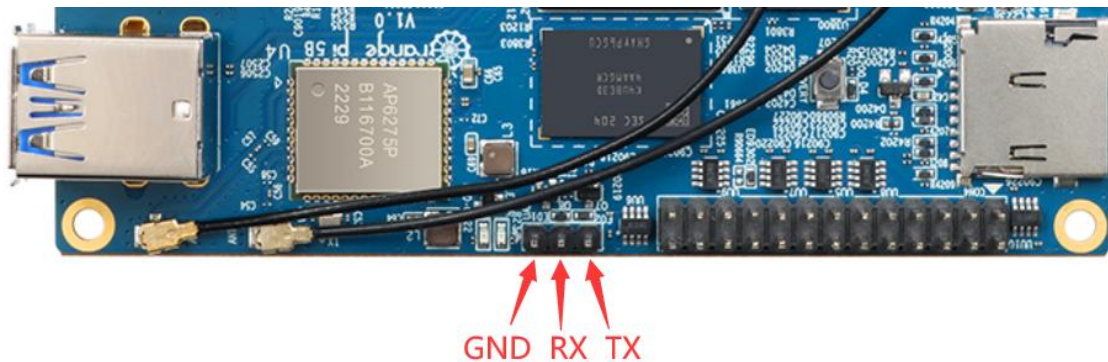
1) First, you need to prepare a 3.3V USB to TTL module, and then insert the USB interface end of the USB to TTL module into the USB interface of the computer.

**For better compatibility, it is recommended to use the CH340 USB to TTL module instead of the CP2102 USB to TTL module.**

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



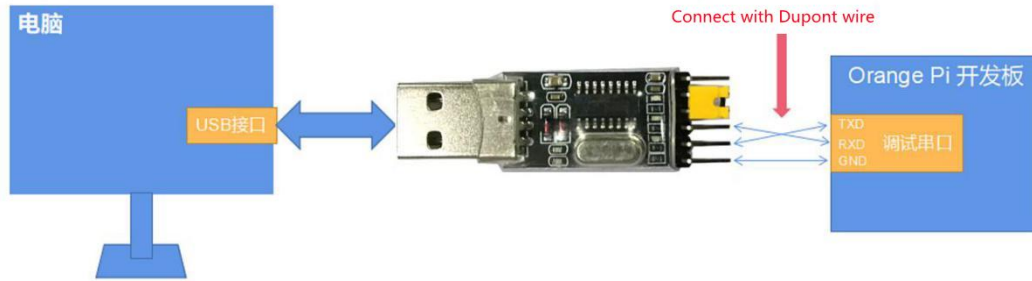
2) The corresponding relationship between GND, RXD and TXD pins of the debugging serial port of the development board is shown in the figure below



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

- a. 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. 12. 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
```

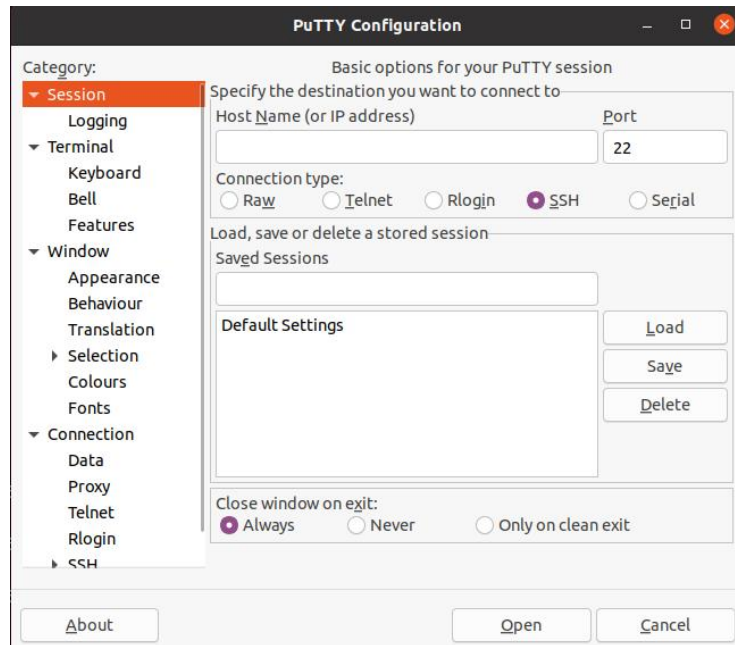
2) Then use the following command to install putty on Ubuntu PC

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

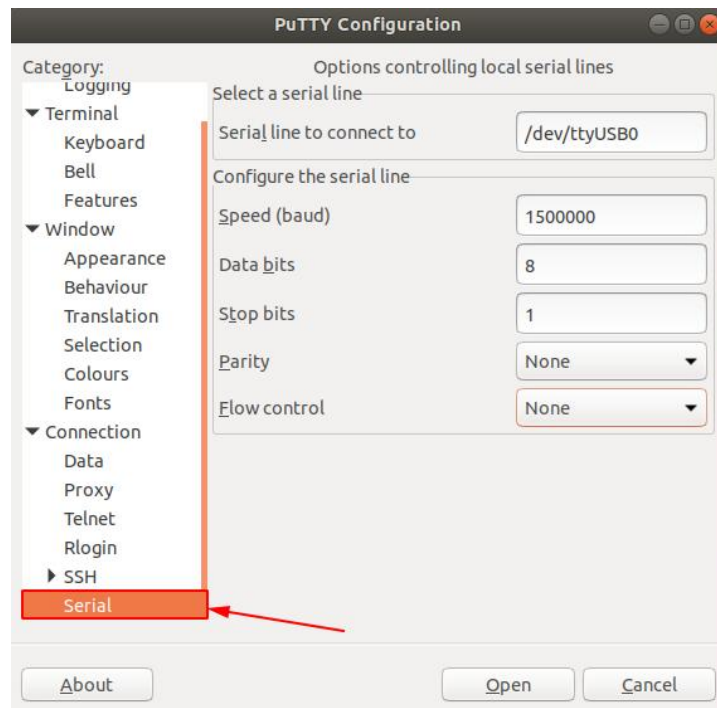
3) Then run putty, **remember to add sudo permission**

```
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

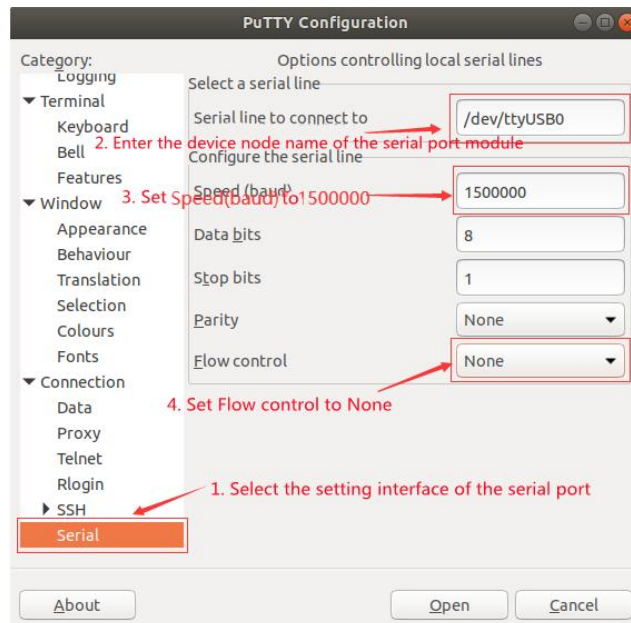


6) Then set the parameters of the serial port

- 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)

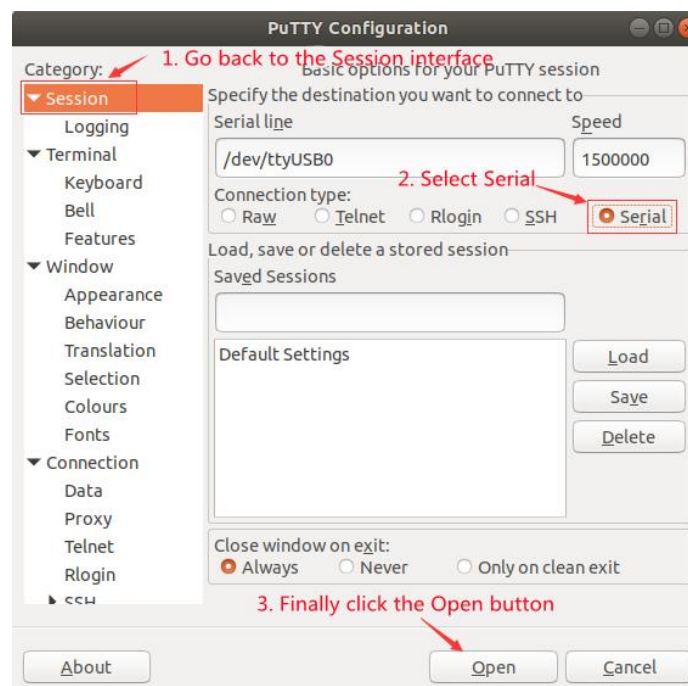


c. Set Flow control to None



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

- First select the Connection type as Serial
- 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

```
/dev/ttyUSB0 - PuTTY
R0=0x18
MR4=0x1
MR5=0x1
MR8=0x8
MR12=0x72
MR14=0x72
MR18=0x0
MR19=0x0
MR24=0x8
MR25=0x0
R0=0x18
MR4=0x1
MR5=0x1
MR8=0x8
MR12=0x72
MR14=0x72
MR18=0x0
MR19=0x0
MR24=0x8
MR25=0x0
channel 0 training pass!
channel 1 training pass!
change freq to 416MHz 0,1
Channel 0: LPDDR4,416MHz
Bus Width=32 Col=10 Bank=8 Row=15/15 CS=2 Die Bus-Width=16 Size=2048MB
Channel 1: LPDDR4,416MHz
Bus Width=32 Col=10 Bank=8 Row=15/15 CS=2 Die Bus-Width=16 Size=2048MB
256B stride
R0=0x18
```

### 2. 12. 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.

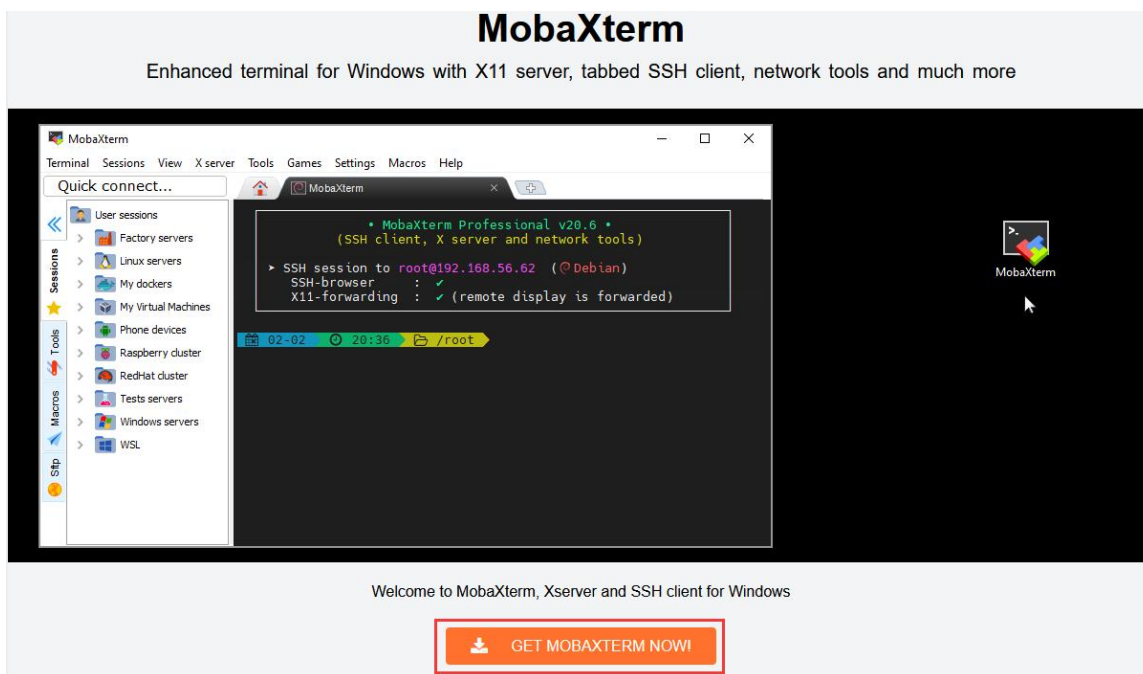
#### 1) Download MobaXterm

- 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

Home Edition	Professional Edition
<b>Free</b>	<b>\$69 / 49€ per user*</b>
<small>Full X server and SSH support Remote desktop (RDP, VNC, Xdmcp) Remote terminal (SSH, telnet, rlogin, Mosh) X11-Forwarding Automatic SFTP browser Master password protection Plugins support Portable and installer versions Full documentation Max. 12 sessions Max. 2 SSH tunnels Max. 4 macros Max. 360 seconds for Tftp, Nfs and Cron</small>	<small>* Excluding tax. Volume discounts <a href="#">available</a></small> <b>Every feature from Home Edition +</b> Customize your startup message and logo Modify your profile script Remove unwanted games, screensaver or tools Unlimited number of sessions Unlimited number of tunnels and macros Unlimited run time for network daemons Enhanced security settings 12-months updates included Deployment inside company Lifetime right to use
<a href="#">Download now</a>	<a href="#">Subscribe online / Get a quote</a>


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






MobaXterm Home Edition

Download MobaXterm Home Edition (current version):


 MobaXterm Home Edition v22.2  
(Portable edition)

 MobaXterm Home Edition v22.2  
(Installer edition)


Download previous stable version: [MobaXterm Portable v22.1](#) [MobaXterm Installer v22.1](#)

By downloading MobaXterm software, you accept [MobaXterm terms and conditions](#)

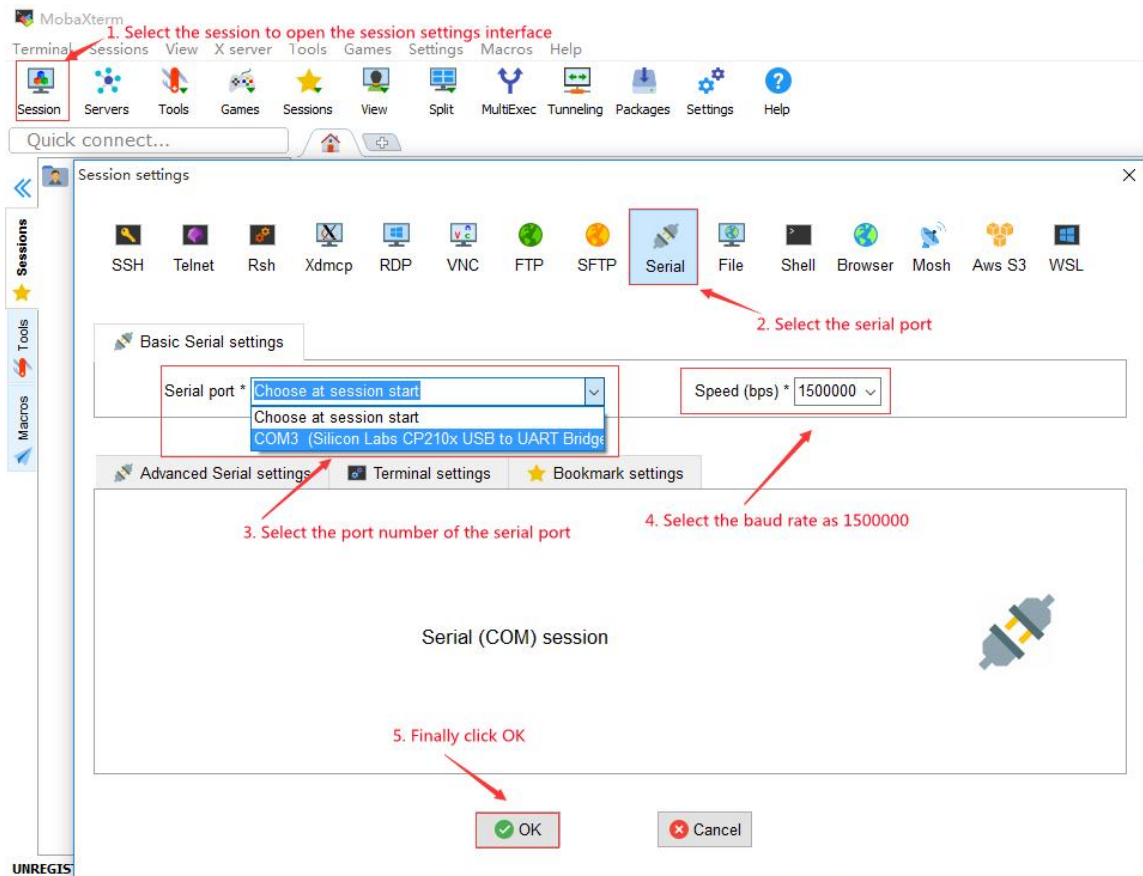
You can download the third party plugins and components sources [here](#)

 If you use MobaXterm inside your company, you should consider subscribing to [MobaXterm Professional Edition](#): your subscription will give you access to professional support and to the "Customizer" software. This customizer will allow you to generate personalized versions of MobaXterm including your own logo, your default settings and your welcome message. Please [contact us](#) for more information.

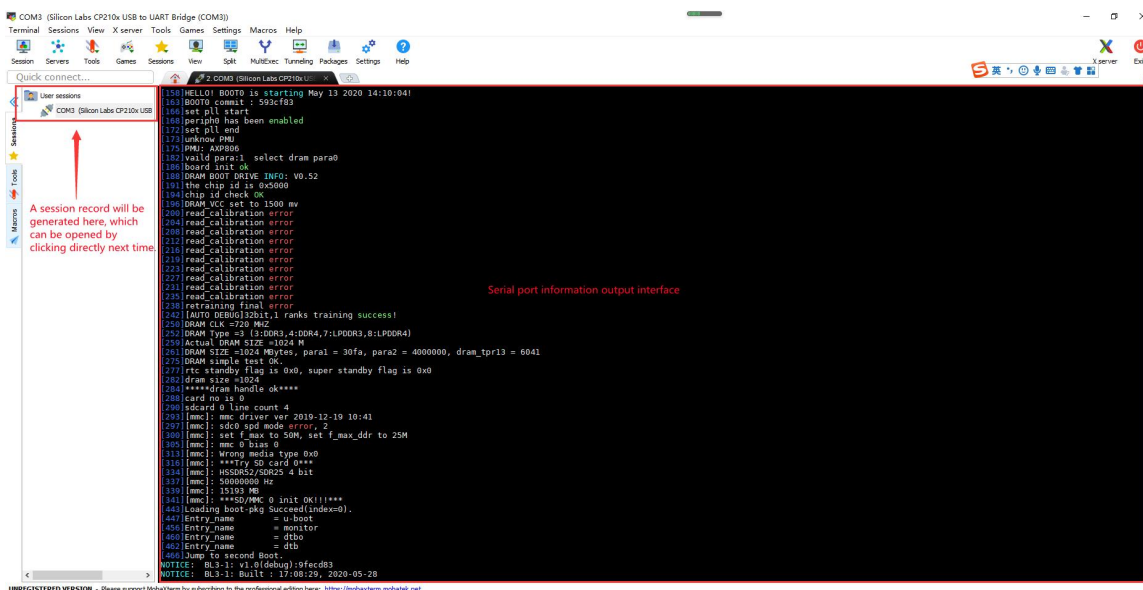
2) After downloading, use decompression software to decompress the downloaded compressed package, you can get the executable software of MobaXterm, and then double-click to open

名称	修改日期	类型	大小
 CygUtils.plugin	2022/9/24 20:16	PLUGIN 文件	17,484 KB
 MobaXterm_Personal_22.2	2022/10/22 16:53	应用程序	16,461 KB

- 3) After opening the software, the steps to set up the serial port connection are as follows
- Select the serial port type
  - Select the port number of the serial port (select the corresponding port number according to the actual situation), if you can't see the port number, please use **360 Driver Master** to scan and install the driver of the USB to TTL serial port chip
  - Select the baud rate of the serial port as **1500000**
  - 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. 13. 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 cord and power adapter used can meet the power supply requirements of the development board. If the use is unstable, please switch back to the Type-C power supply.

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



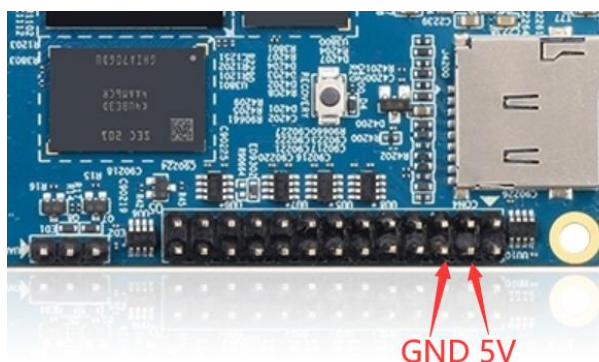
The power cord shown in the picture above can be bought on Taobao, please search and buy by yourself.

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. b. The red DuPont line needs to be plugged into the 5V pin of the development board 26pin
- c. 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 26pin interface on the development board is shown in the figure below, **remember not to reverse the connection**



### 3. Ubuntu/Debian Server and Xfce desktop system usage instructions

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

If you are using the Ubuntu22.04 Gnome image, please first check the instructions in the [Ubuntu22.04 Gnome Wayland Desktop System Instructions](#) chapter.

[Ubuntu22.04 Gnome Wayland Desktop System Instructions](#) — For content that does not exist in the [Ubuntu22.04 Gnome Wayland Desktop System Instructions](#) chapter, you can refer to the instructions in this chapter, but some details will be different, so please pay special attention to this.

If you are using the OPi OS Arch image, please check the [Orange Pi OS Arch System Instructions](#) chapter.

### 3. 1. Supported Linux image types and kernel versions

Linux image type	kernel version	server version	desktop version
Debian 11 - Bullseye	Linux5.10	support	support
Debian 12 - Bookworm	Linux5.10	support	support
Ubuntu 20.04 - Focal	Linux5.10	support	support
Ubuntu 22.04 - Jammy	Linux5.10	support	support
Debian 11 - Bullseye	Linux6.1	support	support
Debian 12 - Bookworm	Linux6.1	support	support
Ubuntu 20.04 - Focal	Linux6.1	support	support
Ubuntu 22.04 - Jammy	Linux6.1	support	support

### 3. 2. Linux system adaptation

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

Function	Debian11	Debian12	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
eMMC	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



<b>HDMI video</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>HDMI audio</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>OV13850 camera</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>OV13855 camera</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>LCD1</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>LCD2</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>Gigabit Ethernet port</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>Network port status light</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>MIC</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>headphone playback</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>headphone recording</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>LED</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>GPU</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>NPU</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>VPU</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>switch button</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>watchdog test</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>Chromium hard solution video</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>

### 3. 2. 2. Linux6.1 system adaptation situation

<b>Function</b>	<b>Debian11</b>	<b>Debian12</b>	<b>Ubuntu20.04</b>	<b>Ubuntu22.04</b>
<b>USB2.0x2</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>USB3.0x1</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>USB Type-C 3.0</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>DP display</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>eMMC</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>AP6275P-WIFI</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>AP6275P-Bluetooth</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>GPIO (26pin)</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>UART (26pin)</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>SPI (26pin)</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>I2C (26pin)</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>CAN (26pin)</b>	<b>NO</b>	<b>NO</b>	<b>NO</b>	<b>NO</b>
<b>PWM (26pin)</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>



<b>3pin debugging serial port</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>TF card start</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>HDMI video</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>HDMI audio</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>OV13850 camera</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>OV13855 camera</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>LCD1</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>LCD2</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>Gigabit Ethernet port</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>Network port status light</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>MIC</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>headphone playback</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>headphone recording</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>LED</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>GPU</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>NPU</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>VPU</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>switch button</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>watchdog test</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>
<b>Chromium hard solution video</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>	<b>OK</b>

### 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:~\$** prompt indicates that this command is entered in **the linux system of the development board**. The **\$** at the end of the prompt indicates that the current user of the system is an ordinary user. When executing a privileged command, you need to add **sudo**
- c. **root@orangePi:~#** prompt indicates that this command is entered in **the linux system of the development board**, and the last **#** of the prompt indicates that the current user of the system is the root user, who can execute any desired command
- d. **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. **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**

```
orangepi5 login: orangepi (automatic login)

  O P I 5

Welcome to Orange Pi 1.0.0 Bullseye with Linux 5.10.110-rockchip-rk3588

System load: 27%      Up time: 0 min
Memory usage: 7% of 7.51G  IP: 192.168.1.219
CPU temp: 59°C      Usage of /: 14% of 29G

[ General system configuration (beta): orangepi-config ]

Last login: Thu Dec  1 13:11:02 UTC 2022 on tty1
orangepi@orangepi5:~$
```

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

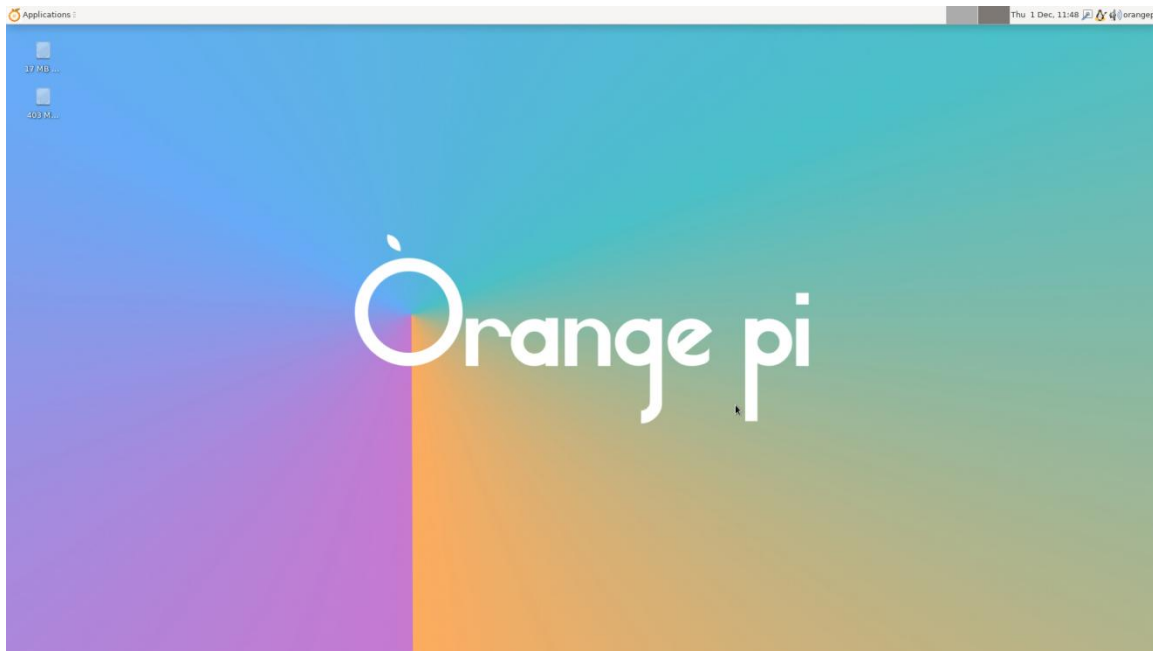
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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

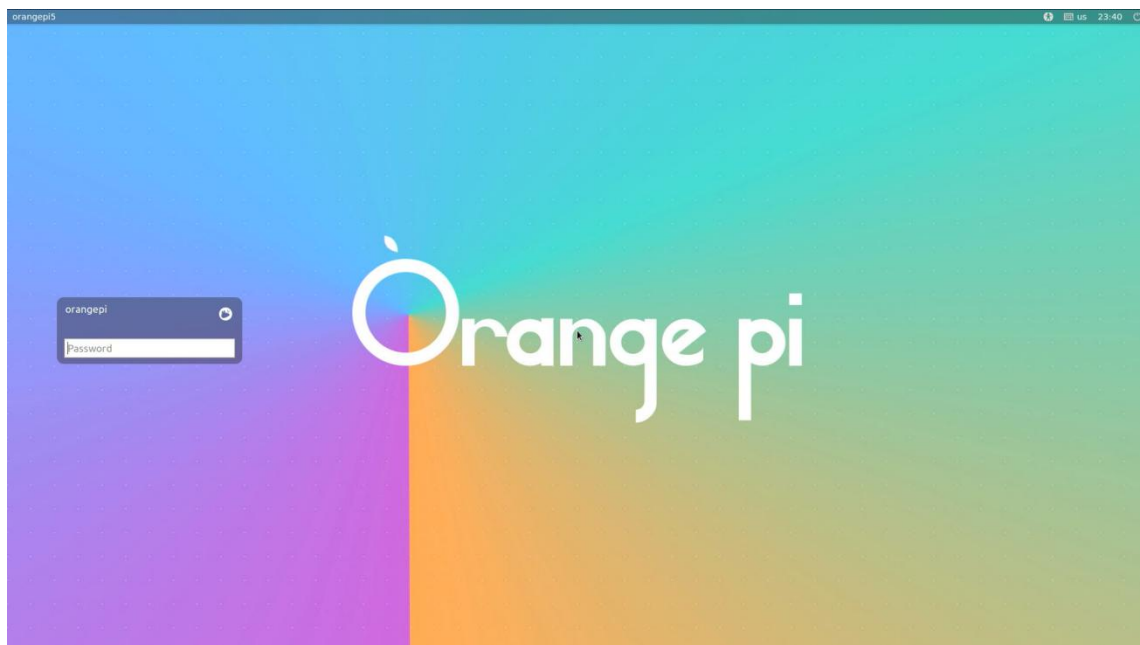
- 1) After the desktop version system is started, it will automatically log in to the desktop without entering a password



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



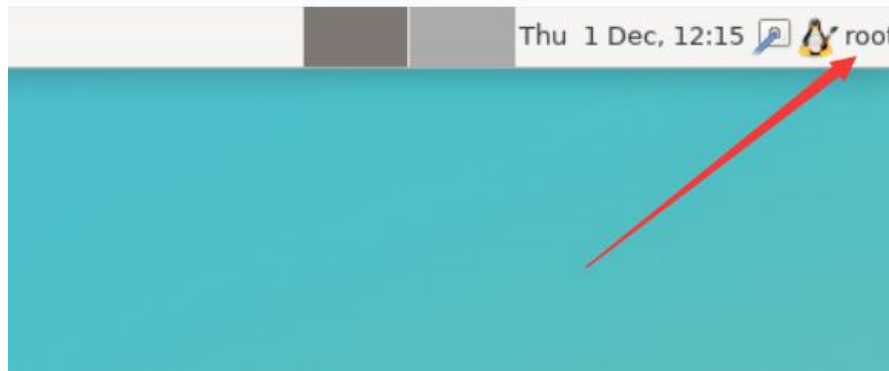


### 3. 4. 4. The setting method of root user automatic login in Linux desktop version system

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

```
orange@orange:~$ 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 orange user again

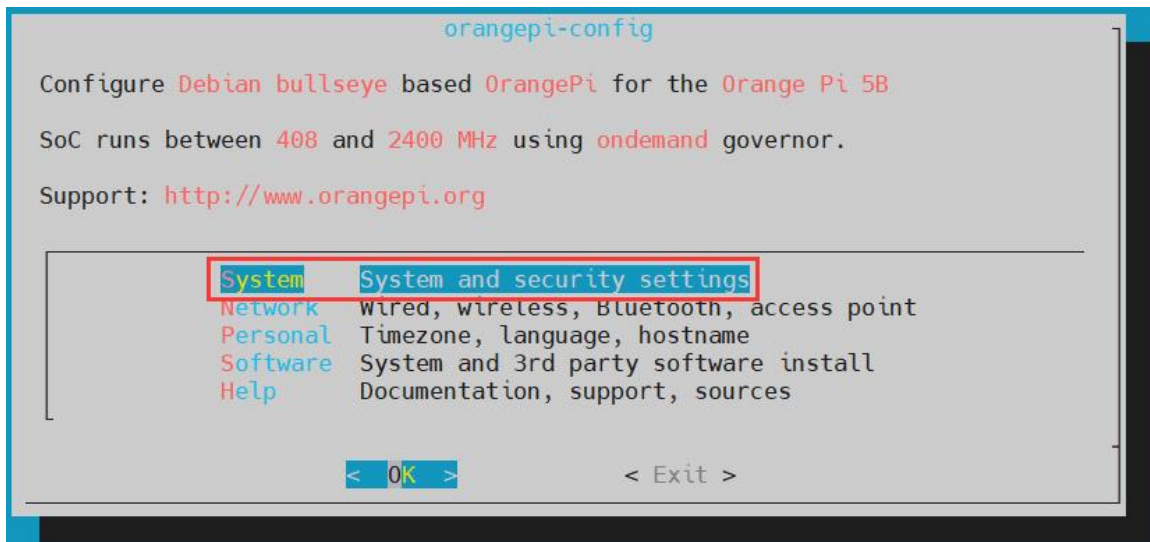
```
orange@orange:~$ sudo desktop_login.sh orange
```

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

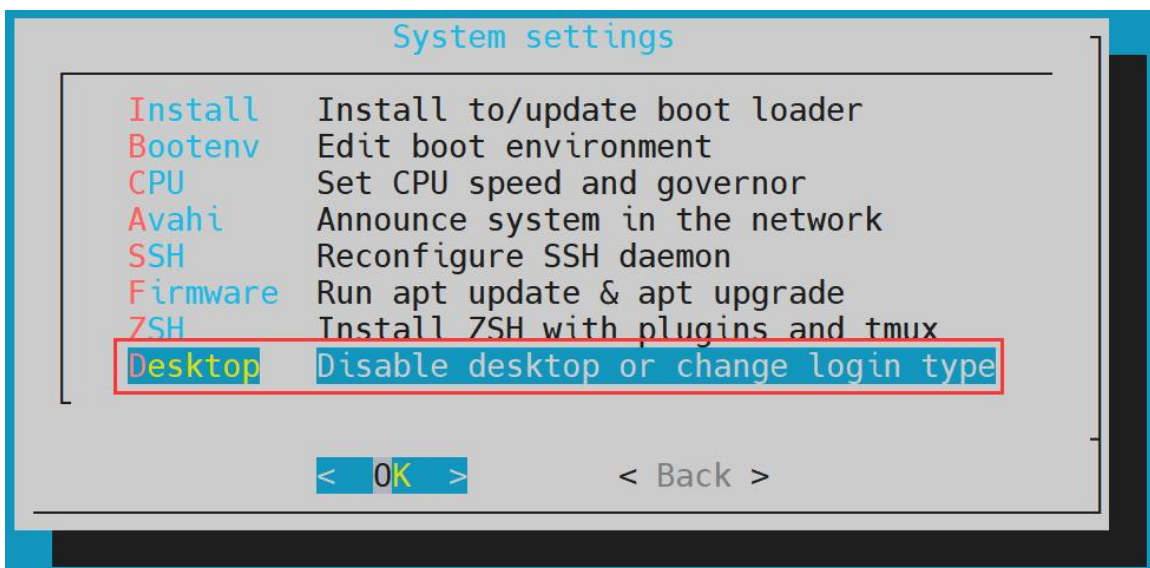
- 1) First enter the following command in the command line, **please remember to add sudo permission**

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

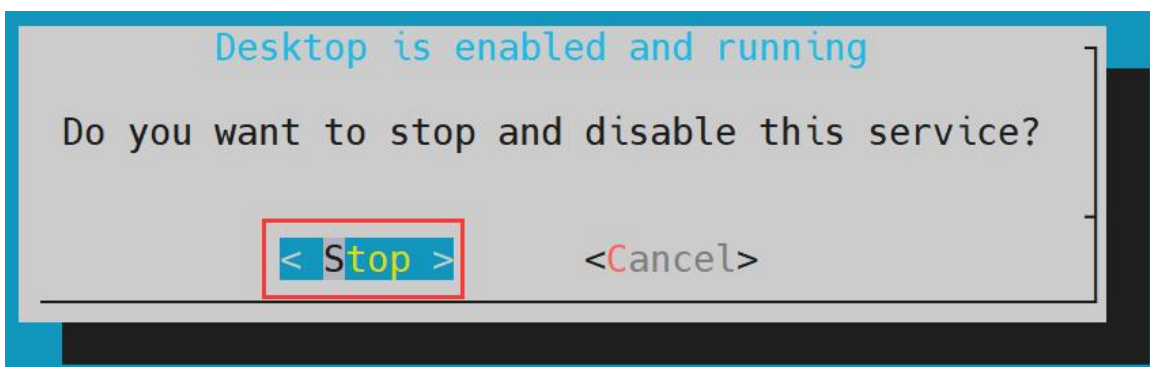
- 2) Then select **System**



3) Then select **Desktop**



4) Then select **<Stop>**





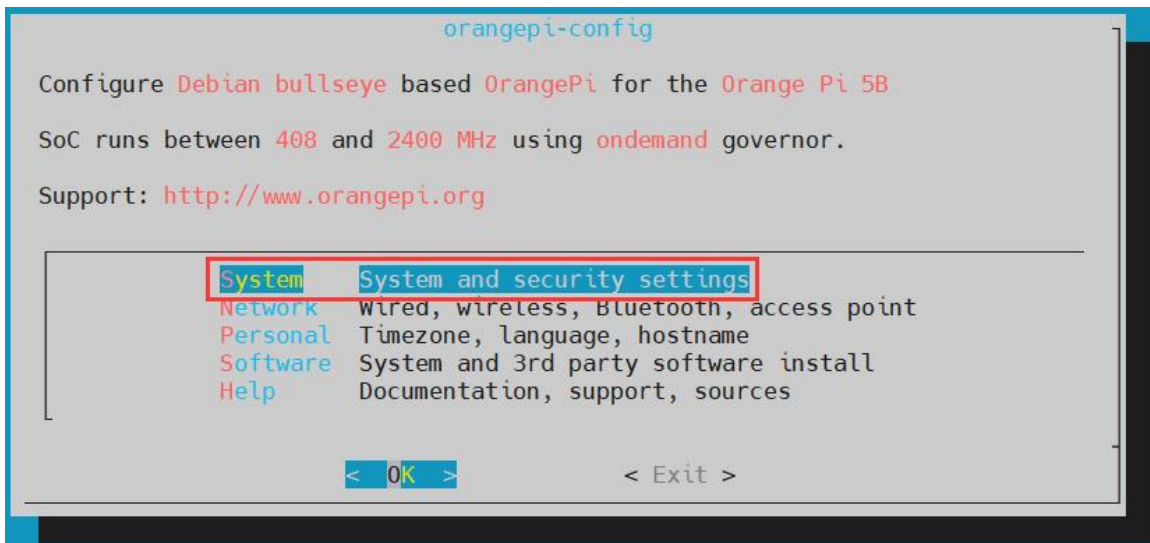
5) Then restart the Linux system and you will find that the desktop will not be displayed

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

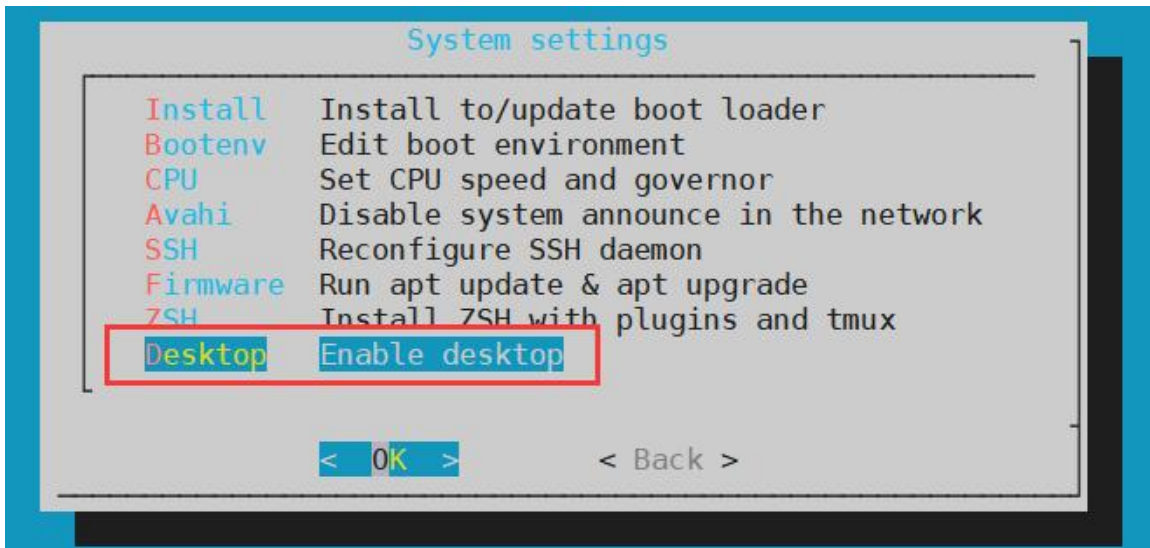
- a. First enter the following command in the command line, **please remember to add sudo permission**

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

- b. Then select **System**



- c. Then select **Desktop**    **Enable desktop**

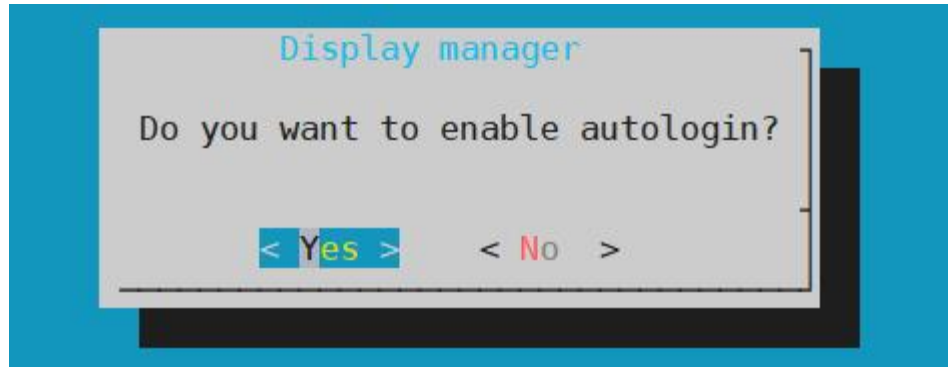


- d. Then choose whether to automatically log in to the desktop. If you select **<Yes>**, you will automatically log in to the desktop. If you select **<No>**, the input





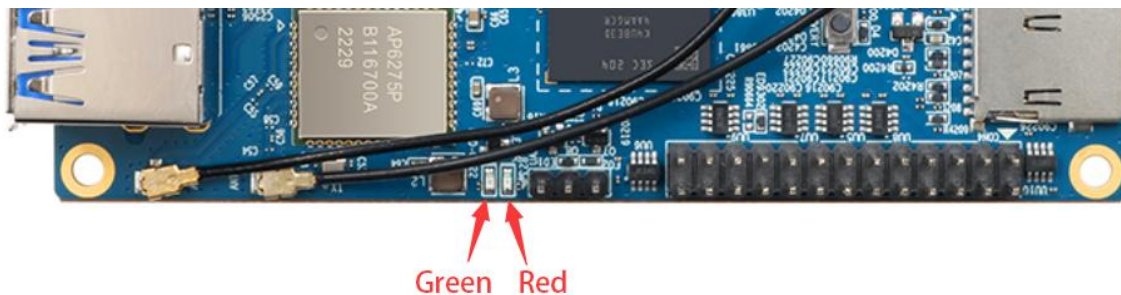
interface for user and password will be displayed, and you need to enter the password to enter the desktop.



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

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

- a. First enter the setting directory of the green light

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

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



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

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

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

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

```
root@orangepi:/sys/class/leds/status_led# echo 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**, **without any other configuration**

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

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

```
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 prompted, 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

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

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



```

root@orangepi:~# nmcli dev wifi
IN-USE  BSSID          SSID          MODE  CHAN  RATE        SIGNAL  BARS  SECURITY
28:6C:07:6E:87:2E  orangepi      Infra    9     260 Mbit/s  97      █████ WPA1 WPA2
D8:D8:66:A5:BD:D1  orangepi      Infra   10     270 Mbit/s  90      █████ WPA1 WPA2
A0:40:A0:A1:72:20  orangepi      Infra    4     405 Mbit/s  82      █████ WPA2
28:6C:07:6E:87:2F  orangepi_5G   Infra  149     540 Mbit/s  80      █████ WPA1 WPA2
CA:50:E9:89:E2:44  ChinaNet_TC15 Infra    1     130 Mbit/s  79      █████ WPA1 WPA2
A0:40:A0:A1:72:31  NETGEAR      Infra  100     405 Mbit/s  67      █████ WPA2
D4:EE:07:08:A9:E0  orangepi      Infra    4     130 Mbit/s  55      █████ WPA1 WPA2
88:C3:97:49:05:13  orangepi      Infra    6     130 Mbit/s  52      █████ WPA1 WPA2
00:BD:82:51:53:C2  orangepi      Infra   12     130 Mbit/s  49      █████ WPA1 WPA2
C0:61:18:FA:49:37  orangepi      Infra  149     270 Mbit/s  47      █████ WPA1 WPA2
04:79:70:8D:0C:B8  orangepi      Infra  153     270 Mbit/s  47      █████ WPA2
04:79:70:FD:0C:B8  orangepi      Infra  153     270 Mbit/s  47      █████ WPA2
9C:A6:15:DD:E6:0C  orangepi      Infra   10     270 Mbit/s  45      █████ WPA1 WPA2
B4:0F:3B:45:D1:F5  orangepi      Infra   48     270 Mbit/s  45      █████ WPA1 WPA2
E8:CC:18:4F:7B:44  orangepi      Infra  157     135 Mbit/s  45      █████ WPA1 WPA2
B0:95:8E:D8:2F:ED  orangepi      Infra   11     405 Mbit/s  39      █████ WPA1 WPA2
C0:61:18:FA:49:36  orangepi      Infra   11     270 Mbit/s  24      █████ WPA1 WPA2
root@orangepi:~#

```

- 3) Then use the nmcli command to connect to the scanned WIFI hotspot, where:
  - a. **wifi\_name** needs to be replaced with the name of the WIFI hotspot you want to connect to
  - b. **wifi\_passwd** needs to be replaced with the password of the WIFI hotspot you want to connect to

```

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

```

- 4) Through the **ip addr show wlan0** command, you can view the IP address of wifi

```

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

```

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



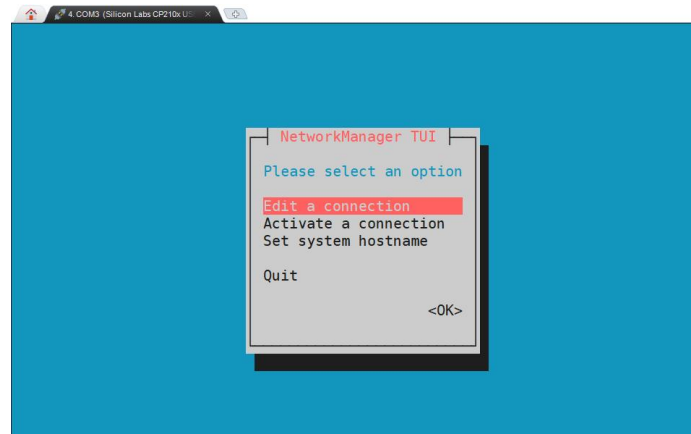
```
orangepi@orangepi:~$ ping www.orangepi.org -I wlan0
PING www.orangepi.org (182.92.236.130) from 192.168.1.49 wlan0: 56(84) bytes of
data.
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=1 ttl=52 time=43.5 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=2 ttl=52 time=41.3 ms
64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

### 3. 6. 2. 2. The server image connects to WIFI in a graphical way

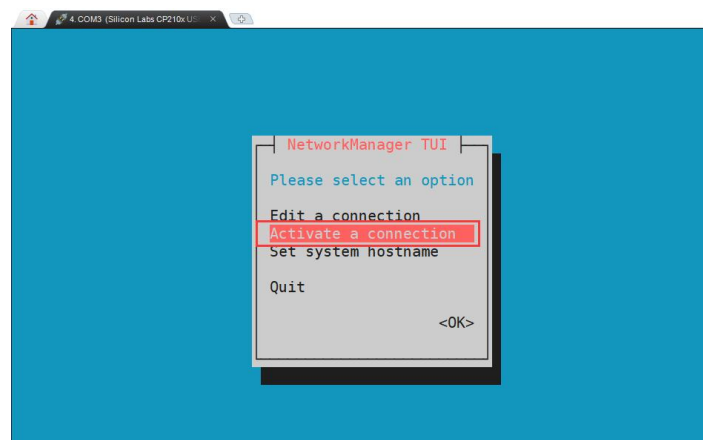
- 1) First log in to the linux system, there are the following three ways
  - a. If the development board is connected with a network cable, you can remotely log in to the Linux system through ssh
  - b. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the linux system (please use MobaXterm for the serial port software, and 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
```

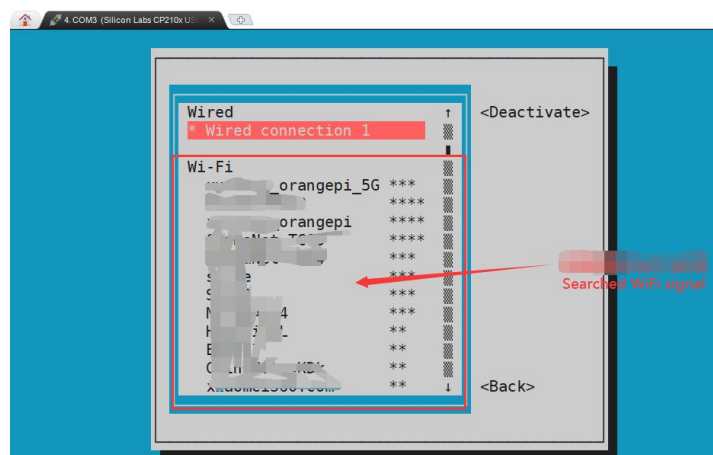
- 3) Enter the nmtui command to open the interface as shown below



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

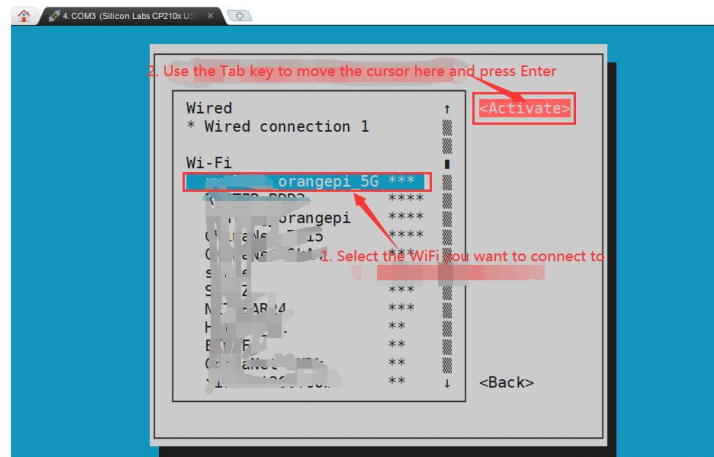


5) Then you can see all the searched WIFI hotspots

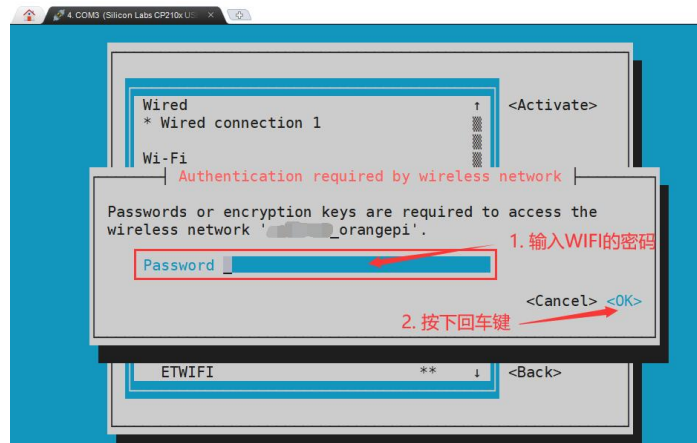


6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the cursor on **Activate** and press Enter

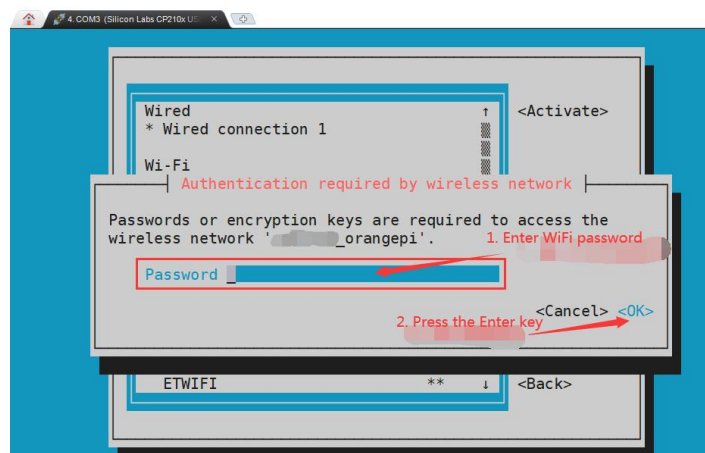




7) Then a dialog box for entering a **password** will pop up, enter the corresponding password in Password and press Enter to start connecting to WIFI



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

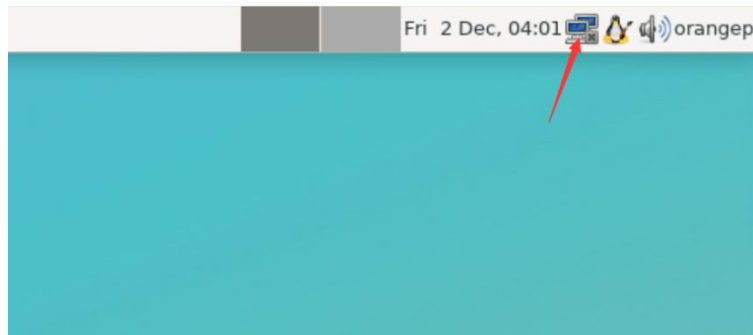
```
orangePi@orangePi:~$ ip addr show wlan0
11: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 24:8c:d3:aa:76:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
        valid_lft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

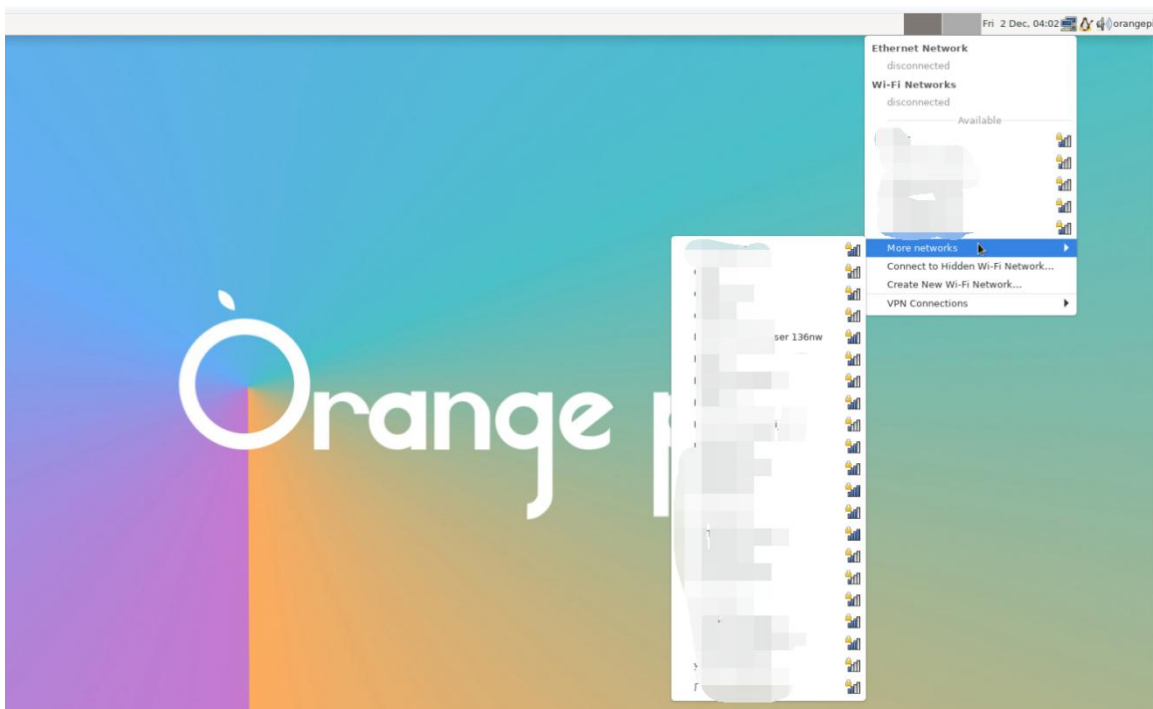
```
orangePi@orangePi:~$ ping www.orangePi.org -I wlan0
PING www.orangePi.org (182.92.236.130) from 192.168.1.49 wlan0: 56(84) bytes of
data.
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=1 ttl=52 time=43.5 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=2 ttl=52 time=41.3 ms
64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

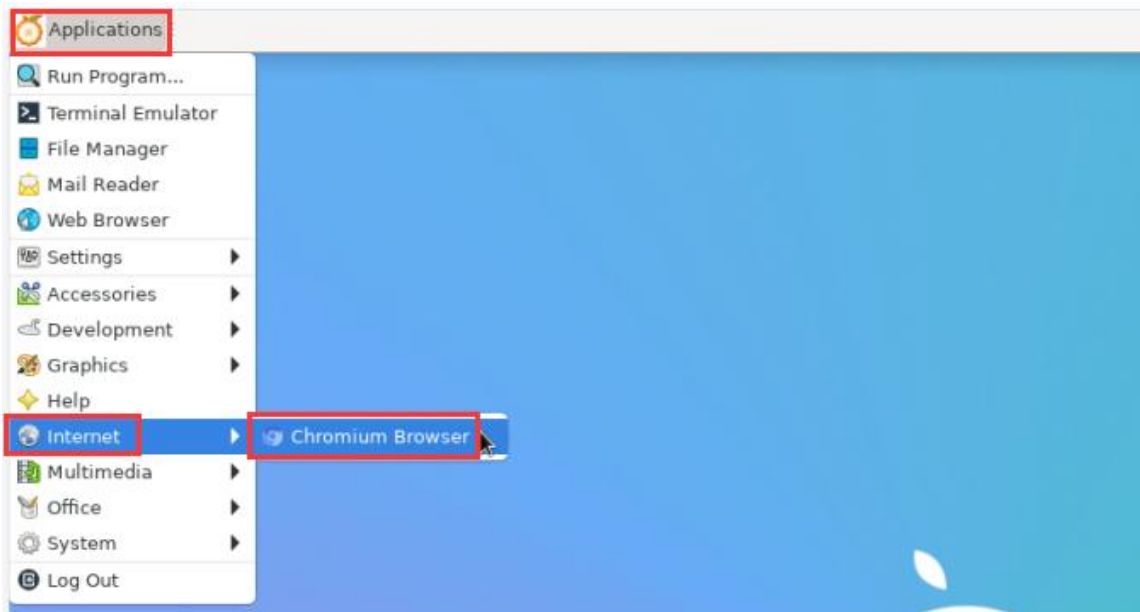


3) Then enter the password of the WIFI hotspot, and then click **Connect** to start connecting to WIFI

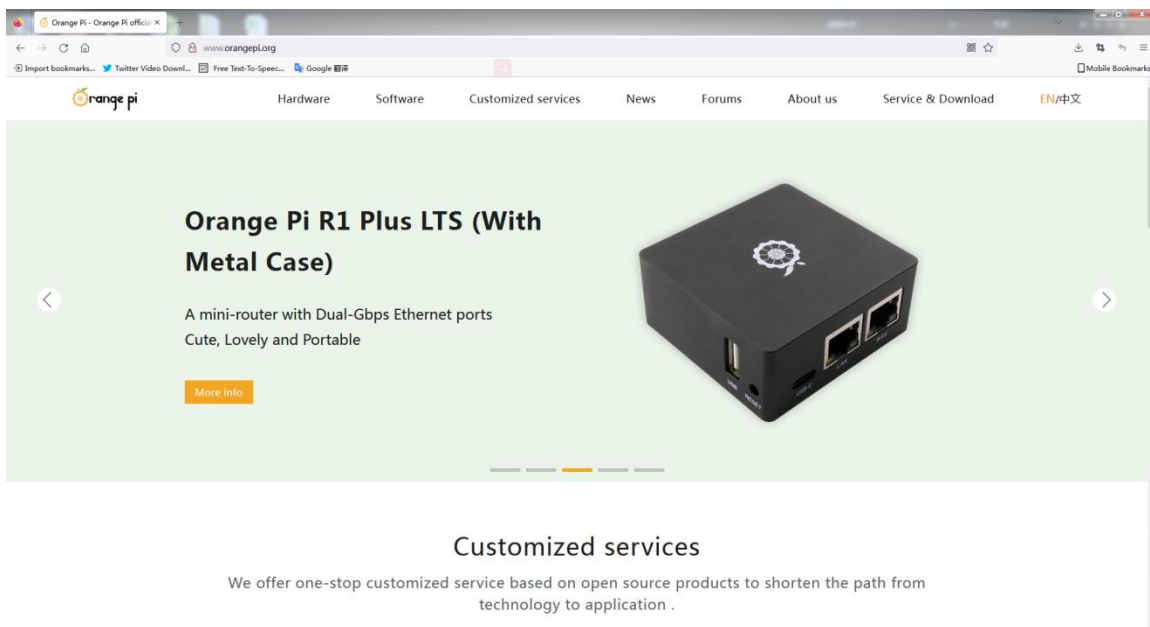




4) After connecting to WIFI, you can open the browser to check whether you can access the Internet. The entrance of the browser is shown in the figure below



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.

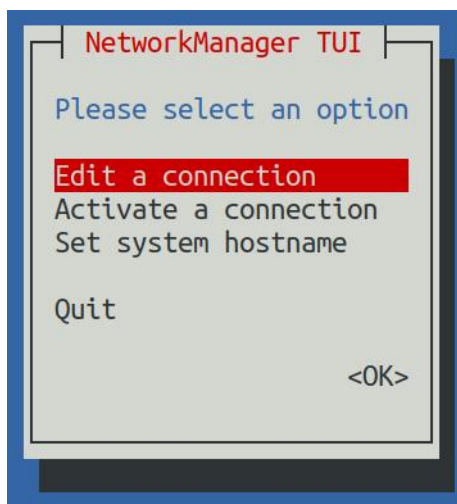


### 3. 6. 3. 1. Use the nmtui command to set a static IP address

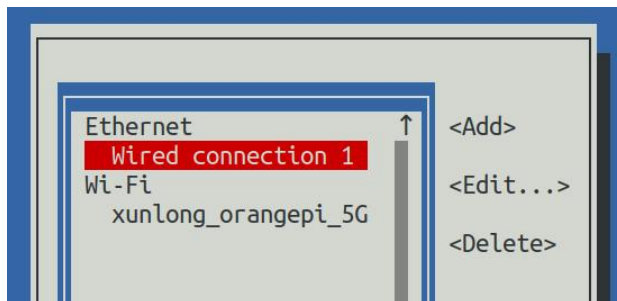
1) First run the **nmtui** command

```
orangeypi@orangeypi:~$ nmtui
```

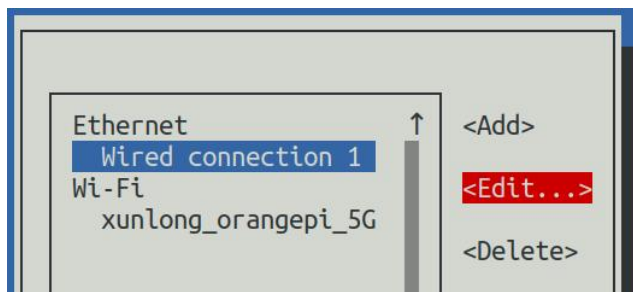
2) Then select **Edit a connection** and press Enter



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**.

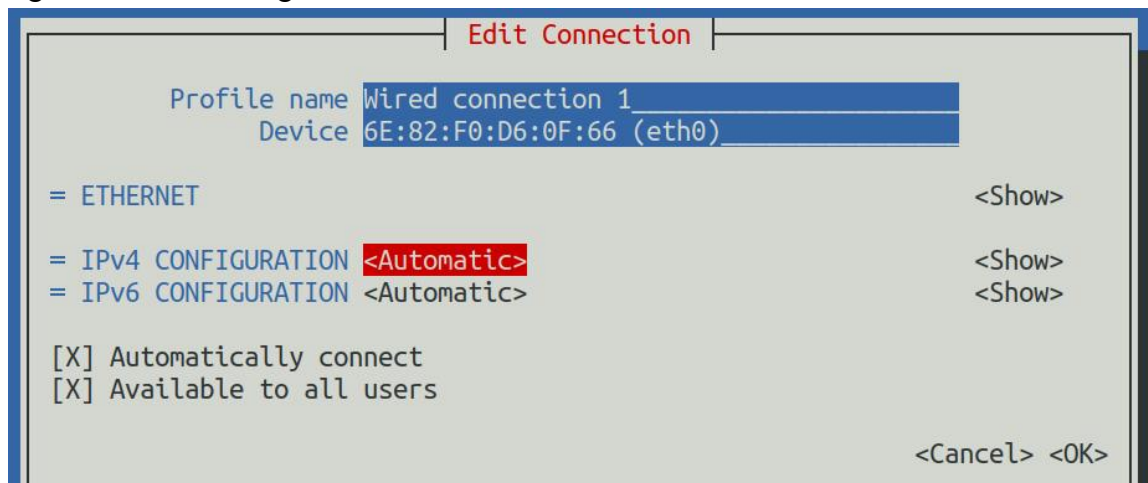


4) Then select **Edit** via the **Tab** key and press the Enter key

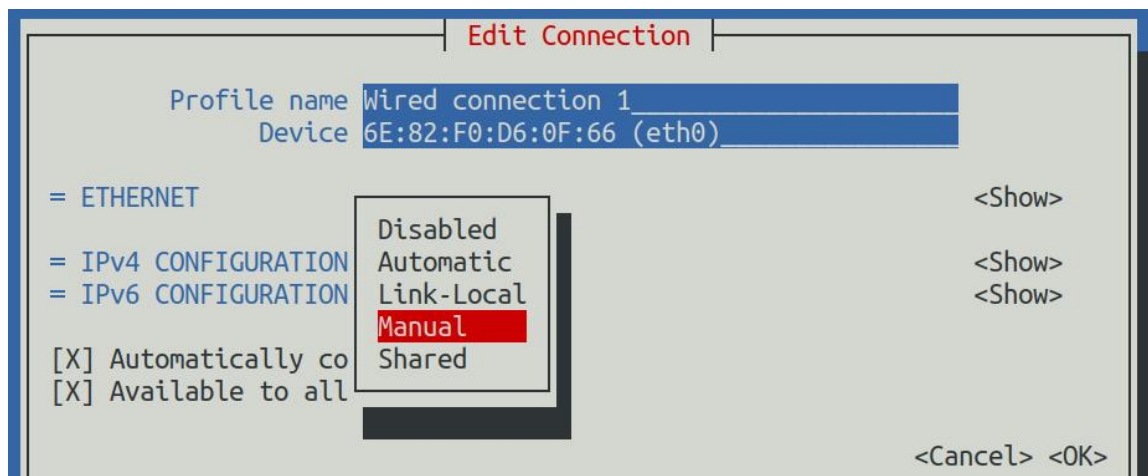




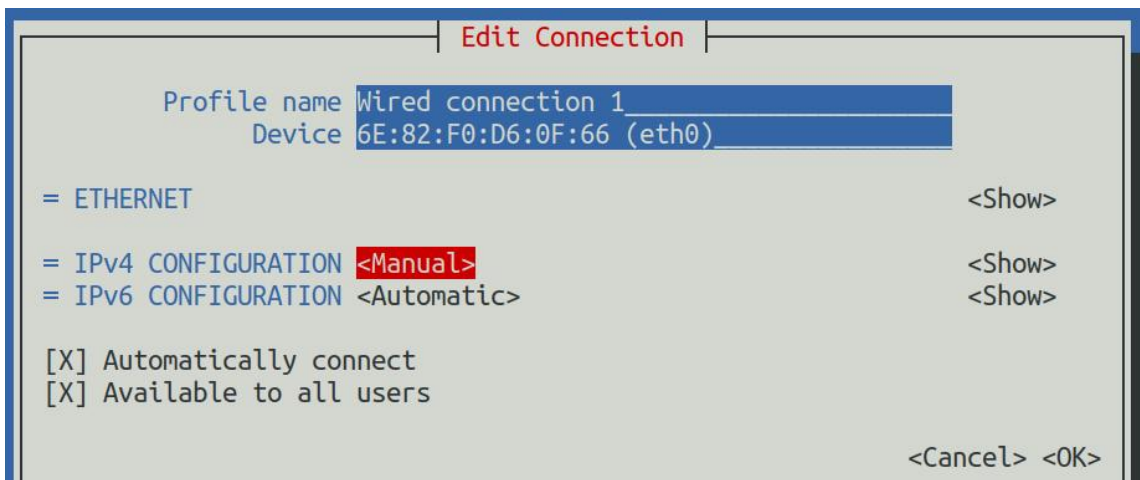
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4



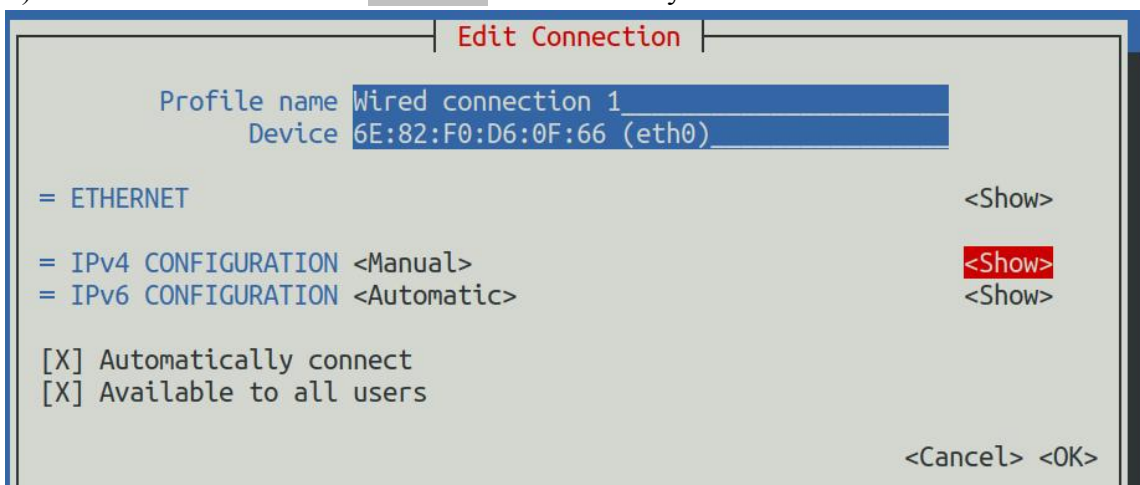
6) Then press Enter, select **Manual** through the up and down arrow keys, and press Enter to confirm



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

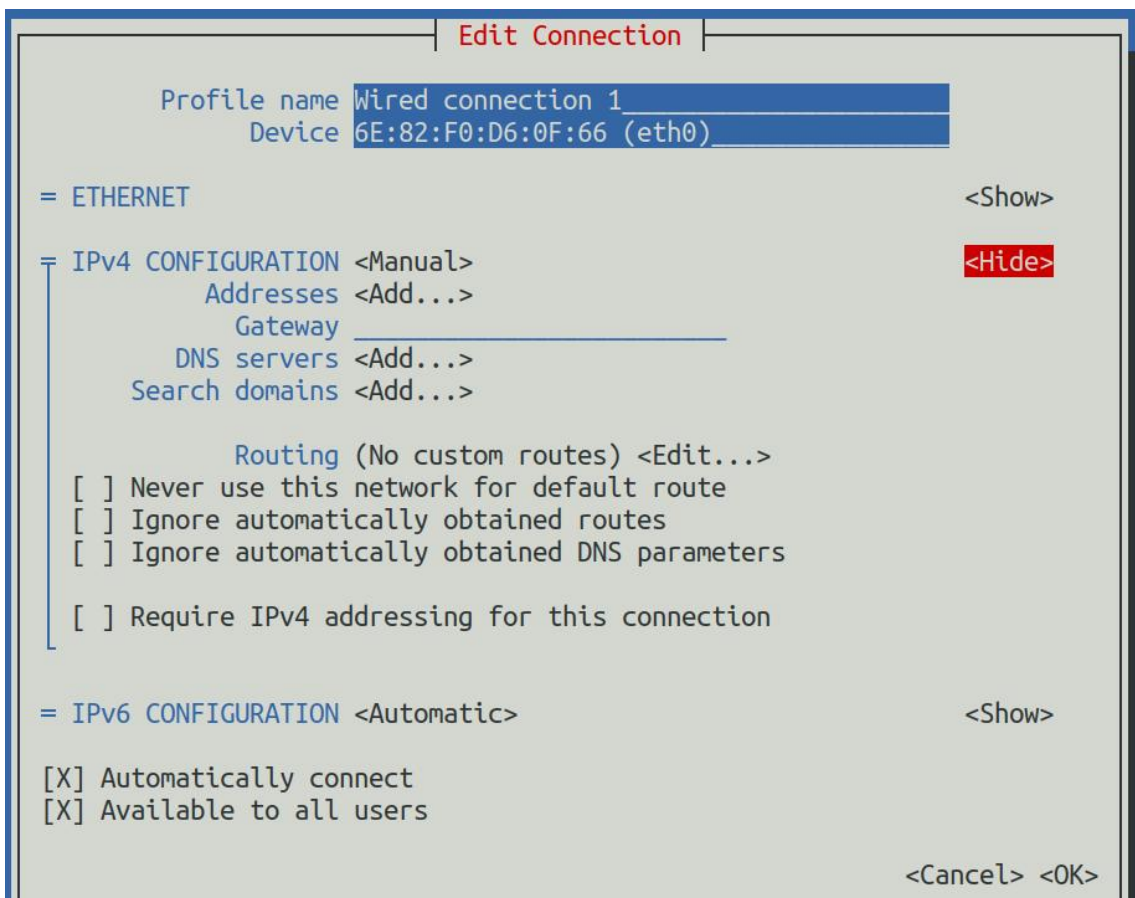


8) Then move the cursor to **<Show>** via the Tab key

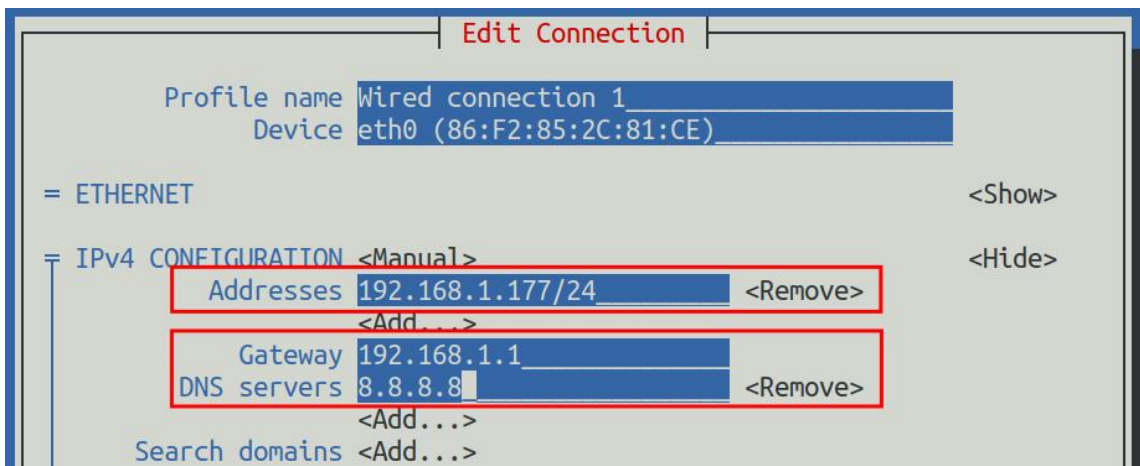


9) Then press Enter, and the following setting interface will pop up after entering





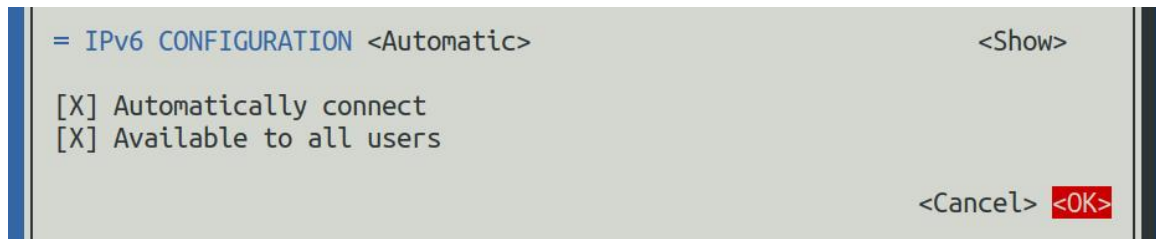
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the position shown in the figure below (there are many other setting options in it, please explore by yourself), **please set according to your specific needs, The values set in the image below are just an example**



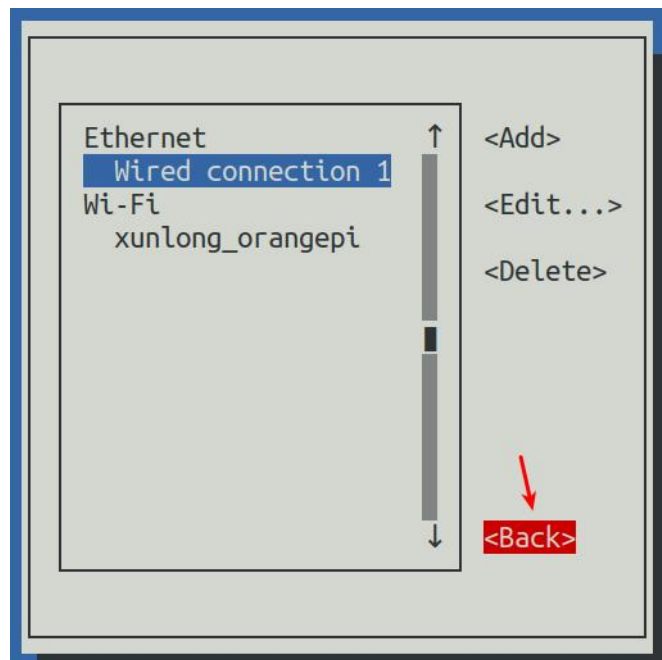
11) After setting, move the cursor to **<OK>** in the lower right corner, and press Enter to



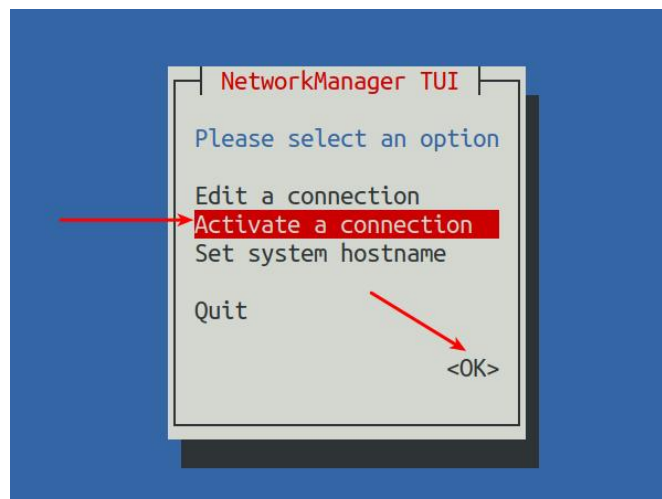
confirm



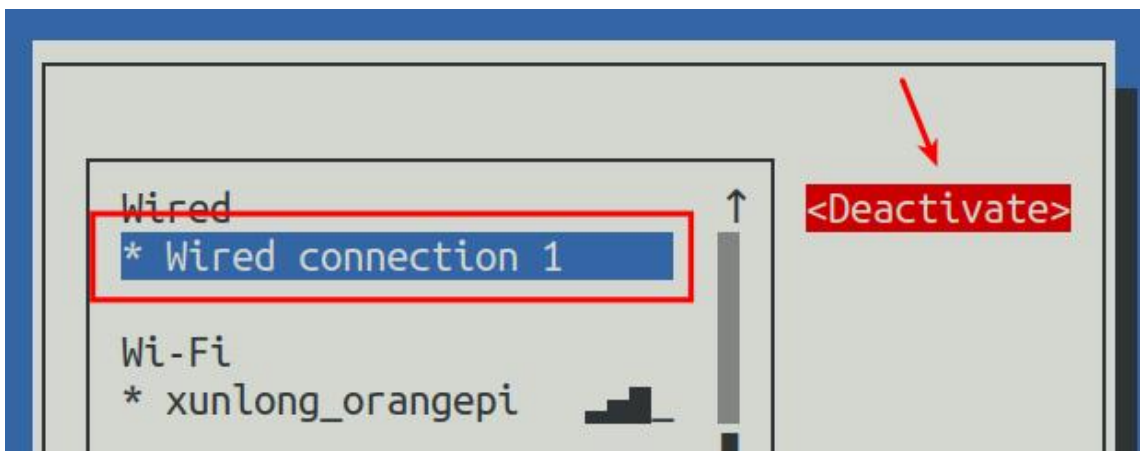
12) Then click **<Back>** to return to the previous selection interface



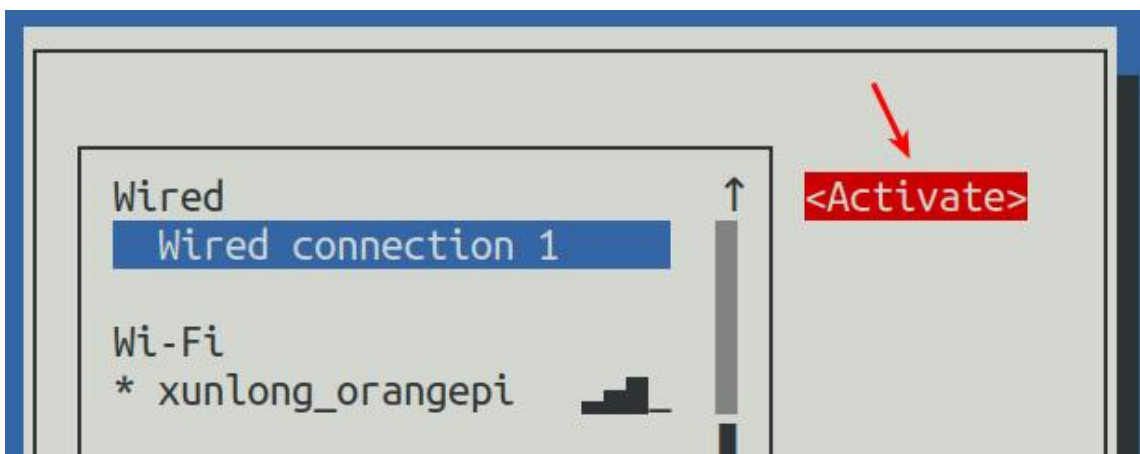
13) Then select **Activate a connection**, then move the cursor to **<OK>**, and finally click Enter



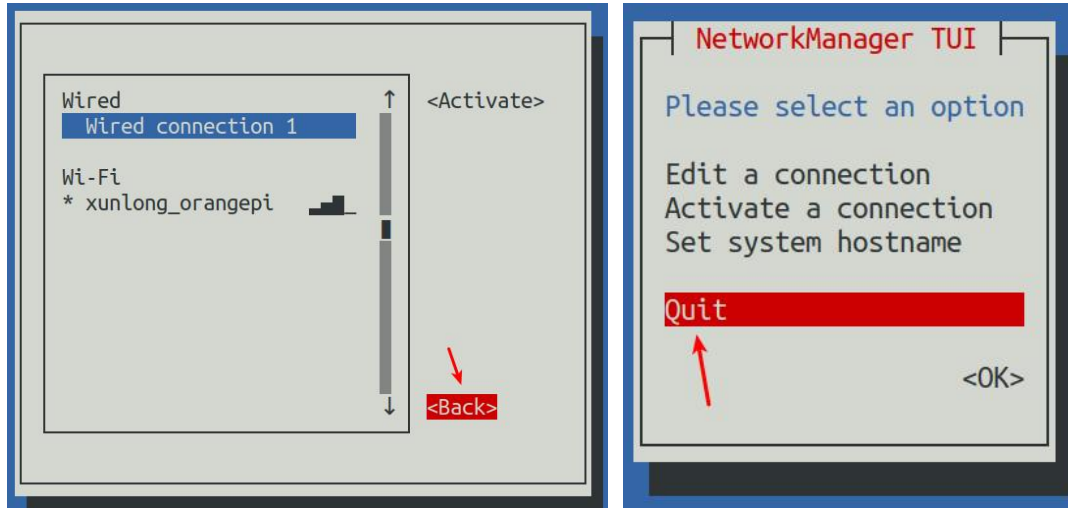
- 14) Then select the network interface that needs to be set, such as **Wired connection 1**, then move the cursor to **<Deactivate>**, and press Enter to disable **Wired connection 1**



- 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

```
orangeipi@orangeipi:~$ 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**

```
orangeipi@orangeipi:~$ 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
<b>orangepi</b>	cfc4f922-ae48-46f1-84e1-2f19e9ec5e2a	wifi	wlan0
<b>Wired connection 1</b>	9db058b7-7701-37b8-9411-efc2ae8bfa30	ethernet	eth0

3) Then enter the following command, where

- a. **"Wired connection 1"** means to set the static IP address of the Ethernet port. If you need to set the static IP address of the WIFI, please change it to the corresponding name of the WIFI network interface (you can get it through the **nmcli con show** command)
- b. **ipv4.addresses** is followed by the static IP address to be set, which can be modified to the value you want to set
- c. **ipv4.gateway** 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. 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](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, usually eth0

\* **access-point-name:** hotspot name

\* **passphrase:** the password of the hotspot

### 3. 6. 4. 1. create\_ap method to create WIFI hotspot in NAT mode

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

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

```
orangepi@orangepi5b:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi
```

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

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

```
orangepi@orangepi5b:~$ 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@orangepi5b:~$ 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. For example, use the -g parameter to specify the network segment of the access point AP as 192.168.2.1.

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

```
orangepi@orangepi5b:~$ 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

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

```
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

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

```
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. 4. 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

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

```
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**

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

```
orange@orange:~$ 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 **orange**, 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

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

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

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

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

```
orangeapi@orangeapi:~$ sudo create_ap -m bridge wlan0 eth0 orangeapi orangeapi --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 logging in with ssh, you first need to ensure that the Ethernet or wifi network is connected, and then use the ip addr command or check the router to obtain the IP address of the development board.**

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

1) Obtain the IP address of the development board

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

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

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

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

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



```
test@test:~$ ssh orangepi@192.168.1.192
orangepi@192.168.1.192's password:

Welcome to Orange Pi 1.0.0 Bullseye with Linux 5.10.110-rockchip-rk3588

System load: 1%          Up time: 9 min    Local users: 2
Memory usage: 3% of 7.51G  IP: 192.168.1.192
CPU temp: 48°C          Usage of /: 33% of 15G

[ 0 security updates available, 10 updates total: apt upgrade ]
Last check: 2023-02-07 10:45

Last login: Tue Feb  7 10:53:56 2023 from 192.168.1.5
orangepi@orangepi5b:~$
```

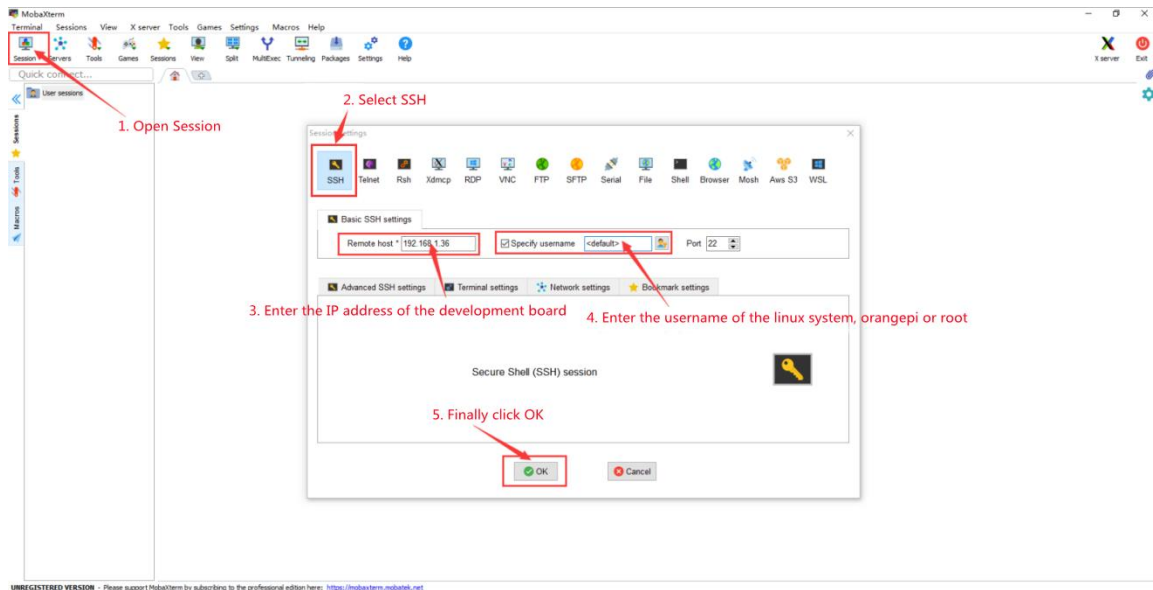
If ssh cannot log in to the linux system normally, please first check whether the IP address of the development board can be pinged. If the ping is ok, you can log in to the linux system through the serial port or HDMI display and then enter the following command on the development board and try again.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

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

If it still doesn't work, please reset the system and try again

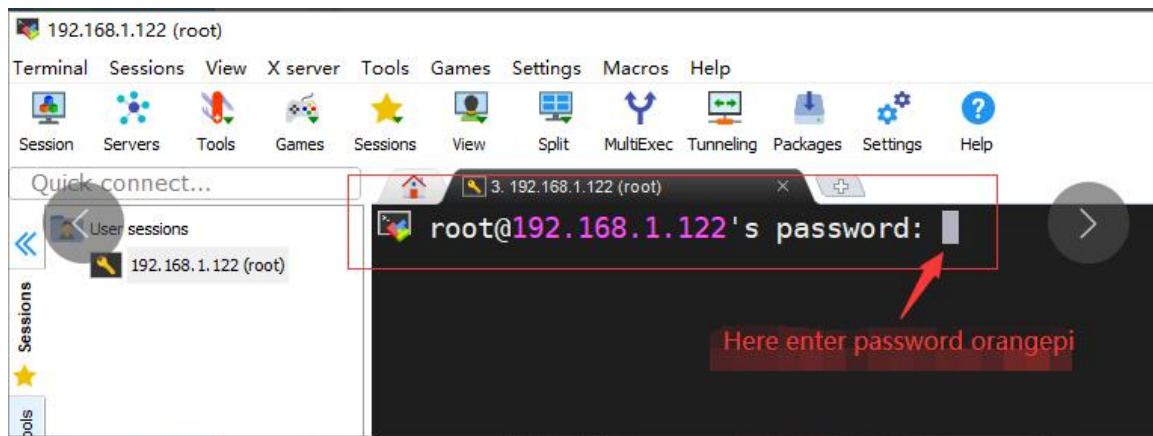
### 3. 7. 2. SSH remote login development board under Windows

- 1) First obtain the IP address of the development board
- 2) Under Windows, you can use MobaXterm to remotely log in to the development board, first create a new ssh session
  - a. Open **Session**
  - b. Then select **SSH** in **Session Setting**
  - c. Then enter the IP address of the development board in the **Remote host**
  - d. Then enter the user name **root** or **orangepi** of the linux system in **Specify username**
  - e. Finally click **OK**

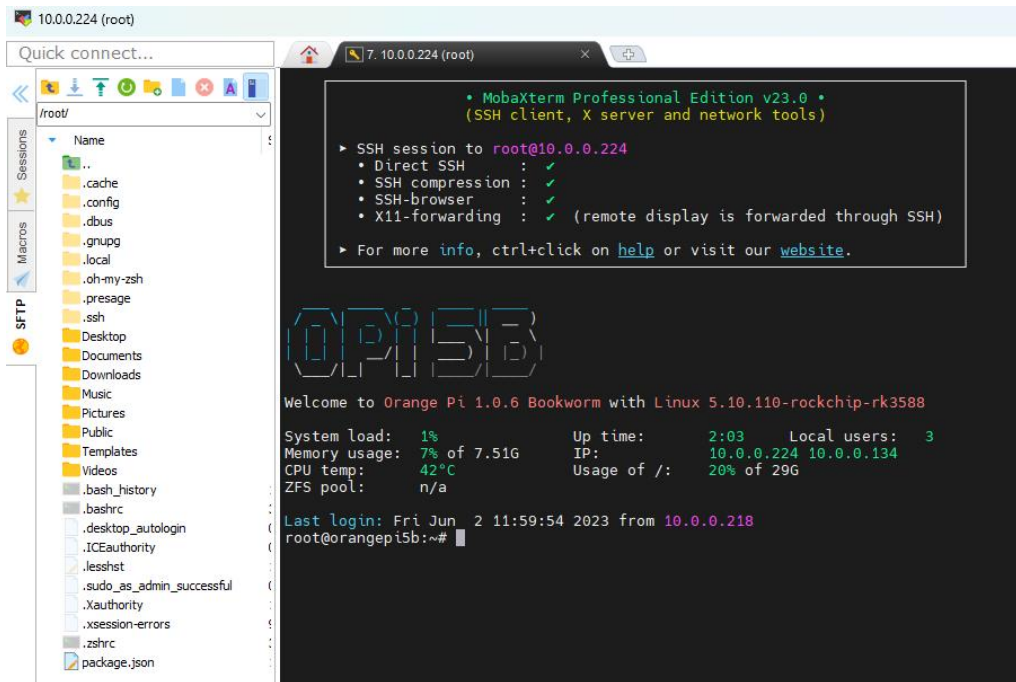


3) Then you will be prompted to enter a password. The default passwords for root and orangepi users are orangepi

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



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



## 3.8. 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@orangePi5b:/# <--- 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 you do not have the adb tool in your Windows system, you can use the adb program in the RKDevTool software (this software is useful in the section on how to burn the Android image to SPIFlash+NVMe SSD).**

桌面 > RKDevTool\_Release\_v2.92 > bin

名称	修改日期	类型	大小
adb	2019/6/24 9:13	应用程序	1,807 KB
AdbWinApi.dll	2019/6/24 9:13	应用程序扩展	96 KB
AdbWinUsbApi.dll	2019/6/24 9:13	应用程序扩展	62 KB
AFPTool	2021/8/23 9:04	应用程序	874 KB
RKImageMaker	2021/8/16 14:05	应用程序	870 KB

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



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

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

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

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin 的目录
2022/08/09 13:19 <DIR> .
2022/08/09 13:19 <DIR> ..
2019/06/24 09:13      1,850,368 adb.exe
2019/06/24 09:13      97,792 AdbWinApi.dll
2019/06/24 09:13      62,976 AdbWinUsbApi.dll
2021/08/23 09:04      894,976 AFPTool.exe
2021/08/16 14:05      890,368 RKImageMaker.exe
                5 个文件      3,796,480 字节
                2 个目录 64,033,034,240 可用字节

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

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

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

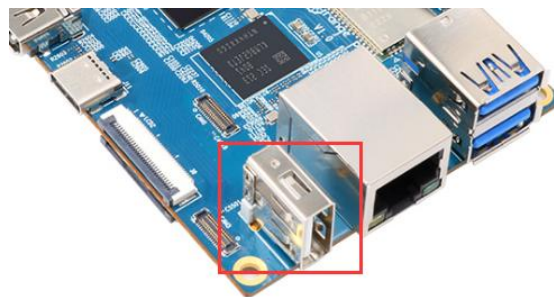
C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>
```

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

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



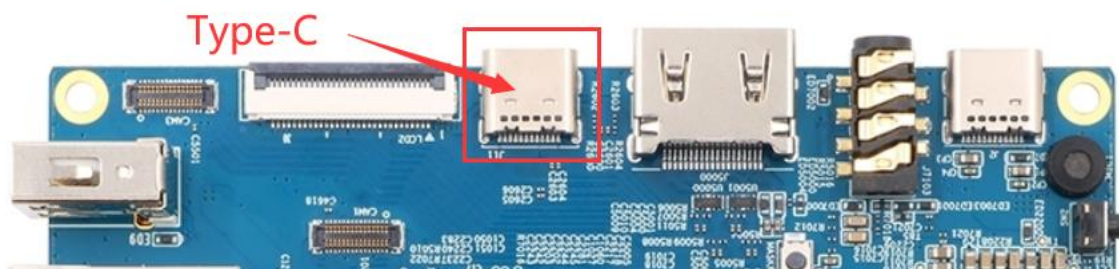
2) Then please make sure that there is no USB device plugged into the USB interface below







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@orangePi5b:/#** <--- 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 you do not have the adb tool in your Windows system, you can use the adb program in the RKDevTool software (this software is useful in the section on how to burn the Android image to SPIFlash+NVMe SSD).**

桌面 > RKDevTool\_Release\_v2.92 > bin

名称	修改日期	类型	大小
adb	2019/6/24 9:13	应用程序	1,807 KB
AdbWinApi.dll	2019/6/24 9:13	应用程序扩展	96 KB
AdbWinUsbApi.dll	2019/6/24 9:13	应用程序扩展	62 KB
AFPTool	2021/8/23 9:04	应用程序	874 KB
RKImageMaker	2021/8/16 14:05	应用程序	870 KB

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

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

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

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin 的目录
2022/08/09 13:19 <DIR>          .
2022/08/09 13:19 <DIR>          ..
2019/06/24 09:13             1,850,368 adb.exe
2019/06/24 09:13             97,792 AdbWinApi.dll
2019/06/24 09:13             62,976 AdbWinUsbApi.dll
2021/08/23 09:04             894,976 AFPTool.exe
2021/08/16 14:05             890,368 RKImageMaker.exe
                    5 个文件          3,796,480 字节
                    2 个目录 63,988,027,392 可用字节

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>adb devices
List of devices attached
e0f9f71bc424c305    device

C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin>adb push adb.exe /root
adb.exe: 1 file pushed. 3.2 MB/s (1850368 bytes in 0.552s)

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



### 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. **orangeypi**: 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/orangeypi**: The path in the Linux system of the development board can also be modified to other paths

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

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

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

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

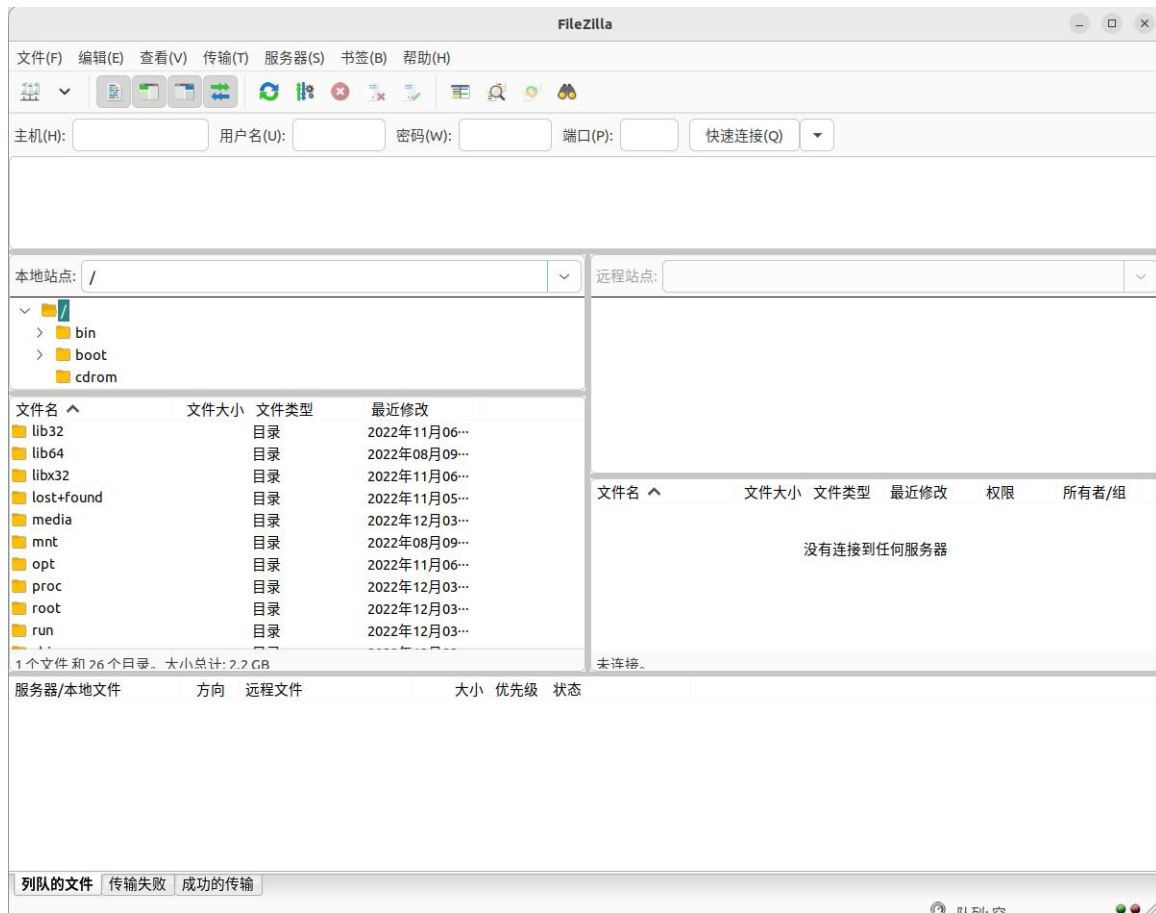
```
test@test:~$ sudo apt install -y filezilla
```

2) Then use the following command to open filezilla

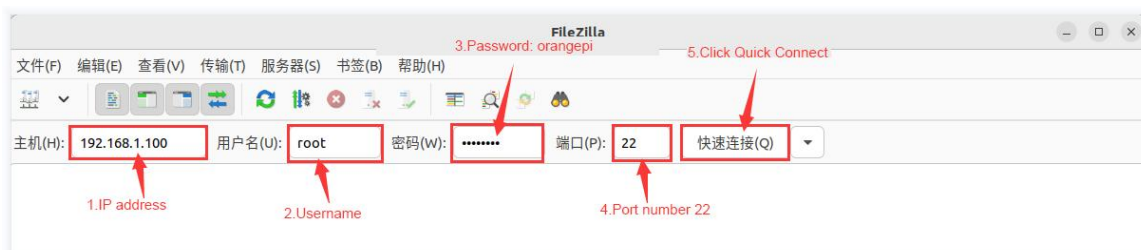
```
test@test:~$ filezilla
```



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



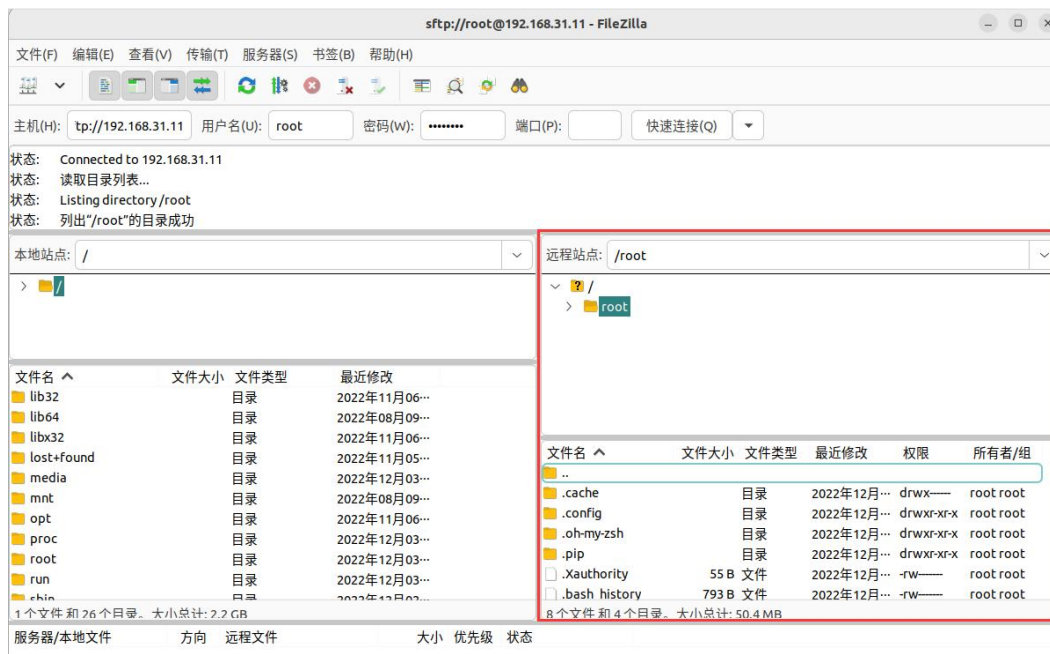
5) Then choose to **save the password**, and then click **OK**



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



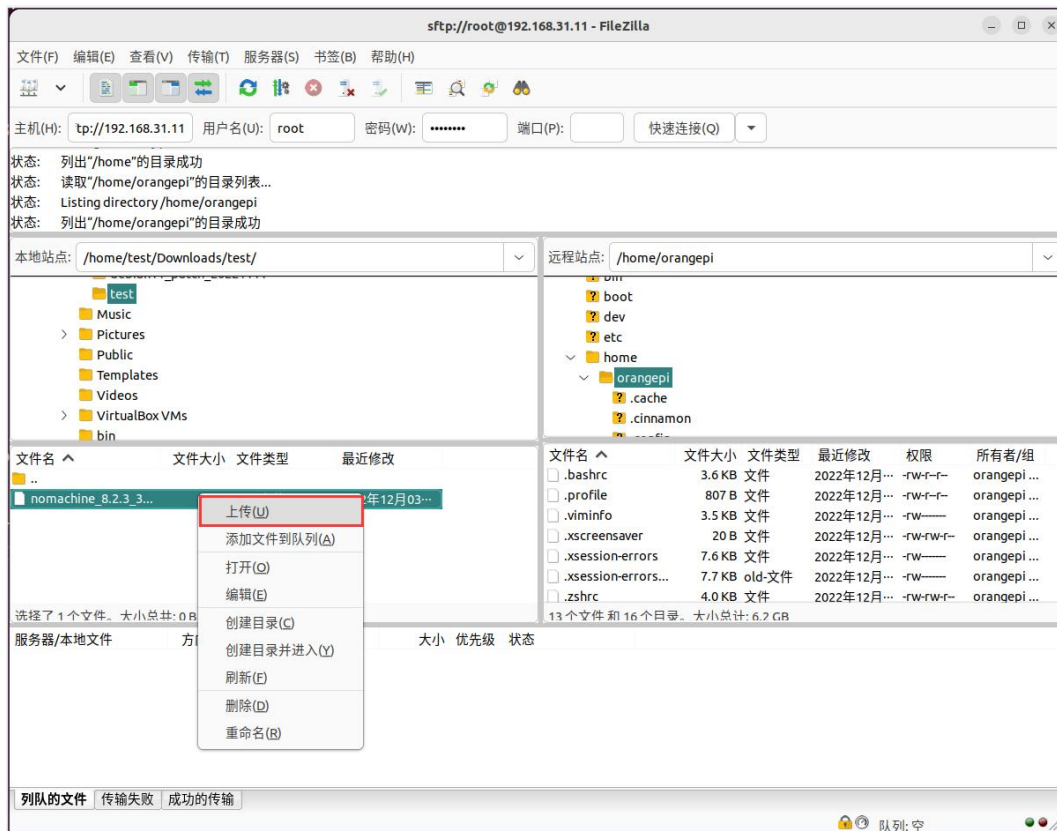
7) After the connection is successful, you can see the directory structure of the development board linux file system on the right side of the filezilla software



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 in 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 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

### 3.9.2. 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?type=client>



**Please select your edition of FileZilla Client**

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

At the bottom of the table, there are four buttons: 'Download', 'Select', 'Select', and 'Select'. A red arrow points to the 'Download' button.

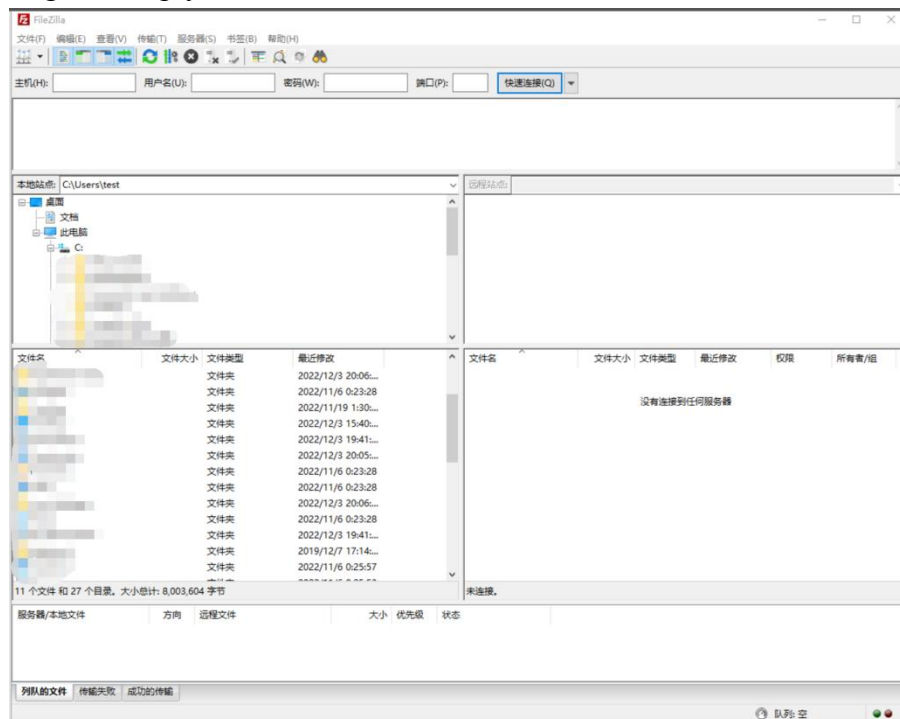
2) The downloaded installation package is as follows, and then double-click to install directly

**FileZilla\_Server\_1.5.1\_win64-setup.exe**

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



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



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





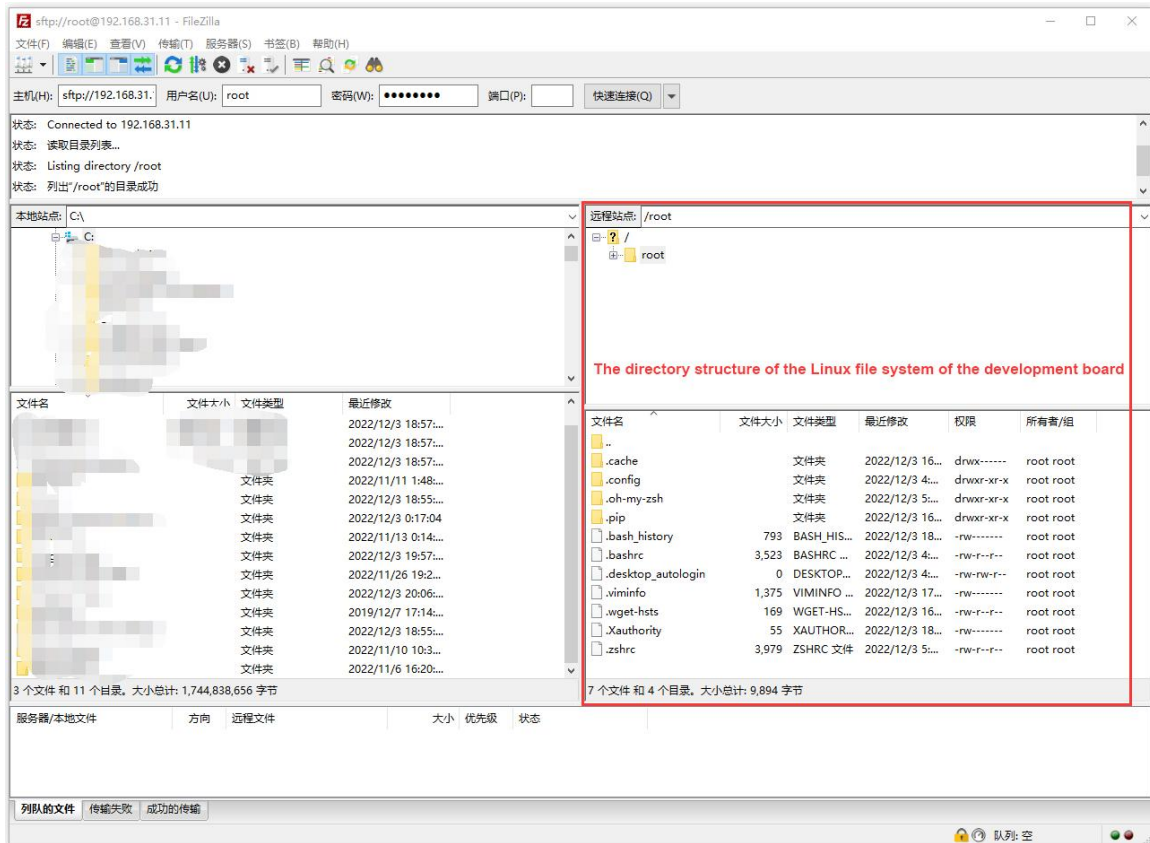
6) Then choose to **save the password**, and then click **OK**



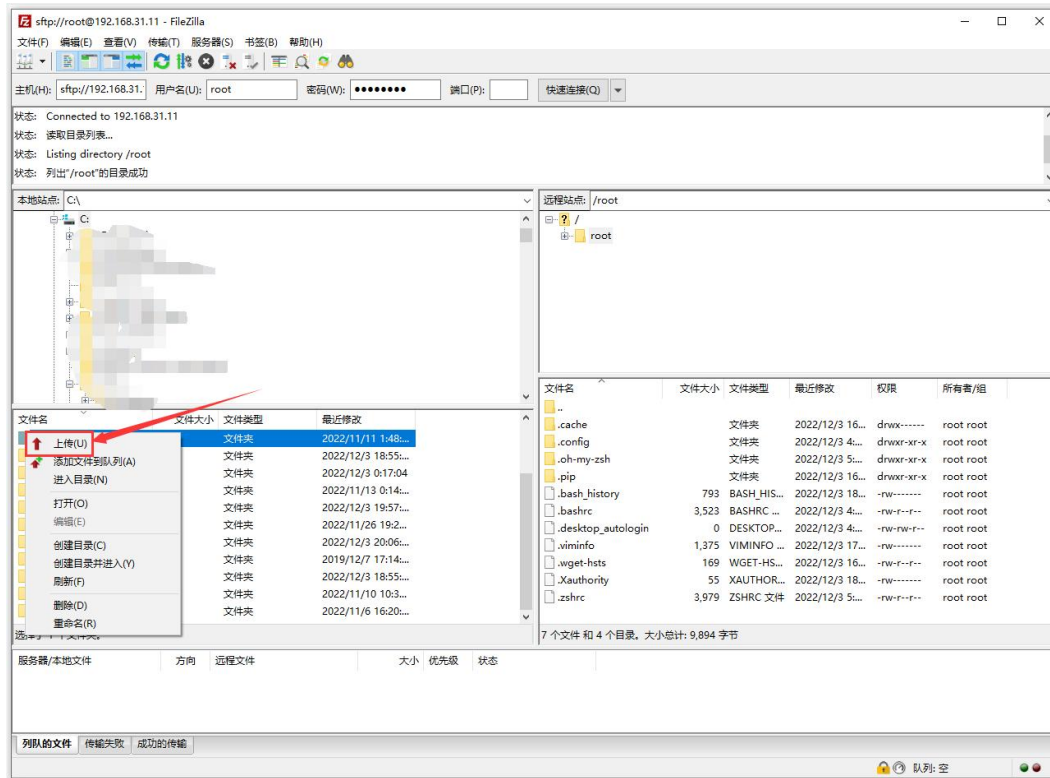
7) Then choose to **always trust this host**, and then click **OK**



8) 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



9) 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



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

11) 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 laptops 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

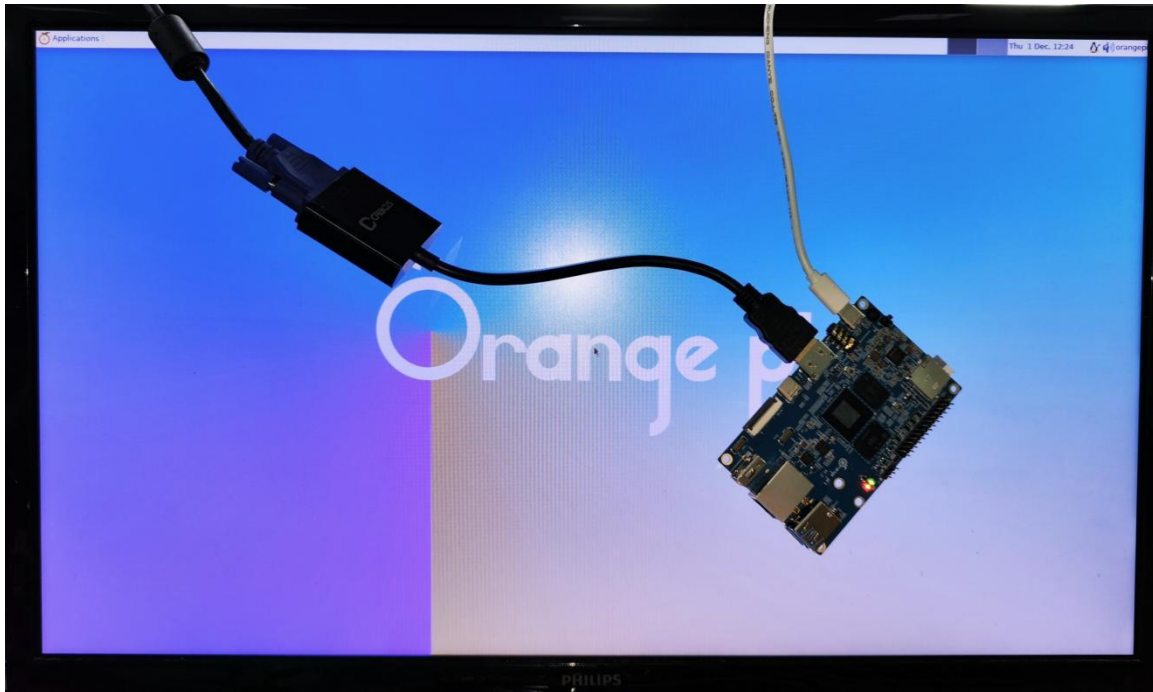


- b. A VGA cable



- c. A monitor or TV that supports VGA interface

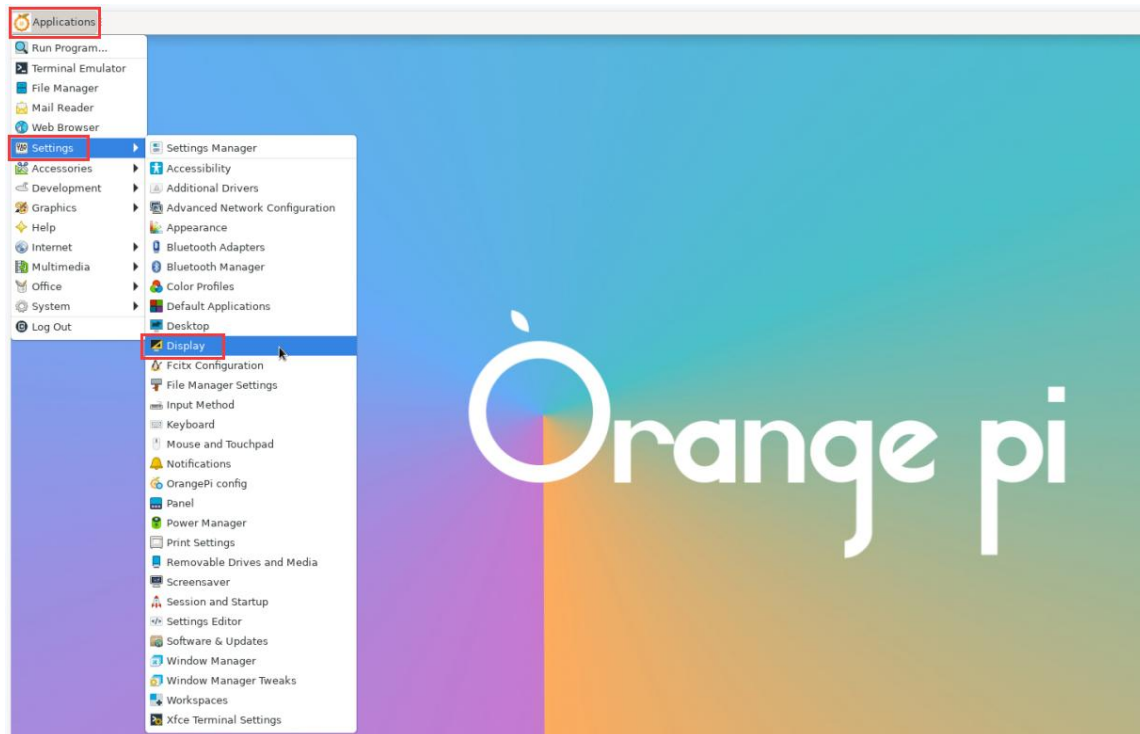
- 2) HDMI to VGA display test as shown below



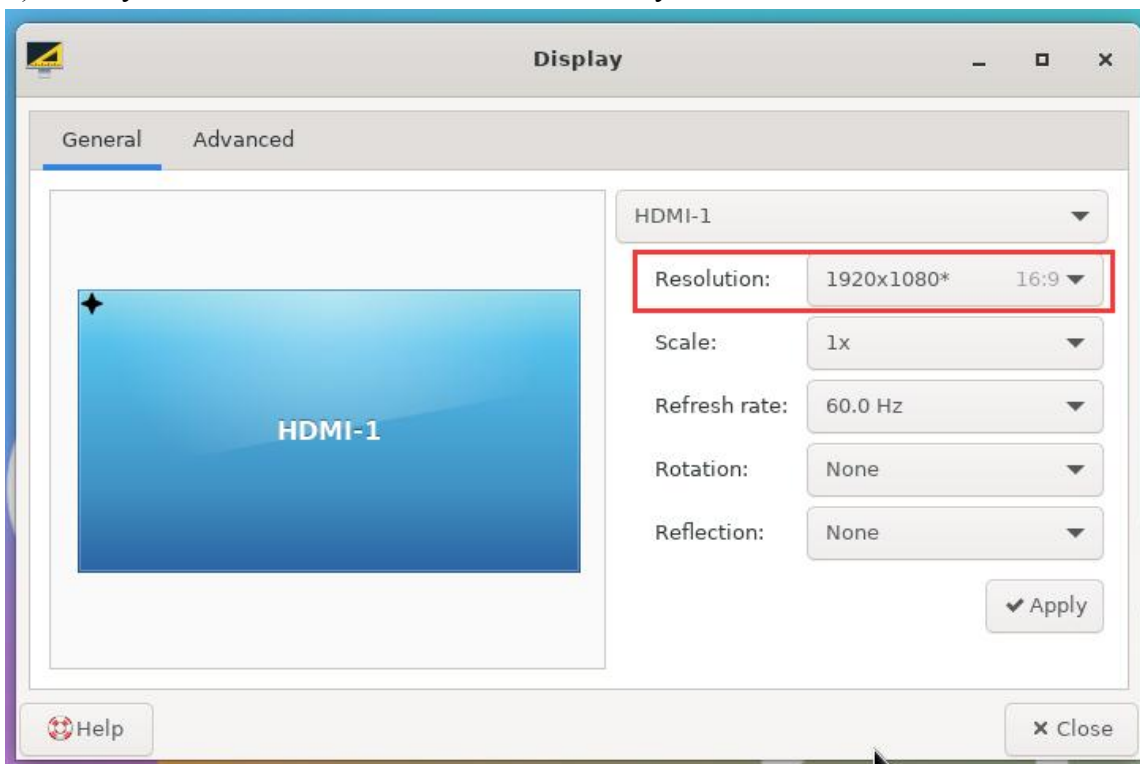
When using HDMI to VGA display, the development board and the Linux system of the development board do not need to make any settings, only the HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI to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

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

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

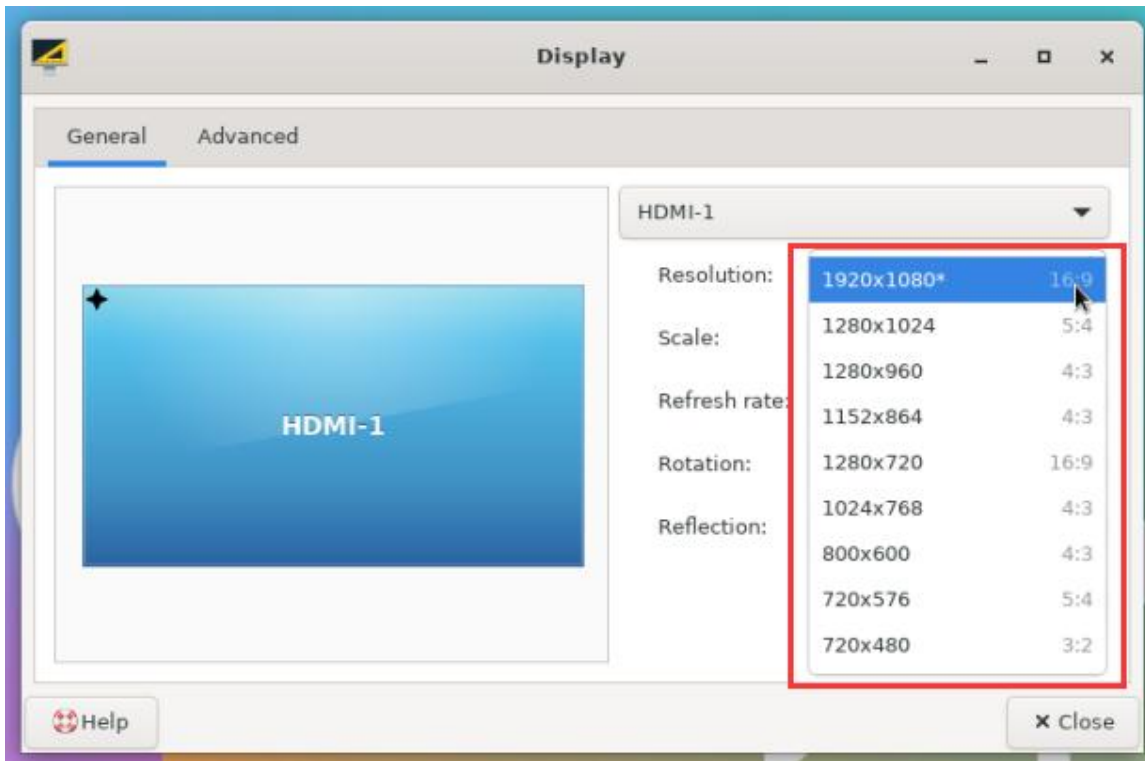


2) Then you can see the current resolution of the system

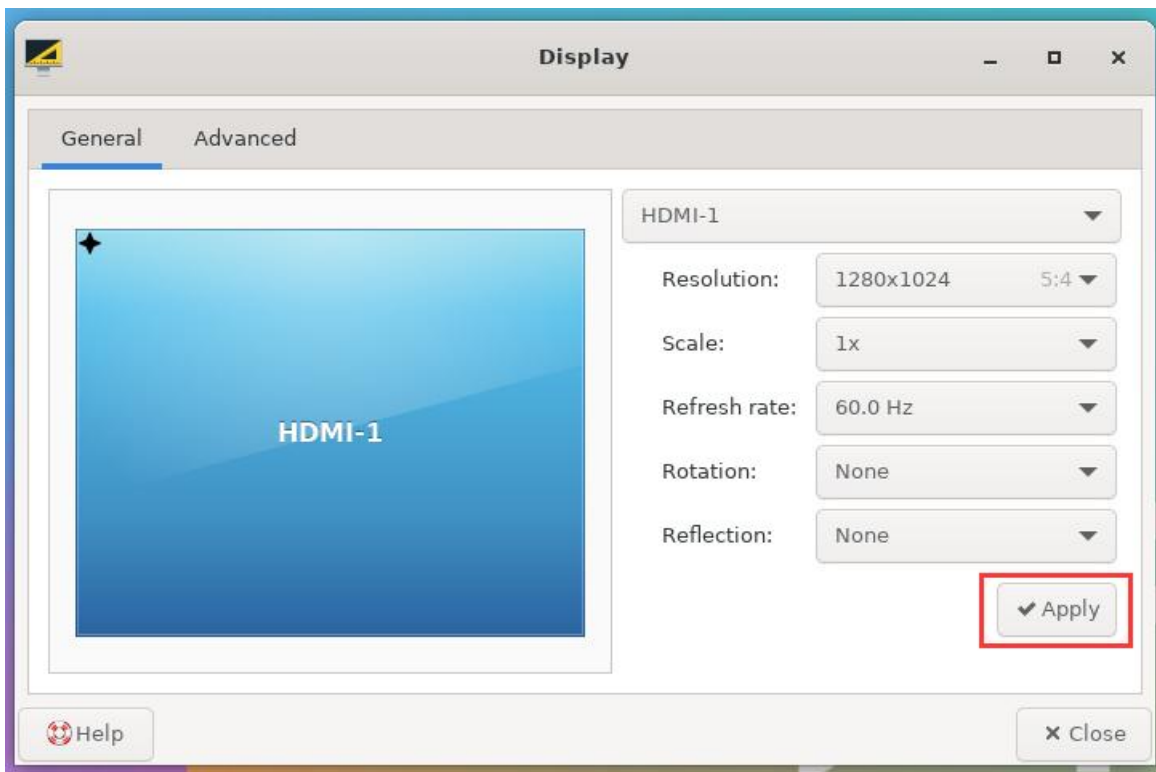


3) Click the drop-down box of Resolution to see all resolutions currently supported by

the display



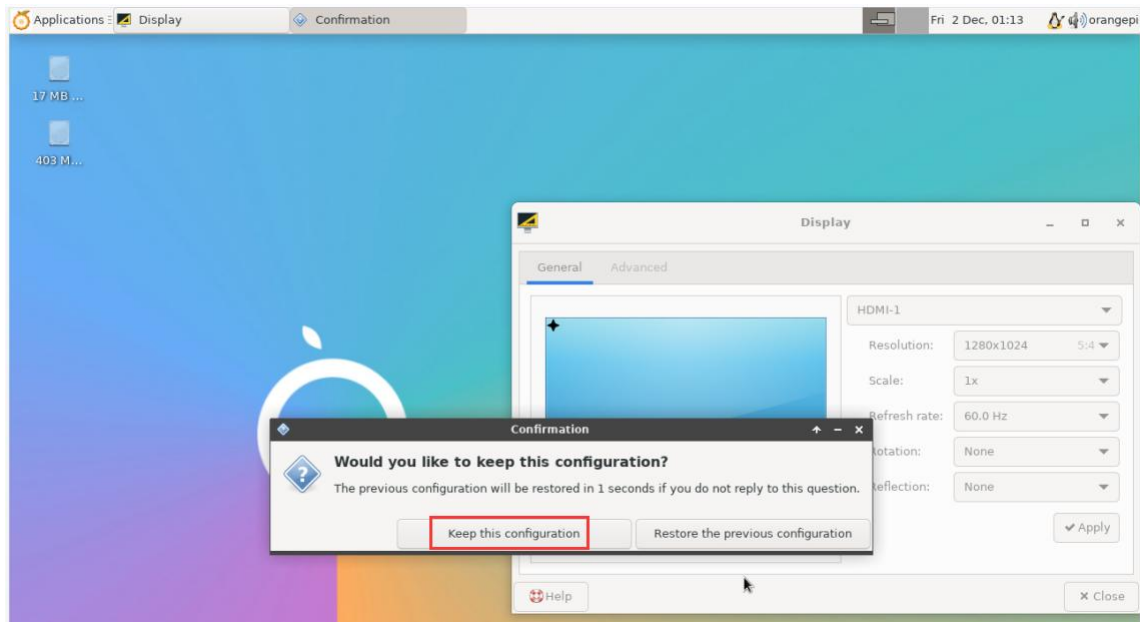
4) Then select the resolution you want to set, and click Apply







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



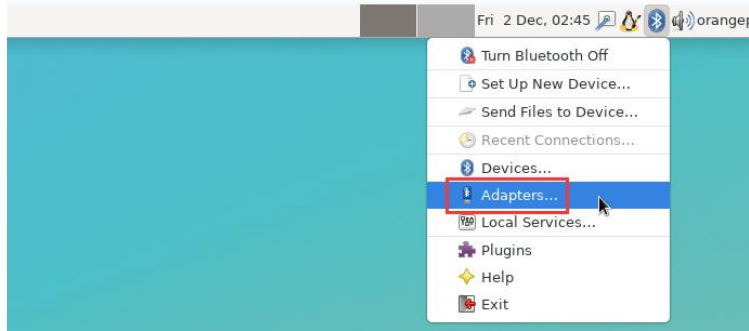
## 3. 11. How to use Bluetooth

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

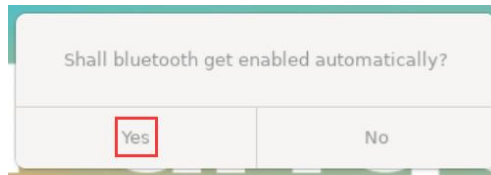
1) Click the Bluetooth icon in the upper right corner of the desktop



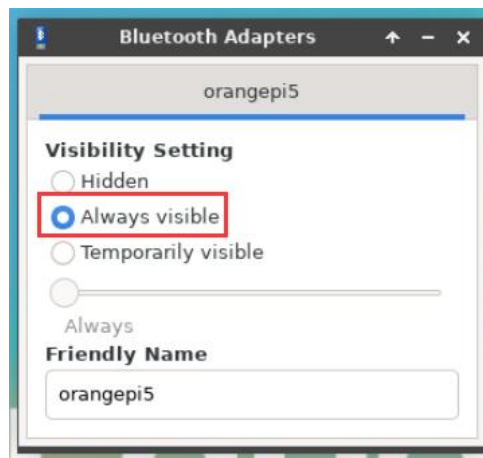
2) Then select the adapter



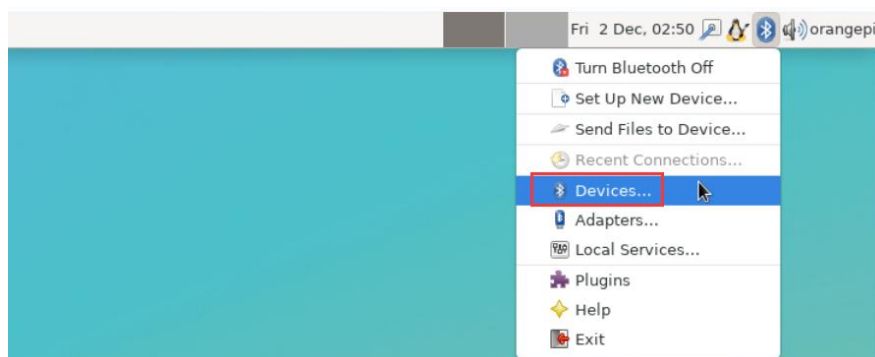
3) If there is a prompt on the following interface, please select **Yes**



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

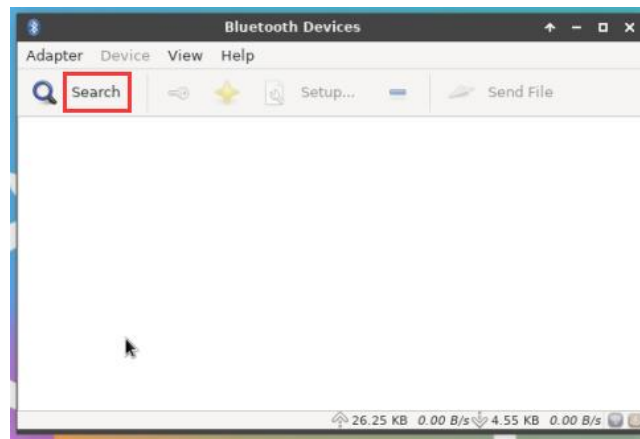


5) Then open the configuration interface of the Bluetooth device

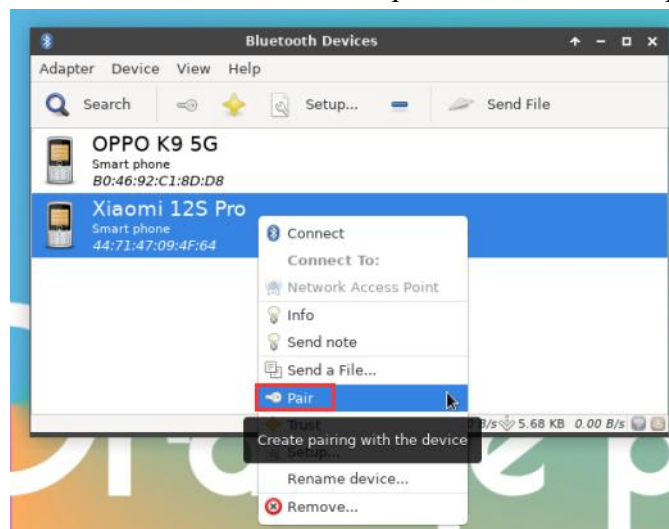




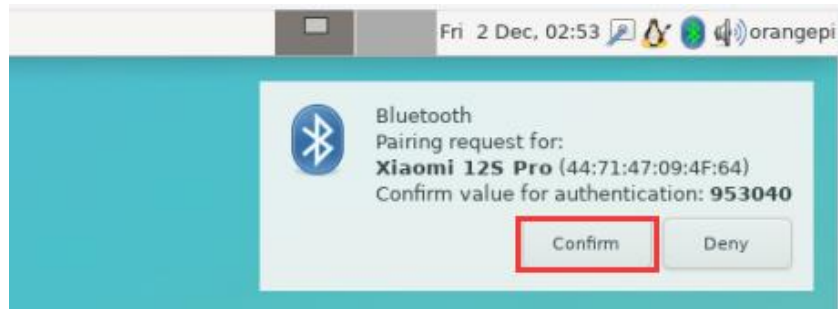
6) Click **Search** to start scanning the surrounding Bluetooth devices



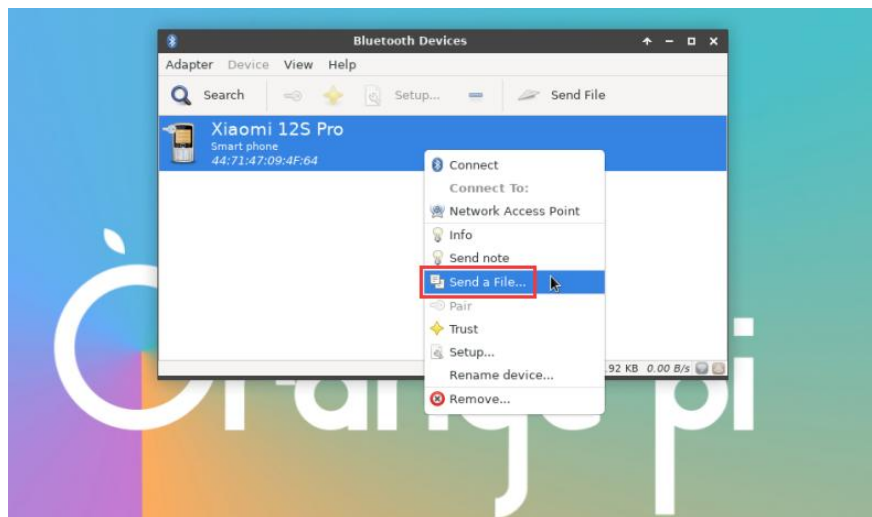
6) Then select the Bluetooth device you want to connect to, and then click the right button of the mouse to pop up the operation interface of the Bluetooth device, select **Pair** to start pairing, and the demonstration here is to pair with an Android phone



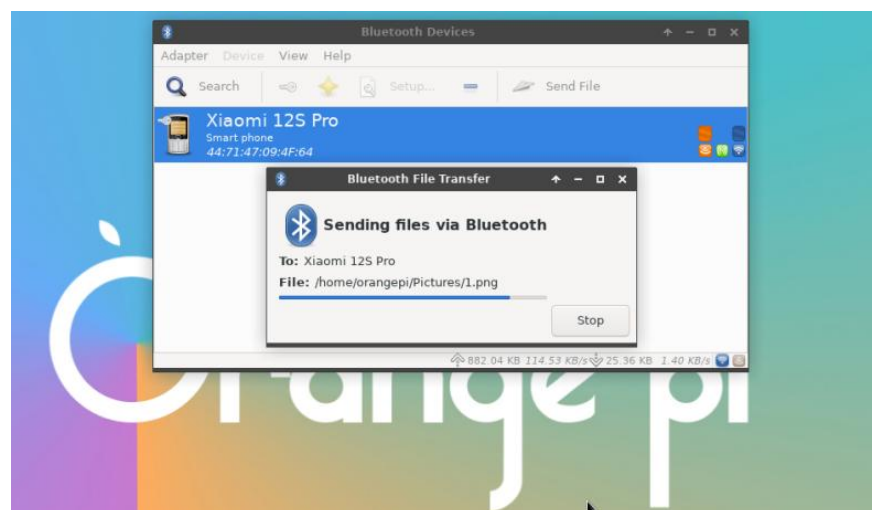
7) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop, just select **Confirm** to confirm, and the phone also needs to confirm at this time



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

```
orangeipi@orangeipi:~$ 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

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

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

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

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

1) First, you need to prepare a USB camera that supports the 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



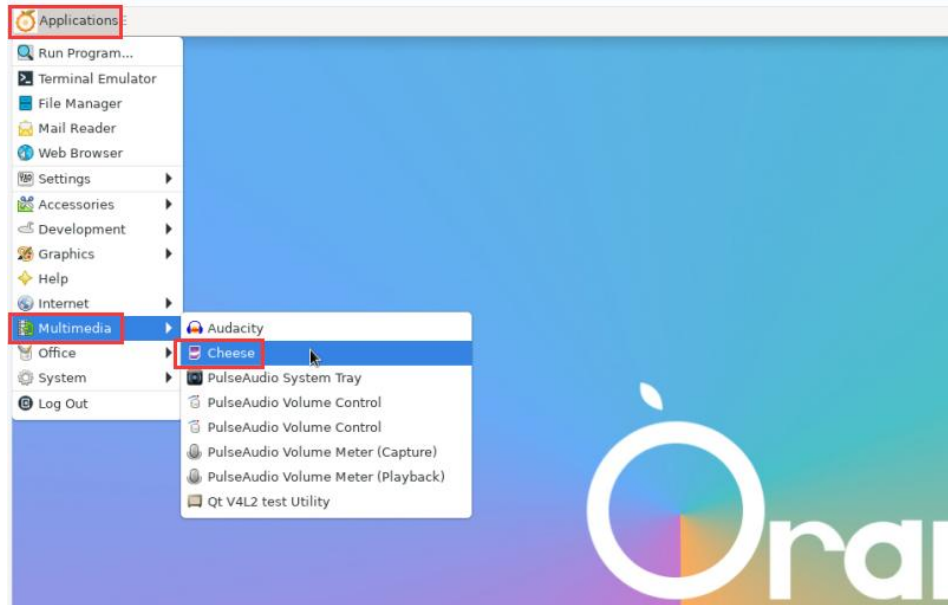
2) Through the v4l2-ctl command, you can see that the device node information of the USB camera is `/dev/video0`

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

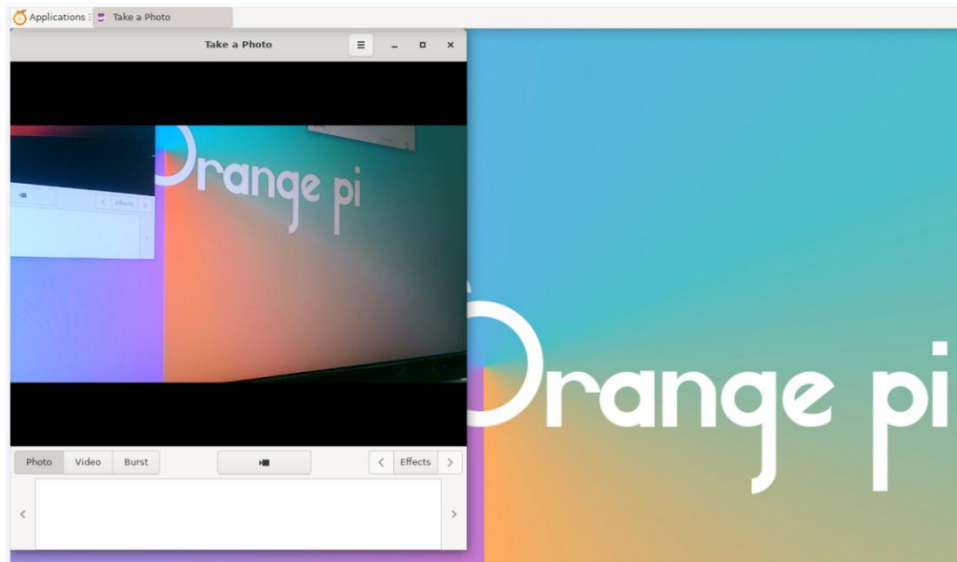
**Note that the l in v4l2 is a lowercase letter l, not the number 1.**

**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

```
orange@orange:~$ sudo apt update
```

```
orange@orange:~$ 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

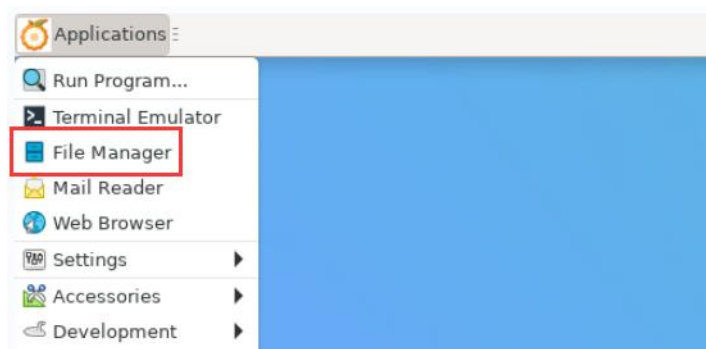
```
orangePi@orangePi:~$ scp image.jpg test@192.168.1.55:/home/test （Modify the IP
address and path according to the actual situation）
```

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

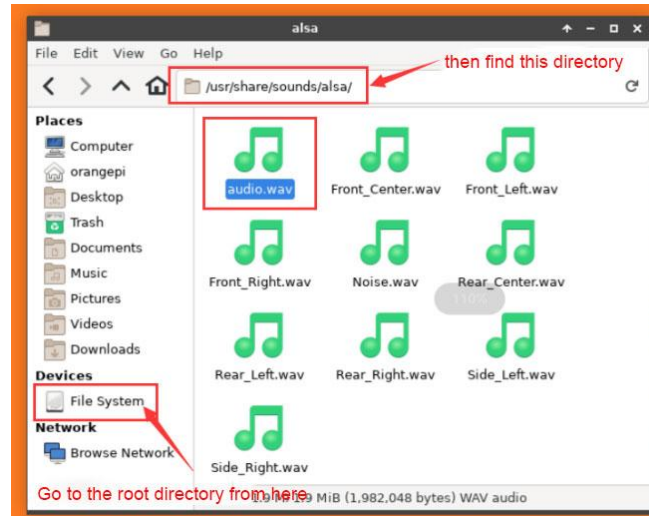
## 3. 13. Audio Test

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

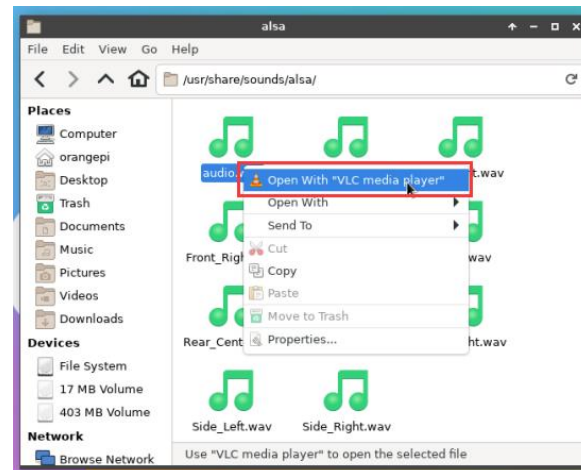
- 1) First open the file manager



- 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)

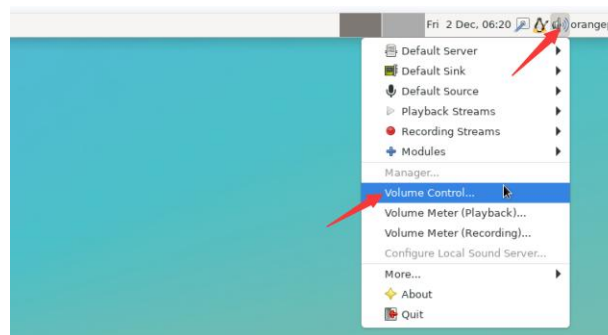


3) Then select the audio.wav file, right click and select open with vlc to start playing



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

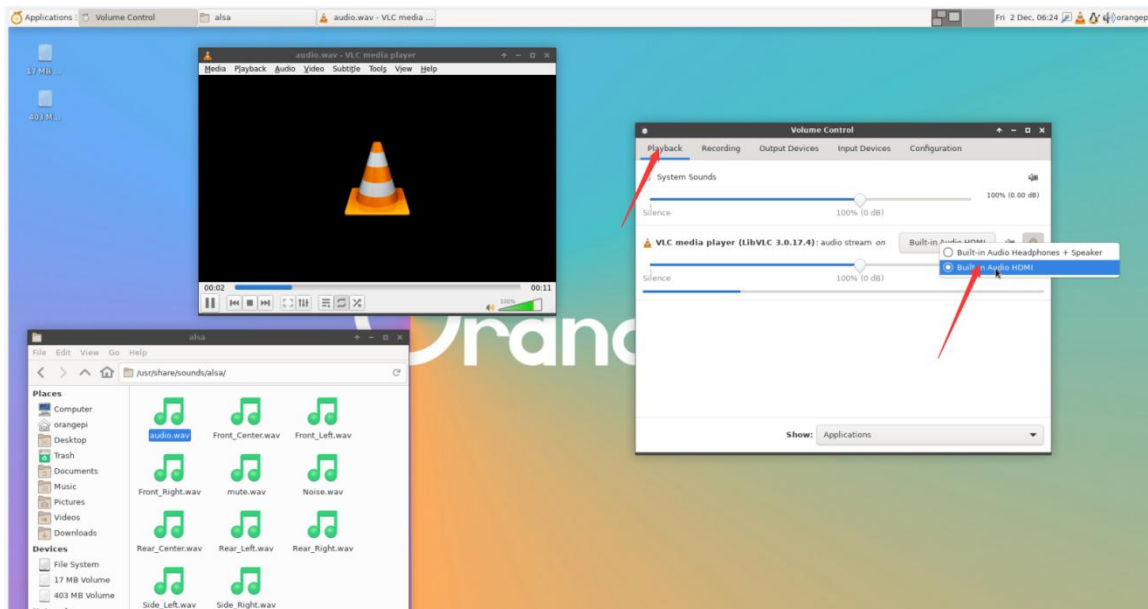
a. First open the volume control interface



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, you 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: rockchip-es8388 [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.

```

orangeypi@orangeypi:~$ 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**

```

orangeypi@orangeypi:~$ 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: rockchip-hdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
Subdevices: 1/1
Subdevice #0: subdevice #0
card 2: rockchip-es8388 [rockchip-es8388], device 0: dailink-multicodecs ES8323.6-0010-0 [dailink-multicodecs ES8323.6-0010-0]
Subdevices: 1/1
Subdevice #0: subdevice #0

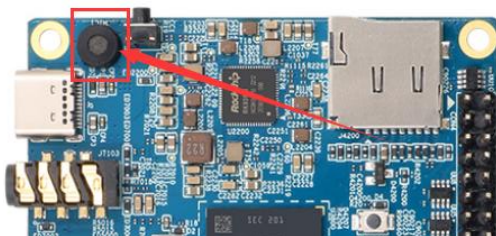
```

- 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.

```
orange@orange:~$ 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.

```
orange@orange:~$ 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.

```
orange@orange:~$ 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. Temperature sensor

The command to view the system temperature sensor is:

```
orange@orange:~$ 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

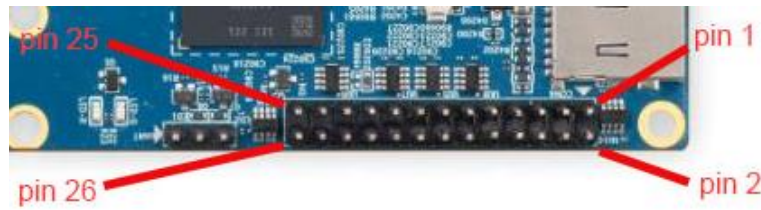
bigcore1_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.2°C

soc_thermal-virtual-0
Adapter: Virtual device
temp1:          +47.2°C (crit = +115.0°C)

```

### 3. 15. 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 5B development board



2) The functions of the 26 pin interface pins of the Orange Pi 5B 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 (feb0010)	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 (feb0030)	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1 (feb0020)	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		PowerKey			
	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 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7
			GND		9
	PWM14_M1 (feb0020)	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		PowerKey			
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.

The 22nd pin in the 26pin is floating and has no purpose by default. But if you solder the lower resistor (100R) on the development board, it can be used as a power-on/off pin (connecting this pin to GND will trigger a shutdown action).



3) There are a total of 16 GPIO ports in the 26pin interface, and the voltage of all GPIO ports is **3.3v**

### 3. 16. 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\\_x.xx.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@orangepi5b:~# gpio readall
```

					PI5B						
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	RXD.0	3	131	
		GND			9	10	0	TXD.0	4	132	
138	5	CAN1_RX	IN	1	11	12	1	CAN2_TX	6	29	
139	7	CAN1_TX	IN	1	13	14		GND			
28	8	CAN2_RX	IN	1	15	16	1	SDA.1	9	59	
		3.3V			17	18	1	SCL.1	10	58	
49	11	SPI4_TXD	IN	1	19	20		GND			
48	12	SPI4_RXD	IN	1	21	22		PowerKey			
50	13	SPI4_CLK	IN	1	23	24	1	SPI4_CS1	14	52	
		GND			25	26	1	PWM1	15	35	

```
root@orangepi5b:~#
```

### 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 5B needs to download the code of the wiringOP next branch, please don't miss the -b next parameter.

If there is a problem downloading the code from GitHub, you can directly use the wiringOP source code that comes with the Linux image. The storage location is: `/usr/src/wiringOP`.

### 2) Compile and install wiringOP

```
orangepi@orangepi:~$ cd wiringOP
orangepi@orangepi:~/wiringOP$ sudo ./build clean
```



```
orange@orange:~/wiringOP$ sudo ./build
```

3) Test the output of the gpio readall command as follows

```
root@orange5b:~# gpio readall
```

					PI5B						
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	RXD.0	3	131	
		GND			9	10	0	TXD.0	4	132	
138	5	CAN1_RX	IN	1	11	12	1	CAN2_TX	6	29	
139	7	CAN1_TX	IN	1	13	14		GND			
28	8	CAN2_RX	IN	1	15	16	1	SDA.1	9	59	
		3.3V			17	18	1	SCL.1	10	58	
49	11	SPI4_TXD	IN	1	19	20		GND			
48	12	SPI4_RXD	IN	1	21	22		PowerKey			
50	13	SPI4_CLK	IN	1	23	24	1	SPI4_CS1	14	52	
		GND			25	26	1	PWM1	15	35	

```
root@orange5b:~#
```

### 3. 17. 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.

```
orange@orange:~$ sudo vim /boot/orangeEnv.txt
```

```
overlays=i2c1-m2 lcd1 ov13850-c1 pwm13-m2 spi4-m0-cs1-spidev uart0-m2
```

#### 3. 17. 1. 26pin GPIO port test

The linux system released by Orange Pi has a pre-installed blink\_all\_gpio program, which will set all 16 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:

```
orange@orange5b:~$ sudo blink_all_gpio #Remember to add sudo permission
```



```
[sudo] password for orangepi:
```

```
#A password is required here
```

1) A total of **16** GPIO ports can be used in the 26pins of the development board. The following uses pin 7—the corresponding GPIO is GPIO1\_C6—the corresponding wPi serial number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepi5b:~# gpio readall
```

GPIO	wPi	Name	Mode	V	Physical	PI5B	V	Mode	Name	wPi	GPIO
47	0	3.3V			1	2			5V		
		SDA.5	IN	1	3	4			5V		
46	1	SCL.5	IN	1	5	6			GND		
54	2	PWM15	IN	1	7	8	0	IN	RXD.0	3	131
		GND			9	10	0	IN	TXD.0	4	132

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@orangepi5b:~# gpio readall
```

GPIO	wPi	Name	Mode	V	Physical	PI5B	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	IN	RXD.0	3	131
		GND			9	10	0	IN	TXD.0	4	132

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





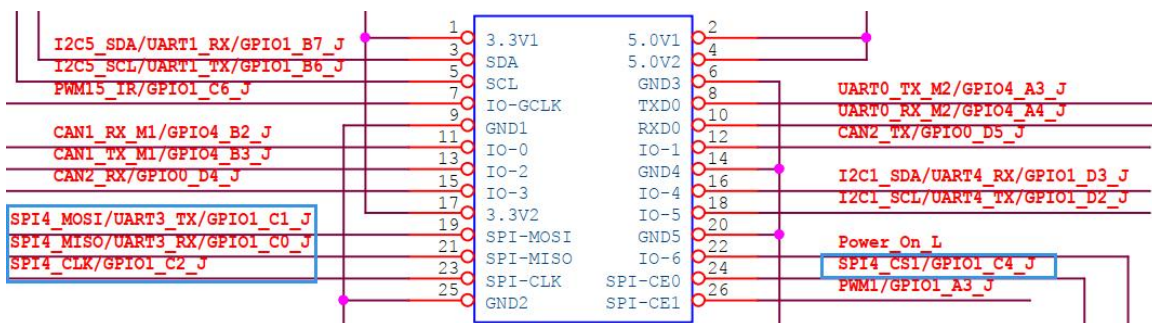
```
root@orangepi5b:~# gpio readall
```

PI5B										
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	RXD.0	3	131
		GND			9	10	0	TXD.0	4	132

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. 17. 2. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the spi available for Orange Pi 5B 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 FF 40 00 00 00 00 95 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 | .....

```

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 FF 40 00 00 00 00 95 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 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF F0 0D | .....@.....

```

### 3. 17. 3. 26pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi 5B 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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	1	2		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	3	4		5V			
			GPIO1_C6	54	5	6		GND			
		PWM15_IR_M2 (feb0030)	GND		7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
	PWM14_M1 (feb0020)	CAN1_RX_M1	GPIO4_B2	138	9	10	132	GPIO4_A4	UART0_RX_M2		
		CAN1_TX_M1	GPIO4_B3	139	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN2_RX_M1	GPIO0_D4	28	13	14		GND			
PWM3_IR_M0 (fd8b0030)	I2C1_SCL_M2		GPIO1_C1	49	15	16	59	GPIO1_D3	UART4_RX_M0 (feb70000)	I2C1_SDA_M4	PWM1_M1 (fd8b0010)
			GPIO1_C2	50	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_C0	48	19	20		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C2	50	21	22		PowerKey			
	PWM3_IR_M2 (fd8b0030)	SPI4_CLK_M0	GND		23	24	52	GPIO1_C4	SPI4_CS1_M0		
					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 26pin (i2c1\_m2), and can also be derived from pins 16 and 18 of 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

```
orangeypi@orangeypi:~$ sudo i2cdetect -y 1      #i2c1 command
orangeypi@orangeypi:~$ sudo i2cdetect -y 3      #i2c3 command
orangeypi@orangeypi:~$ sudo i2cdetect -y 5      #i2c5 command
```

```
orangeypi@orangeypi5:~$ sudo i2cdetect -y 5
[sudo] password for orangeypi:
    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:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
orangeypi@orangeypi5:~$
```

### 3. 17. 4. 26pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5B are four groups of uart buses: uart0, uart1, uart3 and uart4

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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	1	2		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	3	4		5V			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	5	6		GND			
			GND		7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
	PWM14_M1 (feb0020)	CAN1_RX_M1	GPIO4_B2	138	9	10	132	GPIO4_A4	UART0_RX_M2		
		CAN1_TX_M1	GPIO4_B3	139	11	12	29	GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
PWM3_IR_M0 (fd8b0030)	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	13	14		GND			
			3.3V	17	15	16	59	GPIO1_D3	UART4_RX_M0 (feb70000)	I2C1_SDA_M4	PWM1_M1 (fd8b0010)
I2C3_SCL_M0	UART3_TX_M0 (feb60000)	SPI4_MOSI_M0	GPIO1_C1	49	17	18	58	GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1 (fd8b0000)
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	19	20		GND			
	PWM3_IR_M2 (fd8b0030)	SPI4_CLK_M0	GPIO1_C2	50	21	22		PowerKey			
			GND	25	23	24	52	GPIO1_C4	SPI4_CS1_M0		
							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.

```
orangeypi@orangeypi:~$ 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. 17. 5. How to test PWM using /sys/class/pwm

1) As can be seen from the table below, the available pwm for Orange Pi 5B 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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	1	2		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	3	4		GND			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	5	6					
			GND		7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
					9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1 (feb0020)	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)
					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		PowerKey			
	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 shown below, please do not open pwm1-m1 and pwm1-m2 at the same time:

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt  
overlays=pwm1-m1
```

Select the settings for pwm1\_m2 as shown below:

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt  
overlays=pwm1-m2
```

Select the settings of pwm3\_m0 as shown below, please do not open pwm3-m0 and pwm3-m2 at the same time:

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt  
overlays=pwm3-m0
```

Select the settings for pwm3\_m2 as shown below:

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
```

**overlays=pwm3-m2**

2) After opening a pwm, there will be an extra pwmchipX in `/sys/class/pwm/` (X is a specific number), for example, after opening pwm15, check the pwmchipX under `/sys/class/pwm/` two became three

```
orange@orange:~$ 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:

```
orange@orange:~$ 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
orange@orange:~$
```

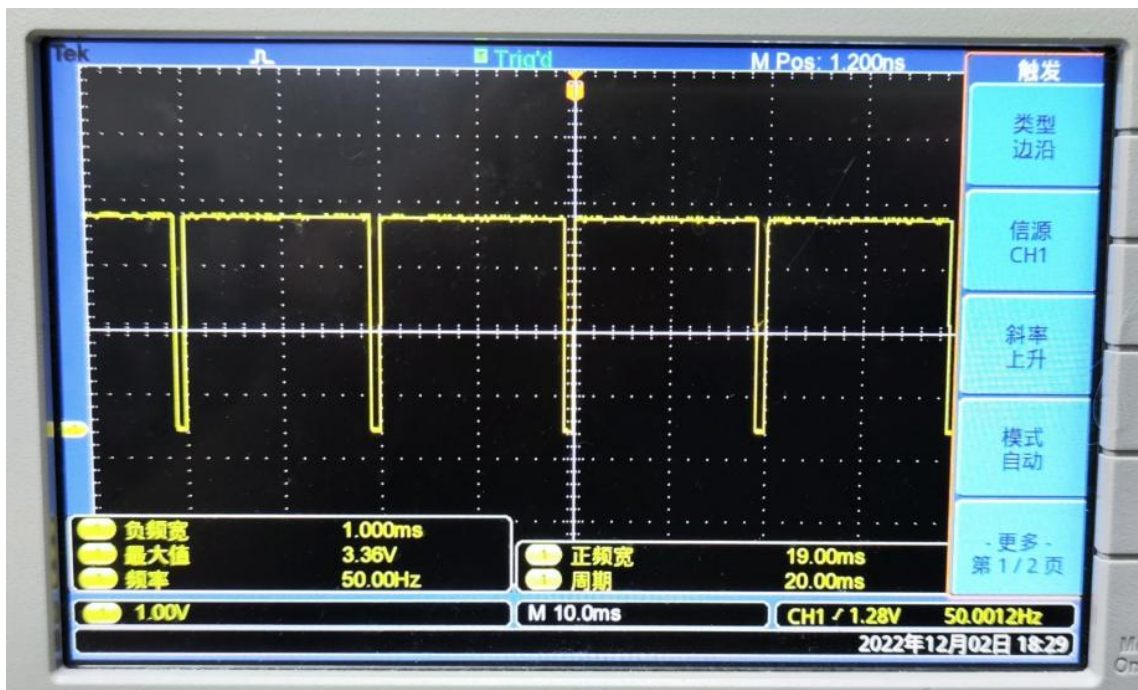
4) Then it can be known from the table below that the base address of the pwm15 register is febf0030, and then look at the output of the `ls /sys/class/pwm/ -l` command, you can see that pwmchip2 is linked to febf0030.pwm, so pwm15 corresponds to pwmchip as pwmchip2

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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		5V			
		PWM15_IR_M2 (febf0030)	GPIO1_C6	54	7	8	131	GND			
			GND		9	10	132	GPIO4_A3	UART0_TX_M2 (fd890000)		
	PWM14_M1 (feb0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO4_A4	UART0_RX_M2		
		CAN1_TX_M1	GPIO4_B3	139	13	14		GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
PWM3_IR_M0 (fd8b0030)	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GND			
			3.3V		17	18	58	GPIO1_D3	UART4_RX_M0 (feb70000)	I2C1_SDA_M4	PWM1_M1 (fd8b0010)
			GPIO1_C1	49	19	20		GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1 (fd8b0000)
I2C3_SCL_M0	UART3_TX_M0 (feb60000)	SPI4_MOSI_M0	GPIO1_C0	48	21	22		GND			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C2	50	23	24	52	PowerKey			
	PWM3_IR_M2 (fd8b0030)	SPI4_CLK_M0			25	26	35	GPIO1_C4	SPI4_CS1_M0		
			GND					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@orange:~# echo 0 > /sys/class/pwm/pwmchip2/export
root@orange:~# echo 20000000 > /sys/class/pwm/pwmchip2/pwm0/period
root@orange:~# echo 1000000 > /sys/class/pwm/pwmchip2/pwm0/duty_cycle
root@orange:~# echo 1 > /sys/class/pwm/pwmchip2/pwm0/enable
```





6) The test method of pwm15 demonstrated above is similar to other pwm test methods.

### 3. 17. 6. CAN test method

Please note that Linux 6.1 systems currently do not support CAN function.

#### 3. 17. 6. 1. How to open CAN

1) As can be seen from the table below, the available CAN buses for Orange Pi 5B are CAN1 and CAN2, a total of two CAN 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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		5V			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7	8	131	GND	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A3	UART0_RX_M2		
	PWM14_M1 (feb0020)	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		PowerKey			
	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 Linux systems, CAN in 26pin is turned off by default and needs to be turned on manually before it can be used.

Add the configuration in red font below to `/boot/orangepiEnv.txt`, and then restart



**the Linux system to open CAN1 and CAN2 at the same time. If you only need to open one, just fill in one.**

```
orangePi@orangePi:~$ sudo vim /boot/orangepiEnv.txt
```

```
overlays=can1-m1 can2-m1
```

2) After entering the Linux system, use the **sudo ifconfig -a** command. If you can see the CAN device, it means that CAN has been opened correctly. 入 linux

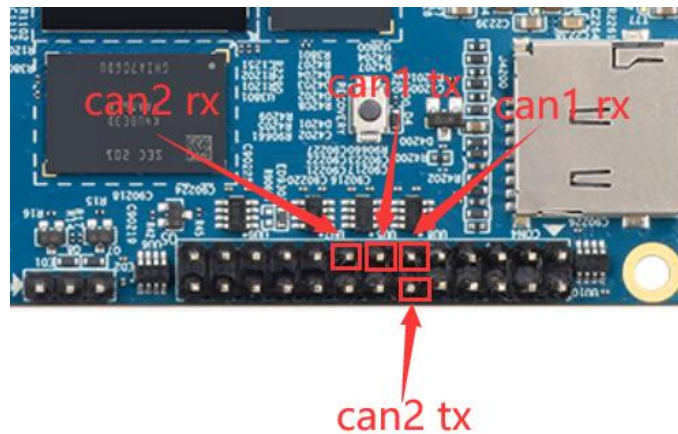
```
orangePi@orangePi:~$ sudo ifconfig -a
can0: flags=128<NOARP>  mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
    device interrupt 91

can1: flags=128<NOARP>  mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
    device interrupt 92
```

3) The corresponding pins of CAN1 and CAN2 are

	CAN1	CAN2
TX pin	Corresponds to pin 13	Corresponds to pin 12
RX pin	Corresponds to pin 11	Corresponds to pin 15





### 3. 17. 6. 2. Use CANalyst-II analyzer to test sending and receiving messages

1) The CANalyst-II analyzer used in the test is shown in the figure below



2) CANalyst-II analyzer data download link

<https://www.zhcxgd.com/3.html>

3) First install the USBCANToolSetup software



4) The shortcut after installation of USBCANToolSetup is



5) You also need to install the USB driver

桌面 > images > CAN分析仪资料20200101_顶配 > 硬件驱动程序				
名称	修改日期	类型	大小	
3.USB驱动的安装与卸载说明书	2020/1/1 19:01	Foxit PDF Reade...	602 KB	
USB驱动安装工具Setup(V1.40)	2020/1/1 18:42	应用程序	14,935 KB	
硬件驱动程序(手动安装)	2019/5/22 16:16	360压缩 RAR 文件	11,023 KB	

6) The USB interface of the CANalyst-II analyzer needs to be connected to the USB interface of the computer.



7) To test the CAN function, you also need to prepare a CAN transceiver as shown in the figure below. The main function of the CAN transceiver is to convert the TTL signal of the CAN controller into the differential signal of the CAN bus.

- The 3.3V pin of the CAN transceiver needs to be connected to the 3.3V pin in the 26pin of the development board.
- The GND pin of the CAN transceiver needs to be connected to the GND pin on the 26pin of the development board.
- The CAN TX pin of the CAN transceiver needs to be connected to the TX pin of the CAN bus in the 26-pin development board
- The CAN RX pin of the CAN transceiver needs to be connected to the RX pin of the CAN bus in the 26-pin development board.
- The CANL pin of the CAN transceiver needs to be connected to the H interface of the analyzer
- The CANL pin of the CAN transceiver needs to be connected to the L interface



of the analyzer



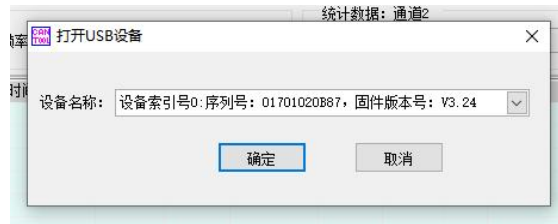
8) Then you can open the USB-CAN software



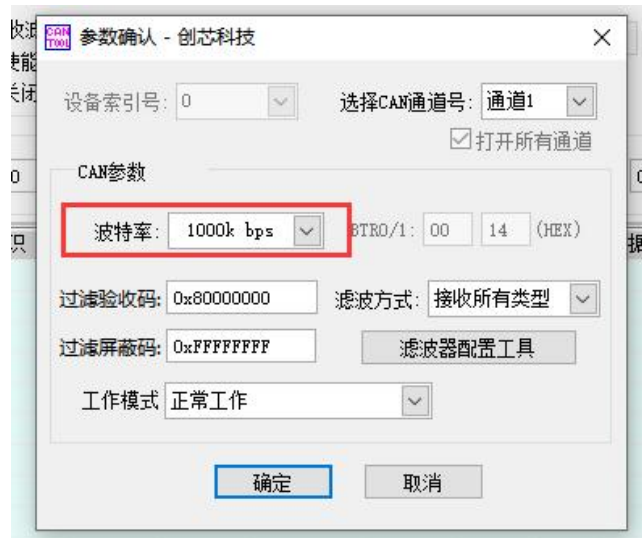
9) Then click Start Device



10) Then click OK



11) Set the baud rate to 1000k bps



12) After successful opening, the USB-CAN software will display the serial number and other information.



13) Development board receives CAN message test

a. First set the baud rate of the CAN bus to **1000kbps** in the Linux system of the



development board.

```
orangepi@orangepi:~$ sudo ip link set can0 down
orangepi@orangepi:~$ sudo ip link set can0 type can bitrate 1000000
orangepi@orangepi:~$ sudo ip link set can0 up
```

b. Then run the **candump can0** command to prepare to receive messages

```
orangepi@orangepi:~$ sudo candump can0
```

c. Then send a message to the development board in the USB-CAN software



d. If the development board can receive messages sent by the analyzer, it means that the CAN bus can be used normally.

```
orangepi@orangepi:~$ sudo candump can0
can0 001 [8] 01 02 03 04 05 06 07 08
```

#### 14) Development board sends CAN message test

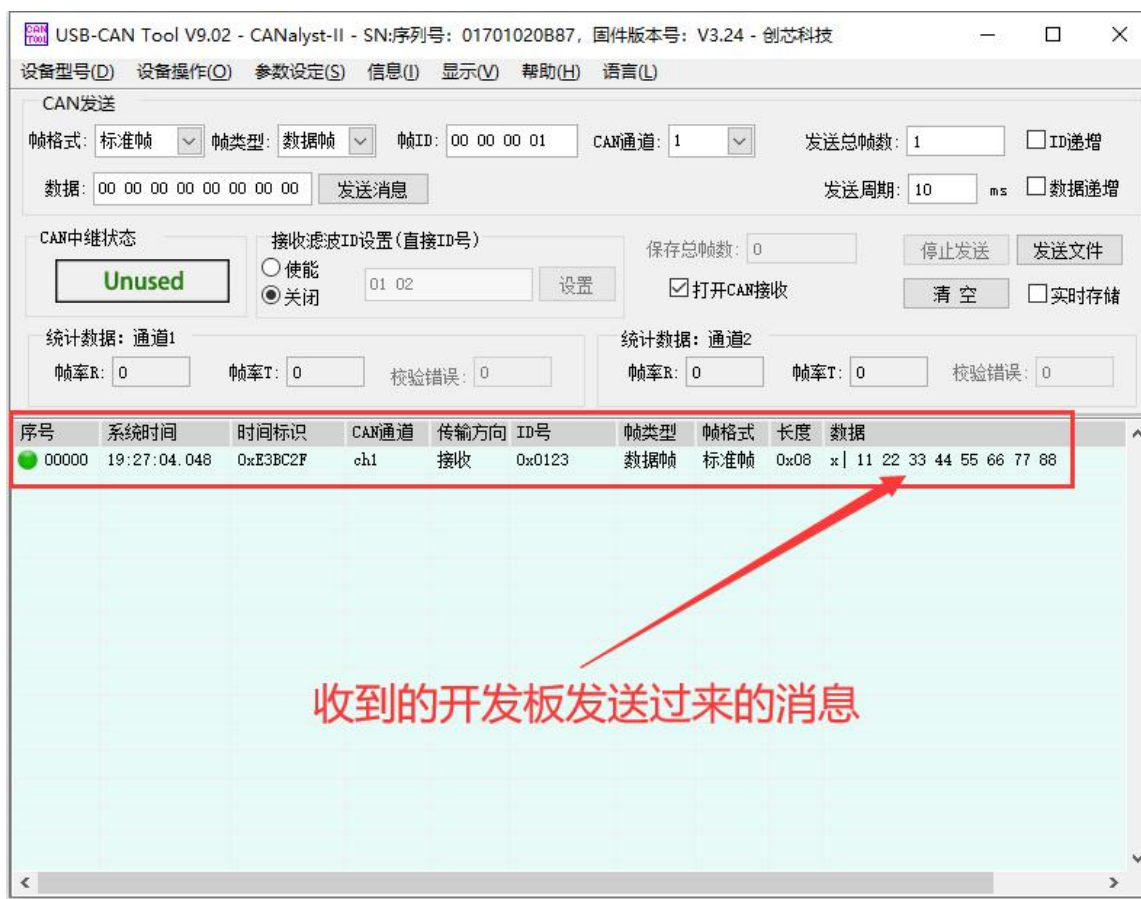
a. First set the CAN baud rate to **1000kbps** in the Linux system

```
orangepi@orangepi:~$ sudo ip link set can0 down
orangepi@orangepi:~$ sudo ip link set can0 type can bitrate 1000000
orangepi@orangepi:~$ sudo ip link set can0 up
```

b. Execute the **cansend** command in the development board to send a message

```
orangepi@orangepi:~$ sudo cansend can0 123#1122334455667788
```

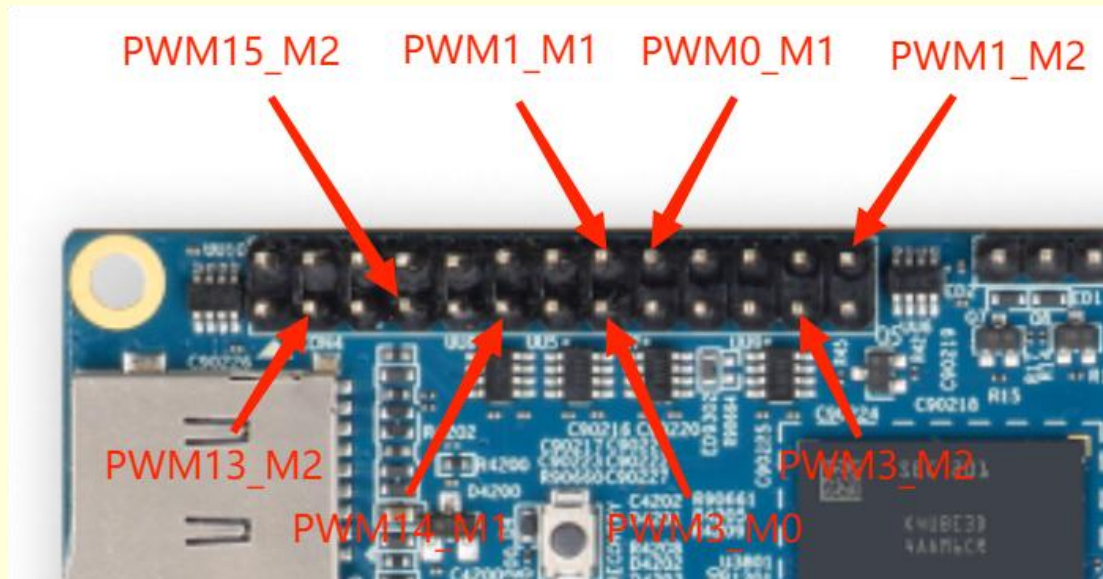
c. If the USB-CAN software can receive the message from the development board, the communication is successful.



### 3. 18. How to use wiringOP hardware PWM

The development board can use a total of 6 PWM channels: PWM0, PWM1, PWM3, PWM13, PWM14 and PWM15. The locations of their pins are as shown in the figure below:





What needs to be noted is: PWM1\_M1 和 PWM1\_M2、  
PWM3\_M0 和 PWM3\_M2

They are all the same PWM, just connected to different pins. If both pins are configured as PWM functions, when one of the PWM pins is set, the same setting will be made to the other PWM pin.

### 3. 18. 1. How to set PWM using wiringOP' s gpio command

#### 3. 18. 1. 1. Set the corresponding pin to PWM mode

1) The corresponding relationship between the 8 PWM pins and the wPi serial number is as shown in the following table:

PWM pin	wPi serial number
PWM0_M1	10
PWM1_M1	9
PWM1_M2	16
PWM3_M0	8
PWM3_M2	14
PWM13_M2	0
PWM14_M1	5
PWM15_M2	2





2) The command to set the pin to PWM mode is as follows, taking PWM0\_M1 as an example. The third parameter needs to enter the wPi serial number corresponding to the PWM0\_M1 pin.

```
orange@orange:~$ gpio mode 10 pwm
```

3) After the pin is set to PWM mode, it will output a square wave with a frequency of 200Hz and a duty cycle of 50% by default. At this time, we use an oscilloscope to measure the corresponding PWM pin and you can see the following waveform.



### 3. 18. 1. 2. Method of adjusting PWM duty cycle

1) The calculation formula of PWM duty cycle is as follows. We can adjust the PWM duty cycle by setting the values of CCR and ARR.

$$\text{PWM duty cycle} = \text{CCR} / \text{ARR}$$

in:

The value range of CCR is 0~65535, and the default value is 500.

The value range of ARR is 0~65535, and the default value is 1000.

It should be noted that our CCR value needs to be smaller than the ARR value because the duty cycle cannot be greater than 1.



When setting  $CCR > ARR$ , the following error message will be prompted:

**gpio: CCR should be less than or equal to ARR (XXX)**

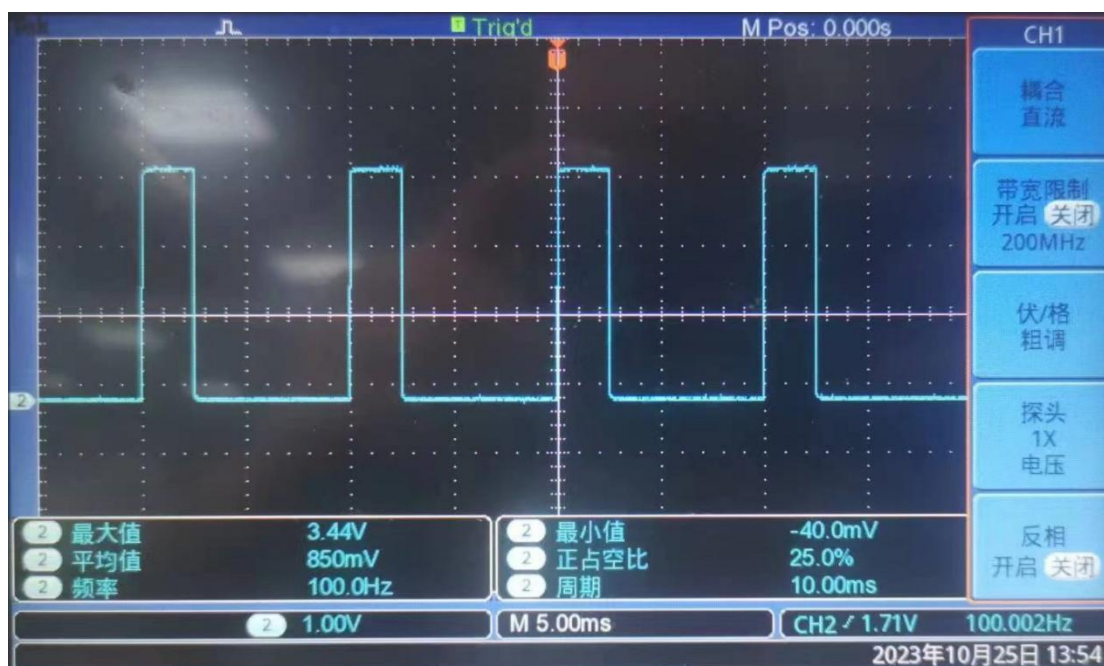
When setting  $ARR < CCR$ , the following error message will be prompted:

**gpio: ARR should be greater than or equal to CRR (XXX)**

2) We can use the following command to set the ARR of the PWM0\_M1 pin to 2000

```
orange@orange:~$ gpio pwmr 10 2000
```

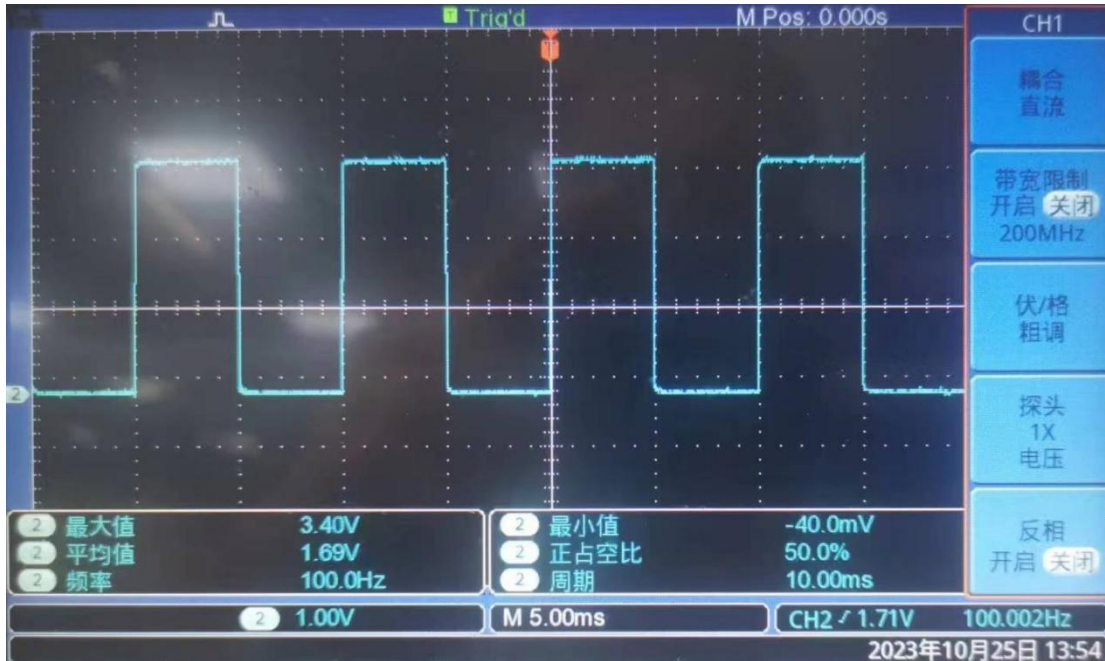
3) After running the above command, you can observe through the oscilloscope that the PWM duty cycle changes from the default 50% (500/1000) to 25% (500/2000)



4) We can use the following command to set the CCR of the PWM0\_M1 pin to 1000

```
orange@orange:~$ gpio pwm 10 1000
```

5) After running the above command, you can observe through the oscilloscope that the PWM duty cycle changes from 25% (500/2000) to 50% (1000/2000)



### 3. 18. 1. 3. Method of adjusting PWM frequency

The calculation formula of PWM frequency is as follows:

$$\text{PWM frequency} = \text{clock source frequency} / (\text{frequency division factor} * \text{ARR})$$

in:

The default value of the clock source frequency is 24000000Hz.

The value range of the frequency division coefficient is 2 ~ 512, and the default value is 120.

The value range of ARR is 0 ~ 65535, and the default value is 1000.

The default value of PWM frequency is  $24000000 / (120 * 1000) = 200\text{Hz}$ .

It should be noted that if the frequency division coefficient is set to an odd number, the actual frequency division coefficient is the set value minus one.

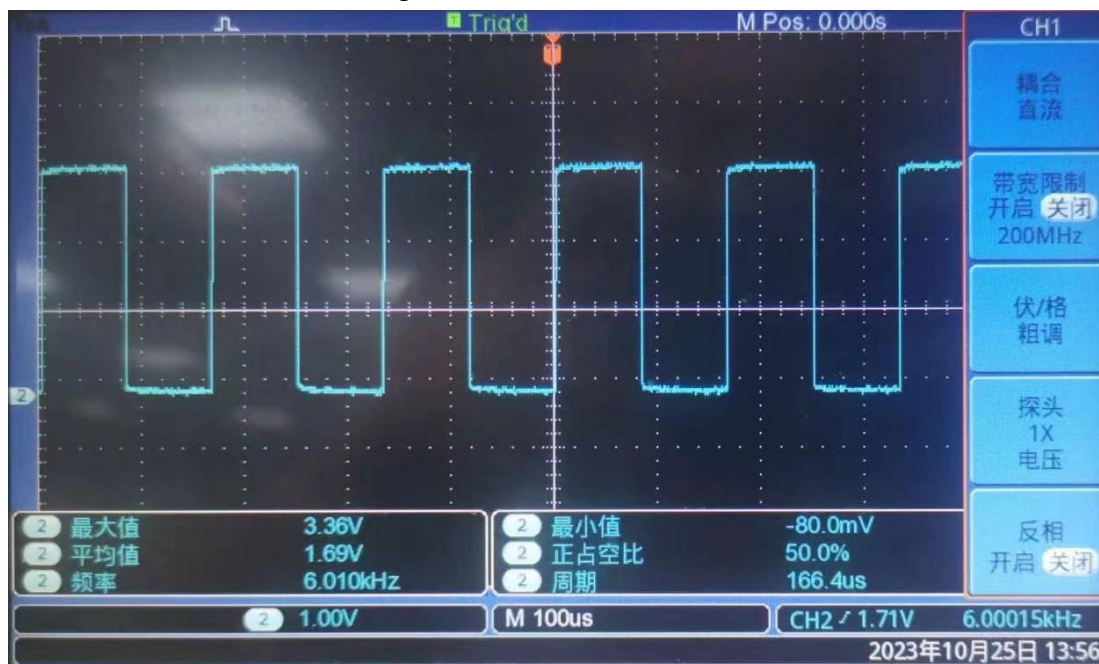
#### 3. 18. 1. 3. 1. Method to adjust PWM frequency by setting frequency division coefficient

1) We can use the following command to set the frequency division factor of the PWM0\_M1 pin to 4



```
orange@orange:~$ gpio pwm 10 4
```

2) According to the above formula, the calculated value of PWM frequency is 6000Hz. It can be observed through the oscilloscope that the measured value of PWM frequency is 6010Hz, and the error can be ignored.



### 3. 18. 1. 3. 2. Method of directly setting the PWM frequency

1) We can use the `gpio pwmTone` command to set the frequency of the PWM pin. For example, use the following command to set the PWM frequency of the PWM0\_M1 pin to 500Hz.

```
orange@orange:~$ gpio pwmTone 10 500
```

**When setting the PWM frequency, you need to ensure:**

**The set frequency value is  $< 24000000 / (\text{frequency division factor} * 2)$ .**

**For example, the default frequency division coefficient is 120. If the frequency division coefficient is not modified, the set frequency value should be less than 100000Hz.**

**If the setting value is too large, the following error will appear:**





**gpio: The PWM frequency you set is too high to be possible**

- 2) Then you can observe through the oscilloscope that the PWM frequency becomes 500Hz



### 3. 18. 2. How to use the PWM test program

- 1) In the example directory of wiringOP, there is a program named pwm.c. This program demonstrates how to operate PWM using the PWM-related API in wiringOP.

```
orangePi@orangePi:~$ cd /usr/src/wiringOP/examples/
orangePi@orangePi:/usr/src/wiringOP/examples$ ls pwm.c
pwm.c
```

- 2) The command to compile **pwm.c** into an executable program is as follows

```
sorangePi@orangePi:/usr/src/wiringOP/examples$ gcc -o pwm pwm.c -lwiringPi
```

- 3) Then you can execute the PWM test program. When executing the PWM test program, you need to specify the PWM pin. For example, you can use the following command to test the PWM0\_M1 pin:

```
sorangePi@orangePi:/usr/src/wiringOP/examples$ sudo ./pwm 10
```

- 4) After the pwm program is executed, the following contents will be tested in sequence:
- a. Adjust the PWM duty cycle by setting ARR



- b. Adjust the PWM duty cycle by setting CCR
- c. Adjust the PWM frequency by setting the frequency division coefficient
- d. Directly set the PWM frequency

5) After each test is completed, the PWM waveform output will be stopped for 5 seconds. After all test contents are completed, a new round of testing will be restarted.

6) The detailed execution process of the PWM test program is as follows:

- a. Adjust the PWM duty cycle by setting ARR: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 50% to 25%, maintains it for 5 seconds, and then PWM The waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 25% to 50% and remains for 5 seconds.
- b. Adjust the PWM duty cycle by setting CCR: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 50% to 100%, maintains it for 5 seconds, and then PWM The waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 100% to 50% and remains for 5 seconds.
- c. Adjust the PWM frequency by setting the frequency division coefficient: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 9 times, the PWM frequency becomes 1/10 of the default PWM frequency, which is 2347Hz, and remains for 5 seconds. , and then the PWM waveform changes every 0.5 seconds. After changing 9 times, the PWM frequency changes to the default PWM frequency, which is 23475Hz, and remains for 5 seconds.
- d. Directly set the PWM frequency: It can be observed through the oscilloscope that the PWM frequency first changes to 2000Hz, and then the PWM frequency increases by 2000Hz every two seconds. After changing 9 times, the PWM frequency changes to 20000Hz and remains for 5 seconds.

### 3. 19. How to install and use wiringOP-Python

**wiringOP-Python is the Python language version of wiringOP, which is used to**



operate the hardware resources of the development board, such as GPIO, I2C, SPI and UART, in the Python program.

In addition, please note that all the following commands are operated under the **root** user.

### 3. 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 there is a problem downloading the code from GitHub, you can directly use the wiringOP-Python source code that comes with the Linux image. The storage location is: `/usr/src/wiringOP-Python`.

```
root@orangepi:~# git clone --recursive https://github.com/orangepi-xunlong/wiringOP-Python -b next
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# git submodule update --init --remote
```

3) Then use the following command to compile wiringOP-Python and install it into the Linux system of the development board

```
root@orangepi:~# cd wiringOP-Python
root@orangepi:~/wiringOP-Python# python3 generate-bindings.py > bindings.i
root@orangepi:~/wiringOP-Python# sudo python3 setup.py install
```

4) Then enter the following command, if there is help information output, it means that wiringOP-Python is installed successfully, press the **q** key to exit the help information interface

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; help(wiringpi)"
Help on module wiringpi:
```



**NAME**

wiringpi

**DESCRIPTION**

```
# This file was automatically generated by SWIG (http://www.swig.org).
# Version 4.0.2
#
# Do not make changes to this file unless you know what you are doing--modify
# the SWIG interface file instead.
```

5) The steps to test whether wiringOP-Python is successfully installed under the python command line are as follows:

- a. First use the python3 command to enter the command line mode of python3

```
root@orangepi:~# python3
```

- b. Then import the python module of wiringpi

```
>>> import wiringpi;
```

- c. Finally, enter the following command to view the help information of wiringOP-Python, and press the **q** key to exit the help information interface

```
>>> help(wiringpi)
```

```
Help on module wiringpi:
```

**NAME**

wiringpi

**DESCRIPTION**

```
# This file was automatically generated by SWIG (http://www.swig.org).
# Version 4.0.2
#
# Do not make changes to this file unless you know what you are doing--modify
# the SWIG interface file instead.
```

**CLASSES**

```
builtins.object
    GPIO
    I2C
```



```

Serial
nes

class GPIO(builtins.object)
|   GPIO(pinmode=0)
|
>>>

```

### 3. 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 use the gpio command in wiringOP to check the correspondence between the board wPi number and the physical pin.

```

root@orangepi5b:~# gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 47 | 0 | 3.3V | | | 1 | 2 | | | 5V | | |
| 46 | 1 | SDA.5 | IN | 1 | 3 | 4 | | | 5V | | |
| 54 | 2 | SCL.5 | IN | 1 | 5 | 6 | | | GND | | |
| | | PWM15 | IN | 1 | 7 | 8 | 0 | IN | 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 | | | PowerKey | | |
| 50 | 13 | SPI4_CLK | IN | 1 | 23 | 24 | 1 | IN | SPI4_CS1 | 14 | 52 |
| | | GND | | | 25 | 26 | 1 | IN | PWM1 | 15 | 35 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
root@orangepi5b:~#

```

1) A total of 16 GPIO ports can be used in the 26pins of the development board. The following uses pin 7—the corresponding GPIO is GPIO1\_C6—the corresponding wPi serial number is 2—as an example to demonstrate how to set the high and low levels of the GPIO port



```
root@orangePi5b:~# gpio readall
```

PI5B											
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	RXD.0	3	131	
		GND			9	10	0	TXD.0	4	132	

2) The steps to test directly with the command are as follows:

- 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) ; "
```

- 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)"
```

- 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:

- First use the python3 command to enter the command line mode of python3

```
root@orangePi:~# python3
```

- Then import the python module of wiringpi

```
>>> import wiringpi
>>> from wiringpi import GPIO
```

- 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.

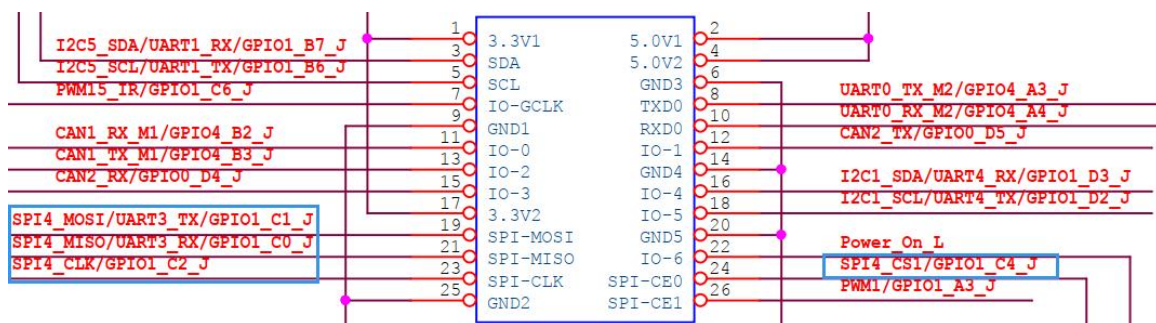
```
>>> wiringpi.digitalWrite(2, GPIO.HIGH)
```

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 5B 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:** Specifies the channel number of the SPI
- b. **--port:** Specify the port number of the 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 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 |.....|
```

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 SPI4 and then run **spidev\_test.py** The output is 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 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 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 F0 0D |.....@.....|
```

### 3. 19. 4. 26pin I2C test

1) As can be seen from the table below, the available i2c for Orange Pi 5B 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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		5V			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7	8	131	GND	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A3	UART0_RX_M2		
	PWM14_M1 (feb0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO4_A4	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GPIO0_D5			
PWM3_IR_M0 (fd8b0030)	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	59	GND			
			3.3V		17	18	58	GPIO1_D3	UART4_RX_M0 (feb70000)	I2C1_SDA_M4	PWM1_M1 (fd8b0010)
I2C3_SCL_M0	UART3_TX_M0 (feb60000)	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GPIO1_D2	UART4_TX_M0	I2C1_SCL_M4	PWM0_M1 (fd8b0000)
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22		GND			
	PWM3_IR_M2 (fd8b0030)	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	PowerKey			
			GND		25	26	35	GPIO1_C4	SPI4_CS1_M0		
								GPIO1_A3	PWM1_M2 (fd8b0010)		

As can be seen from the above table, i2c1 can be derived from pins 12 and 15 of 26pin (i2c1\_m2), and can also be derived from pins 16 and 18 of 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
```

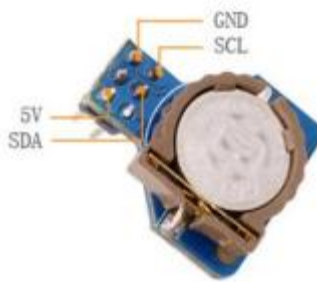
2) After starting the linux system, first confirm that there is an i2c device node under /dev

```
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 command
orangepi@orangepi:~$ sudo i2cdetect -y 3 #i2c3command
orangepi@orangepi:~$ sudo i2cdetect -y 5 #i2c5command
```





```

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. 26pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5B are four groups of uart buses: uart0, uart1, uart3 and uart4

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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		5V			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1 (feb0020)	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		PowerKey			
	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 watch the door dog test

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 shown below:

- a. The second parameter 10 indicates the counting of the door dog. If the dog is not fed in this time, the system will restart.
- b. We can feed the dog by pressing any key (except ESC) on the keyboard. After the dog is fed, the program will print a line of Keep Alive.

```
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. View 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 when



the system starts the system next time

```
orange@orange:~$ enable_docker.sh
```

3) Then you can use the following command to test the docker.

```
orange@orange:~$ 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. Download the method of downloading and installing ARM64 version of Balenaetcher

1) The download address of Balenaetcher ARM64 is:

- 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
```



May 1

ryanfortner

v1.7.9

9529280

Compare

**balenaEtcher v1.7.9**

Latest

Update and rename compile-etcher\_v1.7.3.sh to compile-etcher\_v1.7.9.sh

## ▼ Assets 10

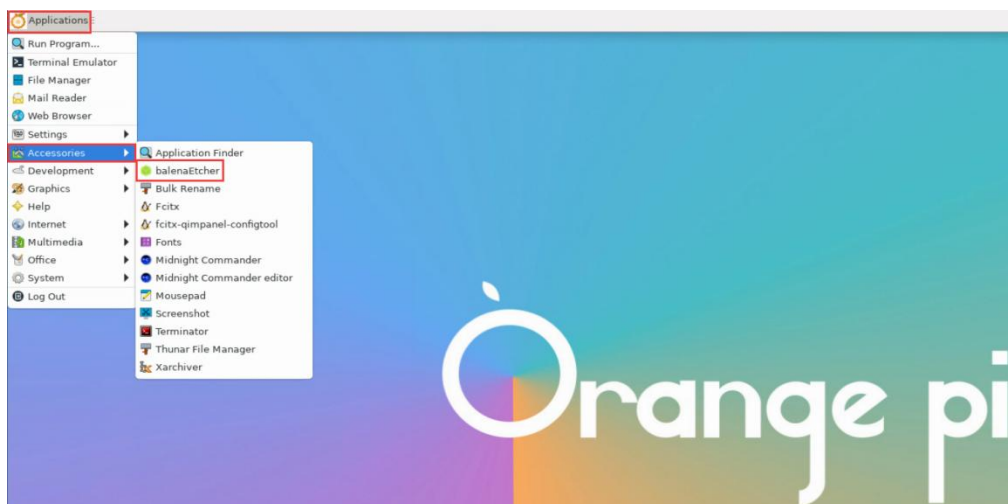
balena-etcher-electron-1.7.9+5945ab1f.aarch64.rpm	64.3 MB	May 1
balena-etcher-electron-1.7.9+5945ab1f.armv7l.rpm	58.4 MB	May 1
balena-etcher-electron_1.7.9+5945ab1f_arm64.deb	87.9 MB	May 1
balena-etcher-electron_1.7.9+5945ab1f_armv7l.deb	76.5 MB	May 1
balenaEtcher-1.7.9+5945ab1f-arm64.AppImage	97.3 MB	May 1
balenaEtcher-1.7.9+5945ab1f-armv7l.AppImage	80.9 MB	May 1

## 2) How to install and use the deb version of Balenaetcher:

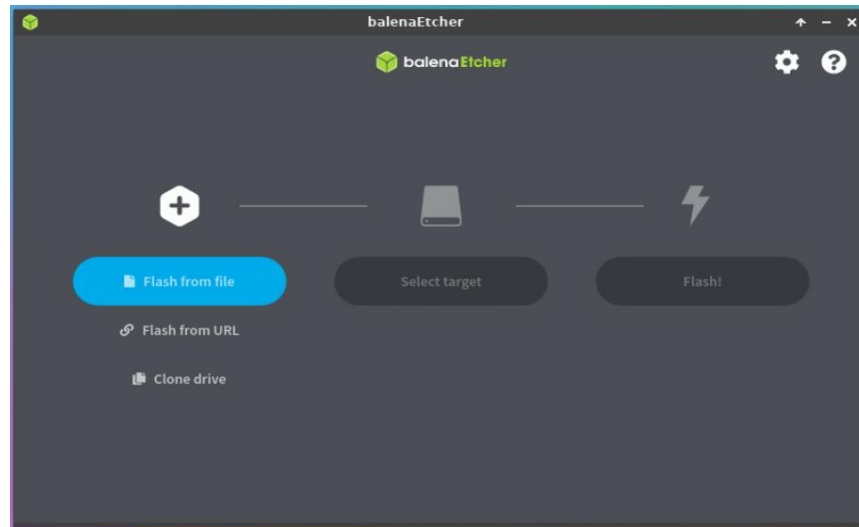
- a. deb version of BalenaEtcher installation commands as shown below:

```
orangepi@orangepi:~$ sudo apt install -y \
--fix-broken ./balena-etcher-electron_1.7.9+5945ab1f_arm64.deb
```

- b. After the BalenaEtcher installation of deb version is completed, it can be opened in the Application



- c. The interface after BalenaEtcher is opened is shown below:

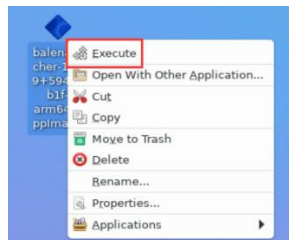


### 3) How to use the AppImage version of Balenaetcher:

- a. First add permissions to BalenaEtcher

```
orange@orange:~/Desktop$ chmod +x balenaEtcher-1.7.9+5945ab1f-arm64.AppImage
```

- b. Then select the AppImage version BalenaEtcher, click right -click, and then click Execute to open Balenaetcher



## 3. 24. The installation method of the BT linux panel

**BT 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 (excerpt from the [official website of the BT](#)).**

- 1) The order of compatibility recommendation of the BT Linux system is

```
Debian11 > Ubuntu 22.04 > Debian12
```

- 2) Then enter the following command in the Linux system to start the installation of the BT





```
orange@orange:~$ sudo install_bt_panel.sh
```

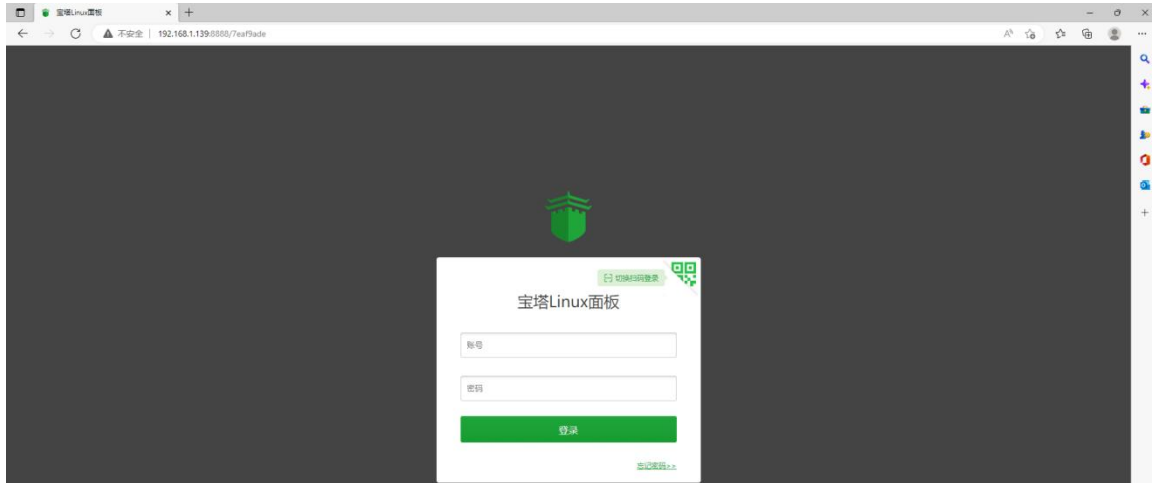
3) Then the pagoda installation program will remind whether to install the **BT-Panel** to the **/www** folder. At this time, enter Y

```
+-----+
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+-----+
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+-----+
| The WebPanel URL will be http://SERVER_IP:8888 when installed.
+-----+
Do you want to install Bt-Panel to the /www directory now?(y/n): y
```

4) Then you have to 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 8 minutes. There may be some differences according to the difference in network speed.

```
=====
Congratulations! Installed successfully!
=====
外网面板地址: http://183.15.204.10:8888/7eaf9ade
内网面板地址: http://192.168.1.139:8888/7eaf9ade
username: nslvetif
password: fec12d4b
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
=====
Time consumed: 12 Minute!
root@orange:~#
```

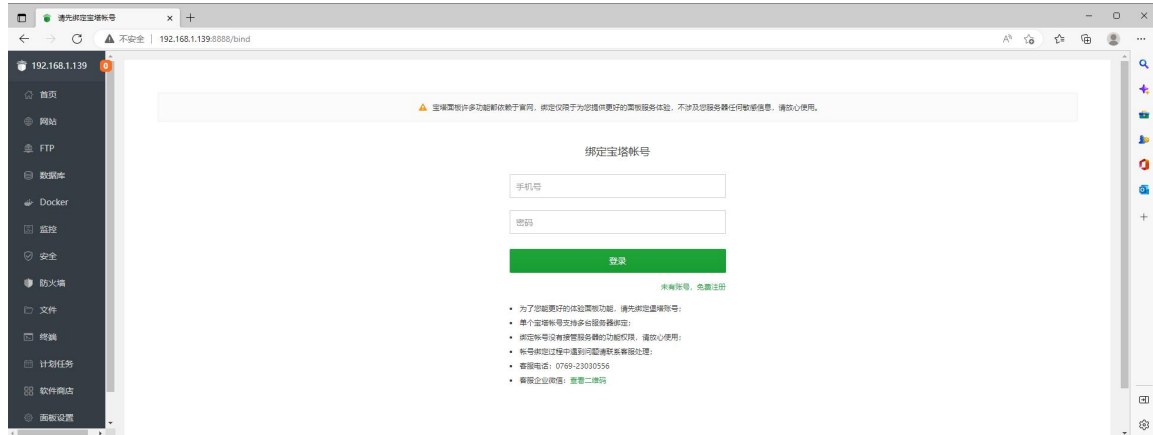
5) Enter the **panel address** displayed above in the browser to open the login interface of the pagoda Linux panel, and then enter the **username** and **password** displayed in the corresponding position.



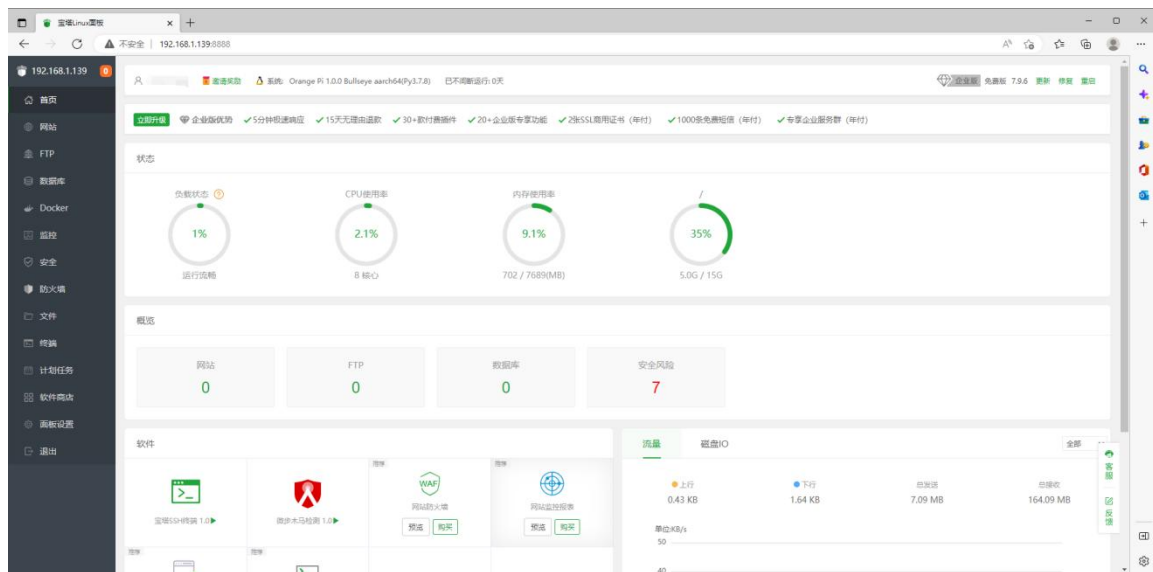
6) After successfully logging in to the BT, the following welcome interface will pop up. First, please take the intermediate user notice to read to the bottom, and then you can choose "I have agreed and read" User Agreement ", and then click" Enter the panel " You can enter the BT.



7) After entering the pagoda, you will first prompt that you need to bind the account of the pagoda official website. If you do n't have an account, you can go to the official website of the pagoda (<https://www.bt.cn>) to register one.

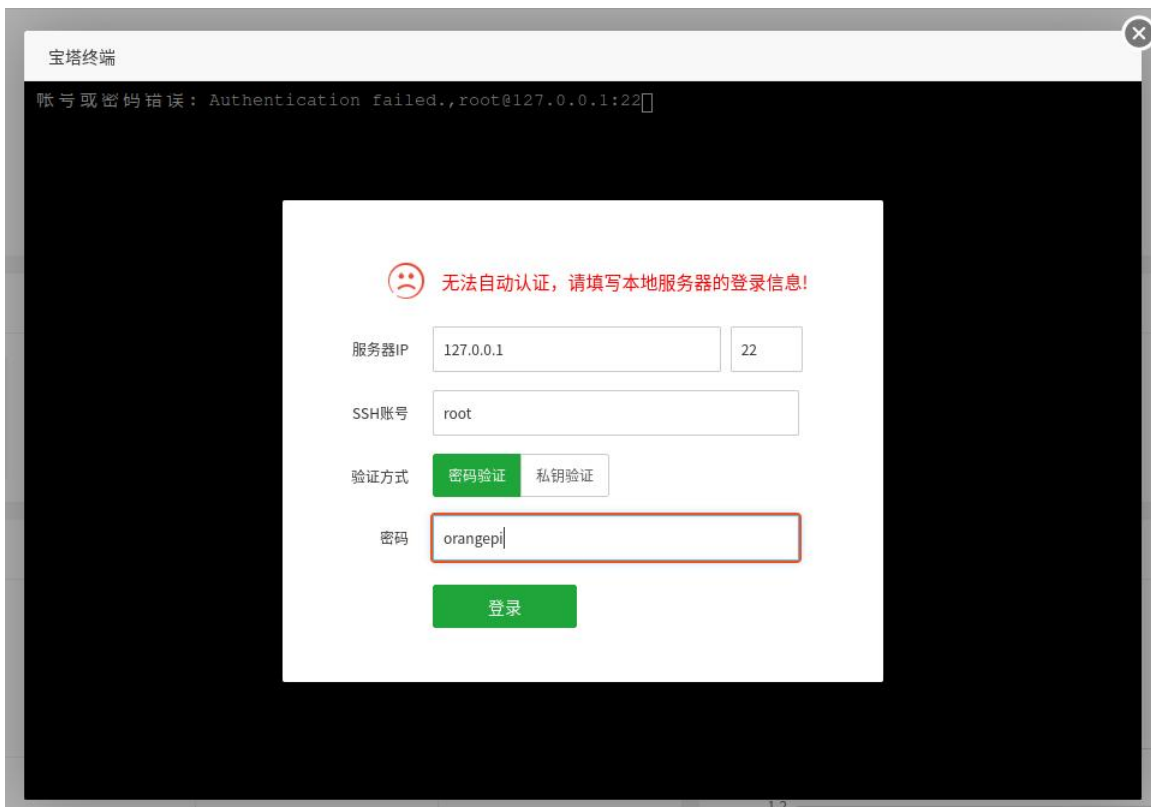


8) The final display interface is shown in the figure below. You can intuitively see some status information of the development board Linux system, such as load status, CPU usage, memory usage and storage space usage.

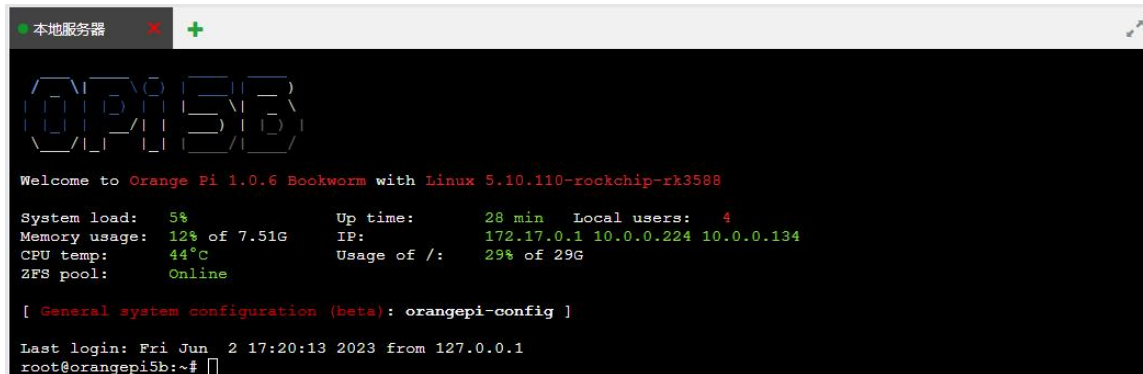


## 9) Test the SSH terminal login of the BT

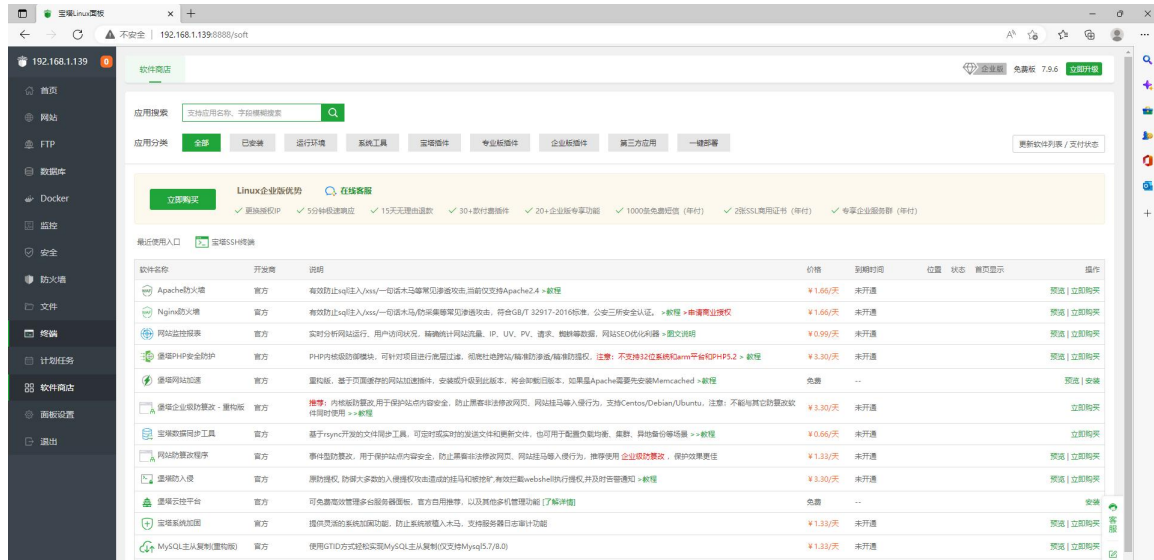
- After opening the SSH terminal of the pagoda, you will first prompt that you need to enter the password of the development board system. At this time, enter **orange** in the password box (the default password, if you have modification, please fill in the modified one).



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. You can also deploy various applications in one click. Please explore it yourself.



## 11) Pagoda command line tool test

```

orangepi@orangepi5:~$ sudo bt
[sudo] password for orangepi:
=====宝塔面板命令=====
(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 BT, 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 Chinese input method

**Note that before installing the Chinese input method, please ensure that the Linux system used in the development board is the desktop version system.**

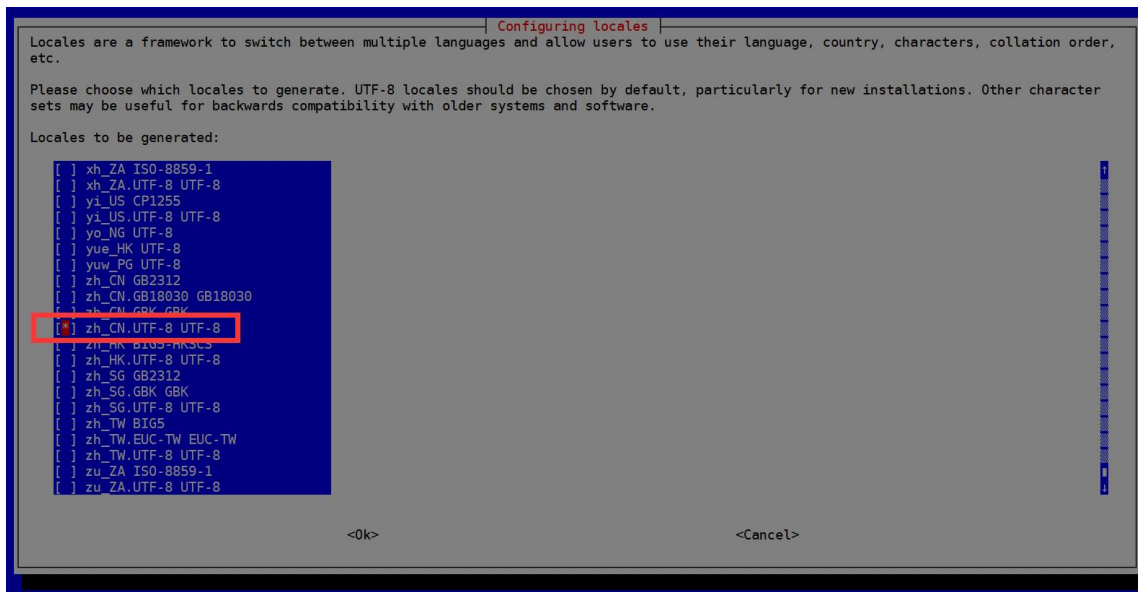
#### 3.25.1. Debian System installation method

1) First set the default **local** as Chinese

a. Enter the command below to start configuration **locale**

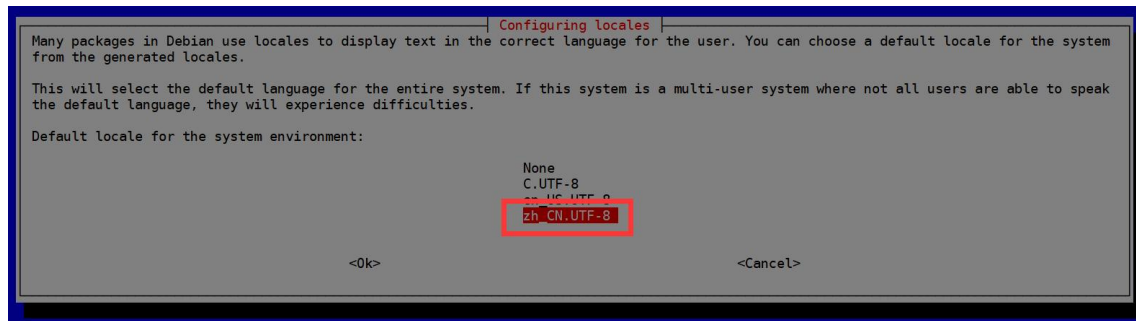
```
orangeipi@orangeipi:~$ sudo dpkg-reconfigure locales
```

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, and then return to the back to return. Just car)



c. Then set the default **locale** to **zh\_CN.UTF-8**





- d. After exiting the interface, **Locale** will be set. The output displayed by the command line is shown below

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
```

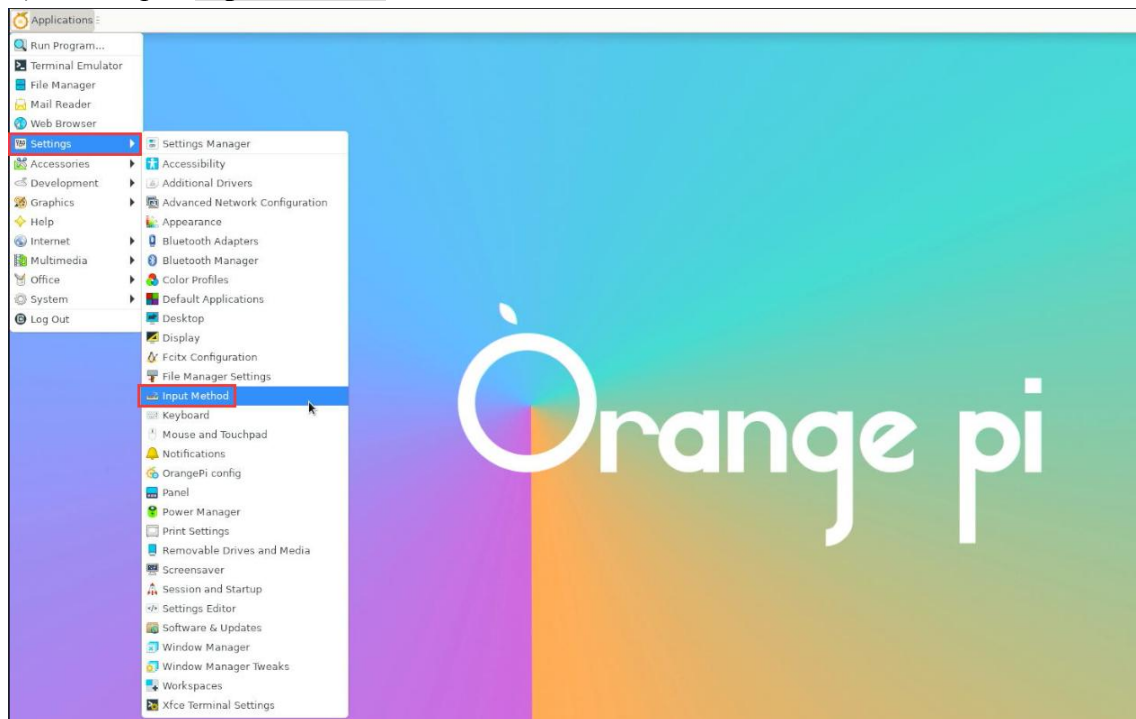
```
Generating locales (this might take a while)...
```

```
en_US.UTF-8... done
```

```
zh_CN.UTF-8... done
```

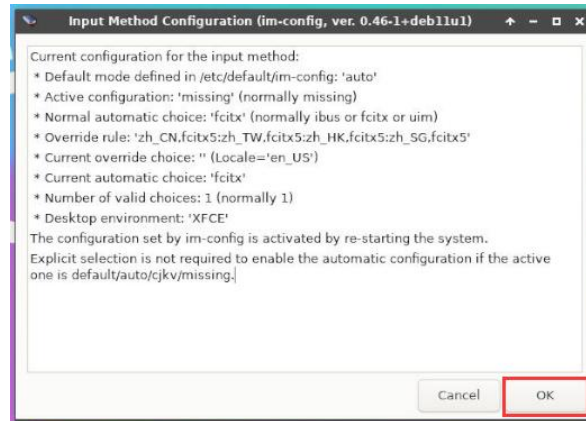
```
Generation complete.
```

## 2) Then open **Input Method**

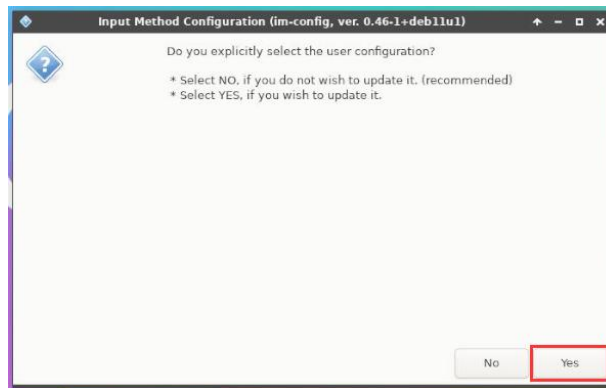


## 3) Then choose **OK**

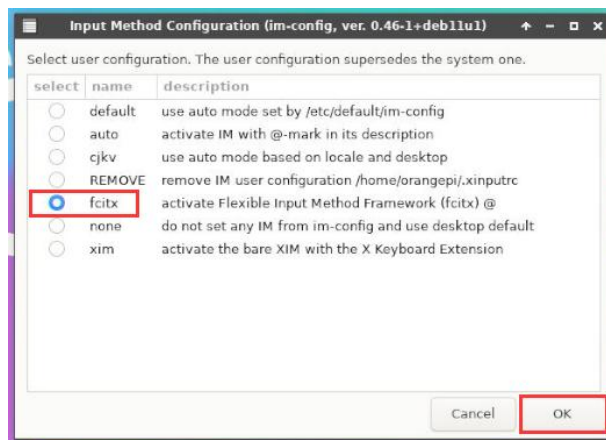




4) Then choose **Yes**



5) Then choose **fcitx**

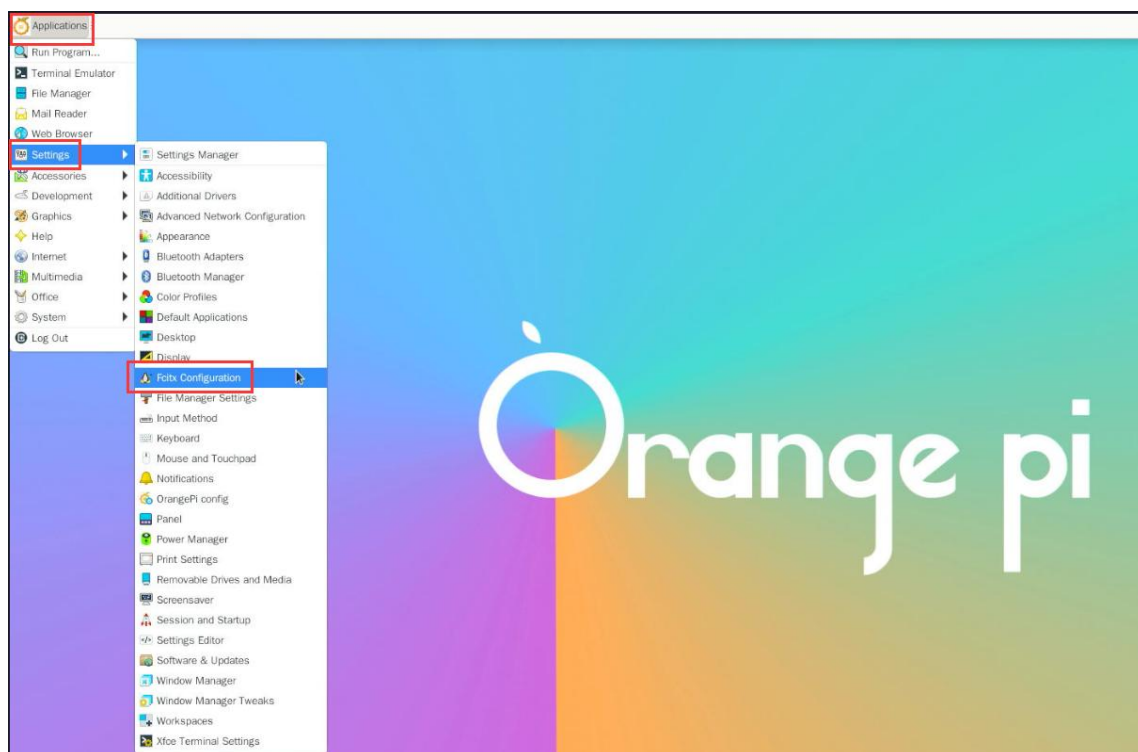


6) Then choose **OK**

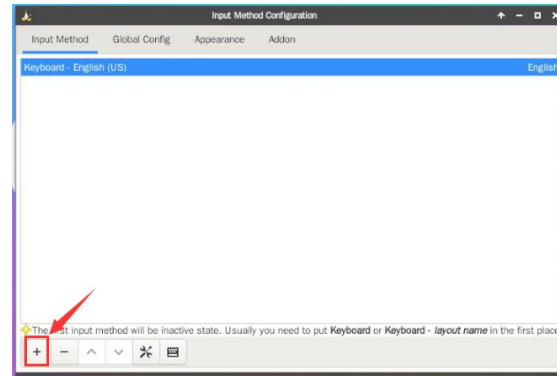


7) Then restart the Linux system to make the configuration effective

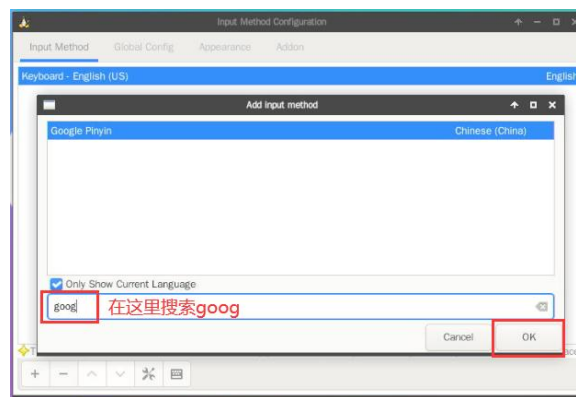
8) Then open **Fcitx configuration**



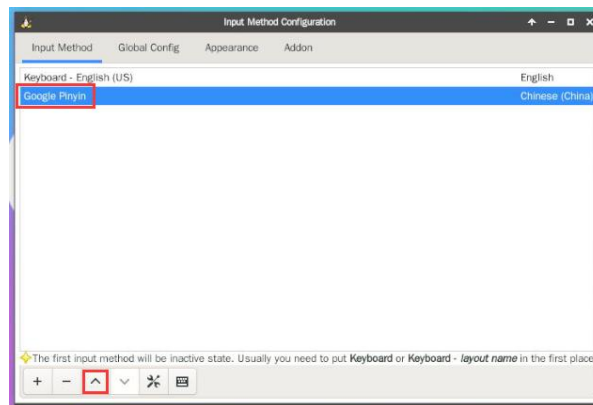
9) Then click the “+” of the position shown in the figure below

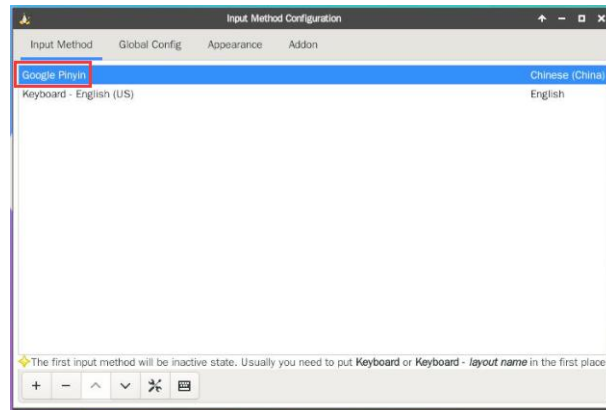


10) Then search **Google Pinyin** and click **OK**



11) Then put **Google Pinyin** to the front.

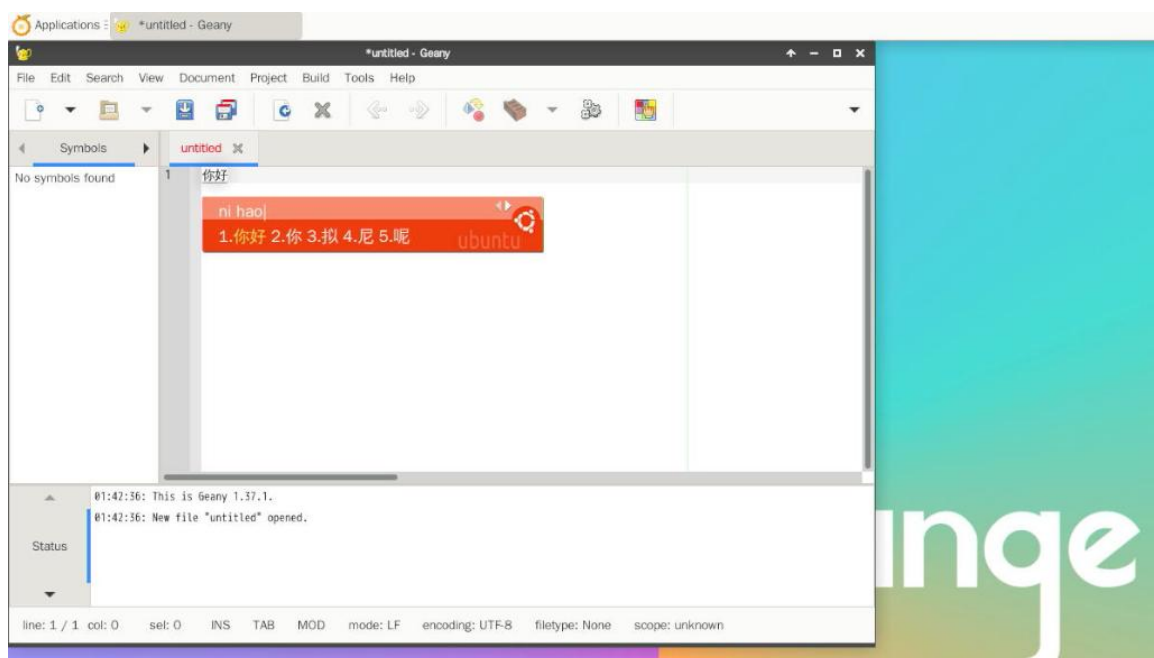




12) Then open the **Geany** editor and test the Chinese input method



13) The Chinese input method test is shown below



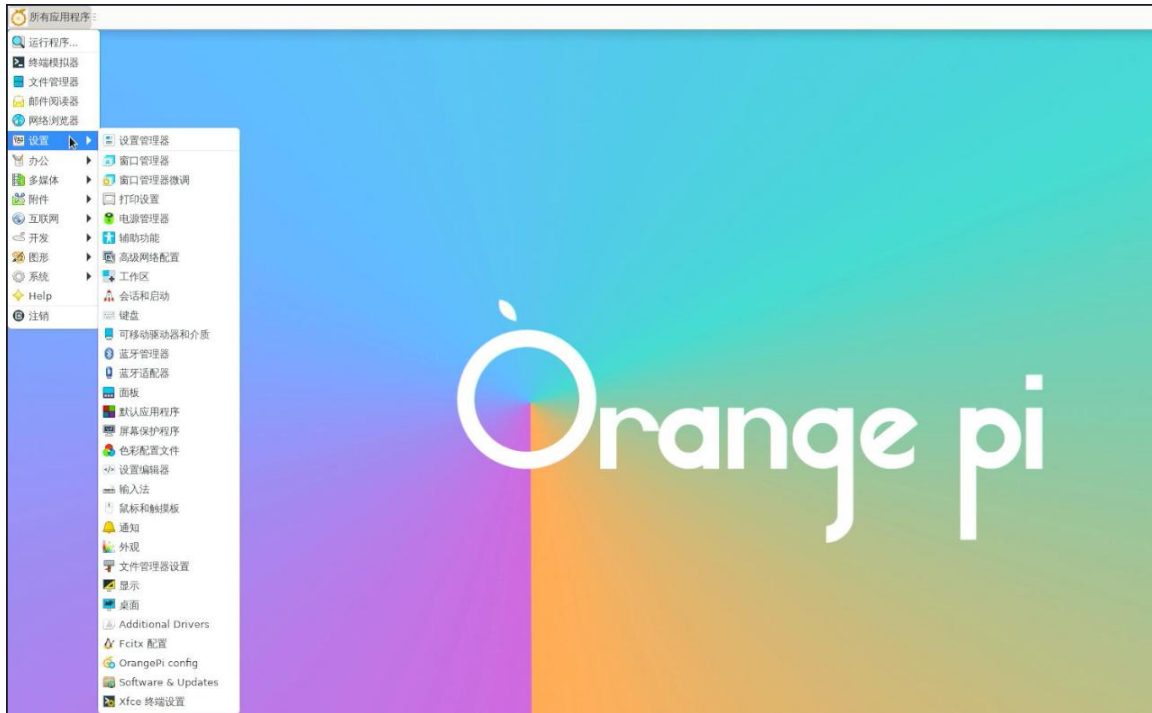


14) You can switch between Chinese and English input methods through the **Ctrl+Space** shortcut key

15) If the entire system is required as Chinese, you can set the variables **in/etc/default/locale** to set to **zh\_CN.UTF-8**

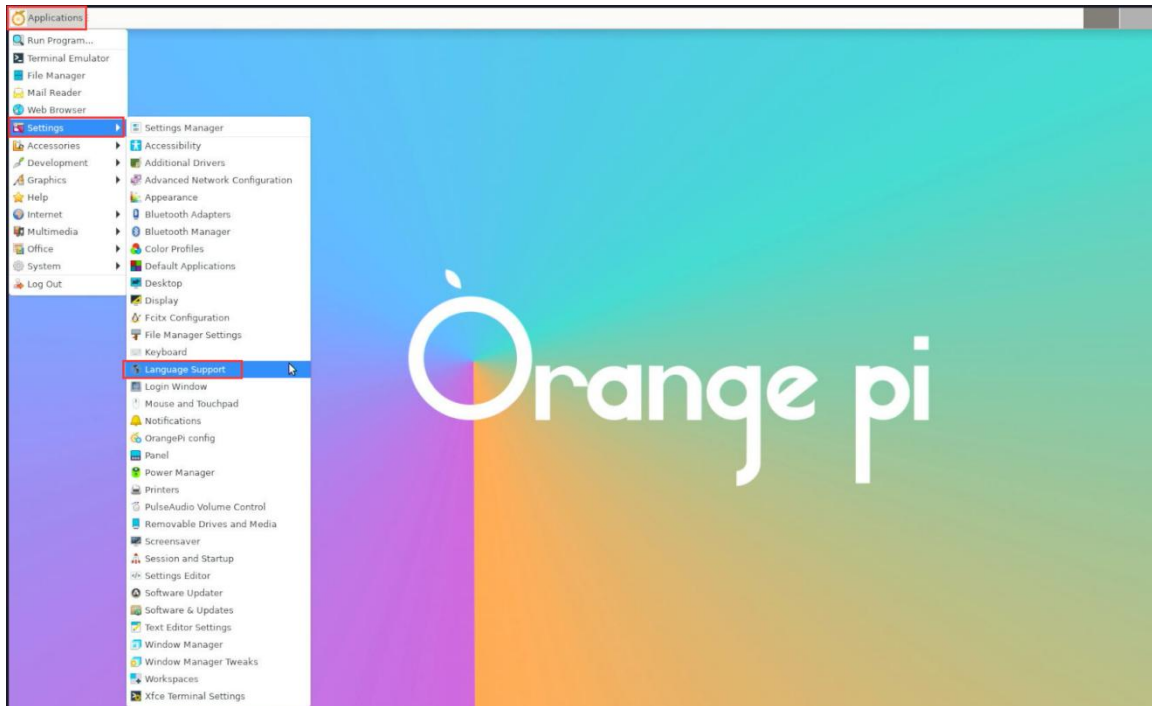
```
orange@orange:~$ sudo vim /etc/default/locale
# File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** to see the system display as Chinese

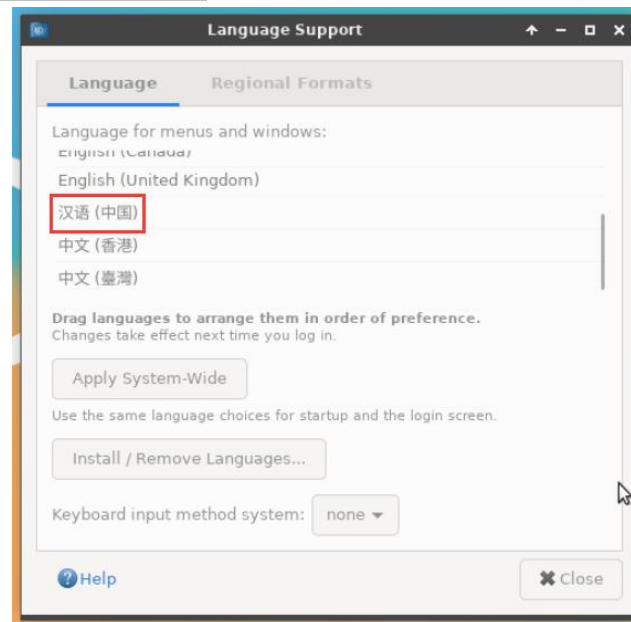


### 3. 25. 2. UBuntu 20.04 system installation method

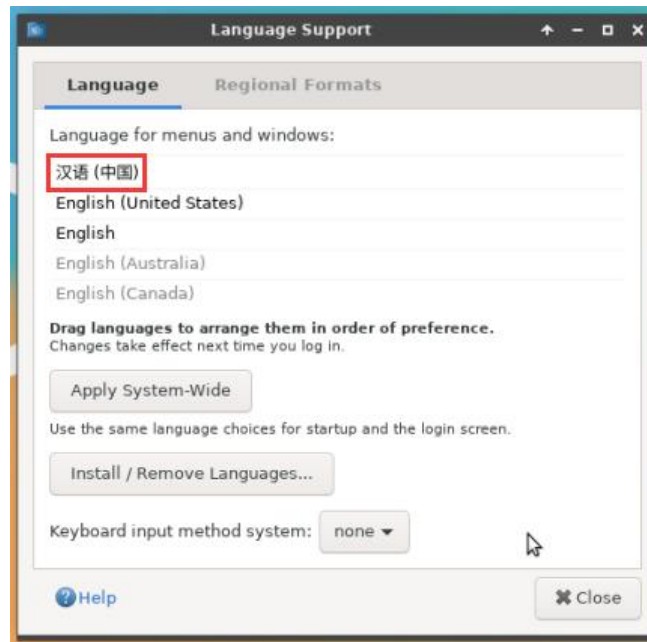
1) First open **Language Support**



2) Then find the **Chinese (China)** option

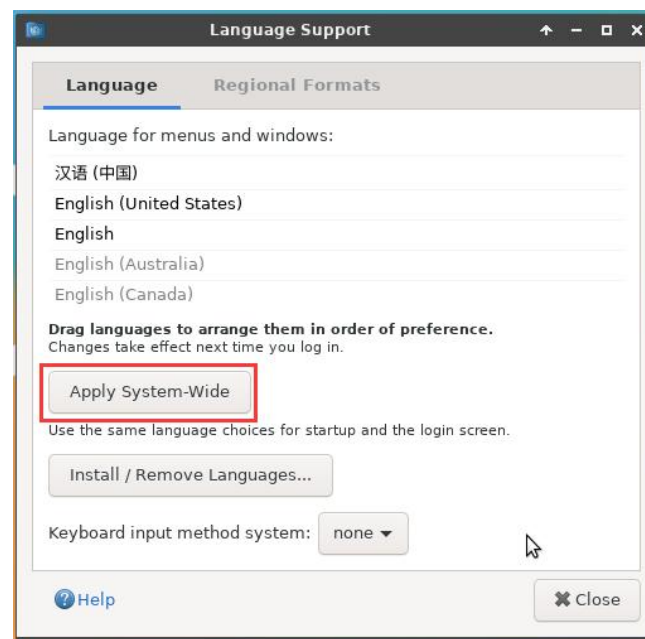


3) Then please use the mouse to select **Chinese (China)** and hold it down, and then drag it up to the beginning. The display after the dragging is shown in the figure below::



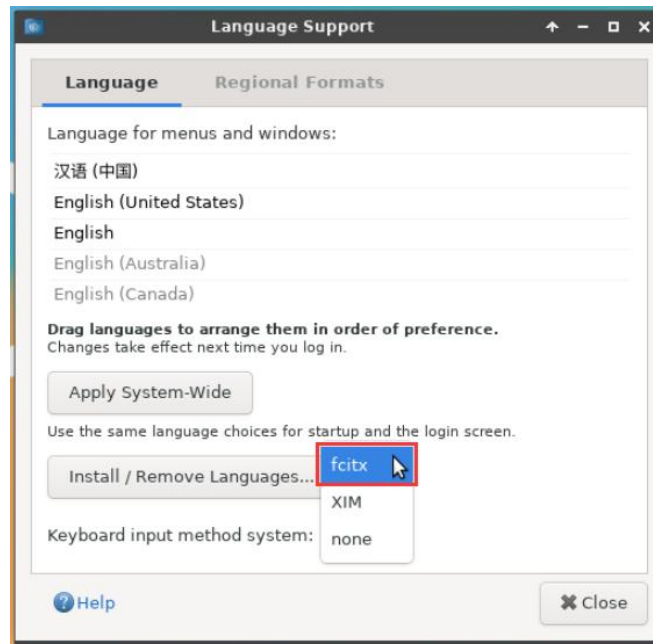
**Note that this step is not easy to drag, please try more patiently.**

4) Then select the **Apply System-Wide** to apply the Chinese settings to the entire system.



5) Then set the **keyboard input method system** to **fcitx**



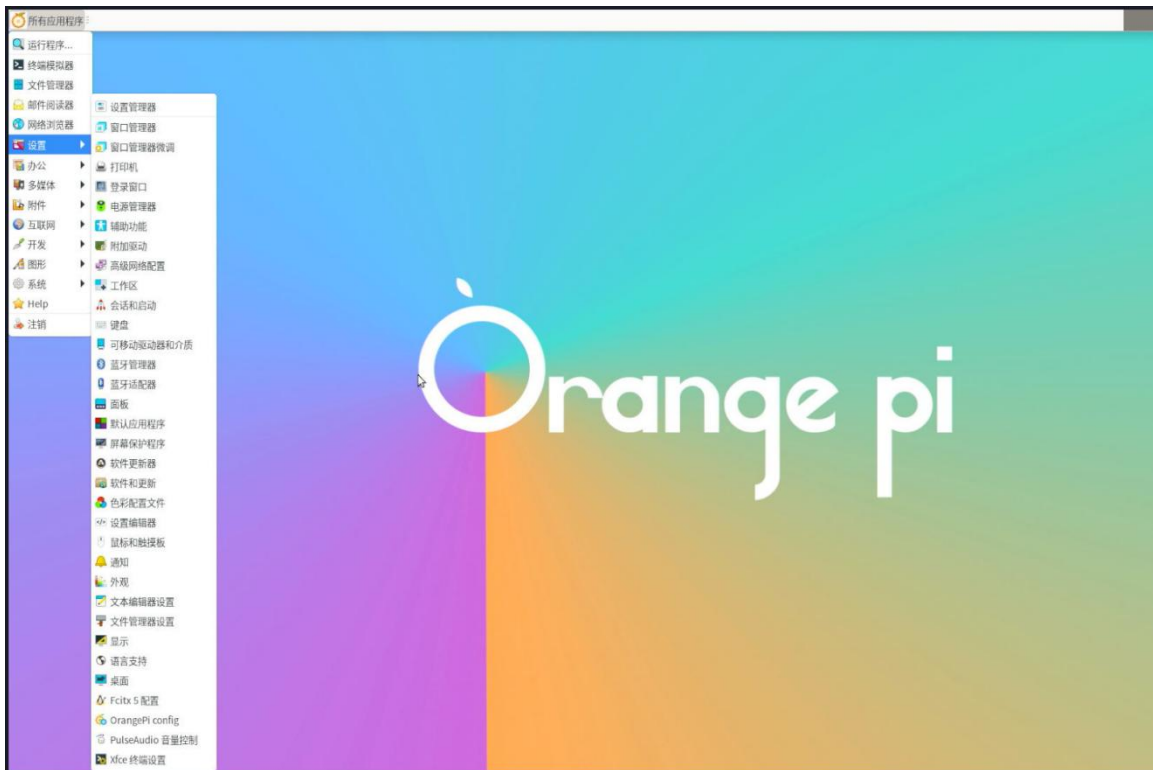


6) **Then restart the linux system to make the configuration effective**

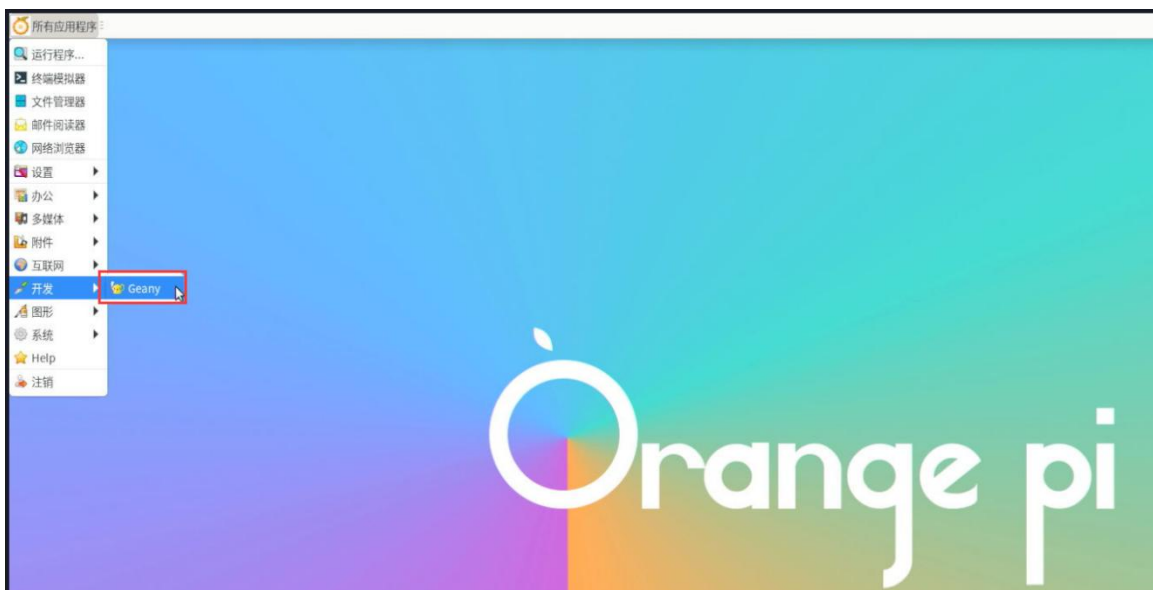
7) After re -entering the system, select **do not ask me again** at the interface below, and then determine whether the standard folder should be updated to Chinese according to your preference



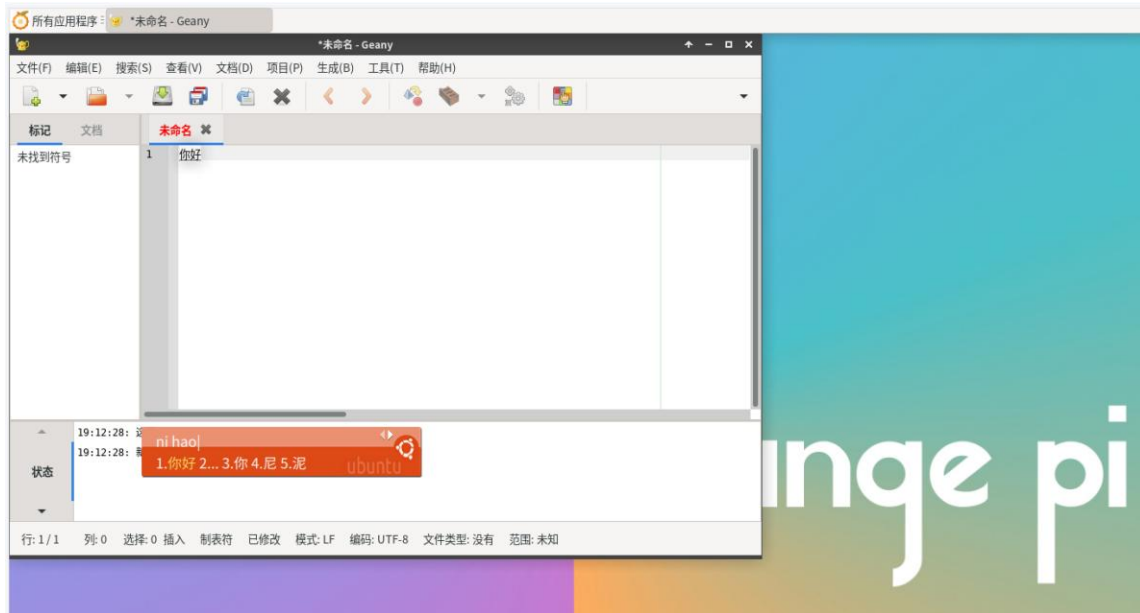
8) Then you can see that the desktop is displayed as Chinese



9) Then we can open the **Geany** test in the Chinese input method. The way to open the way is shown in the figure below

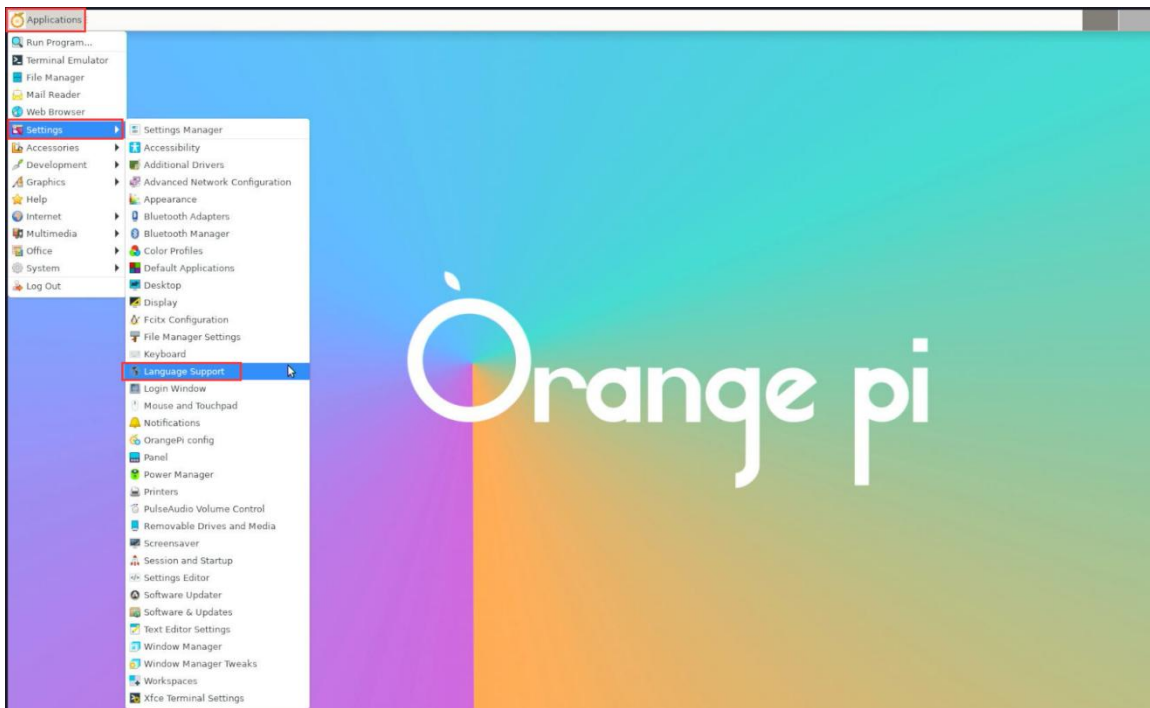


10) After opening **Geany**, the default is an English input method. We can switch into Chinese input method through the **Ctrl+Space** shortcut keys, and then we can enter Chinese.

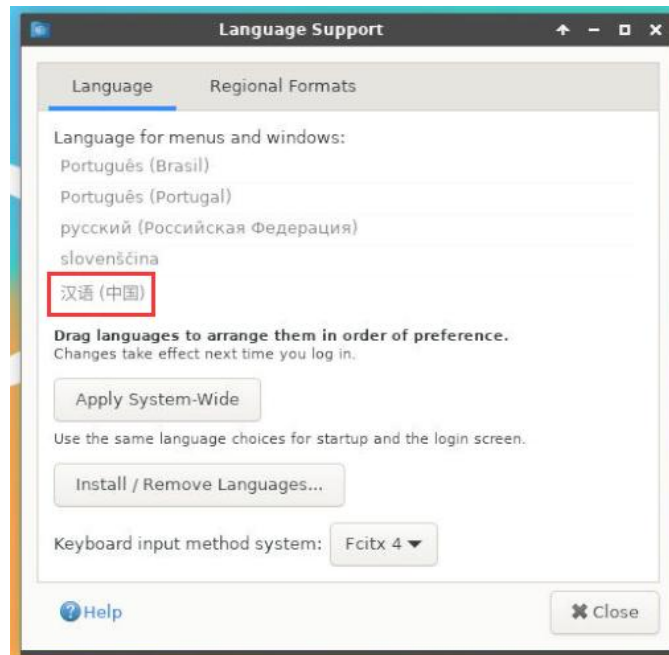


### 3. 25. 3. ubuntu 22.04 installation method

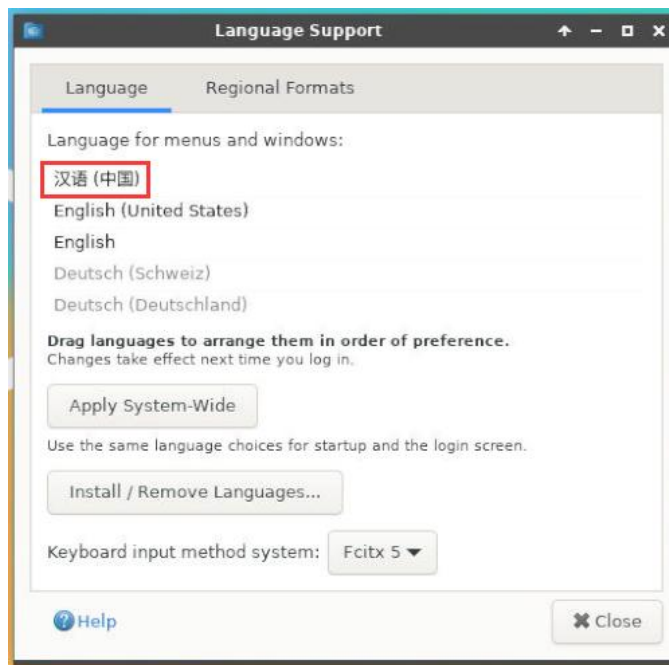
#### 1) First open **Language Support**



#### 2) Then find the **Chinese (China)** option

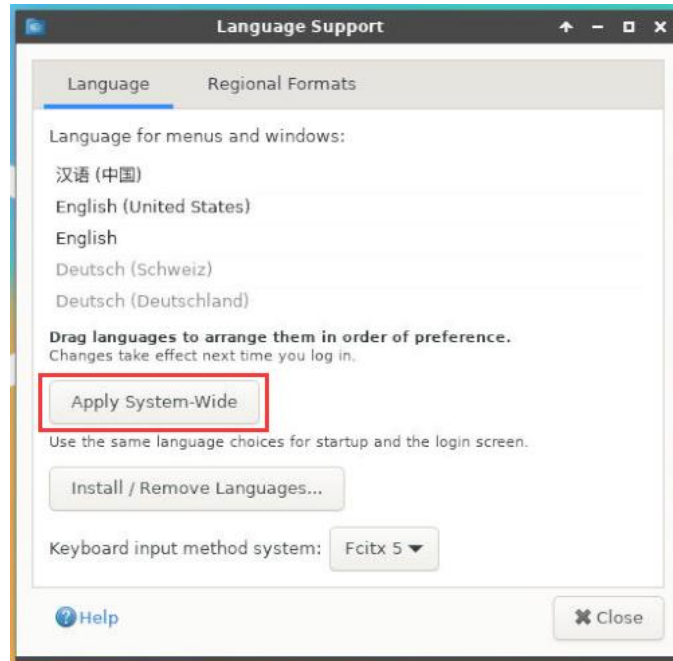


3) Then please use the mouse to select **Chinese (China)** and hold it down, and then drag it up to the beginning. The display after the dragging is shown in the figure below:



**Note that this step is not easy to drag, please try more patiently.**

4) Then select the **Apply System-Wide** to apply the Chinese settings to the entire system

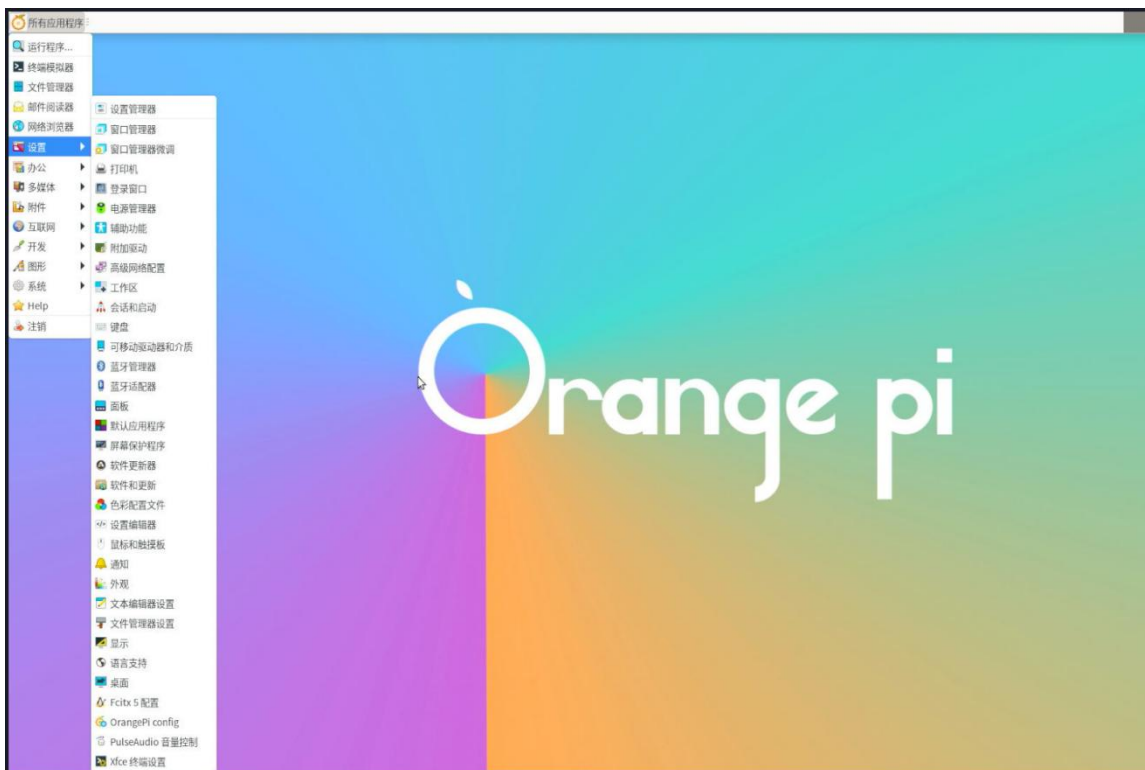


5) **Then restart the linux system to make the configuration effective**

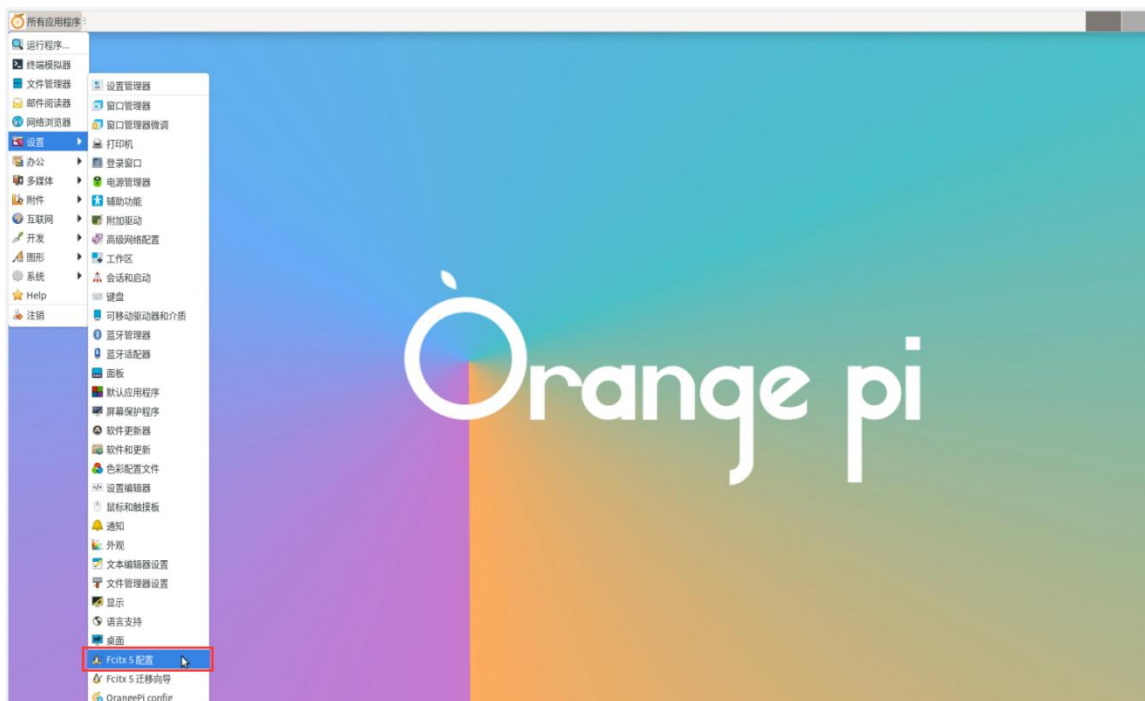
6) After re -entering the system, please choose **not to ask me again** at the interface below, and then determine whether the standard folder should be updated to Chinese according to your preference.



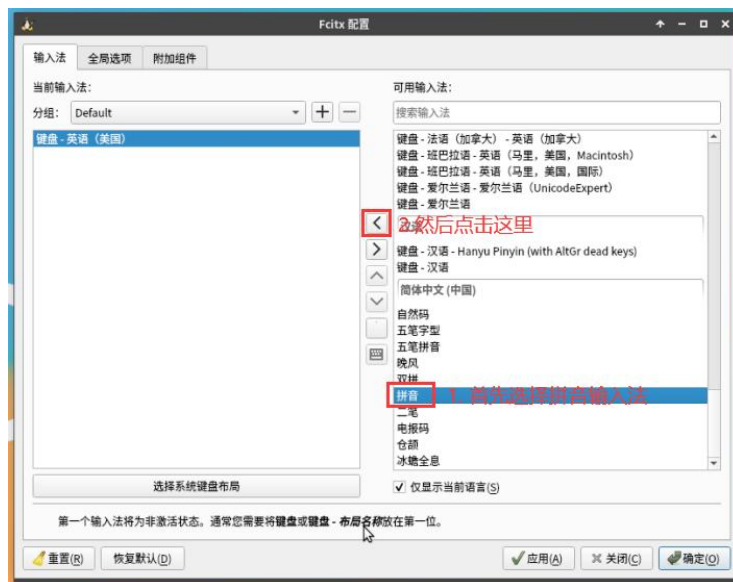
7) Then you can see that the desktop is displayed as Chinese



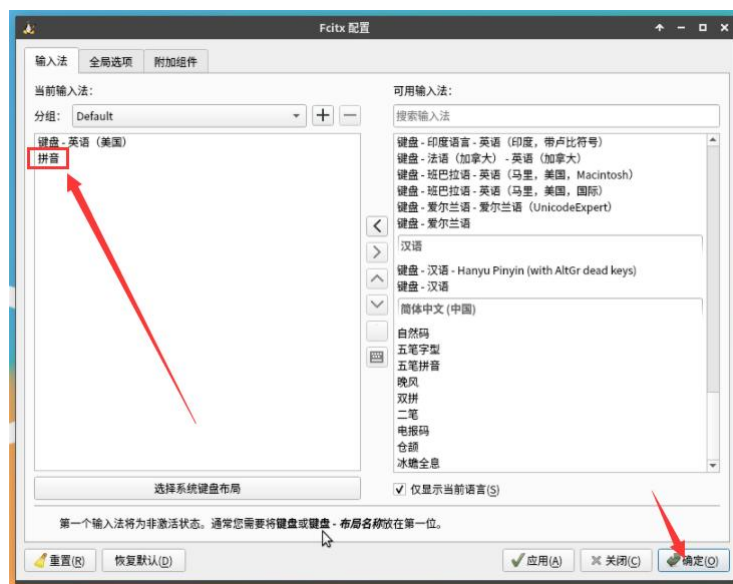
8) Then open the Fcix5 configuration program



9) Then choose to use Pinyin input method

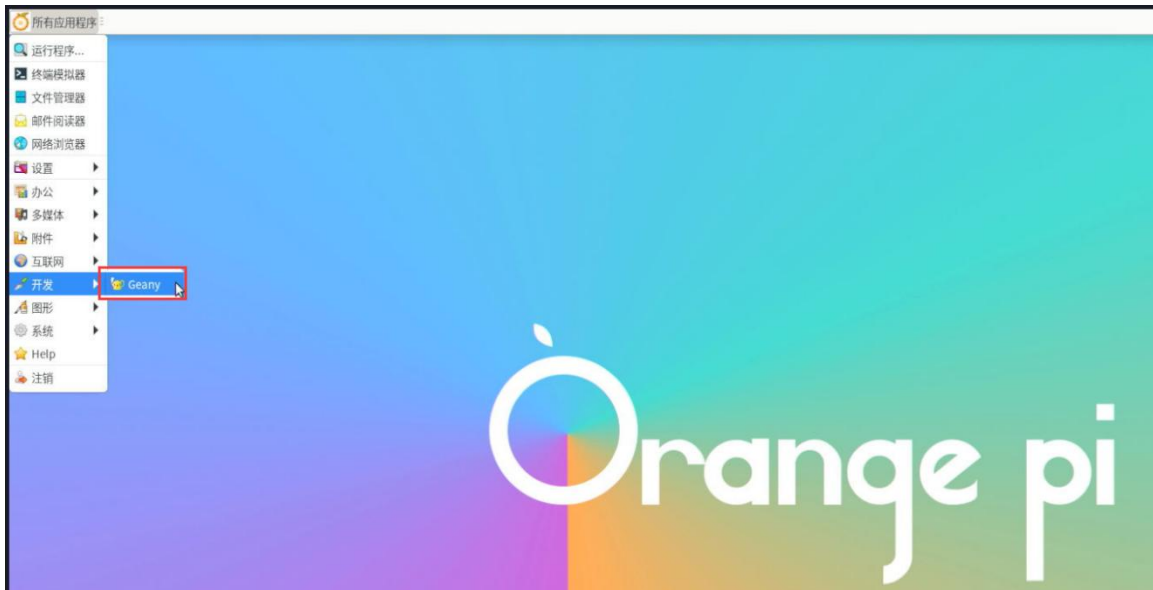


10) The interface after selecting is shown below, then click OK

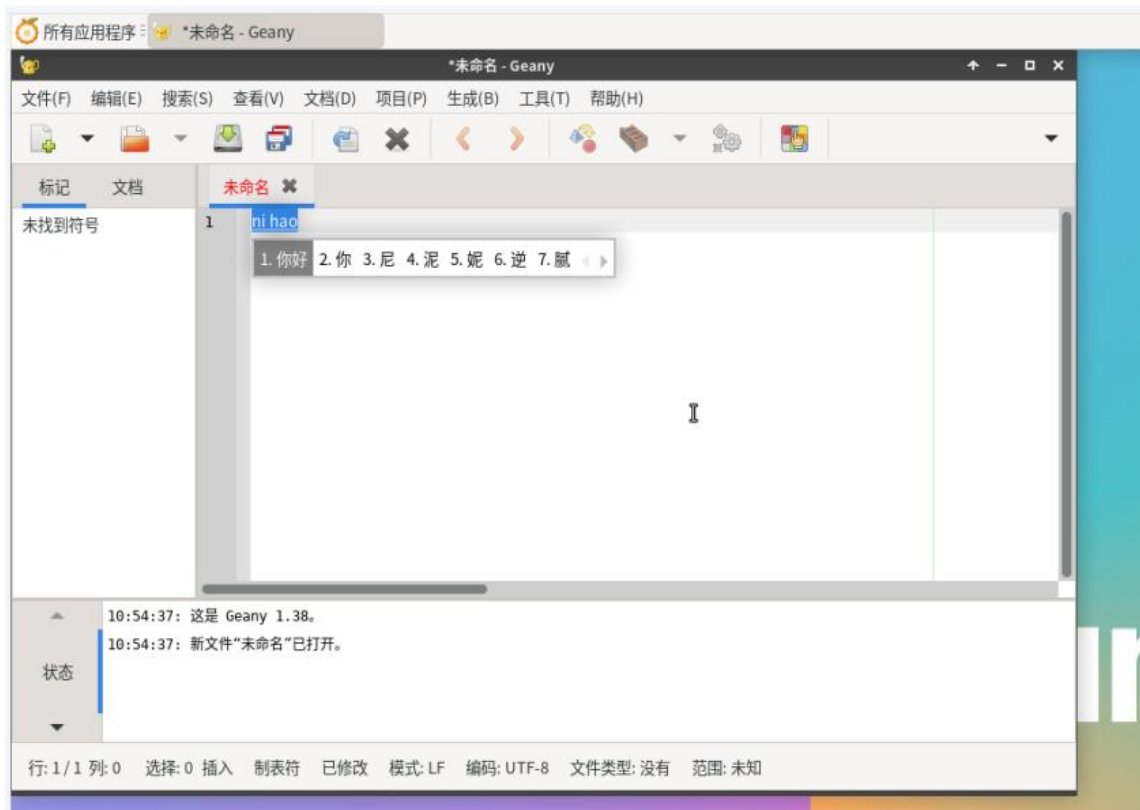


11) Then we can open the **Geany** test in the Chinese input method, The way to open is shown in the figure below





12) After opening **Geany**, the default is an English input method. We can switch into Chinese input method through the **Ctrl+Space** shortcut keys, and then we can enter Chinese.





### 3. 26. Remote login Linux system desktop method

The Ubuntu Gnome Wayland image does not support remote login to the desktop using Nomachine and VNC as described here.

#### 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 multiple devices. The following demonstrates the Linux system desktop of the Orange PI development board through Nomachine in Windows. For installation methods of other platforms, please refer to the official documentation of nomachine.

Please ensure that Windows computers and development boards are in the same local area network before operation, and can log in to the Ubuntu or Debian system that can log in to the development board normally.

1) First download the NoMachine software Linux **arm64** deb version of the installation package, 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>



Home / Download / NoMachine for ARM - arm64

NoMachine for ARM - **arm64**

Version: 8.5.3\_1  
 Package size: 48.34 MB  
 Package type: DEB  
 MD5 signature: 2291f8d8ec76f0a914285acaaa93e34d  
 For: Ubuntu 14.04/16.04/18.04/20.04, Debian 8/9/10



Although your ARMv8 device may not be listed here, we encourage you to try the packages. Please consult the installation and configuration [notes](#) about Linux for ARM packages for more details about devices and specific distributions we have tested.

[Download](#)

- b. In addition, you can also download the installation package of **Nomachine** in the **official tools**.



官方工具

[下载](#)

First enter the **remote login software-NoMachine** folder

Remote Login Software-**NoMachine**

Then download the arm64 version of the deb installation package

- ☐ nomachine\_8.5.3\_2.dmg
- ☐ nomachine\_8.5.3\_1\_amd64.deb
- ☐ nomachine\_8.5.3\_1\_x64.exe
- ☒ **nomachine\_8.5.3\_1\_arm64.deb**

- c. Then upload the downloaded **nomachine\_x.x.x\_x\_arm64.deb** to the Linux system of the development board
- d. Then use the following command to install **NoMachine** in the Linux system of the development board

```
orange@orange:~$ sudo dpkg -i nomachine_x.x.x_x_arm64_arm64.deb
```

2) Then download the installation package of the Windows version of the NoMachine software. The download address is as follows

**Note that this download link may change.**



<https://downloads.nomachine.com/download/?id=9>

## NoMachine for Windows - 64bit



Version: 8.5.3\_1  
Package size: 57.4 MB  
Package type: EXE  
MD5 signature: d585ad1e4f341444cacd3ae8add3b6ee  
For: Windows 7/8/8.1/10/11/Windows Server 2008/2012/2016/2019

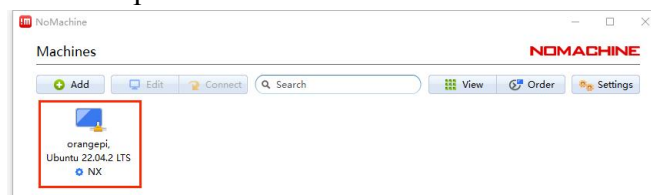
Download

3) Then install NoMachine in Windows, **Please restart the computer after installation**

4) Then open **NoMachine** in Window

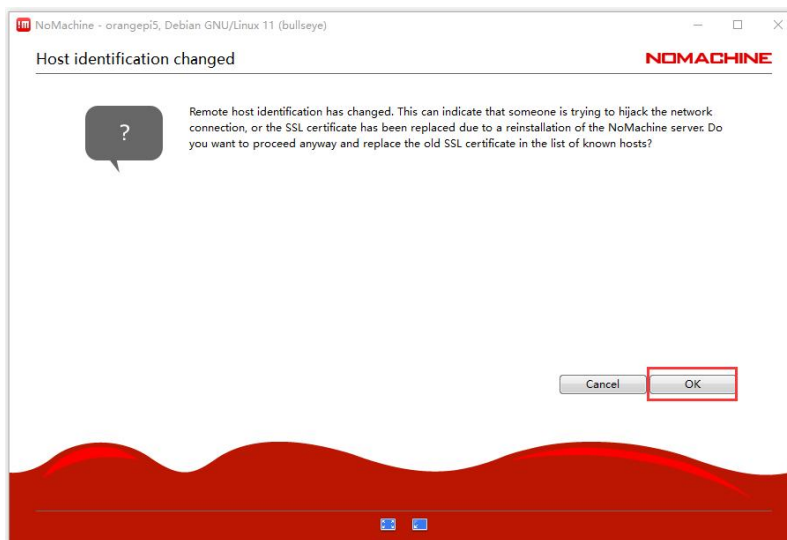


5) After Nomachine is started, it will automatically scan other devices installed in the local area network. After entering the main interface of Nomachine, you can see that the development board is already in the connected device list, and then click the location shown in the red box below in the figure below. You can start logging in to the linux system desktop of the development board





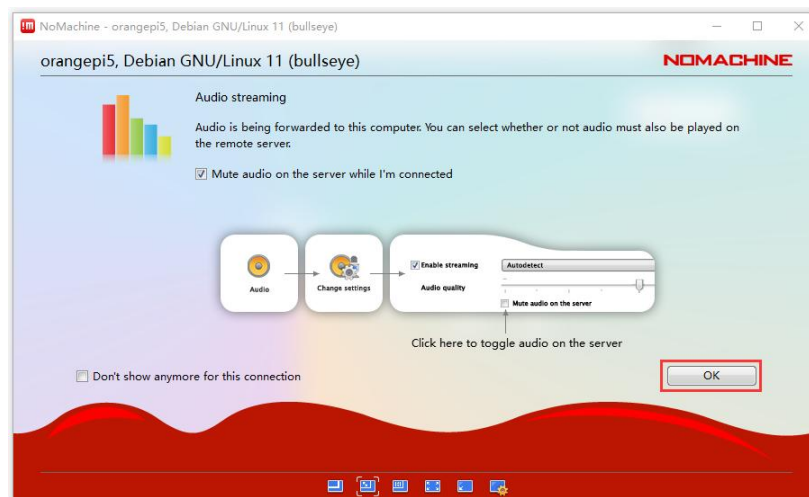
6) Then click **OK**



7) Then enter the username and password of the linux system in the corresponding position in the figure below, and then click **login** to start login

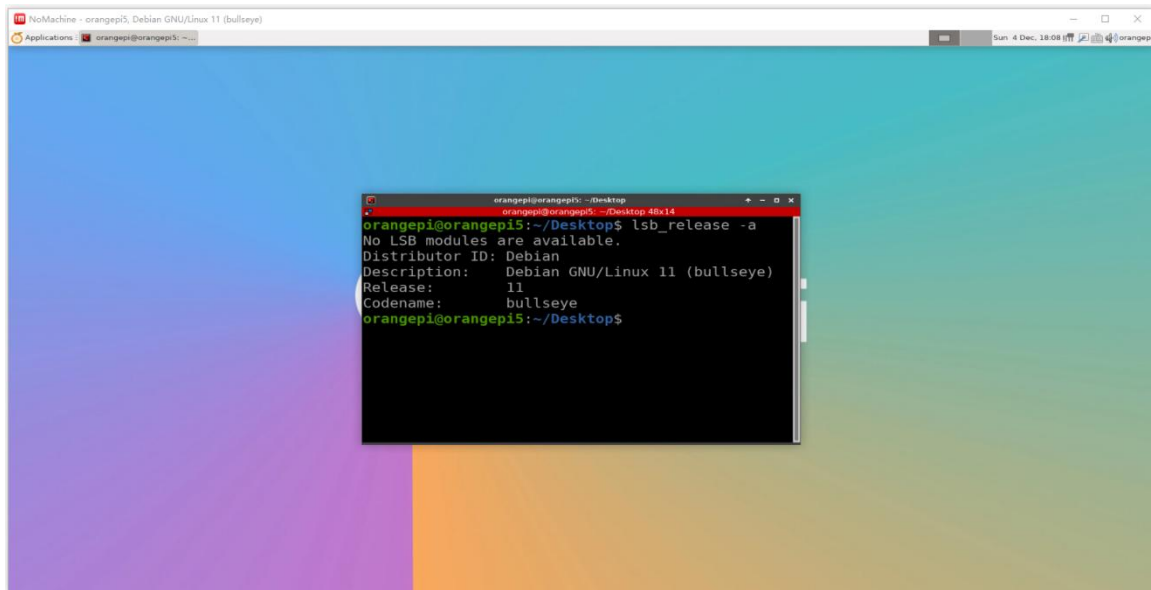


8) Then click OK in the next interface

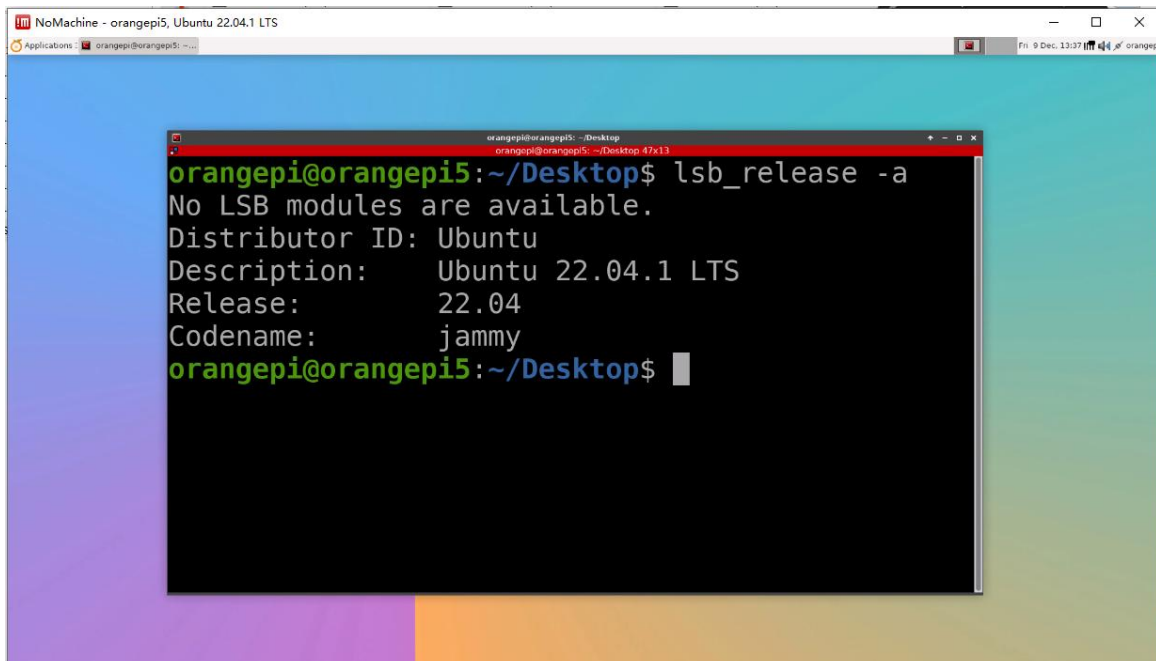


9) Finally, you can see the desktop of the development board Linux system

a. Debian11



b. Ubuntu22.04



### 3. 26. 2. Use VNC remote login

Before operation, please ensure that the Windows computer and the development board are in the same local area network, and can log in to the Ubuntu or Debian system that can log in to the development board normally.

**Ubuntu 20.04 tests many problems with VNC, please do not use this method.**





- 1) First run the **SET\_VNC.SH** script settings, and **remember to add Sudo permissions**.

```
orangePi@orangePi:~$ sudo set_vnc.sh
You will require a password to access your desktops.

Password:      #Set the VNC password here, 8 -bit characters
Verify:        #Set the VNC password here, 8 -bit characters
Would you like to enter a view-only password (y/n)? n
xauth:  file /root/.Xauthority does not exist

New 'X' desktop is orangePi5b:1

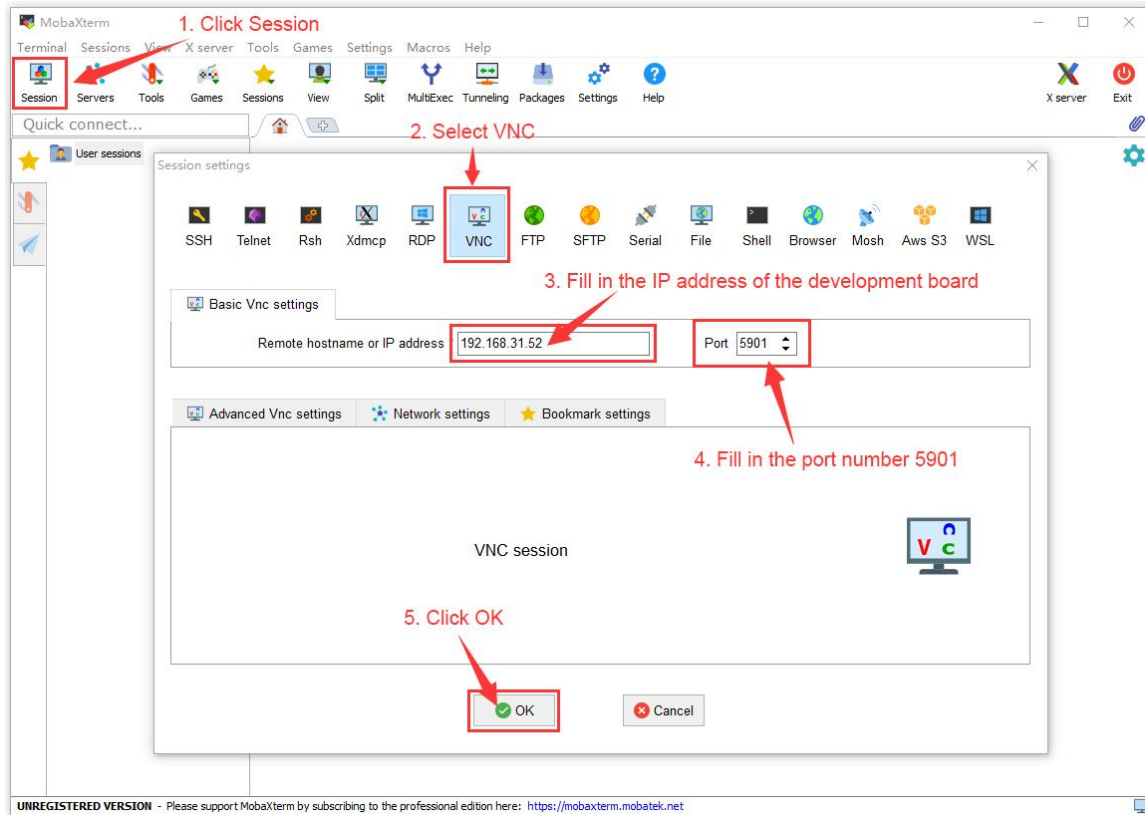
Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangePi5b:1.log

Killing Xtightvnc process ID 3047

New 'X' desktop is orangePi5b:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangePi5b:1.log
```

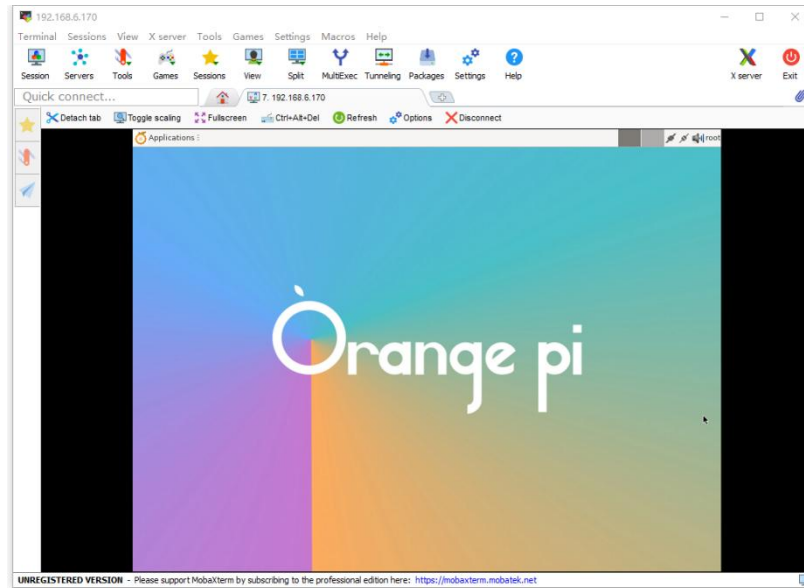
- 2) The steps to connect the development board Linux system desktop using MobaxTerm software are shown below:
  - a. First click session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm



b. Then enter the password of the VNC set before



c. After successful login, the interface is displayed as shown below, and then you can remotely operate the desktop of the development board Linux system.



### 3. 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. 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**

```
orange@orange:~$ gcc -o hello_world hello_world.c
orange@orange:~$ ./hello_world
Hello World!
```

## 2) Debian Bullseye defaults to install Python3

- a. The specific version of Python is shown below

```
orange@orange:~$ 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

```
orange@orange:~$ vim hello_world.py
print('Hello World!')
```

- c. The results of running **Hello\_World.py** are shown below

```
orange@orange:~$ python3 hello_world.py
Hello World!
```

## 3) Debian Bullseye's compilation tool and operating environment that is not installed in Java by default

- a. You can use the following command to install openjdk. The latest version in Debian Bullseye is OpenJDK-17

```
orange@orange:~$ sudo apt install -y openjdk-17-jdk
```

- b. After installation, you can check the version of Java

```
orange@orange:~$ java --version
```

- c. Write the java version **hello\_world.java**

```
orange@orange:~$ 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. Debian Bookworm system

1) Debian Bookworm is installed with the gcc compilation tool chain by default, which can directly compile C language programs in the Linux system of the development board.

- a. The version of a.gcc is as follows

```
orangepi@orangepi:~$ gcc --version  
gcc (Debian 12.2.0-14) 12.2.0  
Copyright (C) 2022 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR  
PURPOSE.
```

- b. Write the **hello\_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c  
#include <stdio.h>  
  
int main(void)  
{  
    printf("Hello World!\n");  
  
    return 0;  
}
```

- c. Then compile and run **hello\_world.c**

```
orangepi@orangepi:~$ gcc -o hello_world hello_world.c  
orangepi@orangepi:~$ ./hello_world  
Hello World!
```

2) Debian Bookworm has Python3 installed by default

- a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
```



```
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

**Use the Ctrl+D shortcut key to exit python's interactive mode.**

- b. Write the **hello\_world.py** program in Python language

```
orangepi@orangepi:~$ vim hello_world.py
print('Hello World!')
```

- c. The result of running **hello\_world.py** is as follows

```
orangepi@orangepi:~$ python3 hello_world.py
Hello World!
```

3) Debian Bookworm does not install Java compilation tools and operating environment by default.

- a. You can use the following command to install openjdk. The latest version in Debian Bookworm is openjdk-17

```
orangepi@orangepi:~$ sudo apt install -y openjdk-17-jdk
```

- b. After installation, you can check the Java version.

```
orangepi@orangepi:~$ java --version
```

- c. Write a Java version of **hello\_world.java**

```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

- d. Then compile and run **hello\_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

### 3. 27. 3. 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. 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!
```

## 2) Ubuntu Focal's default installation with 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. Write the java version **hello\_world.java**

```
orangePi@orangePi:~$ vim hello_world.java
public class hello_world
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

d. Then compile and run **hello\_world.java**

```
orangePi@orangePi:~$ javac hello_world.java
orangePi@orangePi:~$ java hello_world
Hello World!
```

### 3. 27. 4. Ubuntu Jammy system

4) Ubuntu Jammy 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 (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**

```
orange@orange:~$ gcc -o hello_world hello_world.c
orange@orange:~$ ./hello_world
Hello World!
```

## 5) Ubuntu jammy is installed with Python3 by default

- a. Python3 specific version is shown below

```
orange@orange:~$ 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

```
orange@orange:~$ vim hello_world.py
print('Hello World!')
```

- c. The results of running **Hello\_World.py** are shown below

```
orange@orange:~$ python3 hello_world.py
Hello World!
```

## 6) Ubuntu Jammy defaults to compile tools and operating environment without installing Java

- a. You can use the following command to install openjdk-18

```
orange@orange:~$ sudo apt install -y openjdk-18-jdk
```

- b. After installation, you can check the version of Java

```
orange@orange:~$ java --version
openjdk 18-ea 2022-03-22
OpenJDK Runtime Environment (build 18-ea+36-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 18-ea+36-Ubuntu-1, mixed mode, sharing)
```

- c. Write the java version **hello\_world.java**



```
orangepi@orangepi:~$ vim hello_world.java
public class hello_world
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

d. Then compile and run **hello\_world.java**

```
orangepi@orangepi:~$ javac hello_world.java
orangepi@orangepi:~$ java hello_world
Hello World!
```

### 3. 28. QT installation method

1) You can install QT5 and QT Creator with the following scripts

```
orangepi@orangepi:~$ install_qt.sh
```

2) After the installation is installed, 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. Ubuntu22.04's own QT version is **5.15.3**

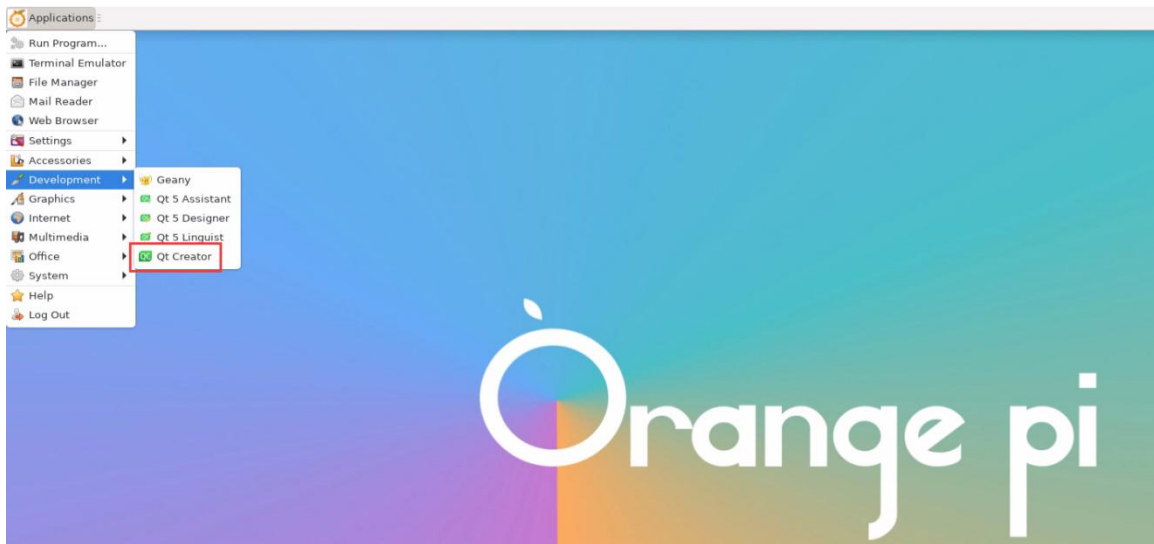
```
orangepi@orangepi:~$ install_qt.sh
.....
QMake version 3.1
Using Qt version 5.15.3 in /usr/lib/aarch64-linux-gnu
```

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**



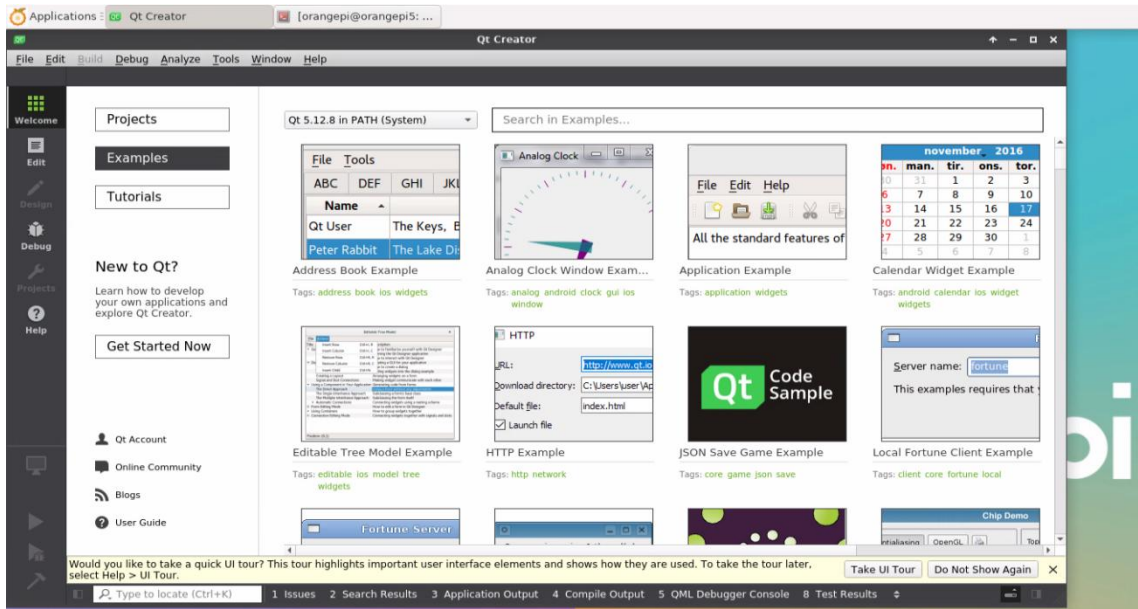
You can also use the following command to open QT Creator

```
orange@orange:~$ qtc
```

**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.**

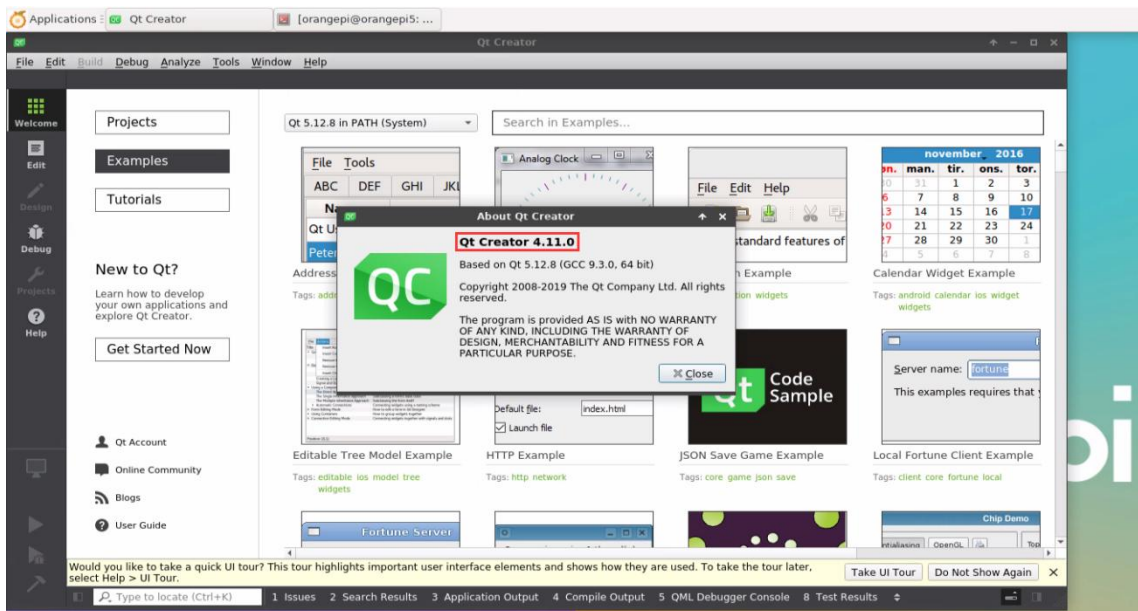
```
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
```

4) The interface after the QT Creator is opened is shown below

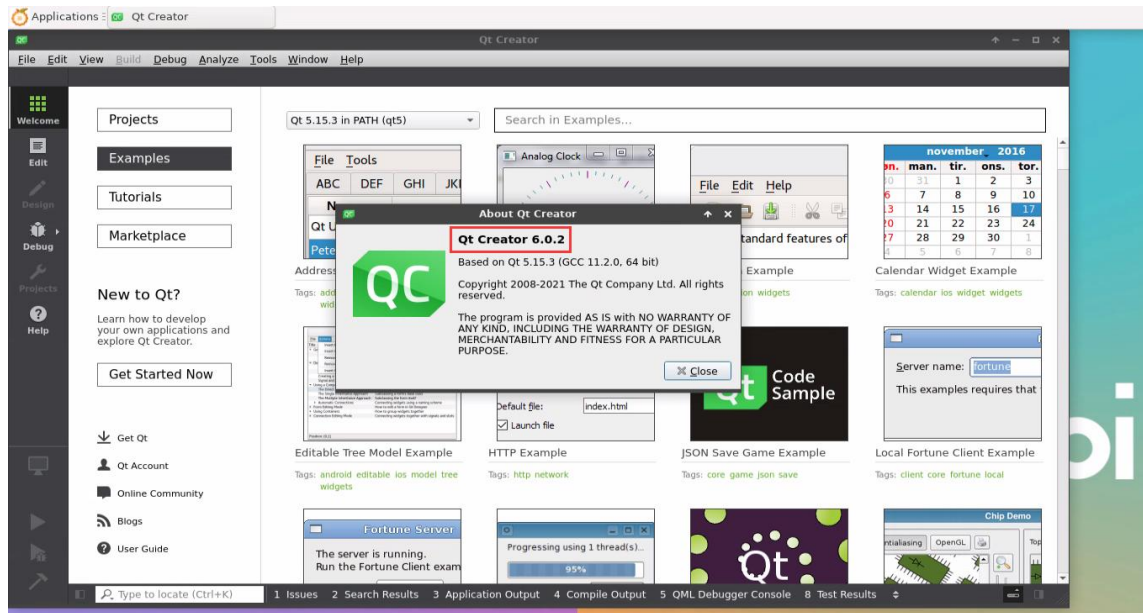


5) The version of QT Creator is shown below

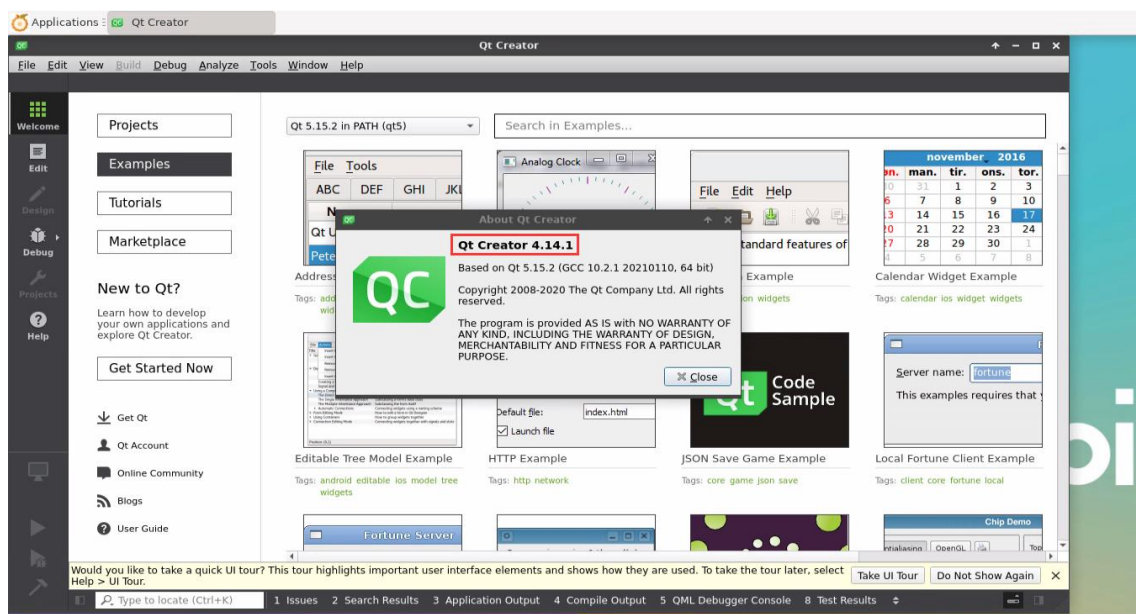
a. The default version of QT Creator in **Ubuntu20.04** is shown below



b. The default version of QT Creator in **Ubuntu22.04** is shown below



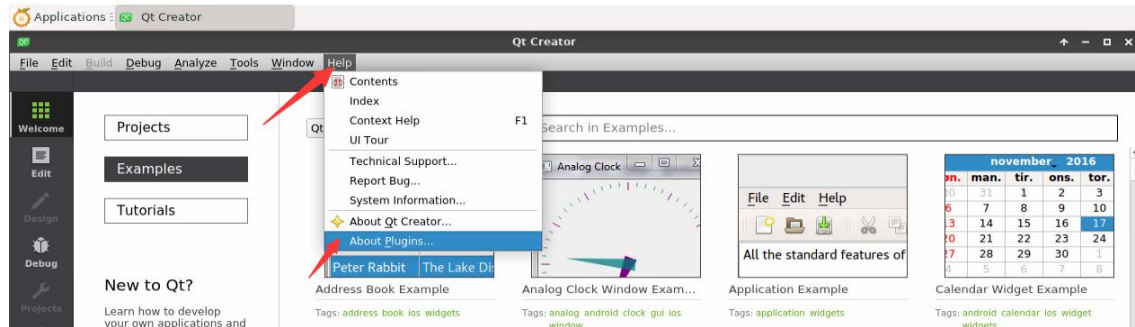
c. The default version of QT Creator in **Debian11** is shown below



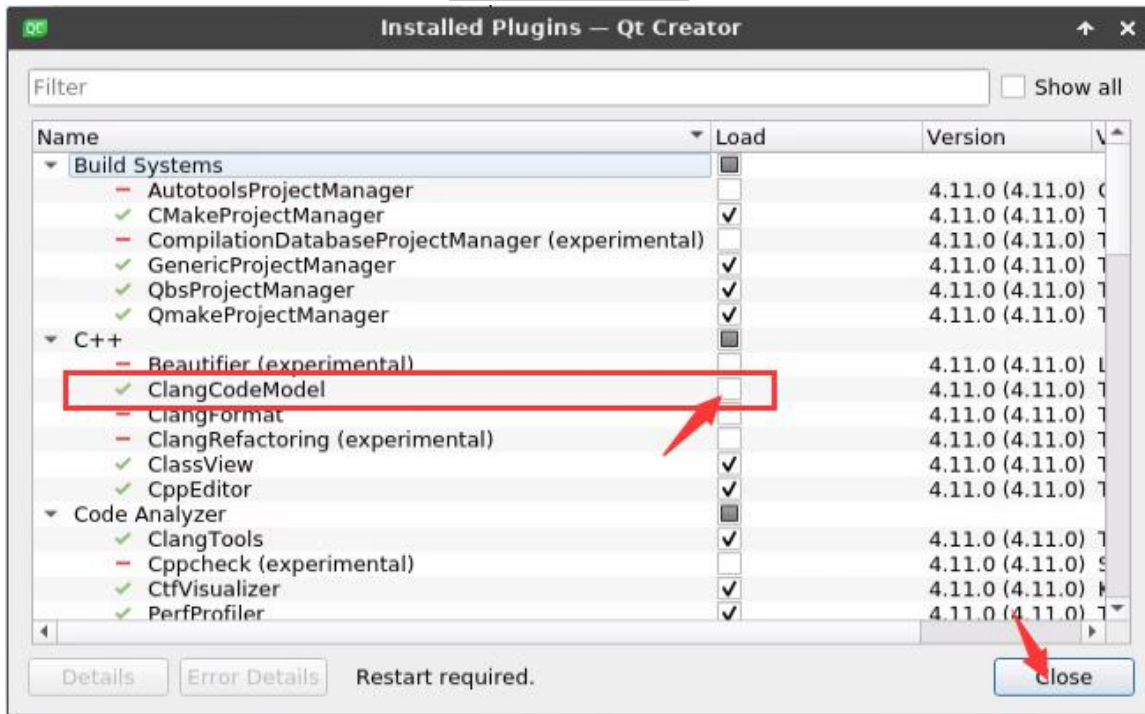
6) Then set it QT

a. First open **Help->About Plugins...**





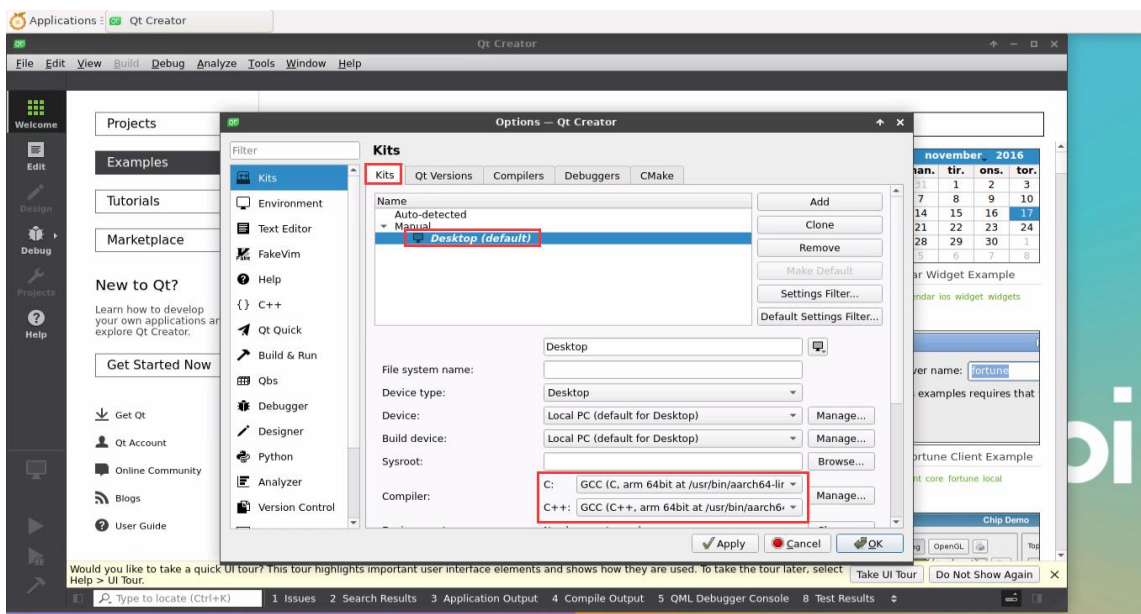
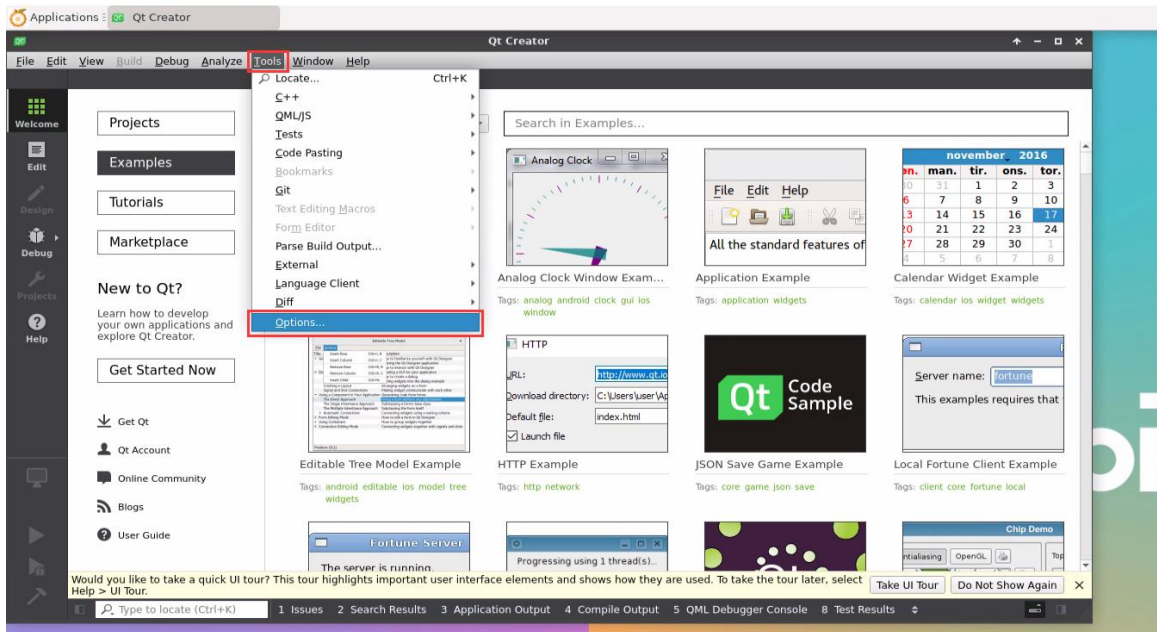
b. Then remove the hook of **ClangCodeModel**



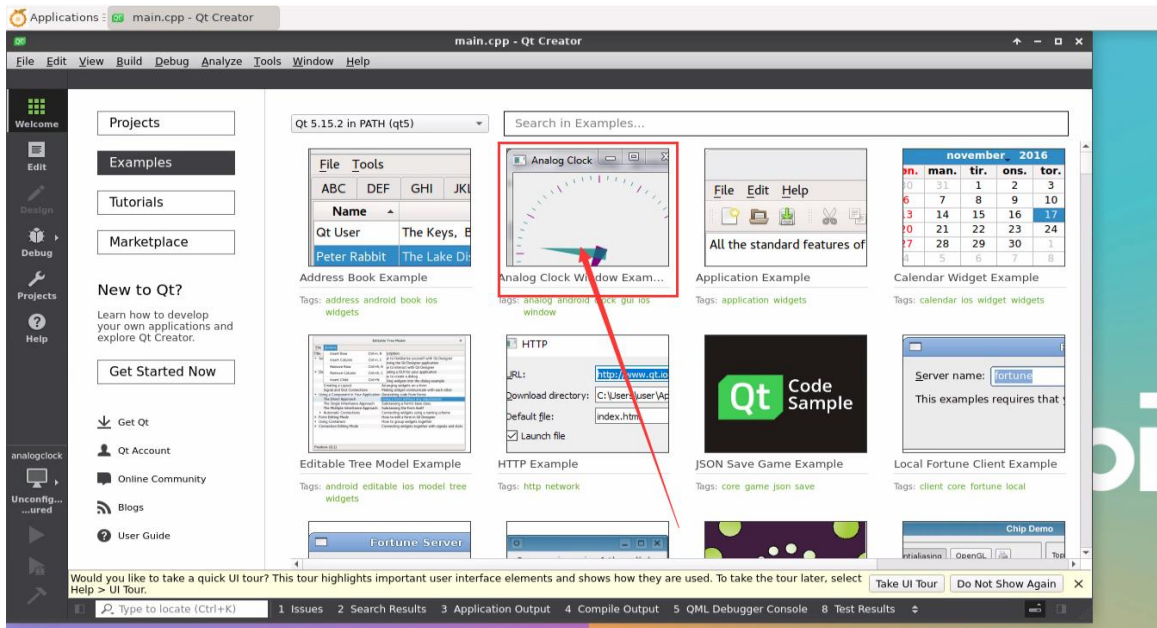
c. **After setting, you need to restart QT Creator**

d. Then make sure that the GCC compiler used by QT Creator, if the default is Clang, please modify it to GCC

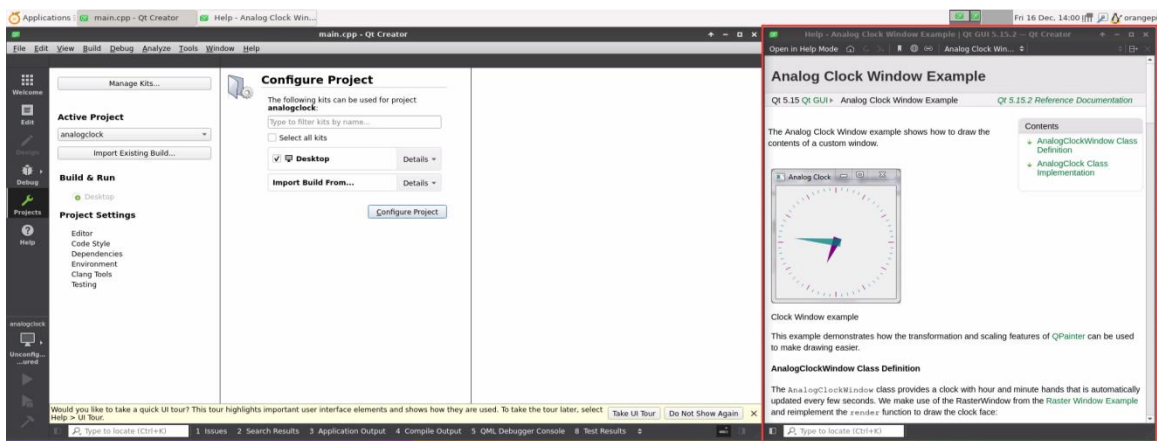




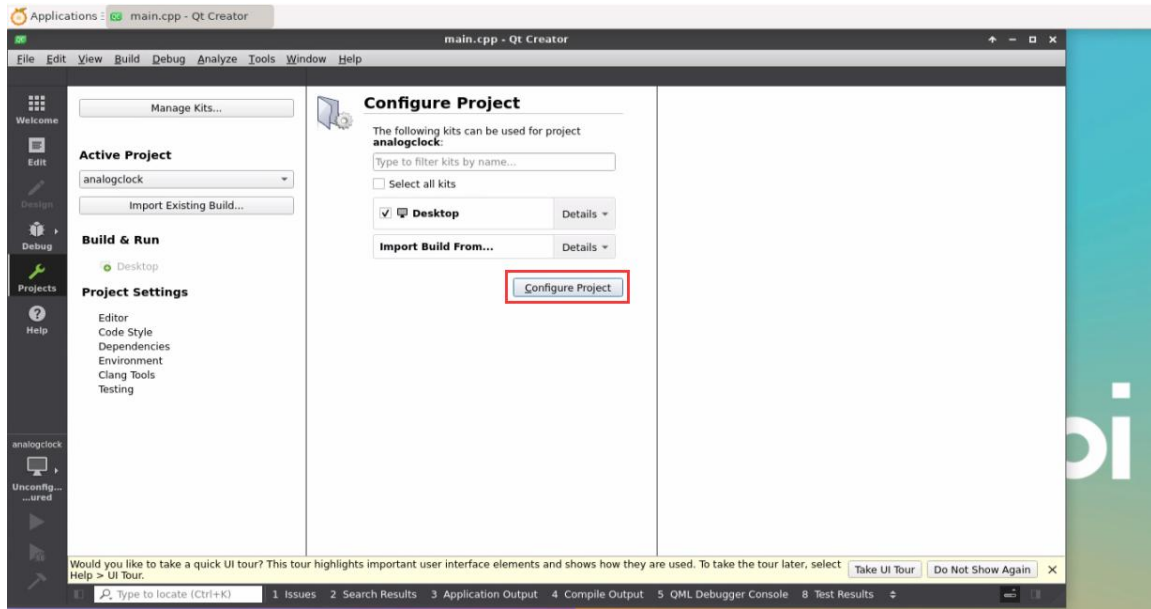
7) Then you can open an example code



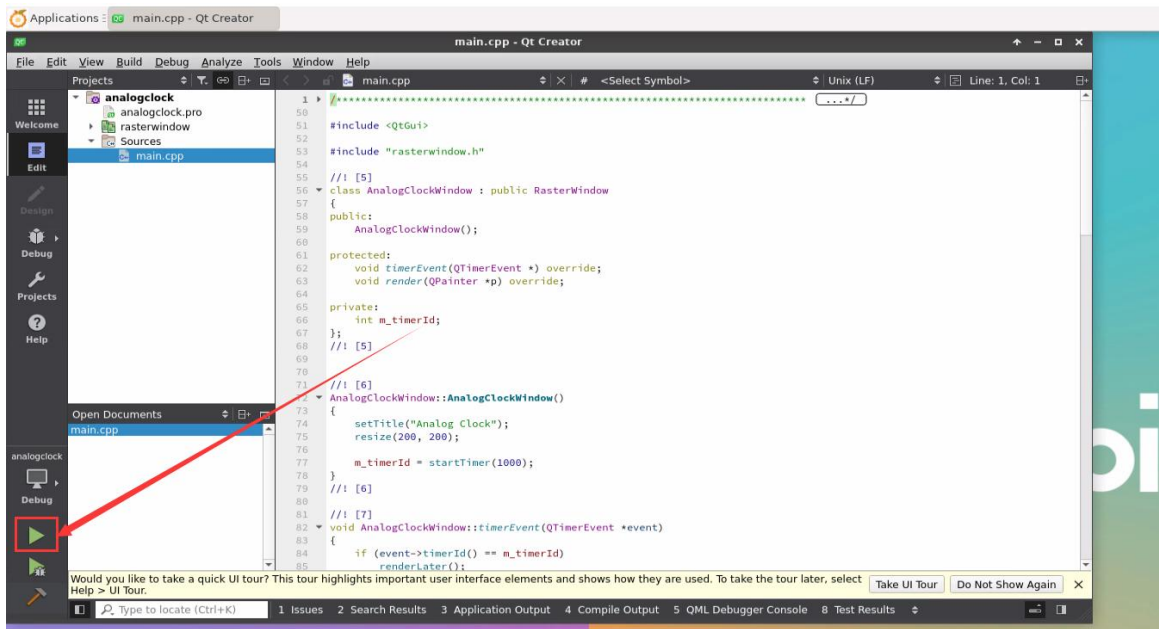
8) After clicking the example code, you will automatically open the corresponding description document. You can carefully look at the instructions for usage



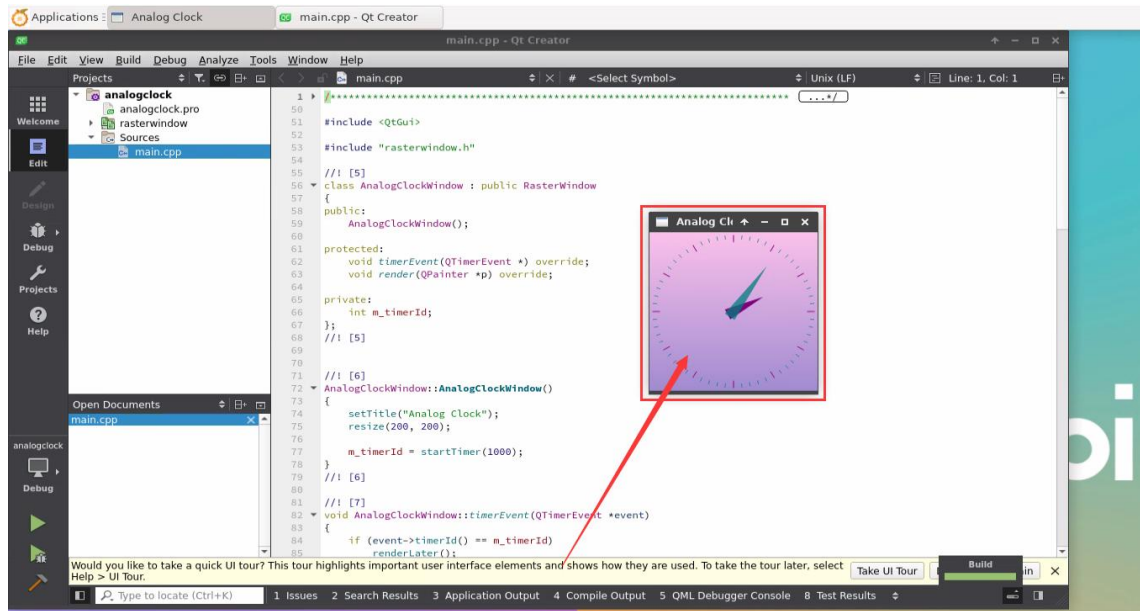
9) Then click down **Configure Project**



10) Then click the sample code under the green triangle compilation and run 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://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. Ubuntu20.04 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** The official installation document of Ubuntu recommends 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>

## Select Your Platform

Supported:



Ubuntu

Focal

amd64

armhf

arm64



Debian

Buster

amd64

arm64

[Source installation](#)

4) Then use the following script to install ros1

```
orangepi@orangepi5b:~$ install_ros.sh ros1
```

5) Before using the ROS tool, first of all, you need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code





**Note that running the following commands need to ensure that the development board can access GitHub normally, otherwise it will report an error 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 the following errors after the ros1 is installed, please think other methods to allow the linux system of the development board to access GitHub normally. Then manually run the following command.

<https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml>

Hit <https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml>

**ERROR: error loading sources list:**

**The read operation timed out**

```
orange_pi@orange_pi:~$ source /opt/ros/noetic/setup.bash
```

```
orange_pi@orange_pi:~$ sudo rosdep init
```

```
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
```

```
Recommended: please run
```

```
rosdep update
```

```
orange_pi@orange_pi:~$ 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"
```



```

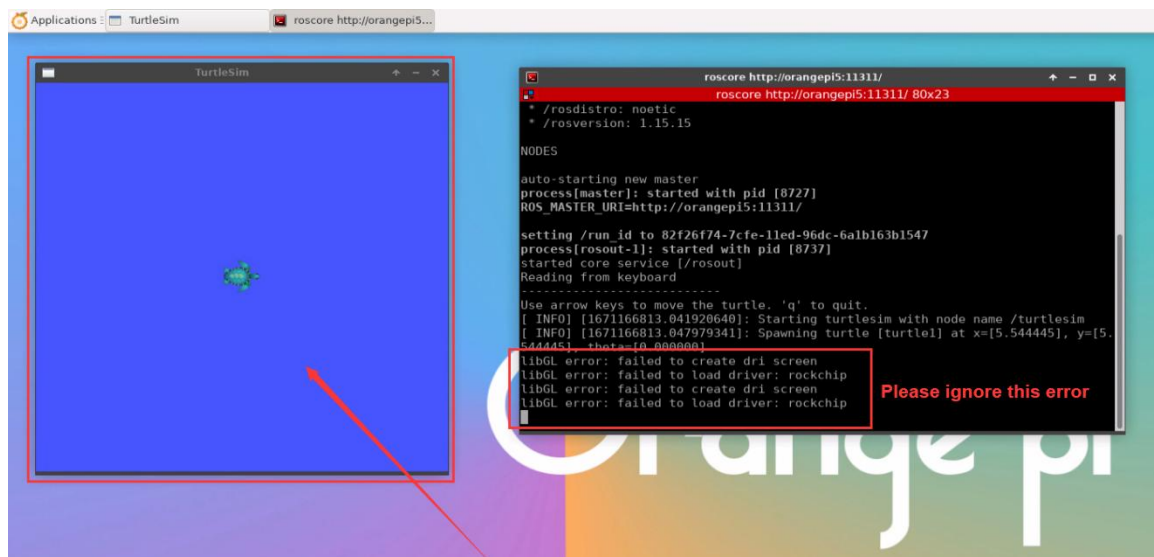
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated 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 the routine of a small turtle to test whether the ROS can be used normally

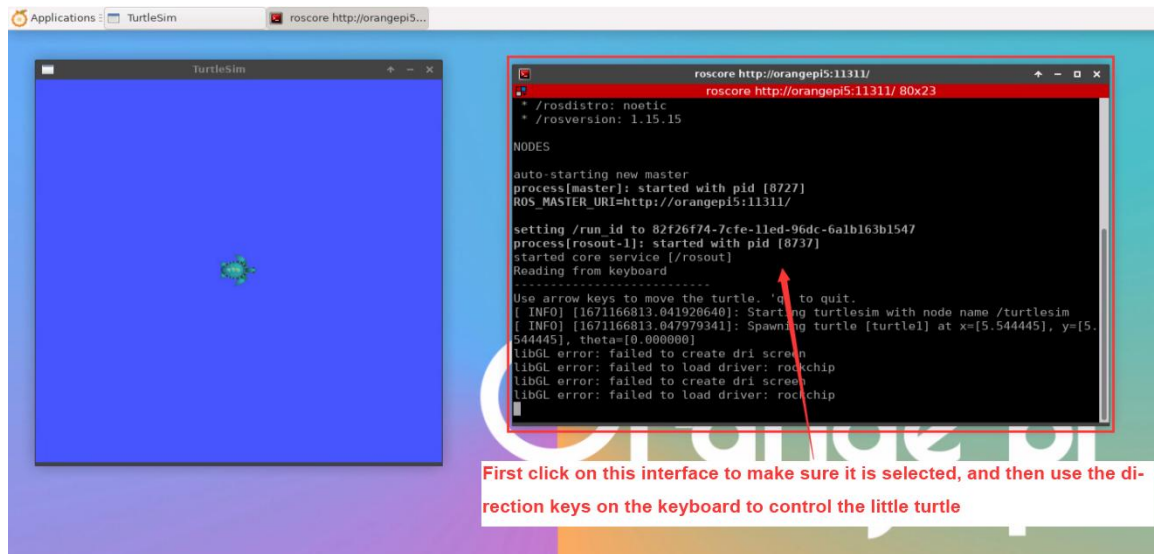
```
orangepi@orangepi:~$ test_ros.sh
```

7) After running the **test\_ros.sh** script, a small turtle shown in the figure below will pop up

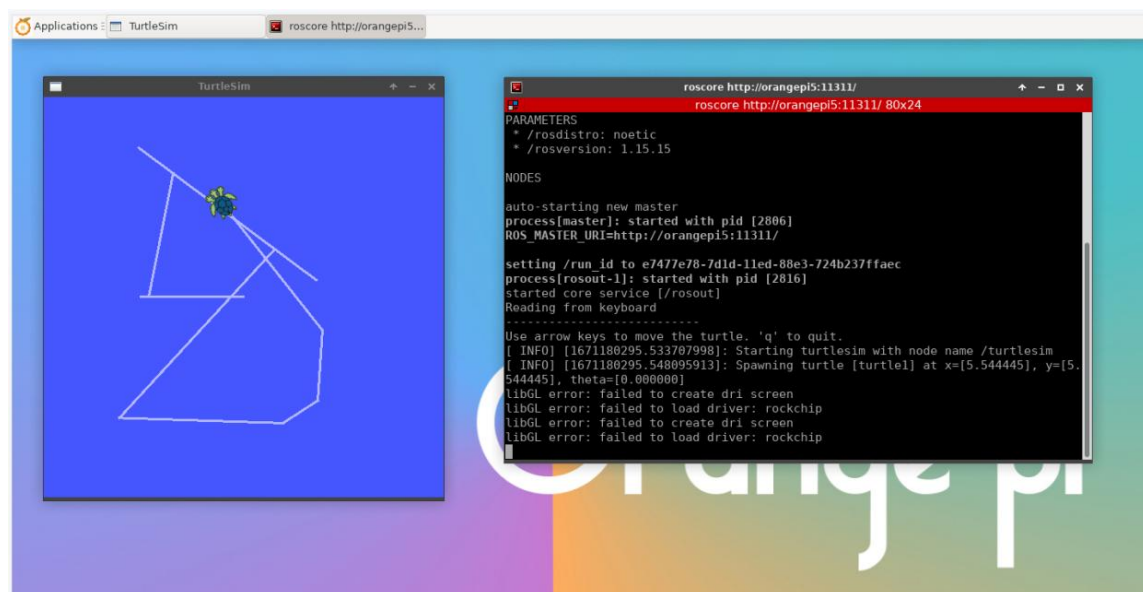


8) Then please keep the terminal window just open at the top



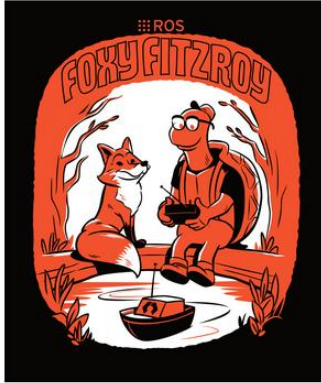


9) At this time, press the direction button on the keyboard to control the small turtles up, down, left and right.



### 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](https://docs.ros.org/en/galactic/Installation.html)

[http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html](https://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html)

3) ROS 2 **Galactic Geochelone**'s official installation document 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 Ros2you Can Install Ros2 with **Install\_ros.sh** Script

```
orange_pi@orange_pi:~$ install_ros.sh ros2
```

5) **install\_ros.sh** script will automatically run the **ros2 -h** command after ROS2. 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 the **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



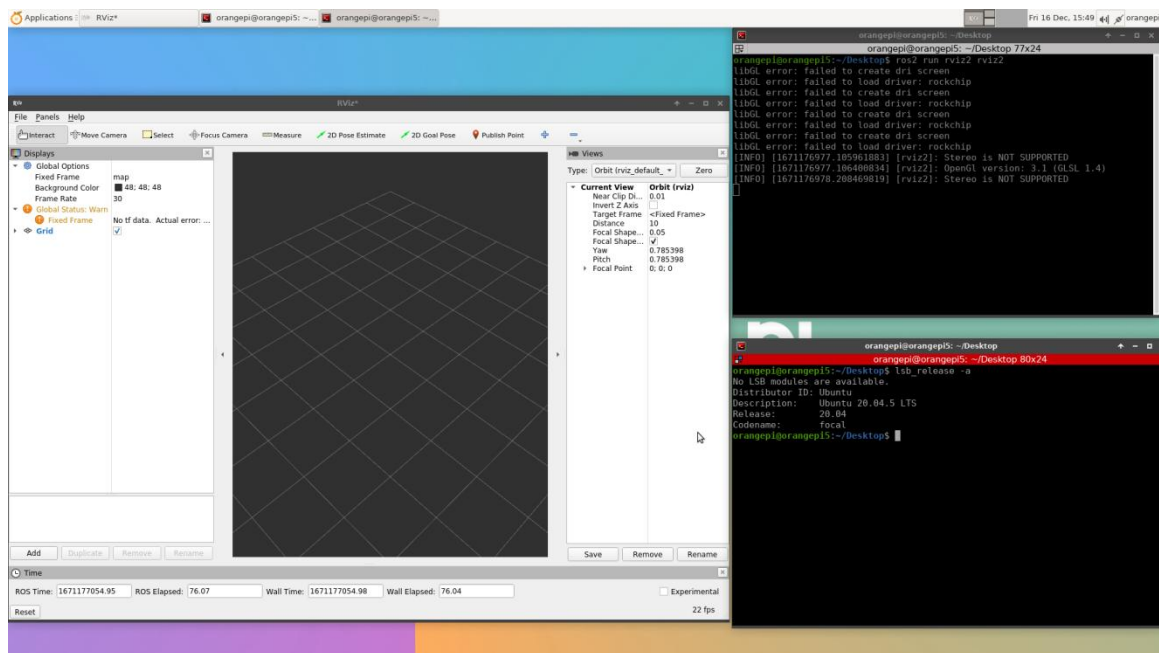
```
orangepi@orangepi5b:~$ test_ros.sh
```

```
[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 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.

3) Then you can use the **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

```
orangeypi@orangepi5b:~$ test_ros.sh
```

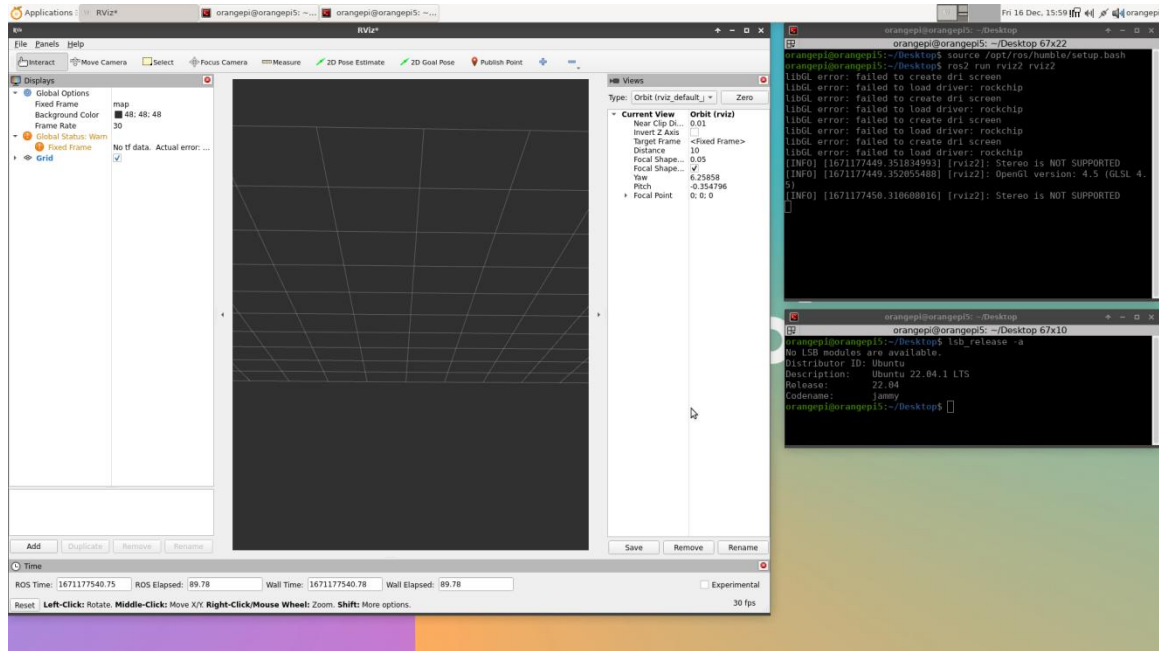
```
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
```



```
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2

```
orangeypi@orangeypi:~$ source /opt/ros/humble/setup.bash
orangeypi@orangeypi:~$ 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. How to install kernel header files

1) The Linux image released by OPi comes with the deb package of the kernel header file by default, and the storage location is `/opt/`

```
orangeypi@orangeypi:~$ ls /opt/linux-headers*
/opt/linux-headers-legacy-rockchip-rk3588_1.x.x_arm64.deb
```

2) Use the following command to install the deb package of the kernel header file

内核头文件 deb 包的名字需要替换为实际的名字，请不要照抄。

```
orangeypi@orangeypi:~$ sudo dpkg -i /opt/linux-headers-legacy-rockchip-rk3588_1.x.x_arm64.deb
```



3) After installation, you can see the folder where the kernel header file is located under **/usr/src**.

```
orangepi@orangepi:~$ ls /usr/src
# If it is a system with Linux 5.10 kernel, the output is as follows
linux-headers-5.10.160-rockchip-rk3588
# If it is a system with Linux 6.1 kernel, the output is as follows
linux-headers-6.1.43-rockchip-rk3588
```

4) Then you can write a hello kernel module to test the kernel header file

a. First write the code of hello kernel module, as shown below:

```
orangepi@orangepi:~$ vim hello.c
#include <linux/init.h>
#include <linux/module.h>

static int hello_init(void)
{
    printk("Hello Orange Pi -- init\n");

    return 0;
}

static void hello_exit(void)
{
    printk("Hello Orange Pi -- exit\n");

    return;
}

module_init(hello_init);
module_exit(hello_exit);

MODULE_LICENSE("GPL");
```

b. Then write the Makefile that compiles the hello kernel module, as shown below:

```
orangepi@orangepi:~$ vim Makefile
ifneq ($(KERNELRELEASE),)
obj-m:=hello.o
```





```

else
KDIR :=/lib/modules/$(shell uname -r)/build
PWD  :=$(shell pwd)
all:
    make -C $(KDIR) M=$(PWD) modules
clean:
    rm -f *.ko *.o *.mod.o *.mod *.symvers *.cmd  *.mod.c *.order
endif

```

- c. Then use the make command to compile the hello kernel module. The output of the compilation process is as follows:

**If there is a problem when compiling the code you copied here, please download the source code from the [official tool](#) and upload it to the Linux system of the development board for testing.**



hello kernel module source code and Makefile

```

orangeypi@orangeypi:~$ make
make -C /lib/modules/5.10.160-rockchip-rk3588/build M=/home/orangeypi modules
make[1]: Entering directory '/usr/src/linux-headers-5.10.160-rockchip-rk3588'
CC [M] /home/orangeypi/hello.o
MODPOST /home/orangeypi/Module.symvers
CC [M] /home/orangeypi/hello.mod.o
LD [M] /home/orangeypi/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.10.160-rockchip-rk3588'

```

- d. After compilation, the **hello.ko** kernel module will be generated.

```

orangeypi@orangeypi:~$ ls *.ko
hello.ko

```

- e. Use the **insmod** command to insert the **hello.ko** kernel module into the kernel.

```

orangeypi@orangeypi:~$ sudo insmod hello.ko

```

- f. Then use the **dmesg** command to view the output of the **hello.ko** kernel module. If you can see the following output, it means that the **hello.ko** kernel module is loaded correctly.

```

orangeypi@orangeypi:~$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init

```

- g. Use the **rmmod** command to uninstall the **hello.ko** kernel module

```

orangeypi@orangeypi:~$ sudo rmmod hello

```

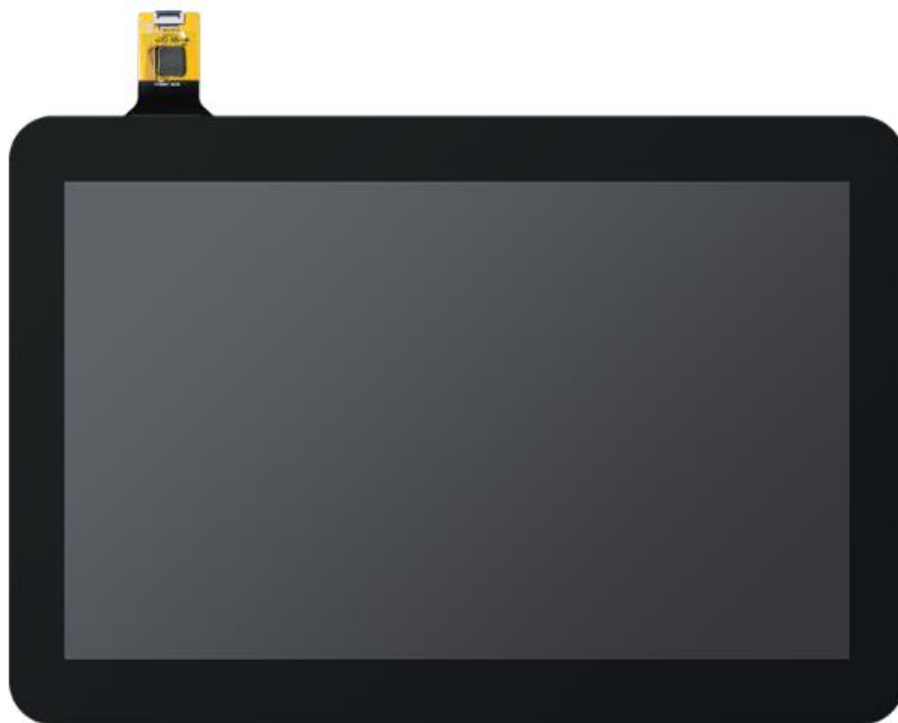


```
orangepi@orangepi:~$ dmesg | grep "Hello"  
[ 2871.893988] Hello Orange Pi -- init  
[ 3173.800892] Hello Orange Pi -- exit
```

### 3. 31. How to use 10.1 inch MIPI LCD screen

#### 3. 31. 1. 10.1 -inch MIPI screen assembly method

- 1) First prepare the required accessories
  - a. 10.1 -inch MIPI LCD display+touch screen



- b. Screen divert plate+31pin to 40pin line



c. 30pin mipi line



d. 12pin touch screen row line



2) According to the figure below, the 12PIN touch screen row, 31PIN to 40PIN ducts, and 30pin MIPI cables get on the screen dial board. **Pay attention to the blue insulation face of the touch screen line line.** If you get a mistake, it will cause no display or unable to touch the problem



3) Place the connected rotor connected to the row line on the MIPI LCD screen, and connect the MIPI LCD screen and the rotary board through 31PIN to 40Pin row



4) Then connect the touch screen and the rotor board through the 12PIN touch screen line, pay attention to the orientation of the insulating surface.



5) Finally connect to the LCD interface of the development board through the 30pin MIPI duct.

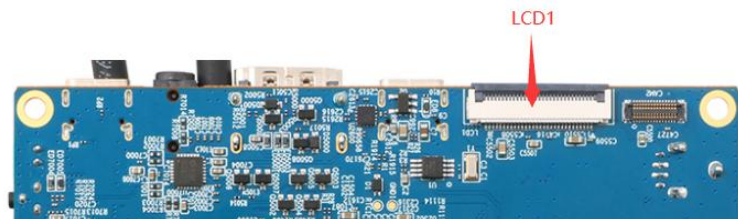


### 3. 31. 2. Open 10.1 -inch MIPI LCD screen configuration method

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 location of the lcd1 interface is:



b. The position of the lcd2 interface is:



**The configuration of the Linux image lcd dtbo of v.1.0.0 and v.1.0.2 versions is opposite to the above definition. Please pay attention when using it.**

**The Linux image of v.1.0.4 and later versions has changed the configuration of LCD DTBO, which is consistent with the LCD serial number displayed on the silk screen on the development board.**

3) The steps of opening the mipi lcd configuration are shown below:

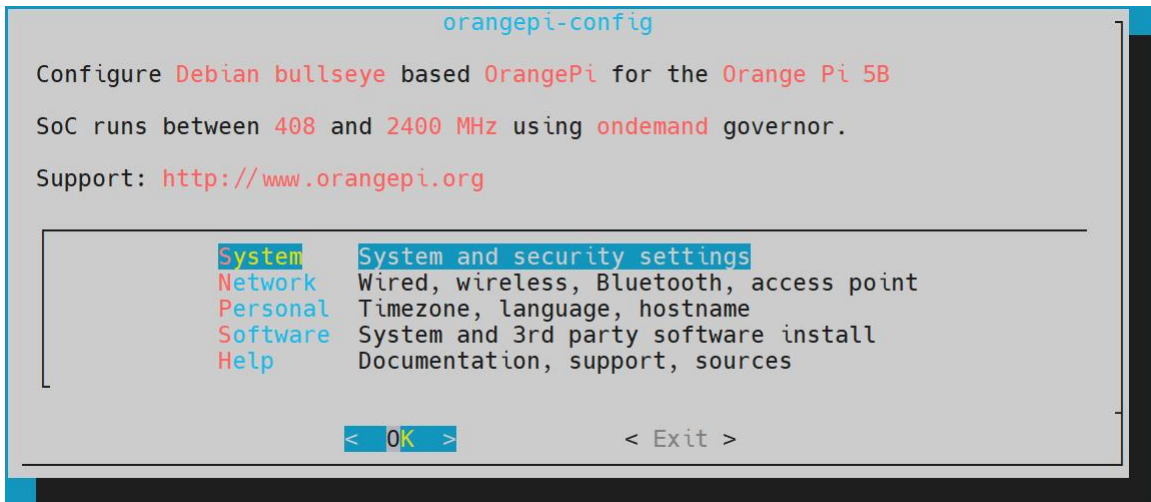




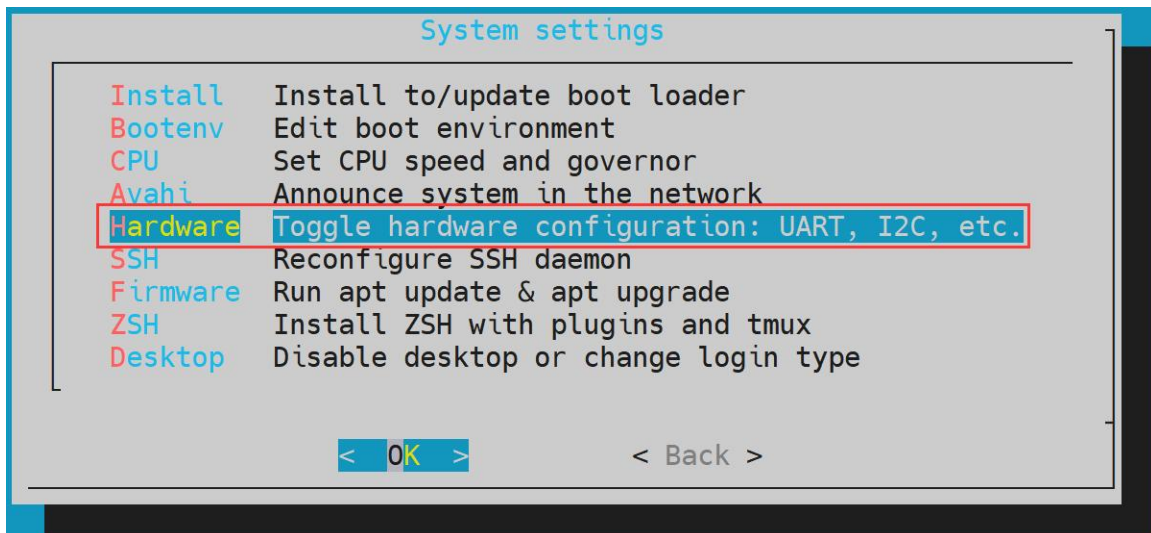
- a. First run the **orange-pi-config**. Ordinary users remember to add **sudo** permissions

```
orange-pi@orange-pi:~$ sudo orange-pi-config
```

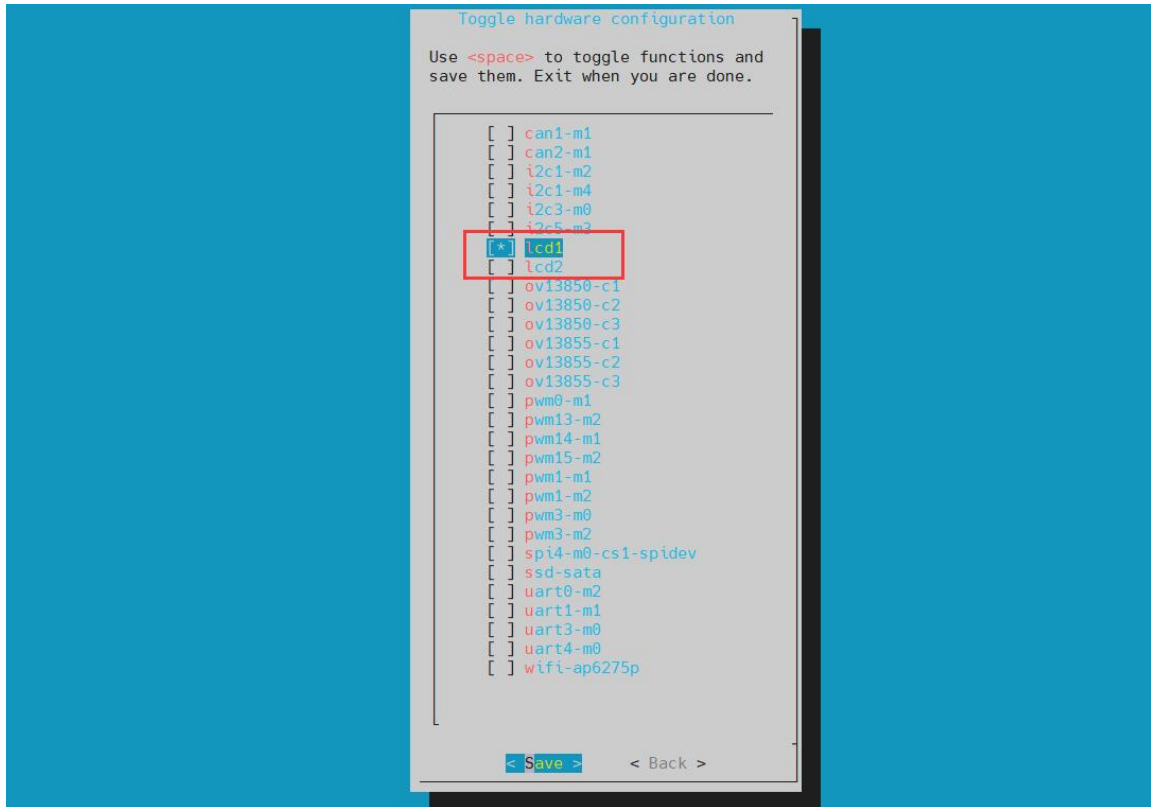
- b. Then choose **System**



- c. Then choose **Hardware**



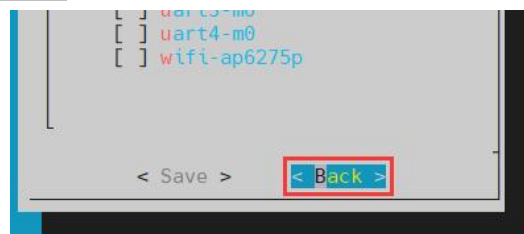
- d. Then use the arrow keys of the keyboard to lcd1 or lcd2 (which one you want to use, and two screens can be opened at the same time), and then use the **space** to select



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

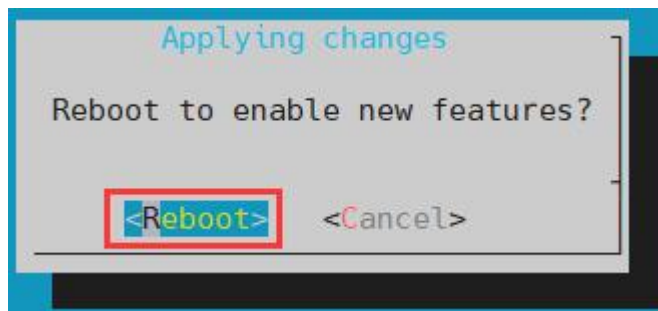


f. Then select **<back>**



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





The above settings will eventually add the **overlays=lcd1** or **overlays=lcd2** or **overlays=lcd1 lcd2** line configuration to `/boot/orangepiEnv.txt`. You can check it first after setting it up. If this line of configuration does not exist, there is something wrong with the settings.

If you find it troublesome to use `orangepi-config`, you can also use the vim editor to open `/boot/orangepiEnv.txt`, and then add the **overlays=lcd1** or **overlays=lcd2** or **overlays=lcd1 lcd2** line configuration.

```
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 (the default vertical screen):



### 3. 31. 3. The server version of the image rotation display direction method

1) Add **Extraargs = fbcon = rotate: The direction to rotate** into **/boot/orangepiEnv.txt**. This configuration can set the direction of the LINUX system of the server version, where the numerals behind **fbcon=rotate** can 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

```
orange@orange:~$ sudo vim /boot/orangepiEnv.txt
overlays=lcd1
extraargs=cma=64M fbcon=rotate:3
```

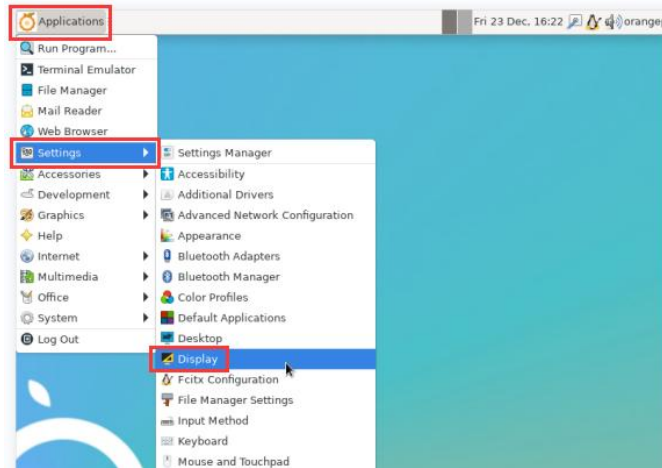
**Note that if /boot/orangepienv.txt is configured in the default default exiArgs = CMA = 64M, FBCon = Rotate: 3 This configuration can be added to extraargs = cma = 64m (need to be separated with spaces).**

2) Then **restart** the Linux system to see that the direction of the lcd screen display has been rotated



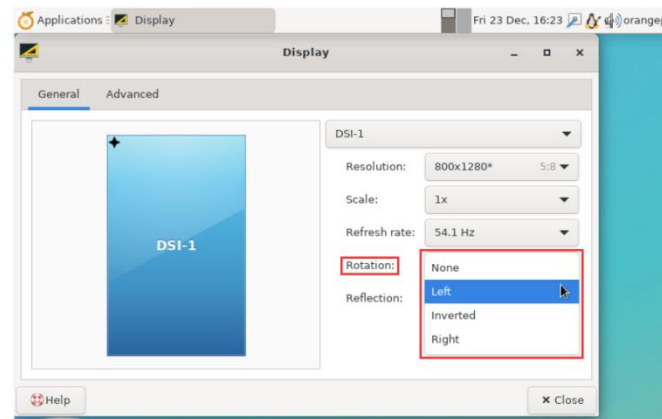
### 3.31.4. The desktop image rotation display and touch direction method

1) First open **Display** settings in the Linux system

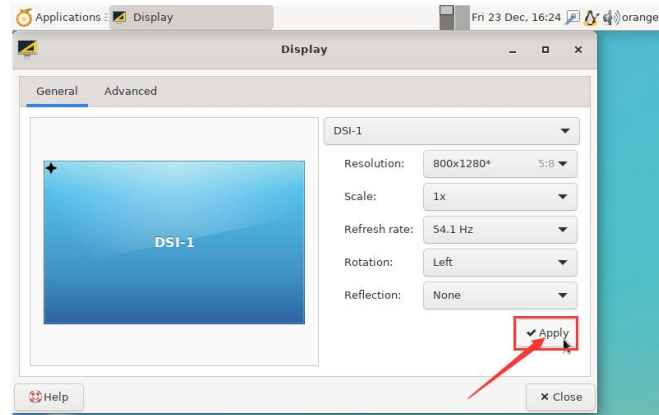


2) Then select the direction you want to rotate in the **Rotation**

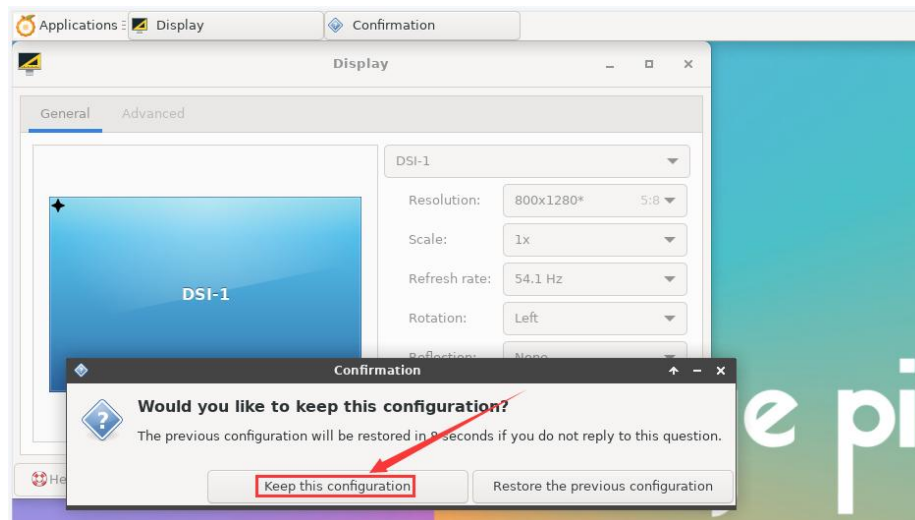
- a. **None**: Not rotate
- b. **Left**: Rotate 90 degrees to the left
- c. **Inverted**: Flipping up and down, equivalent to rotating 180 degrees
- d. **Right**: Rotate 90 degrees to the right



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. Then you can test whether the touch can be used normally.

a. **None**: Not rotate

```
orange_pi@orange_pi:~$ set_lcd_rotate.sh none
```

b. **Left**: Rotate 90 degrees to the left

```
orange_pi@orange_pi:~$ set_lcd_rotate.sh left
```

c. **Inverted**: Flipping up and down, equivalent to rotating 180 degrees



```
orange@orange:~$ set_lcd_rotate.sh inverted
```

d. **Right:** Rotate 90 degrees to the right

```
orange@orange:~$ set_lcd_rotate.sh right
```

**set\_lcd\_rotate.sh** The script mainly does four things:

1. Rotate the direction displayed by Framebuffer
2. The direction of rotating touch
3. Turn off logo
4. Restart the system

The direction of the rotating 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**. Among them, **"x x x x x x x x x"** is different in different directions.

7) References document for touch rotation

<https://wiki.ubuntu.com/X/InputCoordinateTransformation>

### 3. 32. Instructions for opening the logo use

1) The default logo is displayed by default in the desktop version

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the switch to the logo.

```
orange@orange:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=false
```

3) Set up **bootlogo** variable in **/boot/orangepiEnv.txt** to **turn** the switch to the turn -off logo

```
orange@orange:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
```



4) The location of the logo picture in the Linux system is

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

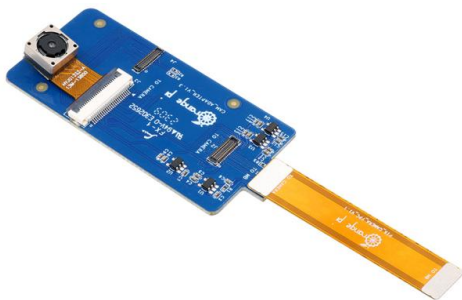
### 3. 33. OV13850 and OV13855 MIPI test methods for testing

**Please note that in Linux 6.1 system, in order to ensure that the 3A service can run normally and obtain normal camera images, the Docker service needs to be disabled. If the Docker service is not disabled, the image captured by the camera will not contain the 3A effect and will appear as a dark image. Here's how to disable Docker:**

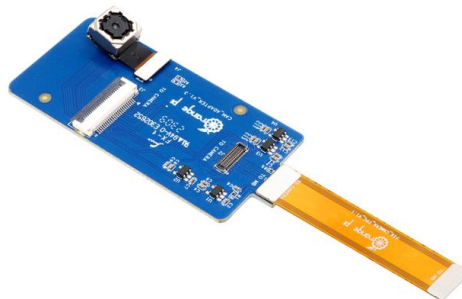
```
orangepi@orangepi:~$ sudo systemctl disable docker.socket docker.service containerd.service  
orangepi@orangepi:~$ sudo reboot
```

At present, the development board supports two MIPI cameras, OV13850 and OV13855, and the specific pictures are shown below:

a. OV13850 camera at 13MP MIPI interface



b. OV13855 camera at 13MP MIPI interface



The rotary board used by OV13850 and OV13855 cameras is the same as the FPC cable, but the two cameras are different from the position on the rotary board. The FPC exhaust line is shown in the following figure. Please note that the FPC line is directed.

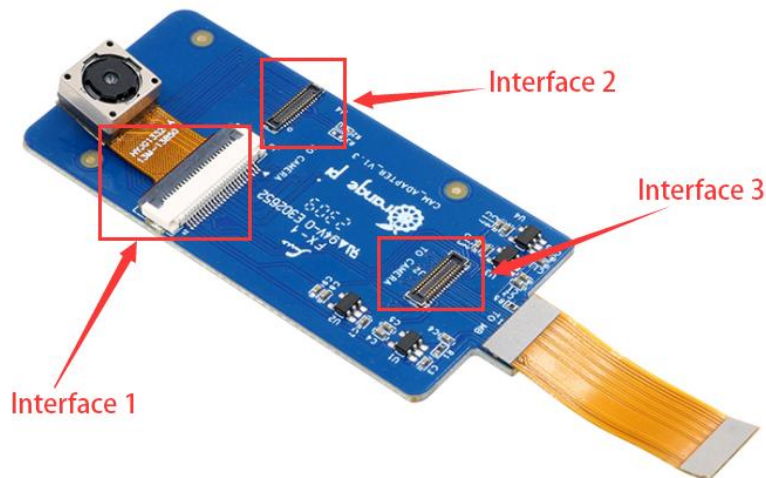


The end marked **TO MB** that it needs to be inserted into the camera interface of the development board. The end marked **TO CAMERA** needs to be inserted on the camera to the board.

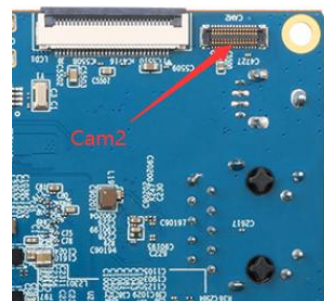
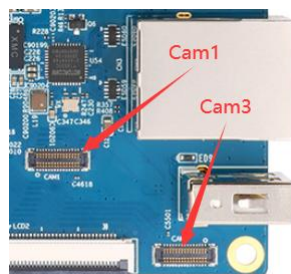


There are a total of 3 cameras on the camera to connect to the board, which can only be used at the same time, as shown in the figure below, about:

- a. **1 interface OV13850 camera**
- b. **2 interface OV13855 camera**
- c. 3 interface is not used, just ignore it



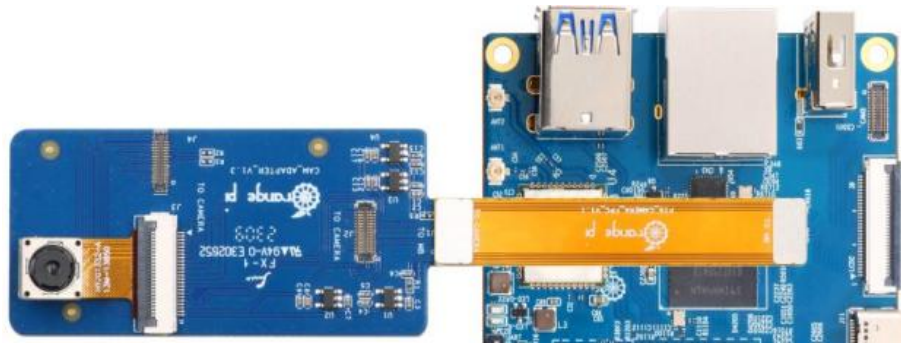
Orange Pi 5B development board has a total of 3 camera interfaces. We define the positions of CAM1, CAM2, and CAM3 as shown in the figure below:







The method of the Cam1 interface inserted in the camera is shown below:



The method of the Cam2 interface inserted in the camera is shown below:



The method of the Cam3 interface inserted in the camera is shown below:

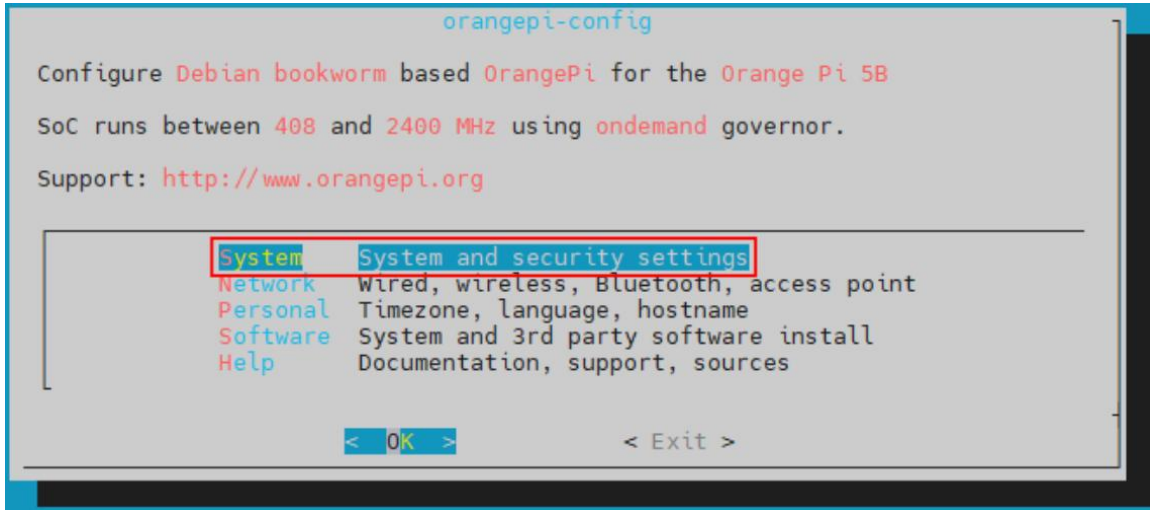


After connecting the camera to the development board, we can use the following method to test the next camera:

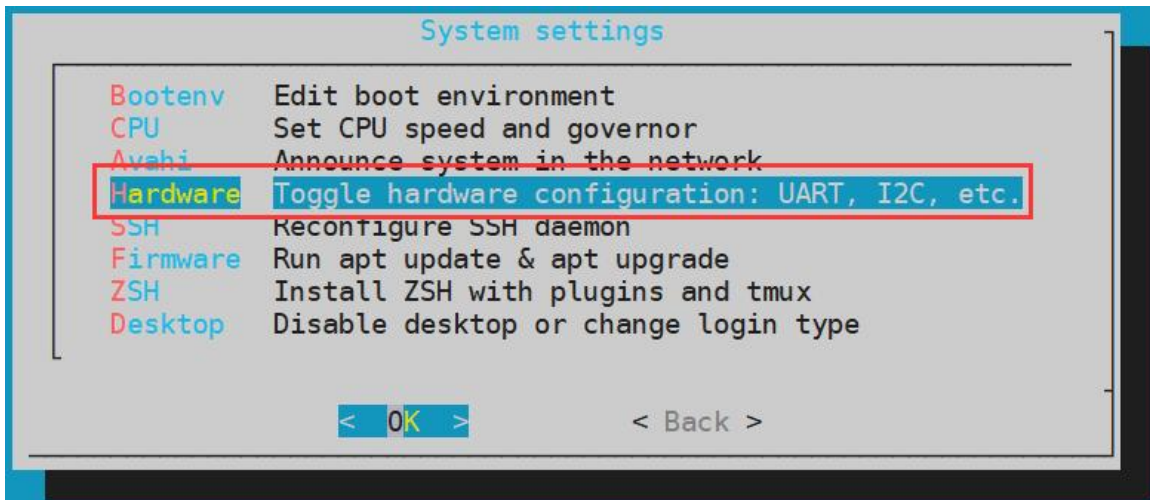
- First run the **orangepi-config**. Ordinary users remember to add **sudo** permissions

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

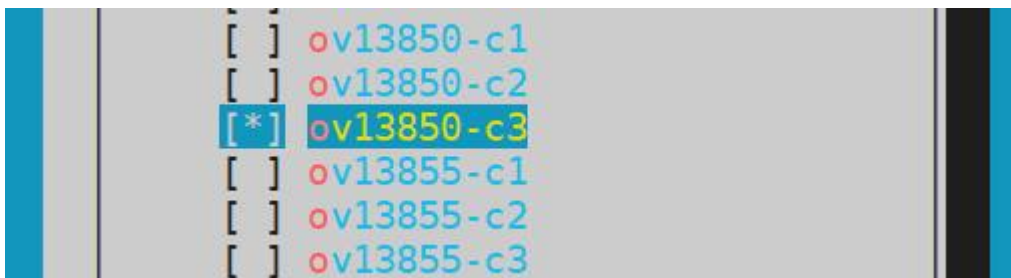
- Then choose **System**



- c. Then choose **Hardware**



- d. Then use the direction key of the keyboard to position the position shown in the figure below, and then use the **space** to select the camera you want to open. Among them, **ov13850-c1** indicates that the **ov13855-c2** camera is used in the CAM1 interface of the development board. Use an OV13855 camera in the CAM2 interface, and other configurations can be pushed.



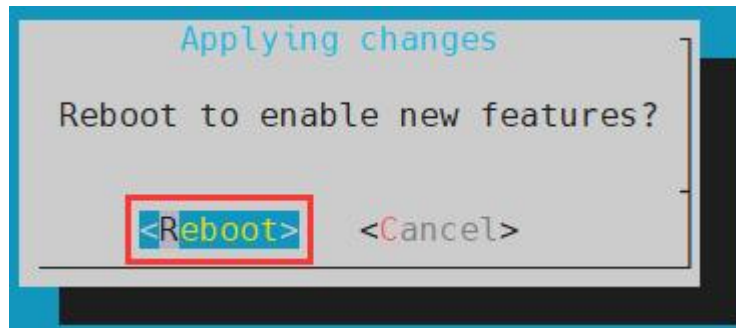
- e. Then select **<Save>** Save



- f. Then select **<Back>**



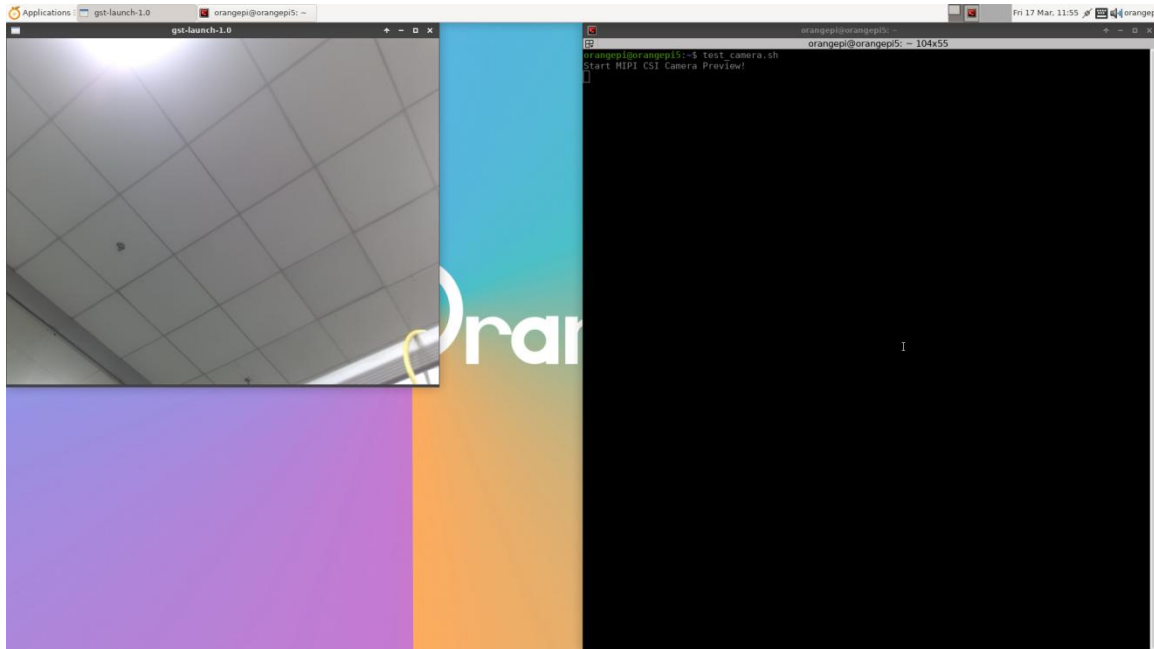
- g. Then select the **<Reboot>** restart system to make the configuration take effect



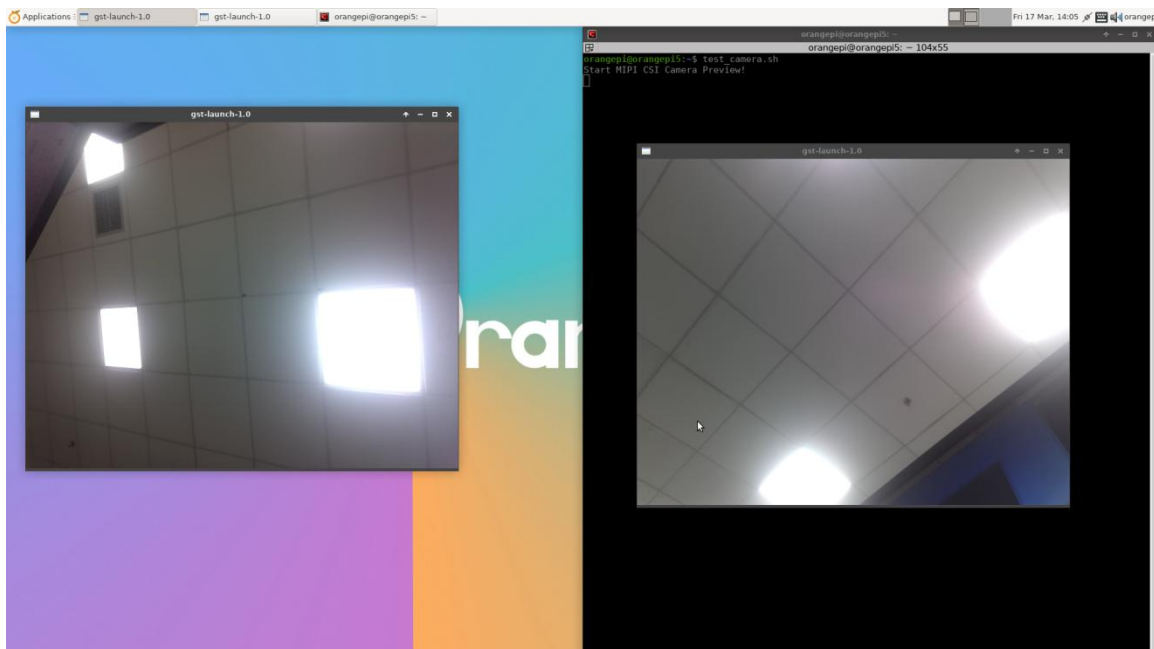
- h. Then open a terminal in the desktop system and run the script below

```
orange@orange:~$ test_camera.sh
```

- i. Then you can see the preview of the camera



In addition to single cameras, we can also use two cameras at the same time. It should be noted that the current test dual camera should be used for combinations of **Cam1+Cam3** (supporting OV13850 and OV13855 mix and match). After receiving the dual camera, like the previous steps, open the configuration of the Cam1+Cam3 through **orange-pi-config**, restart the system, and then open the terminal on the desktop to run the **test\_camera.sh** script to see the preview screen of the two cameras, as follows Shown in the figure:





Please refer to the link below for the camera dts configuration. If you need it, you can modify it by yourself;

<https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-camera1.dtsi>

<https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-camera2.dtsi>

<https://github.com/orangepi-xunlong/linux-orangepi/blob/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-5-camera3.dtsi>

dt overlay configuration is in the directory below:

<https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk3588/arch/arm64/boot/dts/rockchip/overlay>

### 3. 34. How to use the ZFS file system

The latest versions of Ubuntu20.04, Ubuntu22.04, Debian11 and Debian12 **desktop systems** have zfs pre-installed and can be used directly.

The pre-installed zfs version in Ubuntu20.04 and Ubuntu22.04 desktop systems is **2.1.6**.

The pre-installed zfs version in Debian11 and Debian12 desktop systems is **2.1.11**.

After the system starts, please first confirm whether the zfs kernel module has been loaded. If you can see zfs-related content using the lsmod command, it means that the system has pre-installed zfs.

```
orangepi@orangepi:~$ lsmod | grep "zfs"
```

zfs	2801664	0
zunicode	327680	1 zfs
zzstd	471040	1 zfs
zlua	139264	1 zfs
zcommon	69632	1 zfs
znvpair	61440	2 zfs,zcommon
zavl	16384	1 zfs



icp	221184	1 zfs
spl	77824	6 zfs,icp,zzstd,znvpair,zcommon,zavl

### 3.34.1. How to install ZFS

**Before installing zfs, please make sure that the Linux image you are using is the latest version. In addition, if zfs is already installed in the system, there is no need to install it again.**

Before installing zfs, you first need to install the kernel header files. For how to install the kernel header files, please refer to the instructions in the section "[How to Install the Kernel Header Files.](#)"

In Ubuntu20.04, Ubuntu22.04 and Debian11 systems, zfs cannot be installed directly through apt. This is because the zfs version in the default apt source is lower than 2.1.6 and is incompatible with the rk linux5.10 kernel. This problem is fixed in zfs 2.1.6 and later versions

In order to solve this problem, we provide a deb package of zfs that can be installed normally, which can be downloaded from the [official tool](#) of the development board. Open the [official tool](#) and enter the [zfs-related deb package folder used by Ubuntu and Debian systems](#). You can see three types of deb packages for Ubuntu20.04, Ubuntu22.04 and Debian11. Please download the required version.



After downloading the corresponding version of the zfs deb package, please upload them to the Linux system of the development board. For the upload method, please refer to the instructions in the [Methods of Uploading Files to the Development Board Linux System.](#)

After the upload is completed, use the `cd` command on the command line of the development board Linux system to enter the directory of the deb package, and then use the following command to install the zfs deb package.

```
orangeapi@orangepi:~$ sudo apt install ./*.deb
```





After the installation is complete, use the following command to see the zfs-related kernel modules:

```
orangePi@orangePi:~$ ls /lib/modules/5.10.160-rockchip-rk3588/updates/dkms/
icp.ko  spl.ko  zavl.ko  zcommon.ko  zfs.ko  zlua.ko  znvpair.ko  zunicode.ko
zzstd.ko
```

If it is a Linux 6.1 system, use the following command to see the zfs-related kernel modules:

```
orangePi@orangePi:~$ ls /lib/modules/6.1.43-rockchip-rk3588/updates/dkms/
icp.ko  spl.ko  zavl.ko  zcommon.ko  zfs.ko  zlua.ko  znvpair.ko  zunicode.ko
zzstd.ko
```

Then restart the Linux system and you will see that the zfs kernel module will be automatically loaded:

```
orangePi@orangePi:~$ lsmod | grep "zfs"
zfs                2801664  0
zunicode           327680  1 zfs
zzstd              471040  1 zfs
zlua               139264  1 zfs
zcommon            69632  1 zfs
znvpair            61440  2 zfs,zcommon
zavl               16384  1 zfs
icp                221184  1 zfs
spl                77824  6 zfs,icp,zzstd,znvpair,zcommon,zavl
```

In Debian12, the default version of zfs is 2.1.11, so we can install zfs directly through the following command. Again, we need to make sure that the system has the deb package of the kernel header file installed before installation.

```
orangePi@orangePi:~$ sudo apt install -y zfsutils-linux zfs-dkms
```

### 3. 34. 2. How to create a ZFS pool

**ZFS is based on storage pools. We can add multiple physical storage devices to the pool and then allocate storage space from this pool.**

**The following content is demonstrated based on the development board being connected to an NVMe SSD and a USB flash drive.**

1) First, we can use the **lsblk** command to view all storage devices on the development





board. Currently, the development board is connected to an NVMe SSD and a USB flash drive. The output is as follows:

```
orangepi@orangepi:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    1  28.8G  0 disk
├─sda1       8:1    1  28.8G  0 part
└─sda9       8:9    1    8M  0 part
mtdblock0    31:0    0   16M  0 disk
mmcblk0     179:0    0  29.7G  0 disk
├─mmcblk0p1 179:1    0    1G  0 part /boot
└─mmcblk0p2 179:2    0  28.4G  0 part /var/log.hdd
zram0        254:0    0   7.7G  0 disk [SWAP]
zram1        254:1    0  200M  0 disk /var/log
nvme0n1      259:0    0 476.9G  0 disk
├─nvme0n1p1 259:3    0 476.9G  0 part
└─nvme0n1p9 259:4    0    8M  0 part
orangepi@orangepi:~$
```

2) Then enter the following command to create a ZFS pool, including two storage devices: NVMe SSD and USB flash drive.

```
orangepi@orangepi:~$ sudo zpool create -f pool1 /dev/nvme0n1 /dev/sda
```

3) Then use the **zpool list** command to see that the system has created a ZFS pool named **pool1**, and the size of ZFS pool **pool1** is the size of the NVME SSD plus the size of the USB flash drive.

```
orangepi@orangepi:~$ zpool list
NAME      SIZE  ALLOC  FREE  CKPOINT  EXPANDSZ  FRAG    CAP  DEDUP  HEALTH  ALTROOT
pool1    504G   114K   504G      -          -         0%    0%   1.00x  ONLINE  -
```

4) Then execute **df -h** and you can see that **pool1** is mounted to the **/pool1** directory.

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           1.6G   18M   1.6G   2% /run
/dev/mmcblk0p2  29G   6.0G   22G  22% /
tmpfs           7.7G   46M   7.7G   1% /dev/shm
tmpfs           5.0M   4.0K   5.0M   1% /run/lock
tmpfs           7.7G  944K   7.7G   1% /tmp
/dev/mmcblk0p1 1022M  115M   908M  12% /boot
/dev/zram1      188M   4.5M  169M   3% /var/log
tmpfs           1.6G   80K   1.6G   1% /run/user/1000
pool1          489G   9.3M  489G   1% /pool1
```



5) Use the following command to see that the file system type of pool1 is zfs

```
orangepi@orangepi:~$ mount | grep pool1
pool1 on /pool1 type zfs (rw,xattr,noacl)
```

6) Then we can test copying a file to the ZFS pool

```
orangepi@orangepi:~$ sudo cp -v /usr/local/test.mp4 /pool1/
'/usr/local/test.mp4' -> '/pool1/test.mp4'
```

### 3. 34. 3. Test the data deduplication function of ZFS

1) The data deduplication function of ZFS is turned off by default. We need to execute the following command to turn it on.

```
orangepi@orangepi:~$ sudo zfs set dedup=on pool1
```

2) Then do a simple test, first enter pool1, and then execute the following command to generate a random file of 1G size

```
orangepi@orangepi:~$ cd /pool1/
root@orangepi:/pool1$ sudo dd if=/dev/urandom of=test.1g bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 5.04367 s, 213 MB/s
```

3) Then use the following command to copy 1000 copies of a random file of 1G size

```
root@orangepi:/pool1$ for ((i=0; i<1000; i++)); do sudo cp test.1g $i.test.1g; done
```

4) Then use **du -lh** to see that there is currently a total of 1002G of data in the pool, but in fact the size of the ZFS pool is only **504GB** (the total capacity of SSD + U disk), which cannot hold such large data.

```
root@orangepi:/pool1$ du -lh
1002G
```

5) Then use the **zpool list** command to see that only 1.01G is actually occupied, because these 1001 files are duplicates, indicating that the data deduplication function is effective.

```
orangepi@orangepi:/pool1$ zpool list
NAME      SIZE  ALLOC  FREE  CKPOINT  EXPANDSZ  FRAG    CAP  DEDUP  HEALTH  ALTROOT
pool1     504G  1.01G  503G    -          -         0%    0%    6.00x  ONLINE  -
```

### 3. 34. 4. Test the data compression function of ZFS

1) Because the stored data is different, the disk space saved by compression will also be



different, so we choose to compress a relatively large plain text file for the compression test. Execute the following command to package the `/var/log/` and `/etc/` directories. into tarball

```
orange@orange:~$ cd /pool1/
root@orange:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

2) Then the file size that can be seen through the `ls -lh` command and the space occupied in the ZFS pool are both **27M**

```
orange@orange:/pool1$ ls -lh
total 27M
-rw-r--r-- 1 root root 27M Jun  1 14:46 text.tar
orange@orange:/pool1$ zpool list
NAME      SIZE  ALLOC   FREE CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1    504G  26.7M   504G      -          -         0%    0%    1.00x    ONLINE  -
orange@orange:/pool1$
```

3) Then we enable compression in ZFS pool pool1

```
root@orange:/pool1$ sudo zfs set compression=lz4 pool1
```

4) Then execute the following command again to package the `/var/log/` and `/etc/` directories into a tar package

```
root@orange:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

5) At this time, you can see that the `text.tar` file size is still 27M, but it only occupies 9.47M space in the ZFS pool, indicating that the file is compressed.

```
orange@orange:/pool1$ ls -lh
total 9.2M
-rw-r--r-- 1 root root 27M Jun  1 14:54 text.tar
orange@orange:/pool1$ zpool list
NAME      SIZE  ALLOC   FREE CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1    504G  9.47M   504G      -          -         0%    0%    1.00x    ONLINE  -
orange@orange:/pool1$
```

### 3. 35. How to install and use CasaOS

**CasaOS is an open source home cloud system based on the Docker ecosystem, which allows you to run a variety of home applications on your own development board, such as NAS, home automation, media servers, etc.**

#### 3. 35. 1. CasaOS installation method

1) First you need to install docker. Docker is already pre-installed in the system released by Orange Pi. This step can be skipped. You can use the following command to check



the version of docker installed.

```
orangepi@orangepi:~$ docker --version  
Docker version 24.0.2, build cb74dfc      # Ubuntu Jammy 系统的输出
```

2) Then enter the following command in the linux system to start the installation of CasaOS

```
orangepi@orangepi:~$ curl -fsSL https://get.casaos.io | sudo bash
```

3) When you see the terminal outputting the following print information, it means that CasaOS has been installed.

```
CasaOS v0.4.4.2 is running at:
```

```
Open your browser and visit the above address.
```

```
CasaOS Project : https://github.com/IceWhaleTech/CasaOS
```

```
CasaOS Team    : https://github.com/IceWhaleTech/CasaOS#maintainers
```

```
CasaOS Discord : https://discord.gg/knqAbbBbeX
```

```
Website       : https://www.casaos.io
```

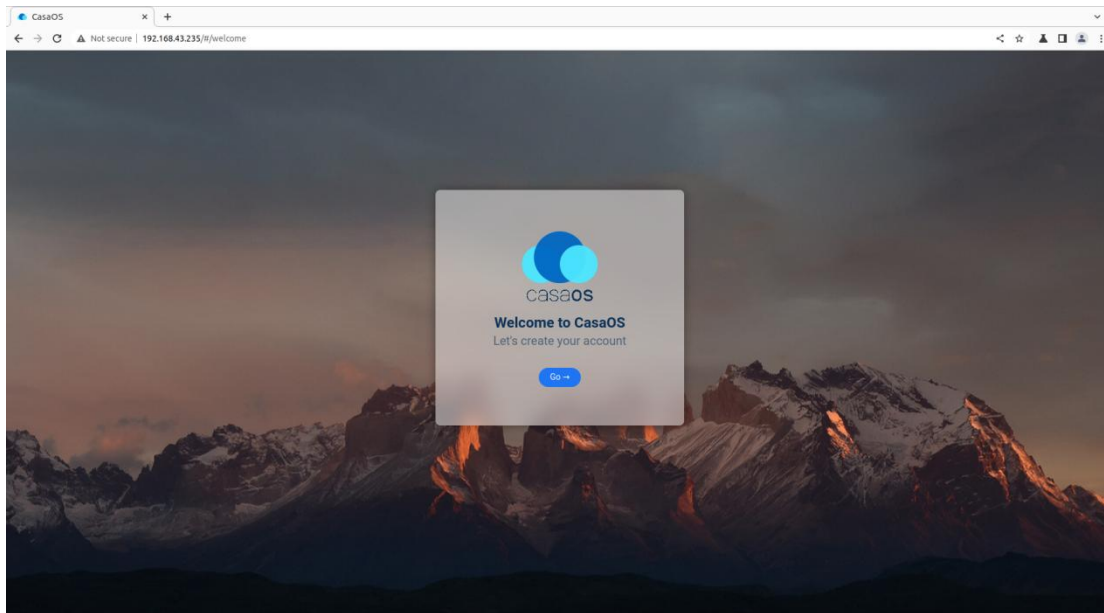
```
Online Demo   : http://demo.casaos.io
```

```
Uninstall    : casaos-uninstall
```

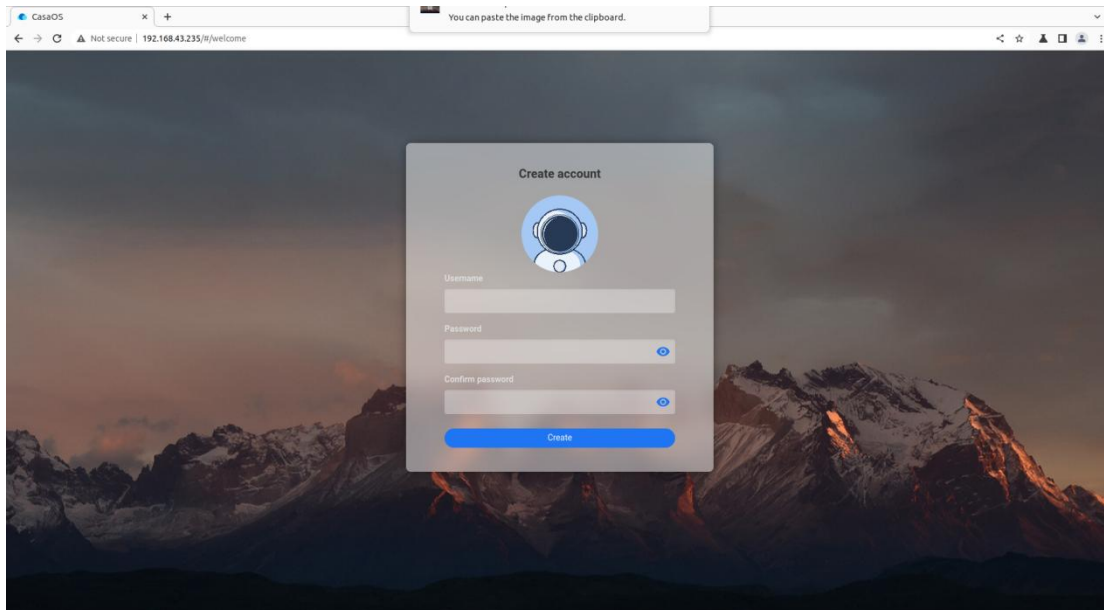
### 3.35.2. How to use CasaOS

1) After installing CasaOS, enter **http://the IP address of the development board** in the browser to open CasaOS

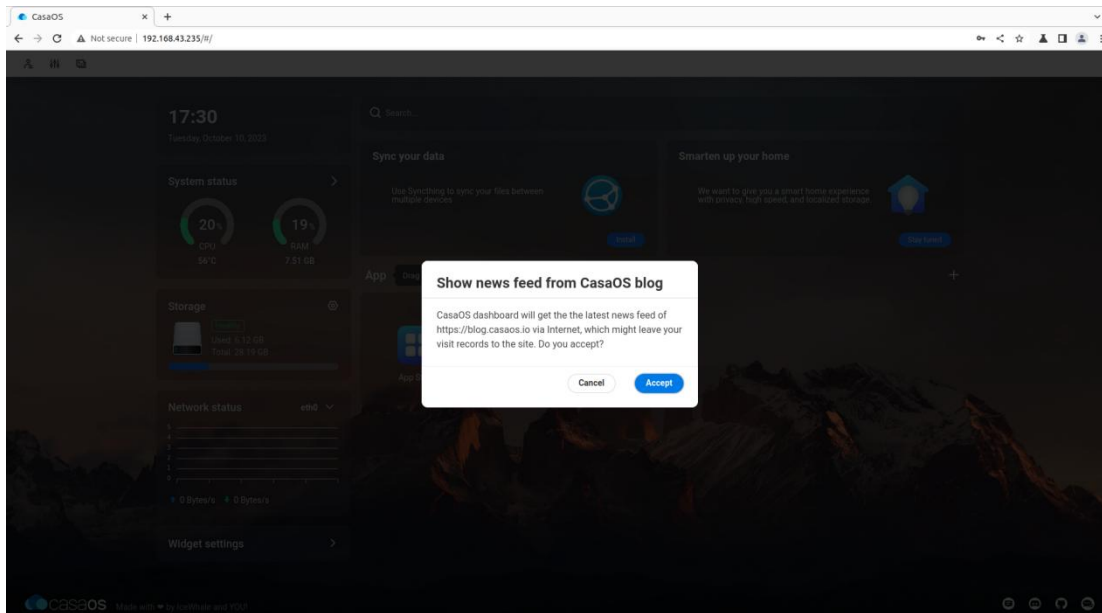
2) After opening CasaO, the following welcome interface will pop up. Click "Go" to proceed to the next step.



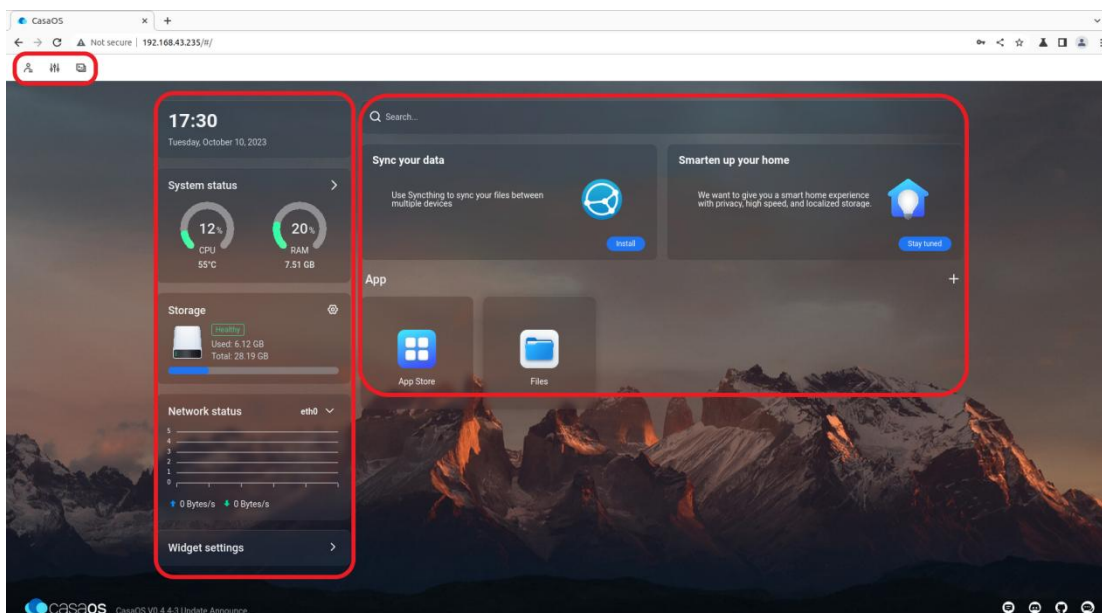
3) When you log in to CasaOS for the first time, the login interface is the interface for setting the account and password. When you log in again, only the interface for entering the account and password will appear. After setting the account and password, click "Create" to proceed to the next step.



4) Click "Accept" directly in the interface below to proceed to the next step.

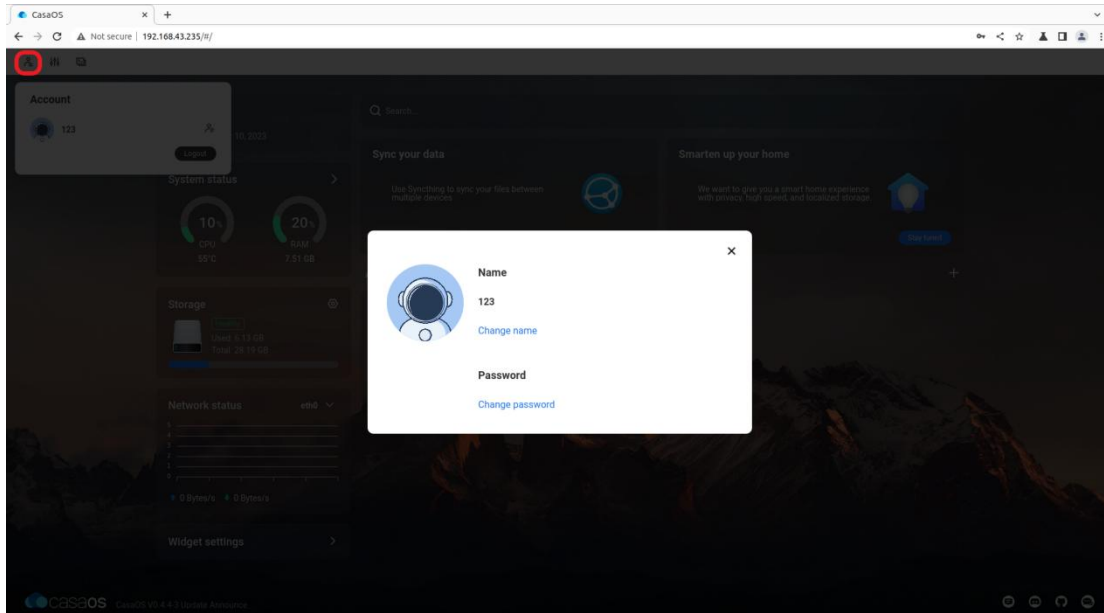


5) Now enter the main page of CasaOS. There are three icons in the upper left corner for function settings. On the left is the performance panel, which can display the current time and status information of CPU, RAM, storage, and network. On the right is the function panel. It has functions such as search, application recommendation, application store and file management.

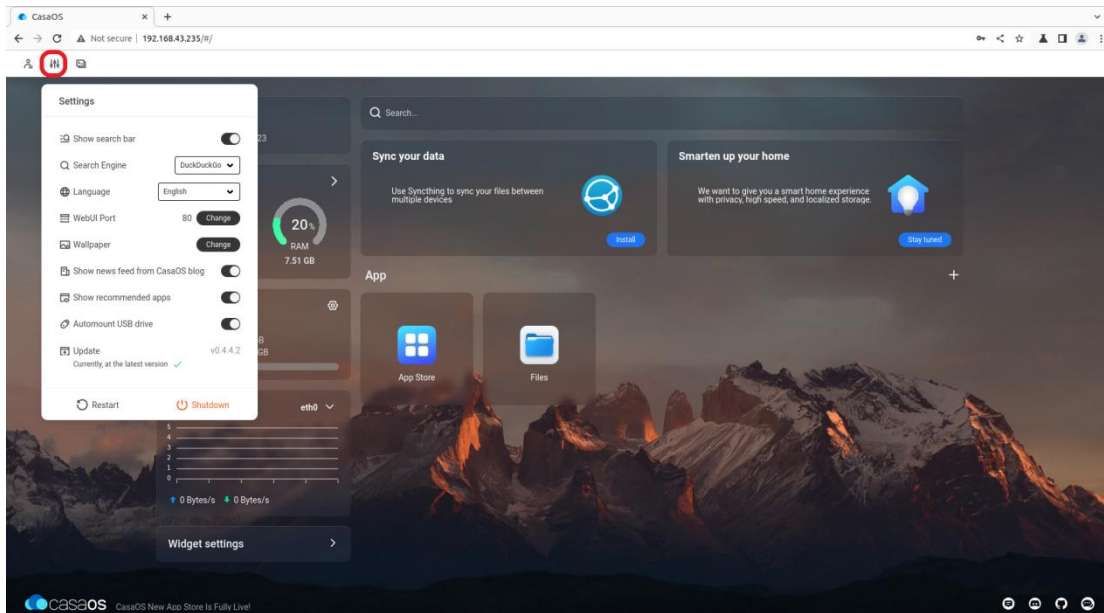


6) You can click the first icon in the upper left corner to modify the account number and password



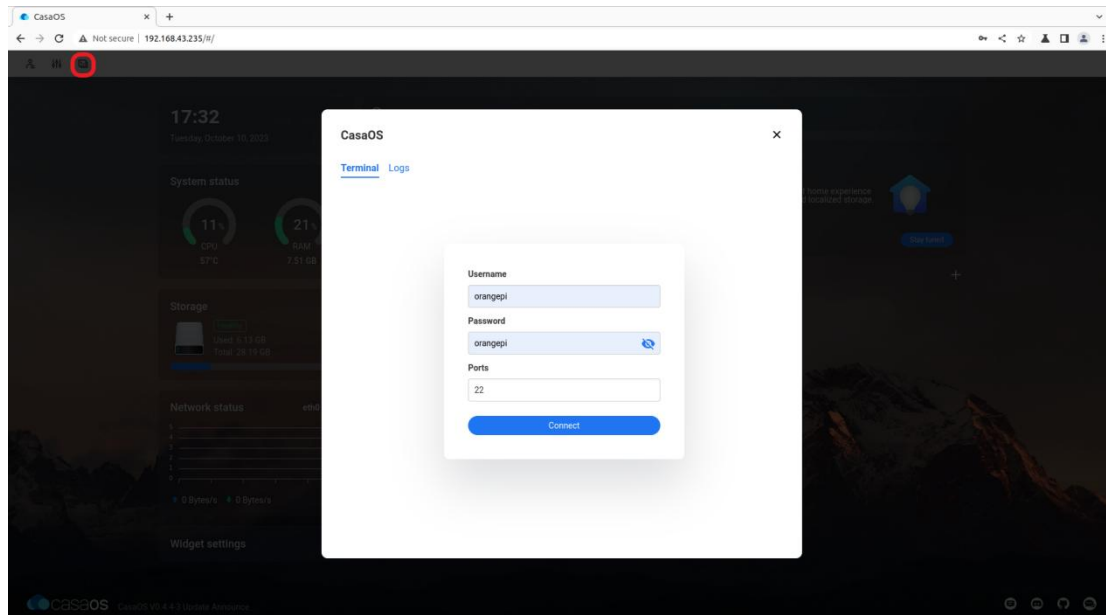


7) You can click the second icon to set basic functions

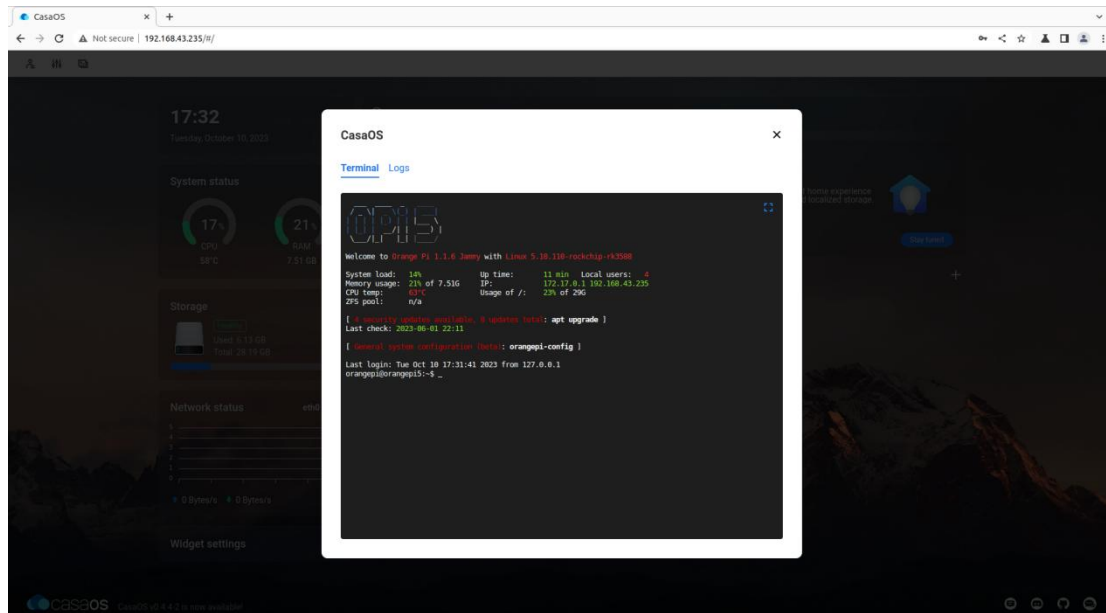


8) The third icon in the upper left corner mainly has two functions, namely switching to command line mode and printing log information. When switching to command line mode, you need to enter your account and password. The account and password here refer to the development board. Linux system account and password, the port system defaults to number 22

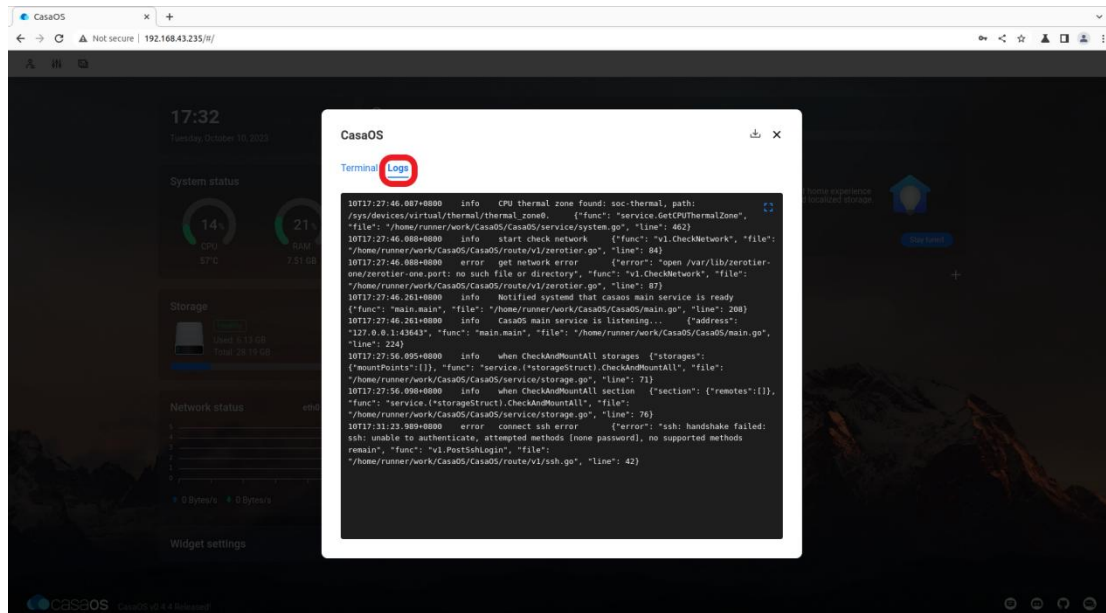




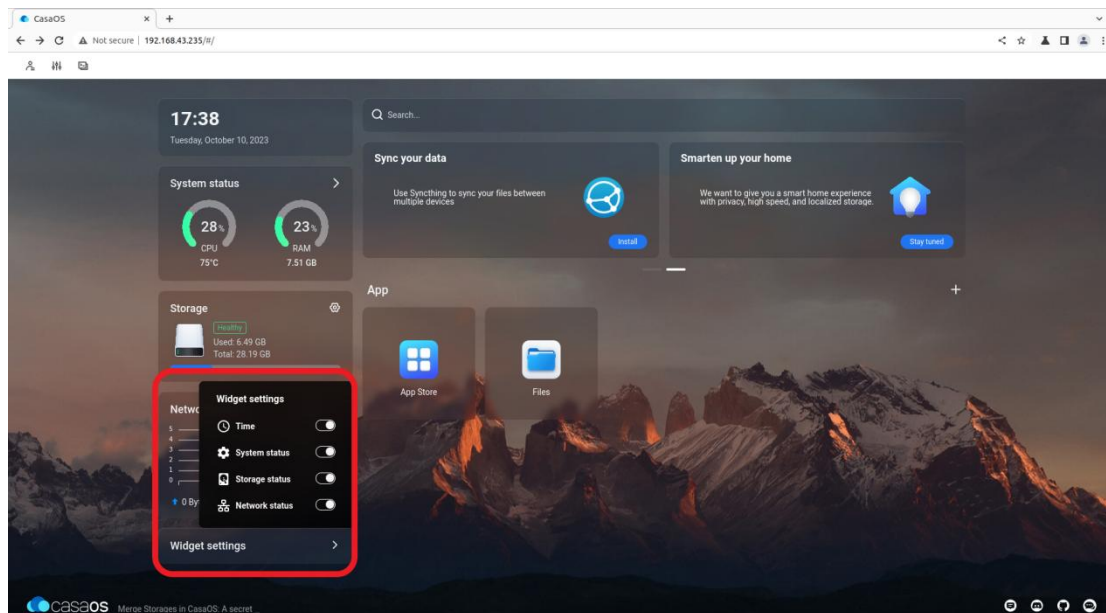
9) Then click "Connect" to enter the command line interface:



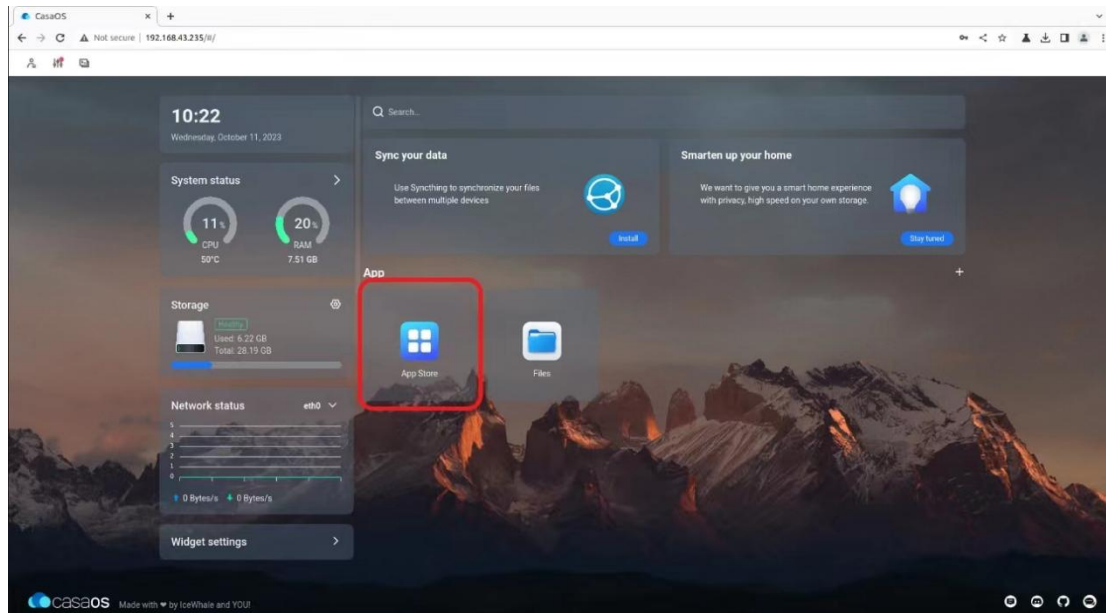
10) Another function under the third icon is to print CasaOS logs. Click "Logs" to enter. The interface is as follows:



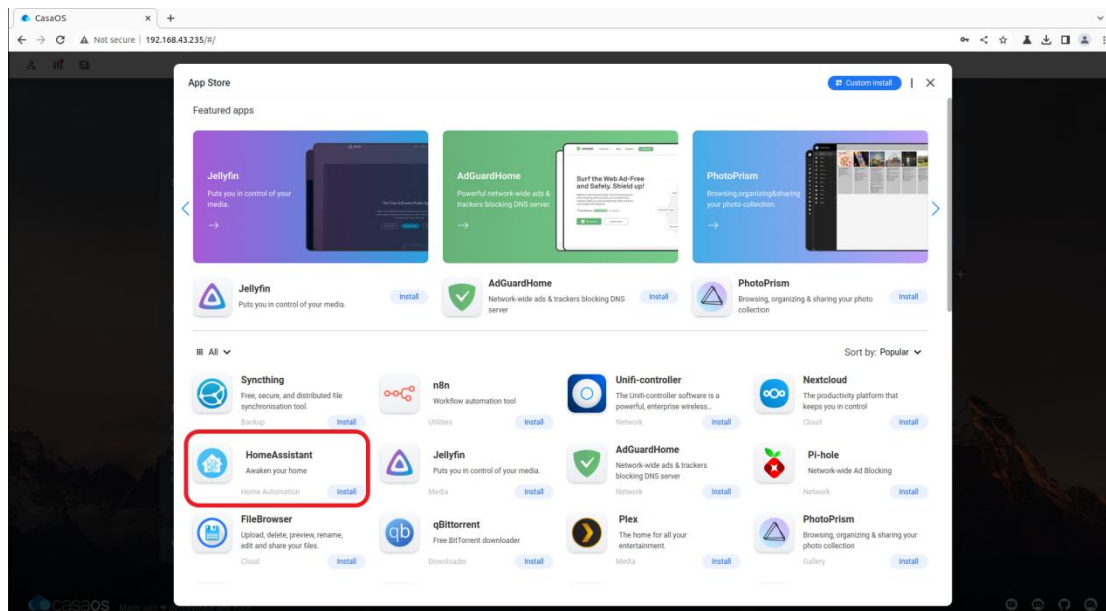
11) Click "Widget settings" in the lower left corner to set whether to display the widgets of the performance panel on the main page.



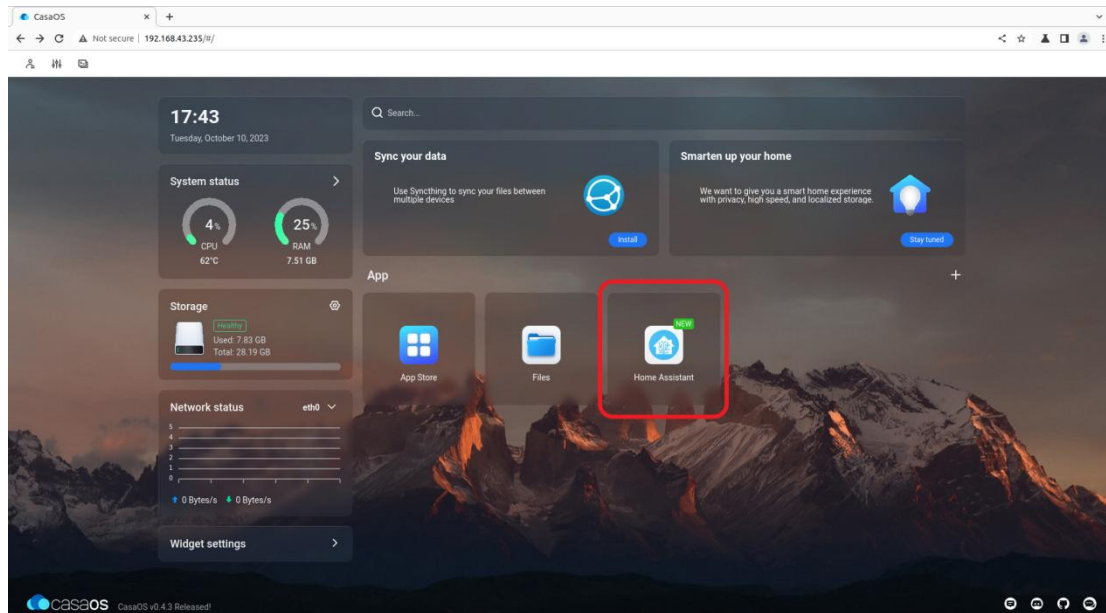
12) Click "APP Store" on the main interface to open the app store. Currently, there are a total of 70+ APPs available in the app store.



13) Here we take Home Assistant as an example to download, find Home Assistant in the APP Store, and then click the corresponding "install"

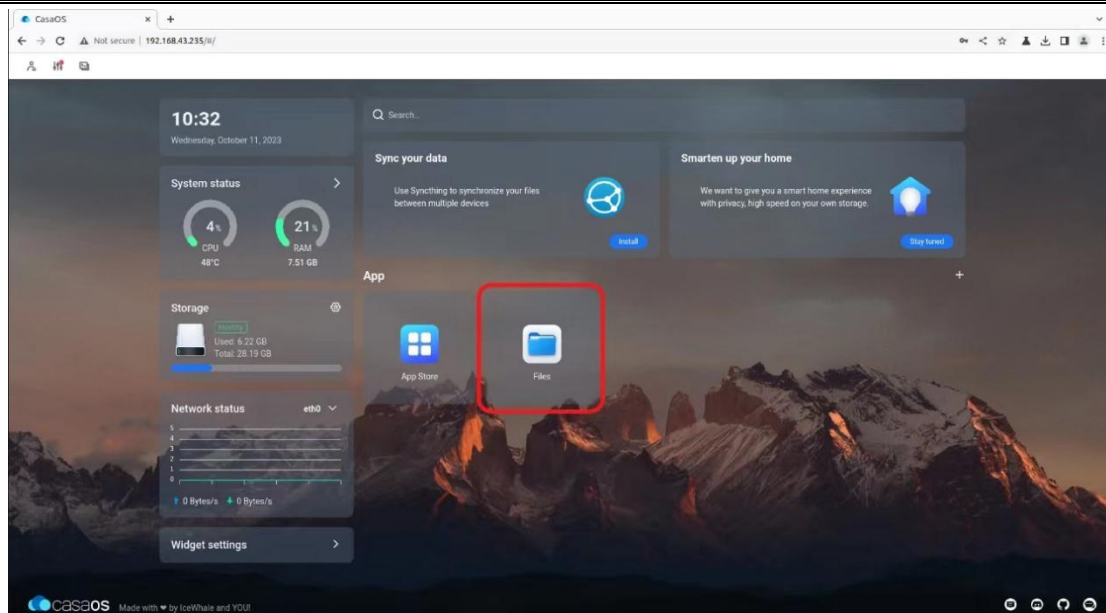


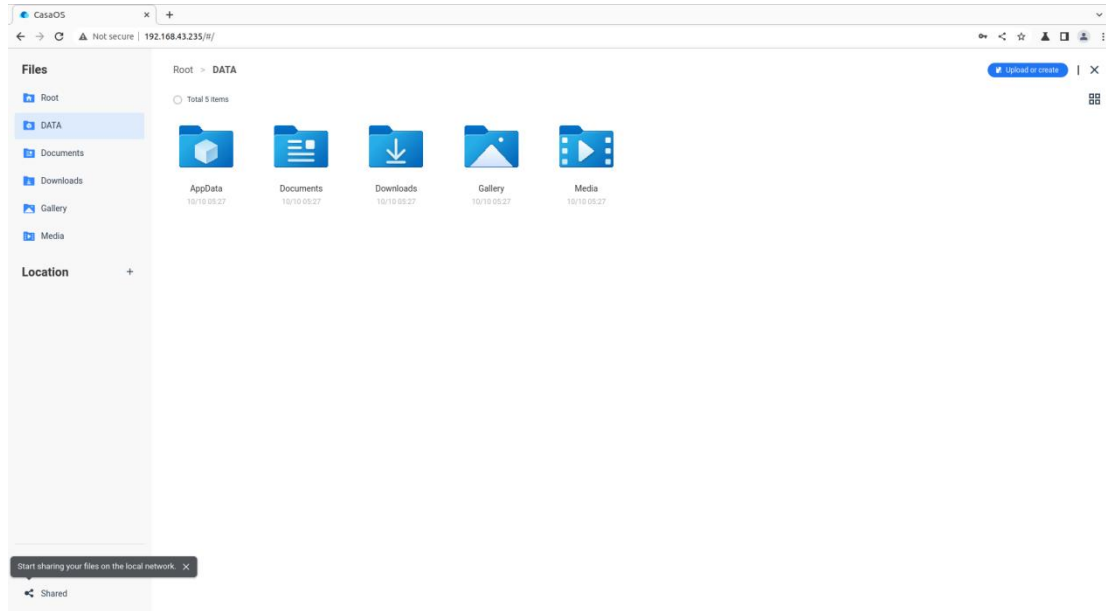
14) After the download is completed, HostAssitant will appear on the main page.



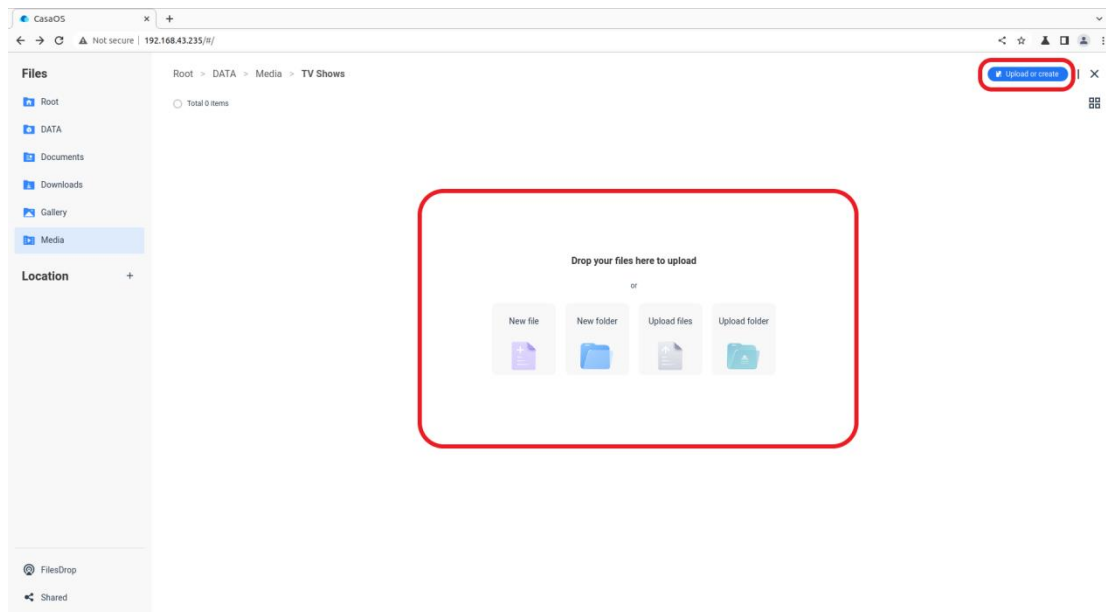
15) Click "Files" in the main interface to open the file system that comes with CasaOS, and then you can upload and save files.

**Please ensure that other devices and the development board are in the same LAN.**





16) When uploading files, you need to switch to the target folder, then drag the local file to the indicated area in the picture, or click "Upload or Create" in the upper right corner to select the file to upload.



17) If you want to uninstall CasaOS, you can use the following command:

```
orange@orange5:~$ curl -fsSL https://get.icewhale.io/casaos-uninstall.sh |bash
```

### 3. 36. Methods of using NPU

#### 3. 36. 1. Preparation tools

- 1) A PC with Ubuntu20.04 operating system installed

**According to the official documentation of RKNN-Toolkit2, the operating systems supported by the current version of RKNN-Toolkit2 are as follows:**

- a. Ubuntu18.04 (x64)**
- b. Ubuntu20.04 (x64)**
- c. Ubuntu22.04 (x64)**

**In this document, we use Ubuntu20.04 (x64) operating system for demonstration. Please test other versions of operating systems by yourself.**

- 2) An Orange Pi 5B development board with Debian 11 system installed

- 3) A data cable with Type-C interface for using adb function



#### 3. 36. 2. Install RKNN-Toolkit2 on Ubuntu PC

**Toolkit2 is a development kit used on the Ubuntu PC platform. Users can use the Python interface provided by the tool to easily complete functions such as model conversion, inference, and performance evaluation.**

- 1) On the Ubuntu PC side, open a command line window and enter the following commands to install python3 and pip3

```
test@test:~$ sudo apt-get install python3 python3-dev python3-pip
```

- 2) You can use the following command to view the installed version of python3

```
test@test:~$ python3 --version
```



```
Python 3.8.10
```

3) Then enter the following command to install the dependency package of RKNN-Toolkit2

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 \
libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

4) Then enter the following command to download the 1.5.2 version of RKNN-Toolkit2

```
test@test:~$ git clone https://github.com/rockchip-linux/rknn-toolkit2 -b v1.5.2
```

5) Then enter the following command to install the dependency package of the corresponding version of python3. This command will use the dependencies listed in the pip3 installation file requirements\_cp38-1.5.2.txt

```
test@test:~$ pip3 install -r rknn-toolkit2/doc/requirements_cp38-1.5.2.txt -i \
https://mirror.baidu.com/pypi/simple
```

6) Then enter the following command to use pip3 to install the RKNN-Toolkit2 software package. After the installation is complete, you can use RKNN-Toolkit2

```
test@test:~$ pip3 install rknn-toolkit2/packages/rknn_toolkit2-1.5.2+b642f30c-cp38-cp38-linux_x86_64.whl
```

### 3. 36. 3. Use RKNN-Toolkit2 for model conversion and model inference

**RKNN-Toolkit2 supports converting Caffe, TensorFlow, TensorFlow Lite, ONNX, DarkNet, PyTorch and other models into RKNN models, and then runs the RKNN model through simulation on the Ubuntu PC or using the NPU of the development board for inference.**

**Relevant examples are provided in the example folder of RKNN-Toolkit2 to help users better understand how to operate. We take the ONNX model with yolov5 function as an example to illustrate.**

#### 3. 36. 3. 1. Simulate running model on Ubuntu PC

**RKNN-Toolkit2 is equipped with a built-in simulator, which allows users to simulate the inference process of the model on the Rockchip NPU on the Ubuntu PC.**

**In this way, model conversion and inference can be completed on the Ub**





**untu PC side, helping users test and verify their models faster.**

1) First switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

2) Then run the test.py script, which first converts the yolov5s\_relu.onnx model into an RKNN model that can be run on the simulator, and then uses the simulator to simulate and run the model to perform inference on the bus.jpg image in the current directory

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

3) After the test.py script is successfully run, you will see the following print information, indicating that the model successfully detected four people and a bus in the bus.jpg picture.

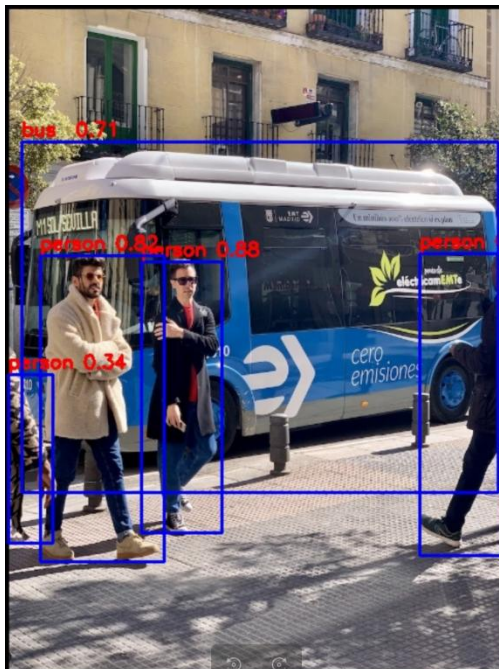
```
done
--> Running model
W inference: The 'data_format' has not been set and defaults is nhwc!
done
class: person, score: 0.884139358997345
box coordinate left,top,right,down: [209.1040009856224, 244.4304337501526, 286.5742521882057,
506.7466902732849]
class: person, score: 0.8676778078079224
box coordinate left,top,right,down: [478.5757632255554, 238.58572268486023, 559.5273861885071,
526.479279756546]
class: person, score: 0.8246847987174988
box coordinate left,top,right,down: [110.57257843017578, 238.58099019527435,
230.54625701904297, 534.0008579492569]
class: person, score: 0.3392542004585266
box coordinate left,top,right,down: [79.96397459506989, 354.9062474966049, 122.13020265102386,
516.2529321908951]
class: bus , score: 0.7012234926223755
box coordinate left,top,right,down: [94.43931484222412, 129.53470361232758, 553.1492471694946,
468.0852304697037]
D NPUTransfer: Transfer client closed, fd = 3
```

4) The converted model file yolov5s\_relu.rknn and the inference picture result result.jpg



are saved in the current directory

5) The result.jpg picture shows the object categories and confidence rates detected in the bus.jpg picture using the yolov5s\_relu.rknn model.



### 3. 36. 3. 2. Using the NPU running model of the development board on Ubuntu PC

RKNN-Toolkit2 provides users with a Python interface for using the NPU of the development board for inference through adb. It allows users to use the NPU of the development board on the Ubuntu PC to run the model for inference.

In this way, the Ubuntu PC side can use the machine learning library provided by Python to optimize and adjust the model based on the actual effect when the model is run on the NPU of the development board.

#### 3. 36. 3. 2. 1. Use Type-C data cable to connect adb

Use adb to operate the development board on the Ubuntu PC. For how to use adb, please see the instructions in the section "[How to use ADB](#)".



### 3.36.3.2.2. Update the rknn\_server and librknrt.so of the development board

**librknrt.so is a board-side runtime library.**

**rknn\_server is a background proxy service running on the development board. It is used to receive the protocol transmitted from the PC through USB, then execute the corresponding interface in the board-side runtime library, and return the results to the PC.**

1) First enter the following command on Ubuntu PC to download version 1.5.2 of RKNPU2

```
test@test:~$ git clone https://github.com/rockchip-linux/rknpu2 -b v1.5.2
```

2) Then enter the following command on the Ubuntu PC to update the rknn\_server of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/rknn_server/aarch64/usr/bin/* /usr/bin
```

3) Then enter the following command on the Ubuntu PC side to update the librknrt.so library of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/librknn_api/aarch64/librknrt.so /usr/lib
```

4) Open the terminal of the development board through the adb tool

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

5) Open the rknn\_server service of the development board

```
root@orange5:/# restart_rknn.sh
root@orange5:/# start rknn server,version:1.5.2(8babfeabuild@2023-08-25T10:30:31)
I NPUTransfer: Starting NPU TransferServer,Transfer version 2.1.0(b5861e7@2020-11-23T11:50:51)
```

6) You can use the following command to check. If the process ID of rknn\_server appears, it means that rknn\_server has been opened, so that the operating environment of the development board is set up.



```
root@orangepi5:/# pgrep rknn_server
3131
```

### 3. 36. 3. 2. 3. Modify the parameters in the example

1) On the Ubuntu PC side, you can view the device ID of the development board connected to the Ubuntu PC by running the following command. This ID will be used below.

```
test@test:~$ adb devices
List of devices attached
4f9f859e5a120324    device
```

2) Switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

3) Use vim editor to modify the test.py file

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ vim test.py
```

4) In the test.py file, we need to modify the following content:

- a. In the preprocessing configuration, change the target platform to rk3588, so that after model conversion, you will get an RKNN model suitable for the NPU of the RK3588S development board.

```
# pre-process config
print('--> Config model')
rknn.config(mean_values=[[0, 0, 0]], std_values=[[255, 255, 255]], target_platform='rk3588')
print('done')
```

- b. In the initialization running environment, add a description of the target platform and device ID. The target platform is rk3588, and the device ID is the device ID of the development board obtained through adb. The operation of running the model for inference will be on the RK3588S development board. Performed on NPU

```
# Init runtime environment
print('--> Init runtime environment')
ret = rknn.init_runtime(target='rk3588', device_id='4f9f859e5a120324')
if ret != 0:
    print('Init runtime environment failed!')
    exit(ret)
print('done')
```

c. After modification, save and exit.

### 3. 36. 3. 2. 4. Run the example on Ubuntu PC

1) Enter the following command to run the test.py script. The script first converts the yolov5s\_rel.u.onnx model to an RKNN model, and then loads the model to the NPU of the development board to perform inference on the out.jpg image in the current directory.

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

2) In the printed information, we can see that Ubuntu PC uses the NPU of the development board to run the model for inference through the adb tool.

```
--> Init runtime environment
I target set by user is: rk3588
I Check RK3588 board npu runtime version
I Starting ntp or adb, target is RK3588
I Device [4f9f859e5a120324] not found in ntb device list.
I Start adb...
I Connect to Device success!
I NPUTransfer: Starting NPU Transfer Client, Transfer version 2.1.0
(b5861e7@2020-11-23T11:50:36)
```

3) After the test.py script runs successfully, the converted model file yolov5s\_rel.u.rknn and the inference image result result.jpg are saved in the current directory.

4) The running result is the same as the section on simulating [the running model on Ubuntu PC](#).



### 3. 36. 4. Call the C interface to deploy the RKNN model and run it on the development board

RKNPU2 provides a C programming interface for chip platforms with Rockchip NPU, which can help users deploy RKNN models exported using RKNN-Toolkit2 and accelerate the implementation of AI applications.

In the example folder of RKNPU2, examples of deploying RKNN models with different functions to the development board are provided. We take deploying the RKNN model with yolov5 function to the RK3588 Debian 11 platform as an example to illustrate.

#### 3. 36. 4. 1. Download cross-compilation tools

Since the development board runs a Linux system, it needs to be compiled using the gcc cross compiler. It is recommended to use gcc-9.3.0-x86\_64\_aarch64-linux-gnu this version of gcc

Enter the following command to download this version of gcc. After downloading, you will get a folder named gcc-buildroot-9.3.0-2020.03-x86\_64\_aarch64-rockchip-linux-gnu

```
test@test:~$ git clone https://github.com/airockchip/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
```

#### 3. 36. 4. 2. Modify the compilation tool path in the script

1) Switch to the rknpu2/examples/rknn\_yolov5\_demo directory

```
test@test:~$ cd ~/rknpu2/examples/rknn_yolov5_demo
```

2) Use vim editor to modify the content in the build-linux\_RK3588.sh file

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ vim build-linux_RK3588.sh
```

3) In the build-linux\_RK3588.sh file, we need to change the value of the variable TOOL\_CHAIN to the path of the gcc-buildroot-9.3.0-2020.03-x86\_64\_aarch64-rockchip-linux-gnu folder. In this way, when running the build-android\_RK3588.sh script, the cross-compilation tool in the gcc-buildroot-9.3.0-2020.03-x86\_64\_aarch64-rockchip-linux-gnu folder will be used for



## compilation

```
TARGET_SOC="rk3588"
GCC_COMPILER=aarch64-linux-gnu

export TOOL_CHAIN=~/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
export LD_LIBRARY_PATH=${TOOL_CHAIN}/lib64:$LD_LIBRARY_PATH
export CC=${GCC_COMPILER}-gcc
export CXX=${GCC_COMPILER}-g++
```

4) After modification, save and exit

### 3. 36. 4. 3. Compile rknn\_yolov5\_demo

1) Run `build-linux_RK3588.sh`. This script generates a program through cross-compilation that is suitable for the RK3588 development board and can run the RKNN model on it for inference.

```
test@test:~/rknp2/examples/rknn_yolov5_demo$ ./build-linux_RK3588.sh
```

2) After running `build-linux_RK3588.sh`, there will be an additional folder named `install` in the current directory. The `rknn_yolov5_demo_Linux` folder under this folder contains the program generated by cross-compilation and its related files.

```
test@test:~/rknp2/examples/rknn_yolov5_demo$ ls install
rknn_yolov5_demo_Linux
```

### 3. 36. 4. 4. Deploy rknn\_yolov5\_demo to the development board

On the Ubuntu PC side, you can use the following command to upload the `rknn_yolov5_demo_Linux` folder to the development board through the `adb` tool, thereby implementing the deployment of `rknn_yolov5_demo` on the development board.

```
test@test:~/rknp2/examples/rknn_yolov5_demo$ adb push \
install/rknn_yolov5_demo_Linux /data/rknn_yolov5_demo_Linux
```





### 3. 36. 4. 5. Run rknn\_yolov5\_demo on the development board

1) Enter the file system of the development board through adb shell on Ubuntu PC

```
test@test:~$ adb shell
root@orangepi5:/#
```

2) Switch to the rknn\_yolov5\_demo\_Linux directory

```
root@orangepi5:/# cd /data/rknn_yolov5_demo_Linux/
root@orangepi5:/data/rknn_yolov5_demo_Linux# ls
lib  model  rknn_yolov5_demo  rknn_yolov5_video_demo
```

3) Then run the rknn\_yolov5\_demo program to perform inference. In the following command, the program uses the yolov5s-640-640.rknn model to perform inference on the bus.jpg image. The entire running process will be completed on the development board.

```
root@orangepi5:/data/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo \
./model/RK3588/yolov5s-640-640.rknn ./model/bus.jpg
```

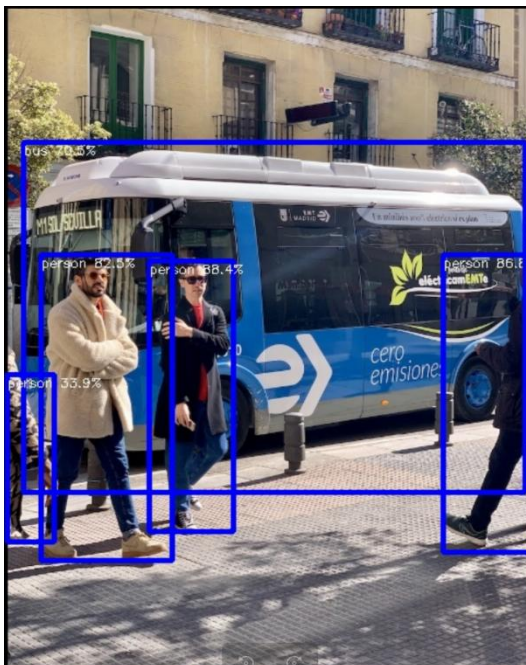
4) After the operation is completed, the inference result out.jpg image is saved in the current directory.

```
root@orangepi5:/data/rknn_yolov5_demo_Linux# ls
lib  model  out.jpg  rknn_yolov5_demo  rknn_yolov5_video_demo
```

5) On the Ubuntu PC side, we can use the following command to download the out.jpg image through the adb tool, and then use the image viewer to view it

```
test@test:~$ adb pull /data/rknn_yolov5_demo_Linux/out.jpg ~/Desktop/
/data/rknn_yolov5_demo_Linux/out.jpg: ...led. 1.9 MB/s (191507 bytes in 0.095s)
```

6) The out.jpg picture shows the object categories and confidence rates detected in the bus.jpg picture using the yolov5s-640-640.rknn model.



### 3. 37. RK3588 How to use Baidu Feipian

Using Baidu Feipian on the rk3588 development board includes converting the pdmodel model to the rknn model on the PC side and deploying the rknn model using the FastDeploy deployment tool developed by Baidu Feipian on the board side. The following content is implemented in the environment where the PC system is Ubuntu22.04 and the board system is Debian 11. Please test it yourself in other environments.

#### 3. 37. 1. Ubuntu PC environment construction

The tools and uses that need to be installed on Ubuntu PC are as follows:

Tool name	use
Anaconda3	For creating and managing Python environments
Paddle2ONNX	Used to convert pdmodel model to ONNX model
RKNN-Toolkit2	Used to convert ONNX model to RKNN model



### 3. 37. 1. 1. Install Anaconda3 on PC

1) Open the browser on the ubuntu PC, enter the following URL in the address bar to download and install the Anaconda3 script. After the download is completed, you will get the **Anaconda3-2023.07-1-Linux-x86\_64.sh** file

[https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-x86\\_64.sh](https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-x86_64.sh)

2) Then open the terminal and run the **Anaconda3-2023.07-1-Linux-x86\_64.sh** script to install Anaconda3

```
test@test:~/Downloads$ sh Anaconda3-2023.07-1-Linux-x86_64.sh
```

3) Then the installation script will output the following prompt message. At this time, click the Enter key to continue the installation.

```
ly@ly:~/Downloads$ sh Anaconda3-2023.07-1-Linux-x86_64.sh

Welcome to Anaconda3 2023.07-1

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>> 
```

4) After clicking the Enter key, some introduction information about Anaconda3 will appear. Keep clicking the "↓" key



```
=====
End User License Agreement - Anaconda Distribution
=====

Copyright 2015-2023, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between y
ou and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution
(which was formerly known as Anaconda Individual Edition).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusi
ve, non-transferable license to:

    * Install and use the Anaconda Distribution (which was formerly known as Anaco
nda Individual Edition),
    * Modify and create derivative works of sample source code delivered in Anacon
da Distribution from Anaconda's repository, and;
    * Redistribute code files in source (if provided to you by Anaconda as source)
and binary forms, with or without modification subject to the requirements set
forth below, and;

--更多--
```

5) Then the installation script will prompt whether to accept the license terms. At this time, enter yes and press Enter.

```
The following packages listed on https://www.anaconda.com/cryptography are inclu
ded in the repository accessible through Anaconda Distribution that relate to cr
yptography.

Last updated February 25, 2022

Do you accept the license terms? [yes|no]
[no] >>> 
```

6) Then the installation script will remind you to install Anaconda3 to the home directory. Press the Enter key to confirm.



```
Anaconda3 will now be installed into this location:  
/home/ly/anaconda3  
  
- Press ENTER to confirm the location  
- Press CTRL-C to abort the installation  
- Or specify a different location below  
  
[/home/ly/anaconda3] >>>
```

7) Then the installation script will prompt whether to initialize Anaconda3, enter yes, and then press the Enter key

```
installation finished.  
Do you wish the installer to initialize Anaconda3  
by running conda init? [yes|no]  
[no] >>> 
```

8) When you see the following printout in the terminal, it means Anaconda3 has been successfully installed.

```
If you'd prefer that conda's base environment not be activated on startup,  
set the auto_activate_base parameter to false:  
  
conda config --set auto_activate_base false  
  
Thank you for installing Anaconda3!
```

### 3. 37. 1. 2. Install RKNN-Toolkit2 on PC

1) Open the terminal on the ubuntu PC and create an environment with python version 3.8 through the Anaconda3 tool

```
(base)test@test:~$ conda create -n fastdeploy python=3.8
```

2) Activate the python3.8 environment just created

```
(base)test@test:~$ conda activate fastdeploy
```

3) Then install pip3 development tools and package management tools

```
(fastdeploy)test@test:~$ sudo apt-get install python3-dev python3-pip
```



4) Then install the dependency packages of RKNN-Toolkit2

```
(fastdeploy)test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 libs  
m6 libgl1-mesa-glx libprotobuf-dev gcc
```

5) rknn\_toolkit2 has a specific dependency on numpy, so numpy==1.16.6 needs to be installed first

```
(fastdeploy)test@test:~$ pip install numpy==1.16.6
```

6) Install git tools

```
(fastdeploy)test@test:~$ sudo apt install git
```

7) Then execute the following command to download RKNN-Toolkit2. After the download is completed, you will get the rknn-toolkit2 folder

```
(fastdeploy)test@test:~$ git clone https://github.com/rockchip-linux/rknn-toolkit2
```

8) Then execute the following command to install the RKNN-Toolkit2 corresponding to the python3.8 version

```
(fastdeploy)test@test:~$ pip install rknn-toolkit2/rknn-toolkit2/packages/rknn_tool  
kit2-1.6.0+81f21f4d-cp38-cp38-linux_x86_64.whl
```

### 3. 37. 1. 3. Install Paddle2ONNX on PC

You can execute the following command to install paddle2onnx

```
(fastdeploy)test@test:~$ pip install paddle2onnx
```

### 3. 37. 2. Board environment construction

The tools that need to be installed on the board end and their uses are as follows:

Tool name	use
Anaconda3	For creating and managing Python environments
rknpu2	Basic driver of rknpu2
FastDeploy	After compilation, you get the FastDeploy reasoning library





### 3. 37. 2. 1. Install Anaconda3 on the board side

1) Open the browser on the board end, enter the following URL in the address bar to download and install the Anaconda3 script. After the download is completed, you will get the **Anaconda3-2023.07-1-Linux-aarch64.sh** file

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-aarch64.sh>

2) Open the terminal and run the **Anaconda3-2023.07-1-Linux-aarch64.sh** script to install Anaconda3

```
orangeypi@orangeypi:~/Downloads$ sh Anaconda3-2023.07-1-Linux-aarch64.sh
```

3) The installation script will then output the following prompt message. Click the Enter key to continue the installation.

```
orangeypi@orangeypi:~/Downloads$ sh Anaconda3-2023.07-1-Linux-aarch64.sh
Welcome to Anaconda3 2023.07.1
In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

4) After clicking the Enter key, some introduction information about Anaconda3 will appear. Keep clicking the "↓" key

```
=====
End User License Agreement - Anaconda Distribution
=====

Copyright 2015-2023, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution (which was formerly known as Anaconda Individual Edition).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

* Install and use the Anaconda Distribution (which was formerly known as Anaconda Individual Edition),
* Modify and create derivative works of sample source code delivered in Anaconda Distribution from Anaconda's repository, and;
* Redistribute code files in source (if provided to you by Anaconda as source) and binary forms, with or without modification subject to the requirements set forth below, and;

Anaconda may, at its option, make available patches, workarounds or other updates to Anaconda Distribution. Unless the updates are provided with their separate governing terms, they are deemed part of Anaconda Distribution licensed to you as provided in this Agreement. This Agreement does not entitle you to any support for Anaconda Distribution.

Anaconda reserves all rights not expressly granted to you in this Agreement.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Anaconda nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
* The purpose of the redistribution is not part of a commercial product for resale. Please contact the Anaconda team for a third party redistribution commercial license
* Commercial usage of the repository is non-compliant with our Terms of Service. Please contact us to learn more about our commercial offerings.

You acknowledge that, as between you and Anaconda, Anaconda owns all right, title, and interest, including all intellectual property rights, in and to Anaconda Distribution and, with respect to third-party products distributed with or through Anaconda Distribution, the applicable third-party licensors own all right, title and interest, including all intellectual property rights, in and to such products. If you send or transmit any communications or materials to Anaconda suggesting or recommending changes to the software or documentation, including without limitation, new features or functionality relating thereto, or any comments, questions, suggestions or the like ("Feedback"), Anaconda is free to use such Feedback. You hereby assign to Anaconda all right, title, and interest in
--More--
```

5) Then the installation script will prompt whether to accept the license terms. At this time, enter yes and press Enter.





```
The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to cryptography.
Last updated February 25, 2022

Do you accept the license terms? [yes|no]
[no] >>> |
```

6) Then the installation script will remind you to install Anaconda3 to the home directory. Press the Enter key to confirm.

```
Anaconda3 will now be installed into this location:
/home/orangepi/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/orangepi/anaconda3] >>> |
```

7) Then the installation script will prompt whether to initialize Anaconda3, enter yes, and then press the Enter key

```
Installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> |
```

8) When you see the following printout in the terminal, it means Anaconda3 has been successfully installed.

```
If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Anaconda3!
```

9) If you use the conda command in the terminal and the command does not exist, you need to modify the ~/.bashrc file.

```
orangepi@orangepi:~$ vi ~/.bashrc
```

10) Add the following code at the end of the ~/.bashrc file

```
export PATH=/home/orangepi/anaconda3/bin:$PATH
```

11) Then enter the following command in the terminal to make the changes take effect.

```
orangepi@orangepi:~$ source ~/.bashrc
```

12) Then enter the following command in the terminal to initialize conda

```
(base)orangepi@orangepi:~$ conda init bash
```

13) Then close the current terminal and reopen a terminal. At this time, you can use the conda command normally.



### 3. 37. 2. 2. Install the rknpu2 driver on the board side

1) Open the terminal on the board and create an environment with python version 3.9 through the Anaconda3 tool

```
(base)orangepi@orangepi:~$ conda create -n fastdeploy python=3.9
```

2) Activate the python3.9 environment just created

```
(base)orangepi@orangepi:~$ conda activate fastdeploy
```

3) Download the rknpu2\_device\_install\_1.4.0.zip file through wget

```
(fastdeploy)orangepi@orangepi:~$ wget https://bj.bcebos.com/fastdeploy/third_libs/rknpu2_device_install_1.4.0.zip
```

4) Then execute the following command to decompress rknpu2\_device\_install\_1.4.0.zip. After decompression, you will get the rknpu2\_device\_install\_1.4.0 folder and \_\_MACOSX folder

```
(fastdeploy)orangepi@orangepi:~$ unzip rknpu2_device_install_1.4.0.zip
```

5) Switch to the rknpu2\_device\_install\_1.4.0 directory

```
(fastdeploy)orangepi@orangepi:~$ cd rknpu2_device_install_1.4.0/
```

6) There is the rknn\_install\_rk3588.sh script in this directory. Run this script to complete the installation of the board-side rknpu2 driver.

```
(fastdeploy)orangepi@orangepi:~/rknpu2_device_install_1.4.0$ sudo bash rknn_install_rk3588.sh
```

### 3. 37. 2. 3. Compile FastDeploy C++ SDK on the board end

1) You need to use the cmake command when compiling. You can execute the following command to install the cmake tool.

```
(fastdeploy)orangepi@orangepi:~$ sudo apt-get install -y cmake
```

2) Then download the FastDeploy SDK. After the command is executed, you will get the FastDeploy folder.

```
(fastdeploy)orangepi@orangepi:~$ git clone https://github.com/PaddlePaddle/FastD
```



## eploy.git

3) Switch to the FastDeploy directory

```
(fastdeploy)orange@orange:~$ cd FastDeploy
```

4) Create the compilation directory build and switch to the build directory

```
(fastdeploy)orange@orange:~/FastDeploy$ mkdir build && cd build
```

5) Before compiling, you need to use cmake to configure the project information that needs to be compiled. After executing the following command, there will be some more files in the current directory, including the Makefile file used for compilation.

```
(fastdeploy)orange@orange:~/FastDeploy/build$ cmake .. -DENABLE_OR_T_BACKEND=ON \  
-DENABLE_RKNPU2_BACKEND=ON \  
-DENABLE_VISION=ON \  
-DRKNN2_TARGET_SOC=RK3588 \  
-DCMAKE_INSTALL_PREFIX=${PWD}/fastdeploy-0.0.3
```

6) Execute the following command to start compilation

```
(fastdeploy)orange@orange:~/FastDeploy/build$ make -j8
```

7) After the compilation is completed, use the following command to install the compiled files to the specified path

```
(fastdeploy)orange@orange:~/FastDeploy/build$ make install
```

8) After the compilation is completed, the fastdeploy-0.0.3 folder is mainly obtained. In this folder, there is the script file fastdeploy\_init.sh for configuring environment variables. After using this script to configure the environment variables, you can use some of the compiled library files.

```
(fastdeploy)orange@orange:~/FastDeploy/build$ source fastdeploy-0.0.3/fastdeploy_init.sh
```

### 3. 37. 3. Example of deploying model using FastDeploy

The ResNet50\_vd model is a model used for target classification. The following uses the ResNet50\_vd model as an example to illustrate the process of deploying the pdmodel model using FastDeploy.



### 3. 37. 3. 1. Ubuntu PC model conversion

1) Open the terminal on the PC and activate the python3.8 environment previously created using Anaconda3

```
test@test:~$ conda activate fastdeploy
```

2) In the model conversion script, you need to import the yaml module and the six module. You can execute the following command to install them.

```
(fastdeploy)test@test:~$ pip install pyyaml six
```

3) Execute the following command to download the ResNet50\_vd\_infer.tgz file

```
(fastdeploy)test@test:~$ wget https://bj.bcebos.com/paddlehub/fastdeploy/ResNet50_vd_infer.tgz
```

4) After decompressing the ResNet50\_vd\_infer.tgz file, you can get the ResNet50\_vd\_infer folder, which contains the pdmodel model file inference.pdmodel and other related files.

```
(fastdeploy)test@test:~$ tar -xvf ResNet50_vd_infer.tgz
```

5) You can use the following command to convert the pdmodel model to an onnx model through paddle2onnx. After executing this command, there will be an extra converted onnx model file ResNet50\_vd\_infer.onnx in the ResNet50\_vd\_infer folder.

```
(fastdeploy)test@test:~$ paddle2onnx --model_dir ResNet50_vd_infer \
--model_filename inference.pdmodel \
--params_filename inference.pdiparams \
--save_file ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--enable_dev_version True \
--opset_version 10 \
--enable_onnx_checker True
```

6) Then use the following command to fix the shape to [1,3,224,224]. After executing the command, the ResNet50\_vd\_infer.onnx file will be modified.

```
(fastdeploy)test@test:~$ python -m paddle2onnx.optimize --input_model \
ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--output_model ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--input_shape_dict '{"inputs':[1,3,224,224]}'
```



7) To convert the onnx model to the rknn model, you need to use the script in the FastDeploy SDK. Execute the following command to download FastDeploy

```
(fastdeploy)test@test:~$ git clone https://github.com/PaddlePaddle/FastDeploy.git
```

8) Then transfer the ResNet50\_vd\_infer folder to the corresponding directory of FastDeploy

```
(fastdeploy)test@test:~$ mv ResNet50_vd_infer \  
FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

9) Switch to the directory for model conversion

```
(fastdeploy)test@test:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

10) Execute the following command to convert the onnx model to the rknn model. Finally, the rknn model file ResNet50\_vd\_infer\_rk3588\_unquantized.rknn is obtained in the ResNet50\_vd\_infer directory.

```
(fastdeploy)test@test:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2$ python ./rknpu2_tools/export.py \  
--config_path ./rknpu2_tools/config/ResNet50_vd_infer_rknn.yaml \  
--target_platform rk3588
```

11) When deploying on the board end, the name of the rknn model file used is ResNet50\_vd\_infer\_rk3588.rknn, so the ResNet50\_vd\_infer\_rk3588\_unquantized.rknn file needs to be renamed to ResNet50\_vd\_infer\_rk3588.rknn

```
(fastdeploy)test@test:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2$ mv ResNet50_vd_infer/ResNet50_vd_infer_rk3588_unquantized.rknn \  
ResNet50_vd_infer/ResNet50_vd_infer_rk3588.rknn
```

### 3. 37. 3. 2. Board end model deployment

1) Open the terminal on the board and activate the python3.9 environment created previously using Anaconda3

```
orangeypi@orangeypi:~$ conda activate fastdeploy
```

2) Run the fastdeploy\_init.sh script to configure the environment

```
(fastdeploy)orangeypi@orangeypi:~$ source FastDeploy/build/fastdeploy-0.0.3/fastdeploy_init.sh
```



3) Switch to the sample directory where the ResNet50 model is deployed in FastDeploy

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp
```

4) Create a directory structure in this directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ mkdir build images ppclas_model_dir thirdpartys
```

5) Copy the compiled fastdeploy-0.0.3 folder to the thirdpartys folder

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cp -r ~/FastDeploy/build/fastdeploy-0.0.3/ thirdpartys/
```

6) Copy the files in the ResNet50\_vd\_infer folder on the PC to the ppclas\_model\_dir directory

7) Switch to the images directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cd images
```

8) Download the test image in the images directory through wget

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ wget https://gitee.com/paddlepaddle/PaddleClas/raw/release/2.4/deploy/images/ImageNet/ILSVRC2012_val_00000010.jpeg
```

9) Then switch to the compilation directory build

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ cd ../build/
```

10) Use cmake to configure the content that needs to be compiled. After executing this command, some files will appear in the current directory, including Makefile files.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cmake ..
```

11) Execute the following command to start compilation

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make -j8
```

12) Execute the following command to install the compiled files to the specified path. After executing the command, there will be an install directory in the current directory.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make install
```

13) Switch to the install directory, where inference using the model is completed.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cd install
```



14) Use the following command to classify the content in the ILSVRC2012\_val\_00000010.jpeg image using the converted rknn model

```
(fastdeploy)orange@orange:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknp2/cpp/build/install$ ./rknp_test \
./ppclas_model_dir/ ./images/ILSVRC2012_val_00000010.jpeg
```

15) After executing this command, the following printout will appear in the echo information, indicating that the category ID number of the object in the picture is 644, and the confidence rate is 0.072998

```
ClassifyResult(
label_ids: 644,
scores: 0.072998,
)
```

### 3. 38. 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. Unplug the power supply.。

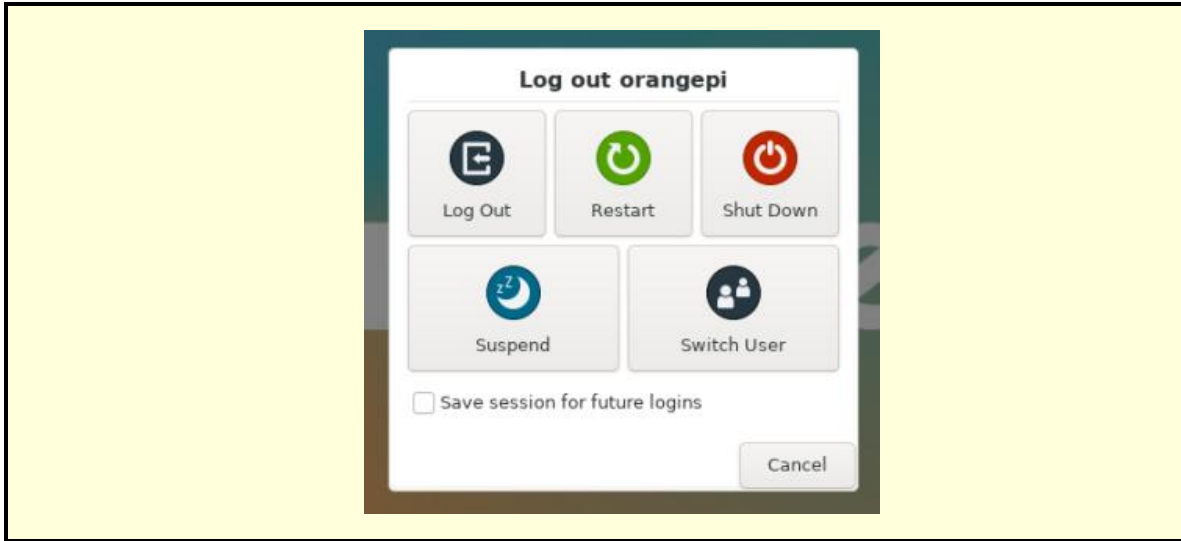
```
orange@orange:~$ sudo poweroff
```

2) In addition, the development board is equipped with a switch button, and you can also **Short Press** the switch button on the development board to turn off。



Note that the Linux desktop system will pop up the confirmation box shown in the figure below after pressing the buttons. You need to click the **Shut Down** option before turning off。





3) After shutting down, press the switch button on the development board to turn on.。



4) Restart the command of the linux system to be

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

## 4. Ubuntu22.04 Gnome Wayland Desktop System Instructions

ubuntu 22.04 gnome image is pre-installed with the panfork mesa user space library by default, and the pre-installed KODI player and Chromium browser support hardware decoding to play videos.

**NOTED:** This image needs to be used under wayland. If you need to use x11, please choose the image of xfce type.

### 4. 1. Ubuntu22.04 Gnome Desktop System Adaptation

Function	Ubuntu22.04 Gnome Wayland
USB2.0x2	OK
USB3.0x1	OK
USB Type-C 3.0	OK
DP Display	OK
eMMC	OK
AP6275P-WIFI	OK
AP6275P-Bluetooth	OK
GPIO (26pin)	OK
UART (26pin)	OK
SPI (26pin)	OK
I2C (26pin)	OK
CAN (26pin)	OK
PWM (26pin)	OK
3pin Debug Serial Port	OK
TF Card Start up	OK
HDMI Video	OK
HDMI Audio	OK
OV13850 Camera	OK
OV13855 Camera	OK
LCD1	OK

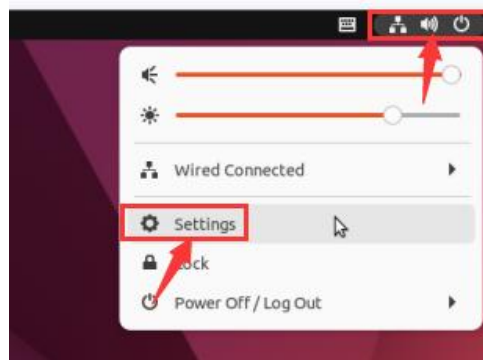


<b>LCD2</b>	<b>OK</b>
<b>Gigabit Ethernet Port</b>	<b>OK</b>
<b>Network Port Status Light</b>	<b>OK</b>
<b>MIC</b>	<b>OK</b>
<b>Headphone Playback</b>	<b>OK</b>
<b>Headphone Recording</b>	<b>OK</b>
<b>LED Light</b>	<b>OK</b>
<b>GPU</b>	<b>OK</b>
<b>NPU</b>	<b>OK</b>
<b>VPU</b>	<b>OK</b>
<b>switch button</b>	<b>OK</b>
<b>watchdog test</b>	<b>OK</b>
<b>Chromium hard solution video</b>	<b>OK</b>
<b>Kodi hard solution video</b>	<b>OK</b>
<b>MPV hard solution video</b>	<b>OK</b>

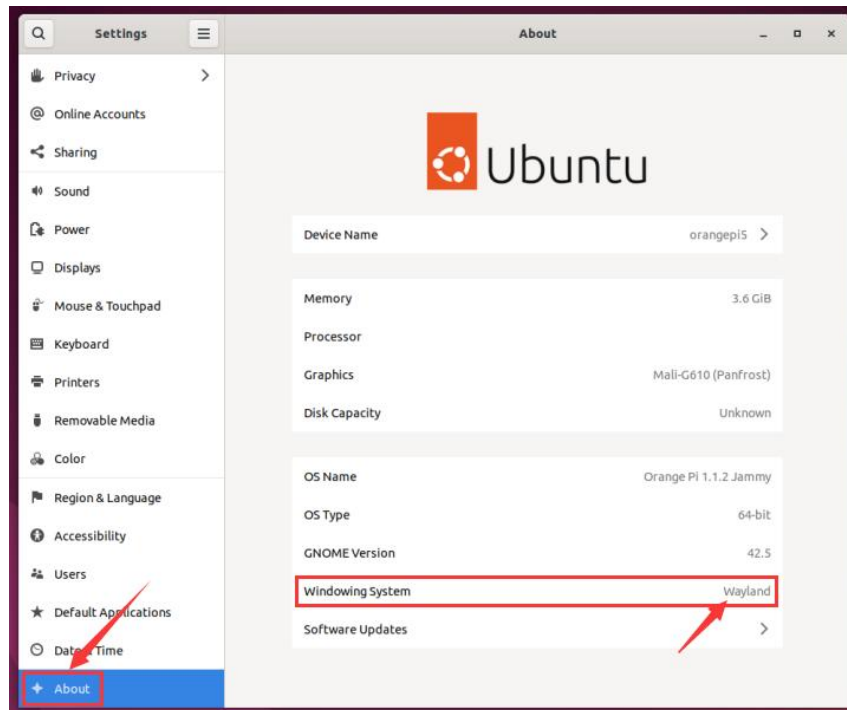
## 4. 2. How to confirm that the window system currently is wayland

1) The default window system used by the system is wayland, the confirmation method is as follows:

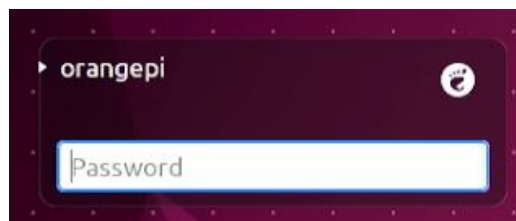
a. First open the **settings**



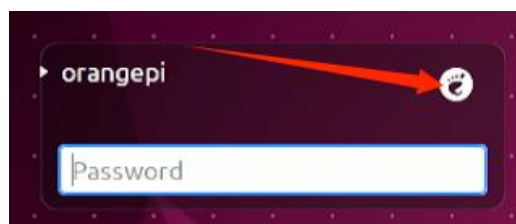
b. Then select **About**, if the **wayland** description displayed in the **Windowing System**, it means set correctly.



2) When **Log Out** exits the system, it will enter the following login interface



3) Before logging in to the system again, please click the location shown in the figure below

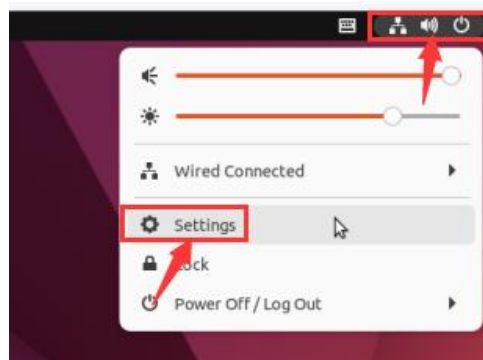


4) Then select **Ubuntu on Wayland**, and then enter the password to log in to the system

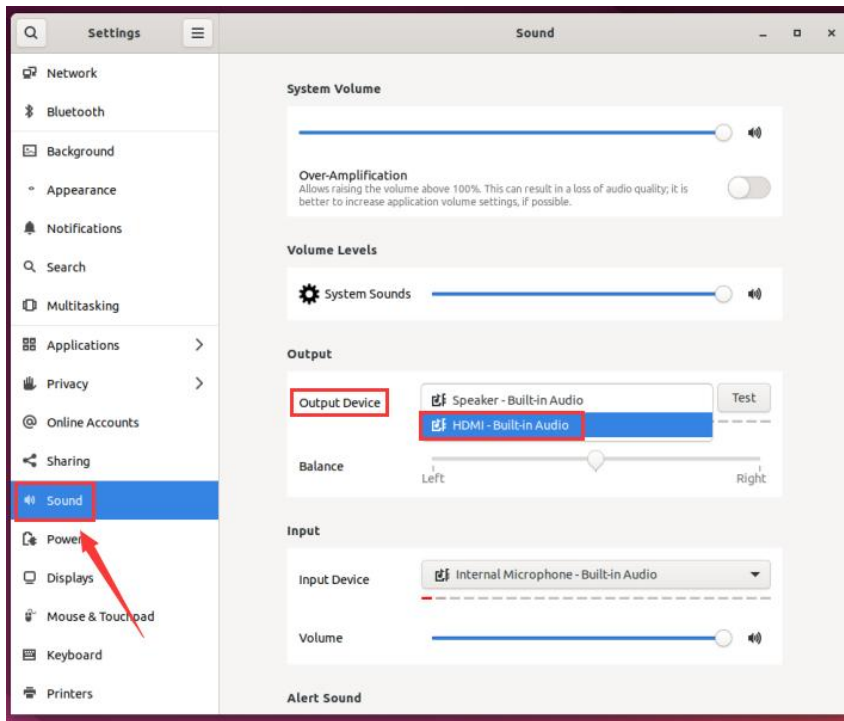


#### 4. 3. How to switch the default audio device

1) First open the settings



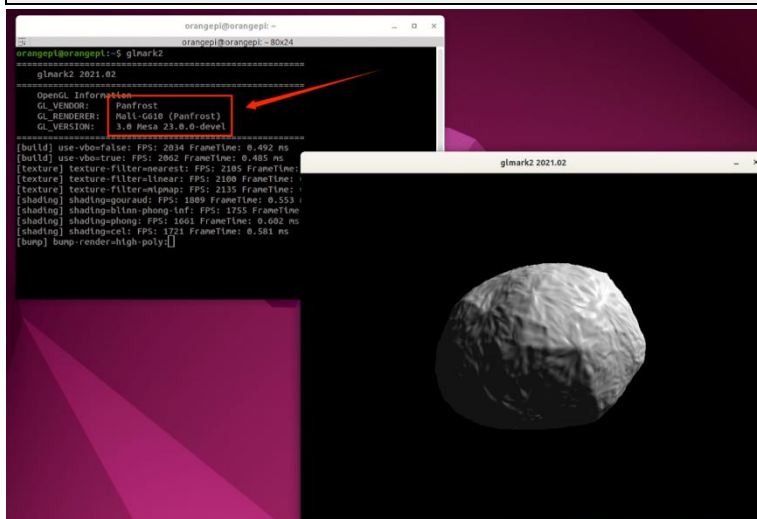
2) Then select **Sound**, and then select the audio device you want to use in **Output Device**



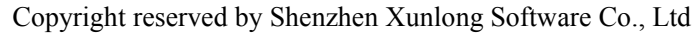
#### 4. 4. How to test GPU

1) Open a terminal on the desktop, and then enter the **glmark2** command. If you can see that **GL\_VENDOR** is displayed as **Panfrost**, it means that the GPU is used.

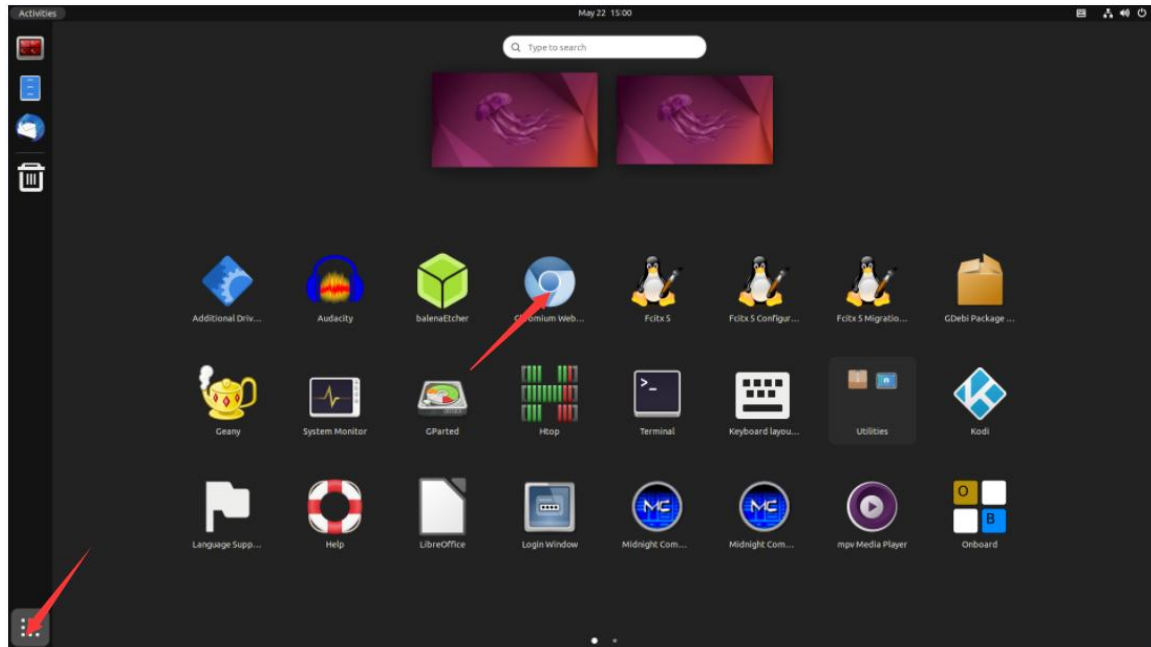
```
orange@orange:~$ glmark2
```



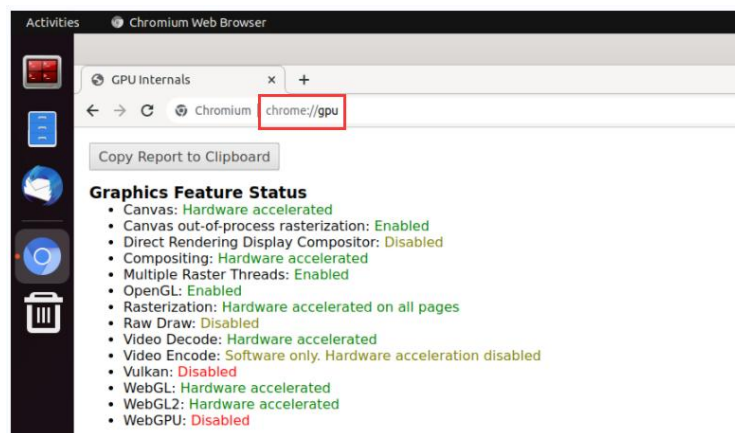
2) The glmark2 running score test is generally more than 1000 points.







2) Then enter **chrome://gpu** in the Chromium browser to check the support of GPU and video decoding



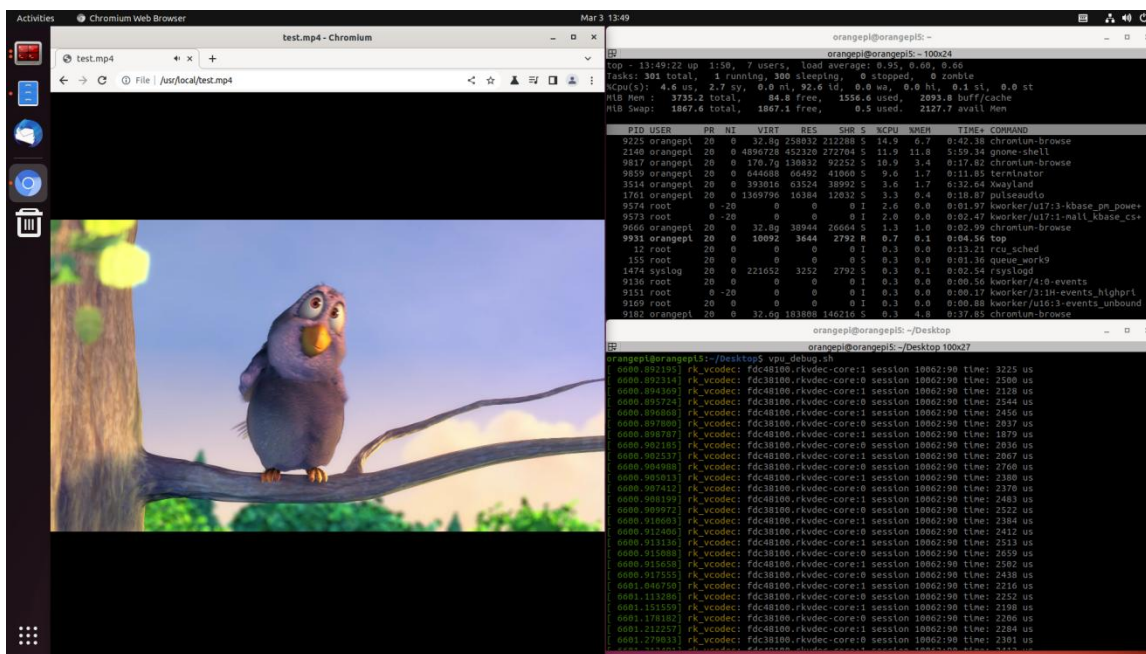
3) Then you can open the video website to play a video file, or enter the following path name in the browser to play a test video file that comes with the system

**/usr/local/test.mp4**

4) When playing a video, you can run the **vpu\_debug.sh** script in the terminal. If there is a printout in the lower right corner of the figure below, it means that hardware is used to decode the video



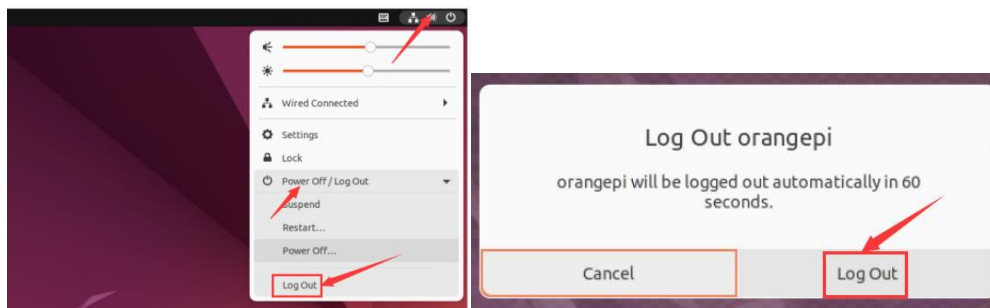
```
orange@orange:~$ vpu_debug.sh
```



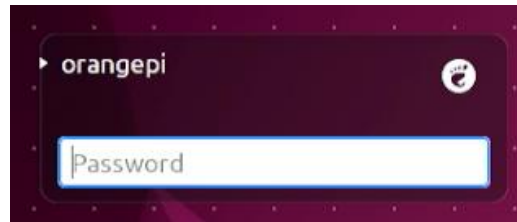
## 4.6. How to play Kodi hardware solution video

**Note:** There will be problems when opening Kodi directly on the Wayland desktop, please strictly follow the method below to open Kodi.

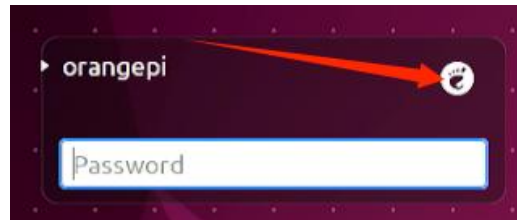
1) First log out of the system



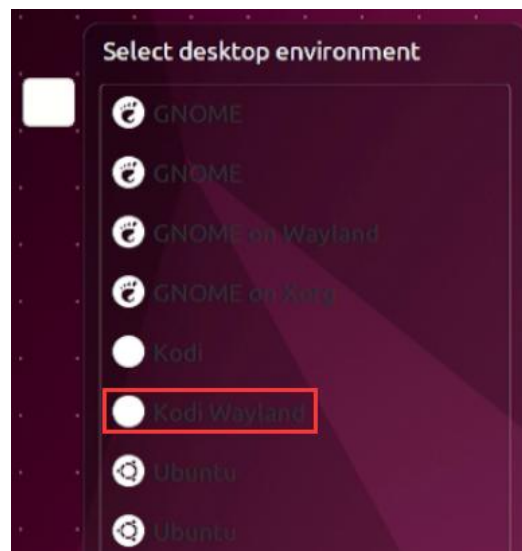
2) When you log out of the system, you will enter the following login interface



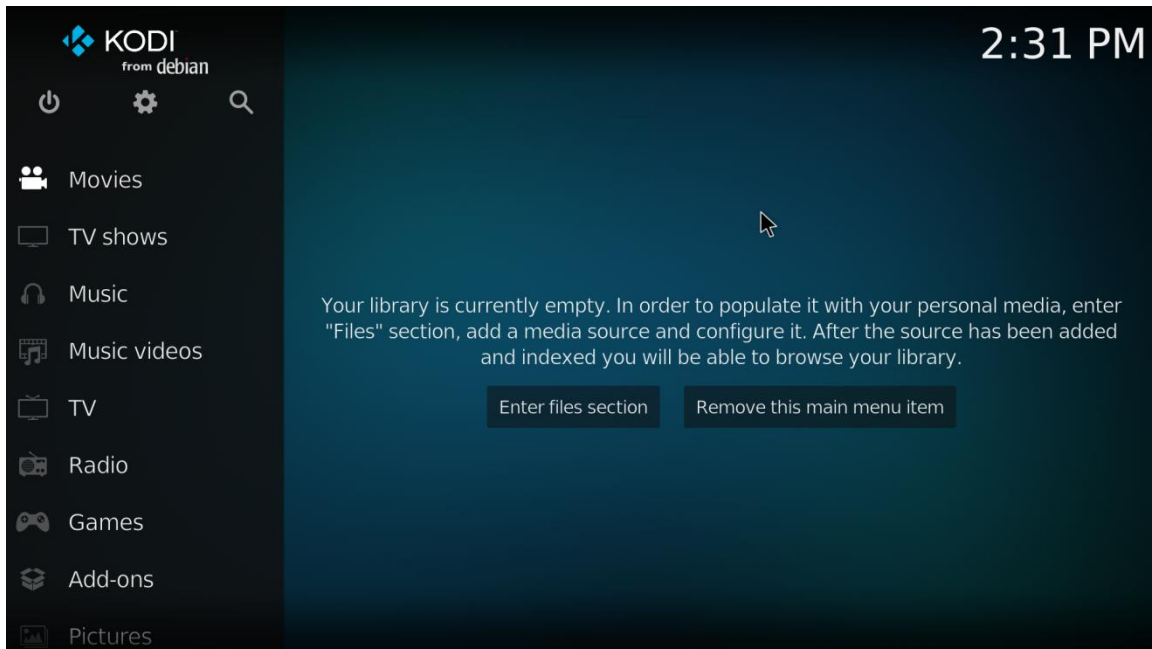
3) Then click the location shown in the figure below



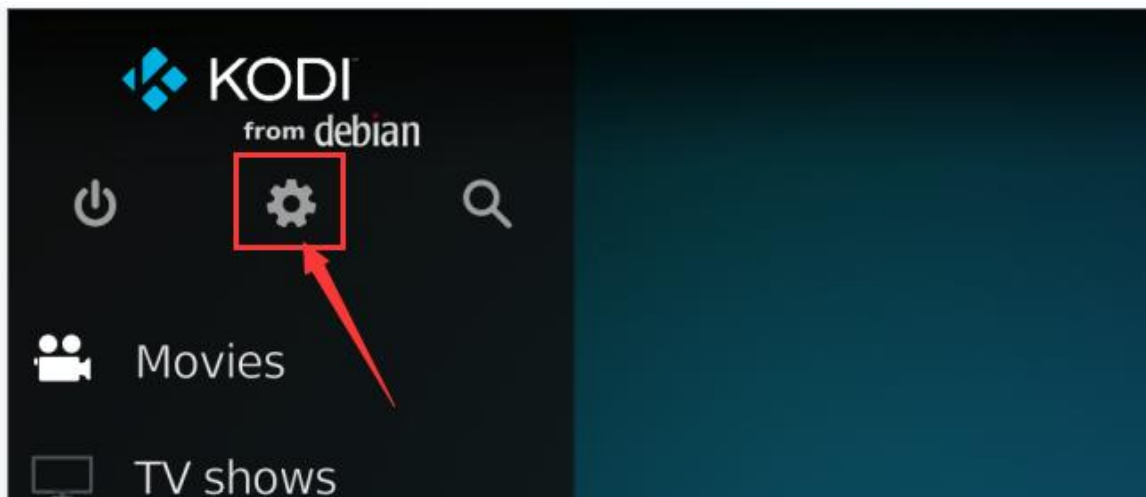
4) Then select **Kodi Wayland**, and then enter the password to log in to the system



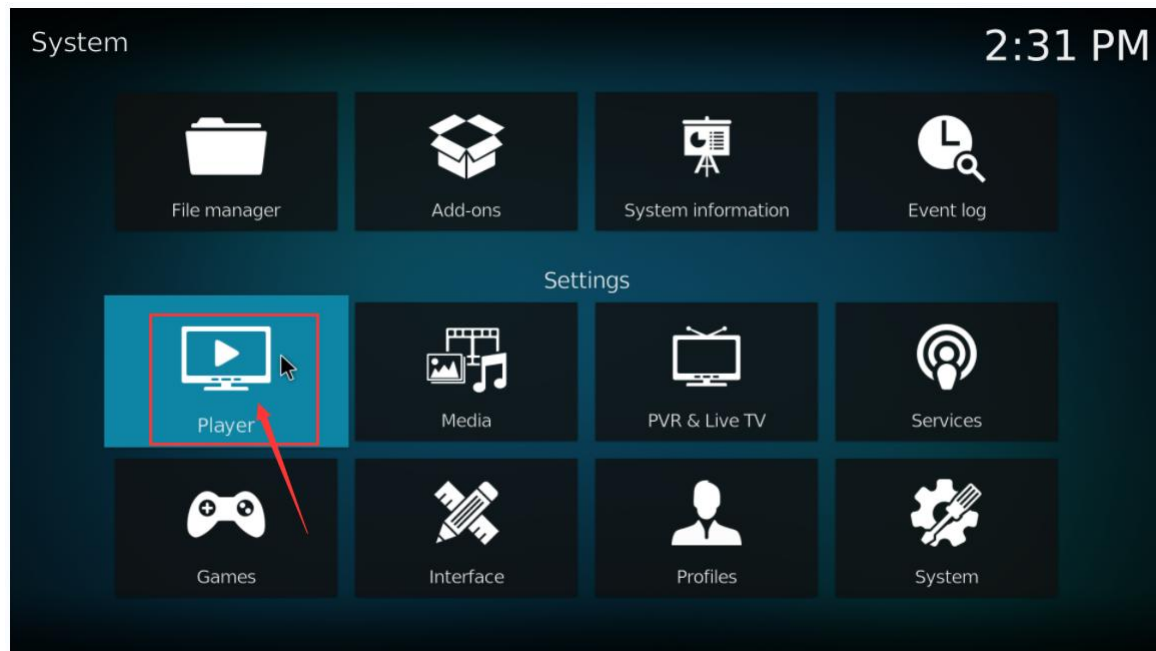
5) The interface after Kodi is opened is shown as follows



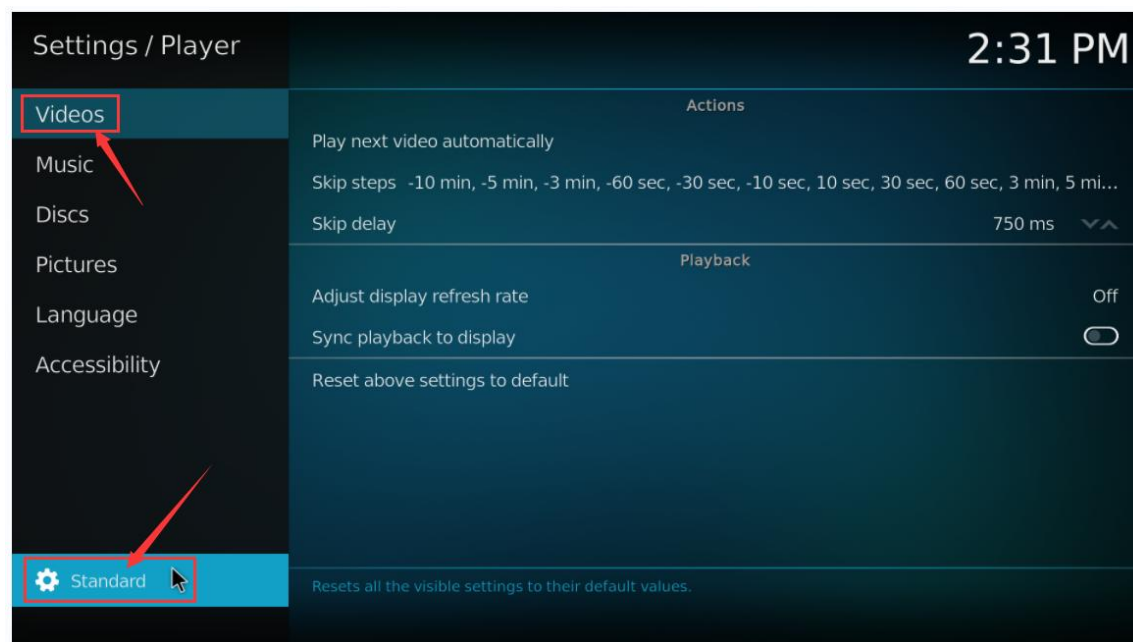
6) Then click Settings



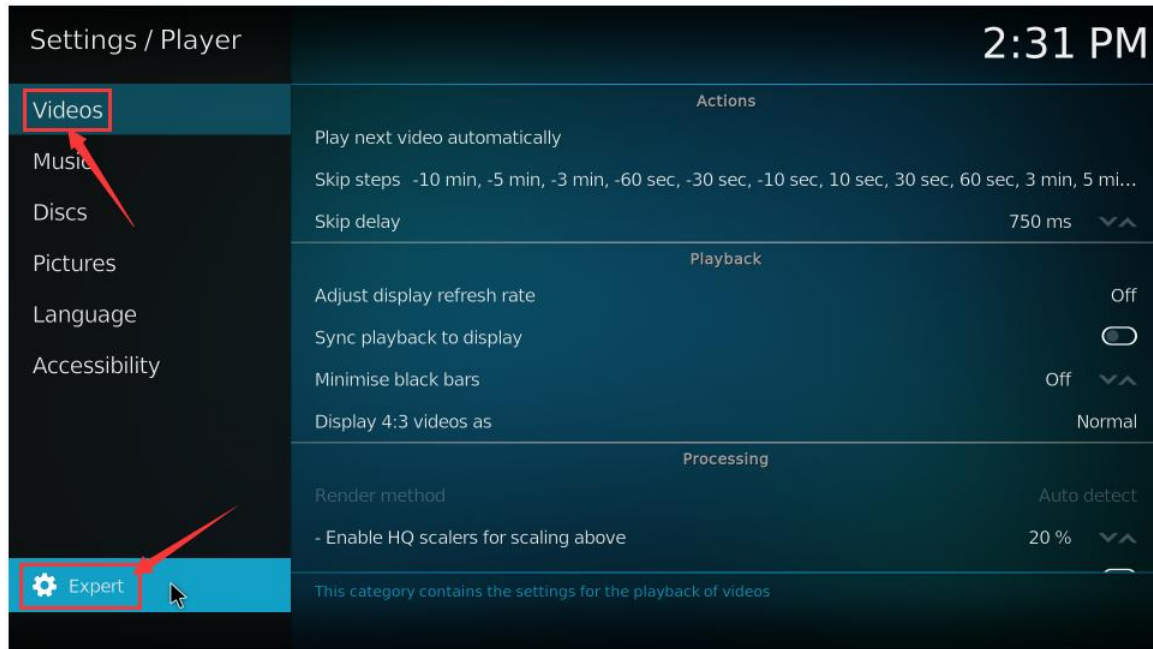
7) And then select **Player**



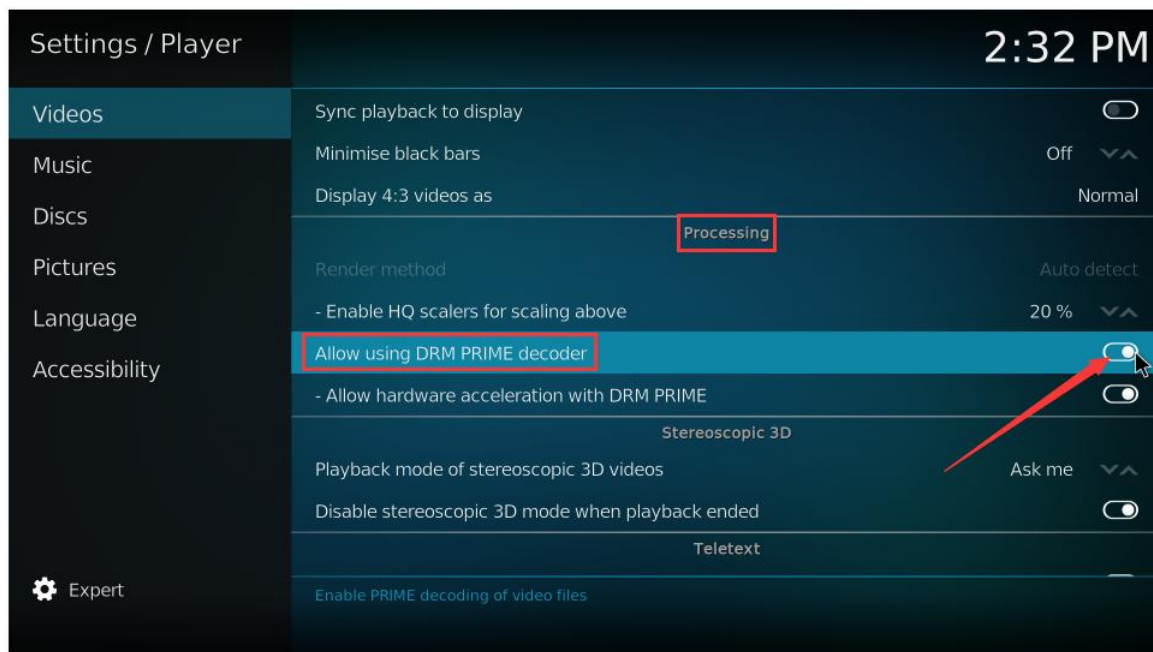
8) Select **Videos**, and click **Standard**



9) After clicking twice, it will switch to **Expert** mode, as shown in the figure below

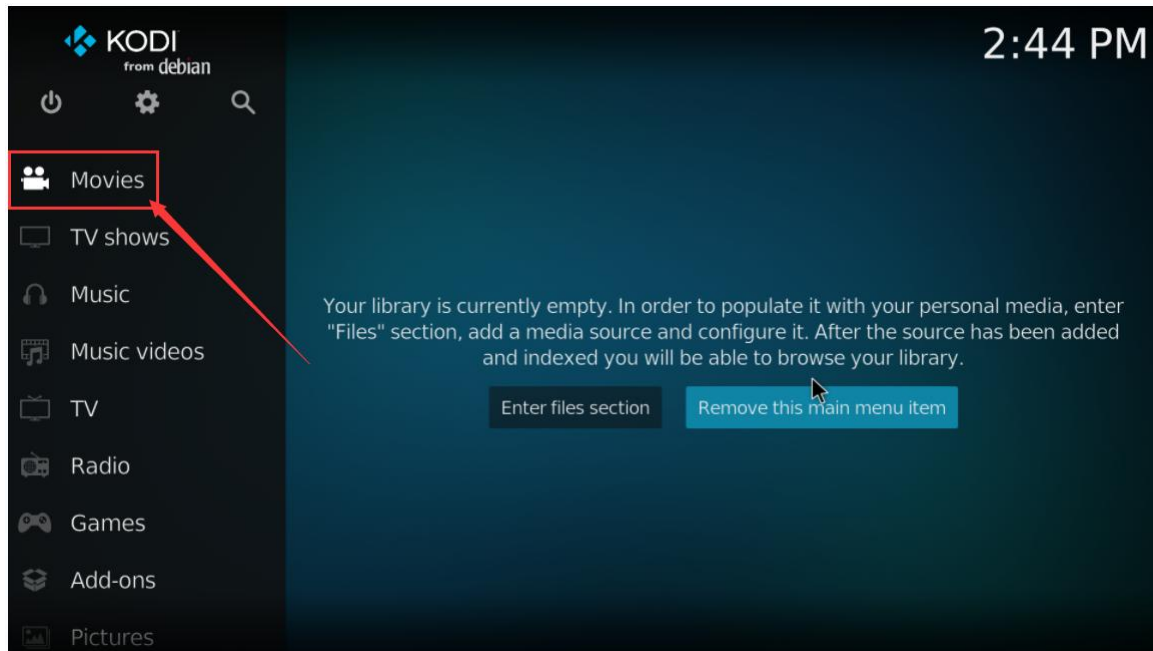


10) Then turn on **Allow using DRM PRIME decoder** in the **Processing** settings

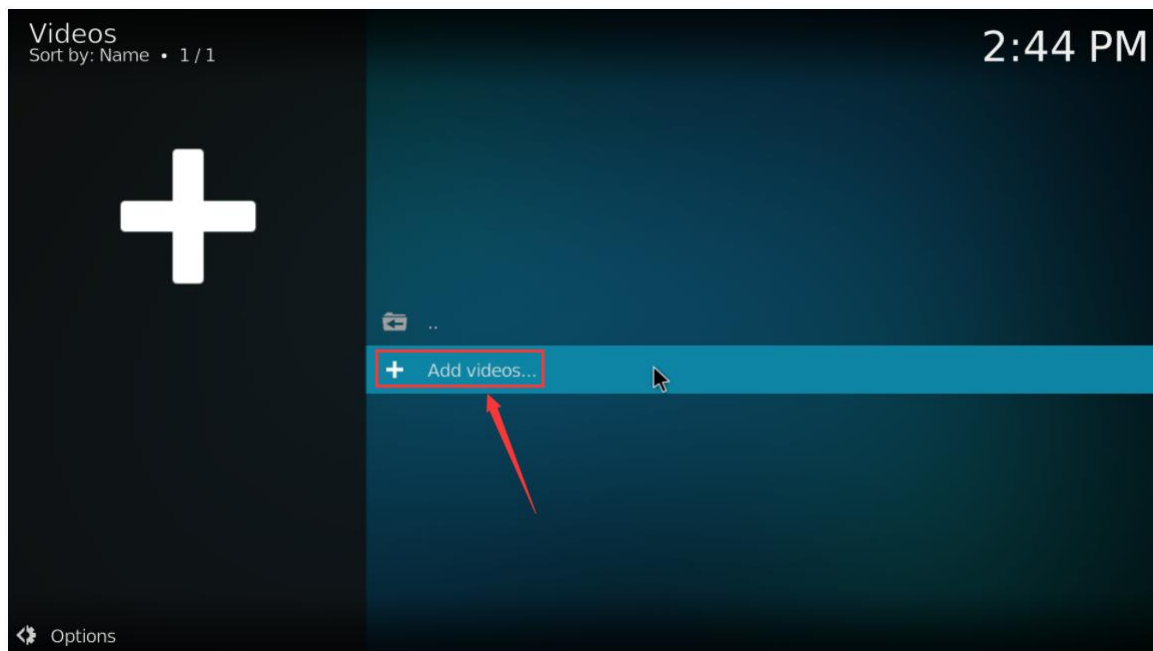


11) Then let's import a test video that comes with the system for testing. You can also upload the video you want to play to the system, and then import and play it.

a. First enter the main interface, then select **Movies**

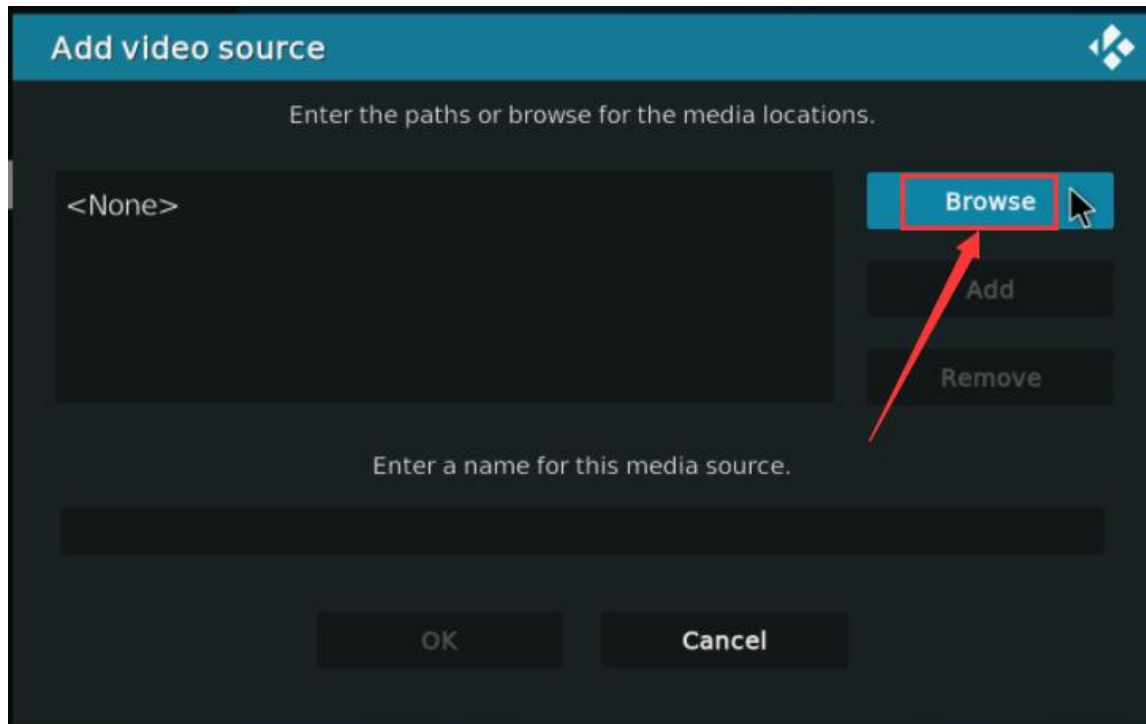


b. Then select **Add videos...**

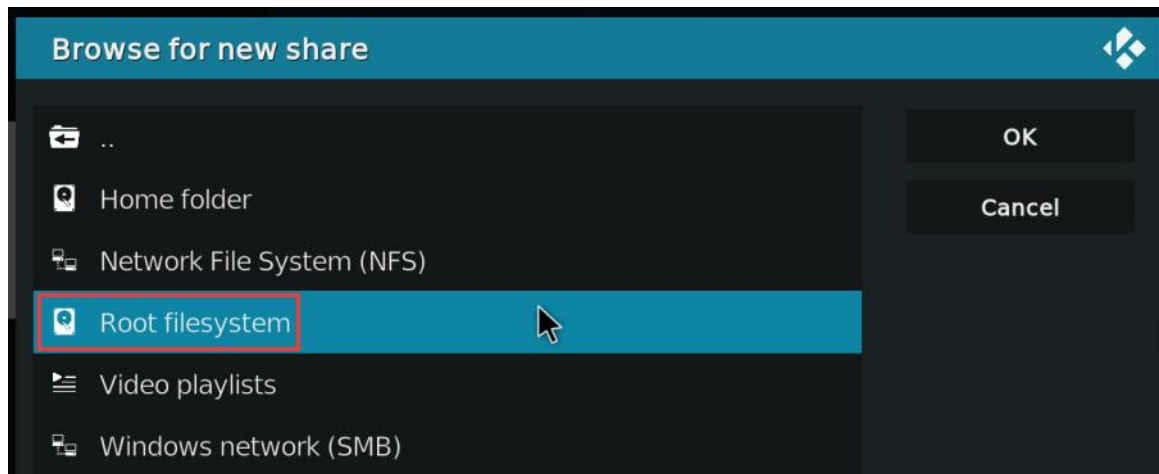


c. And then select **Browse**

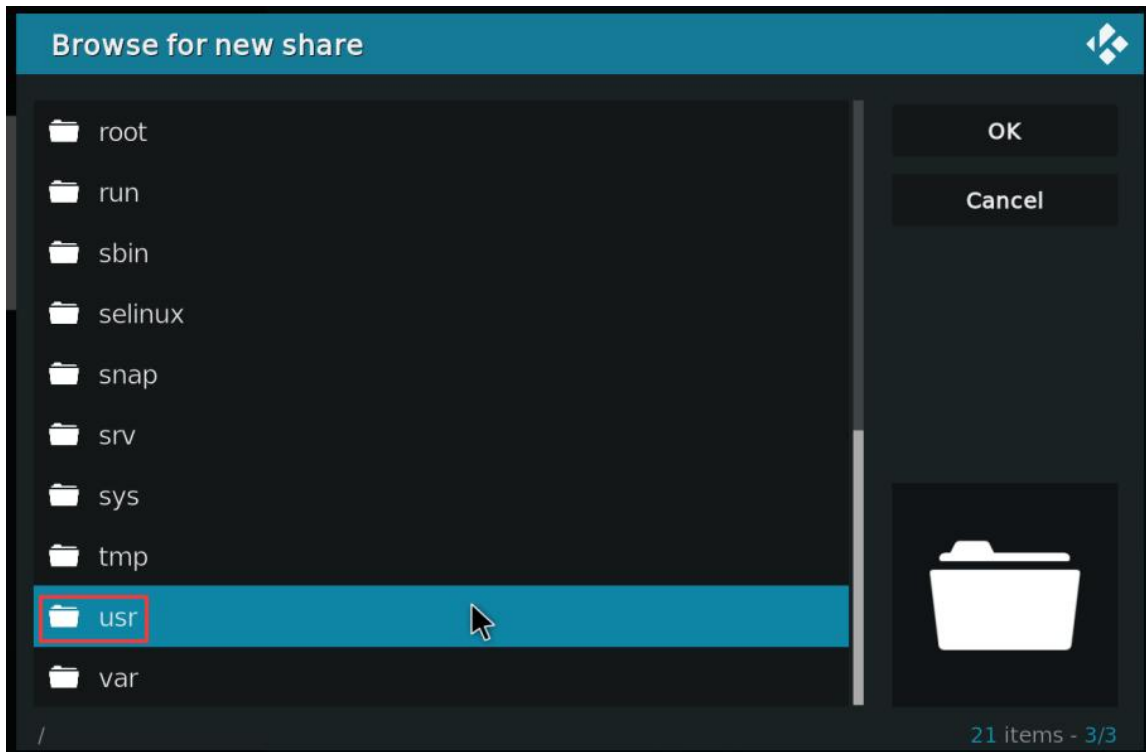




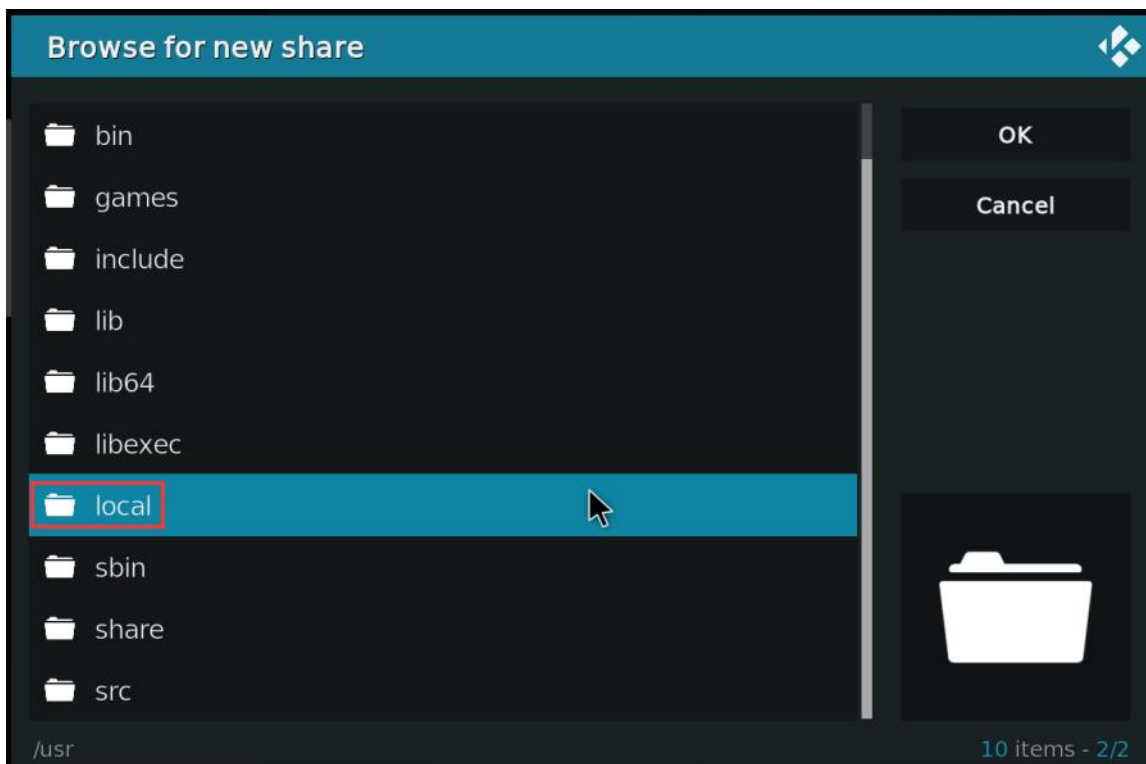
d. Select **Root filesystem**



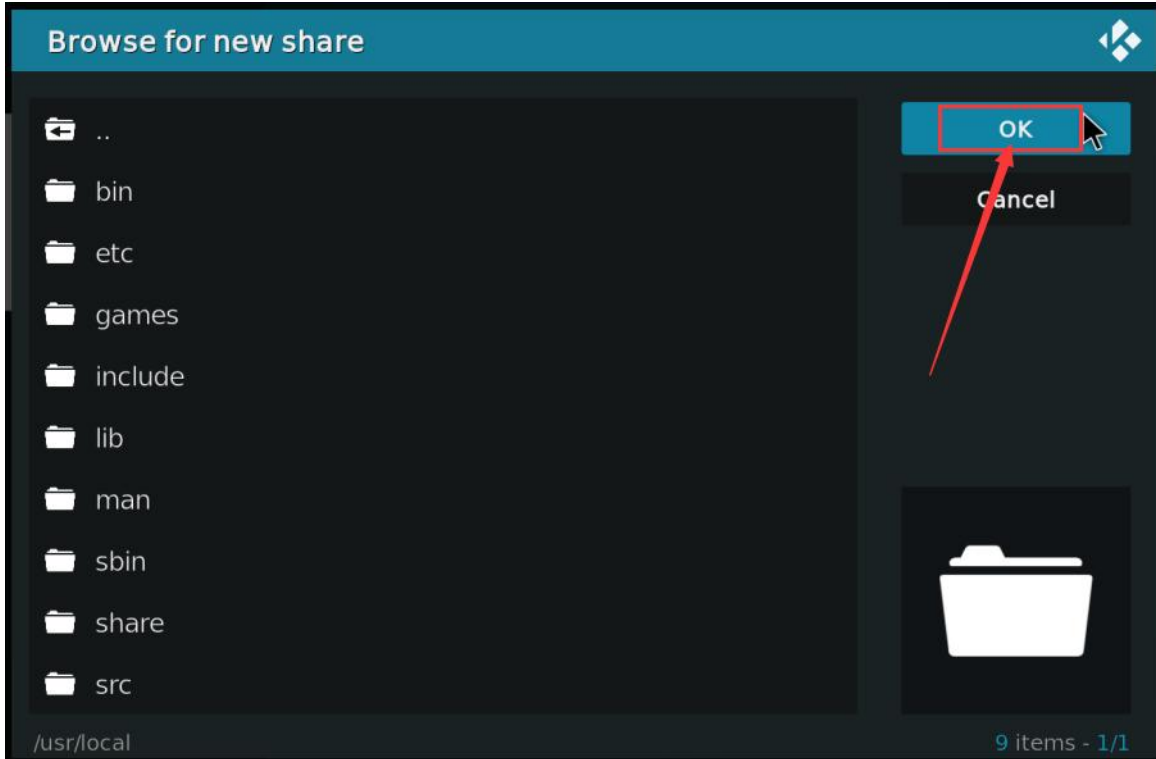
e. Select **usr**



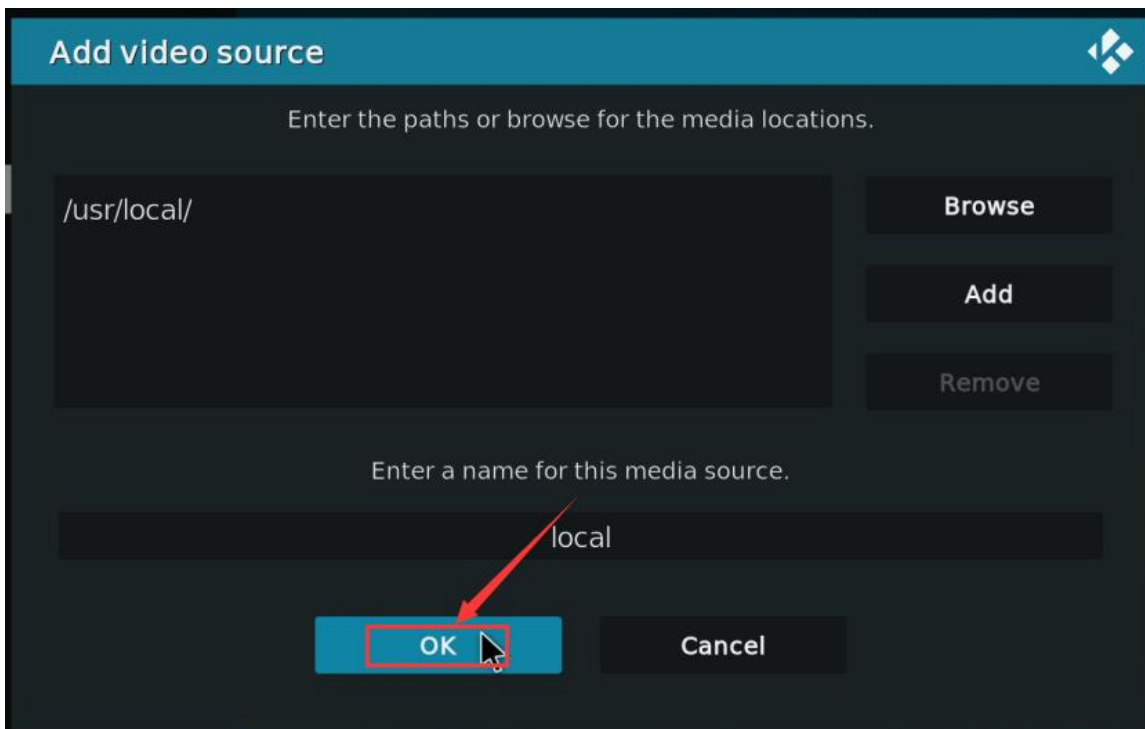
f. Select **local**



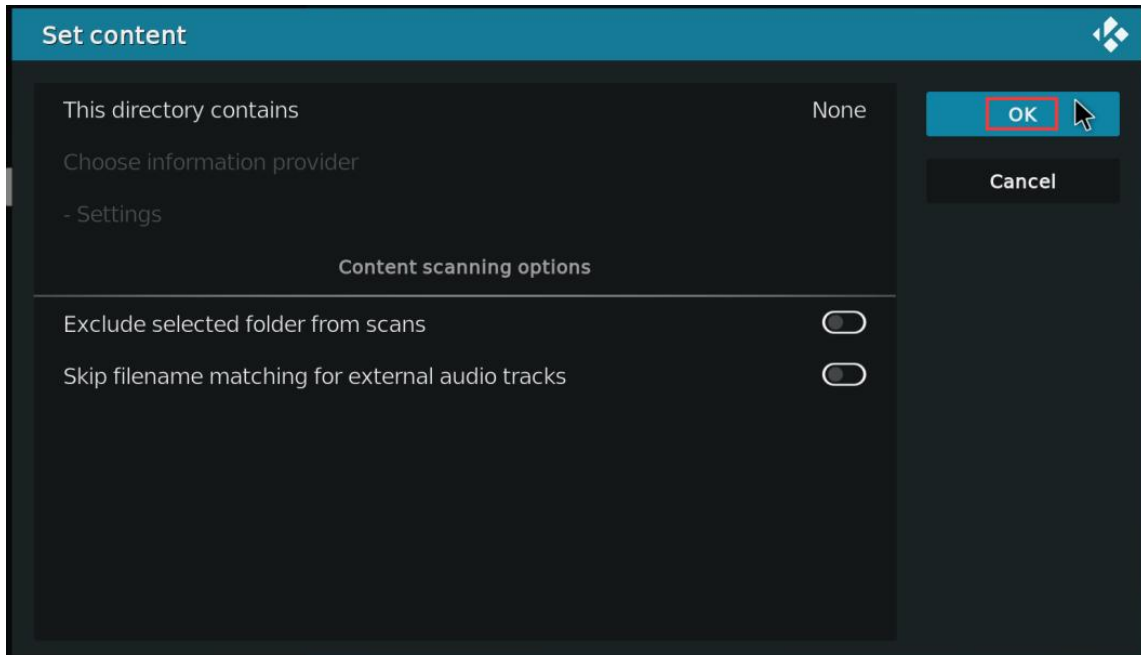
g. Select **OK**



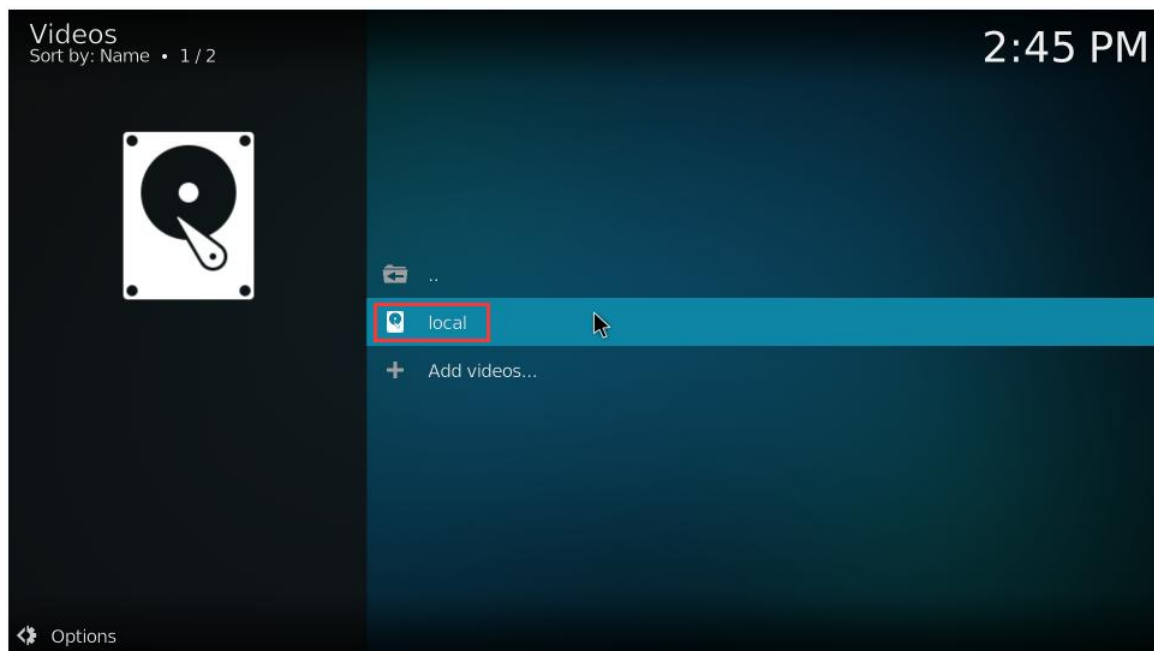
h. Select **OK**



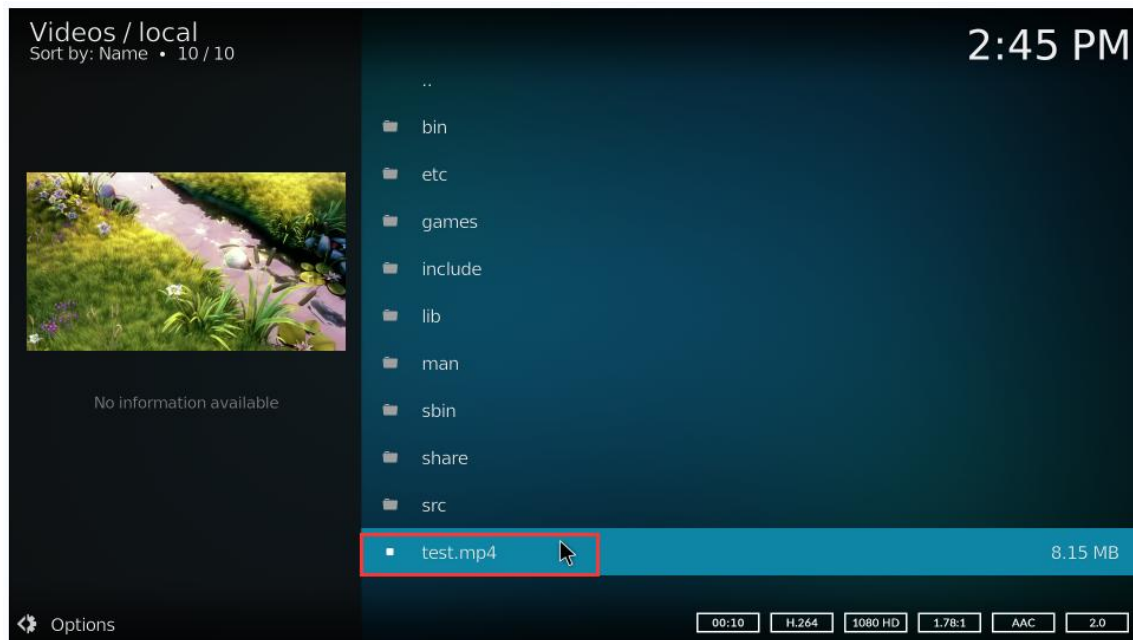
i. Select **OK**



j. Then enter the local folder



k. Then you can play the **test.mp4** test video

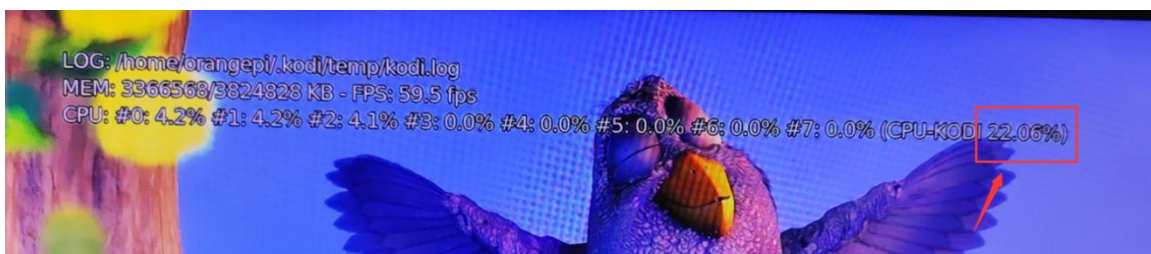


12) When playing the video, you can run the **vpu\_debug.sh** script on the command line (via ssh or serial port). If there is the following printout, it means that the video is decoded by hardware

```
orangeypi@orangeypi:~$ vpu_debug.sh
```

```
[ 1830.938378] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2728 us
[ 1830.938461] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2617 us
[ 1830.941179] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2661 us
[ 1830.941777] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2708 us
[ 1830.944727] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 3444 us
[ 1830.945211] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 3331 us
[ 1830.970563] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2547 us
[ 1831.199650] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2703 us
```

13) The CPU usage of playing the **test.mp4** video file is around **20%~30%** 占用率在 **20%~30%**.





## 4. 7. How to install ROS2 Humble on Ubuntu22.04 Gnome

1) Ros2 can be installed using the `install_ros.sh` script

```
orangeypi@orangeypi:~$ install_ros.sh ros2
```

2) The `install_ros.sh` script will automatically run the `ros2 -h` command after installing ros2. If you can see the following print, it means that the ros2 installation is complete.

```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...
```

ros2 is an extensible command-line tool for ROS 2.

optional arguments:

-h, --help show this help message and exit

Commands:

action	Various action related sub-commands
bag	Various rosbag related sub-commands
component	Various component related sub-commands
daemon	Various daemon related sub-commands
doctor	Check ROS setup and other potential issues
interface	Show information about ROS interfaces
launch	Run a launch file
lifecycle	Various lifecycle related sub-commands
multicast	Various multicast related sub-commands
node	Various node related sub-commands
param	Various param related sub-commands
pkg	Various package related sub-commands
run	Run a package specific executable
security	Various security related sub-commands
service	Various service related sub-commands
topic	Various topic related sub-commands
wtf	Use `wtf` as alias to `doctor`



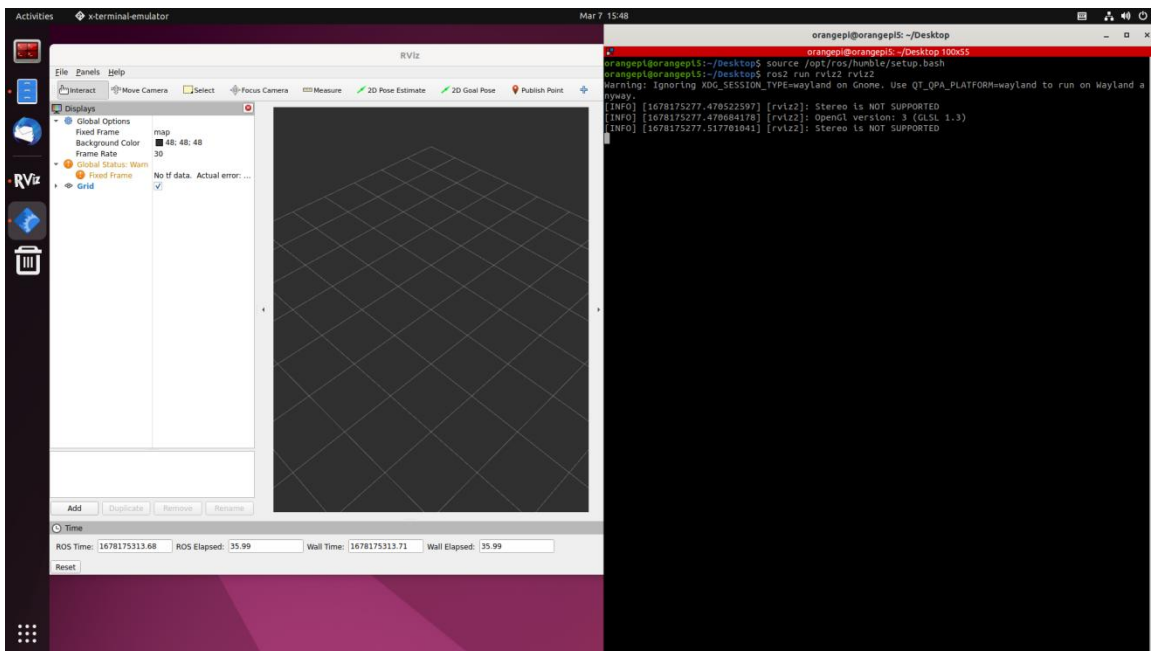
Call `ros2 <command> -h` for more detailed usage.

3) Then you can use the **test\_ros.sh** script to test whether ROS2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally.

```
orangepi@orangepi5b:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2.

```
orangepi@orangepi:~$ source /opt/ros/humble/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



5) Reference documents

<http://docs.ros.org/en/humble/index.html>

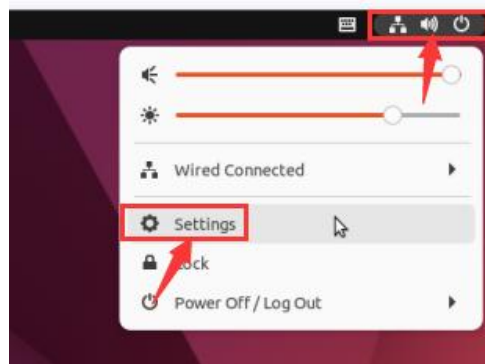
<http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>



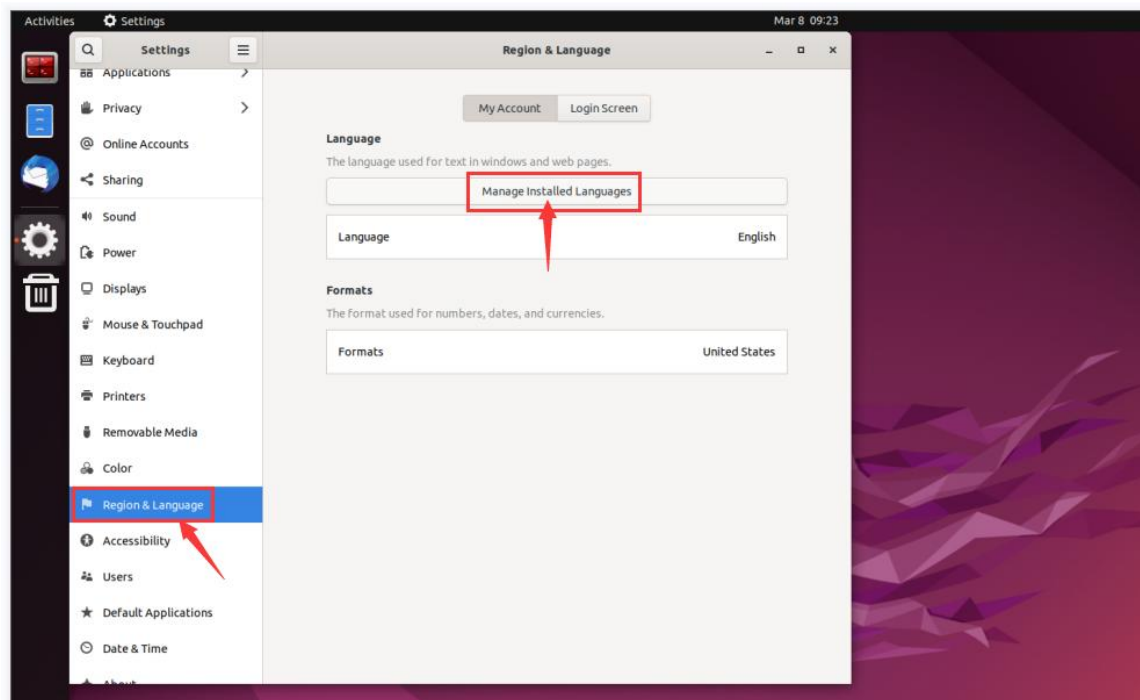


## 4.8. How to set Chinese environment and install Chinese input method

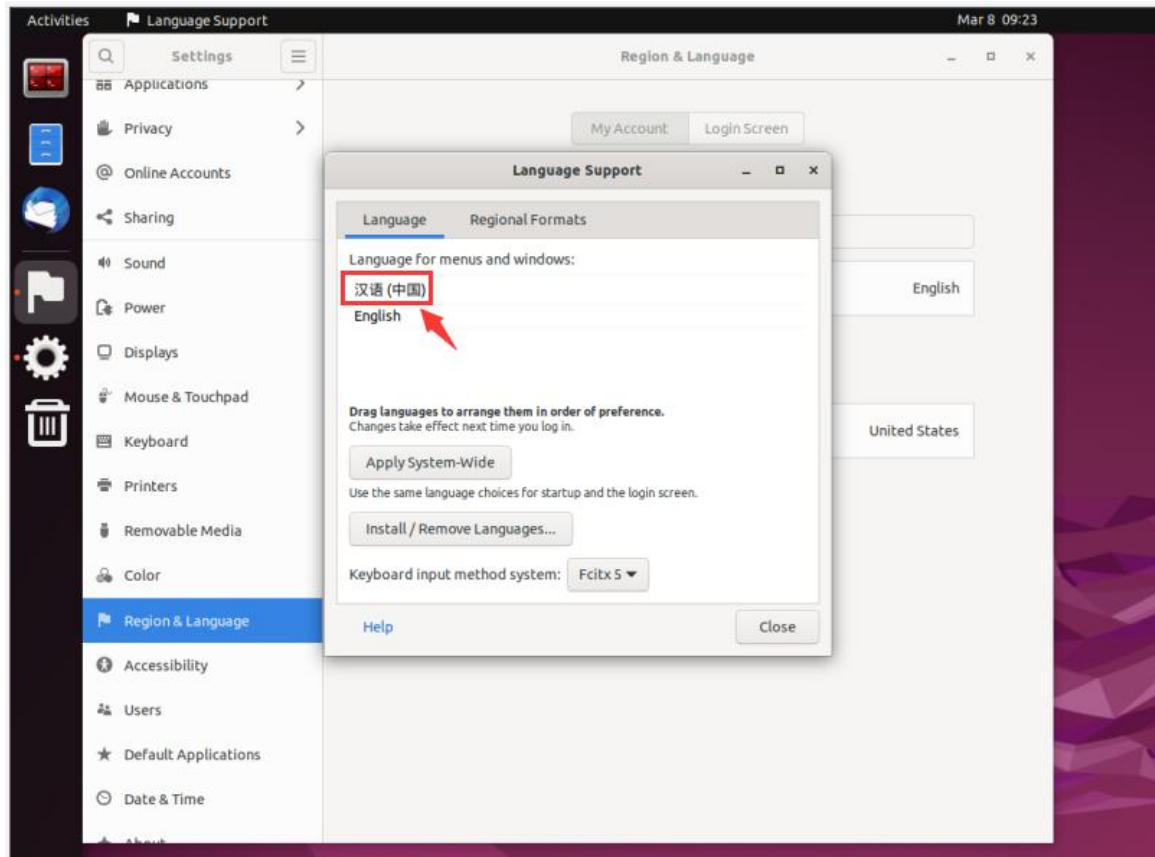
1) First open the settings



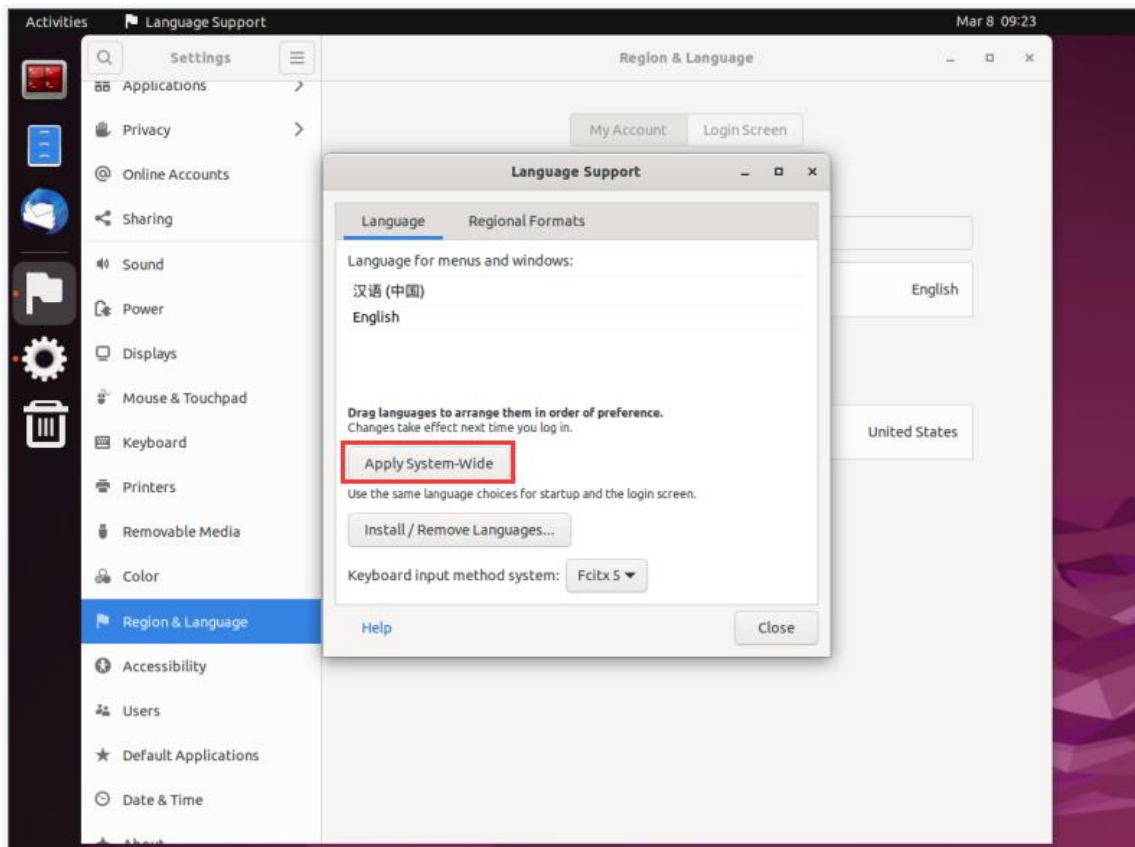
2) Then find the **Region & Language** option, and click the **Manage Installed Languages** option



3) Then please use the left mouse button to select **Chinese (China)** and hold it down, then drag it up to the initial position, and the display after dragging is shown in the figure below:

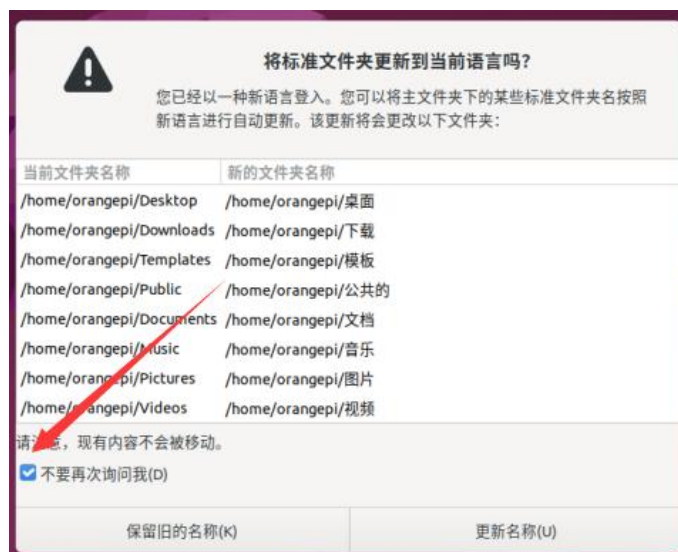


4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system.

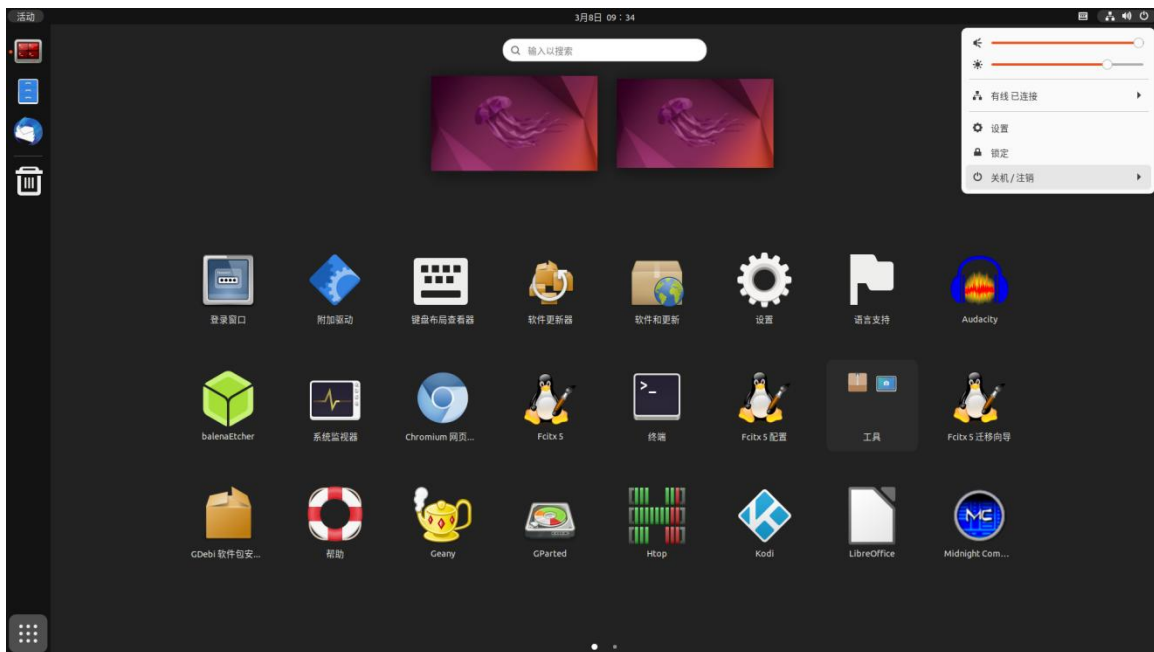


5) **Then restart the Linux system to make the configuration take effect**

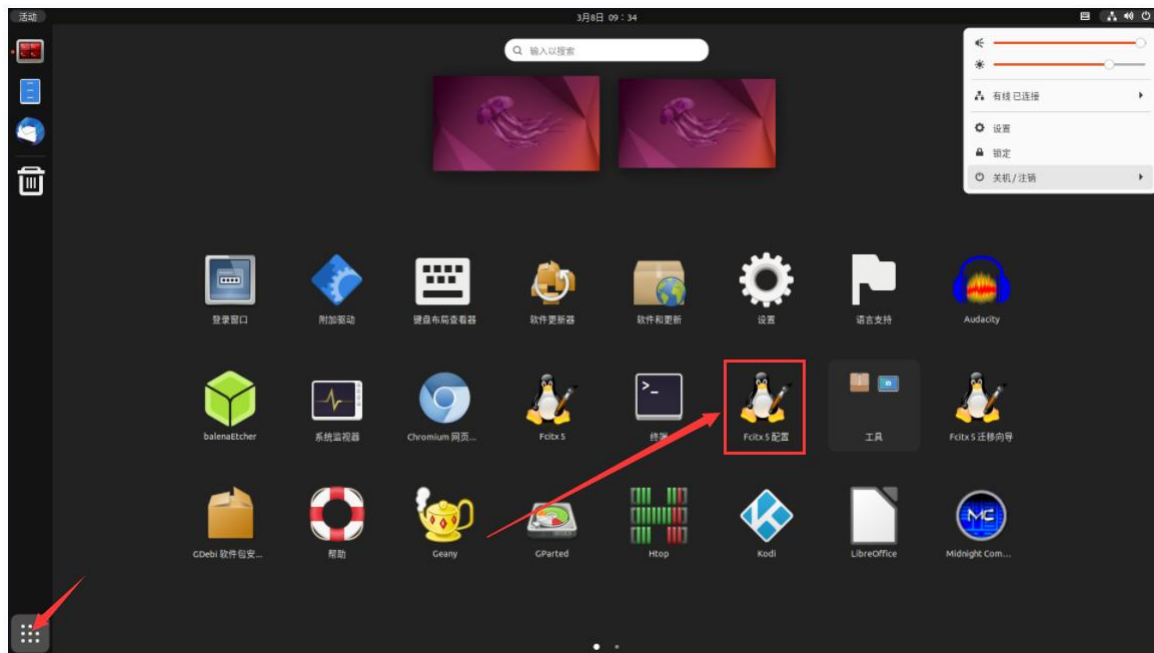
6) After re-entering the system, please choose **not to ask me again** in the following interface, and then please decide whether the standard folder should also be updated to Chinese according to your preferences



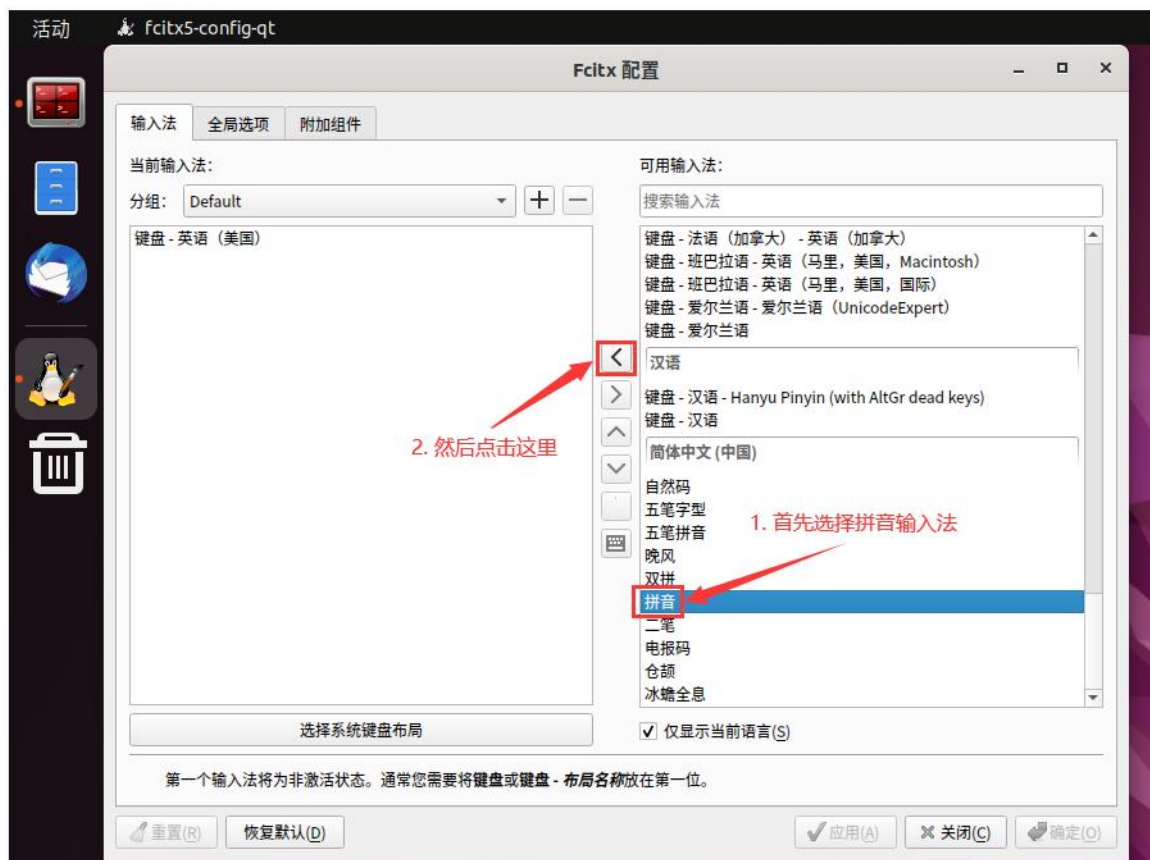
7) Then you can see that the desktop is displayed in Chinese



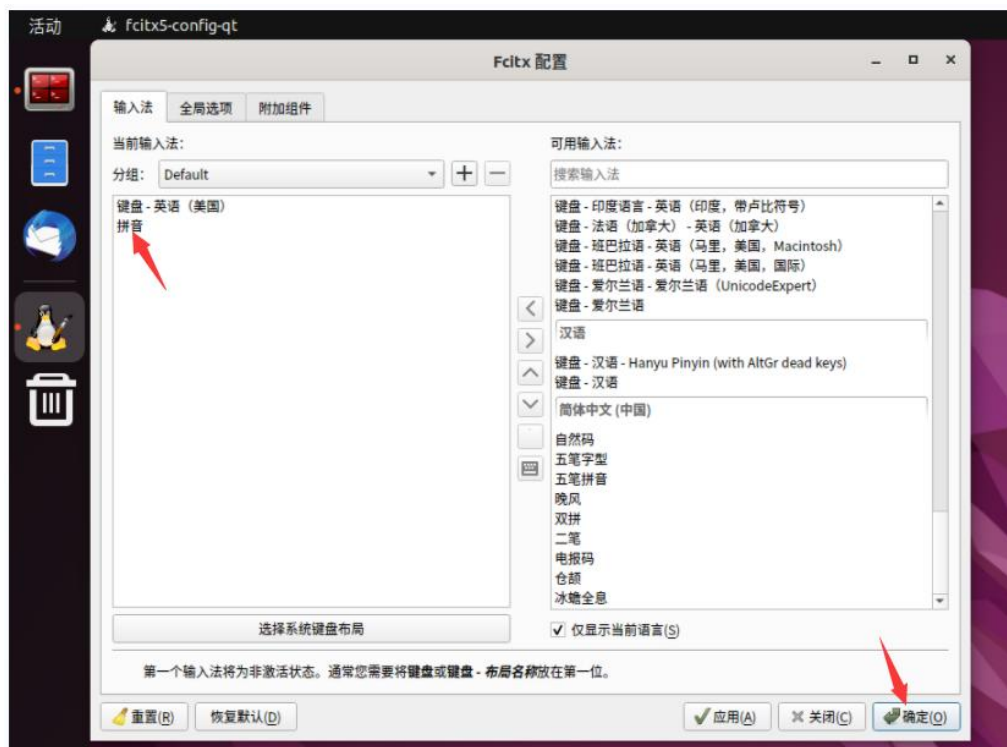
8) Then open the Fcix5 configuration program



9) Then choose to use Pinyin input method

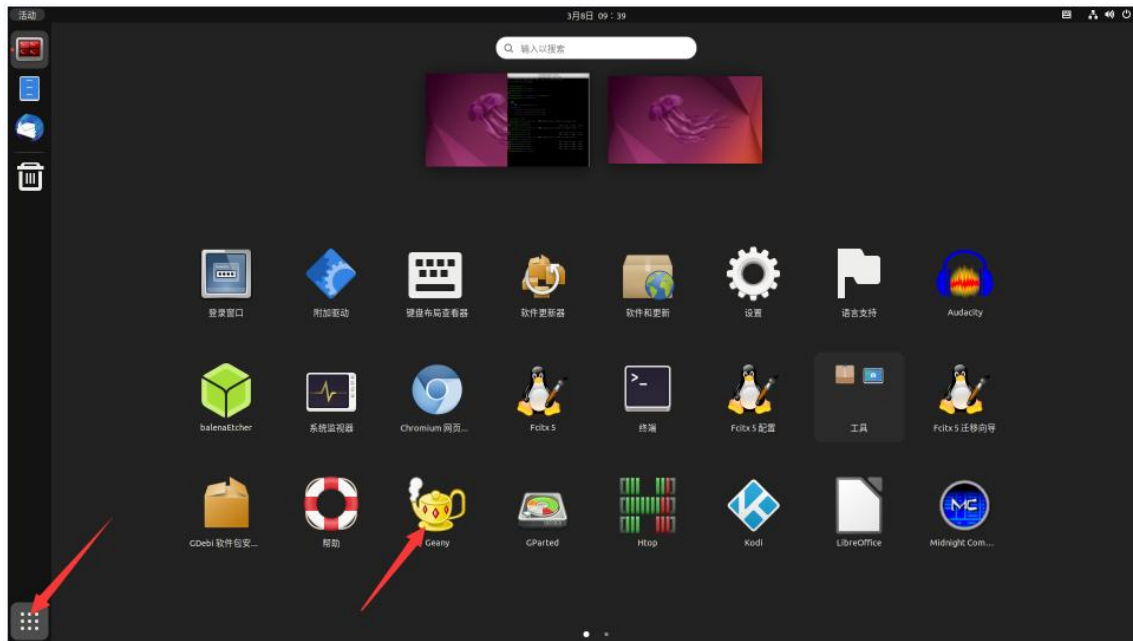


10) The interface after selection is as shown below, and then click OK

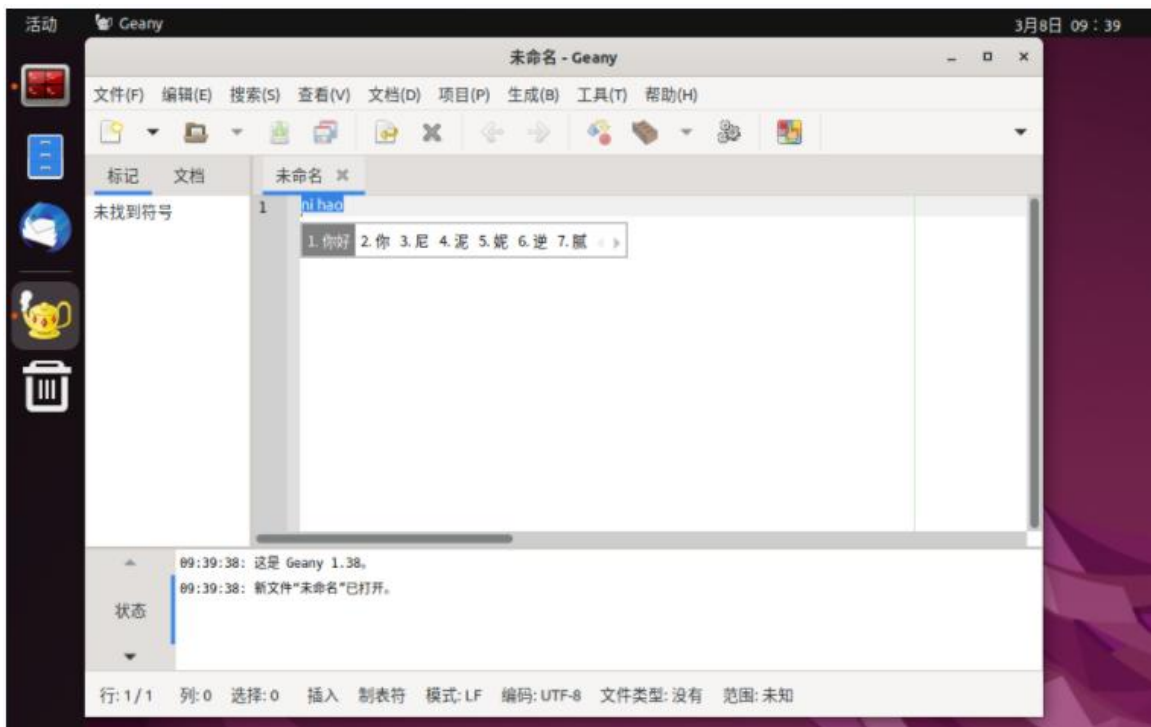




11) Then we can open **Geany** to test the Chinese input method, as shown in the figure below.



12) After opening **Geany**, the default is English input method, we can switch to Chinese input method through **Ctrl+Space** shortcut key, and then we can input Chinese



## 5. Orange Pi OS Arch System Instructions

### 5.1. Orange Pi OS Arch system adaptation status

Function	OPi OS Arch Gnome Wayland
USB2.0x2	OK
USB3.0x1	OK
USB Type-C 3.0	OK
eMMC Start the system	OK
DP display	OK
AP6275P-WIFI	OK
AP6275P-Bluetooth	OK
GPIO (26pin)	OK
UART (26pin)	OK
SPI (26pin)	OK
I2C (26pin)	OK
CAN (26pin)	OK
PWM (26pin)	OK
3pin Debug serial port	OK
TF card startup	OK
HDMI Video	OK
HDMI Audio	OK
OV13850 Camera	OK
OV13855 Camera	OK
LCD1	OK
LCD2	OK
Gigabit network port	OK
Network port status light	OK
MIC	OK
Headphone playback	OK
Headphone recording	OK





<b>LED light</b>	<b>OK</b>
<b>GPU</b>	<b>OK</b>
<b>NPU</b>	<b>NO</b>
<b>VPU</b>	<b>OK</b>
<b>On/off button</b>	<b>OK</b>
<b>watchdog test</b>	<b>OK</b>
<b>Chromiumhard decryption video</b>	<b>NO</b>
<b>MPV hard decoding playback video</b>	<b>OK</b>

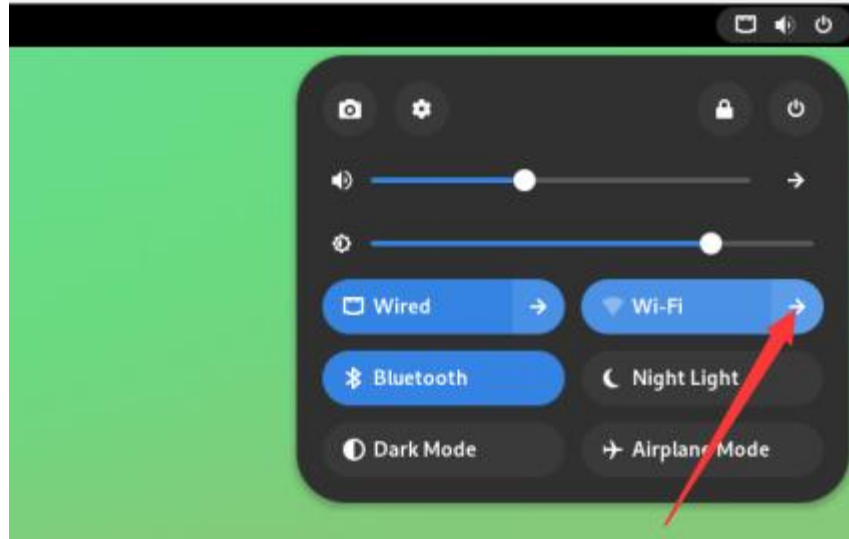
## 5.2. How to use AP6275P PCIe WIFI6+ Bluetooth module

1) The steps to connect to WIFI are as follows:

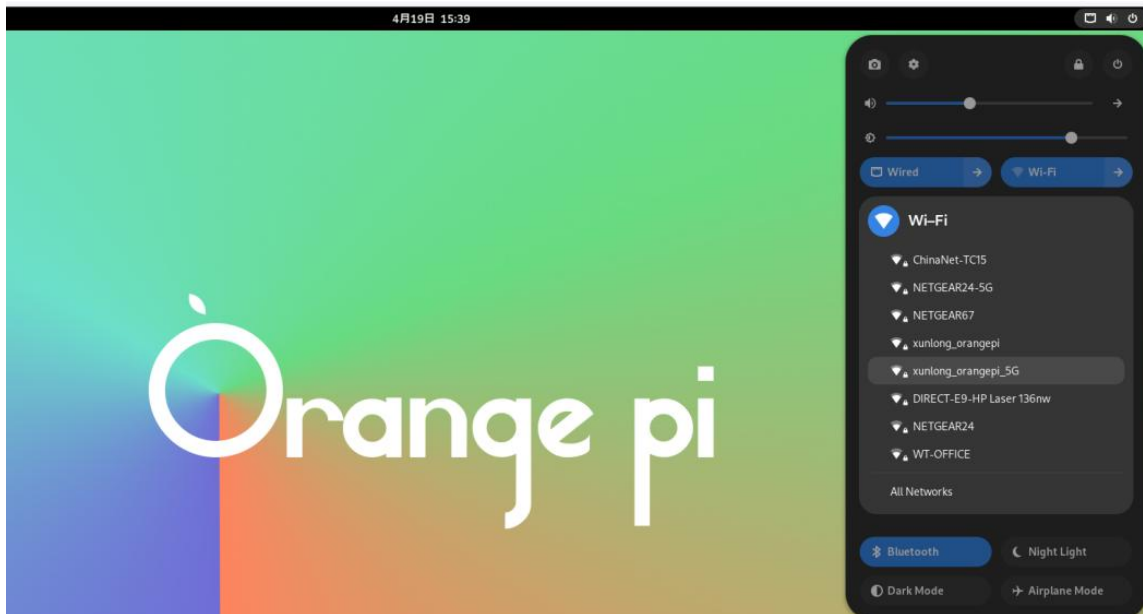
- a. First click on the area in the upper right corner of the desktop



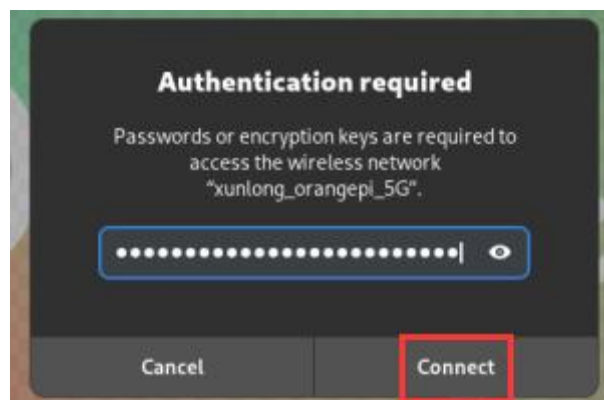
- b. Then select Wi-Fi



c. Then select the WIFI you want to connect to

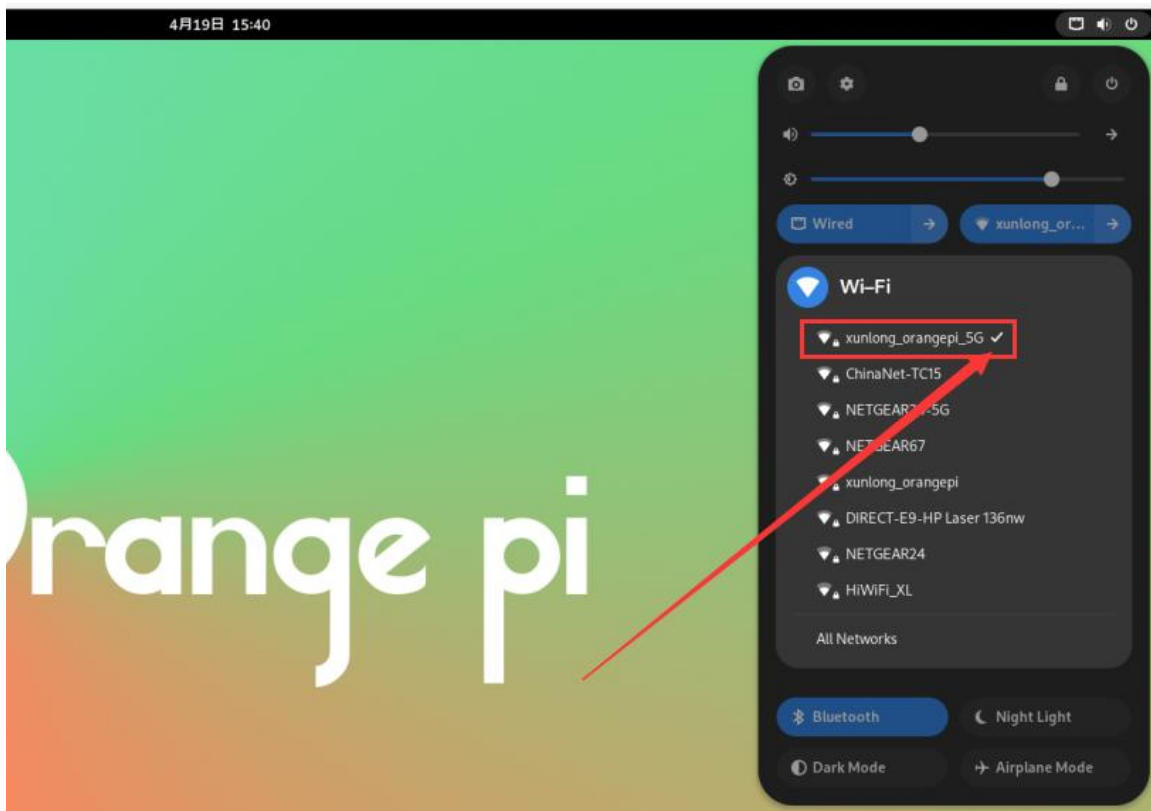


d. Then enter the WIFI password and click **Connect**





- e. Then enter the following interface again and you will see that WIFI is connected.

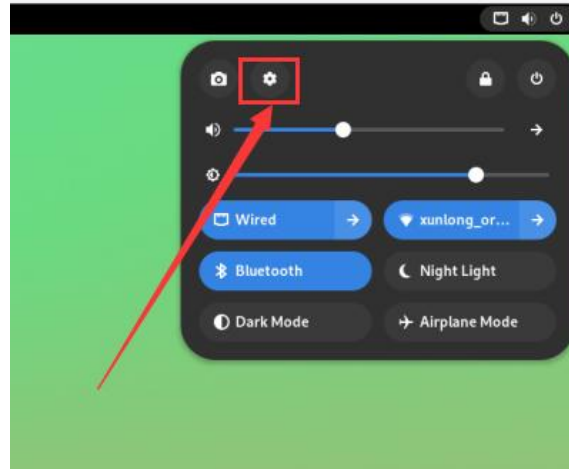


## 2) Bluetooth usage example:

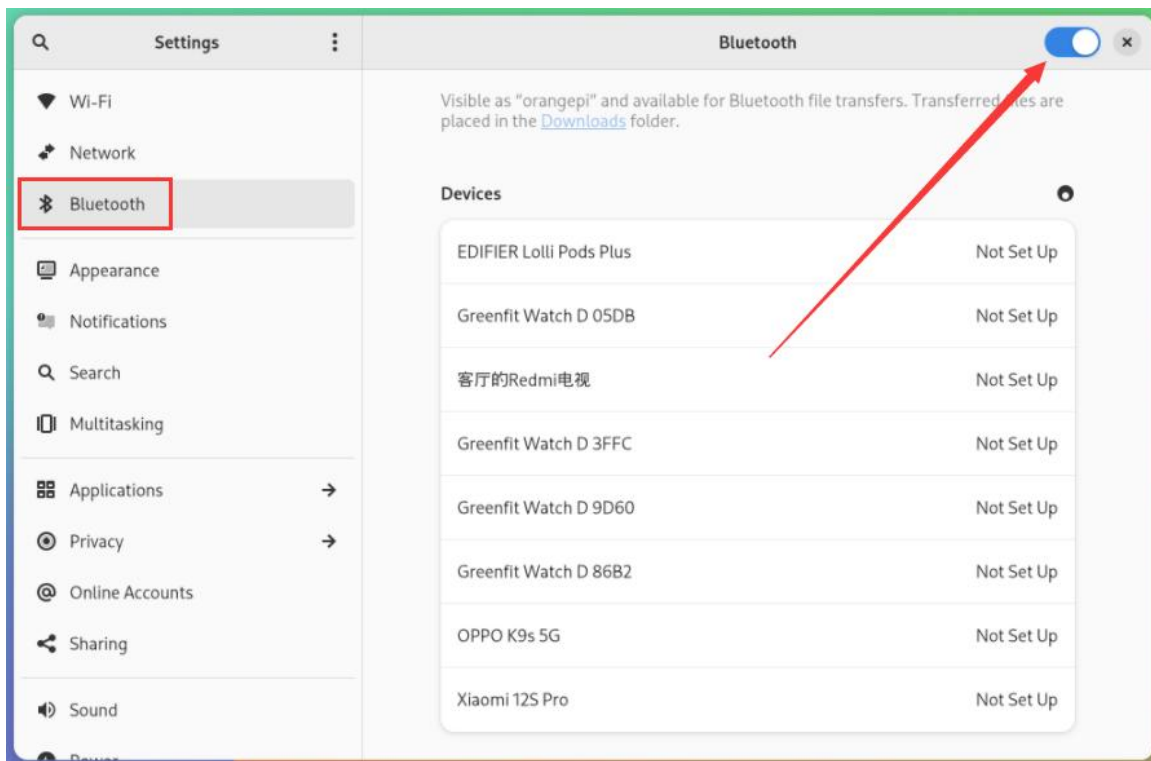
- a. First click on the area in the upper right corner of the desktop



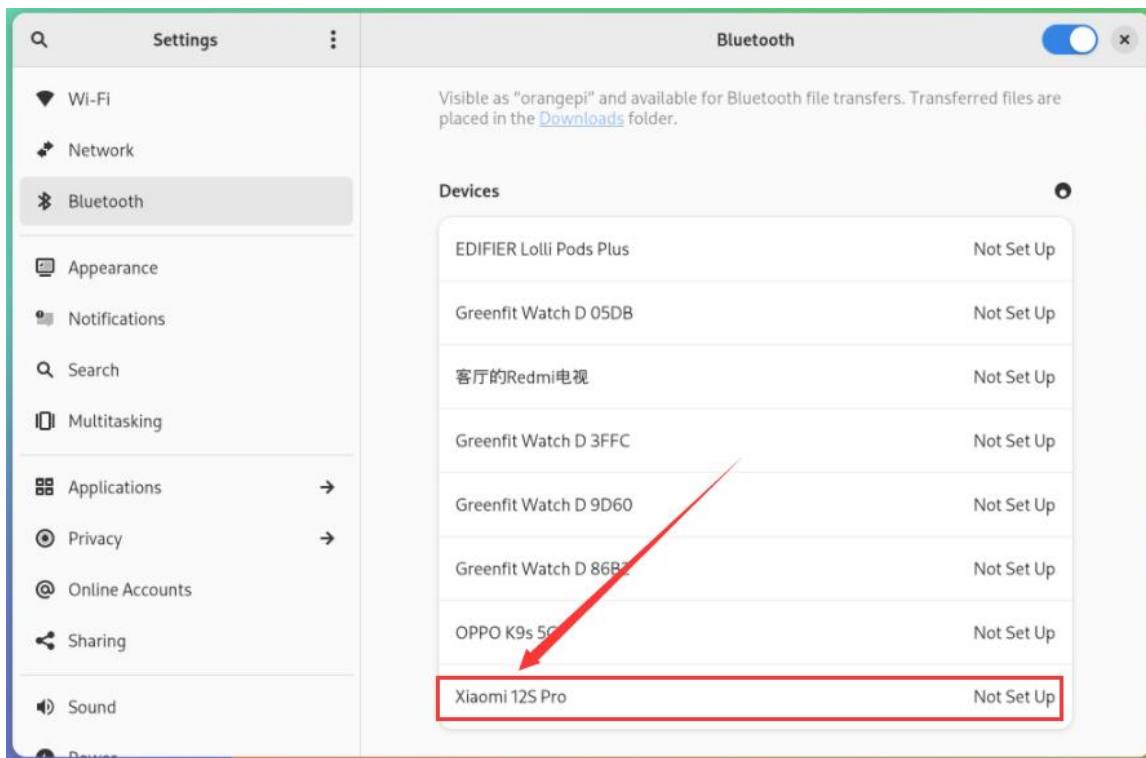
- b. Then open settings



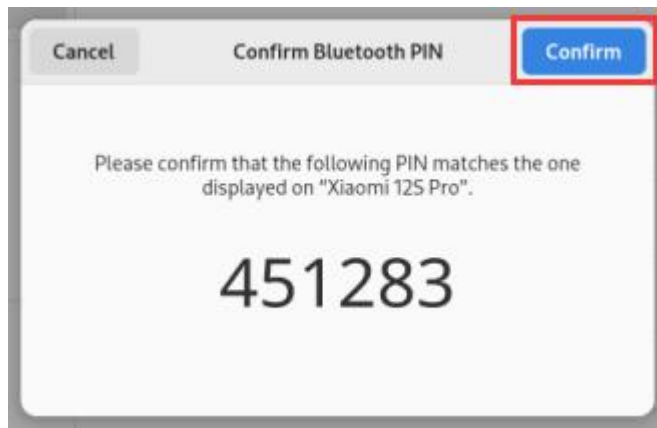
- c. Then select Bluetooth in settings and make sure the switch button in the upper right corner of Bluetooth is turned on



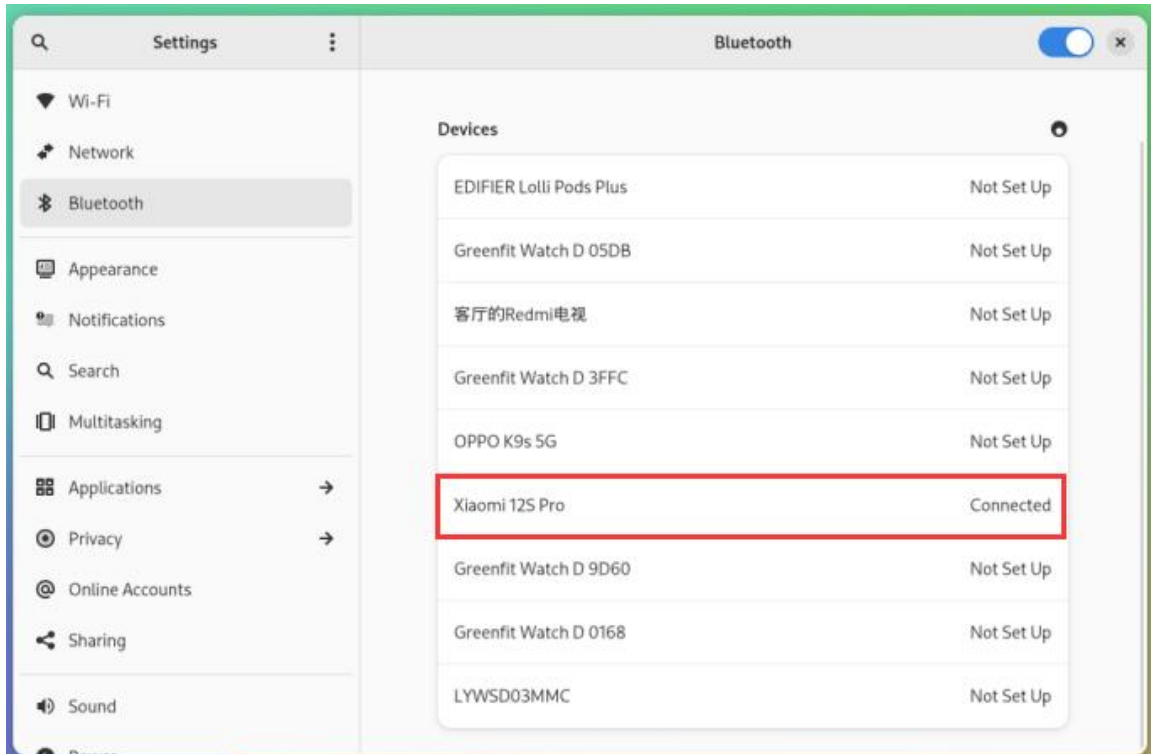
- d. Then select the Bluetooth device you want to configure, such as pairing with an Android phone.



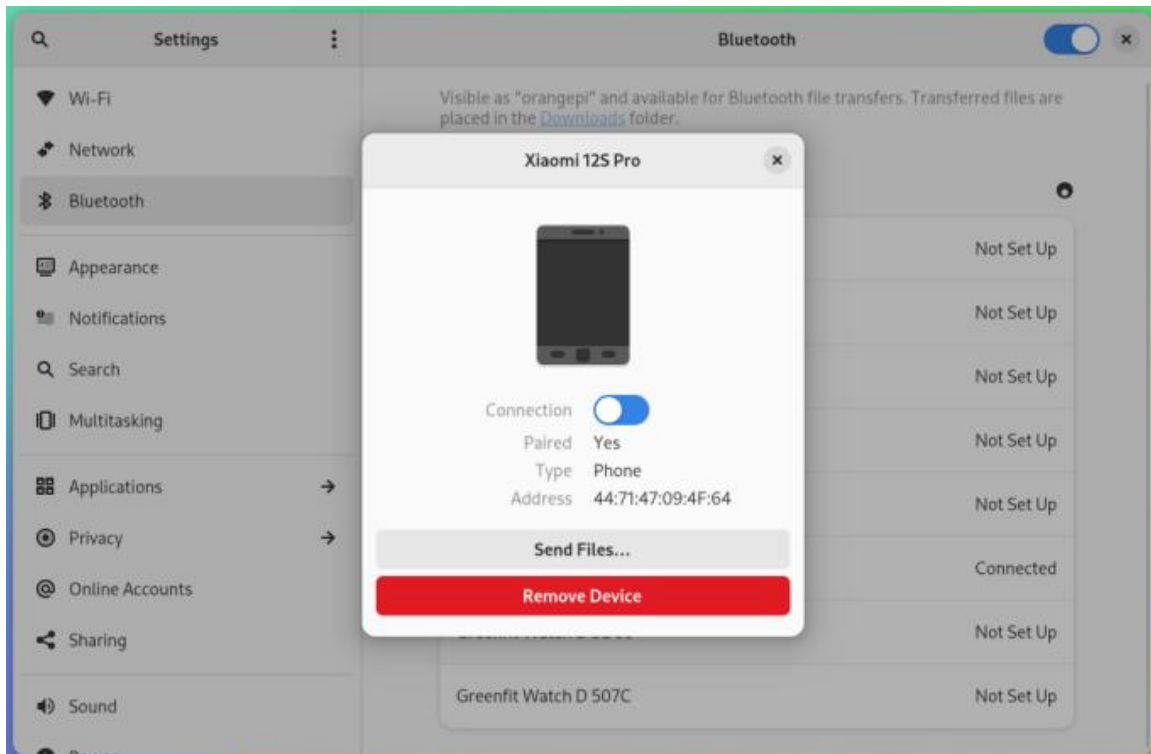
- e. Then click **Confirm**. The mobile phone also needs to confirm pairing.



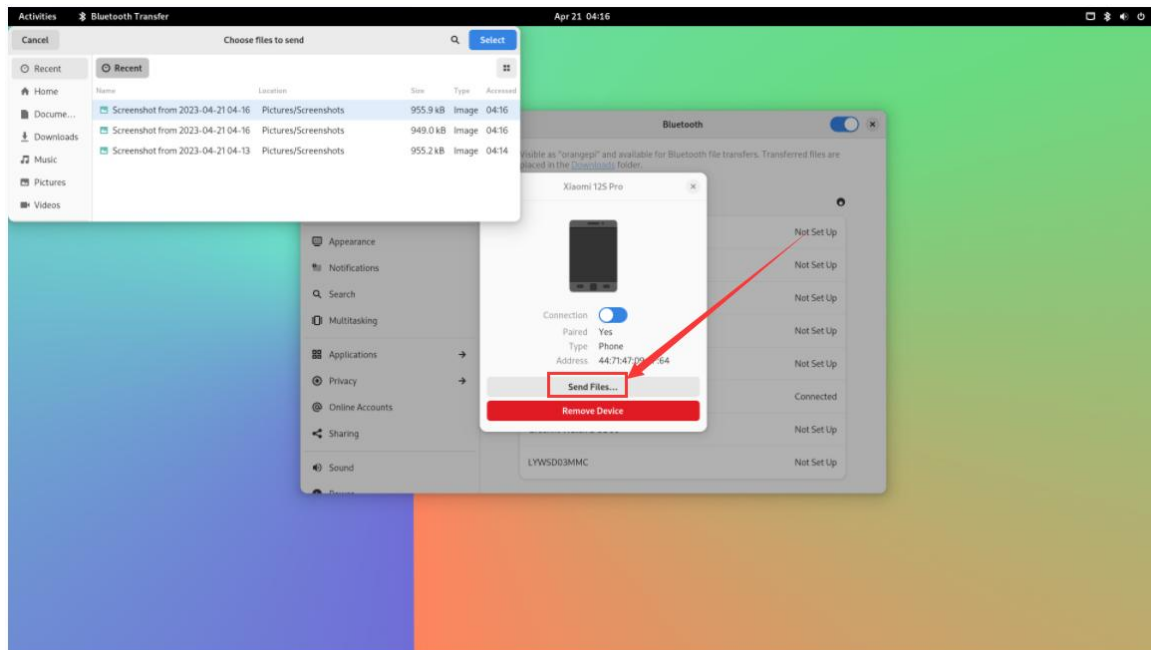
- f. The display after Bluetooth and Android phone are connected is as follows:



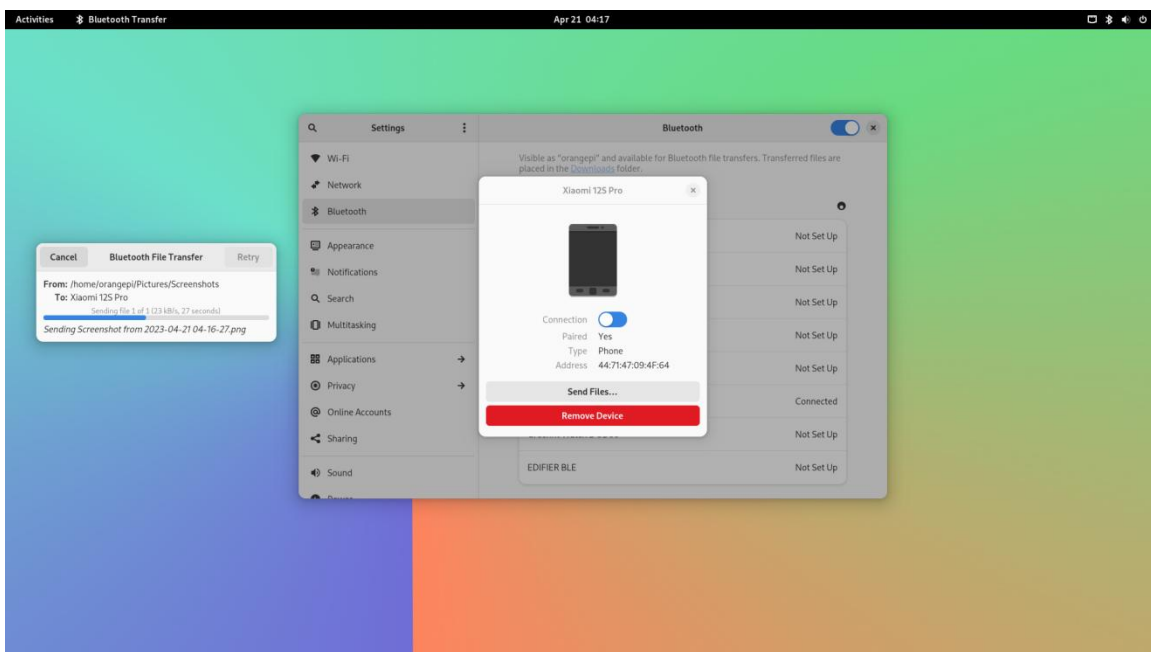
- g. Then click on the paired Bluetooth device and the operation interface shown in the figure below will pop up.



- h. Click **Send Files...** at this time to send a file to the mobile phone



- i. The schematic diagram of sending pictures to the mobile phone via Bluetooth is as follows:





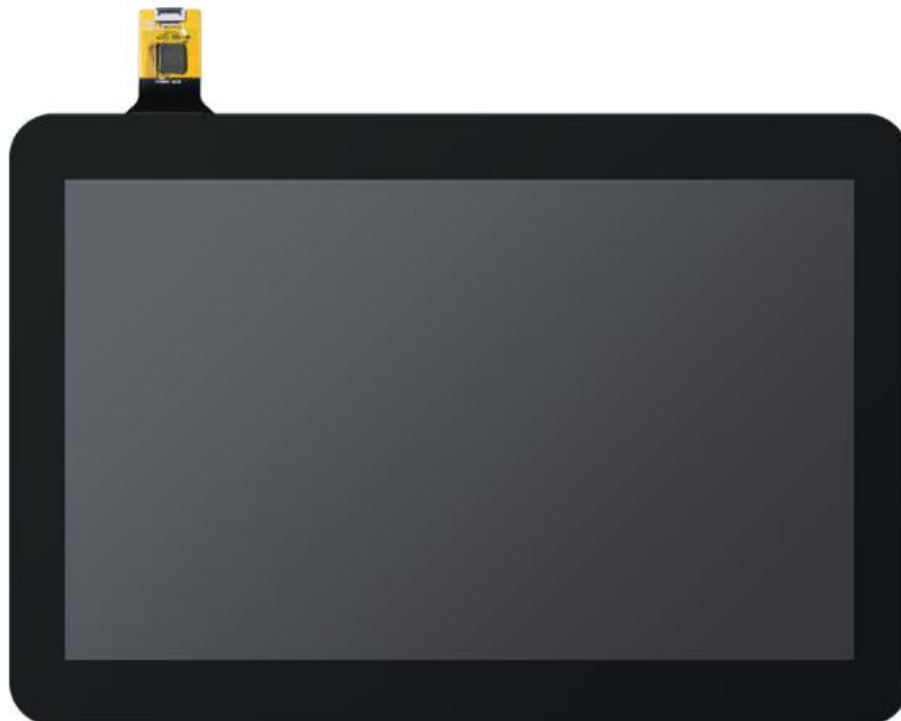


## 5.3. How to use the 10.1-inch MIPI LCD screen

### 5.3.1. 10.1-inch MIPI screen assembly method

1) First prepare the necessary accessories

a. 10.1 inch MIPI LCD display + touch screen



b. Screen adapter board + 31pin to 40pin cable



c. 30pin MIPI cable



d. 12pin touch screen cable



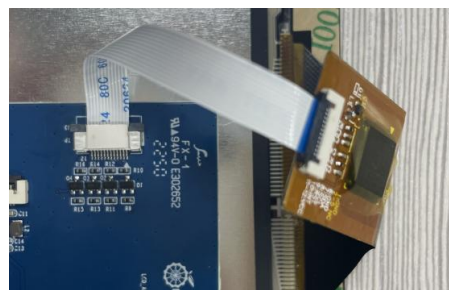
2) Connect the 12pin touch screen cable, 31pin to 40pin cable, and 30pin MIPI cable to the screen adapter board as shown below. Note **that the blue insulated side of the touch screen cable is facing down**, and the insulated side of the other two cables is facing up. , if connected incorrectly, it will cause no display or inability to touch.



3) Place the adapter board connected with the cable on top of the MIPI LCD screen as shown below, and connect the MIPI LCD screen and the adapter board through a 31pin to 40pin cable.



4) Then connect the touch screen and the adapter board through the 12pin touch screen cable, paying attention to the orientation of the insulation surface



5) Finally, connect it to the LCD interface of the development board through a 30pin MIPI cable.

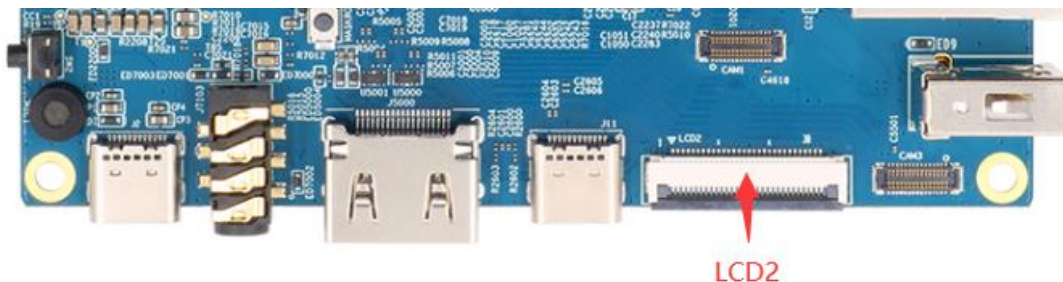


### 5. 3. 2. How to open the 10.1-inch MIPI LCD screen configuration

- 1) The OPi OS Arch image does not have the configuration to open the mipi LCD screen by default. If you need to use the mipi LCD screen, you need to open it manually.
- 2) There are two MIPI LCD screen interfaces on the development board. We define:
  - a. The location of a.lcd1 interface is:



- b. The location of b.lcd2 interface is:



- 3) The method of opening mipi lcd configuration is as follows:
  - a. If you want to open LCD1, please add the following configuration to **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
```



```
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
```

```
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-lcd1.dtbo #Configuration that  
needs to be added
```

- b. If you want to turn on LCD2, please add the following configuration to

**/boot/extlinux/extlinux.conf**

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

```
LABEL Orange Pi
```

```
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
```

```
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-lcd2.dtbo #Configuration that  
needs to be added
```

- c. If you want to open LCD1 and LCD2 at the same time, please add the following configuration to **/boot/extlinux/extlinux.conf** (the configuration of **the two LCDs needs to be written on one line, please do not write it on two lines**)

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

```
LABEL Orange Pi
```

```
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
```

```
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-lcd1.dtbo /dtbs/rockchip/overlay/rk3588-lcd2.dtbo
```

#### 4) **Then restart the OPi OS Arch system**

5) After restarting, you can see the LCD screen display as follows (default is vertical screen):



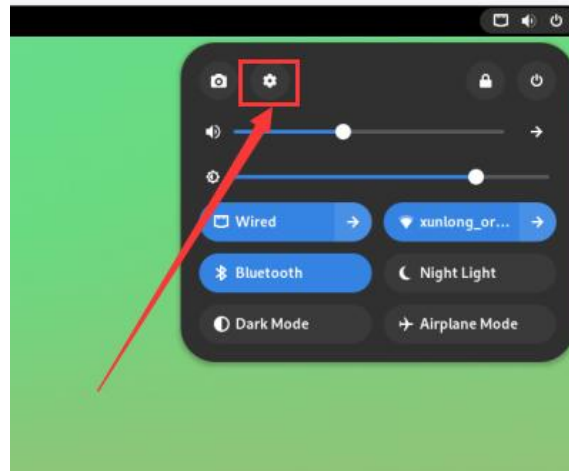
### 5. 3. 3. Methods of rotating display and touch direction

1) First click on the area in the upper right corner of the desktop

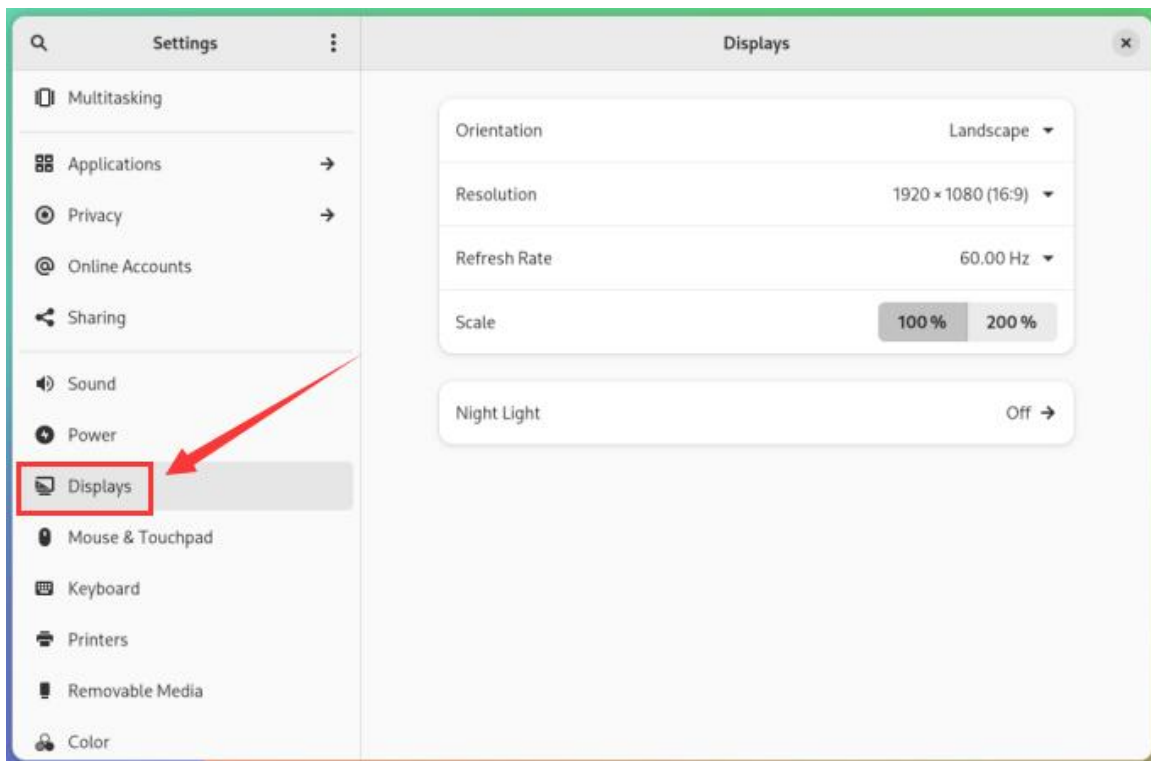


2) Then open settings



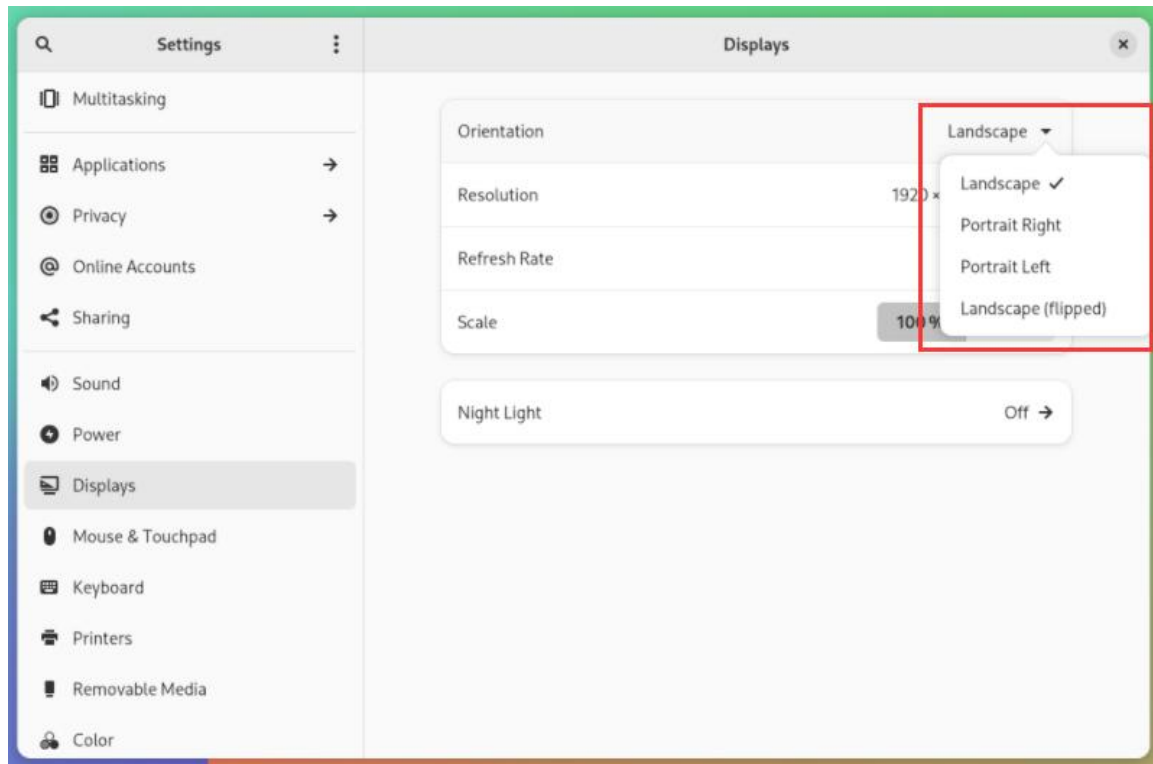


3) Then select **Displays**

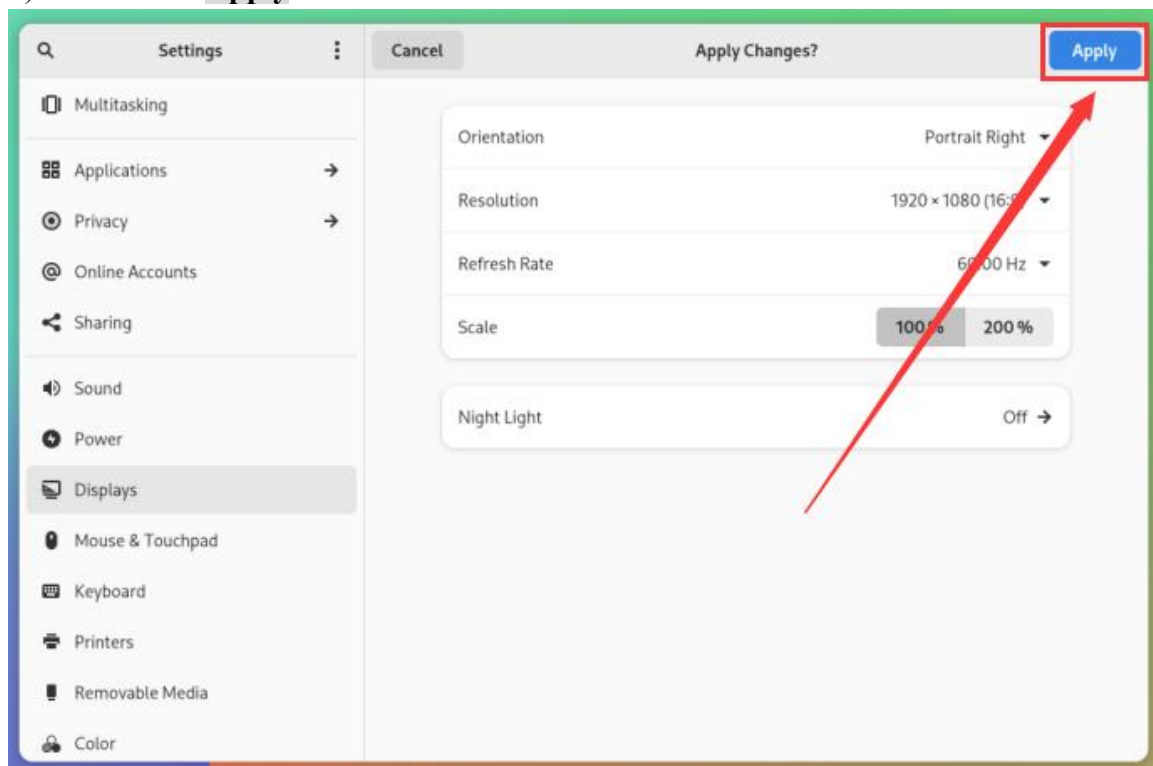


4) Then select the direction you want to rotate in the **Orientation** of **Displays**





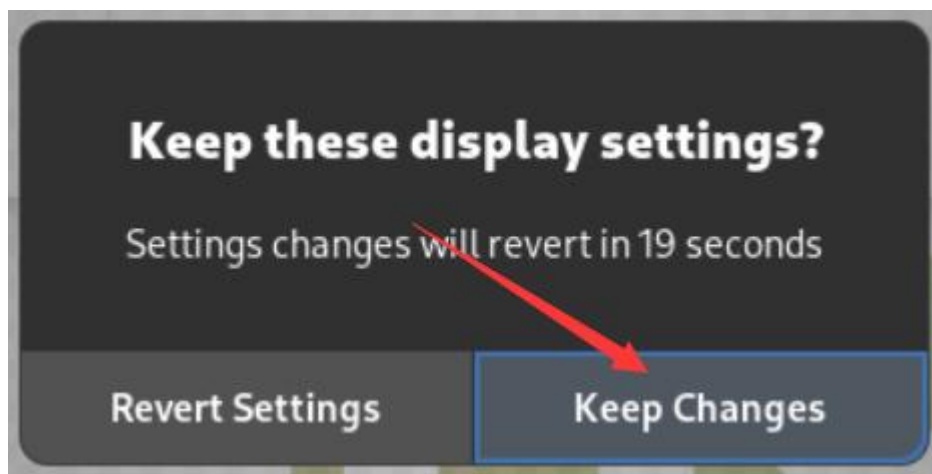
5) Then select **Apply**



6) Then you can see that the screen has been rotated. At this time, you need to select



**Keep Changes** to finalize the rotation.



7) The display after the LCD screen is rotated 90 degrees is as follows:



8) The touch function of the OPi OS Arch system LCD screen will rotate with the rotation of the display direction, no other settings are required.

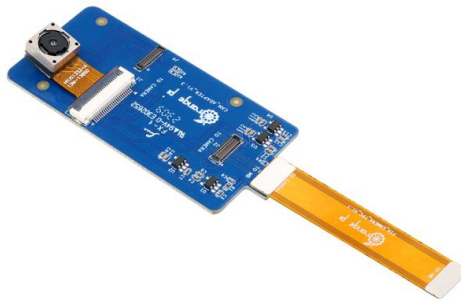
#### **5.4. Testing methods for OV13850 and OV13855 MIPI cameras**

Currently, the development board supports two MIPI cameras, OV13850 and

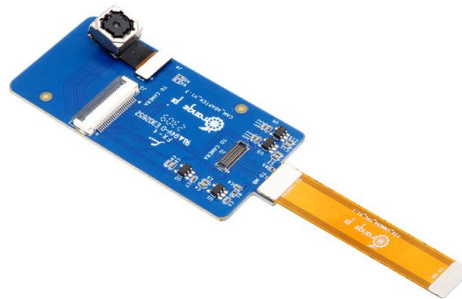


OV13855. The specific pictures are as follows:

- a. OV13850 camera with 13 million MIPI interface



- b. OV13855 camera with 13 million MIPI interface

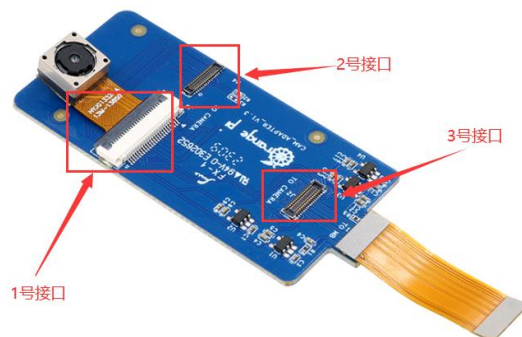


The adapter board and FPC cable used by the OV13850 and OV13855 cameras are the same, but the locations where the two cameras are connected to the adapter board are different. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked TO MB needs to be plugged into the camera interface of the development board, and the end marked TO CAMERA needs to be plugged into the camera adapter board.

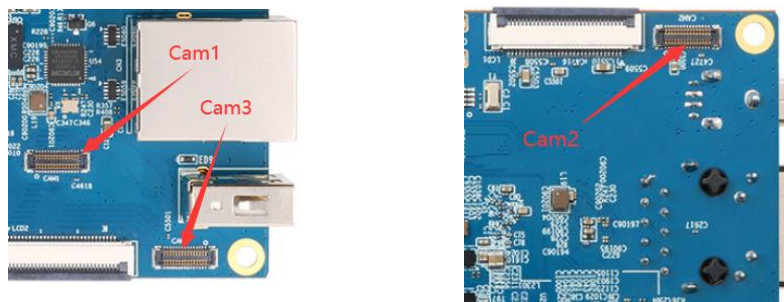


There are a total of 3 camera interfaces on the camera adapter board, and only one can be connected and used at the same time, as shown in the figure below, among which:

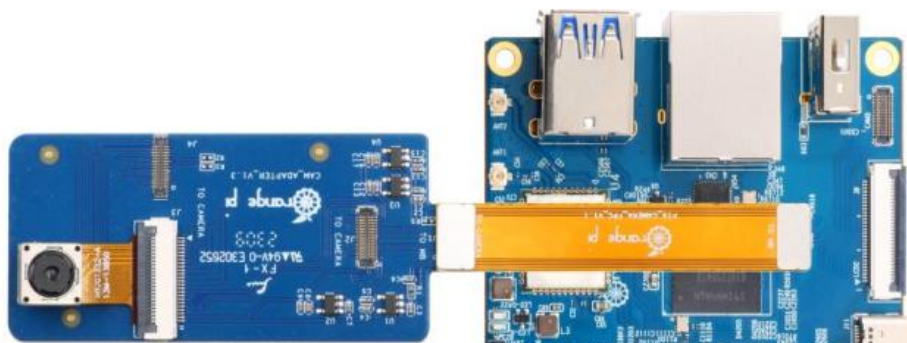
- d. **Connect the OV13850 camera to interface 1**
- e. **No. 2 interface is connected to the OV13855 camera**
- f. Interface No. 3 is not used, just ignore it.



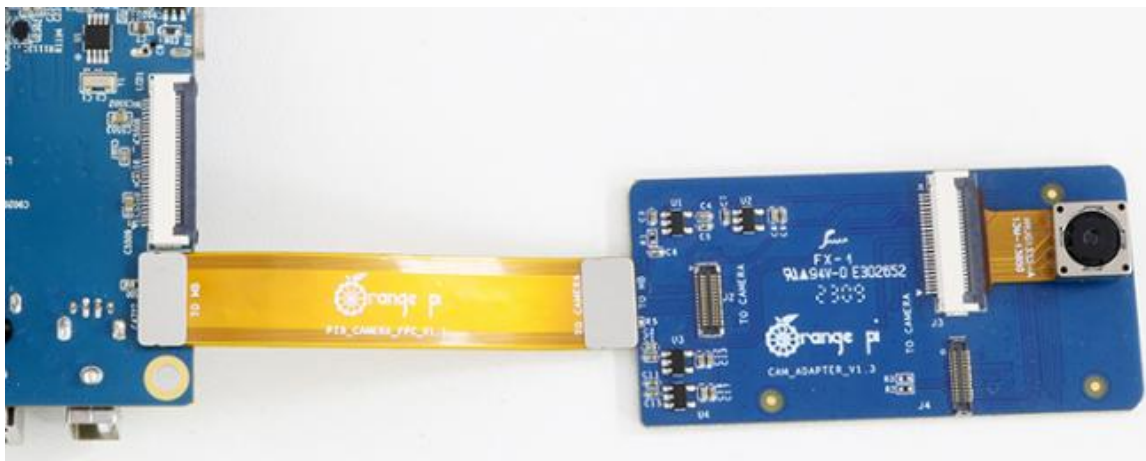
There are a total of 3 camera interfaces on the Orange Pi 5B development board. We define the positions of Cam1, Cam2 and Cam3 as shown in the figure below:



The method of plugging the camera into the Cam1 interface of the development board is as follows:



The method of plugging the camera into the Cam2 interface of the development board is as follows:



The method of plugging the camera into the Cam3 interface of the development board is as follows:



After connecting the camera to the development board, we can use the following method to test the camera:

- First add the following configuration to **/boot/extlinux/extlinux.conf**

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-ov13850-c1.dtbo #Configuration
that needs to be added
```

The red font above demonstrates the configuration of opening the **Cam1 interface to connect to ov13850**. The configuration of other interfaces is as shown in the following table. Just add the corresponding dtbo configuration after **FDTOVERLAYS**. If you want to add multiple configurations at the same time, separate them with spaces.



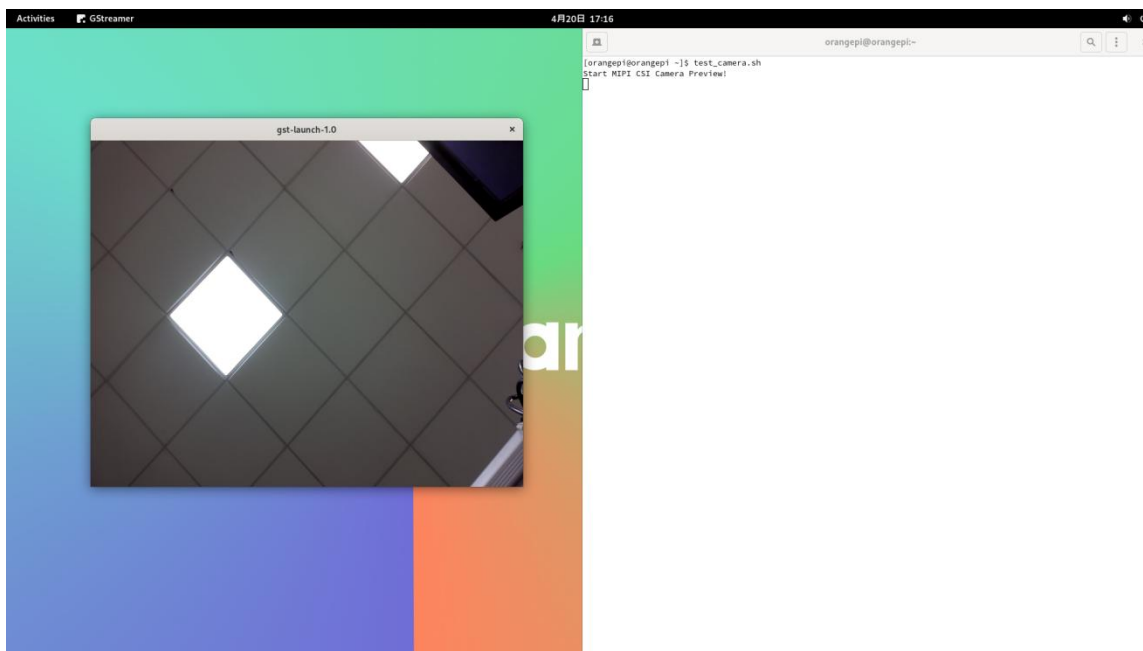
Camera	dtbo configuration
Cam1 connected to ov13850	/dtbs/rockchip/overlay/rk3588-ov13850-c1.dtbo
Cam2 connected to ov13850	/dtbs/rockchip/overlay/rk3588-ov13850-c2.dtbo
Cam3 connected to ov13850	/dtbs/rockchip/overlay/rk3588-ov13850-c3.dtbo
Cam1 connected to ov13855	/dtbs/rockchip/overlay/rk3588-ov13855-c1.dtbo
Cam2 connected to ov13855	/dtbs/rockchip/overlay/rk3588-ov13855-c2.dtbo
Cam3 connected to ov13855	/dtbs/rockchip/overlay/rk3588-ov13855-c3.dtbo

b. **Then restart the OPi OS Arch system**

c. Then open a terminal on the desktop system and run the following script

```
orangeypi@orangeypi:~$ test_camera.sh
```

d. Then you can see the preview screen of the camera





## 5.5. How to install wiringOP

Note that wiringOP is already pre-installed in the OPi OS Arch image released by Orange Pi. Unless the wiringOP code is updated, there is no need to re-download, compile and install it, you can use it directly.

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

```
[orangepi@orangepi ~]$ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 47 | 0 | 3.3V | | | 1 | 2 | | | 5V | | |
| 46 | 1 | SDA.5 | IN | 1 | 3 | 4 | | | 5V | | |
| 54 | 2 | SCL.5 | IN | 1 | 5 | 6 | | | GND | | |
| | | PWM15 | IN | 1 | 7 | 8 | 0 | IN | 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 | | | PowerKey | | |
| 50 | 13 | SPI4_CLK | IN | 1 | 23 | 24 | 1 | IN | SPI4_CS1 | 14 | 52 |
| | | GND | | | 25 | 26 | 1 | IN | PWM1 | 15 | 35 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| | | | | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
[orangepi@orangepi ~]$
```

wiringOP currently mainly adapts to the functions of setting GPIO port input and output, setting GPIO port output high and low levels, and setting pull-up and pull-down resistors. Functions such as hardware PWM cannot be used.

### 1) Download the code of wiringOP

```
[orangepi@orangepi ~]$ sudo pacman -Syy git
[orangepi@orangepi ~]$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

Note that Orange Pi 5B needs to download the code of wiringOP next branch, please don't miss the -b next parameter.

If there is a problem downloading the code from GitHub, you can download the source code





compressed package of wiringOP.tar.gz from the [official tool on the Orange Pi 5B data download page](#).



## 2) Compile and install wiringOP

```
[orangepi@orangepi ~]$ sudo pacman -Syy make gcc
[orangepi@orangepi ~]$ cd wiringOP
[orangepi@orangepi wiringOP]$ sudo ./build clean
[orangepi@orangepi wiringOP]$ sudo ./build
```

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

```
[orangepi@orangepi ~]$ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 47 | 0 | 3.3V | IN | 1 | 1 | 2 | | | 5V | | |
| 46 | 1 | SDA.5 | IN | 1 | 3 | 4 | | | 5V | | |
| 54 | 2 | SCL.5 | IN | 1 | 5 | 6 | | | GND | | |
| | | PWM15 | IN | 1 | 7 | 8 | 0 | IN | 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 | | | PowerKey | | |
| 50 | 13 | SPI4_CLK | IN | 1 | 23 | 24 | 1 | IN | SPI4_CS1 | 14 | 52 |
| | | GND | | | 25 | 26 | 1 | IN | PWM1 | 15 | 35 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| | | | | | | | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
[orangepi@orangepi ~]$
```

## 5.6. 26pin interface GPIO, I2C, UART, SPI, CAN and PWM testing

Note that if you need to set up fdt overlays to open multiple configurations at the same time, please separate them with spaces and write them on one line like the red font configuration below.

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

LABEL Orange Pi



```
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
```

```
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c1-m2.dtbo /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo
```

### 5. 6. 1. 26pin GPIO port test

1) There are a total of 16 GPIO ports in the 26-pin development board that can be used. The following uses pin No. 7 - the corresponding GPIO is GPIO1\_C6 - the corresponding wPi serial number is 2 - as an example to demonstrate how to set the high and low levels of the GPIO port.

```
[orangepi@orangepi ~]$ gpio readall
```

					PISB						
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	RXD.0	3	131	
		GND			9	10	0	TXD.0	4	132	
138	5	CAN1 RX	IN	1	11	12	1	CAN2 TX	6	29	

2) First set the GPIO port to output mode, and the third parameter needs to be the serial number of the wPi corresponding to the input pin.

```
[orangepi@orangepi ~]$ gpio mode 2 out
```

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

```
[orangepi@orangepi ~]$ gpio write 2 0
```

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

```
[orangepi@orangepi ~]$ gpio readall
```

					PI5B						
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	RXD.0	3	131	
		GND			9	10	0	TXD.0	4	132	
138	5	CAN1 RX	IN	1	11	12	1	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 the setting of the high level is successful.



```
[orangeypi@orangeypi ~]$ gpio write 2 1
```

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

```
[orangeypi@orangeypi ~]$ gpio readall
```

GPIO	wPi	Name	Mode	V	PI5B	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	IN	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 for other pins is similar. You only need to modify the serial number of wPi to the serial number corresponding to the pin.

### 5. 6. 2. 26pin GPIO port pull-down resistor setting method

**Note that only the lower 4 GPIO pins of Orange Pi 5 can normally set the pull-up and pull-down resistor functions. The other GPIO pins have external 3.3V pull-ups, so the pull-down settings are invalid.**

```
[orangeypi@orangeypi ~]$ gpio readall
```

GPIO	wPi	Name	Mode	V	PI5B	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	IN	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			PowerKey		
50	13	SPI4_CLK	IN	1	23	24	1	IN	SPI4_CS1	14	52
		GND			25	26	1	IN	PWM1	15	35

```
[orangeypi@orangeypi ~]$
```

1) The following uses pin No. 11 - the corresponding GPIO is GPIO4\_B2 - the corresponding wPi serial number is 5 - as an example to demonstrate how to set the pull-up and pull-down resistors of the GPIO port.



```
[orangeypi@orangeypi ~]$ gpio readall
```

PI5B											
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	RXD.0	3	131	
		GND			9	10	0	TXD.0	4	132	
138	5	CAN1_RX	IN	1	11	12	1	CAN2_TX	6	29	
139	7	CAN1_TX	IN	1	13	14		GND			
28	8	CAN2_RX	IN	1	15	16	1	SDA.1	9	59	
		3.3V			17	18	1	SCL.1	10	58	

2) First, you need to set the GPIO port to input mode, and the third parameter needs to be the serial number of the wPi corresponding to the input pin.

```
[orangeypi@orangeypi ~]$ gpio mode 5 in
```

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

```
[orangeypi@orangeypi ~]$ gpio mode 5 up
```

4) Then enter the following command to read the level of the GPIO port. If the level is 1, it means that the pull-up mode is set successfully.

```
[orangeypi@orangeypi ~]$ gpio read 5
```

1

5) Then execute the following command to set the GPIO port to pull-down mode

```
[orangeypi@orangeypi ~]$ gpio mode 5 down
```

6) Then enter the following command to read the level of the GPIO port. If the level is 0, it means that the pull-down mode is set successfully.

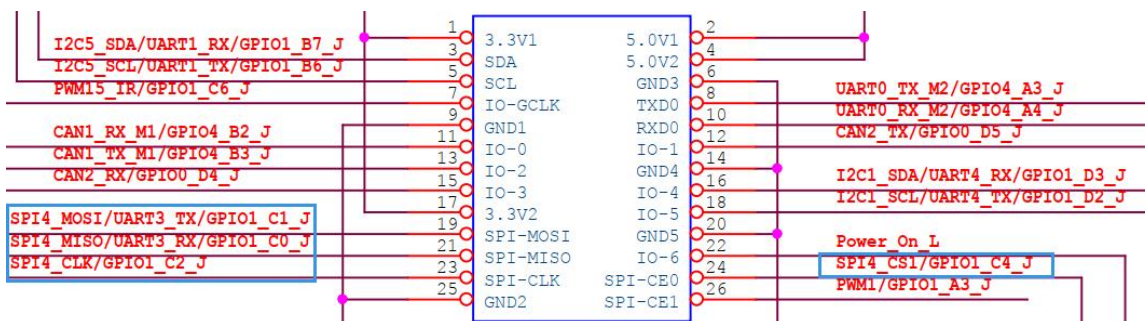
```
[orangeypi@orangeypi ~]$ gpio read 5
```

0

### 5. 6. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the spi available for Orange Pi 5B is spi4





In the OPI OS Arch system, spi4 in 26pin is turned off by default and needs to be turned on manually to use it.

Add the following configuration in red font to `/boot/extlinux/extlinux.conf`, then **restart** the OPI OS Arch system to open spi4

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

```
LABEL Orange Pi
```

```
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
```

```
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-spi4-m0-cs1-spidev.dtbo
```

2) First check whether the device node of **spidev4.1** exists in the OPI OS Arch system. If it exists, it means that SPI4 has been set up and can be used directly.

```
[orangepi@orangepi ~]$ ls /dev/spidev4.1
```

```
/dev/spidev4.1
```

3) Without shorting 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 | .....
```



4) Then short-circuit the two pins of SPI4 mosi (pin 19 in the 26pin interface) and miso (pin 21 in the 26pin interface) and then run `spidev_test`. The output is as follows. You can see the sending and receiving Data is the same



```
[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 FF 40 00 00 00 00 95 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 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D | .....@.....
```

#### 5. 6. 4. 26pin I2C test

1) As can be seen from the table below, the i2c available for Orange Pi 5B is i2c1, i2c3 and i2c5, a total of three groups of i2c buses.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		5V			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7	8	131	GND	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A3	UART0_RX_M2		
	PWM14_M1 (feb0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO4_A4	CAN2_TX_M1	I2C1_SDA_M2	
		CAN1_TX_M1	GPIO4_B3	139	13	14		GPIO4_D5			
PWM3_IR_M0 (fdb0030)	I2C1_SCL_M2	CAN2_RX_M1	GPIO0_D4	28	15	16	58	GND	UART4_RX_M0 (feb70000)	I2C1_SDA_M4	PWM1_M1 (fdb0010)
			3V		17	18	58	GPIO1_D3	UART4_TX_M0	I2C3_SCL_M4	PWM0_M1 (fdb0000)
I2C3_SCL_M0	UART3_TX_M0 (feb60000)	SPI4_MOSI_M0	GPIO1_C1	49	19	20		GPIO1_D2			
I2C3_SDA_M0	UART3_RX_M0	SPI4_MISO_M0	GPIO1_C0	48	21	22		GND			
	PWM3_IR_M2 (fdb0030)	SPI4_CLK_M0	GPIO1_C2	50	23	24	52	PowerKey			
			GND		25	26	35	GPIO1_C4	SPI4_CS1_M0		
								GPIO1_A3	PWM1_M2 (fdb0010)		

As you can see from the above table, i2c1 can be derived from pins 12 and 15 in 26pin (i2c1\_m2), or it can be derived from pins 16 and 18 in 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 OPI OS Arch system, i2c in 26pin is turned off by default and needs to be turned on manually to use it.



Add the configuration in red font below to `/boot/extlinux/extlinux.conf`, and then restart the OPI OS Arch system to open i2c1, i2c3 and i2c5 at the same time. If you only need to open one, just fill in one.

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

```
LABEL Orange Pi
```

```
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
```

```
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c1-m2.dtbo
```

```
/dtbs/rockchip/overlay/rk3588-i2c3-m0.dtbo
```

```
/dtbs/rockchip/overlay/rk3588-i2c5-m3.dtbo
```

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

2) After starting the OPI OS Arch system, first confirm that there is an i2c device node under /dev

```
[orangepi@orangepi ~]$ ls /dev/i2c-*
```

```
/dev/i2c-0 /dev/i2c-10 /dev/i2c-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	Corresponds to pin 12	Corresponds to pin 16	Corresponds to pin 21	Corresponds to pin 3
sck pin	Corresponds to pin 15	Corresponds to pin 18	Corresponds to pin 19	Corresponds to pin 5
3.3v pin	Corresponds to pin 1	Corresponds to pin 1	Corresponds to pin 1	Corresponds to pin 1
5v pin	Corresponds to pin 2	Corresponds to pin 2	Corresponds to pin 2	Corresponds to pin 2
gnd pin	Corresponds to pin 6	Corresponds to pin 6	Corresponds to pin 6	Corresponds to pin 6

**Generally, only one of the 3.3v pin and the 5v pin can be connected. Please**





**choose to connect the 3.3v pin or the 5v pin according to the specific i2c device connected.**

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 pacman -Syy i2c-tools
[orangepi@orangepi ~]$ sudo i2cdetect -y 1      #i2c1 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 3      #i2c3 commands
[orangepi@orangepi ~]$ sudo i2cdetect -y 5      #i2c5 commands
```

### 5. 6. 5. 26pin UART test

1) As can be seen from the table below, the available uarts for Orange Pi 5B are uart0, uart1, uart3 and uart4, a total of four groups of uart buses.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	3.3V		1	2		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B7	47	3	4		5V			
		PWM15_IR_M2 (feb0030)	GPIO1_B6	46	5	6		GND			
			GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1 (feb0020)	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		PowerKey			
	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 OPi OS Arch system, the uart in 26pin is turned off by default and needs to be turned on manually to use it.**

Add the configuration in red font below to **/boot/extlinux/extlinux.conf**, and then restart the OPi OS Arch system to open uart0, uart1, uart3 and uart4 at the same time. If you only need to open one, just fill in one.

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

LABEL Orange Pi

LINUX /Image

FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb

**FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo**

**/dtbs/rockchip/overlay/rk3588-uart1-m1.dtbo**

**/dtbs/rockchip/overlay/rk3588-uart3-m0.dtbo**

**/dtbs/rockchip/overlay/rk3588-uart4-m0.dtbo**

**The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.**



2) After entering the Linux system, first confirm whether there is a device node corresponding to uart under /dev 入 linux

```
[orangepi@orangepi ~]$ ls /dev/ttyS*
/dev/ttyS0 /dev/ttyS1 /dev/ttyS3 /dev/ttyS4 /dev/ttyS9
```

3) Then start testing the uart interface. First use Dupont wire to short-circuit the rx and tx of the uart interface to be tested.

	uart0	uart1	uart3	uart4
Tx pin	Corresponds to pin 8	Corresponds to pin 5	Corresponds to pin 19	Corresponds to pin 18
rx pin	Corresponds to pin 10	Corresponds to pin 3	Corresponds to pin 21	Corresponds 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 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

```

### 5. 6. 6. PWM test method

1) As can be seen from the table below, the available pwms for Orange Pi 5 include pwm0, pwm1, pwm3, pwm13, pwm14 and pwm15, a total of six pwms.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能	复用功能
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3	4		SV			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		SV			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1 (feb0020)	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		PowerKey			
	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:

pwm1 can be derived from pin 16 of 26pin (pwm1\_m1), or it can be derived from pin 26 of 26pin (pwm1\_m2)

pwm3 can be derived from pin 15 of 26pin (pwm3\_m0), or it can be derived from pin 23 of 26pin (pwm3\_m2)

Please select the corresponding pin according to your own needs. Please do not think that these are two different pwm buses.

In the OPi OS Arch system, pwm in 26pin is turned off by default and needs to be turned on manually to use it.

Add the configuration in red font below to `/boot/extlinux/extlinux.conf`, and then restart the OPi OS Arch system to open pwm0, pwm13, pwm14 and pwm15 at the same time. If you only need to open one, just fill in one.

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

```
LABEL Orange Pi
```

```
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb
```

```
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-pwm0-m1.dtbo
```

```
/dtbs/rockchip/overlay/rk3588-pwm13-m2.dtbo
```

```
/dtbs/rockchip/overlay/rk3588-pwm14-m1.dtbo
```

```
/dtbs/rockchip/overlay/rk3588-pwm15-m2.dtbo
```

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

2) When a pwm is opened, there will be an additional pwmchipX (X is a specific number) in `/sys/class/pwm/`. For example, after opening pwm15, the pwmchipX under `/sys/class/pwm/` will be Two became three

```
[orangepi@orangepi ~]$ ls /sys/class/pwm/
pwmchip0  pwmchip1  pwmchip2
```

3) Which pwmchip above corresponds to pwm15? Let's first check the output of the **ls**



**/sys/class/pwm/ -l** command, as shown below:

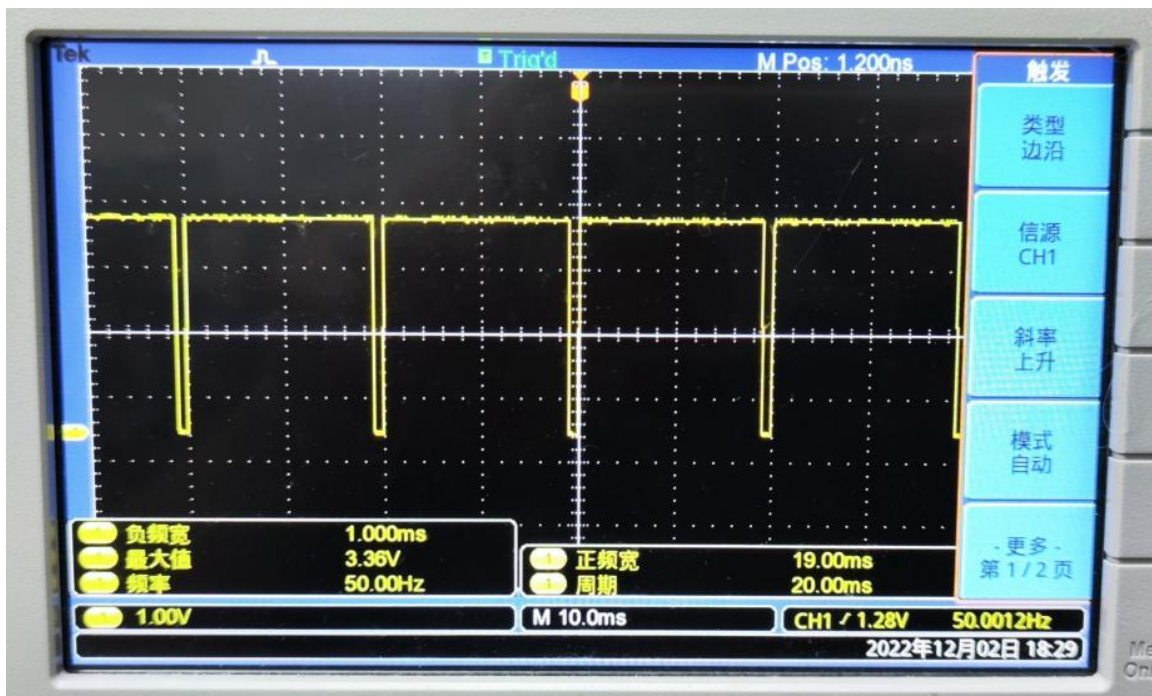
```
[orangeypi@orangeypi ~]$ ls /sys/class/pwm/ -l
total 0
lrwxrwxrwx 1 root root 0 Apr 20 07:33 pwmchip0 -> ../../devices/platform/fd8b0020.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Apr 20 07:33 pwmchip1 -> ../../devices/platform/febd0020.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Apr 20 07:33 pwmchip2 -> ../../devices/platform/feb0030.pwm/pwm/pwmchip2
[orangeypi@orangeypi ~]$
```

4) From the following table, we can see that the base address of the pwm15 register is feb0030. Looking at the output of the **ls /sys/class/pwm/ -l** command, we can see that pwmchip2 is linked to feb0030.pwm, so the corresponding pwmchip of pwm15 is 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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	3.3V		1	2		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B7	47	3	4		5V			
			GPIO1_B6	46	5	6		GND			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7	8	131	GPIO4_A3	UART0_TX_M2 (fd890000)		
			GND		9	10	132	GPIO4_A4	UART0_RX_M2		
	PWM14_M1 (feb0020)	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		PowerKey			
	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@orangeypi orangeypi]# echo 0 > /sys/class/pwm/pwmchip2/export
[root@orangeypi orangeypi]# echo 20000000 > /sys/class/pwm/pwmchip2/pwm0/period
[root@orangeypi orangeypi]# echo 1000000 > /sys/class/pwm/pwmchip2/pwm0/duty_cycle
[root@orangeypi orangeypi]# echo 1 > /sys/class/pwm/pwmchip2/pwm0/enable
```



6) The test method of pwm15 demonstrated above is similar to other pwm test methods.

## 5. 6. 7. CAN test method

1) As can be seen from the table below, the available CAN buses for Orange Pi 5B are CAN1 and CAN2, a total of two CAN 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
PWM13_M2 (feb0010)	UART1_RX_M1 (feb40000)	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V			
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		5V			
		PWM15_IR_M2 (feb0030)	GPIO1_C6	54	7	8	131	GND			
			GND		9	10	132	GPIO4_A3	UART0_TX_M2 (fd890000)		
	PWM14_M1 (feb0020)	CAN1_RX_M1	GPIO4_B2	138	11	12	29	GPIO4_A4	UART0_RX_M2		
		CAN1_TX_M1	GPIO4_B3	139	13	14		GPIO0_D5	CAN2_TX_M1	I2C1_SDA_M2	
		CAN2_RX_M1	GPIO0_D4	28	15	16	59	GND			
PWM3_IR_M0 (fd8b0030)	I2C1_SCL_M2		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		PowerKey			
	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 OPI OS Arch system, CAN in 26pin is turned off by default and needs to be turned on manually to use it.**

**Add the configuration in red font below to `/boot/extlinux/extlinux.conf`, and then restart the OPI OS Arch system to open CAN1 and CAN2 at the same time. If you only need to open one, just fill in one.**

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

LABEL Orange Pi





LINUX /Image

FDT /dtbs/rockchip/rk3588s-orangepi-5b.dtb

**FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-can1-m1.dtbo  
/dtbs/rockchip/overlay/rk3588-can2-m1.dtbo**

**The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.**

2) After entering the OPI OS Arch system, use the **sudo ifconfig -a** command. If you can see the CAN device node, it means that CAN has been opened correctly.

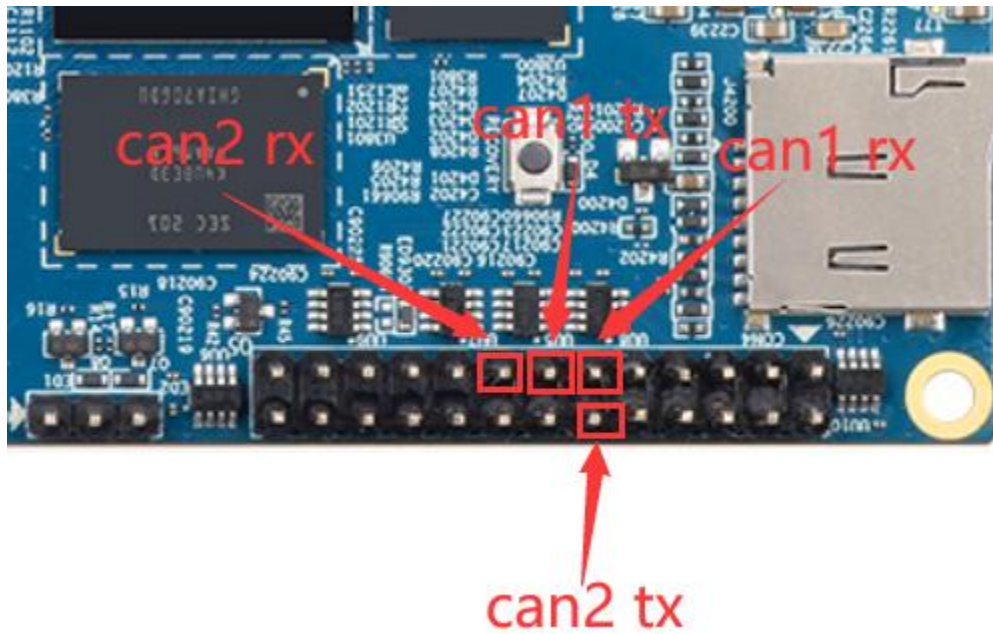
```
[orangepi@orangepi ~]$ sudo pacman -Syy net-tools
[orangepi@orangepi ~]$ sudo ifconfig -a
can0: flags=128<NOARP>  mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
    device interrupt 91

can1: flags=128<NOARP>  mtu 16
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10  (UNSPEC)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
    device interrupt 92
```

3) CAN1 和 CAN2 对应的引脚为

	CAN1	CAN2
TX pin	Corresponds to pin 13	Corresponds to pin 12
RX pin	Corresponds to pin 11	Corresponds to pin 15





4) For the method of using CANalyst-II analyzer to test CAN sending and receiving messages, please refer to the section below [using CANalyst-II analyzer to test sending and receiving](#) messages.

## 6. Linux SDK——Orange Pi Build Instruction

### 6. 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 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.**

#### 6. 1. 1. Use the development board Ubuntu 22.04 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.**

#### 6. 1. 2. Use X64's Ubuntu 22.04 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. orangepi-build.

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 22.04



## Ubuntu 镜像使用帮助

Ubuntu 的软件源配置文件是 `/etc/apt/sources.list`。将系统自带的该文件做个备份，将该文件替换为下面内容，即可使用 TUNA 的软件源镜像。

选择你的ubuntu版本: 22.04 LTS

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

c. The 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.



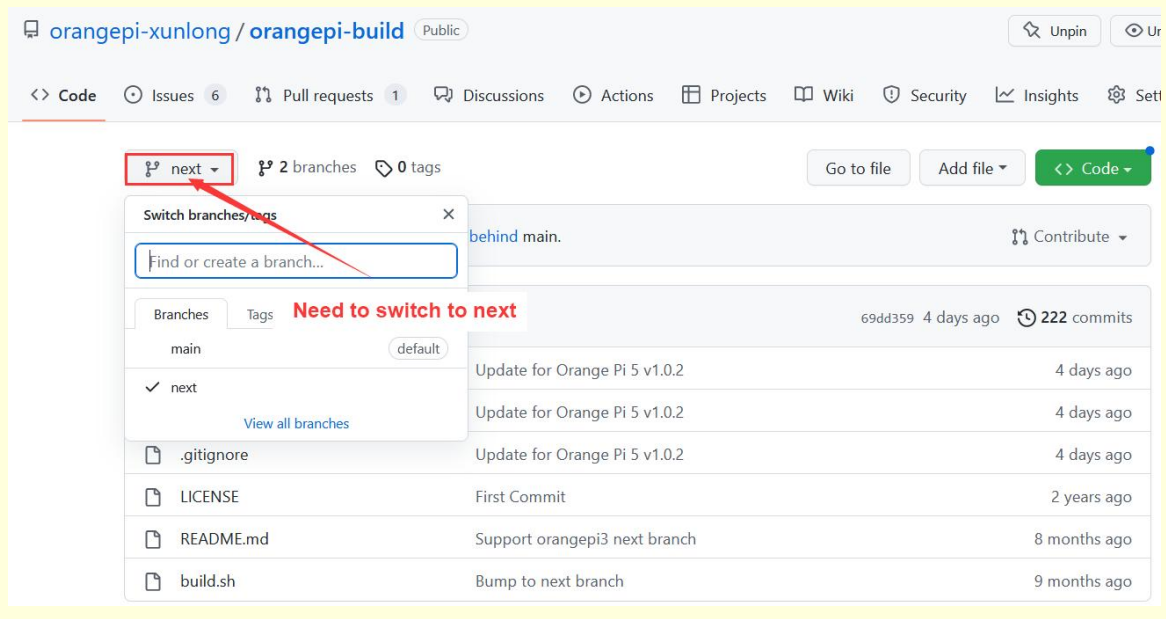
## 6. 2. Get the source code of linux sdk

### 6. 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
current	u-boot 2017.09	Linux6.1

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.

We currently define the linux5.10 bsp kernel provided by RK as the legacy branch, and the linux6.1 bsp kernel as the current branch.

3) orangepi-build will contain the following files and folders after download

- 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
- 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.

### 6. 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 Ubuntu 22.04 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 toolchain 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.

```
[ o.k. ] Checking for external GCC compilers
[ .... ] downloading using http(s) network [ gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz ]
#8d7029 16MiB/24MiB(65%) CN:1 DL:7.9MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz: 24.9MiB [14.4MiB/s] [=====] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz ]
#e30eec 17MiB/33MiB(50%) CN:1 DL:10MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz: 33.9MiB [9.66MiB/s] [=====] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-linux-gnueabi-4.8-2014.04_linux.tar.xz ]
#841c24 48MiB/48MiB(99%) CN:1 DL:2.7MiB]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-linux-gnueabi-4.8-2014.04_linux.tar.xz: 48.8MiB [13.0MiB/s] [=====] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz ]
#3dee3e 72MiB/76MiB(93%) CN:1 DL:3.7MiB ETA:1s]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz: 77.0MiB [14.2MiB/s] [=====] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz ]
#42e728 104MiB/104MiB(99%) CN:1 DL:2.8MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz: 104MiB [13.9MiB/s] [=====] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz ]
#2c065e 108MiB/111MiB(97%) CN:1 DL:3.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz: 111MiB [13.4MiB/s] [=====] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabi.tar.xz ]
#d232ee 256MiB/251MiB(99%) CN:1 DL:2.0MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabi.tar.xz: 251MiB [13.7MiB/s] [=====] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu.tar.xz ]
#88b441 268MiB/269MiB(99%) CN:1 DL:0.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
```

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/](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-11.2-2022.02-x86_64-aarch64-none-linux-gnu
gcc-arm-11.2-2022.02-x86_64-arm-none-linux-gnueabi
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabi
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabi
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
- linux5.10

```
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
```

- 5) The cross compilation tool chain used by compiling the u-boot source code is
- v2017.09

```
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
```

### 6. 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

- The git repository where the Linux5.10 kernel source code is stored is as follows:

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk3588
```

- The git warehouse where the Linux6.1 kernel source code is stored is as follows:

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-6.1-rk35xx
```

- 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
```

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.

- build.sh**: Compile the startup script
- external**: Contains the configuration files needed to compile the image, scripts for specific functions, and the source code of some programs. The rootfs compressed package cached during the image compilation process is also stored in external.
- kernel**: Stores the source code of the Linux kernel. The folder named **orange-pi-5.10-rk35xx** stores the kernel source code of the legacy branch of the RK3588/RK3588S series development board. The folder named **orange-pi-6.1-rk35xx** stores the RK3588 /Kernel source code of the current



branch of the RK3588S series development board. Please do not modify the name of the folder of the kernel source code manually. If modified, the kernel source code will be re-downloaded when the compilation system is running.

- d. **LICENSE**: GPL 2 License
- e. **README.md**: orangepi-build description file
- f. **output**: 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  
**u-boot**: Store the u-boot source code. The folder named **v2017.09-rk3588** stores the u-boot source code of the legacy and current branches of the RK3588/RK3588S series development board. Please do not modify the name of the u-boot source code folder manually. , if modified, the u-boot source code will be re-downloaded when the compilation system is running.
- i. **userpatches**: Store the configuration file needed to be used in the compilation script

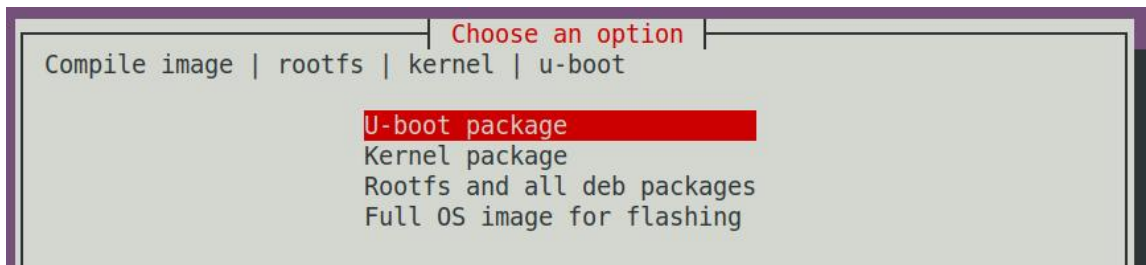
```
test@test:~/orangepi-build$ ls
build.sh  external  kernel  LICENSE  output  README.md  scripts
toolchains  u-boot  userpatches
```

### 6. 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

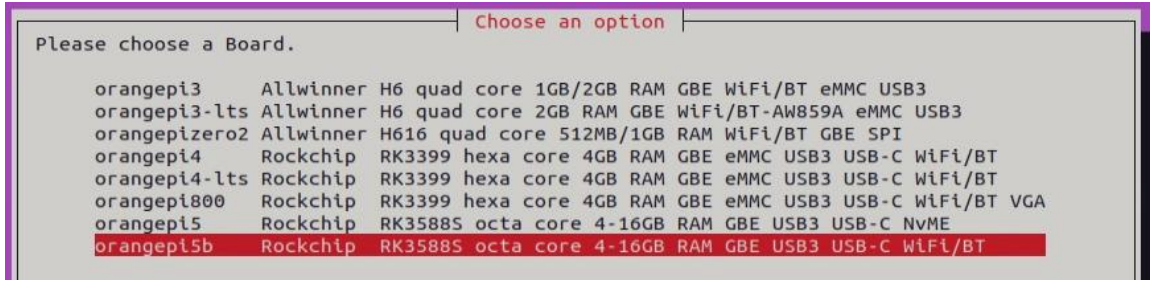


```

Choose an option
Compile image | rootfs | kernel | u-boot
U-boot package
Kernel package
Rootfs and all deb packages
Full OS image for flashing

```

- 3) Then select the model of the development board



4) Then I will start to compile u-boot, and the part of the information prompted during compilation will be as follow

a. u-boot Source code version

```
[ o.k. ] Compiling u-boot [ v2017.09 ]
```

b. The version of the cross compile toolchain

```
[ 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-orangePi5b_1.0.0_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=orangePi5b 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-orangePi5b_1.0.0_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-orangePi5b_1.0.0_arm64.deb . (Note that there is a "." at the  
end of the command)
```



```
test@test:~/orangepi_build/output/debs/u-boot$ ls
linux-u-boot-legacy-orangepi5b_1.0.0_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-orangepi5b_1.0.0_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-build 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:~/orange_pi_build/output/debs/u-boot$ scp \
```

```
linux-u-boot-legacy-orangepi5b_1.0.0_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-orangepi5b-legacy
```

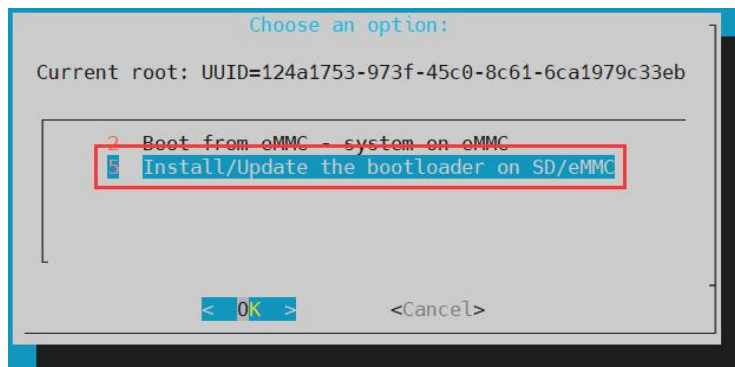
- c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepi5b_1.0.0_arm64.deb
```

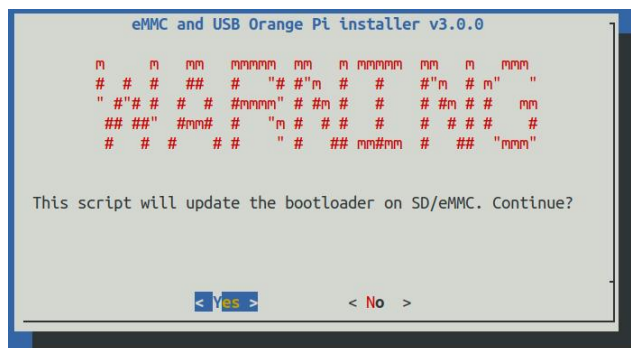
- d. Then run nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

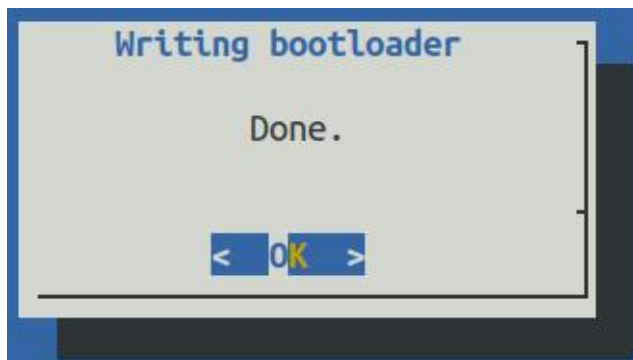
- e. Then select **5 5 Install/Update the bootloader on SD/eMMC** to update the U-Boot (If it is a linux system started by eMMC, the update is u-boot in eMMC)



- f. After pressing the Enter key, a Warning will pop up first



- g. Press the cabbage 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

[orange-pi-build/u-boot/v2017.09-rk3588/configs/orangepi\\_5b\\_defconfig](#)

- b. u-boot 2017.09 Source code, the development board uses a dts file.

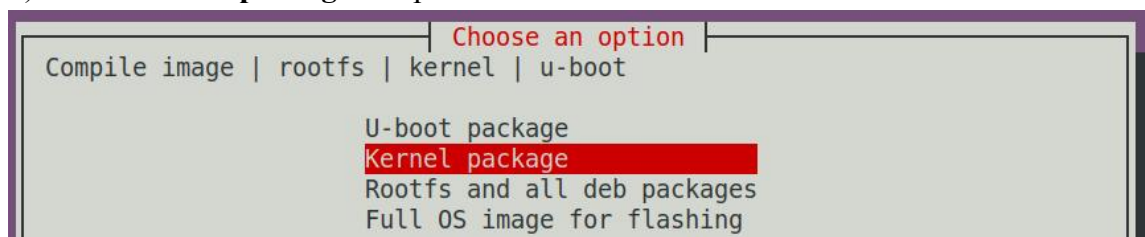
[orange-pi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3588s-orangepi-5b.dts](#)

## 6. 4. Compile the linux kernel

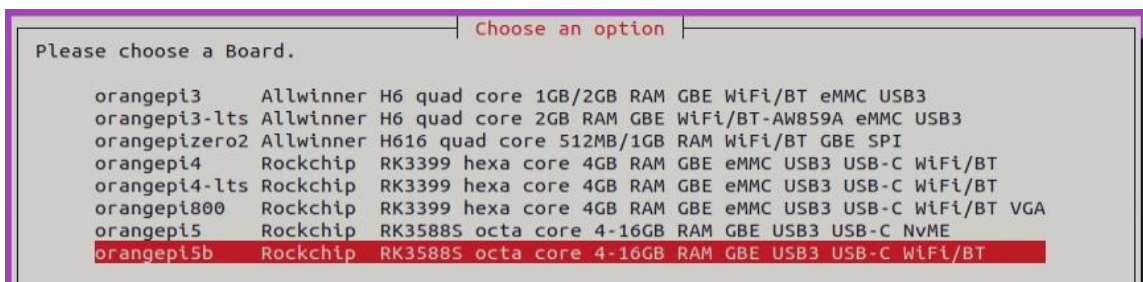
- 1) Run the build.sh script, remember to add Sudo permission

```
test@test:~/orange-pi-build$ sudo ./build.sh
```

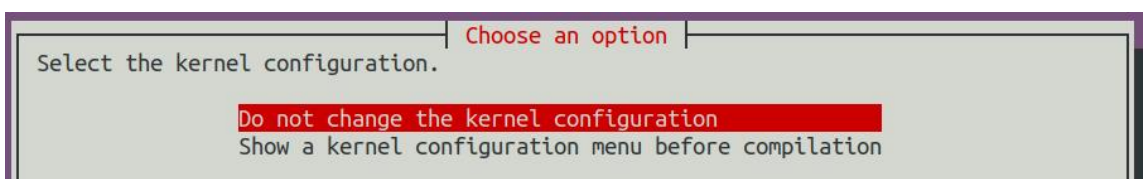
- 2) Select **Kernel package** and press Enter



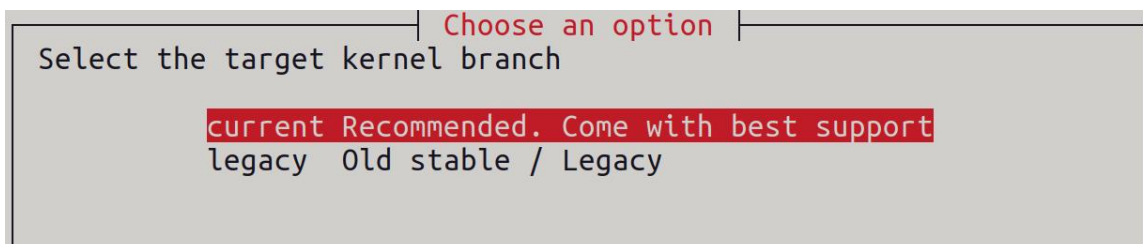
- 3) Then select the model of the development board



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

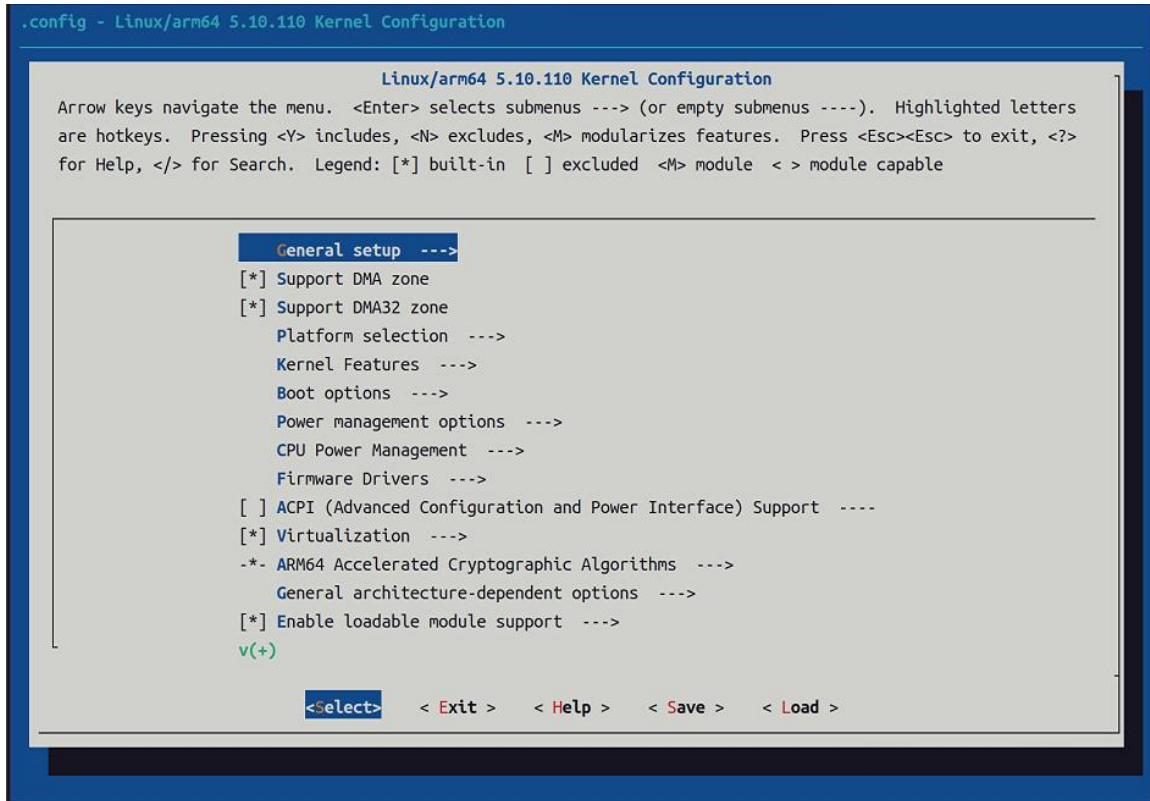


- 5) Then select the branch type of the kernel source code
- The a.legacy branch will compile the linux5.10 kernel source code
  - The b.current branch will compile the linux6.1 kernel source code



6) 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:~/orange-pi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

- b. You can also set **orange-pi-build/user patches/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 rerun the build.sh script.



```

HOSTCC scripts/kconfig/mconf.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/util.o
HOSTCC scripts/kconfig/lxdialog/inputbox.o
HOSTCC scripts/kconfig/lxdialog/textbox.o
HOSTCC scripts/kconfig/lxdialog/yesno.o
HOSTCC scripts/kconfig/lxdialog/menubox.o
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated

```

## 7) Part of the information prompted when compiling the kernel source code

### a. Linux kernel source version

[ o.k. ] Compiling current kernel [ **5.10.160** ]

### b. The version of the cross compilation tool chain used

[ o.k. ] Compiler version [ **aarch64-none-linux-gnu-gcc 11.2.1** ]

### c. The configuration file used by the 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 [ **orange-pi-build/output/debs/** ]

### e. Compiled and generated kernel images deb bag name

[ o.k. ] File name [ **linux-image-legacy-rockchip-rk3588\_1.0.0\_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=orange-pi5b BRANCH=legacy BUILD\_OPT=kernel KERNEL\_CONFIGURE=no** ]

## 8) View compiled core -related deb package

- linux-dtb-legacy-rockchip-rk3588\_1.0.0\_arm64.deb** containing kernels dtb file
- linux-headers-legacy-rockchip-rk3588\_1.0.0\_arm64.deb** Contains kernel header files
- linux-image-legacy-rockchip-rk3588\_1.0.0\_arm64.deb** Contains kernel images and kernel modules



```
test@test:~/orange-pi-build$ ls output/debs/linux-*
output/debs/linux-dtb-legacy-rockchip-rk3588_1.0.0_arm64.deb
output/debs/linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb
output/debs/linux-headers-legacy-rockchip-rk3588_1.0.0_arm64.deb
```

9) 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:~/orange-pi-build$ cd output/debs
test@test:~/orange-pi-build/output/debs$ mkdir test
test@test:~/orange-pi-build/output/debs$ cp \
linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb test/
test@test:~/orange-pi-build/output/debs$ cd test
test@test:~/orange-pi-build/output/debs/test$ dpkg -x \
linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb .
test@test:~/orange-pi-build/output/debs/test$ ls
boot  etc  lib  linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb  usr
```

- b. The decompressive file is shown below

```
test@test:~/orange-pi-build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.160-rockchip-rk3588
│   ├── System.map-5.10.160-rockchip-rk3588
│   └── vmlinuz-5.10.160-rockchip-rk3588
├── etc
│   └── kernel
├── lib
│   └── modules
├── linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb
├── usr
│   ├── lib
│   └── share
```

10) orange-pi-build 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, 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"
```

11) 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_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

- b. Then log in to the development board, install 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_1.0.0_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 infor

- a. The storage location of the kernel configuration file is as follows. Please do not go to the kernel source code to find the kernel configuration file used by the development board.

- a) Linux5.10

```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-legacy.config
```

- b) Linux6.1

```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-current.config
```

- b. The location of the dts file used by the development board

- a) Linux5.10



```
orange-pi-build/kernel/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s-
orange-pi-5.dts
```

b) Linux6.1

```
orange-pi-build/kernel/orange-pi-6.1-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s-
orange-pi-5.dts
```

## 6. 5. Compile rootfs

1) Run the build.sh script, remember to add Sudo permissions

```
test@test:~/orange-pi-build$ sudo ./build.sh
```

2) Select **Rootfs and all deb packages**, and then press Enter

```

Choose an option
Compile image | rootfs | kernel | u-boot

U-boot package
Kernel package
Rootfs and all deb packages
Full OS image for flashing

```

3) Then select the model of the development board

```

Choose an option
Please choose a Board.

orange-pi3      Allwinner H6 quad core 1GB/2GB RAM GBE WiFi/BT eMMC USB3
orange-pi3-lts Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
orange-pi-zero2 Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
orange-pi4      Rockchip RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orange-pi4-lts  Rockchip RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orange-pi800    Rockchip RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
orange-pi5      Rockchip RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVME
orange-pi5b     Rockchip RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT

```

4) Then select the branch type of the kernel source code. Currently, the rootfs maintained by the kernel source code uses the same set.

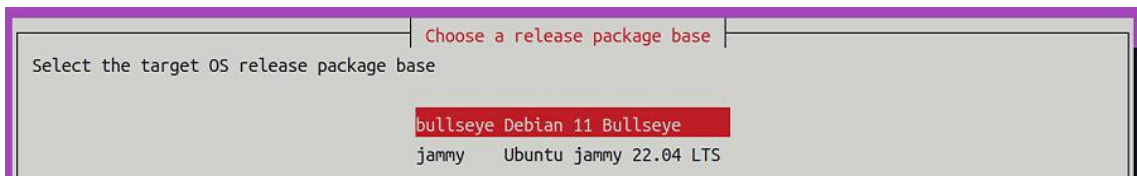
```

Choose an option
Select the target kernel branch

current Recommended. Come with best support
legacy Old stable / Legacy

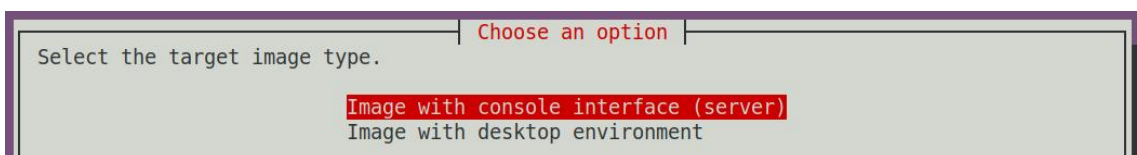
```

5) Then select the type of rootfs

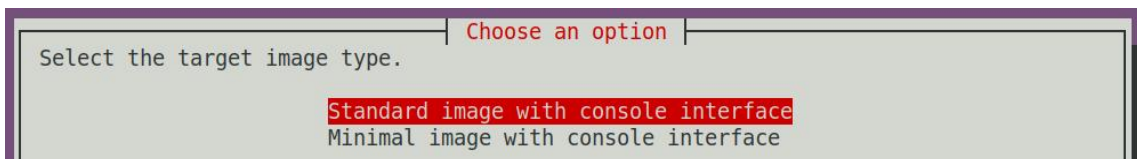


6) Then select the type of the image

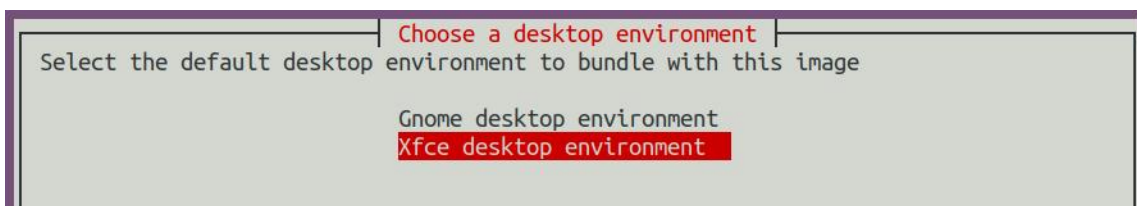
- a. **Image with console interface (server)** Indicates the image of the server version, the volume is small
- b. **Image with desktop environment** Indicates an image with a desktop, which is relatively large



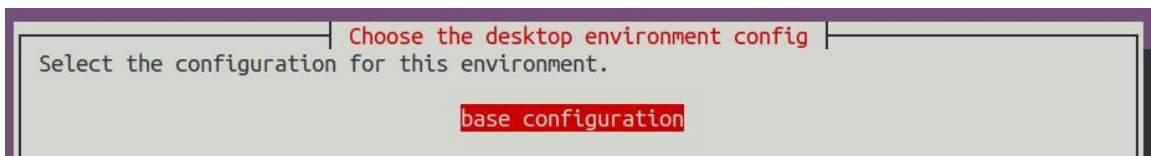
7) If it is a image that compiles the server version, you can also choose to compile the standard version or the minimal version. The pre -installed software pre -installed software will be much less than the Standard version. (**No special needs, please do not choose the minimal version, because many things are not pre -installed by default, some functions may not be used**)



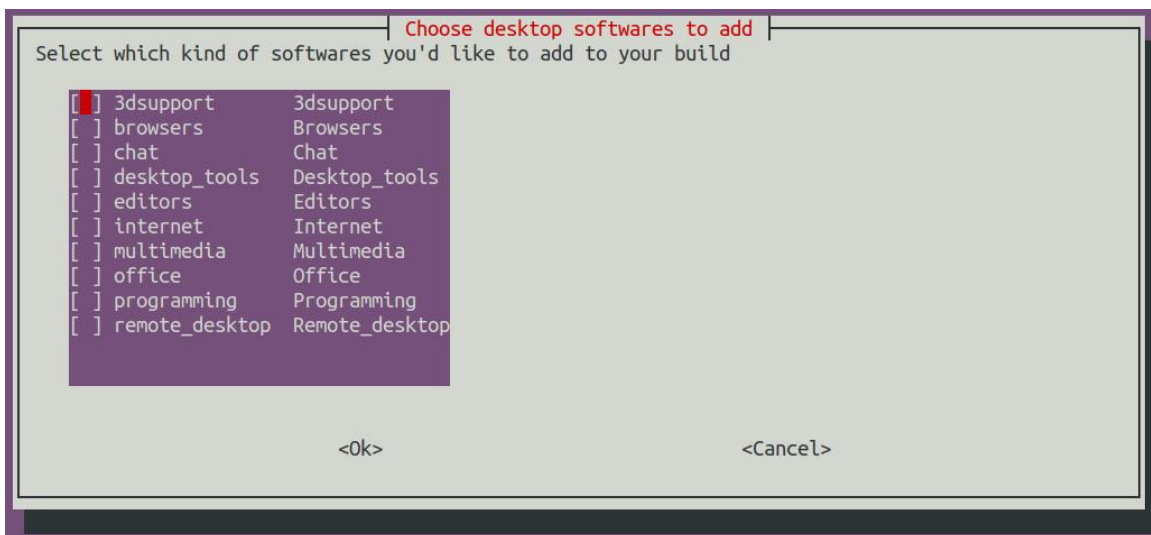
8) If you are compiling a desktop version of the image, you need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop







Then you can choose an additional software package you need to install. Press the Enter key here directly.



9) 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 compile rootfs compressed package

```
[ o.k. ] File name [ jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4 ]
```

d. Time to compile

```
[ o.k. ] Runtime [ 13 min ]
```

10) View compiled rootfs compression package

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**. It is a compressed package for rootfs. The meaning of each field of the name

a) **jammy** Represents the type of rootfs Linux release version

b) **xfce** Indicates that rootfs is the type of desktop version, if it is **cli**, it means





- c) the server version type
- d) **arm64** Indicates the architecture of rootfs
- e) **f930ff6ebbac1a72108a2e100762b18f** It is the MD5 hash value generated by the package names of all software packages installed by rootfs. As long as the list of software packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to determine whether it needs
- 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
```

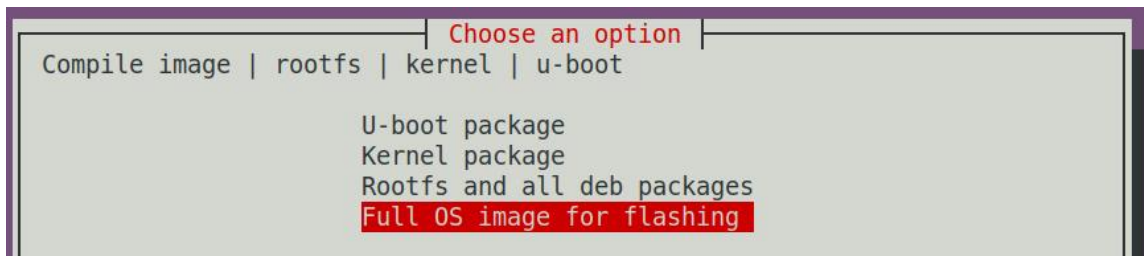
11) If the required rootfs already exists under **external/cache/rootfs**, then compile rootfs again will skip the compilation process directly, and will not start compiling again. When compiling the image, you will also go to the **external/cache/rootfs** to find out if there is a cache rootfs. If you have it, you can use it directly, so that you can save a lot of download compilation time.

## 6. 6. Compile linux images

- 1) Run the build.sh script, remember to add Sudo permission

```
test@test:~/orangepi-build$ sudo ./build.sh
```

- 2) Select **Full OS image for flashing**, and then press Enter



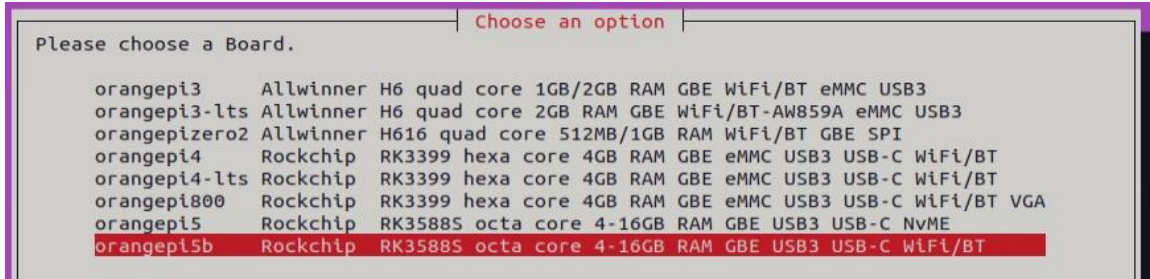
```

Choose an option
Compile image | rootfs | kernel | u-boot

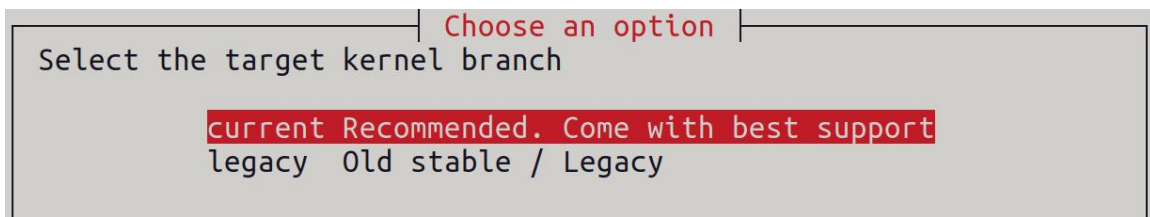
U-boot package
Kernel package
Rootfs and all deb packages
Full OS image for flashing

```

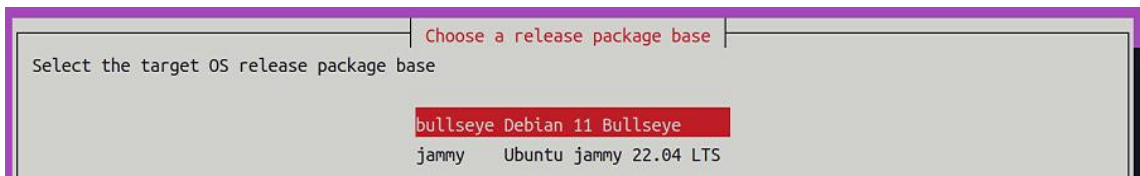
- 3) Then select the model of the development board



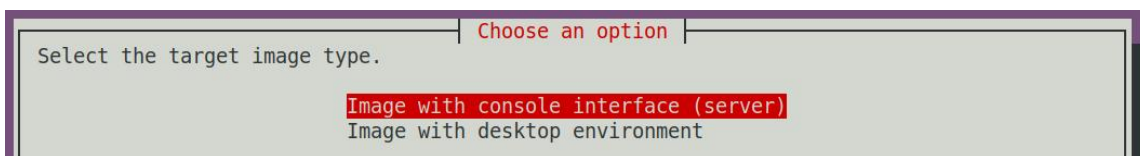
- 4) Then select the branch of the kernel source code
  - a. The legacy branch will compile the linux5.10 kernel source code
  - b. The b.current branch will compile the linux6.1 kernel source code



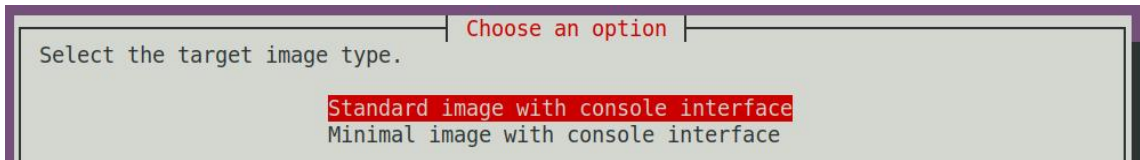
- 5) Then select the type of rootfs



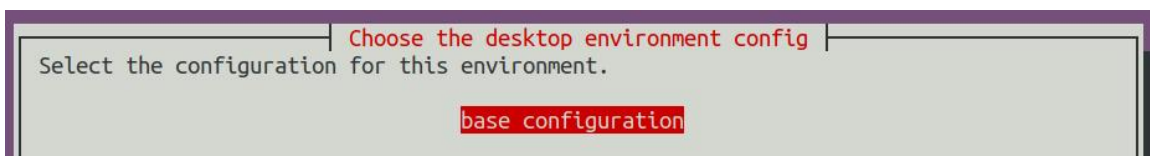
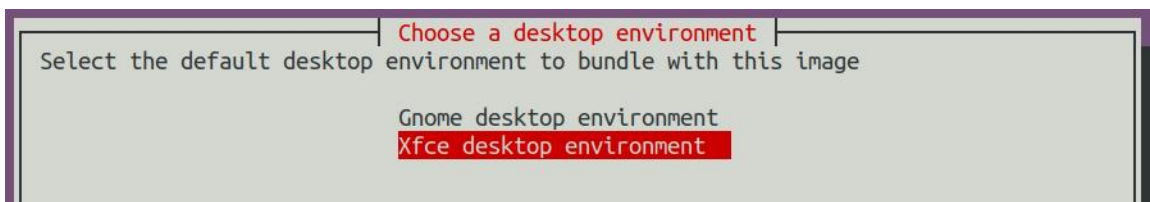
- 6) Then select the type of the image
  - a. **Image with console interface (server)** Indicates the image of the server version, the volume is small
  - b. **Image with desktop environment** Indicates an image with a desktop, which is relatively large



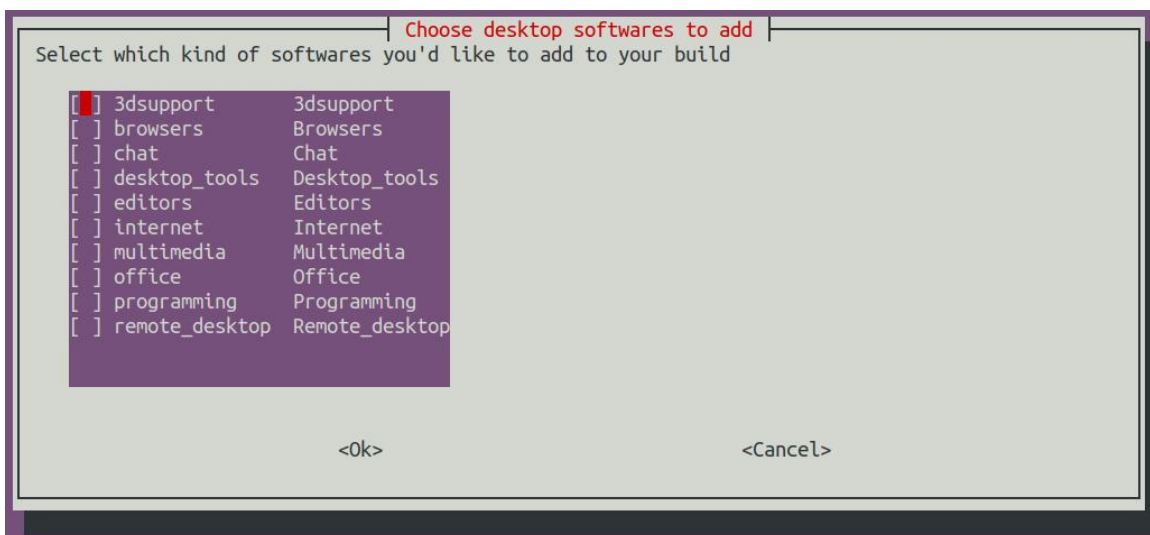
- 7) If it is a image that compiles the server version, you can also choose to compile the standard version or the minimal version. The pre -installed software pre -installed software will be much less than the Standard version. (No special needs, please do not choose the minimal version, because many things are not pre -installed by default, some functions may not be used)



8) If you are compiling a desktop version of the image, you need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop



Then you can choose an additional software package you need to install. Press the Enter key here directly.



9) 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 将 u-boot

10) The following information will be prompted after compiling the image

- a. The storage path of the compiled

```
[ o.k. ] Done building
[ output/images/orangepi5b_1.0.0_debian_bullseye_linux5.10.160_xfce_desktop/orangepi5b_1.0.0_debian_bullseye_linux5.10.160_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=orangepi5b
BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no
```



```
BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

## 7. Linux Development Manual

### 7.1. How to compile the kernel source code separately in the linux system of the development board

- 1) First download the Linux kernel source code of the development board
  - a. If you are using a Linux 5.10 kernel system, you need to download the orange-pi-5.10-rk35xx branch.

```
orangepi@orangepi:~$ git clone --depth=1 -b orange-pi-5.10-rk3588 https://github.com/orangepi-xunlong/linux-orangepi
```

- b. If you are using a Linux 6.1 kernel system, you need to download the orange-pi-6.1-rk35xx branch.

```
orangepi@orangepi:~$ git clone --depth=1 -b orange-pi-6.1-rk35xx https://github.com/orangepi-xunlong/linux-orangepi
```

**If you have problems downloading the code from github, you can go to the official information of the development board to download the compressed kernel source code package, then upload it to the Linux system of the development board, and then decompress it.**



**The command to decompress the kernel source code archive is:**

**a. Linux5.10 kernel**

```
orangepi@orangepi:~$ tar xzf orange-pi-5.10-rk35xx.tar.gz
```

```
orangepi@orangepi:~$ mv orange-pi-5.10-rk35xx linux-orangepi
```

**b. Linux6.1 kernel**

```
orangepi@orangepi:~$ tar xzf orange-pi-6.1-rk35xx.tar.gz
```

```
orangepi@orangepi:~$ mv orange-pi-6.1-rk35xx linux-orangepi
```

**After decompression, please execute the following command to synchronize the source code with github to ensure that the source code is in the latest state:**

```
orangepi@orangepi:~$ cd linux-orangepi
```

```
orangepi@orangepi:~/linux-orangepi$ git pull
```

**2) Then configure the default kernel configuration**

```
orangepi@orangepi:~$ cd linux-orangepi
```

```
orangepi@orangepi:~/linux-orangepi$ make rockchip_linux_defconfig
```

**rockchip\_linux\_defconfig** The path in the kernel source code is arch/arm64/configs/

**3) Then compile the kernel source code**

```
orangepi@orangepi:~/linux-orangepi$ make -j10
```

**4) Then install the kernel module**

```
orangepi@orangepi:~/linux-orangepi$ sudo make modules_install
```

**The installation path of the kernel module is: /lib/modules**

**After executing the sudo make modules\_install command, you can see that**



**there will be an additional kernel module folder under /lib/modules/:**

```
orangepi@orangepi5b:~$ ls /lib/modules
```

# If it is a system with Linux 5.10 kernel, the output is as follows

**5.10.160+** 5.10.160-rockchip-rk3588

# If it is a system with Linux 6.1 kernel, the output is as follows

**6.1.43+** 6.1.43-rockchip-rk3588

5) Then install the kernel image and uInitrd

```
orangepi@orangepi:~/linux-orangepi$ sudo make install
```

**The installation path of the kernel image and uInitrd is: /boot/**

**After executing the sudo make install command, you can see that there will be one more kernel file under /boot/:**

```
orangepi@orangepi5b:~/linux-orangepi$ ls /boot/vmlinuz*
```

# If it is a system with Linux 5.10 kernel, the output is as follows

**/boot/vmlinuz-5.10.160+** /boot/vmlinuz-5.10.160-rockchip-rk3588

# If it is a system with Linux 6.1 kernel, the output is as follows

**/boot/vmlinuz-6.1.43+** /boot/vmlinuz-6.1.43-rockchip-rk3588

**The file /boot/Image is actually loaded when the system starts, and Image is a copy of the vmlinuz file.**

6) Then install the dtb file into **/boot/dtb**

```
orangepi@orangepi:~/linux-orangepi$ sudo make dtbs_install INSTALL_DTBS_PATH=/boot/dtb/
```

7) Then restart the Linux system and the newly compiled kernel will be loaded

```
orangepi@orangepi:~$ uname -r
```

# If it is a Linux 5.10 system, the output is as follows

**5.10.160+**

# If it is a Linux 6.1 system, the output is as follows

**6.1.43+**





## 8. How to use the Android 12 system.

### 8. 1. The Android versions supported

Android Version	Kernel Version
Android 12	Linux5.10
Android 12 Box	Linux5.10

### 8. 2. Adaptation of Android functions

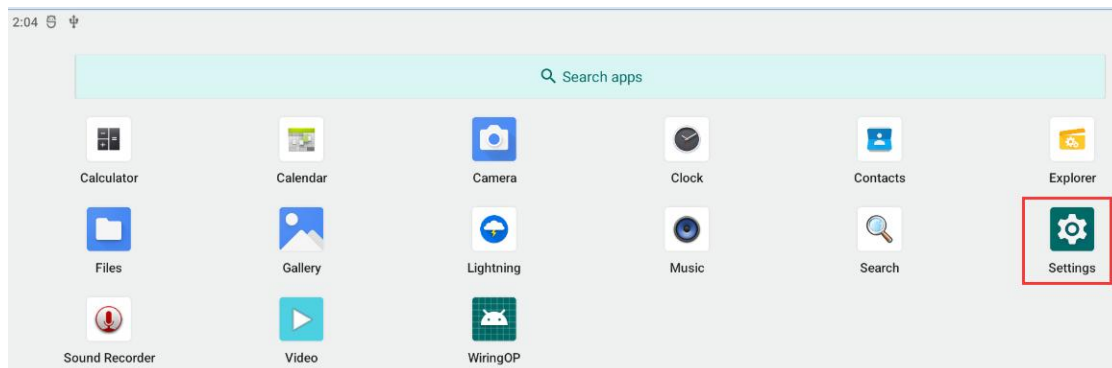
Functions	Android 12	Android12 Box
USB2.0x2	OK	OK
USB3.0x1	OK	OK
USB Type-C 3.0	OK	OK
DP Display	OK	OK
eMMC Start	OK	OK
AP6275P-WIFI	OK	OK
AP6275P-Bluetooth	OK	OK
GPIO (26pin)	OK	OK
UART (26pin)	OK	OK
SPI (26pin)	OK	OK
I2C (26pin)	OK	OK
PWM (26pin)	OK	OK
3pin Debug Serial Port	OK	OK
TF-Card Start	OK	OK
HDMI Video	OK	OK
HDMI Audio	OK	OK
OV13850 Camera	OK	OK



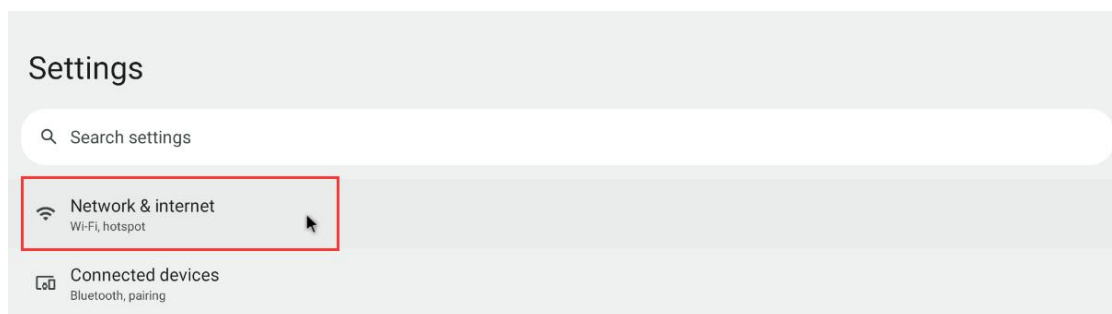
<b>OV13855 Carema</b>	<b>OK</b>	<b>OK</b>
<b>LCD1</b>	<b>OK</b>	<b>NO</b>
<b>LCD2</b>	<b>OK</b>	<b>NO</b>
<b>Gigabit Ethernet Port</b>	<b>OK</b>	<b>OK</b>
<b>Network Port Status Light</b>	<b>OK</b>	<b>OK</b>
<b>MIC</b>	<b>OK</b>	<b>OK</b>
<b>Headphone Playback</b>	<b>OK</b>	<b>OK</b>
<b>Headphone Recording</b>	<b>OK</b>	<b>OK</b>
<b>LED Light</b>	<b>OK</b>	<b>OK</b>
<b>GPU</b>	<b>OK</b>	<b>OK</b>
<b>NPU</b>	<b>OK</b>	<b>OK</b>
<b>VPU</b>	<b>OK</b>	<b>OK</b>
<b>Switch Button</b>	<b>OK</b>	<b>OK</b>
<b>HDMI CEC Function</b>	<b>NO</b>	<b>OK</b>

### 8.3. WIFI connection test method

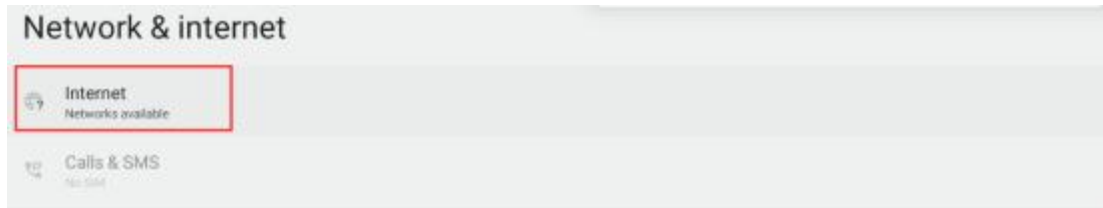
1) First click to enter the **Setting**



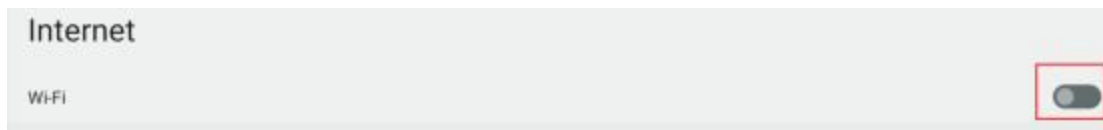
2) Then select **Network & internet**



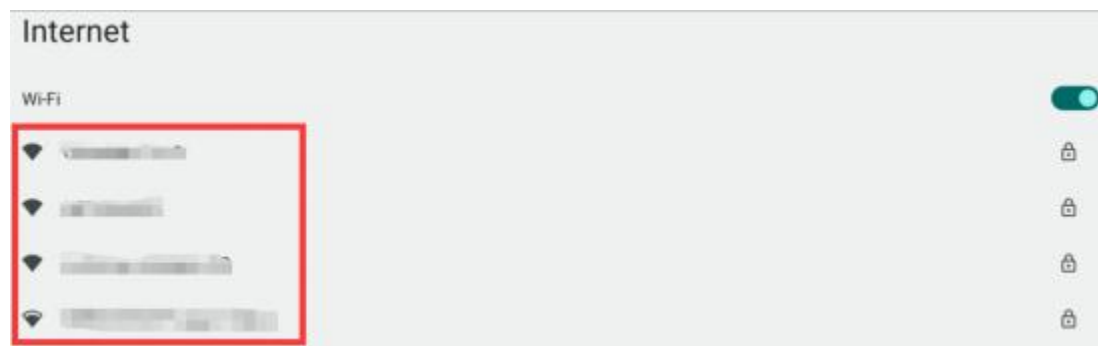
### 3) Select **Internet**



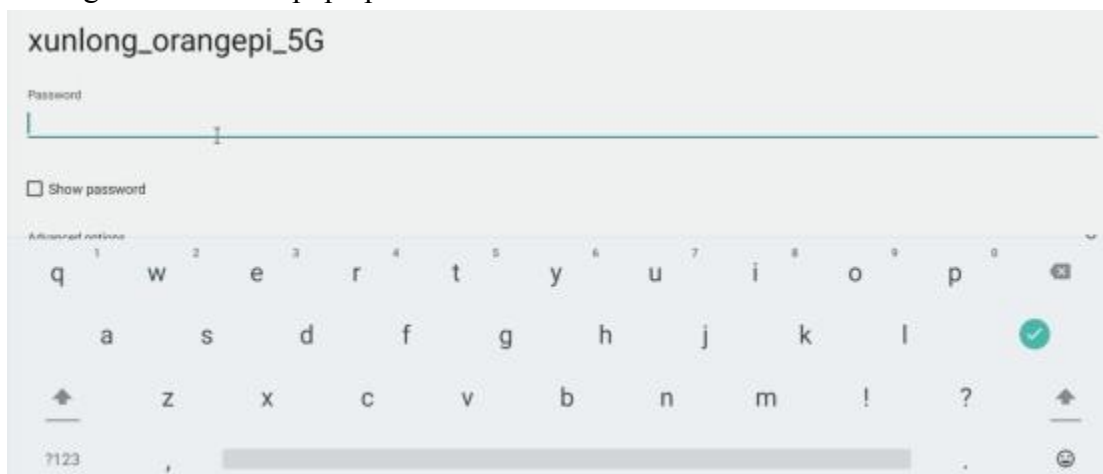
### 4) Switch **Wi-Fi**



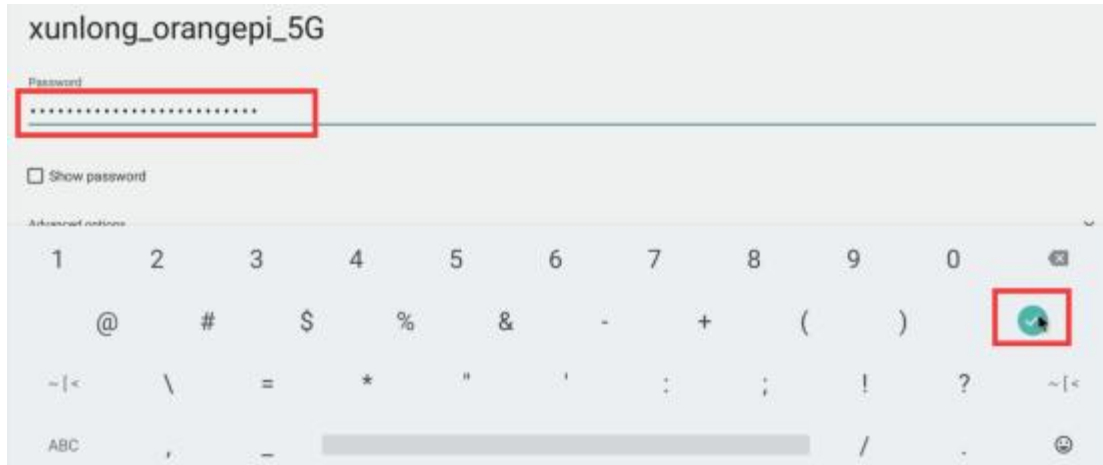
5) After turning on **Wi-Fi**, if everything is normal, you can scan for nearby Wi-Fi hotspots.



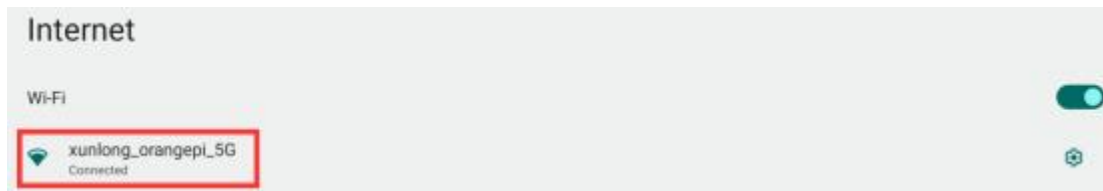
6) Then select the Wi-Fi you want to connect to, and the password input interface shown in the figure below will pop up



- 7) Then use the keyboard to enter the password corresponding to Wi-Fi, and then use the mouse to click the Enter button in the virtual keyboard to start connecting to Wi-Fi



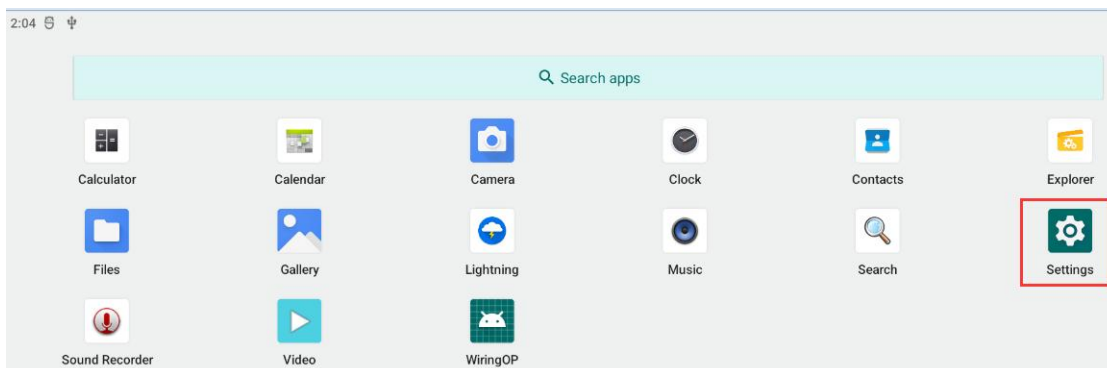
- 8) The display after successful Wi-Fi connection is shown in the figure below:



## 8.4. How to use Wi-Fi hotspot

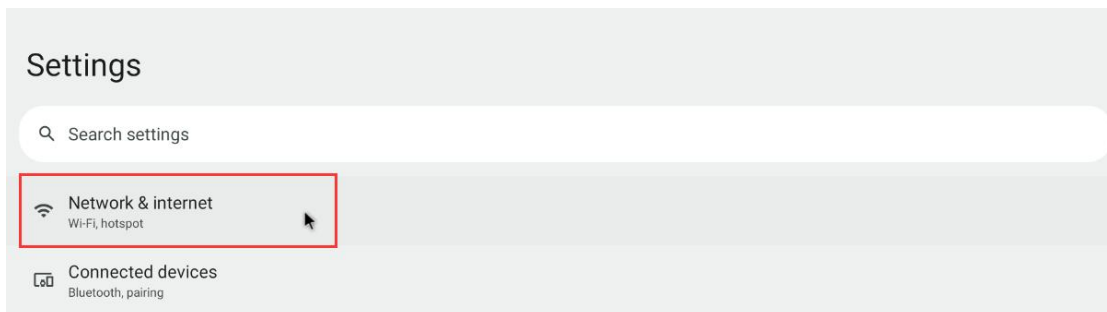
- 1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally.

- 2) And then select **Settings**

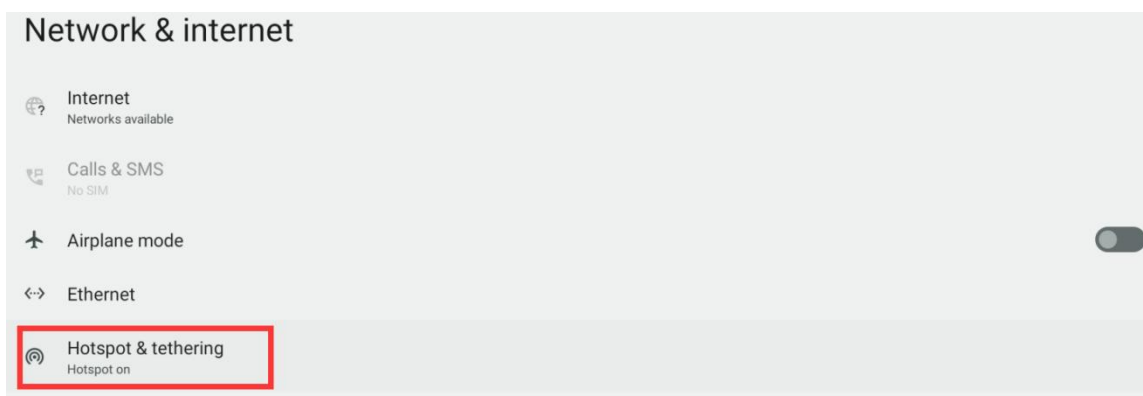




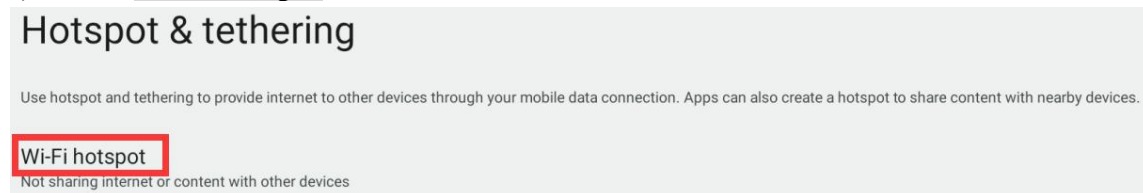
### 3) Select **Network & internet**



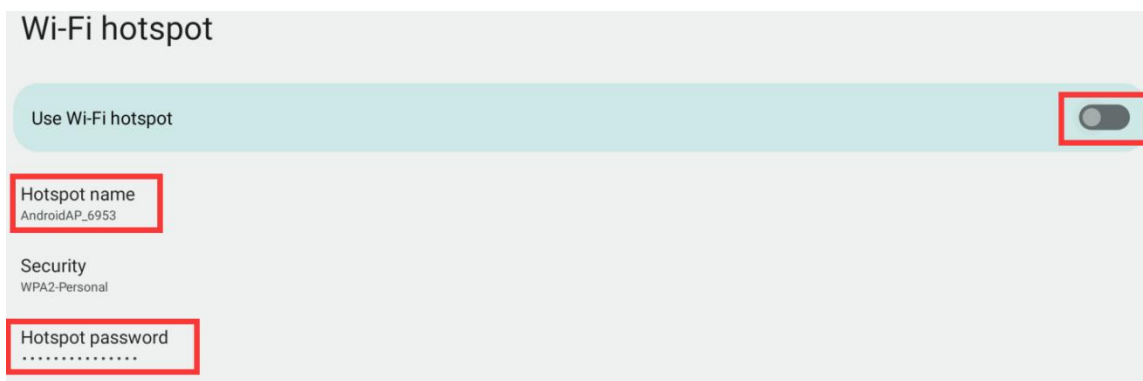
### 4) Select **Hotspot & tethering**



### 5) Select **Wi-Fi hotspot**



6) Then turn on the **Wi-Fi hotspot**, you can also see the name and password of the generated hotspot in the figure below, remember them and use them when connecting to the hotspot (if you need to modify the name and password , you need to turn off the **Wi-Fi hotspot** first before modification)



7) At this point, you can take out your mobile phone. If everything is normal, you can find the **Hotspot name** above in the WIFI list searched by the mobile phone (**Name is AndroidAP\_6953**). Then you can click **AndroidAP 6953** to connect to the hotspot, and the password can be seen under the **Hotspot password** in the above picture

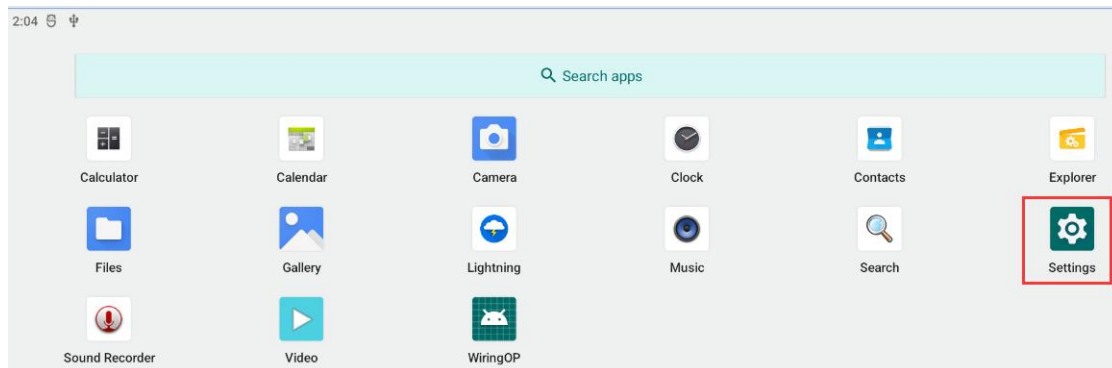


8) After the connection is successful, it will be displayed as shown in the figure below (the interface of different mobile phones will be different, the specific interface is subject to the display of your mobile phone). At this point, you can open a webpage on your mobile phone to see if you can access the Internet. If you can open the webpage normally, it means that the **WI-FI Hotspot** of the development board can be used normally.

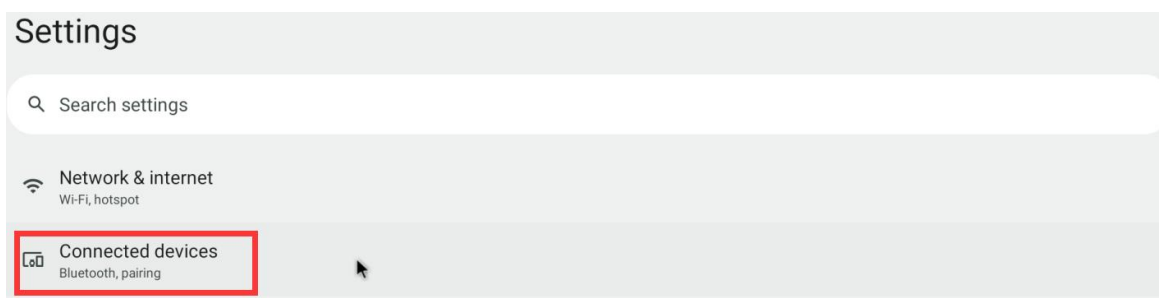


## 8.5. How to test Bluetooth

1) First enter **Setting**

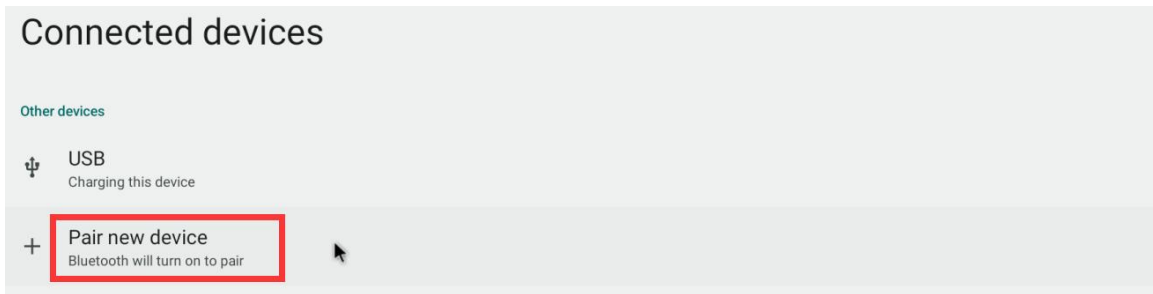


2) And then select **Connected devices**

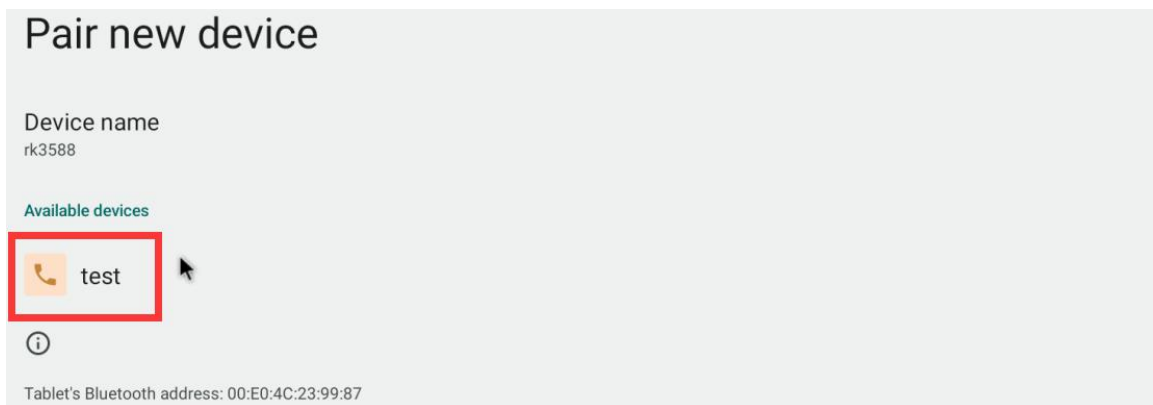


3) Then click **Pair new device** to turn on Bluetooth and start scanning the surrounding Bluetooth devices.

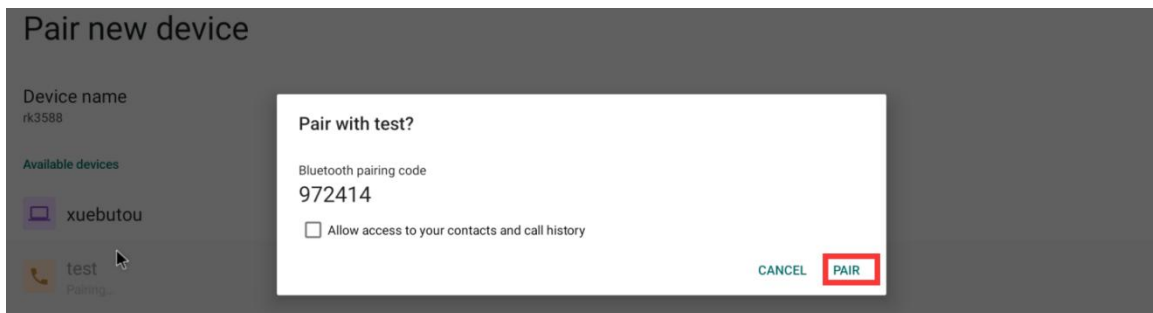




- 4) The search Bluetooth devices will be displayed under **Available devices**



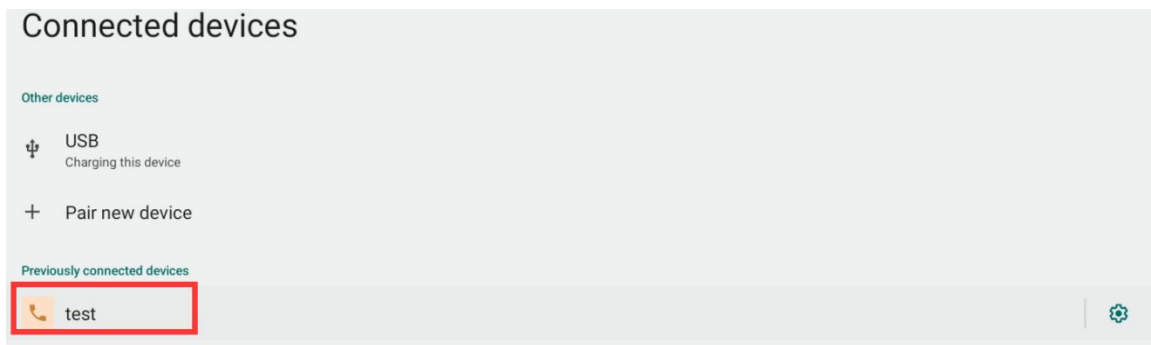
- 5) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option.



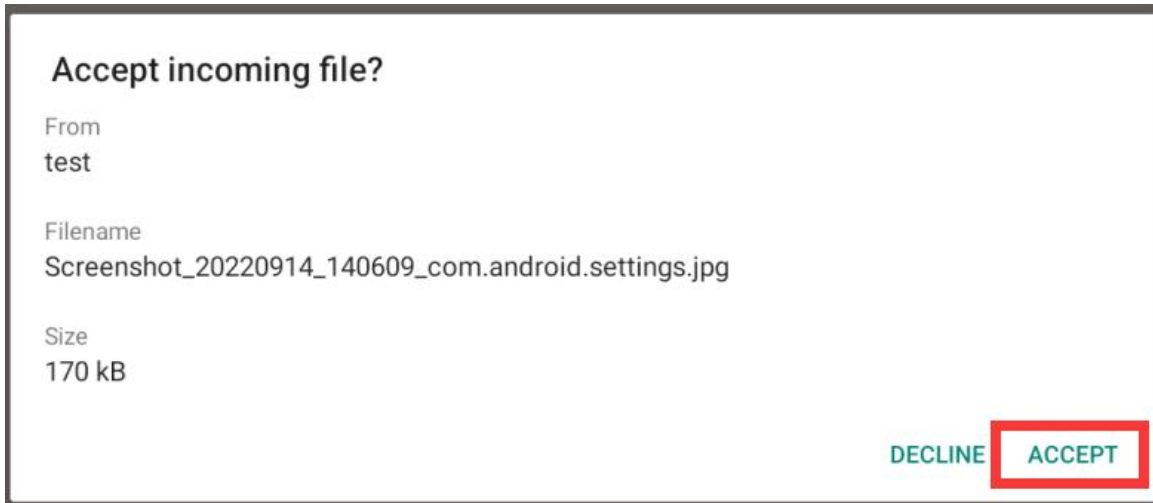
- 6) The test here is the configuration process of the development board and the Bluetooth of the Android mobile phone. At this time, the following confirmation interface will pop up on the mobile phone, and the pairing process will start after clicking the pairing button on the mobile phone.



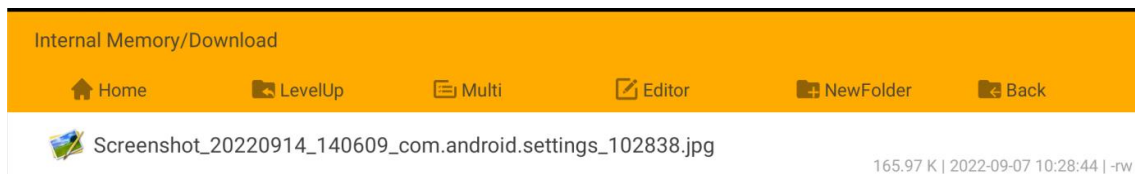
7) After the pairing is complete, you can see the paired Bluetooth devices as shown in the figure below



8) At this time, you can use the Bluetooth of your mobile phone to send a picture to the development board. After sending, you can see the following confirmation interface in the Android system of the development board, and then click **Accept** to start receiving the picture sent by the mobile phone.



9) You can open the **Download** directory in the file manager to view the pictures received by the Android system Bluetooth of the development board.

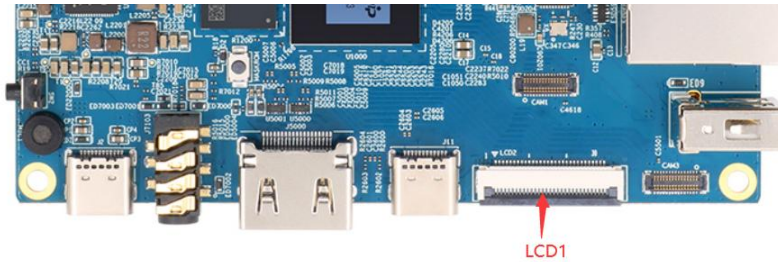


## 8.6. How to use 10.1 inch MIPI screen

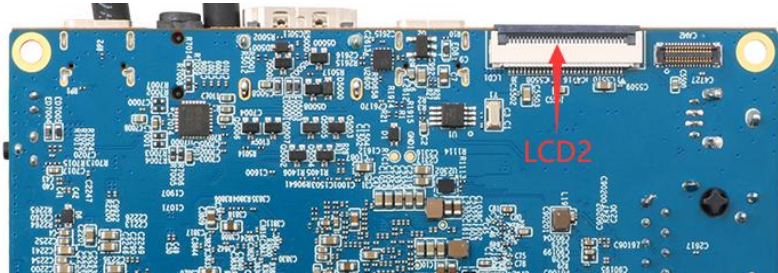
Please make sure that the Android image used is the image of the following version:

**orangepi5b\_RK3588S\_Android12\_lcd\_v1.x.x.img**

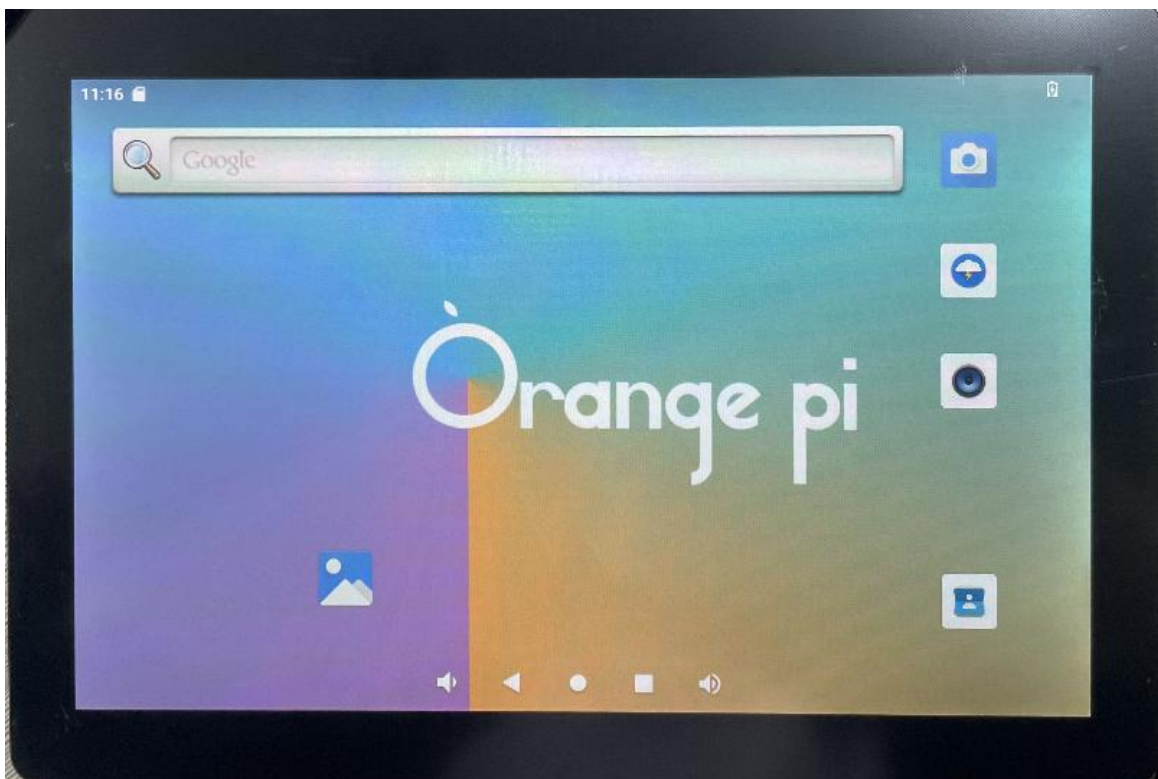
- 1) First you need to assemble the screen, please refer to the assembly [method of the 10.1-inch MIPI screen](#)
- 2) There are two mipi lcd screen interfaces on the development board, we define:
  - c. The location of the lcd1 interface is:



d. The location of the lcd2 interface is:



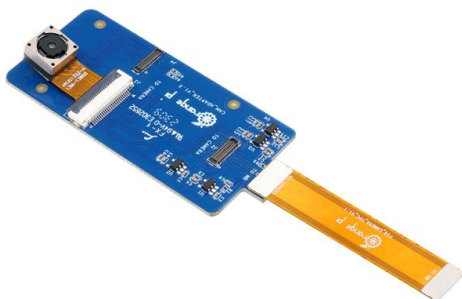
3) Connect the assembled screen to the lcd1 or lcd2 interface, connect the Type-C power supply to the board, and power it on. After the system starts, you can see the screen display as shown in the figure below



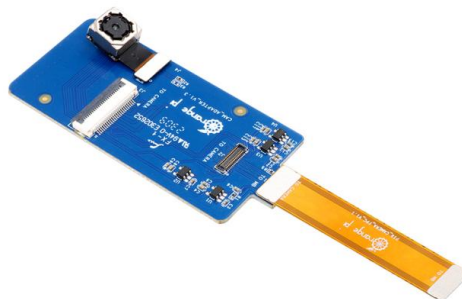
## 8.7. How to test OV13850 and OV13855 MIPI camera

Currently the development board supports two MIPI cameras, OV13850 and OV13855, the specific pictures are as follows:

- a. OV13850 camera with 13 million MIPI interface



- b. OV13855 camera with 13 million MIPI interface



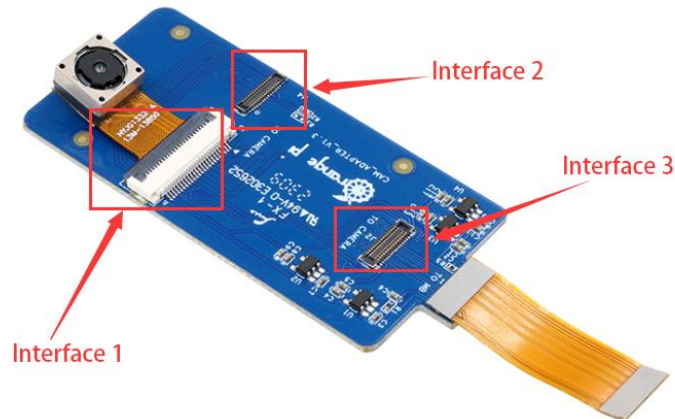
The adapter board and FPC cable used by the OV13850 and OV13855 cameras are the same, but the positions of the two cameras connected to the adapter board are different. FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked **TO MB** needs to be inserted into the camera interface of the development board, and the end marked **TO CAMERA** needs to be inserted into the camera adapter board.



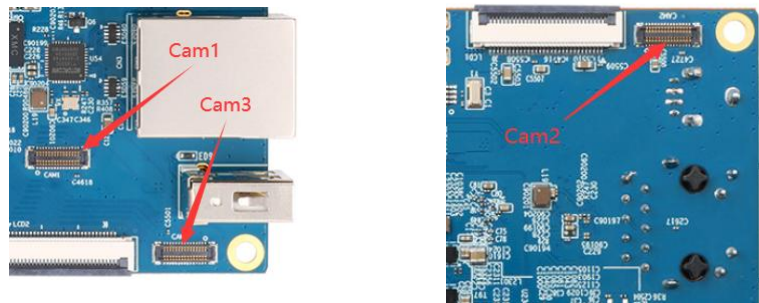
There are a total of 3 camera interfaces on the camera adapter board, and only one can be used at a time, as shown in the figure below:



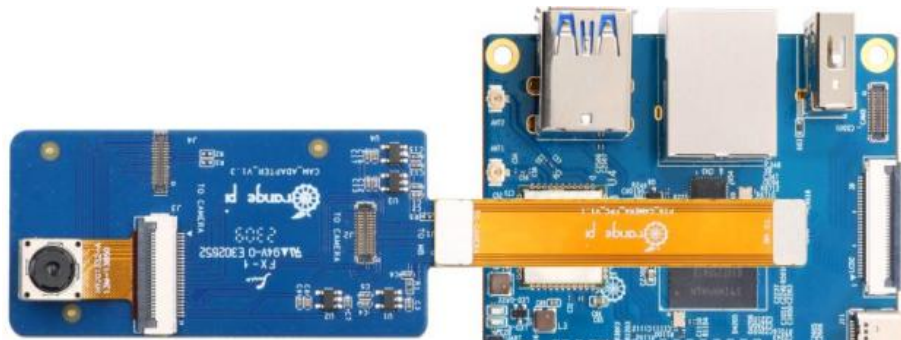
- a. **No. 1 interface is connected to the OV13850 camera**
- b. **No. 2 interface is connected to the OV13855 camera**
- c. No. 3 interface is not used, just ignore it



There are a total of 3 camera interfaces on the Orange Pi 5B development board. We define the positions of Cam1, Cam2 and Cam3 as shown in the figure below:



The method of inserting the camera into the Cam1 interface of the development board is as follows:



The method of inserting the camera into the Cam2 interface of the development board is as follows:





The method of inserting the camera into the Cam3 interface of the development board is as follows:



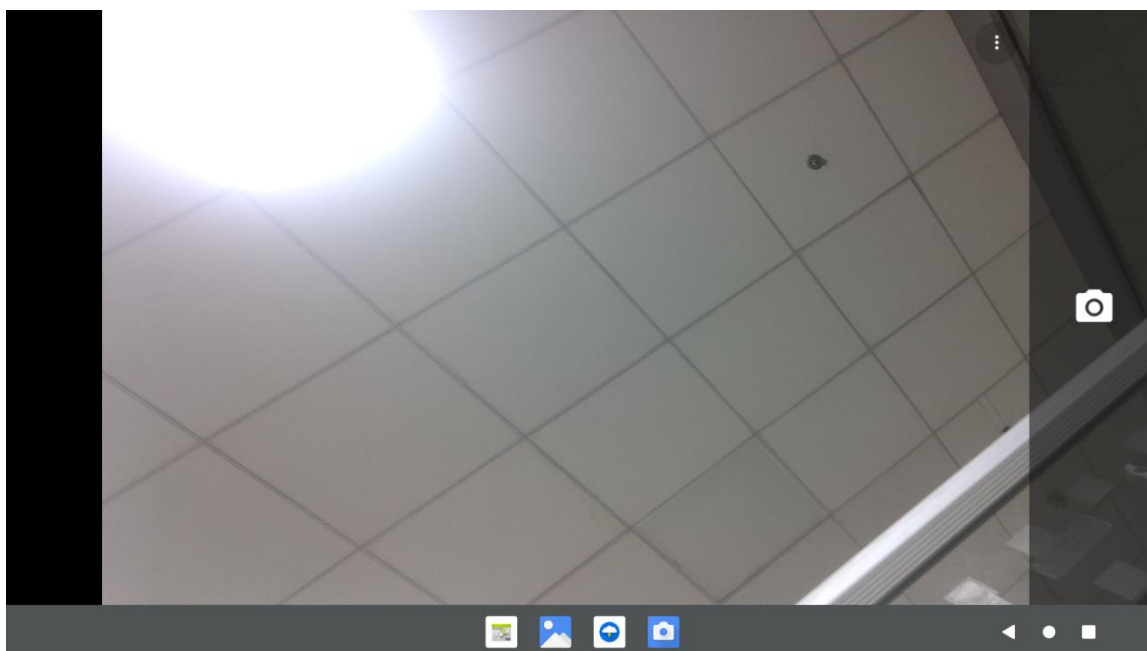
The Android system defaults to the configuration of **Cam1** and **Cam3**, so if you want to use the camera, please choose one of the **Cam1** and **Cam3** interfaces. After connecting the camera to the development board, we can use the following method to test the camera:

- a. Open the camera APP on the desktop





b. Then you can see the preview screen of the camera.

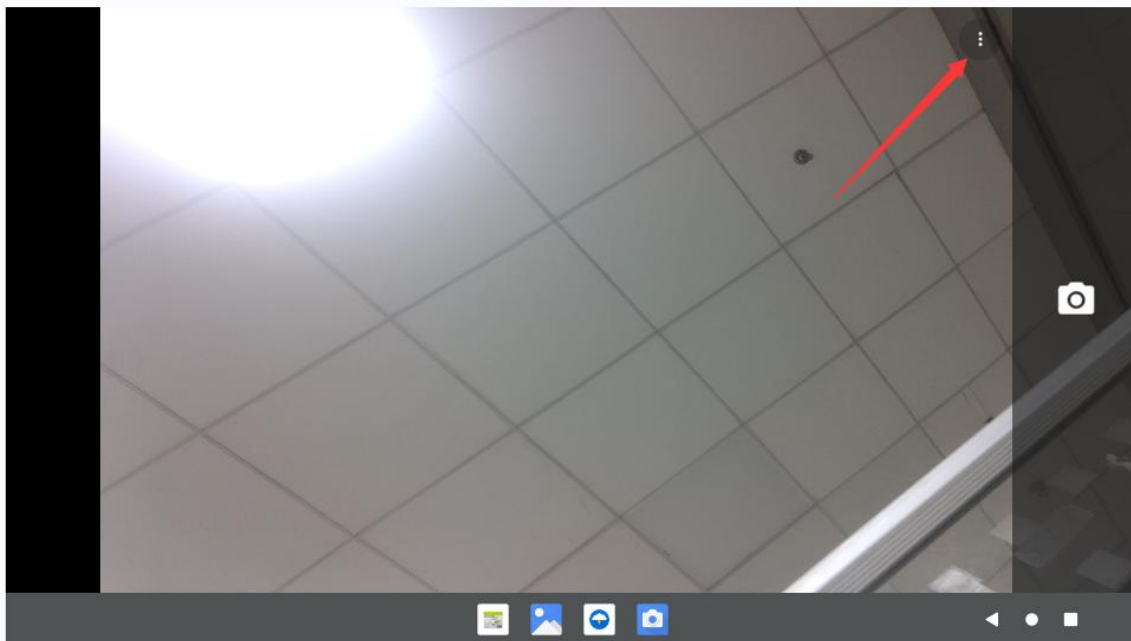


In addition to single camera, we can also connect two cameras at the same time. It should be noted that currently, please use the combination of Cam1+Cam3 for testing dual cameras (support ov13850 and ov13855 mix and match). After connecting the dual cameras, open the camera APP to see the picture of one of the cameras as in the previous steps.

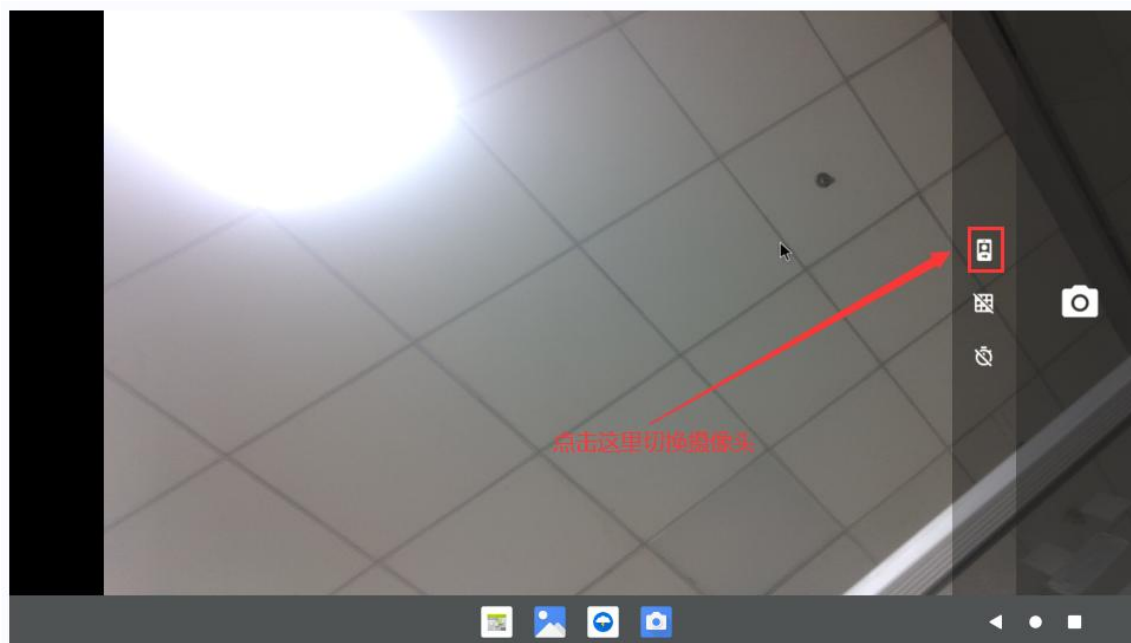


The method to switch to another camera is:

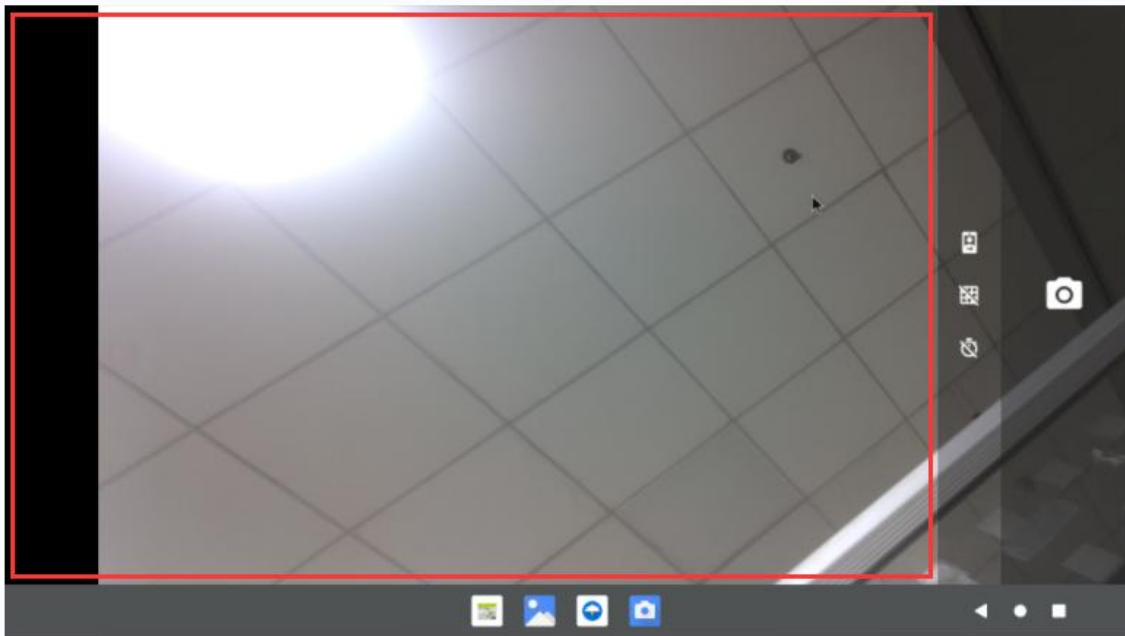
- a. First click the three dots in the upper right corner



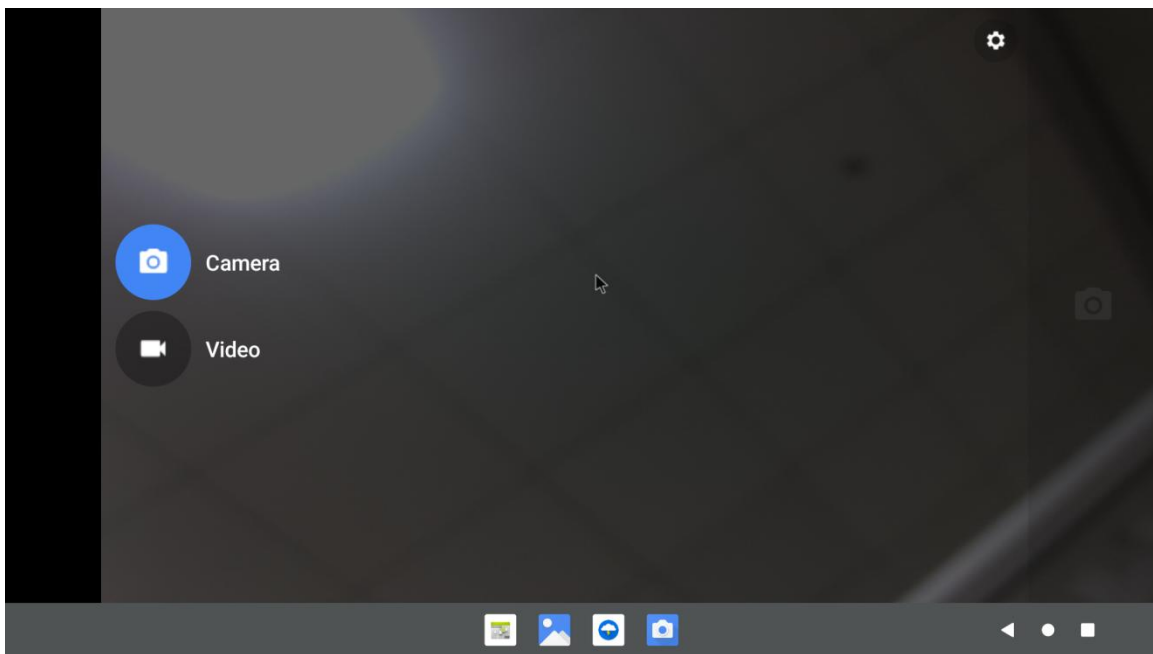
- b. Then click the position shown in the figure below to switch the camera



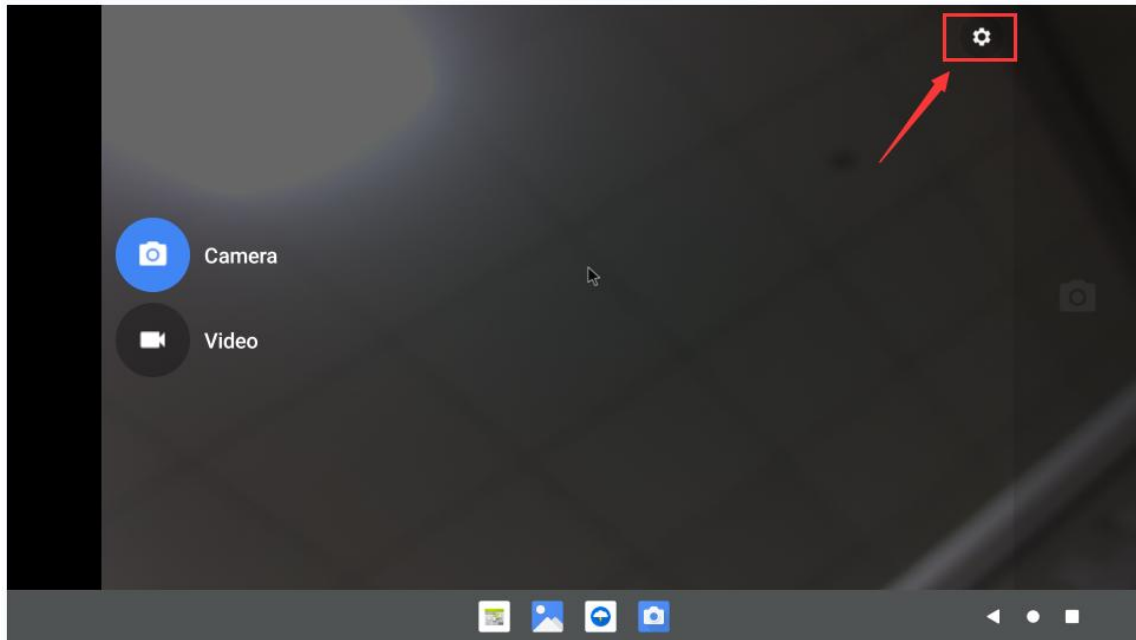
Press and hold the mouse in the area shown in the red box in the picture below of the camera APP and then drag to the right to call up the switching interface for taking pictures and recording



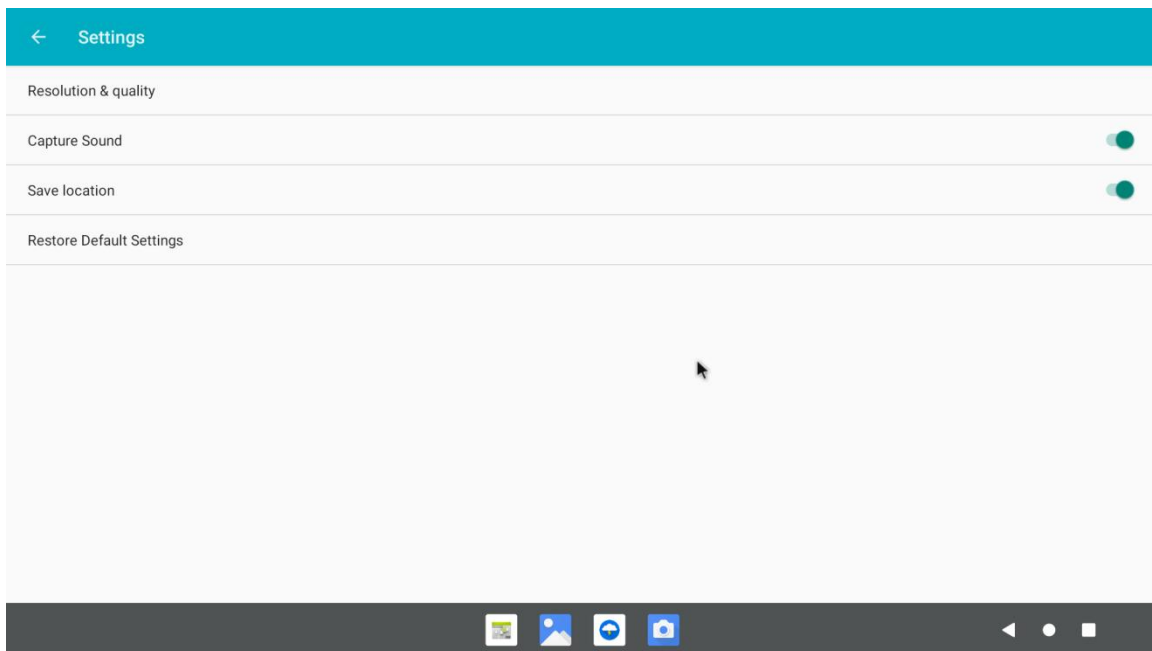
The switching interface of taking pictures and recording is as follows, click **Video** to switch to video recording mode



Click the position shown in the figure below to enter the camera setting interface



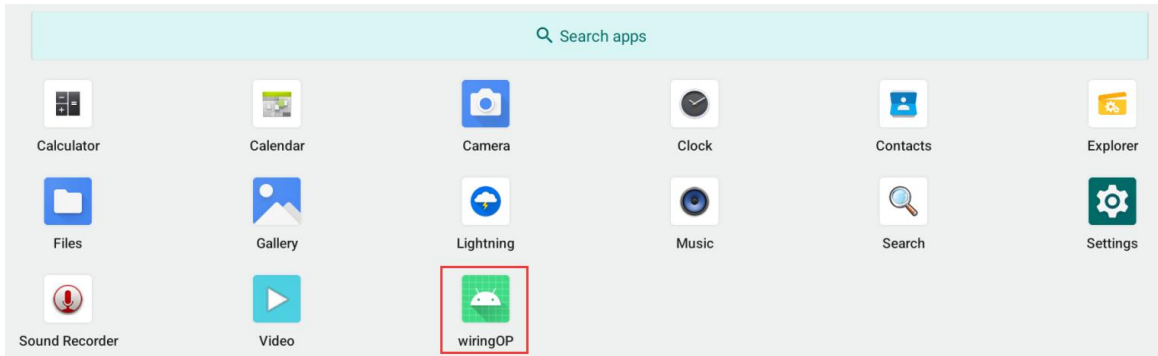
The setting interface of the camera is as follows:



## 8. 8. 26 Pin interface GPIO, UART, SPI and PWM test

### 8. 8. 1. 26pin GPIO port

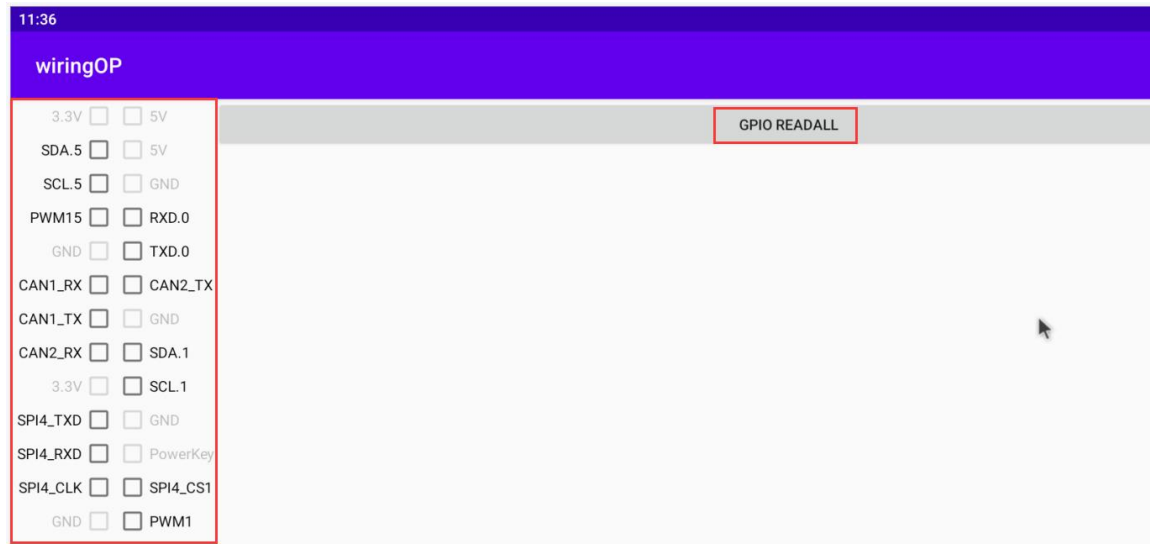
1) First click the Wiringop icon to open the Wiringop App



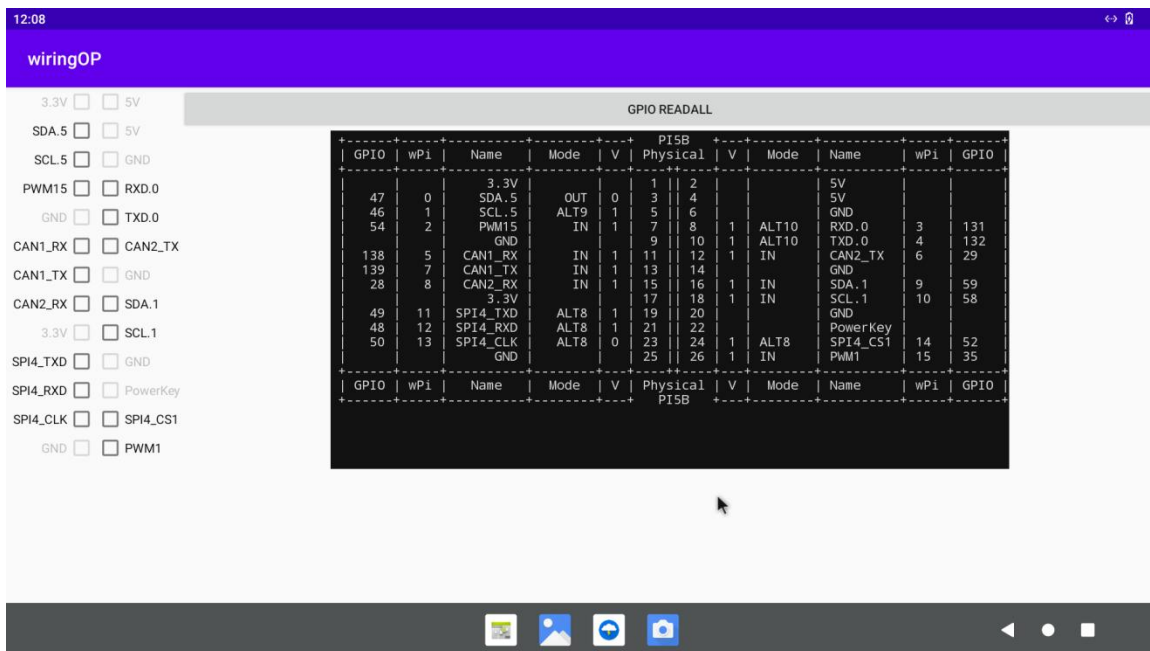
2) The main interface of the Wiringop app is displayed as shown in the figure below, and click the **GPIO\_TEST** button to open the GPIO test interface



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:



5) There are a total of 16 GPIO ports in the 26 pins of the development board that can be used. The following uses 7 pin (the corresponding GPIO is GPIO1\_C6—the corresponding wPi serial number is 2 )as an example to demonstrate how to set the high and low levels of the GPIO port. First click the **CheckBox** button corresponding to 7 pin . When the button is selected, 7 pin will be set to high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **3.3v**, it means setting high level



success

3.3V	<input type="checkbox"/>	5V
SDA.5	<input type="checkbox"/>	5V
SCL.5	<input type="checkbox"/>	GND
PWM15	<input checked="" type="checkbox"/>	RXD.0
GND	<input type="checkbox"/>	TXD.0
CAN1_RX	<input type="checkbox"/>	CAN2_TX
CAN1_TX	<input type="checkbox"/>	GND
CAN2_RX	<input type="checkbox"/>	SDA.1
3.3V	<input type="checkbox"/>	SCL.1
SPI4_TXD	<input type="checkbox"/>	GND
SPI4_RXD	<input type="checkbox"/>	PowerKey
SPI4_CLK	<input type="checkbox"/>	SPI4_CS1
GND	<input type="checkbox"/>	PWM1

6) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is **OUT**, and the pin level is high

wiringOP

3.3V	<input type="checkbox"/>	5V
SDA.5	<input type="checkbox"/>	5V
SCL.5	<input type="checkbox"/>	GND
PWM15	<input checked="" type="checkbox"/>	RXD.0
GND	<input type="checkbox"/>	TXD.0
CAN1_RX	<input type="checkbox"/>	CAN2_TX
CAN1_TX	<input type="checkbox"/>	GND
CAN2_RX	<input type="checkbox"/>	SDA.1
3.3V	<input type="checkbox"/>	SCL.1
SPI4_TXD	<input type="checkbox"/>	GND
SPI4_RXD	<input type="checkbox"/>	PowerKey
SPI4_CLK	<input type="checkbox"/>	SPI4_CS1
GND	<input type="checkbox"/>	PWM1

GPIO READALL

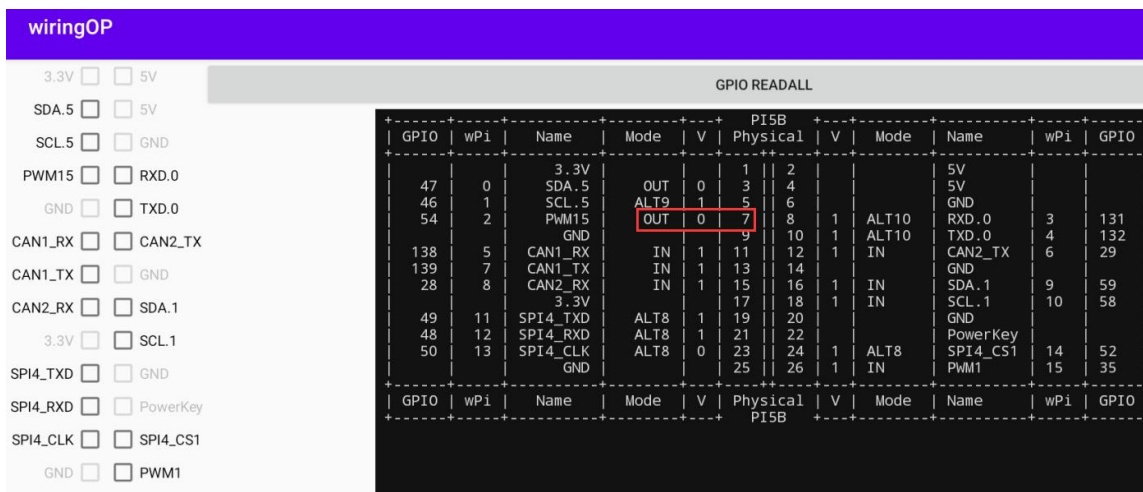
				PI5B							
GPIO	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	GPIO	
47	0	3.3V	OUT	0	1	2		5V			
46	1	SDA.5	ALT9	1	3	4		5V			
54	2	SCL.5	OUT	1	5	6		GND			
		PWM15	OUT	1	7	8	1	ALT10	RXD.0	3	
		GND			9	10	1	ALT10	TXD.0	4	
138	5	CAN1_RX	IN	1	11	12	1	IN	CAN2_TX	6	
139	7	CAN1_TX	IN	1	13	14		GND		29	
28	8	CAN2_RX	IN	1	15	16	1	IN	SDA.1	9	
		3.3V			17	18	1	IN	SCL.1	10	
49	11	SPI4_TXD	ALT8	1	19	20		GND		58	
48	12	SPI4_RXD	ALT8	1	21	22		PowerKey			
50	13	SPI4_CLK	ALT8	0	23	24	1	ALT8	SPI4_CS1	14	
		GND			25	26	1	IN	PWM1	15	
										35	



7) Click the **CheckBox** button in the figure below again to uncheck the status, and pin 7 will be set to low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level is set successfully.

3.3V	<input type="checkbox"/>	<input type="checkbox"/>	5V
SDA.5	<input type="checkbox"/>	<input type="checkbox"/>	5V
SCL.5	<input type="checkbox"/>	<input type="checkbox"/>	GND
PWM15	<input type="checkbox"/>	<input type="checkbox"/>	RXD.0
GND	<input type="checkbox"/>	<input type="checkbox"/>	TXD.0
CAN1_RX	<input type="checkbox"/>	<input type="checkbox"/>	CAN2_TX
CAN1_TX	<input type="checkbox"/>	<input type="checkbox"/>	GND
CAN2_RX	<input type="checkbox"/>	<input type="checkbox"/>	SDA.1
3.3V	<input type="checkbox"/>	<input type="checkbox"/>	SCL.1
SPI4_TXD	<input type="checkbox"/>	<input type="checkbox"/>	GND
SPI4_RXD	<input type="checkbox"/>	<input type="checkbox"/>	PowerKey
SPI4_CLK	<input type="checkbox"/>	<input type="checkbox"/>	SPI4_CS1
GND	<input type="checkbox"/>	<input type="checkbox"/>	PWM1

8) Then click the **GPIO READALL** button, you can see that the current mode of pin 7 is OUT, and the pin level is low

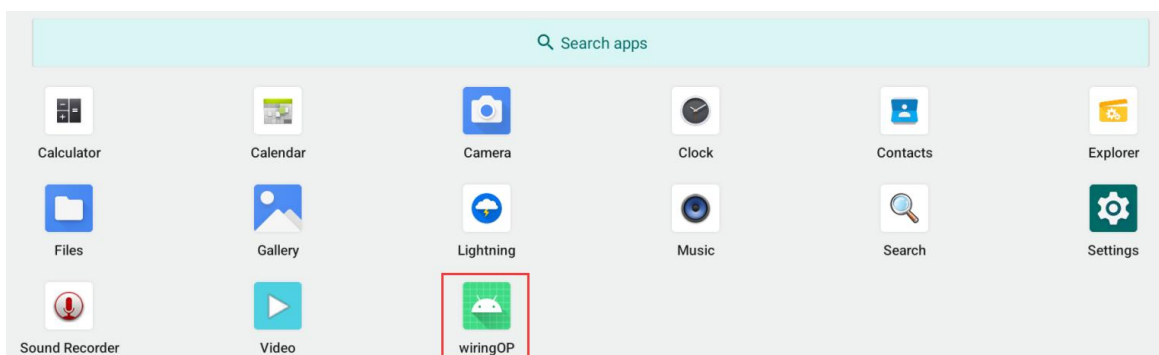


### 8. 8. 2. 26pin UART test

1) In Android default, the UART0 serial port is only opened. The position of UART0 at 26 pin is shown in the figure below. The corresponding device node is **dev/ttyS0**



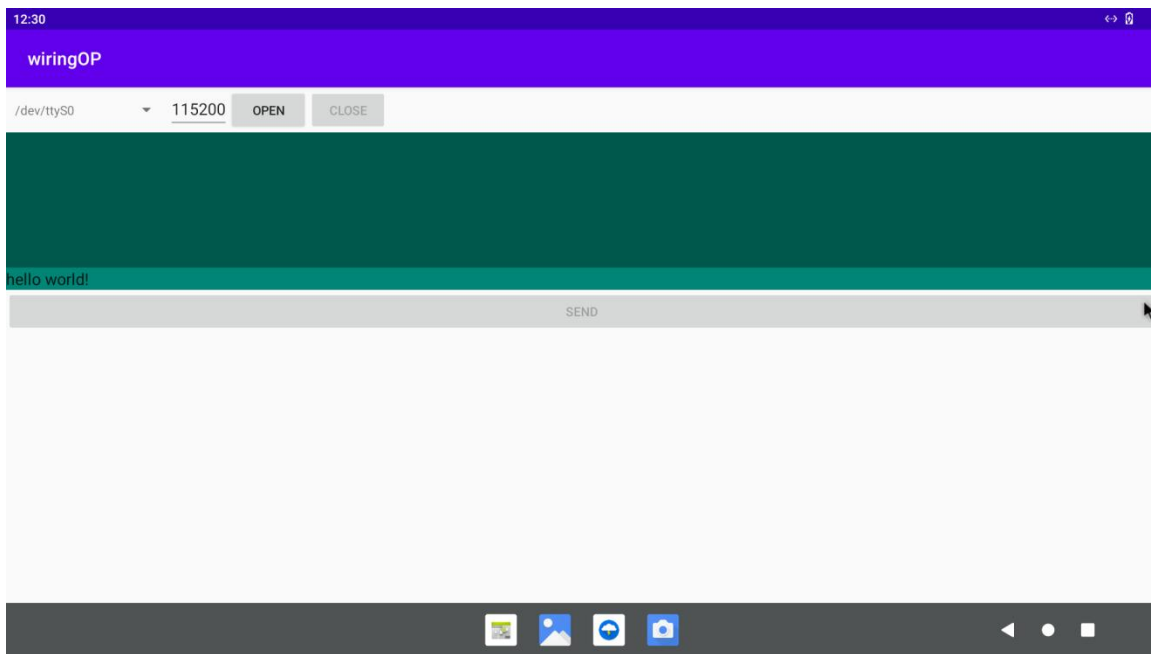
2) First click the WiringOP icon to open the wiringOP APP



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



4) The serial test interface of the APP is shown in the figure below



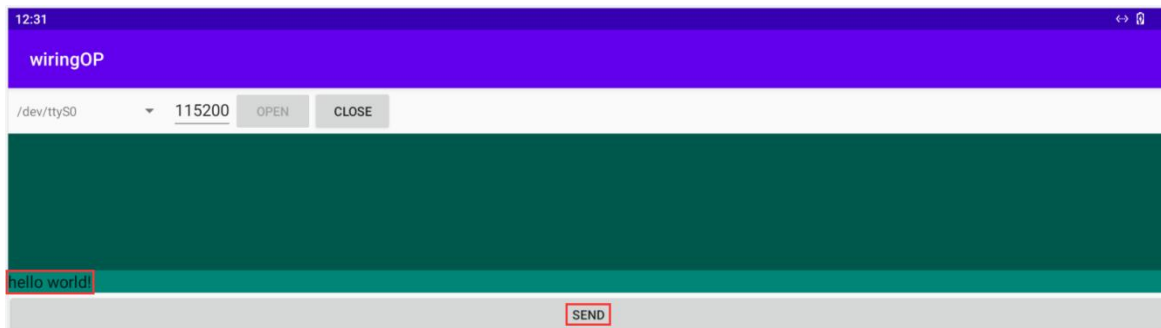
5) Then enter the baud rate you want to set in the editing box, and then click the **OPEN** button to open the **/dev/ttyS0** node. After successful, the **OPEN** button becomes an optional state. The **CLOSE** button and the **SEND** button become an optional state



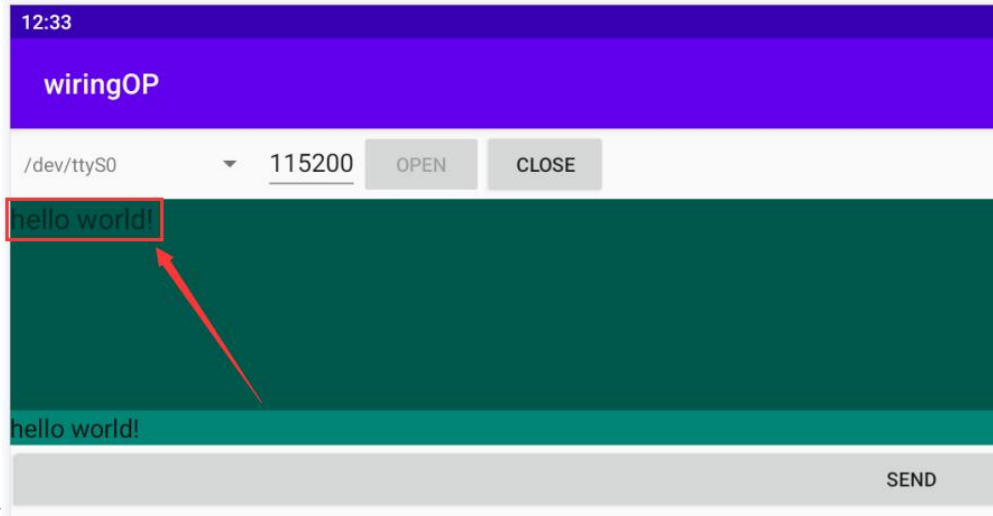
6) Then use the DuPont line to shorten the RXD and TXD



7) Then you can enter a section of characters in the editing box below, click the **SEND** button to start sending



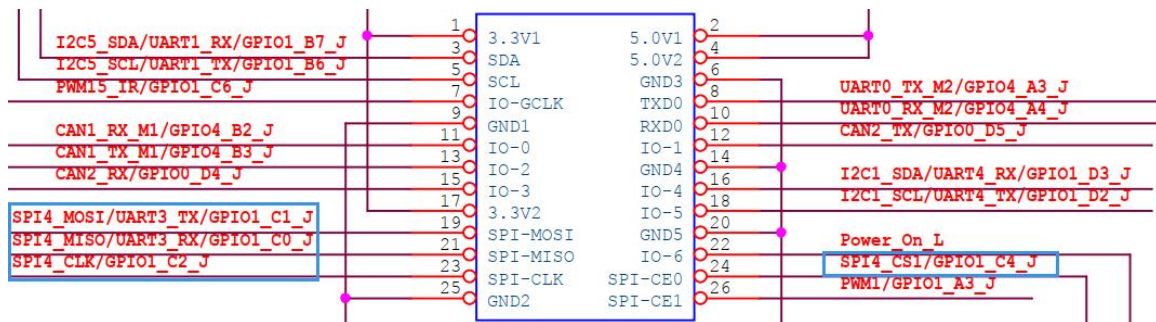
8) If everything is normal, the receiving string will be displayed in the receiving



box

### 8. 8. 3. 26pin SPI test

1) From the schematic diagram of the 26pin interface, the SPI available for Orange Pi 5B is spi4

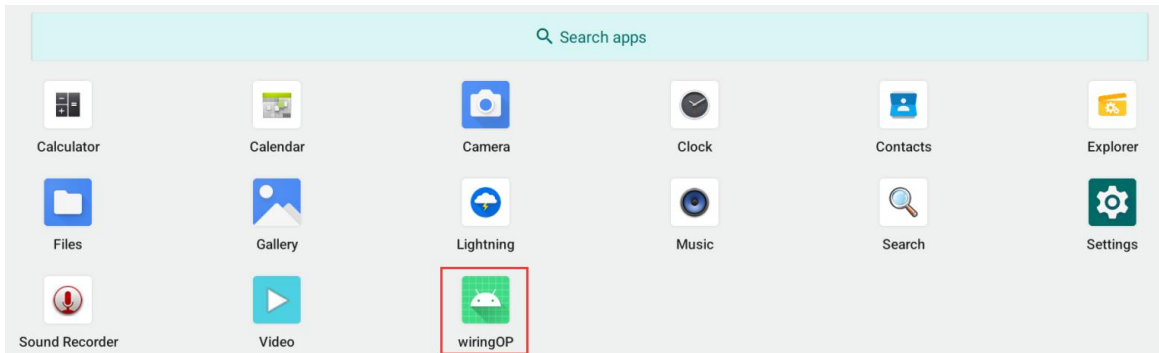


2) Here is the w25q64 module to test the SPI interface, and first access the w25q64 device at the SPI4 interface





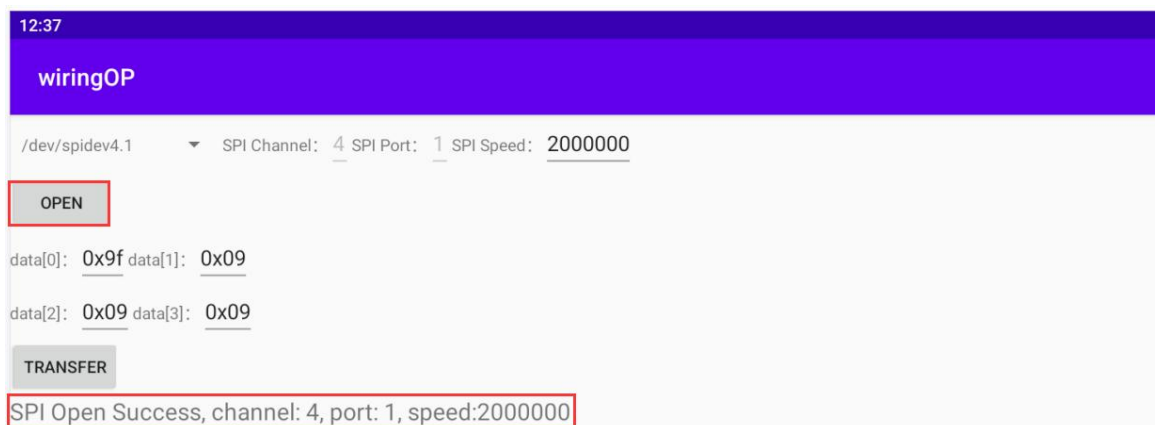
3) Then click the WiringOP icon to open the wiringOP APP



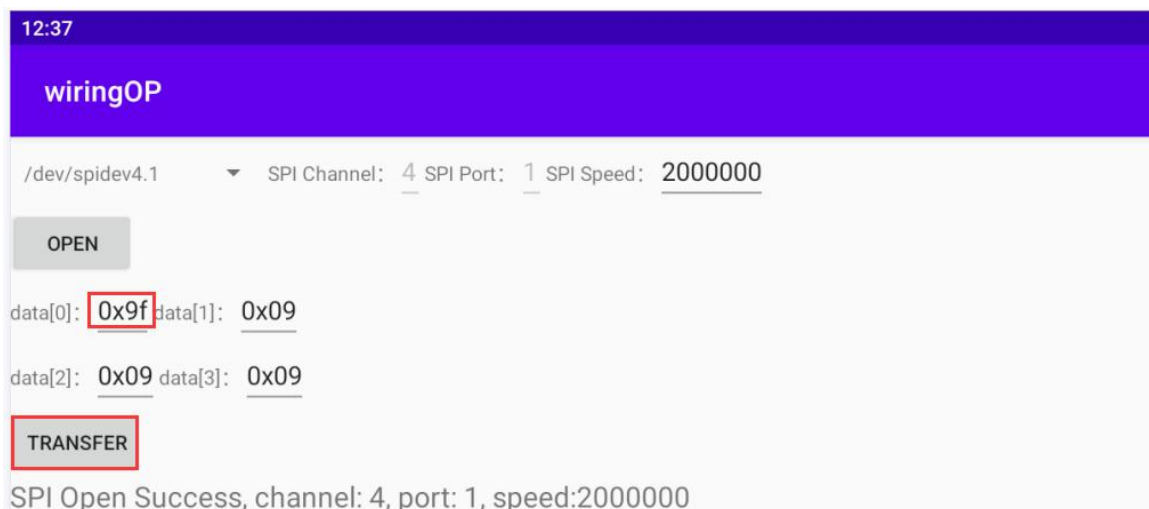
4) The main interface of the WiringOP APP shows as shown in the figure below, click the SPI\_TEST button to open the SPI test interface



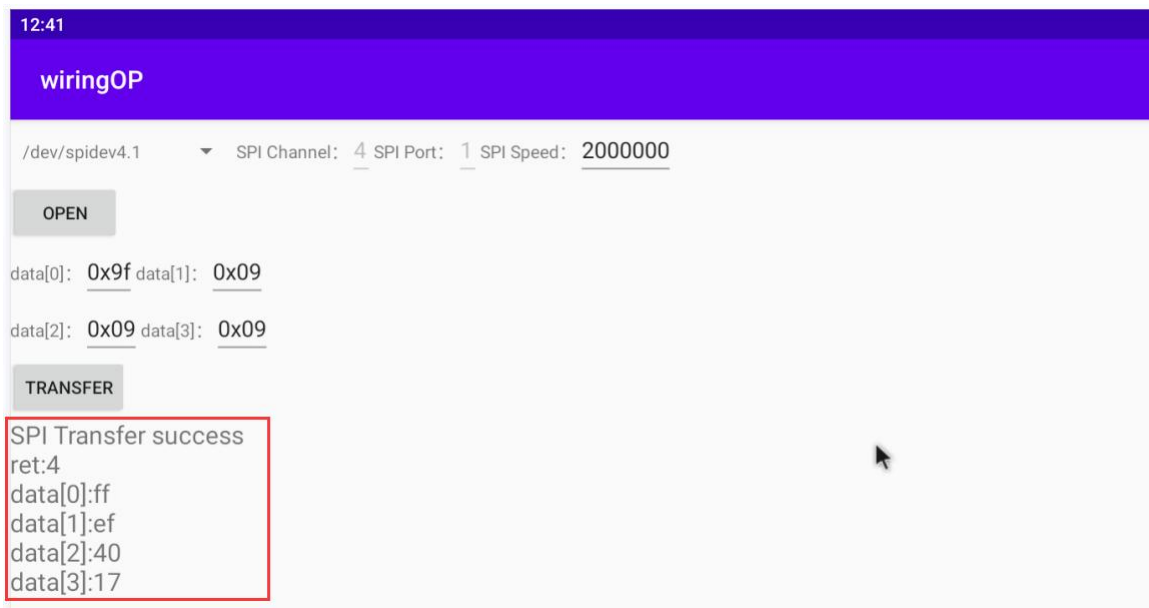
5) Then click the **OPEN** button to initialize SPI



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**



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

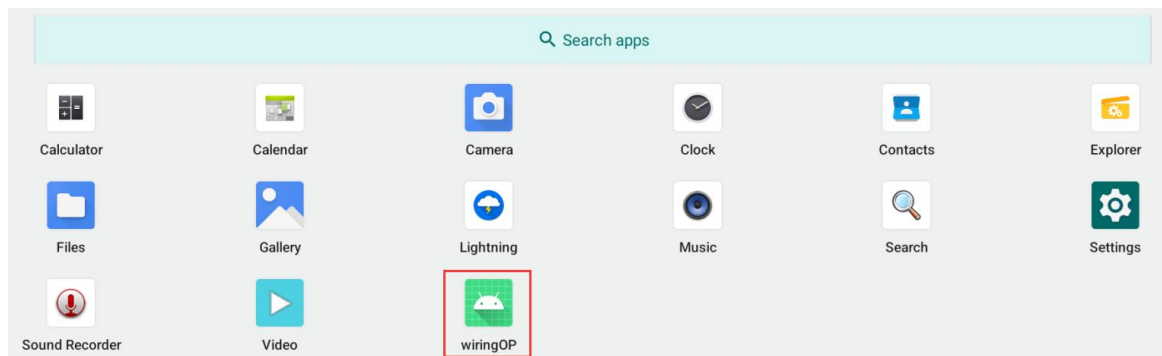


## 8.8.4. 26 pin PWM test

1) Android only opened **PWM 15** by default. The corresponding pins are at the position of 26 Pin



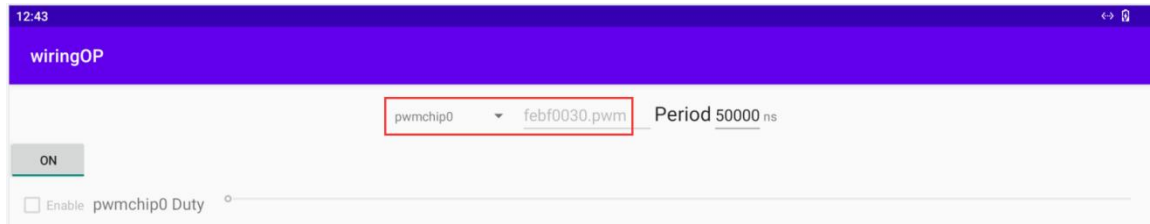
2) First click the WiringOP icon to open the Wiringop App



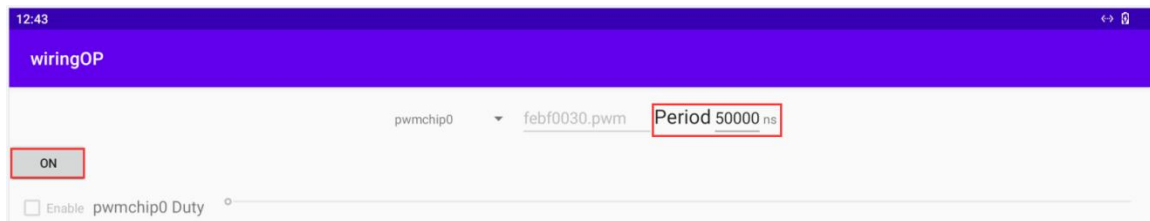
3) Then click the **PWM\_TEST** button to enter the PWM test interface at the main interface of WiringOP



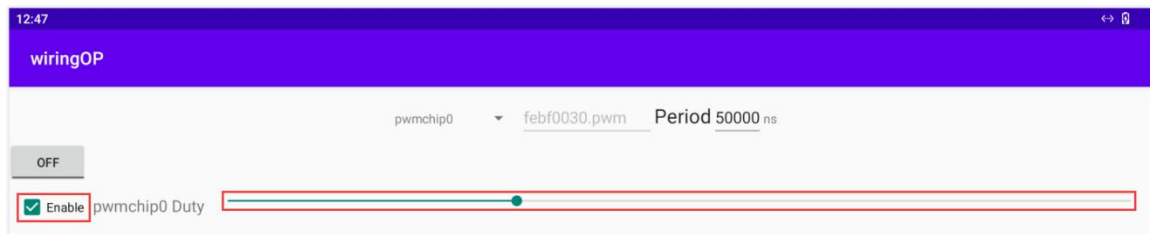
4) The corresponding address corresponding to the PWM15 is **febf0030**. The right side of PWMCHIP0 is exactly the **febf0030.pwm**. If the displayed base address is wrong, please click the drop -down option to select other PWMCHIP until the **febf0030** is displayed on the right.



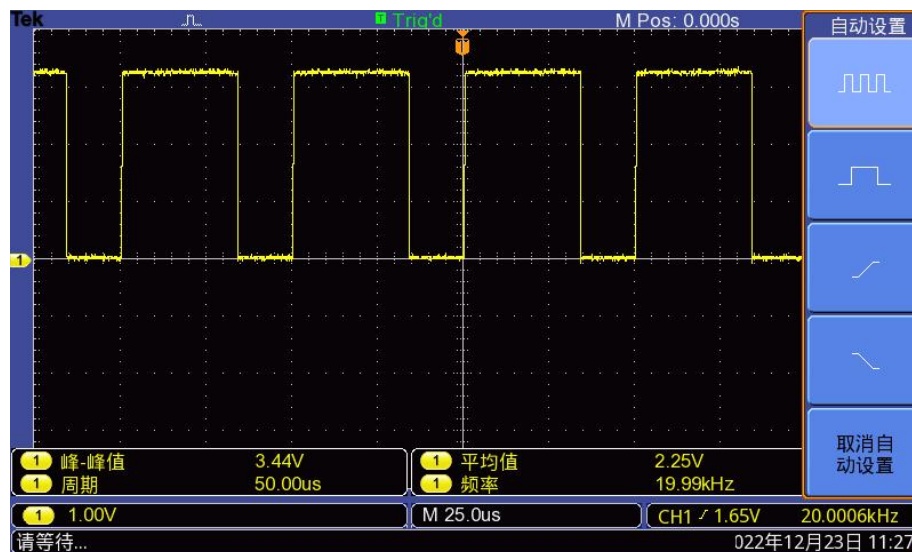
5) Then confirm the PWM cycle. The default configuration is **50000ns**, and the PWM frequency is **20KHz**. You can modify it by yourself. Click on the button to export **PWM15**



6) Then drag the drag bar below to change the PWM duty cycle, and then check Enable to output the PWM waveform



7) Then use an oscilloscope to measure the No. 7 pin in the 26pin of the development board, and you can see the following waveform.



## 8.9. How to use ADB

### 8.9.1. Use a data cable to connect to adb debugging

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



5) Then use the Type-C data cable to connect the development board to the USB interface of the computer (please use the Type-C power supply to power the development board at the same time)

6) Install adb tool on Ubuntu PC

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

7) You can view the identified ADB devices with the following command

```
test@test:~$ adb devices
```



```
List of devices attached
S63QCF54CJ    device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

8) Then you can log in to the android system through the adb shell on the Ubuntu PC

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

9) Execute the command to remount the Android system

```
test@test:~$ adb root
test@test:~$ adb remount
```

10) Then you can transfer files to the Android system

```
test@test:~$ adb push example.txt /system/
```

### 8. 9. 2. adb debug using network connection

Using the network adb does not require a USB Type C interface data cable to connect the computer and the development board, but to communicate through the network, so first make sure that the wired or wireless network of the development board has been connected, and then obtain the IP address of the development board that will be used later.

1) Make sure **service.adb.tcp.port** of Android system is set to port number 5555

```
console:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the port number of network adb

```
console:/ # setprop service.adb.tcp.port 5555
console:/ # stop adbd
console:/ # start adbd
```

3) Install adb tool on Ubuntu PC



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

4) Then connect network adb on Ubuntu PC

```
test@test:~$ adb connect 192.168.1.xxx    (The IP address needs to be changed to  
the IP address of the development board)  
* daemon not running; starting now at tcp:5037  
* daemon started successfully  
connected to 192.168.1.xxx:5555  
  
test@test:~$ adb devices  
List of devices attached  
192.168.1.xxx:5555      device
```

5) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell  
console:/ #
```

## 8. 10. 2.4G USB remote control tested by Android Box

7) 2.4G USB remote control that has been tested so far is shown in the figure below

a. contains a remote



b. A USB wireless receiver



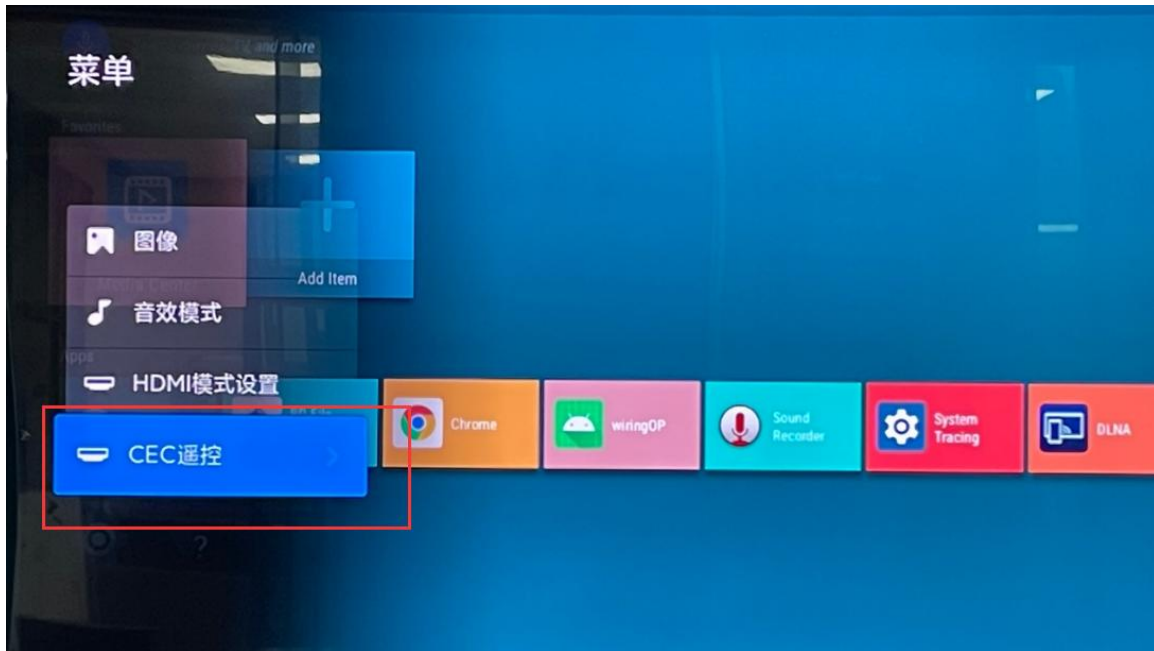
8) The Android Box system does not require any configuration, it can be used after plugging it in

### 8. 11. How to use HDMI CEC function in Android Box system

**HDMI CEC allows users to control all connected devices through HDMI with only one remote control. Based on this function, we can control the development board with the remote control of the TV.**

**Before testing this function, please make sure your TV supports HDMI CEC.**

- 1) First connect the development board to the TV via the HDMI cable, then power on and start
- 2) Then turn on the HDMI CEC function in the TV settings. Different TVs may have different ways to turn it on. Here we take Xiaomi TV as an example. Press the menu button on the remote control, then select CEC remote control and press the confirmation button



3) Then select "On" to open the HDMI CEC remote control



4) At this point, you can control the Android Box system of the development board through the remote control of the TV





## 9. The compilation method of the Android 12 source code

### 9.1. Download the source code of Android 12

1) First download the sub -roll compression package of Android 12 source code from Google Drive

a. Google network disk

RK3588S\_Android\_Source\_Code

名称	所有者	上次修改日期	文件大小
Android_12.tar.gz.md5sum		2022年12月6日	432 个字节
Android_12.tar.gz00		2022年12月6日	4 GB
Android_12.tar.gz01		2022年12月6日	4 GB
Android_12.tar.gz02		2022年12月6日	4 GB
Android_12.tar.gz03		2022年12月6日	4 GB
Android_12.tar.gz04		2022年12月6日	4 GB
Android_12.tar.gz05		2022年12月6日	4 GB
Android_12.tar.gz06		2022年12月6日	4 GB
Android_12.tar.gz07		2022年12月6日	440 MB

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
```



## 9. 2. Compile the source code of Android 12

1) First install the software package required to compile the Android 12 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 follow

- c. **-U**: Compile uboot
- d. **-K**: Compile kernel
- e. **-A**: Compile android
- f. **-u**: Package generates update.img and update\_spi\_nvme.img
- g. **-o**: Compile OTA bag
- h. **-d**: Specify kernel dts

3) Compile Uboot, Kernel, Android and pack it into update.img.

- a. The command to compile and support HDMI 8K display mirroring (LCD off by default) is as follows:

```
test@test:~$ cd Android_12
test@test:~/Android_12$ export BOARD=orangeipi5b
test@test:~/Android_12$ source build/envsetup.sh
test@test:~/Android_12$ lunch rk3588s_s-userdebug
test@test:~/Android_12$ ./build.sh -AUKu
```

- b. The command to compile and support LCD display image is as follows:

```
test@test:~$ cd Android_12
test@test:~/Android_12$ export BOARD=orangeipi5
test@test:~/Android_12$ export DUAL_LCD=true
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.img  
update.img
```

## 10. Appendix

### 10. 1. User Manual Update History

Version	Date	Update Notes
v1.0	2023-04-12	initial version



v1.1	2023-04-21	<ol style="list-style-type: none"> <li>1. Orange Pi OS Arch system adaptation status</li> <li>2. OPi OS Arch: How to use the 10.1-inch MIPI LCD screen</li> <li>3. OPi OS Arch: How to use AP6275P PCIe WIFI6+ Bluetooth module</li> <li>4. OPi OS Arch: Test method for OV13850 and OV13855 MIPI cameras</li> <li>5. OPi OS Arch: How to install wiringOP</li> <li>6. OPi OS Arch: 26pin GPIO、I2C、UART、SPI、CAN 和 PWM 测试</li> <li>7. Exchange the position descriptions of LCD1 and LCD2 to be consistent with the silk screen on the development board.</li> <li>8. Linux: How to set the pull-down resistor for the 26pin GPIO port</li> <li>9. Linux: How to use the CAN bus in 26pin</li> <li>10. OPi OS Arch: How to set up the Chinese environment and install the Chinese input method</li> </ol>
v1.2	2023-06-01	<ol style="list-style-type: none"> <li>1. Ubuntu/Debian: How to use ZFS file system</li> <li>2. Ubuntu/Debian: How to turn off the green light by default at startup</li> <li>3. Added usage instructions for Debian12 system (scattered in multiple sections)</li> </ol>
v1.3	2024-01-26	<ol style="list-style-type: none"> <li>1. How to burn Orange Pi OS (OH) image to TF card</li> <li>2. How to burn Orange Pi OS (OH) image into eMMC</li> </ol>
v1.4	2024-02-01	<ol style="list-style-type: none"> <li>1. Linux6.1 system adaptation situation</li> <li>2. Linux: How to use NPU</li> <li>3. Linux: RK3588 How to use Baidu Flying Plate</li> </ol>

## 10.2. Image Update History

Date	Update Notes
2023-04-12	<p>Orangepi5b_1.0.0_debian_bullseye_server_linux5.10.110.7z</p> <p>Orangepi5b_1.0.0_debian_bullseye_desktop_xfce_linux5.10.110.7z</p> <p>Orangepi5b_1.0.0_debian_bullseye_desktop_kde-plasma_linux5.10.110.7z</p> <p>Orangepi5b_1.0.0_ubuntu_focal_server_linux5.10.110.7z</p> <p>Orangepi5b_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.110.7z</p> <p>Orangepi5b_1.0.0_ubuntu_jammy_server_linux5.10.110.7z</p> <p>Orangepi5b_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.110.7z</p>



	<p>OrangePi5b_1.0.0_ubuntu_jammy_desktop_gnome_linux5.10.110.7z</p> <p>OrangePi5B_RK3588S_Android12_v1.0.0.tar.gz</p> <p>OrangePi5B_RK3588S_Android12_lcd_v1.0.0.tar.gz</p> <p>OrangePi5B_RK3588S_Android12-box_v1.0.0.tar.gz</p> <p>Opios-droid-aarch64-opi5b-23.04-linux5.10.110.tar.gz</p> <p>Opios-droid-aarch64-opi5b-23.04-linux5.10.110-en.tar.gz</p> <p>* initial version</p>
2023-04-14	<p>OrangePi5b_1.0.2_debian_bullseye_server_linux5.10.110.7z</p> <p>OrangePi5b_1.0.2_debian_bullseye_desktop_xfce_linux5.10.110.7z</p> <p>OrangePi5b_1.0.2_debian_bullseye_desktop_kde-plasma_linux5.10.110.7z</p> <p>OrangePi5b_1.0.2_ubuntu_focal_server_linux5.10.110.7z</p> <p>OrangePi5b_1.0.2_ubuntu_focal_desktop_xfce_linux5.10.110.7z</p> <p>OrangePi5b_1.0.2_ubuntu_jammy_server_linux5.10.110.7z</p> <p>OrangePi5b_1.0.2_ubuntu_jammy_desktop_xfce_linux5.10.110.7z</p> <p>OrangePi5b_1.0.2_ubuntu_jammy_desktop_gnome_linux5.10.110.7z</p> <p>* Fix the problem that the Linux system in the TF card cannot be started when burning Android or Orange Pi OS (Droid) system in eMMC</p>
2023-04-21	<p>Opios-arch-aarch64-gnome-opi5b-23.04-linux5.10.110.img.xz</p> <p>* initial version</p> <p>OrangePi5b_1.0.4_debian_bullseye_server_linux5.10.110.7z</p> <p>OrangePi5b_1.0.4_debian_bullseye_desktop_xfce_linux5.10.110.7z</p> <p>OrangePi5b_1.0.4_debian_bullseye_desktop_kde-plasma_linux5.10.110.7z</p> <p>OrangePi5b_1.0.4_ubuntu_focal_server_linux5.10.110.7z</p> <p>OrangePi5b_1.0.4_ubuntu_focal_desktop_xfce_linux5.10.110.7z</p> <p>OrangePi5b_1.0.4_ubuntu_jammy_server_linux5.10.110.7z</p> <p>OrangePi5b_1.0.4_ubuntu_jammy_desktop_xfce_linux5.10.110.7z</p> <p>OrangePi5b_1.0.4_ubuntu_jammy_desktop_gnome_linux5.10.110.7z</p>



	<p>* Exchange the dtbo configuration of LCD1 and LCD2 to be consistent with the silk screen on the development board.</p>
2024-02-01	<p>OrangePi5b_1.0.8_ubuntu_focal_server_linux6.1.43.7z OrangePi5b_1.0.8_ubuntu_jammy_server_linux6.1.43.7z OrangePi5b_1.0.8_debian_bullseye_server_linux6.1.43.7z OrangePi5b_1.0.8_debian_bookworm_server_linux6.1.43.7z OrangePi5b_1.0.8_ubuntu_focal_desktop_xfce_linux6.1.43.7z OrangePi5b_1.0.8_ubuntu_jammy_desktop_xfce_linux6.1.43.7z OrangePi5b_1.0.8_debian_bullseye_desktop_xfce_linux6.1.43.7z OrangePi5b_1.0.8_debian_bookworm_desktop_xfce_linux6.1.43.7z</p> <p>* initial version</p>