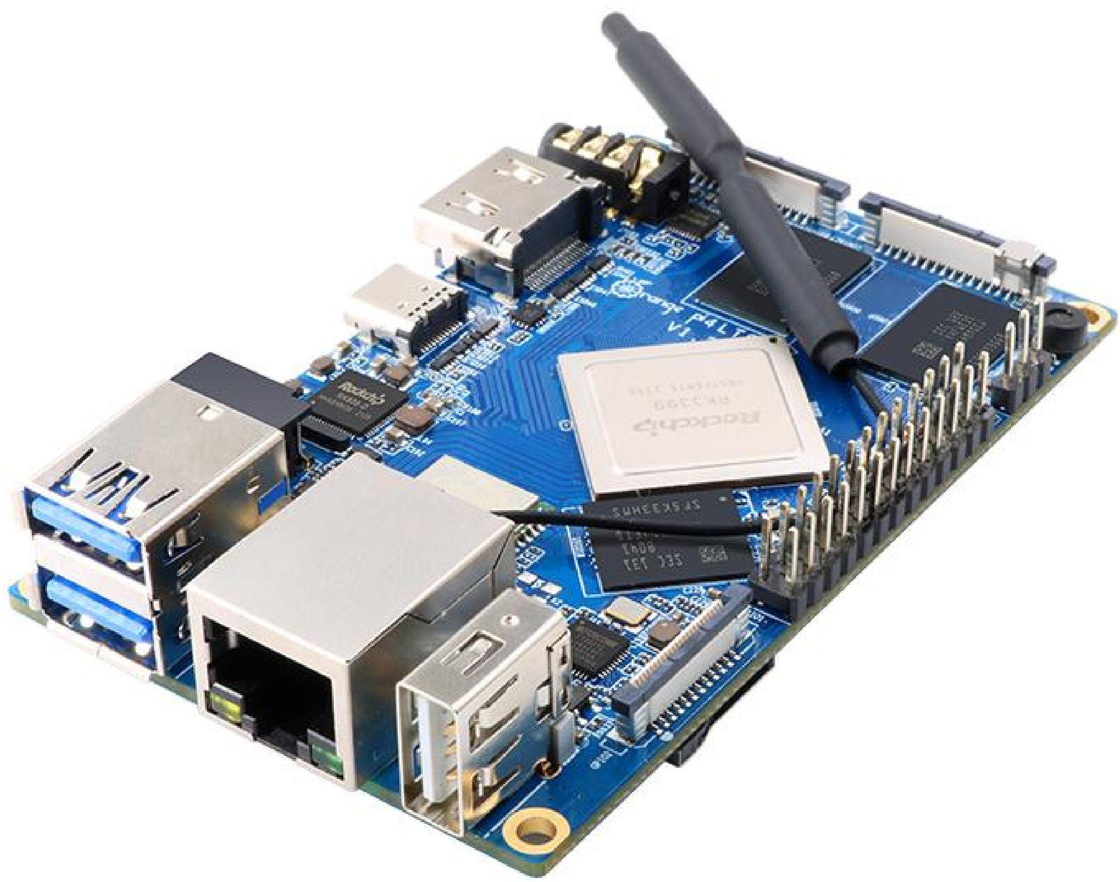
Orange Pi 4 LTS
User Manual



Contents

1. Basic Features of Orange Pi 4 LTS 1
1. 1. What is Orange Pi 4 LTS？ 1
1. 2. Purpose of Orange Pi 4 LTS 1
1. 3. Who is Orange Pi 4 LTS designed for? 1
1. 4. Hardware Features of Orange Pi 4 LTS 2
1. 5. Top view and bottom view of Orange Pi 4 LTS 4
1. 5. 1. RK3399+4GB LPDDR4 with 16GB eMMC version 4
1. 5. 2. RK3399+4GB LPDDR4 without 16GB eMMC version 5
1. 5. 3. RK3399-T+3GB LPDDR4 with 16GB eMMC version 6
1. 5. 4. RK3399-T+3GB LPDDR4 without 16GB eMMC version 7
1. 6. Interface details of Orange Pi 4 LTS 9
1. 6. 1. RK3399+4GB LPDDR4 with 16GB eMMC version 9
1. 6. 2. K3399+4GB LPDDR4 without 16GB eMMC version 10
1. 6. 3. RK3399-T+3GB LPDDR4 with 16GB eMMC version 11
1. 6. 4. RK3399-T+3GB LPDDR4 without 16GB eMMC version 12
2. Introduction to the use of the development board 14
2. 1. Prepare the necessary accessories 14
2. 2. Download the image of the development board and related materials 18
2. 3. Method of burning Linux image to TF card based on Windows PC 19
2. 3. 1. How to use Win32Diskimager to burn Linux image 19
2. 3. 2. How to use balenaEtcher to burn a Linux image 21
2. 4. Method of burning Linux image to TF card based on Ubuntu PC 26
2. 5. How to program Linux image to eMMC 30
2. 6. How to burn Android image to TF card 30
2. 7. The method of burning Android image to eMMC based on Windows PC 33

2. 7. 1. Burn Android firmware to eMMC via TF card 34
2. 7. 2. How to burn Android image into eMMC via Type-C cable 37
2. 8. Method of Burning Android Image to eMMC Based on Ubuntu PC 42
2. 9. How to enter MaskRom mode 44
2. 10. Installation instructions of metal heat dissipation shell 46
2. 11. Start the orange pi development board 51
2. 12. How to debug the serial port 52
2. 12. 1. Connection instructions for debugging serial port 52
2. 12. 2. How to use the debugging serial port on Ubuntu platform 54
2. 12. 3. How to use the debugging serial port on Windows platform 57
3. Instructions for use of Linux system 60
3. 1. Supported linux distribution types and kernel versions 61
3. 2. Linux kernel driver adaptation 63
3. 3. Description of the linux command format in this manual 64
3. 4. Linux system login instructions 66
3. 4. 1. Linux system default login account and password 66
3. 4. 2. How to set the automatic login of the Linux system terminal 66
3. 4. 3. Instructions for automatic login of Linux desktop system 68
3. 4. 4. Setting method for automatic login of root user in Linux desktop system70
3. 4. 5. How to disable the desktop in the Linux desktop system 71
3. 5. Onboard LED light display description 73
3. 6. Operation instructions for the capacity of the rootfs partition of the Linux system in the TF card 74
3. 6. 1. The first boot will automatically expand the capacity of the rootfs partition in the TF card 74
3. 6. 2. The method of prohibiting the automatic expansion of the rootfs partition capacity in the TF card 76
3. 6. 3. How to manually expand the capacity of the rootfs partition in the TF card 77
3. 6. 4. Method to reduce the capacity of rootfs partition in TF card 83
3. 7. How to modify the linux log level (loglevel) 87

3. 8. Network connection test 88
3. 8. 1. Ethernet port test 88
3. 8. 2. WIFI connection test 90
3. 8. 3. How to use Hostapd to establish a WIFI hotspot 98
3. 8. 4. How to set static IP address 103
3. 8. 5. Set up the method of automatically connecting to the network when the Linux system starts up for the first time 111
3. 9. SSH remote login development board 116
3. 9. 1. SSH remote login development board under Ubuntu 116
3. 9. 2. SSH remote login development board under Windows 117
3. 10. HDMI related test items 119
3. 10. 1. HDMI Display Test 119
3. 10. 2. HDMI to VGA display test 120
3. 10. 3. Type C to HDMI display test 121
3. 11. How to use Bluetooth 122
3. 11. 1. Test method for desktop image 122
3. 11. 2. How to use the server version image 125
3. 12. USB interface test 127
3. 12. 1. USB2.0 and USB3.0 interface description 127
3. 12. 2. Connect the mouse or keyboard to test 128
3. 12. 3. Connect USB storage device test 128
3. 12. 4. USB camera test 129
3. 13. Audio Test 132
3. 13. 1. Audio test method of headphone jack 132
3. 13. 2. HDMI audio playback test method 133
3. 13. 3. Test the audio method in the desktop system 133
3. 13. 4. Onboard MIC recording test method 135
3. 13. 5. Headphone recording test method 136
3. 14. Temperature sensor 136
3. 15. How to use Mini PCIE 137
3. 15. 1. Connect SATA hard disk through mini PCIE interface 137
3. 15. 2. Connect the Gigabit Ethernet card through the mini PCIE interface 138

3. 15. 3. Connect wireless network card through mini PCIE interface 141
3. 15. 4. Connect SSD through mini PCIE interface 146
3. 16. GPU test description 150
3. 16. 1. Linux4.4 and Linux5.10 Debian system GPU test instructions 150
3. 16. 2. Ubuntu 22.04 Linux5.18 system GPU test instructions 155
3. 17. MPV hardware decoding playback video test 159
3. 18. Chromium browser hardware decoding video playback test 162
3. 19. Ubuntu22.04 Linux5.18 Kodi hard solution play video instructions 164
3. 20. 26pin interface pin description 172
3. 21. How to install wiringOP 174
3. 22. 26pin interface GPIO, I2C, UART, SPI, PWM test 175
3. 22. 1. 26pin GPIO port test 175
3. 22. 2. 26pin SPI test 176
3. 22. 3. 26pin I2C test 178
3. 22. 4. 26pin UART test 179
3. 22. 5. 26pin PWM test 183
3. 23. How to install and use wiringOP-Python 184
3. 23. 1. Installation method of wiringOP-Python 184
3. 23. 2. 26pin GPIO port test 187
3. 23. 3. 26pin SPI test 189
3. 23. 4. 26pin I2C test 190
3. 23. 5. 26pin UART test 192
3. 24. How to use 0.96-inch OLED module with I2C interface 195
3. 25. Hardware watchdog test 197
3. 26. How to check the serial number of rk3399 chip 198
3. 27. How to program linux image to eMMC 198
3. 28. How to use the OV13850 camera on Linux 4.4 202
3. 28. 1. Camera connection instructions 202
3. 28. 2. How to open a single OV13850 camera 204
3. 28. 3. How to open two OV13850 cameras at the same time 206

3.29. 10.1 inch MIPI LCD screen usage 207
3.29.1. Instructions for LCD1 interface 209
3.29.2. Instructions for the use of LCD2 interface 213
3.30. How to set dual-screen simultaneous display 219
3.31. Set up Chinese environment and install Chinese input method 220
3.31.1. Ubuntu system installation method 220
3.31.2. Installation method of Debian system 226
3.32. How to use the orange pi DS1307 RTC clock module 232
3.33. Installation method of pagoda Linux panel 237
3.34. Ubuntu22.04 method of installing browser 243
3.34.1. Ubuntu22.04 Chromium browser installation method 243
3.34.2. Installation method of Ubuntu22.04 Firefox browser 247
3.35. Partial programming language test supported by Linux system 248
3.35.1. Debian Bullseye System 248
3.35.2. Debian Buster System 250
3.35.3. Ubuntu Jammy system 252
3.35.4. Ubuntu Focal system 254
3.36. The method of remotely logging in to the Linux system desktop 256
3.36.1. Remote login using NoMachine 256
3.37. How to install Docker 265
3.38. Python related instructions 268
3.38.1. How to compile and install Python source code 268
3.38.2. How to replace pip source in Python 269
3.39. Installation method of OpenCV 270
3.39.1. Using apt to install OpenCV 270
3.40. How to install Home Assistant 270
3.40.1. Install via docker 270
3.40.2. Install via python 275
3.41. Installation method of Tensorflow 276
3.41.1. The method of using script to automatically install Tensorflow 276

3. 41. 2. Steps to manually install Tensorflow 277
3. 42. Installation method of OpenMediaVault 278
3. 42. 1. Install OpenMediaVault 5.x on Debian 10 279
3. 42. 2. Install OpenMediaVault 6.x on Debian 11 282
3. 43. ROS installation method 294
3. 43. 1. How to install ROS 1 Noetic 294
3. 43. 2. How to install ROS 2 Galactic 300
3. 44. Installation method of Pi-hole 304
3. 45. Tencent ncnn high-performance neural network forward computing framework test 311
3. 46. Installation and testing method of face_recognition face recognition library 320
3. 46. 1. Automatic installation of face_recognition using script 320
3. 46. 2. Manual installation of face_recognition 321
3. 46. 3. Test method of face_recognition 323
3. 47. QT installation method 334
3. 47. 1. How to install QT5 334
3. 47. 2. How to install QT Creator 337
3. 48. Reset and shutdown methods 347
4. Instructions for use of Android system 348
4. 1. Supported Android Versions 348
4. 2. Android 8.1 function adaptation 348
4. 3. On-board LED light display description 349
4. 4. How to use ADB 349
4. 4. 1. Use the data cable to connect adb debugging 349
4. 4. 2. Using network connection adb debugging 350
4. 5. How to use OV13850 camera 351
4. 6. How to use the HDMI interface 354
4. 7. The method of displaying the system interface through the TypeC interface ... 355
4. 8. 10.1 inch MIPI screen usage 355

4. 9. How to use the USB camera 360

4. 10. How to use Mini PCIE 362

4. 10. 1. Instructions for connecting Mini PCIE to SATA hard disk 362

5. Instructions for using the Linux SDK 367

5. 1. Compilation system requirements 367

5. 2. Get the source code of linux sdk 369

5. 2. 1. Download orangepi-build from github 369

5. 2. 2. Download the cross-compilation toolchain 371

5. 2. 3. Orangepi-build complete directory structure description 372

5. 2. 4. Download orangepi-build from Google Cloud Disk 374

5. 3. Compile u-boot 376

5. 4. Compile the linux kernel 381

5. 5. Compile rootfs 386

5. 6. Compile the linux image 391

6. Instructions for using Android SDK 395

6. 1. Download the source code of Android SDK 395

6. 2. Build Android Compilation Environment 396

6. 3. Compile Android image 397

6. 3. 1. Compiling u-boot 397

6. 3. 2. Compile the kernel 397

6. 3. 3. Compile android 398

6. 3. 4. Packaging the full image 398

Version update history

Version	Date	Update Instructions
v1.7	2022-05-31	1. The linux command format description in this manual 2. Set up the method of automatic login of Linux system terminal 3. Setting method for automatic login of root user in Linux desktop system 4. How to disable the desktop in the Linux desktop version system 5. Operation instructions for the capacity of the rootfs partition of the Linux system in the TF card 6. Linux: How to create a WIFI hotspot using Hostapd 7. Linux: How to set static IP address using nmcli command 8. Linux: How to set up the method of automatically connecting to the network when the Linux system starts up for the first time 9. Test Audio Method in Desktop Linux System 10. Support Ubuntu 22.04 Linux5.17 system 11. Ubuntu 22.04 Linux5.17 system GPU test instructions 12. Ubuntu22.04 method of installing browser 13. Add a description of the linux command format in this manual 14. Add some precautions 15. Linux: How to compile Python source code and how to replace pip source 16. Installation method of OpenCV 17. Installation method of Tensorflow 18. How to install Home Assistant 19. Debian: How to install OpenMediaVault 5.x and 6.x 20. Linux: How to install Pi-hole 21. Linux: Tencent ncnn neural network forward computing framework test method 22. Ubuntu 20.04 ROS1 and ROS2 installation method 23. How to install QT 24. face_recognition face recognition library installation and

		testing methods
v1.8	2022-06-29	1. How to install and use wiringOP-Python
v1.9	2022-07-07	1. Support Ubuntu 22.04 Linux5.18 system 2. Ubuntu22.04 Linux5.18 Kodi hard solution playback video instructions
v2.0	2022-07-29	1. orangepi-build next branch: support on Ubuntu22.04 PC
v2.1	2022-08-01	1. Linux and Android: Support AFJ101BA2131 model MIPI screen

1. Basic Features of Orange Pi 4 LTS

1. 1. What is Orange Pi 4 LTS？

Orange Pi is an open source single-board card computer, a new generation of arm64 development boards, which can run operating systems such as Android 8.1, Ubuntu and Debian. The Orange Pi 4 LTS development board can be divided into the following 4 models according to the SOC used and whether it has an eMMC chip:

Orange Pi 4 LTS Model 1	Use RK3399 chip	with 16GB eMMC	4GB LPDDR4
Orange Pi 4 LTS Model 2	Use RK3399 chip	Without 16GB eMMC	4GB LPDDR4
Orange Pi 4 LTS Model 3	Use RK3399-T chip	with 16GB eMMC	3GB LPDDR4
Orange Pi 4 LTS Model 4	Use RK3399-T chip	Without 16GB eMMC	3GB LPDDR4

1. 2. Purpose of Orange Pi 4 LTS

we can use it to build：

- A computer
- A wireless server
- Games
- HD video player
- Android

Of course there are many more features as Orange Pi is open source.

1. 3. Who is Orange Pi 4 LTS designed for?

The Orange Pi development board is not just a consumer product, it is designed for anyone who wants to use technology to create and innovate. It's a very simple, fun, and useful tool that you can use to shape the world around you.

1. 4. Hardware Features of Orange Pi 4 LTS

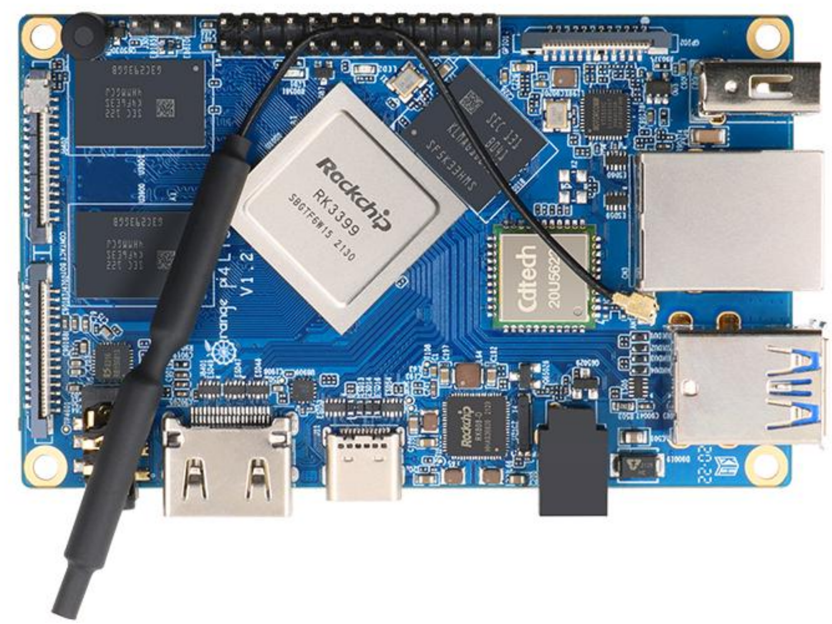
Hardware Features Introduction	
CPU	• Rockchip RK3399 or RK3399-T • The main frequency of RK3399 is up to 1.8GHz • The main frequency of RK3399-T is up to 1.6GHz • 6-core ARM® 64-bit processor • Based on big.LITTLE size core architecture: Dual-core Cortex-A72 (large core) + quad-core Cortex-A53 (small core)
GPU	• High-performance multi-core GPU Mali T864 • OpenGL ES 1.1/2.0/3.0 • OpenCL 1.0/1.1/1.2 • DirectX 9/11.1
Memory	3GB LPDDR4 or 4GB LPDDR4
Onboard storage	• 16GB eMMC or empty sticker • TF card slot
Ethernet	10/100/1000Mbps Ethernet (YT8531C)
WIFI+Bluetooth	• 20U5622, IEEE 802.11 a/b/g/n/ac • BT5.0
video output	HDMI 2.0 (Type-A), support 4K@60 frame output DP 1.2 (DisplayPort) , support 4K@60 frame output Supports dual channel MIPI-DSI (4 wires per channel)
video input	MIPI-CSI x2 camera interface (MIPI_RX0, MIPI_TX1/RX1)
Audio output	• 3.5mm Headphone plug • HDMI
Audio input	• Onboard MIC • Headphone recording
power supply	• DC 5V/4A Or DC 5V/3A • TYPE-C 5V/4A

USB port	• USB2.0 HOST x 2 • USB3.0 HOST x 1 • USB3.0 Type-C x 1
26pin pin header	with I2Cx2, SPIx1 or UARTx1 and multiple GPIOs
Mini-PCIE	24pin mini-PCIE interface
Debug serial port	UART-TX、UART-RX and GND
LED lights	Power indicator and status indicator
button	Reset button x1, upgrade button x1
Supported Operating Systems	Android8.1、Ubuntu、Debian etc. operating system
Appearance specification introduction	
Product Size	91mm×56mm
range Pi™ is the registered trademark of Shenzhen Xunlong Software Co., Ltd.	

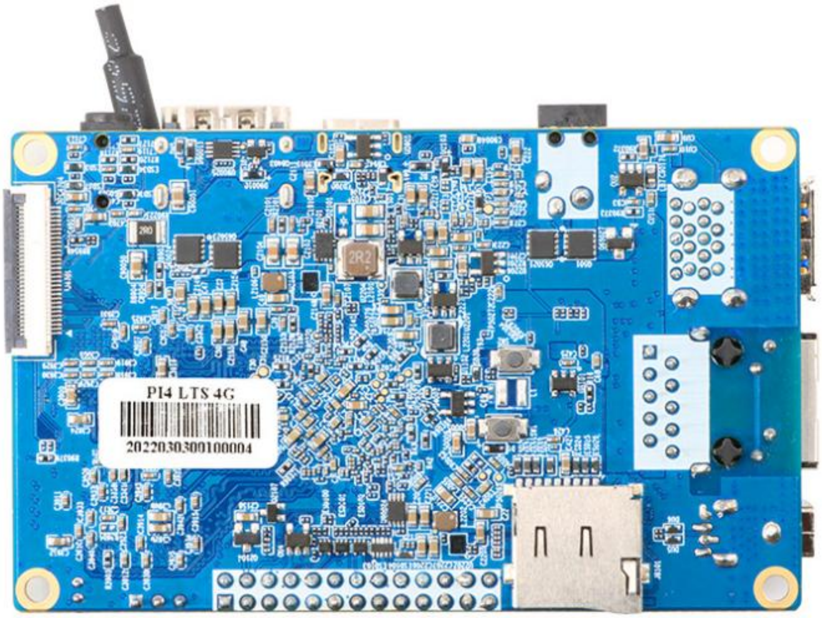
1.5. Top view and bottom view of Orange Pi 4 LTS

1.5.1. RK3399+4GB LPDDR4 with 16GB eMMC version

top view:



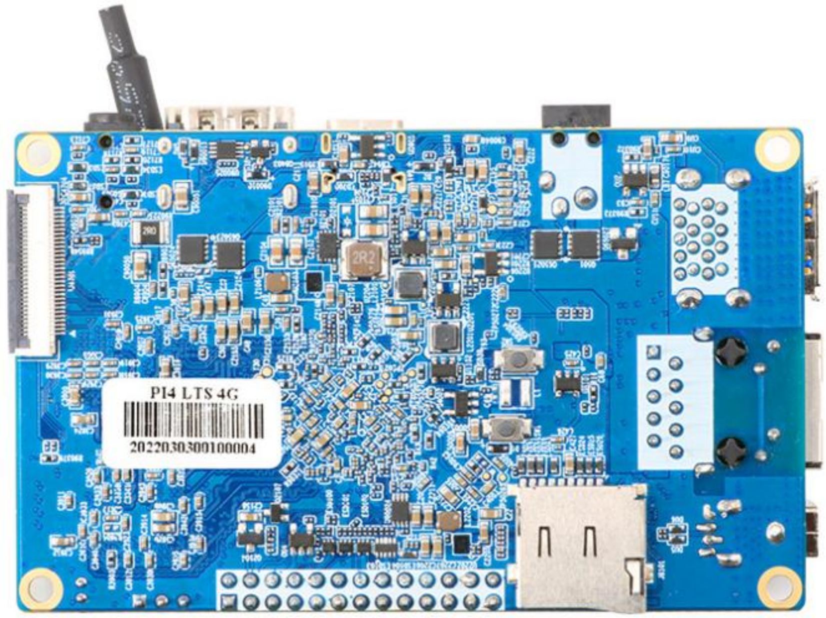
bottom view:



1.5.2. RK3399+4GB LPDDR4 without 16GB eMMC version

top view:

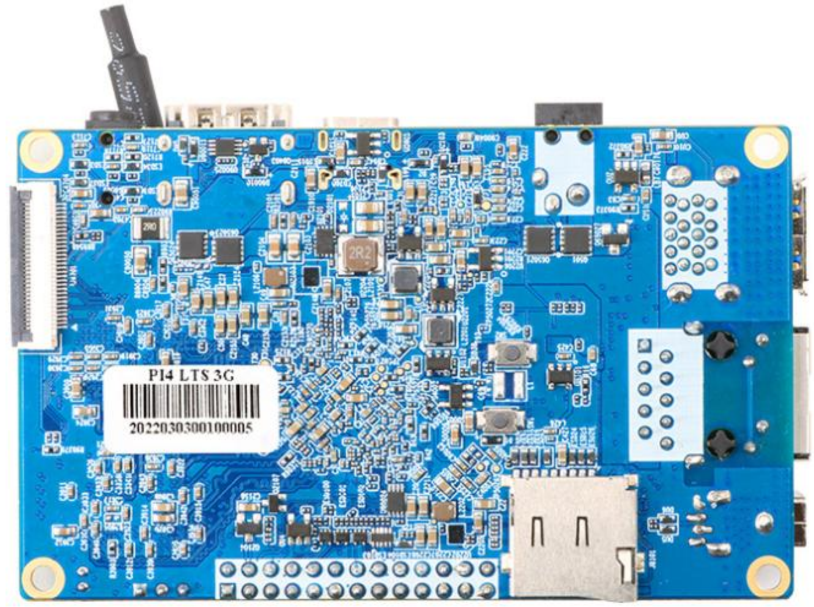
bottom view:



1. 5. 3. RK3399-T+3GB LPDDR4 with 16GB eMMC version

top view:

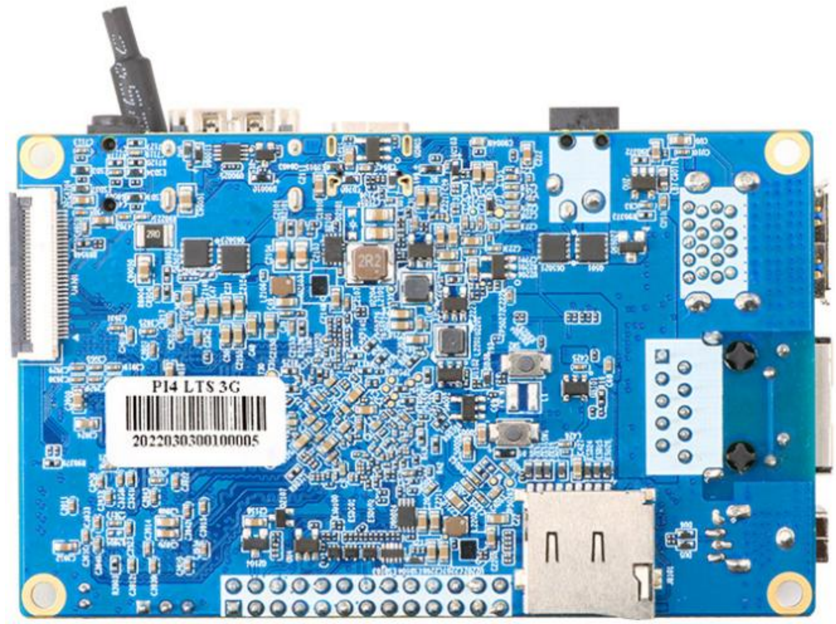
bottom view:



1.5.4. RK3399-T+3GB LPDDR4 without 16GB eMMC version

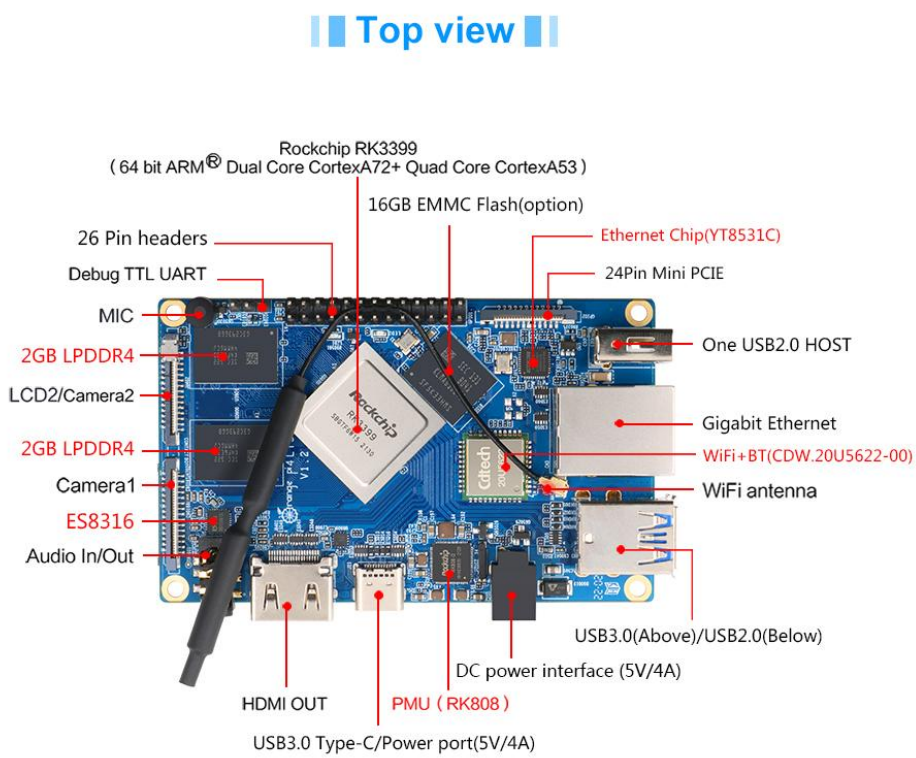
top view:

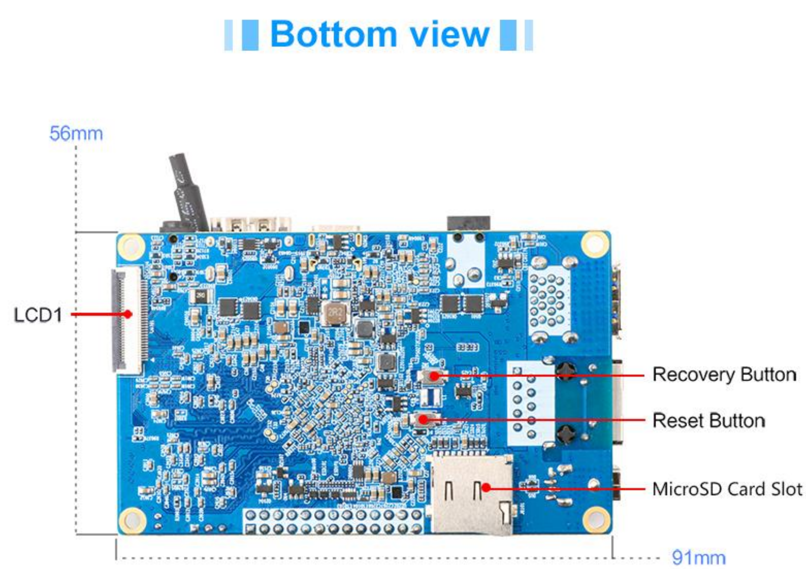
bottom view:


PI4 LTS 3G
20220303001000005

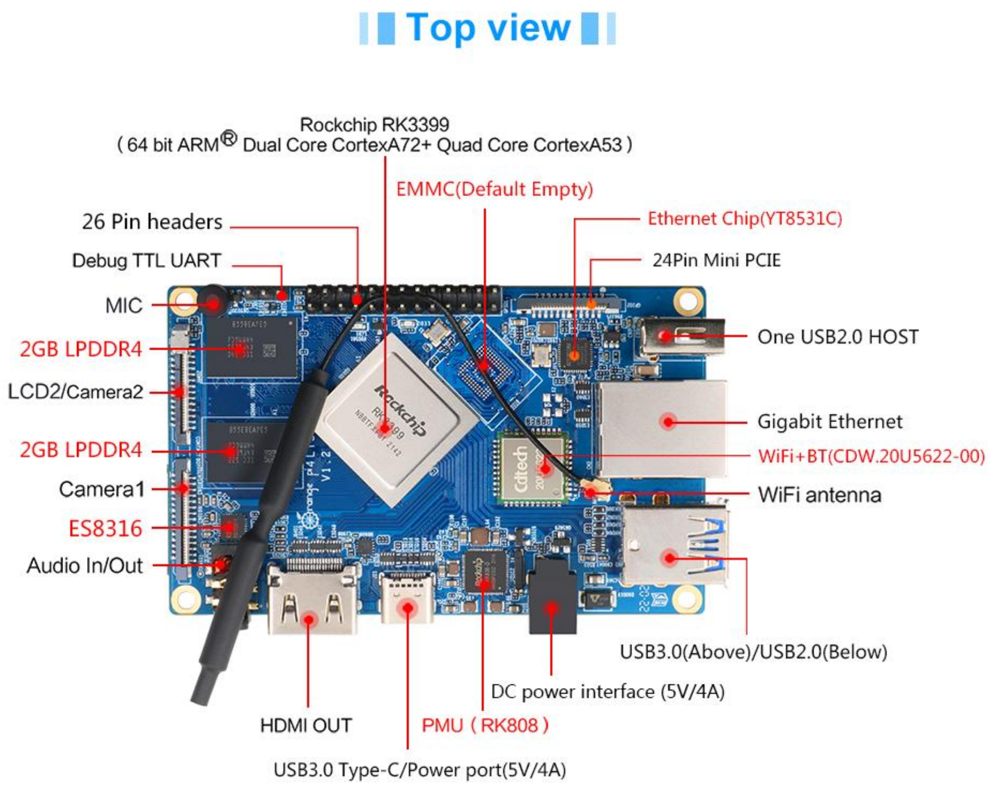
1.6. Interface details of Orange Pi 4 LTS

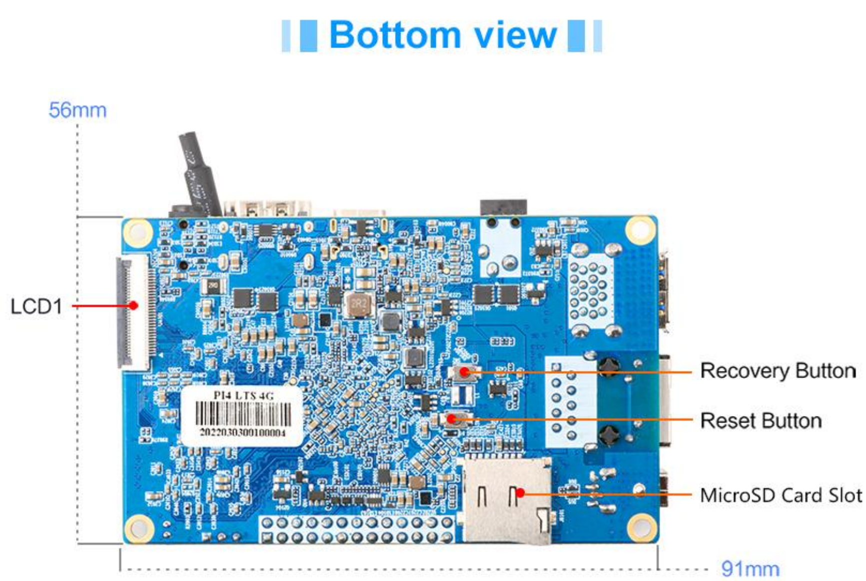
1.6.1. RK3399+4GB LPDDR4 with 16GB eMMC version



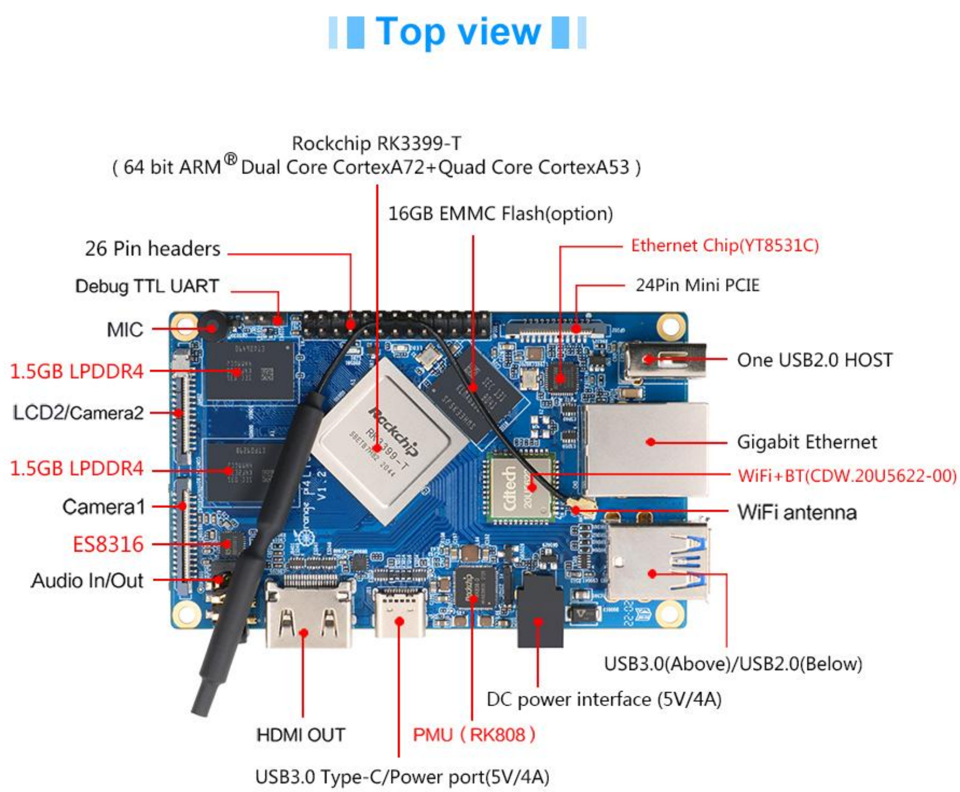


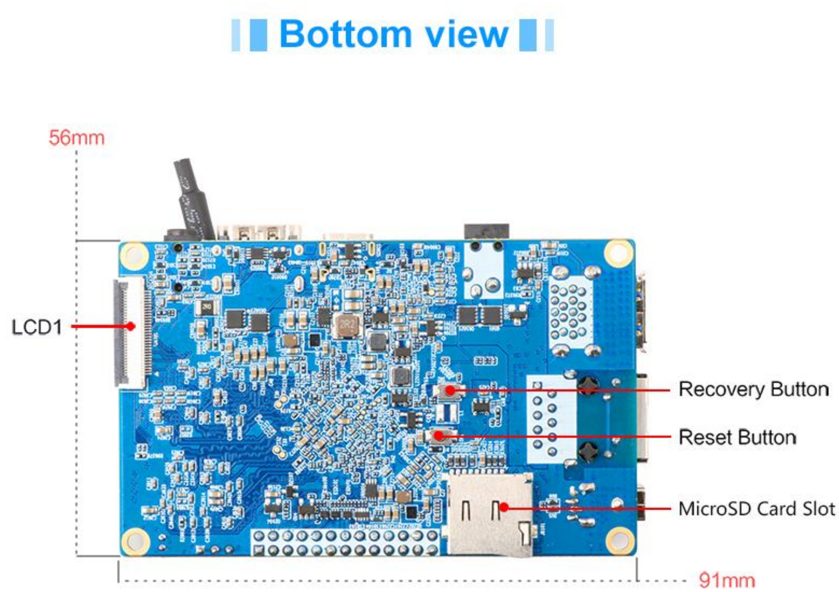
1.6.2. K3399+4GB LPDDR4 without 16GB eMMC version





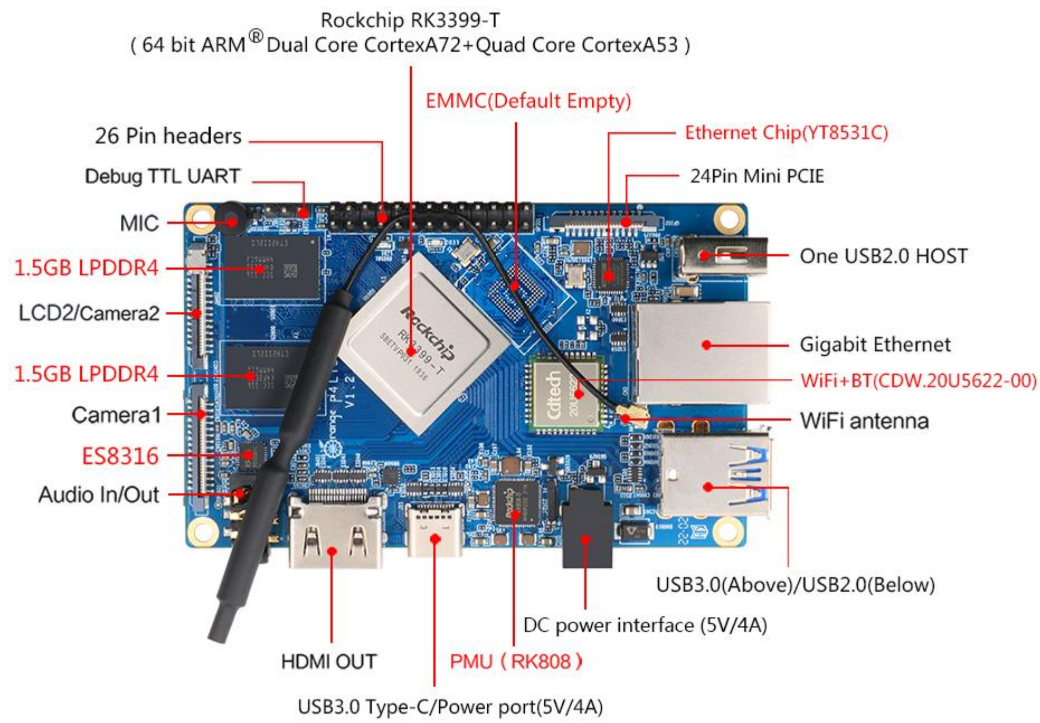
1.6.3. RK3399-T+3GB LPDDR4 with 16GB eMMC version



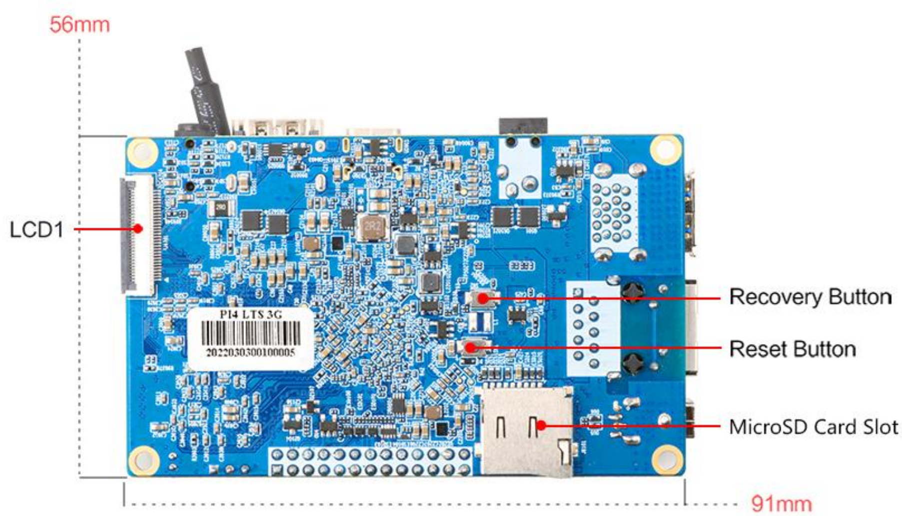


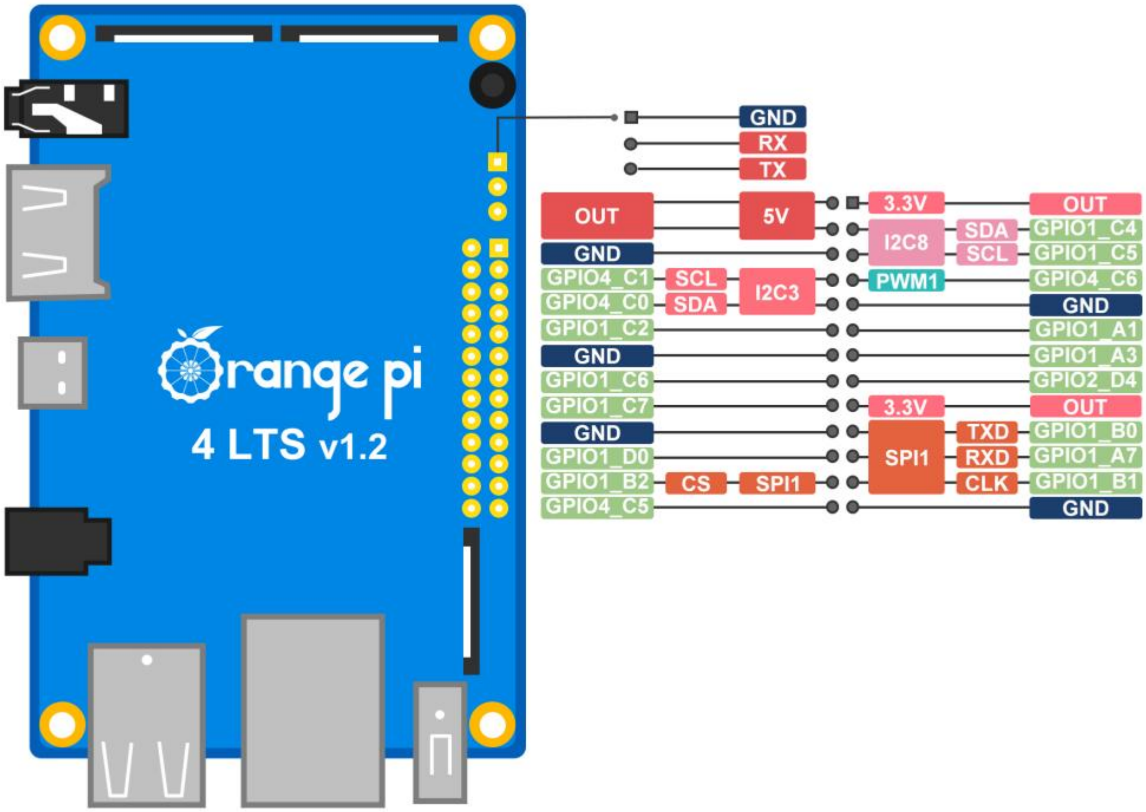
1.6.4. RK3399-T+3GB LPDDR4 without 16GB eMMC version

Top view



Bottom view





The diameters of the four positioning holes are all 3.0mm.

2. Introduction to the use of the development board

2. 1. Prepare the necessary accessories

1) TF card, class10 or above high-speed card with a minimum capacity of 8GB, it is recommended to use SanDisk TF card, Orange Pi test is to use SanDisk TF card, other brands of TF card may cause the problem that the system cannot be started

2) TF card reader, used to read and write TF card

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

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

5) Type-C to USB3.0 adapter, used to connect USB3.0 storage devices or USB devices such as mouse and keyboard

6) 10.1-inch MIPI screen, used to display the system interface of the development board

7) Power adapter, Orange Pi 4 LTS supports 5V/4A or 5V/3A DC power supply, and also supports 5V/4A Type-C power supply

Note, do not use 5V/3A Type-C power supply, it may cause system instability due to insufficient power supply, and Orange Pi 4 LTS cannot be powered through the 5v pin on the 26pin interface.

8) USB interface mouse and keyboard, as long as it is a standard USB interface mouse and keyboard, the mouse and keyboard can be used to control the Orange Pi development board

9) Metal heat dissipation shell, Orange Pi 4 LTS matching metal shell is shown in the figure below, please refer to the instructions in the following section for the installation method of metal heat dissipation shell,Since the RK3399/RK3399-T chip generates a relatively large amount of heat, it is necessary to have a metal case

Note that it is not possible to put another fan inside the metal case.

10) If you don't buy a metal cooling case, it is strongly recommended to add a 5V cooling fan. As shown in the figure below, there are 5V and GND pins on the 26pin interface of the development board that can be connected to the cooling fan. The spacing between the 26pin pin headers is 2.54mm. The power interface of the cooling fan can be purchased from Taobao according to this specification.

Note that after the development board is plugged into the power supply, the 5V pin can be used directly without other settings, and the voltage output by the 5V pin cannot be adjusted and turned off by software.

11) Fast or Gigabit Ethernet cable to connect the development board to the Internet

12) OV13850 13MP camera, dedicated camera for Orange Pi 4 LTS, compatible with MIPI interface

13) **3.3V** USB to TTL module and DuPont cable, when using the serial port debugging function, USB to TTL module and DuPont cable are required to connect the development board and computer

14) A PC with Ubuntu and Windows operating systems installed

1	Ubuntu14.04 PC	Optional, used to compile Android source code
2	Ubuntu22.04 PC	Optional, used to compile Linux source code
3	Windows PC	For burning Android and Linux images

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

1) The download website of the Chinese version of the material is:

http://www.orangepi.cn/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-4-LTS.html

2) The download URL of the English version of the material is:

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-pi-4-LTS.html

3) The information mainly includes
   a. **Android source code**: saved on Google network disk
   b. **Linux source code:** save on github
   c. **User manual and schematic diagram**: The data sheet related to the chip will also be placed here
   d. **Official tools**: mainly include the software that needs to be used during the use of the development board
   e. **Android image**: save on Google network disk
   f. **Ubuntu image**: save on Google network disk
   g. **Debian image**: save on Google network disk

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

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

2. 3. 1. How to use Win32Diskimager to burn Linux image

1) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the TF card must be **class10** or above. It is recommended to use a TF card from a brand such as SanDisk

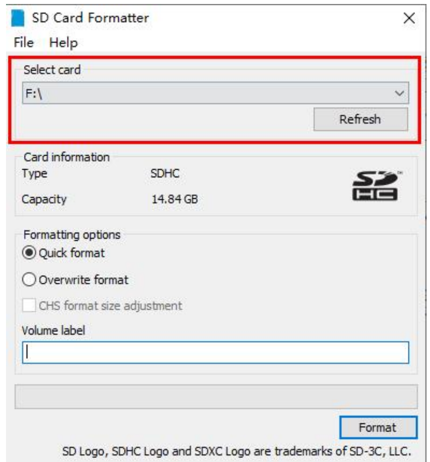
2) Then use the card reader to insert the TF card into the computer

3) Then format the TF card
   a. The **SD Card Formatter** software can be used to format the TF card, and its download address is

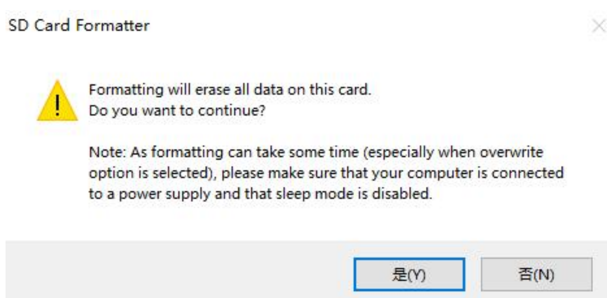
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

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

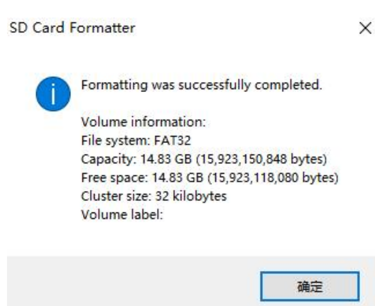
   c. If only the TF card is inserted into the computer, the "**Select card**" column will display the drive letter of the TF card. If multiple USB storage devices are inserted into the computer, you can select the drive letter corresponding to the TF card through the drop-down box.



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



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



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

image file of the operating system. The size is generally more than 1GB

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

a. The download page of Win32Diskimager is

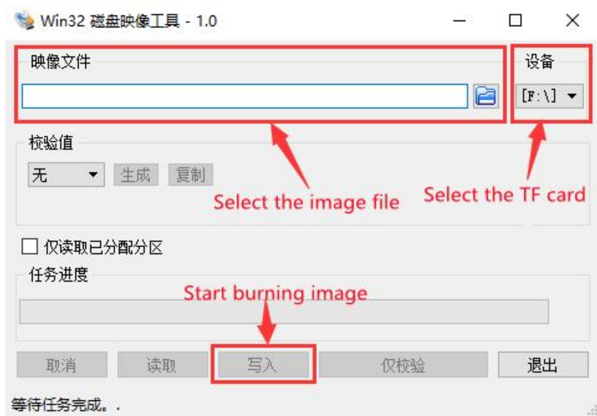
http://sourceforge.net/projects/win32diskimager/files/Archive/

b. After downloading, install it directly. The Win32Diskimager interface 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 the one 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. 2. How to use balenaEtcher to burn a Linux image

1) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the TF card must be **class10** or above. It is recommended to use a TF card from a brand such as SanDisk

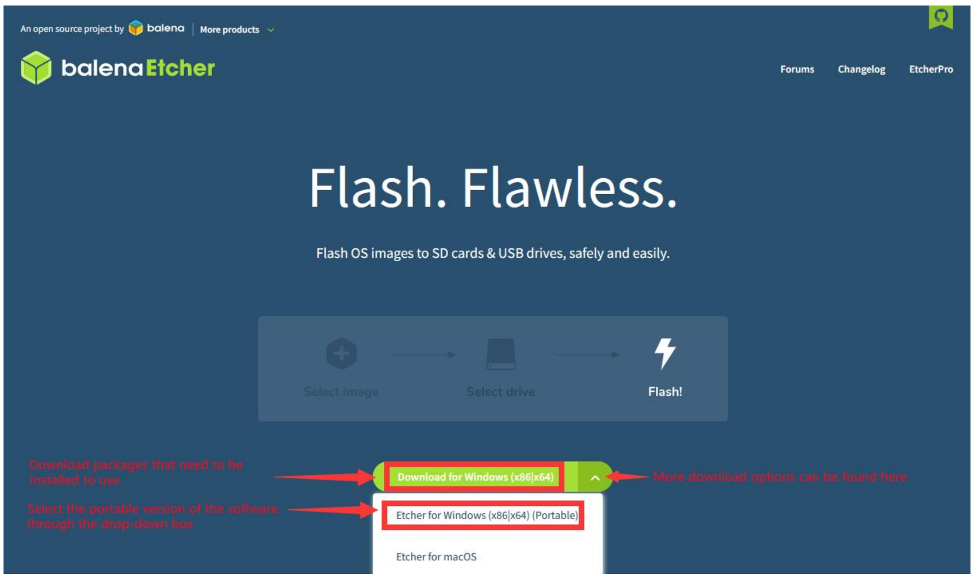
2) Then use the card reader to insert the TF card into the computer

3) Download the compressed package of the Linux operating system image file you want to burn from **the data download page of Orange Pi**, and then use the decompression software to decompress it. In the decompressed file, the file ending with "**.img**" is the image file of the operating system. The size is generally more than 1GB

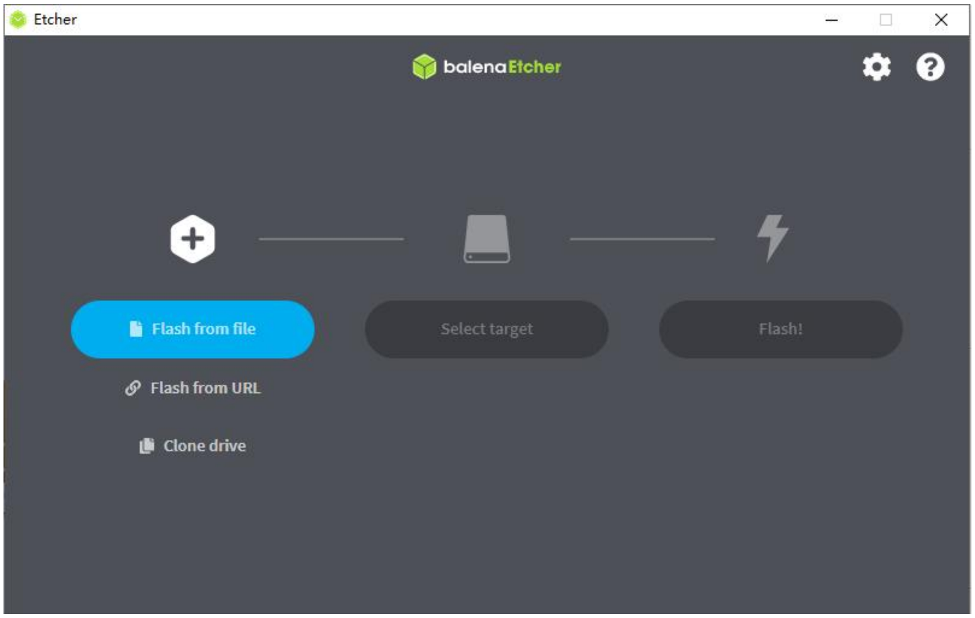
4) Then download the burning software of the Linux image - **balenaEtcher,** the download address is

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

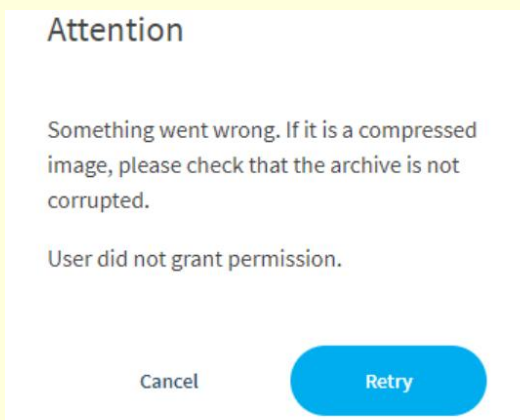
5) After entering the balenaEtcher download page, click the green download button to download the installation package of balenaEtcher. You can also select the Portable version of balenaEtcher through the drop-down box. The Portable version does not need to be installed. Double-click to open it and use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



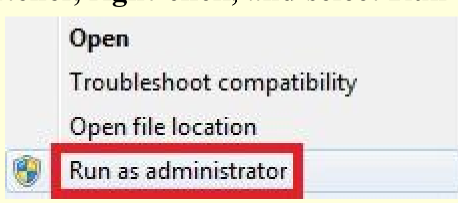
6) If you download a version of balenaEtcher that needs to be installed, please install it before using it. If you download the Portable version of balenaEtcher, just double-click to open it. The opened balenaEtcher interface is shown in the figure below.



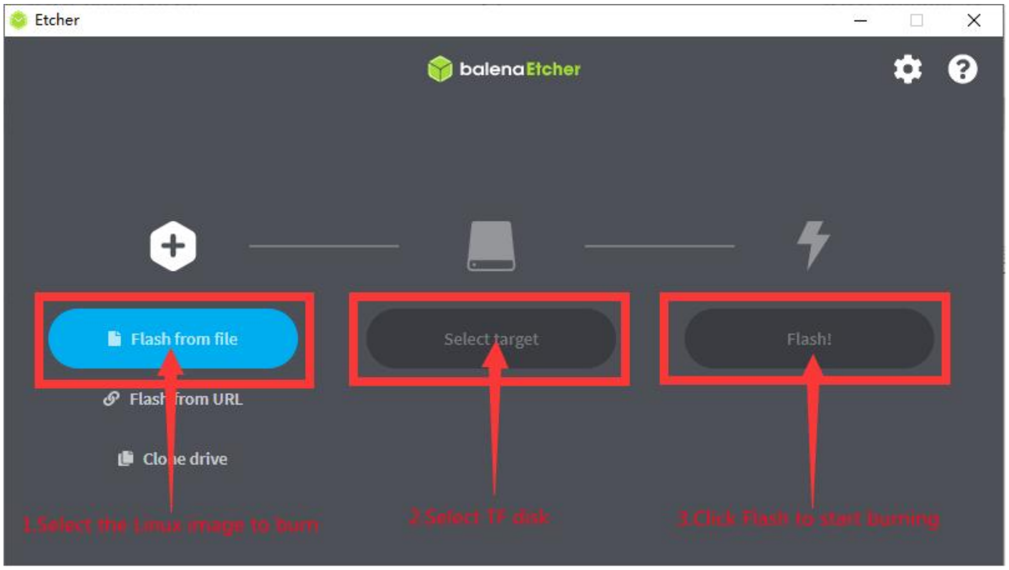
If the following error is displayed when opening balenaEtcher:



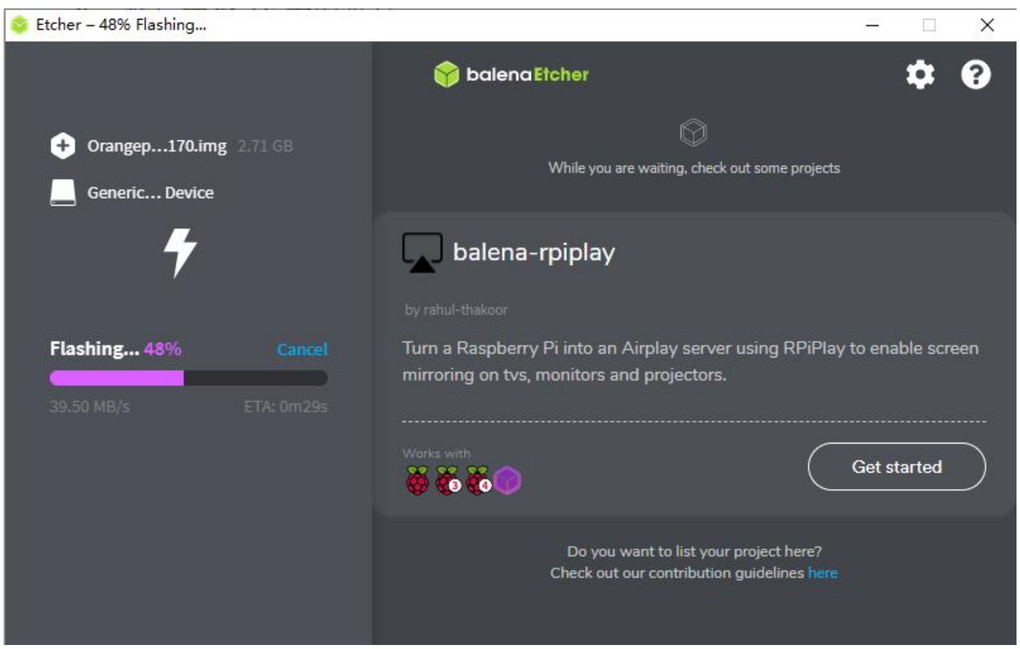
Please select balenaEtcher, right-click, and select Run as administrator.



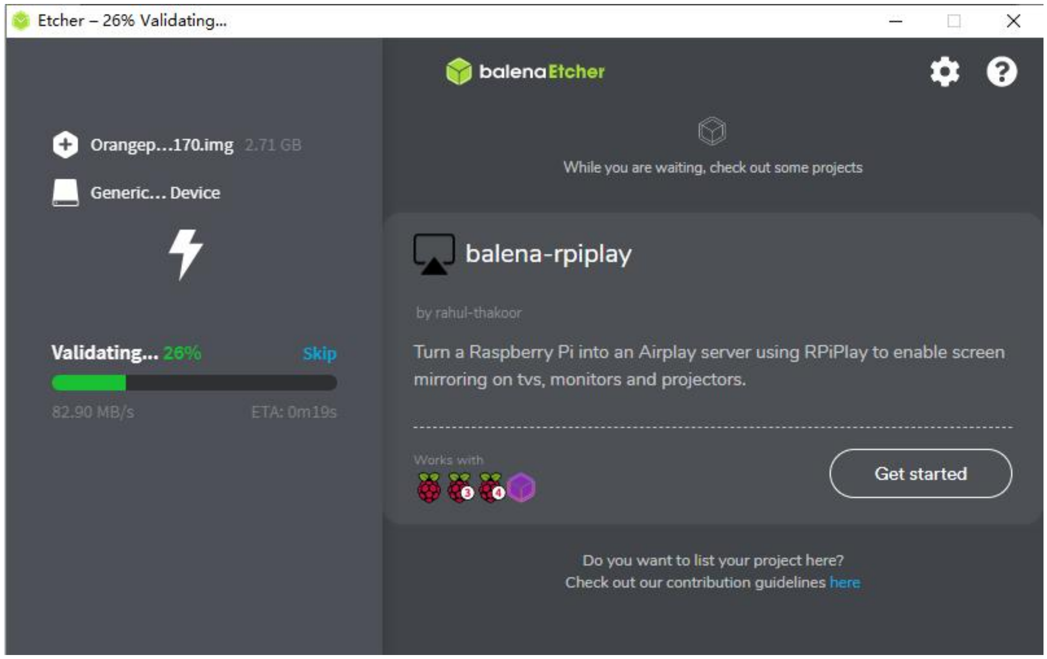
7) The specific steps to use balenaEtcher to burn a Linux image are as follows
   a. First select the path of the Linux image file to be burned
   b. Then select the drive letter of the TF card
   c. Finally, click Flash to start burning the Linux image to the TF card



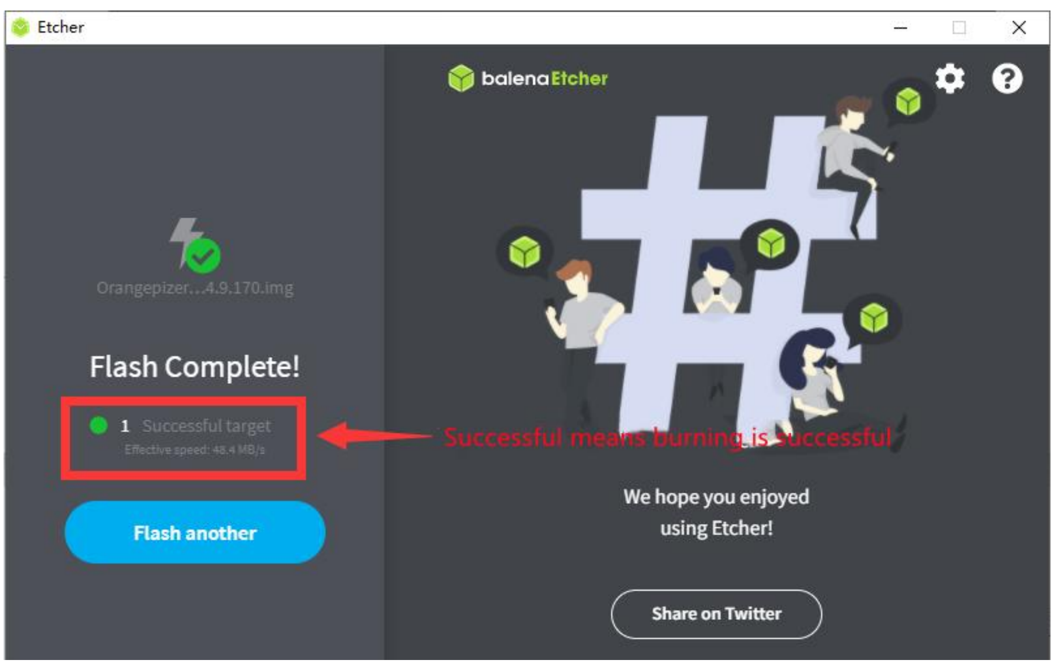
8) The interface displayed in the process of balenaEtcher burning the Linux image is shown in the figure below. In addition, the progress bar shows purple to indicate that the Linux image is being burned to the TF card.



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



10) After the successful burning, the display interface of balenaEtcher is shown in the figure below. If the 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.



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

Note that the Linux image mentioned here specifically refers to a Linux distribution image such as Debian or Ubuntu downloaded from the Orange Pi data download page, and Ubuntu PC refers to a personal computer or virtual machine with Ubuntu installed.

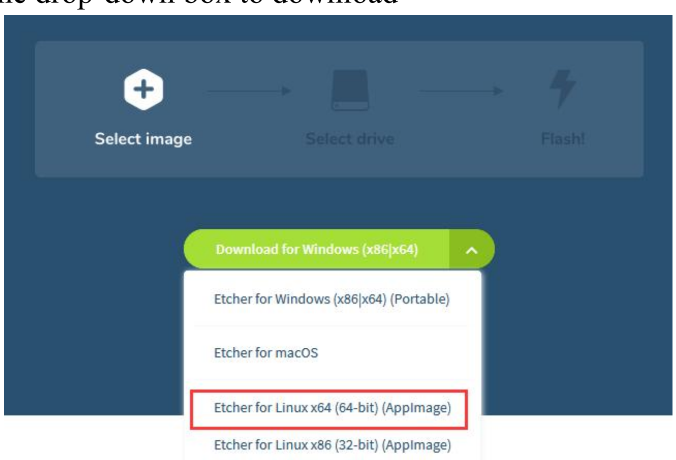
1) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card from a brand such as SanDisk

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

3) Download balenaEtcher software, the download address is

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

4) After entering the balenaEtcher download page, please select the Linux version of the software through the drop-down box to download



5) After downloading, please use the unzip command to decompress the downloaded compressed package. The decompressed **balenaEtcher-1.5.109-x64.AppImage** is the software needed to burn the Linux image

```
test@test:~$ unzip balena-etcher-electron-1.5.109-linux-x64.zip
Archive:    balena-etcher-electron-1.5.109-linux-x64.zip
```

```
  inflating: balenaEtcher-1.5.109-x64.AppImage
test@test:~$ ls
balenaEtcher-1.5.109-x64.AppImage    balena-etcher-electron-1.5.109-linux-x64.zip
```

6) Download the compressed package of the Linux operating system image file you want to burn from the data download page of Orange Pi, and then use the decompression software to decompress it. In the decompressed file, the file ending with "**.img**" is the image file of the operating system. The size is generally more than 1GB

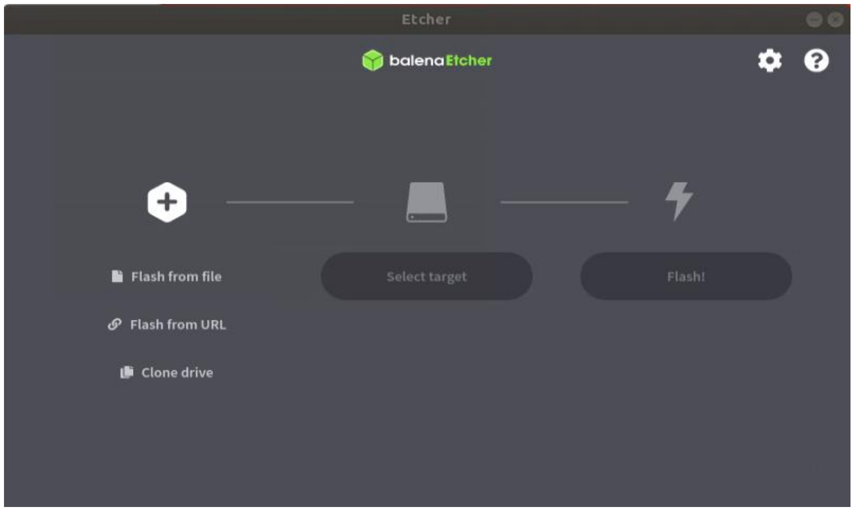
7zThe decompression command for the compressed package at the end is as follows

```
test@test:~$ 7z x Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43.7z
test@test:~$ ls Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43.*
Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43.7z
Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43.sha    #checksum file
Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43.img    #image file
```

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

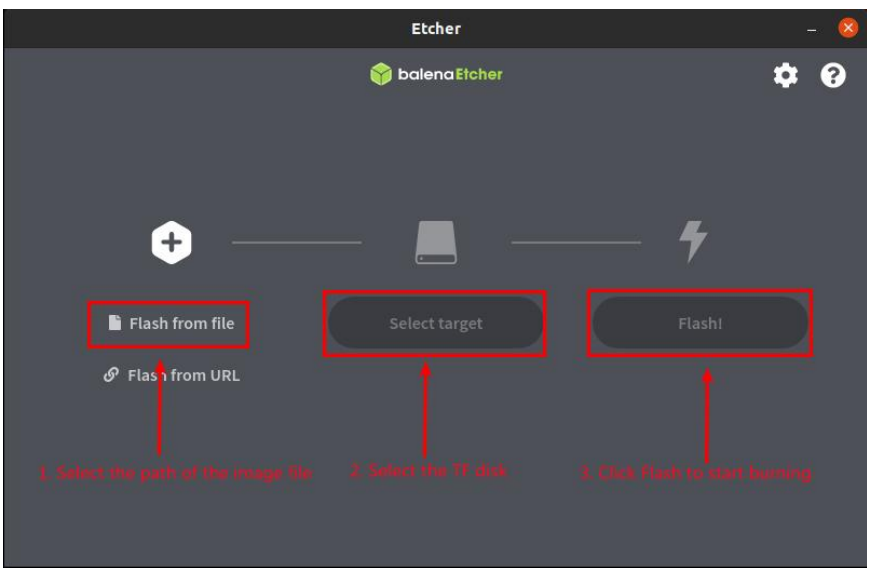
```
test@test:~$ sha256sum -c *.sha
Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43.img: success
```

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 following figure

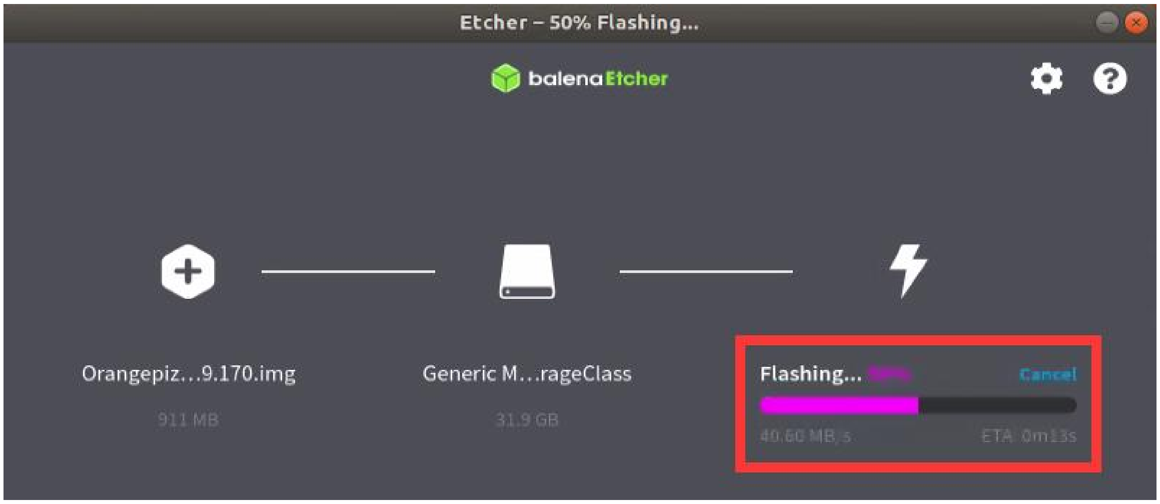


9) The specific steps to use balenaEtcher to burn a Linux image are as follows

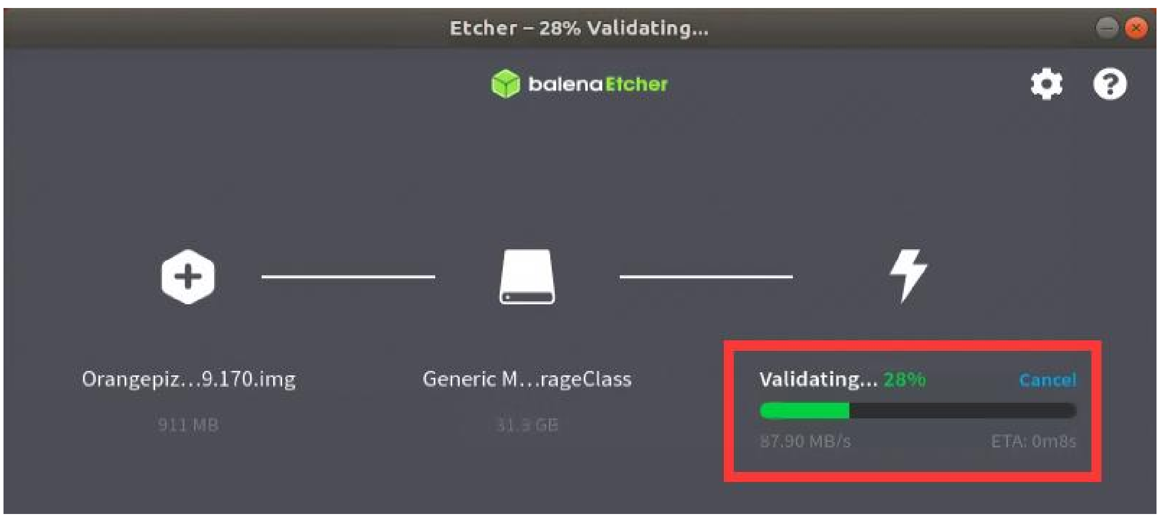
   a. First select the path of the Linux image file to be burned
   b. Then select the drive letter of the TF card
   c. Finally, click Flash to start burning the Linux image to the TF card



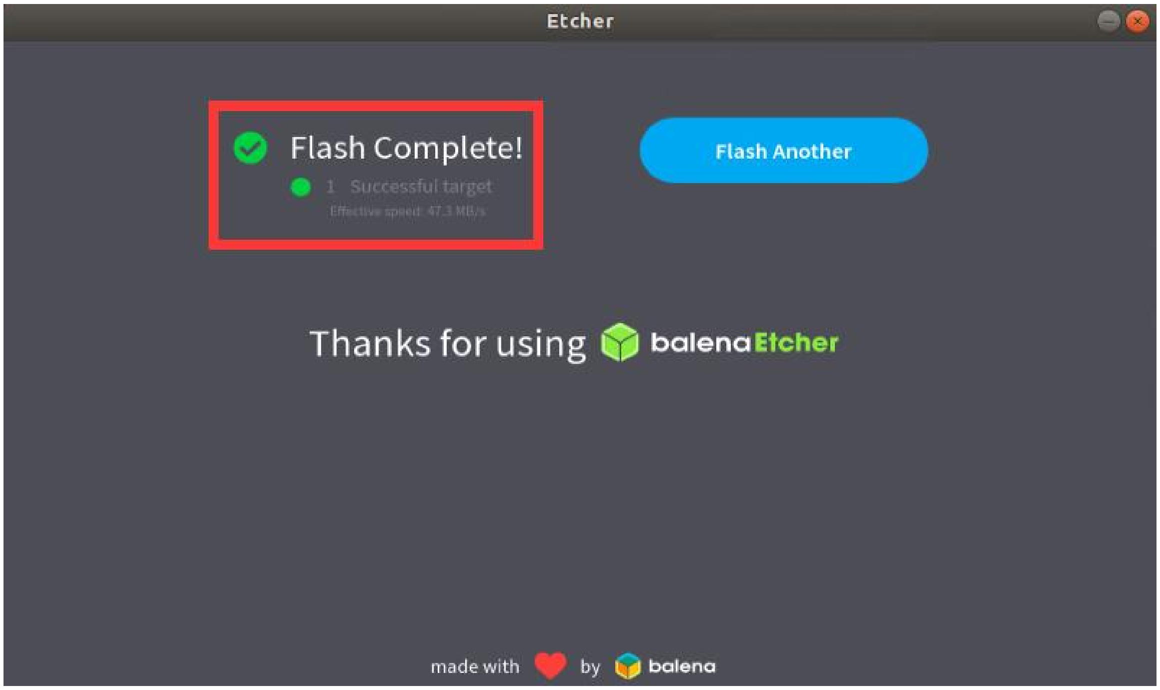
10) The interface displayed in the process of balenaEtcher burning the Linux image is shown in the figure below. In addition, the progress bar shows purple to indicate that the Linux image is being burned to the TF card.



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



12) After the successful burning, the display interface of balenaEtcher is shown in the figure below. If the 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



2. 5. How to program Linux image to eMMC

See the method of burning linux image to EMMC

The blue part above is a hyperlink, click to jump to the corresponding position of the document. In addition, only the purchased Orange Pi 4 LTS with 16GB eMMC chip can burn the Linux image to the eMMC.

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

The Android firmware of the development board can only be burned to the TF card using the SDDiskTool software under the Windows platform. In addition, the SDDiskTool software does not have a Linux platform version, so it is impossible to burn the Android system to the TF card under the Linux platform.

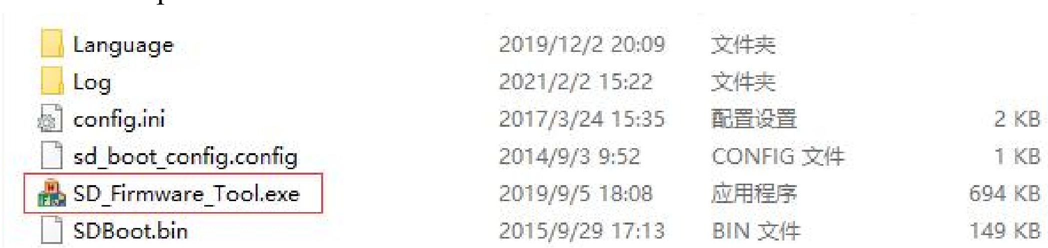
1) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the TF card must be **class10** or above. It is recommended to use a TF card from a brand such as SanDisk

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

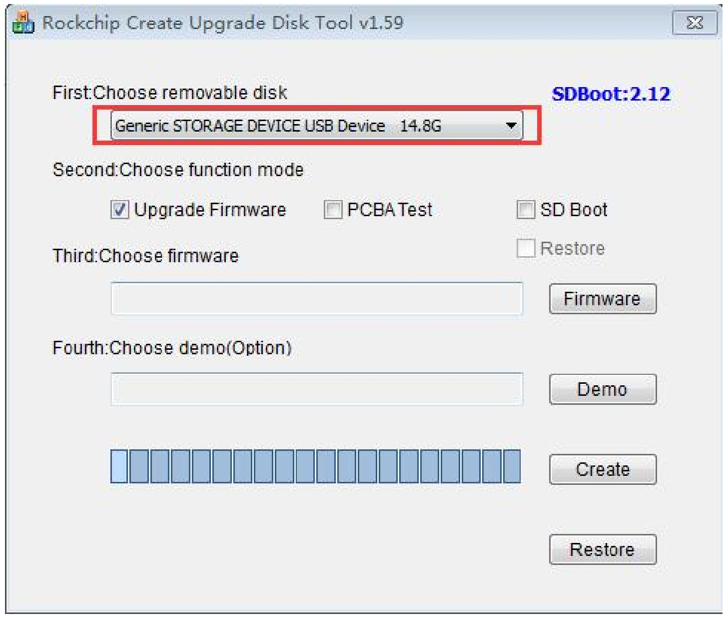
3) Then download the SDDiskTool programming tool and Android firmware from **the data download page of Orange Pi**. The Android firmware on the official website has two versions: eMMC boot and TF card boot. Here you need to select the TF card boot version, and pay attention to the TF card boot version. The name of the Android firmware contains the word "SD"

4) Then use the decompression software to decompress the downloaded Android firmware compressed package. In the decompressed file, the file ending with ".img" is the Android firmware

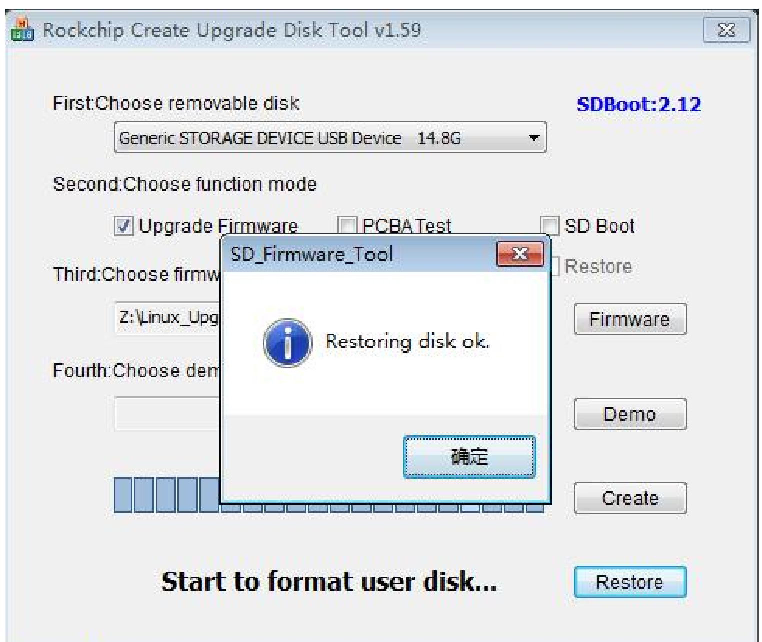
5) Then use the decompression software to decompress **SDDiskTool_v1.59.zip**, this software does not need to be installed, find **SD_Firmware_Tool.exe** in the decompressed folder and open it



6) 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 the same as the drive letter of the TF card you want to burn**. if there is no display, you can try to unplug the TF card



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



8) Then start writing Android image to TF card

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

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

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

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

2. 7. The method of burning Android image to eMMC based on Windows PC

Orange Pi 4 LTS has three upgrade modes, namely MaskRom mode, Loader mode and SD upgrade mode. The first two modes need to be burned through the Type C cable, and the latter mode is burned through the TF card. For the way of programming through Type C cable, if there is no programming system in eMMC, it will enter MaskRom mode by default. If a bootable system has been programmed in eMMC, you can enter Loader mode for programming.

In addition, it should be noted that if the U-boot is damaged due to an accident or other reasons in the system in eMMC, it will not be able to enter the Loader mode for programming. You need to enter the MaskRom mode according to the method of entering the MaskRom mode, and then use the TypeC cable. Connect the computer and the development board for programming.

Using TF card to burn Android image to eMMC is the easiest method. If you

just got the development board, please try to burn with this method first, and then study the method of burning with TypeC cable.

2. 7. 1. Burn Android firmware to eMMC via TF card

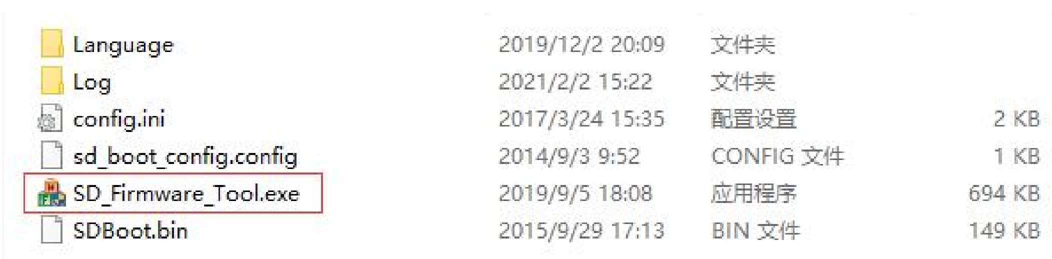
1) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the TF card must be class10 or above. It is recommended to use a TF card from a brand such as SanDisk

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

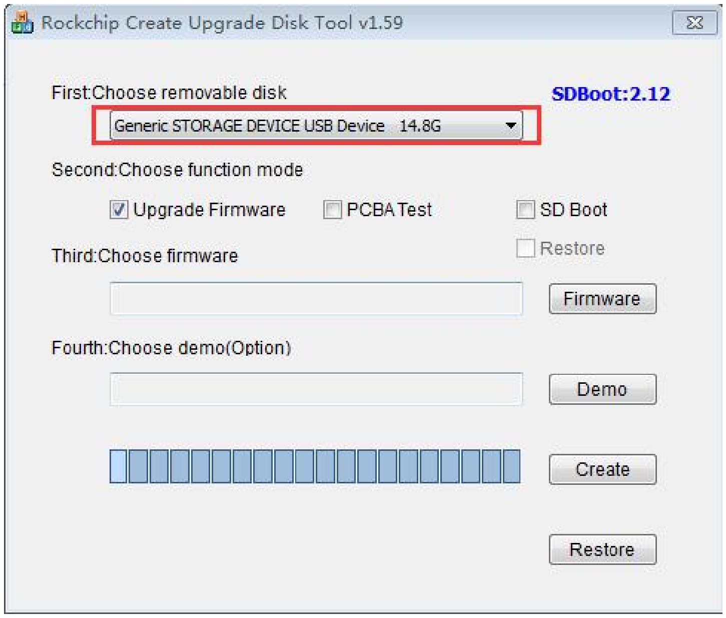
3) Download the Android firmware and SDDiskTool programming tool from **the data download page of Orange Pi**. It should be noted that the firmware on the official website has two versions: eMMC boot and TF card boot. Here you need to select the eMMC boot version, and pay attention to support eMMC boot The firmware does not contain the word "SD", and **please ensure that the version of SDDiskTool is v1.59**

4) Then use the decompression software to decompress the downloaded Android firmware compressed package. In the decompressed file, the file ending with "**.img**" is the Android firmware

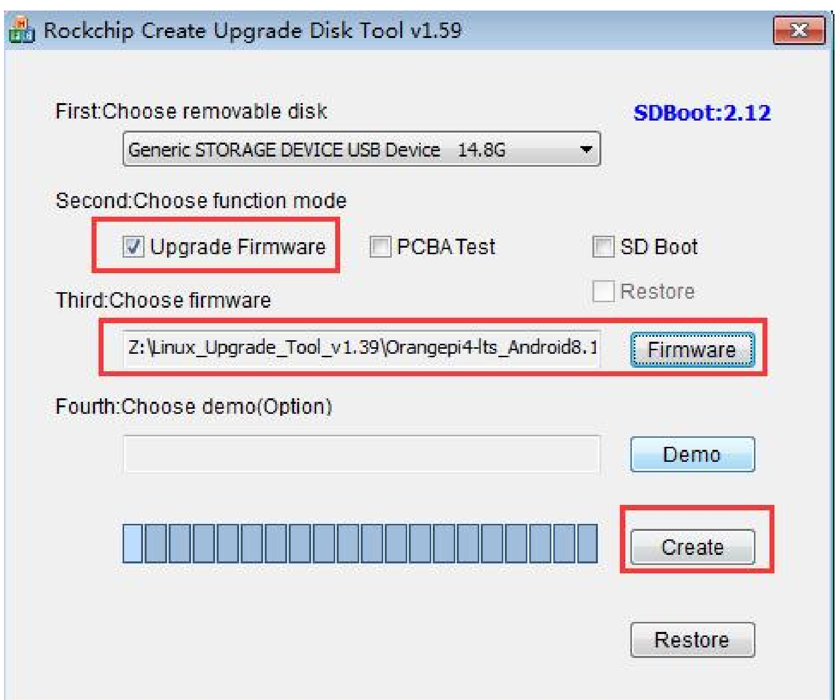
5) Use the decompression software to decompress **SDDiskTool_v1.59.zip**, this software does not need to be installed, just find **SD_Firmware_Tool.exe** in the decompressed folder and open it



6) After opening **SDDiskTool**, if the TF card is recognized normally, the inserted disk device will be displayed in "Select Removable Disk Device". **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

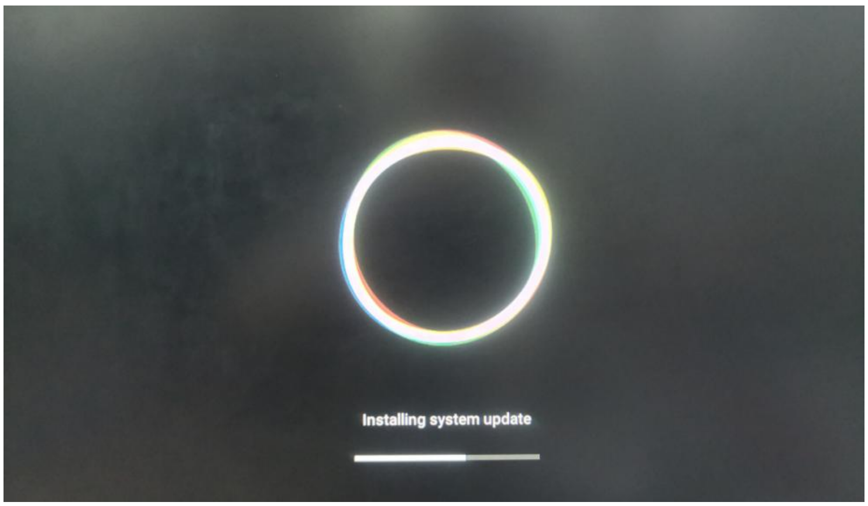


7) Then start writing Android firmware to TF card
   a. First, confirm that the drive letter displayed under "**Select Removable Disk Device**" is the drive letter corresponding to the TF card
   b. Then select "Firmware Upgrade" in "**Select Function Mode**"
   c. Then select the path of the Android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

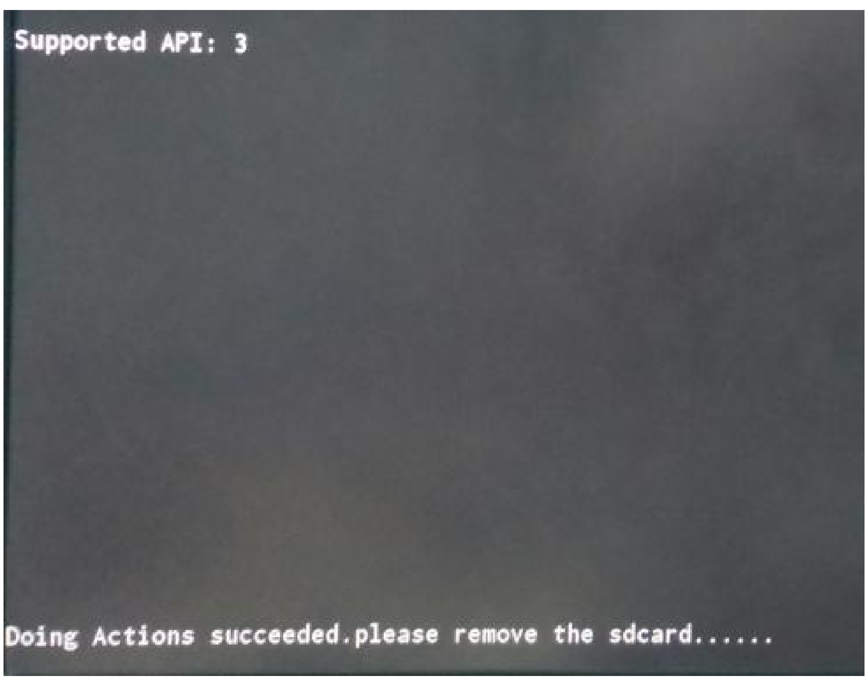


8) After burning, you can exit SDDiskTool, and then you can pull out the TF card from the computer and insert it into the development board. After the development board is powered on, it will start to burn the Android firmware in TF to eMMC.

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



10) When the following information is displayed, it means that the Android firmware has been burned into the eMMC. At this time, the TF card can be pulled out, and then the Android system in the eMMC will start to start.

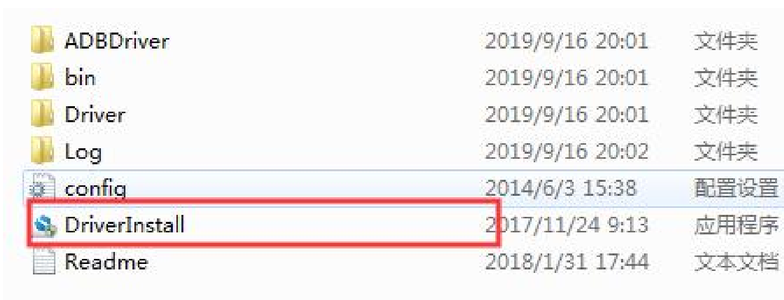


2. 7. 2. How to burn Android image into eMMC via Type-C cable

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

2) **Then download the Rockchip driver DriverAssitant_v4.6, the burning tool AndroidTool and the firmware of Android8.1 from the data download page of Orange Pi. The Android firmware on the official website has two versions: eMMC boot and TF card boot, here you need to choose eMMC Boot version, note that the name of the Android firmware that supports eMMC boot does not contain the word "SD", and please ensure that the version of the AndroidTool tool is v2.58, please do not use AndroidTool software lower than v2.58 to burn Android 8.1 firmware of Orange Pi 4 LTS, AndroidTool tools lower than this version may have problems to program Android 8.1 system**

3) Decompress **DriverAssitant_v4.6.zip** with decompression software, find the **DriverInstall.exe** executable file in the decompressed folder and open it

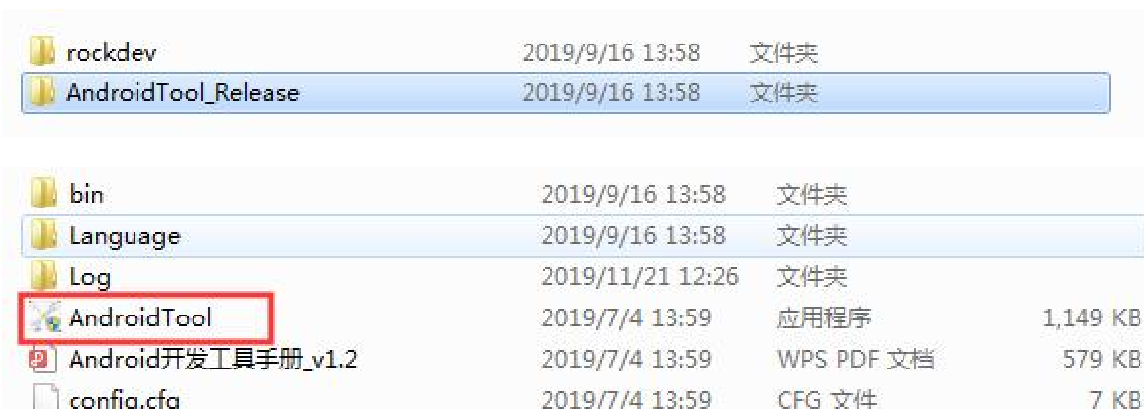


4) The steps to install Rockchip micro driver are as follows

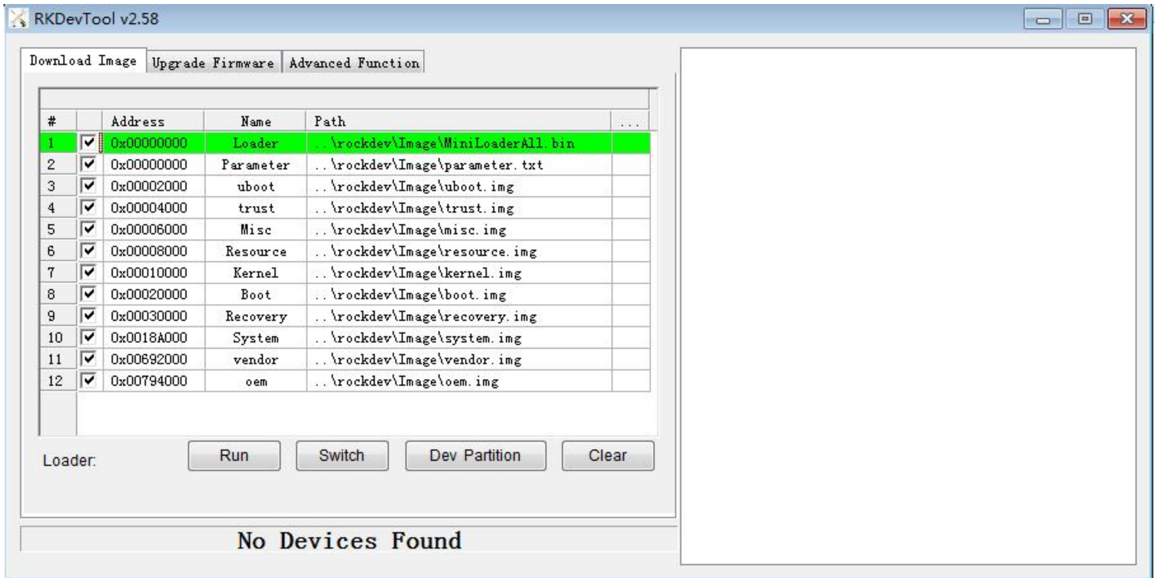
   a. Click the "**Driver Installation**" button

   b. After waiting for a while, a pop-up window will prompt "**Driver installed successfully**"

5) Unzip **AndroidTool_v2.58.zip**, this software does not need to be installed, just find **AndroidTool** in the unzipped folder and open it



6) After opening the **AndroidTool** tool, because the computer has not been connected to the Orange Pi 4 LTS development board through the Type-C cable at this time, the lower left corner will prompt "**No device found**"

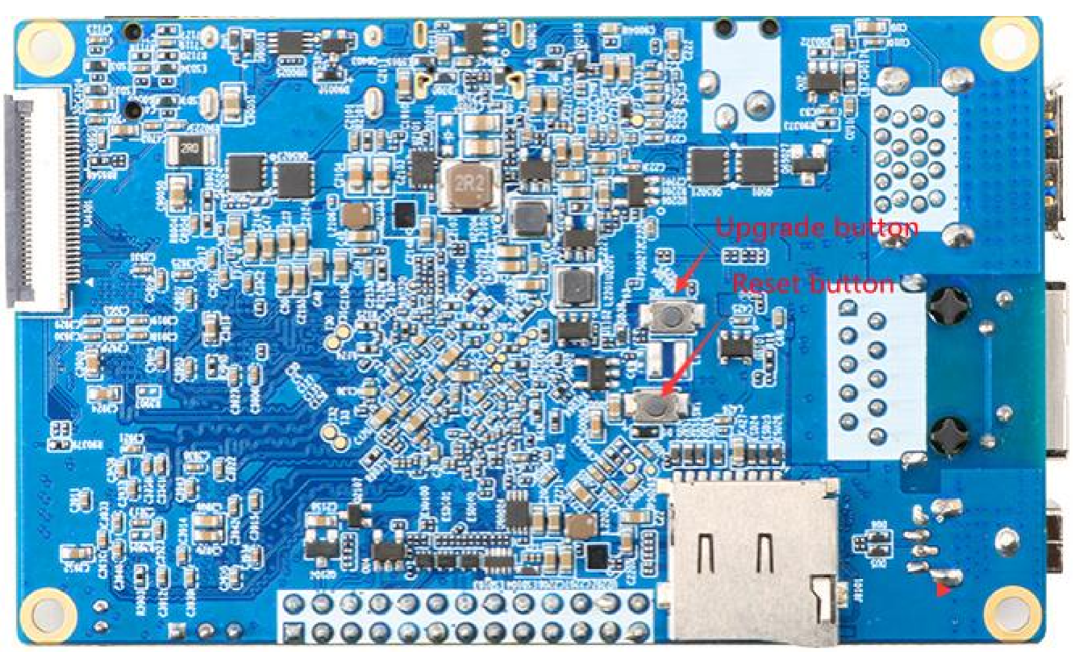


7) Then start the burning of Android firmware

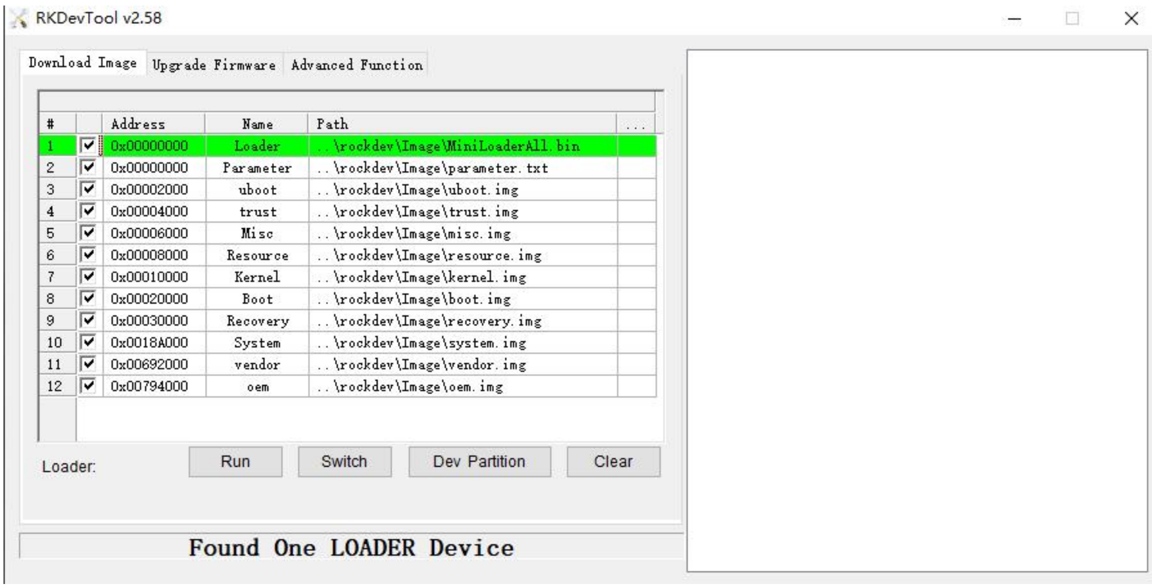
   a. First connect the DC power adapter to the OrangePi 4 LTS development board, and make sure to unplug the TF card

   b. Then connect OrangePi 4 LTS to Windows PC via Type-C cable

   c. First press and hold the upgrade button of Orange Pi 4 LTS, then lightly press the reset button and release it immediately, wait for 3~5 seconds and then release the upgrade button. The position of the button on the development board is shown in the figure below.

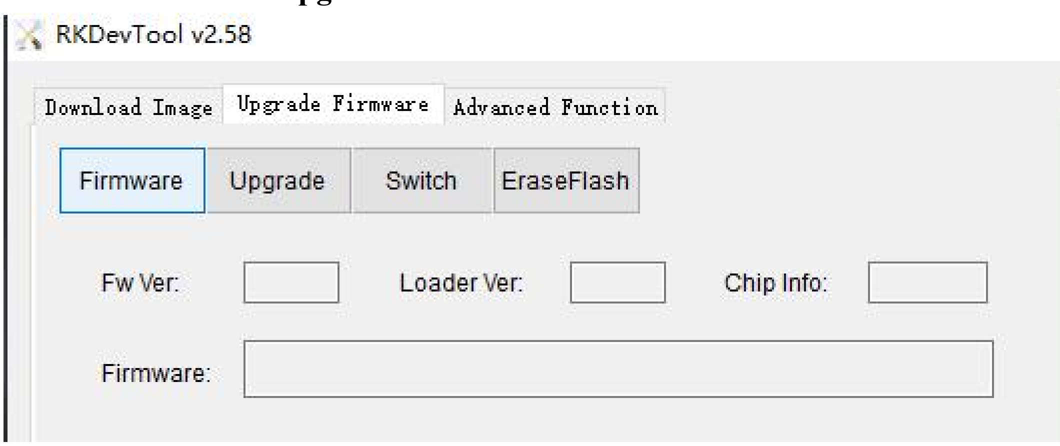


d. If the previous steps are successful, the development board has entered the **Loader mode**, and the **AndroidTool** interface will prompt "**a LOADER device was found**"

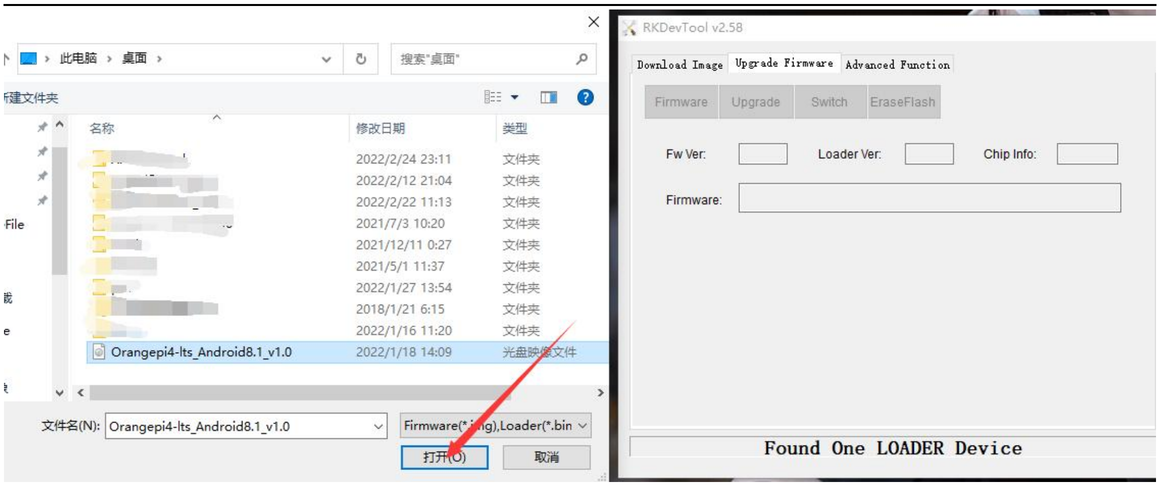


If no system is burned in eMMC, then AndroidTool will prompt "Found a MaskROM device".

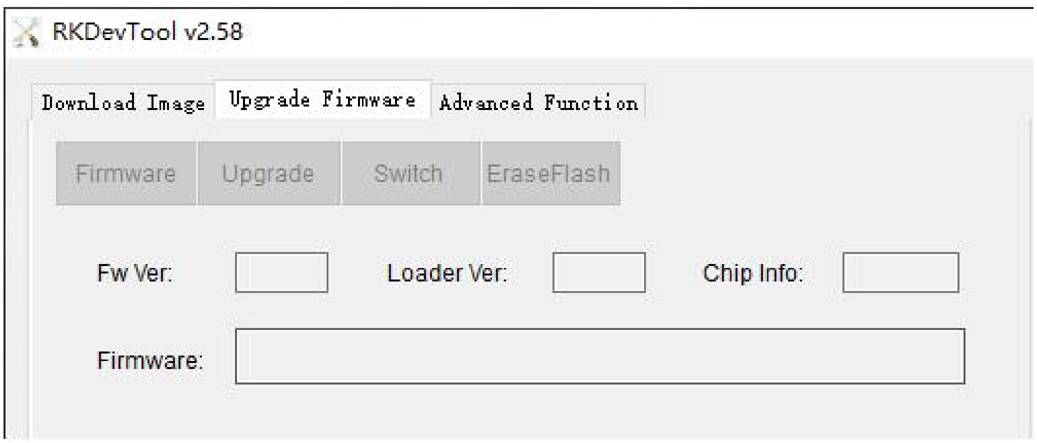
e. Then click the "**Upgrade Firmware**" column of AndroidTool



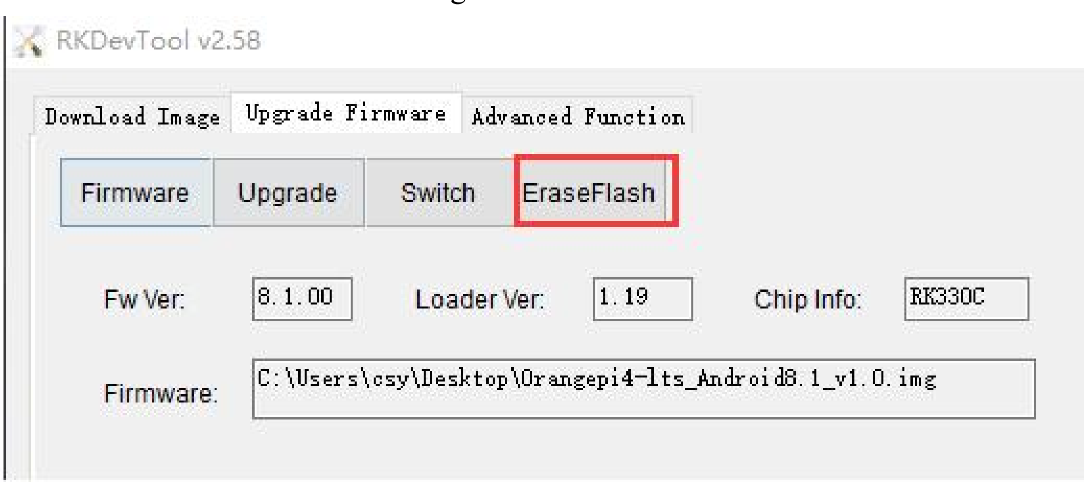
c. Then click the "**Firmware**" button to select the path of the Android firmware, and then click "**Open**", as shown in the following figure



d. After the Android firmware path selection is completed, the firmware will start to be loaded, and the button will turn into a gray unselectable state.

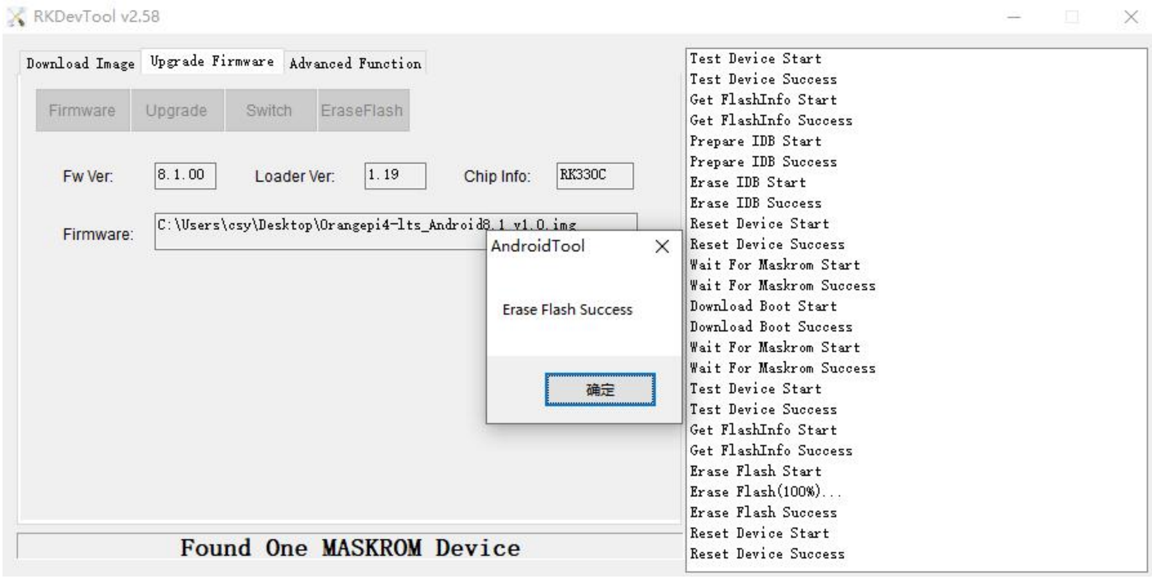


e. After the firmware is loaded, the button becomes selectable, and then click "**Erase Flash**" to start erasing eMMC

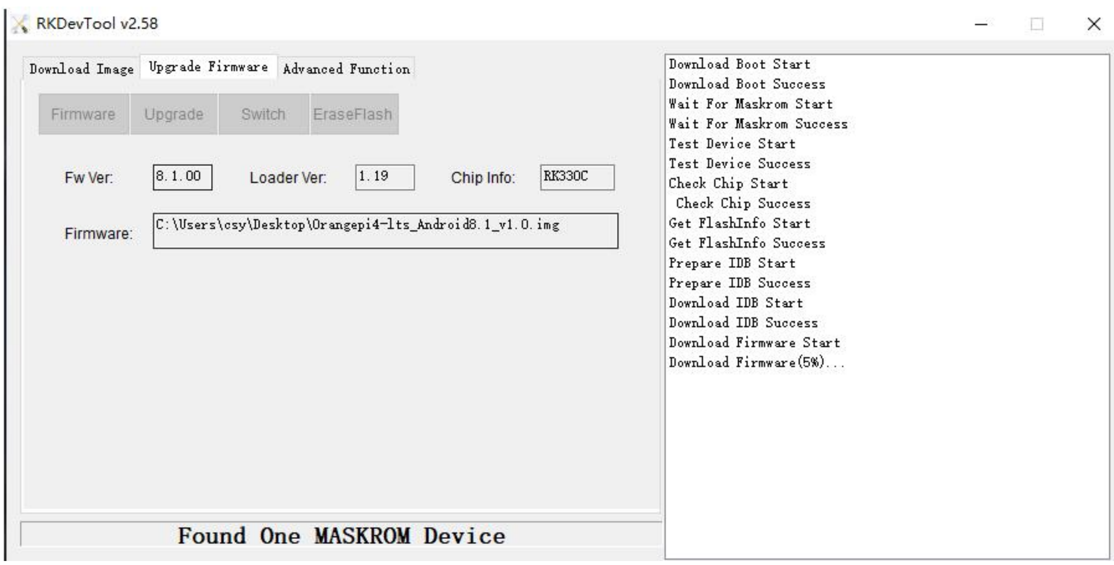


f. The interface of the successful erasing is shown in the figure below, because the system in eMMC has been erased, so you can see the prompt "**found a**

MASKROM device"



g. Finally, click the "**Upgrade**" button to burn. During the burning process, the **AndroidTool** is displayed as shown in the figure below, and the Android system will automatically start after the burning is completed.



2.8. Method of Burning Android Image to eMMC Based on Ubuntu PC

1) First prepare a good quality Type C data cable

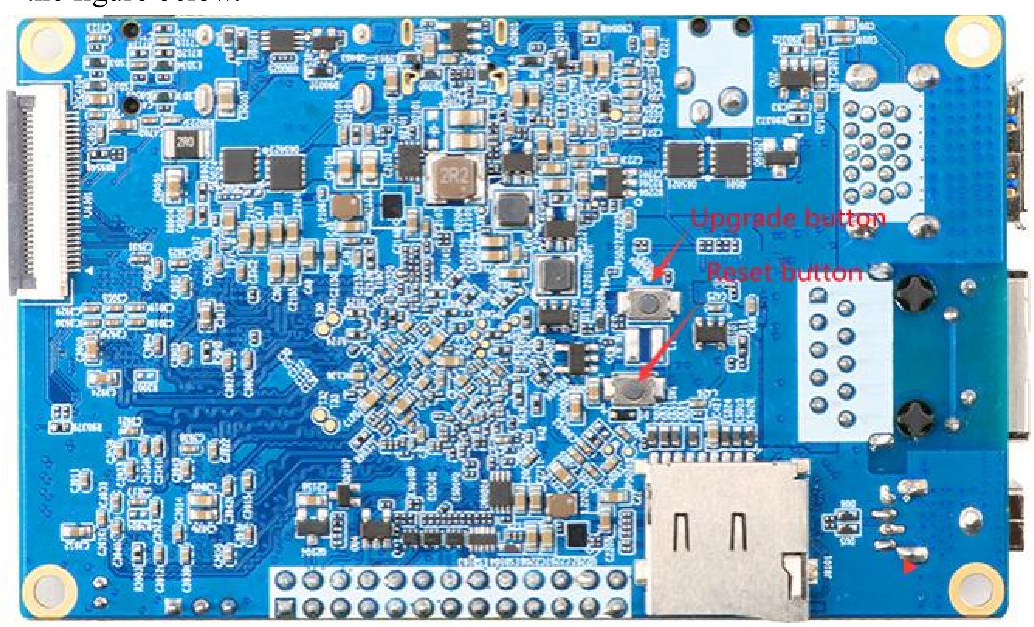
2) Then download the **upgrade_tool** tool and **Android 8.1** firmware from **the data**

download page of Orange Pi. The firmware on the official website has two versions: eMMC boot and TF card boot. Here you need to select the eMMC boot version. Note that the firmware that supports eMMC boot does not contain "SD", please make sure the upgrade_tool version is v1.39, **please do not use the upgrade_tool software lower than v1.39 to burn the Android 8.1 firmware of Orange Pi 4 LTS**, the **upgrade_tool** tool lower than this version Burning Android 8.1 **may be problematic**

3) Then execute the command in the terminal to decompress upgrade_tool and add executable permissions

```
test@test:~$ unzip Linux_Upgrade_Tool_v1.39.zip
test@test:~$ cd Linux_Upgrade_Tool_v1.39
test@test:~/Linux_Upgrade_Tool_v1.39$ sudo chmod +x ./upgrade_tool
```

4) Then start the burning of Android firmware
   a. First connect the DC power adapter to the OrangePi 4 LTS development board, and make sure to unplug the TF card
   b. Then connect OrangePi 4 LTS with Ubuntu PC via Type-C data cable
   c. First press and hold the upgrade button of Orange Pi 4 LTS, then lightly press the reset button and release it immediately, wait for 3~5 seconds and then release the upgrade button. The position of the button on the development board is shown in the figure below.



d. If the previous steps are successful, the development board has entered the **Loader** mode at this time, execute the following command and you will see **Mode=Loader**, indicating that the **Loader** device has been recognized

```
test@test:~/Linux_Upgrade_Tool_v1.39$ ./upgrade_tool LD
Program Data in /home/csy/.config/upgrade_tool
List of rockusb connected(1)
DevNo=1     Vid=0x2207,Pid=0x330c,LocationID=2010201 Mode=Loader
```

If no system is burned in eMMC, then the Maskrom device will be recognized, and the value of Mode will be Maskrom.

e. Then copy the downloaded Android image to the **Linux_Upgrade_Tool_v1.39** directory

f. Then enter the following command in the terminal of the Ubuntu PC to erase the eMMC

```
test@test:~$ sudo ./upgrade_tool ef Orangepi4-lts_Android8.1_v1.0.img
```

g. Finally execute the following command to start burning Android firmware to eMMC

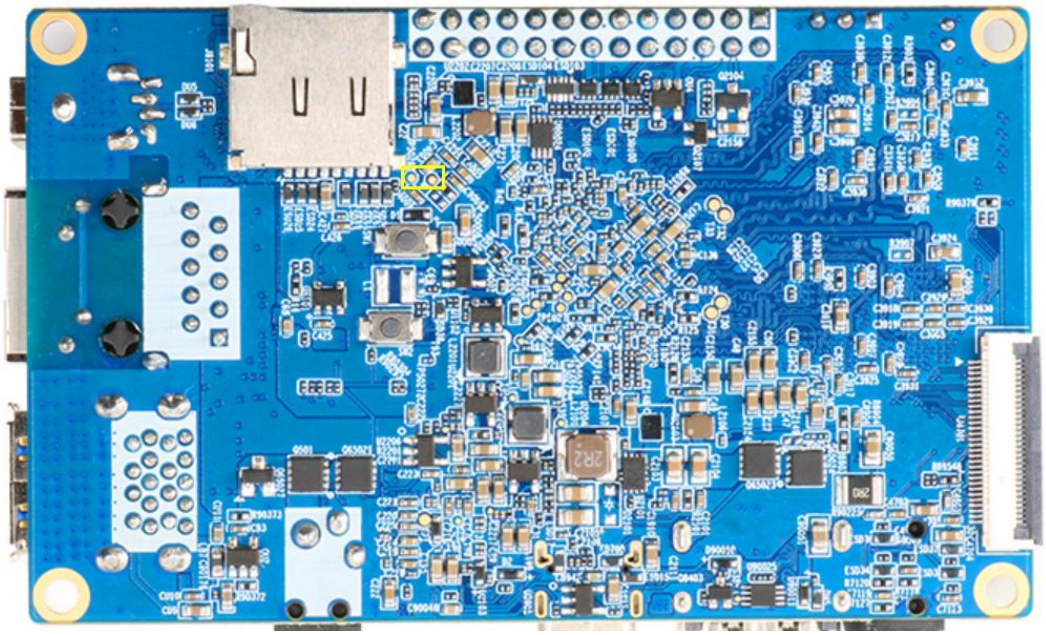
```
test@test:~$ sudo ./upgrade_tool uf Orangepi4-lts_Android8.1_v1.0.img
```

2. 9. How to enter MaskRom mode

Under normal circumstances, it is not necessary to enter the MaskRom mode. Only when the bootloader is damaged and the system cannot be started, it is necessary to enter the Maskrom mode for burning.

1) First make sure that the OrangePi 4 LTS development board is disconnected from all power supplies and unplug the TF card

2) Then use metal tweezers to connect the two test points in the yellow box in the picture below reserved by the OrangePi 4 LTS development board, and keep it still (**make sure that the two test points are short-circuited**)



3) Then plug in the DC power supply to the Orange Pi 4 LTS development board, wait for 2~3 seconds and then release the metal tweezers. At this point the OrangePi 4 LTS development board will enter maskrom mode

4) Then connect the OrangePi 4 LTS development board and the Windows PC with the Type C cable, and then open the **AndroidTool** tool, if all goes well, you can see the **AndroidTool** interface prompts "**a MASKROM device was found**"

5) At this point, you can burn the Android firmware through the **AndroidTool** tool under Windows

2. 10. Installation instructions of metal heat dissipation shell

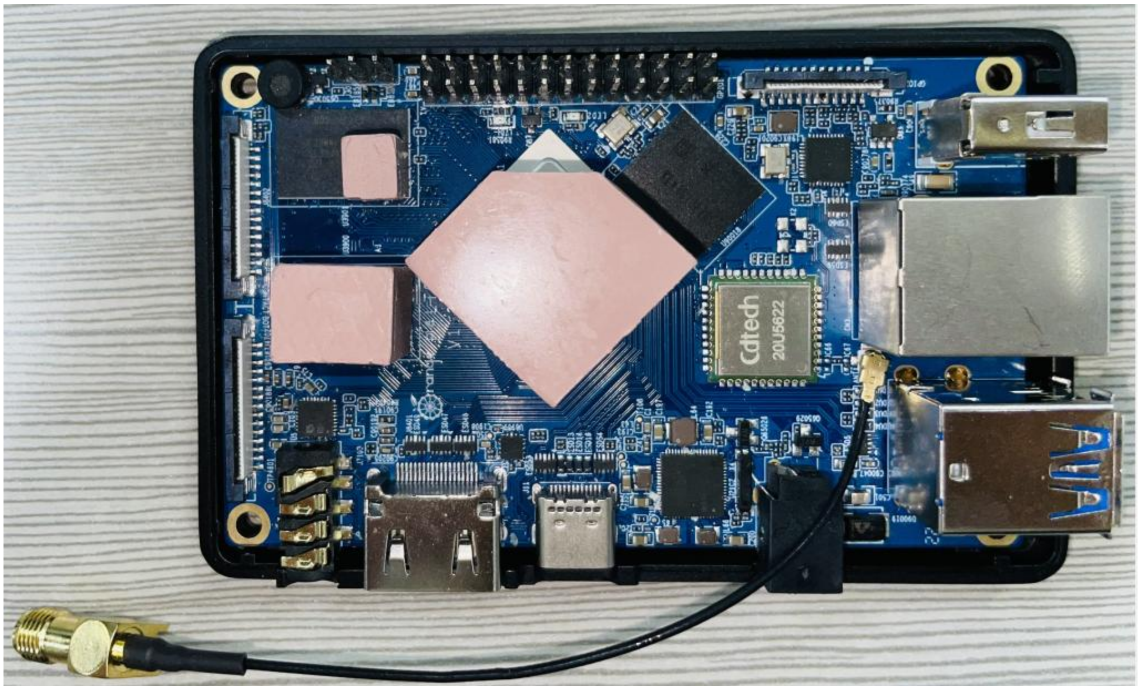
1) The metal heat dissipation shell kit includes 1 aluminum alloy shell, 3 pieces of thermally conductive silicone, 4 M2x8 screws, 1 screwdriver, and 1 set of external WIFI antenna

Note that the external antenna matched with the metal shell is optional, only available after purchase. The rightmost one in the picture below is the external antenna, please do not confuse it with the antenna that comes with the board. The antenna that comes with the board does not need to be purchased separately.

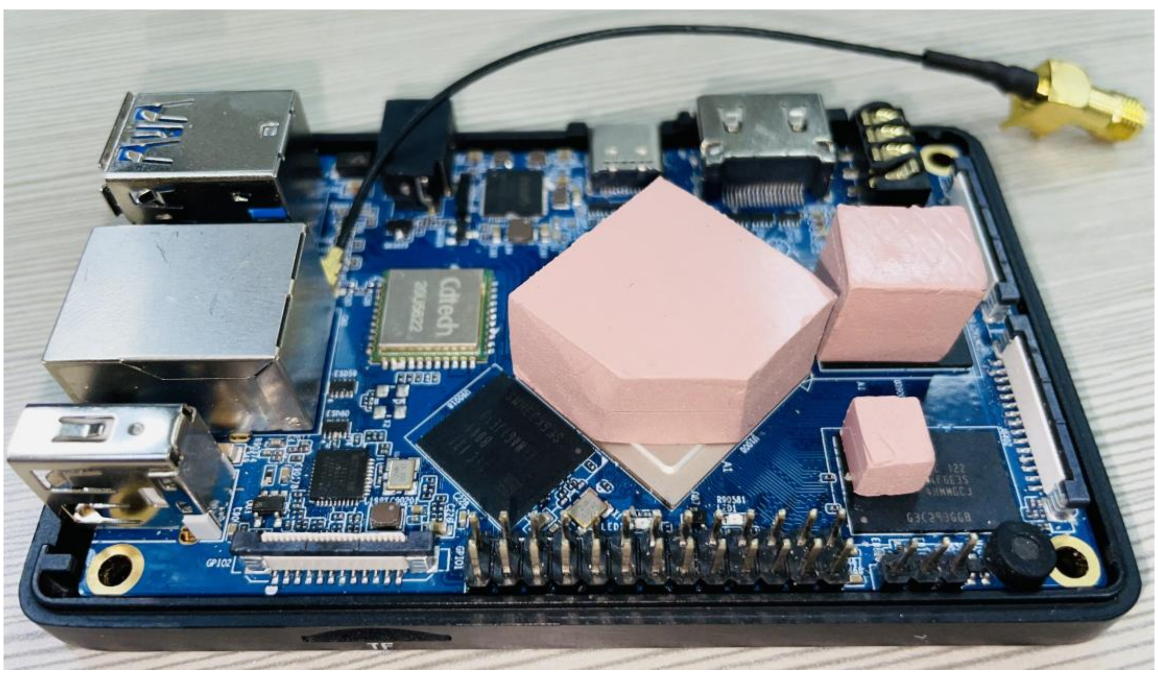
2) First install the external WiFi antenna to the development board

3) Then put the development board on the lower shell of the metal heat dissipation shell, pay attention to the alignment of the four screw holes

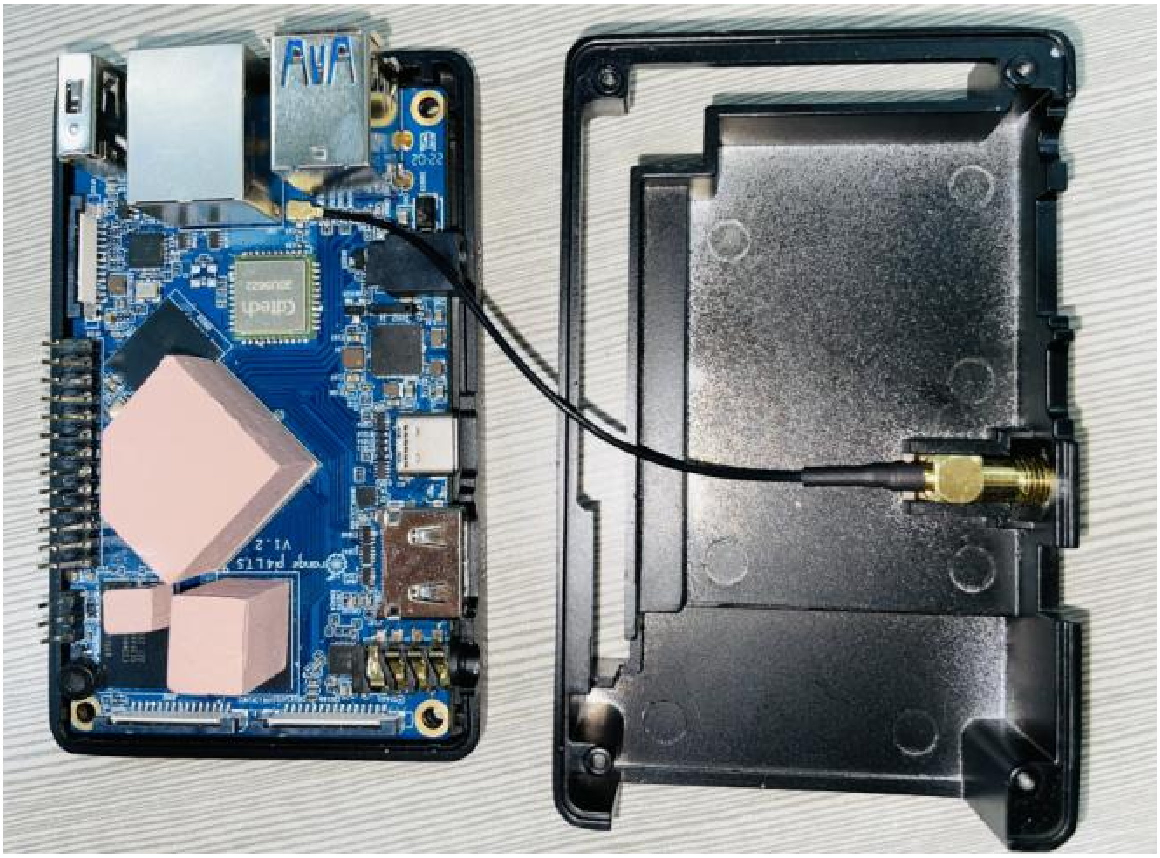
4) Tear off the protective stickers on both sides of the 3 pieces of thermally conductive silica gel, and then put 3 pieces of thermally conductive silica gel on the CPU chip and memory chip of the development board, which need to be placed in the corresponding positions as shown in the figure below. Pay attention to the missing corners of the thermally conductive silica gel of the CPU chip. Side facing 26pin double row pin



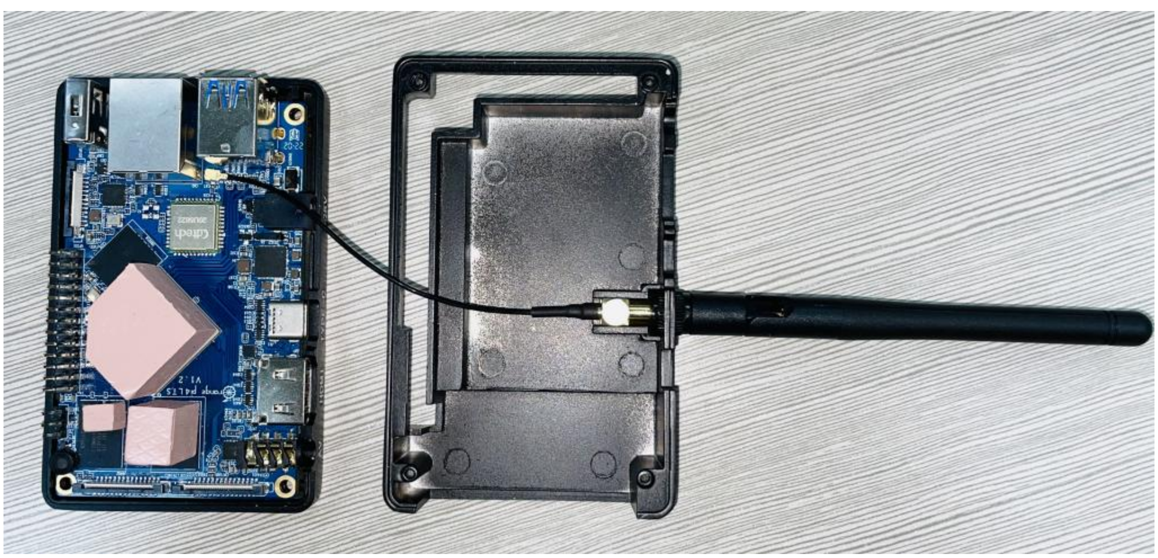
The height of the thermal conductive silica gel at the CPU chip and the memory chip is basically the same, please pay attention to this



5) Then insert the copper column of the WiFi external antenna into the groove of the upper shell of the metal heat dissipation shell, as shown in the figure below, pay attention to the angle and direction



6) Then tighten the glue post of the WiFi external antenna to the copper post of the WiFi external antenna, as shown in the figure below

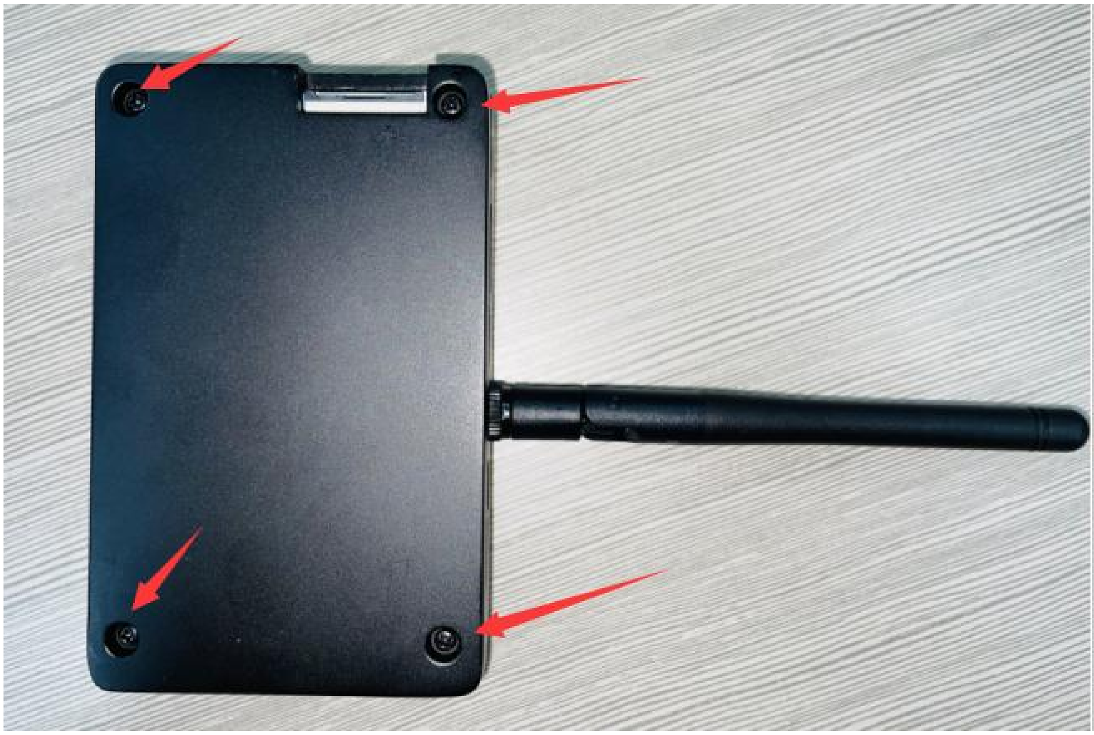


7) Then align the upper case with the lower case as shown in the figure below, and slowly press down to fasten them together. Note that if the fastening fails, the thermal

conductive silicone may be displaced. Readjust the position of the lower thermal conductive silicone, and then try to fasten it again.

8) Finally, let the lower shell of the metal heat dissipation shell face upward, and use a

Phillips screwdriver to tighten the 4 screws to the lower shell of the metal heat dissipation shell, and the assembly is completed.



2. 11. Start the orange pi development board

1) If the purchased development board has an onboard eMMC, the **Debian11** image will be pre-installed by default. You can directly use the Debian image in the eMMC to start and test some functions when you just get the development board.

2) If the purchased development board does not have on-board eMMC, you can insert the TF card with the burned Linux or Android image into the TF card slot of the 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

4) Connect the 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 5V/3A (5V/4A can also) **high-quality** power adapter

a. **Orange Pi 4 LTS cannot be powered through the 5v pin on the 26pin interface**

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

c. **Many unstable phenomena during system power-on and startup are basically caused by power supply problems, so a reliable power adapter is very important**

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

8) If you want to view the output information of the system through the debugging serial port, please use the USB to TTL module and DuPont 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 debug the serial port

2. 12. 1. Connection instructions for debugging serial port

1) First, you need to prepare a **3.3V** USB to TTL module. For better platform compatibility, it is recommended to use the CH340 USB to TTL module. Then insert one end of the USB interface of the USB to TTL module into the USB interface of the computer

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

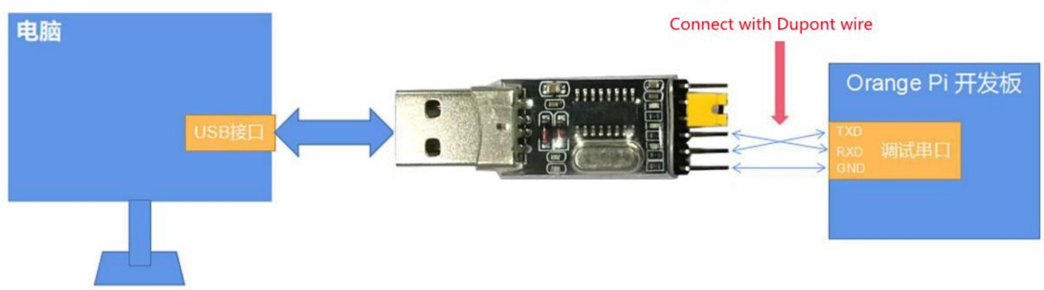
3) The GND, TXD and RXD pins of the USB to TTL module need to be connected to the debug serial port of the development board through a DuPont cable

a. Connect the GND of the USB to TTL module to the GND of the development board

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

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

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



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

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

5) If the CP2102 USB to TTL module is used, in the case of a baud rate of 1500000, some systems may encounter garbled or unusable problems. The specific test situation is as follows

USB to TTL module model	host system	Support situation

CH340	win7	OK
	win10	OK
	ubuntu14.04	OK
	ubuntu18.04	OK
	ubuntu20.04	OK
CP2102	win7	OK
	win10	NO
	ubuntu14.04	OK
	ubuntu18.04	NO
	ubuntu20.04	NO

2. 12. 2. How to use the debugging serial port on Ubuntu platform

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

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

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

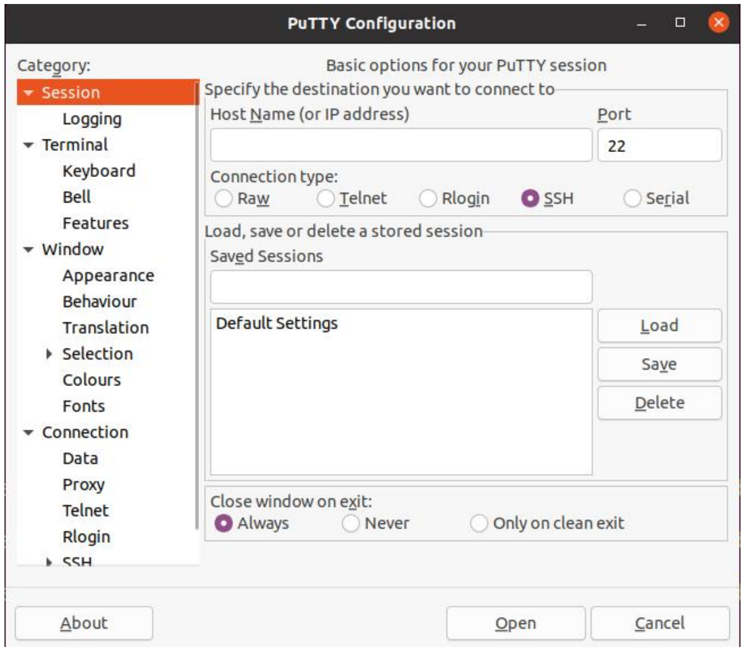
2) Then use the following command to install putty on Ubuntu PC

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install putty
```

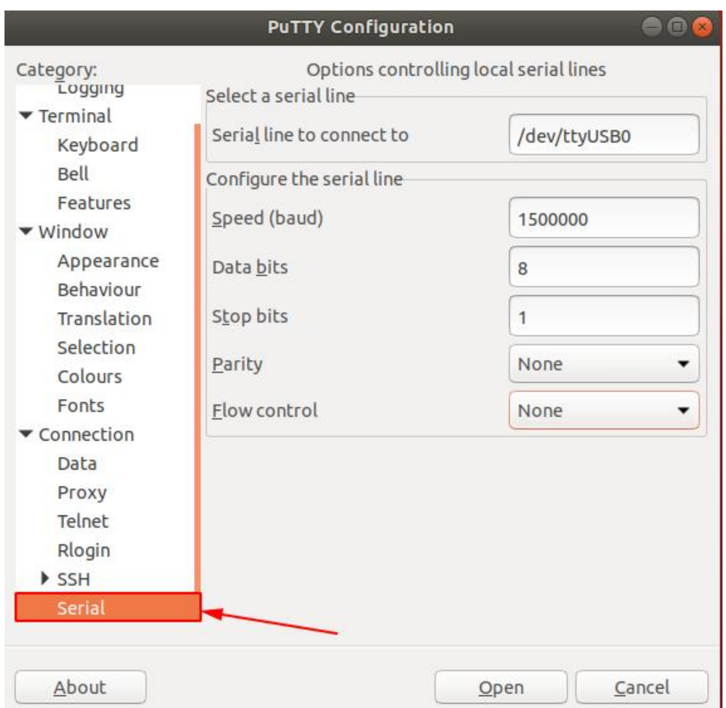
3) Then run putty, **remember to add sudo permissions**

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

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



5) First select the setting interface of the serial port

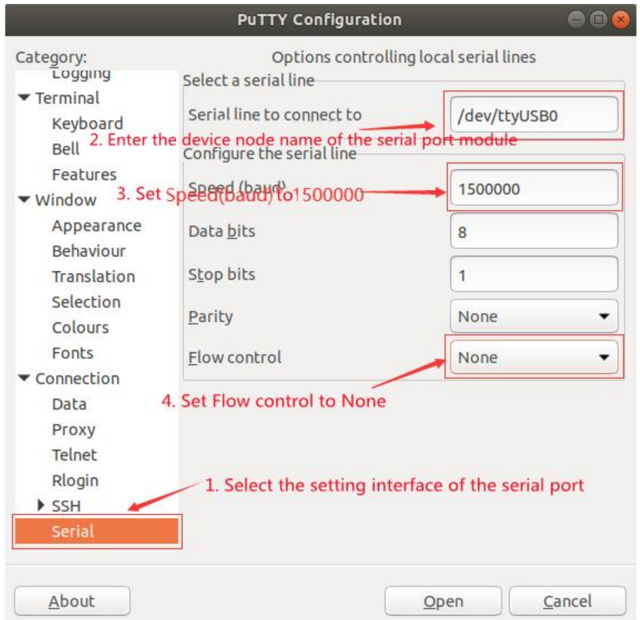


6) Then set the parameters of the serial port

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

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

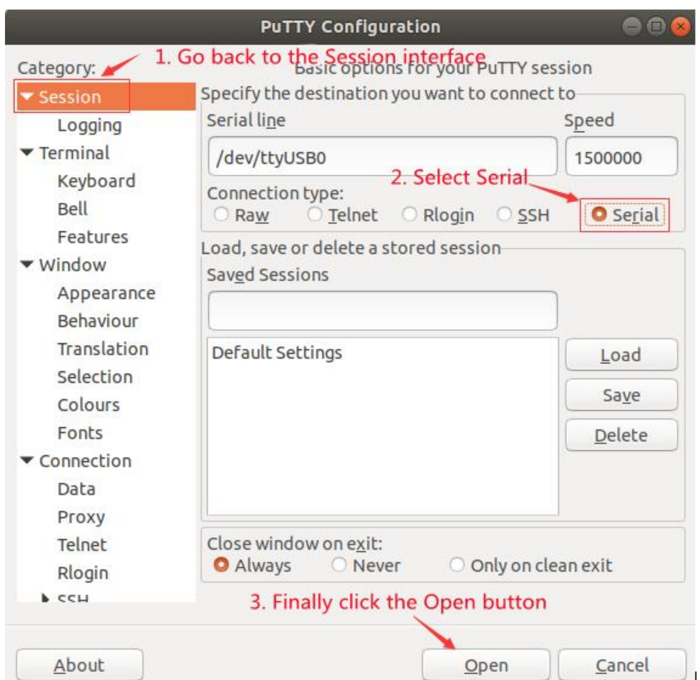
c. Set Flow control to None



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

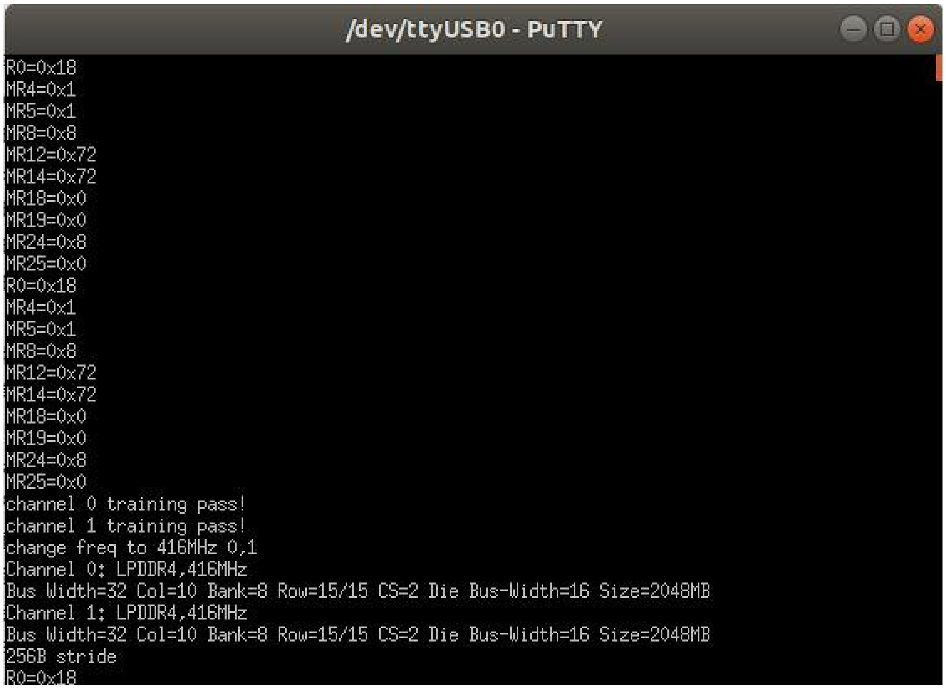
a. First select the Connection type as Serial

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



8) After starting the development board, you can see the Log information output by the

system from the open serial terminal



2. 12. 3. How to use the debugging serial port on Windows platform

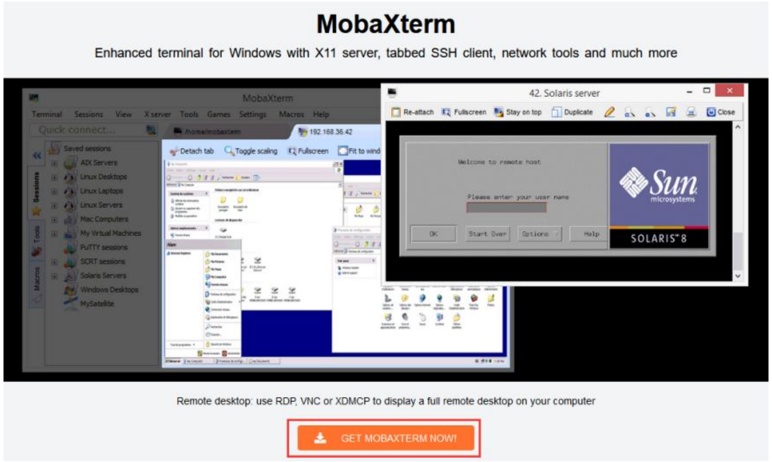
There are many serial debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following shows how to use MobaXterm. This software has a free version and can be used without purchasing a serial number.

1) Download MobaXterm

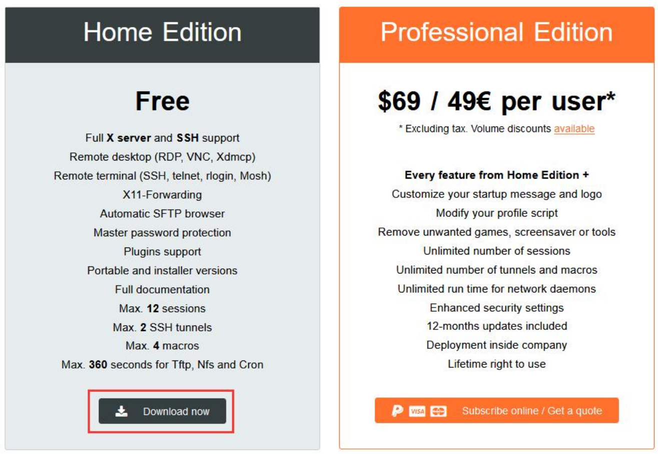
a. Download MobaXterm URL as follows

https://mobaxterm.mobatek.net

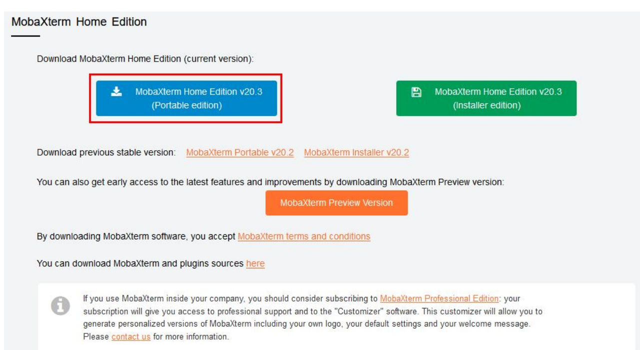
b. After entering the MobaXterm download page, click **GET XOBATERM NOW!**



c. Then choose to download the Home version

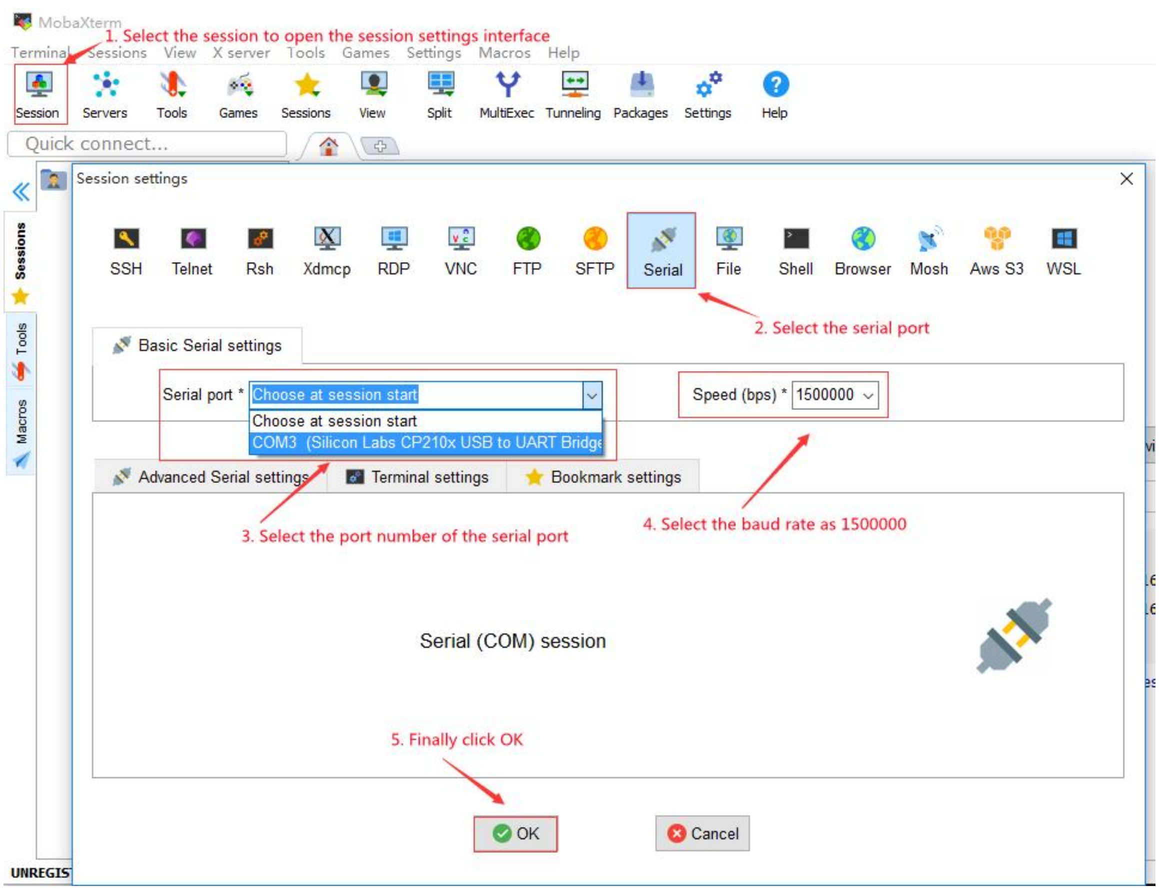


d. Then select the Portable portable version. After downloading, there is no need to install it, just open it and use it

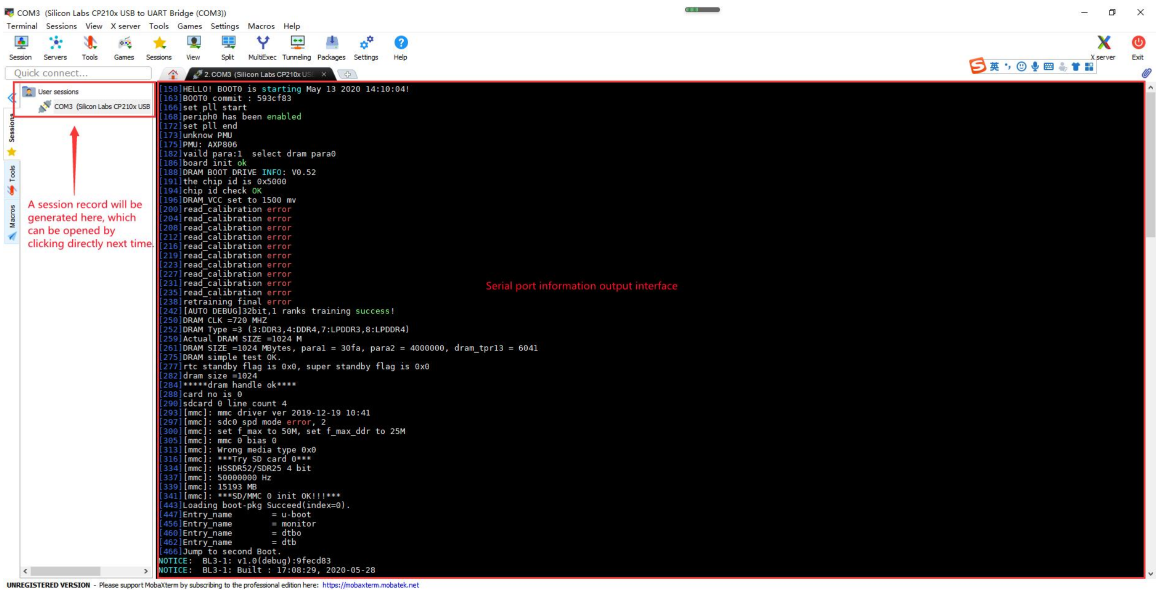


2) After downloading, use the decompression software to decompress the downloaded compressed package, you can get the executable software of MobaXterm, and then double-click to open it

3) After opening the software, the steps to set the serial port connection are as follows
   a. Open the session settings interface
   b. Select the serial port type
   c. Select the port number of the serial port (select the corresponding port number according to the actual situation). If you cannot see the port number, please use the **360 driver master** to scan and install the driver for the USB to TTL serial port chip.
   d. Select the baud rate of the serial port to be **1500000**
   e. Finally click the "**OK**" button to complete the setting



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



3. Instructions for use of Linux system

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

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

We can see from the official websites of Ubuntu and Debian that they both support the arm64 architecture (the SOC of the development board is the arm64 architecture), but please note that the support mentioned here only refers to the arm64 version of the software repository provided by Ubuntu or Debian (including tens of thousands of packages) or rootfs (which is what Orange Pi uses to make Ubuntu or Debian systems). To make an Ubuntu or Debian system that can be used for a certain development board, you need to transplant U-boot and Linux kernel and other things, and also need to fix the bugs encountered and optimize some functions. These are all done by Orange Pi.

Since Orange Pi only maintains Ubuntu and Debian systems, if these Linux distributions such as CentOS, Kali or OpenWRT are not ported by other developers

or adapted by themselves, they cannot be used on the development board of Orange Pi (the hardware runs these systems). is fine).

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

3. 1. Supported linux distribution types and kernel versions

release type	kernel version	Server Edition	desktop version
Debian 10	Linux 4.4	support	support
Debian 11	Linux5.10	support	support
Ubuntu 20.04	Linux5.10	support	support
Ubuntu 22.04	Linux5.18	support	support

All the above images may not be compiled and put on Google network disk for everyone to download and use. Because if they are all released, there will be too many, and many students who are new to Linux may have difficulty choosing.

If you must use a certain version of Ubuntu or Debian for some special needs, but there is no ready-made image to download, you can use the source code provided by Orange Pi to compile the desired image yourself. If it is written to support, then in the code It's all adapted. For the compilation method of Linux image, please refer to Chapter 5.

1) You can see the following download options on **the data download page of Orange Pi**. **Ubuntu image and Debian image are generally referred to as Linux image.**

a. Click the **Ubuntu image** download link to download Ubuntu related image. For example, after opening the Google network disk, you can see the following

Ubuntu image (**the image version number may be updated**)

Orangepi4-lts_3.0.4_ubuntu_focal_desktop_xfce_linux5.10.43.7z	628.9M	2022-05-31 17:05
Orangepi4-lts_3.0.2_ubuntu_jammy_server_linux5.17.11.7z	365.4M	2022-05-30 20:23
Orangepi4-lts_3.0.2_ubuntu_jammy_desktop_xfce_linux5.17.11.7z	841.1M	2022-05-30 20:23
Orangepi4-lts_3.0.2_ubuntu_focal_server_linux5.10.43.7z	287.1M	2022-05-31 17:05
Orangepi4-lts_3.0.0_ubuntu_bionic_linux4.4.179_xfce_desktop.tar.xz	548.9M	2022-06-29 11:43

b. Click the Debian mirror download link to download Debian-related mirrors. For example, after opening the Google network disk, you can see the following Debian mirrors (**the mirror version number may be updated**)

Orangepi4-lts_3.0.4_debian_bullseye_server_linux5.10.43.7z	361.1M	2022-05-31 17:04
Orangepi4-lts_3.0.4_debian_bullseye_desktop_xfce_linux5.10.43.7z	808.3M	2022-05-31 17:04
Orangepi4-lts_3.0.2_debian_buster_server_linux4.4.179.7z	358.9M	2022-05-31 09:26
Orangepi4-lts_3.0.2_debian_buster_desktop_xfce_linux4.4.179.7z	663.8M	2022-05-31 09:26

2) Naming rules for Linux images

Development board model_version number_Linux distribution type_distribution code_server or desktop_kernel version

a. **Model of development board**: All are **Orangepi4-lts**. The model names of different development boards are generally different. Before burning the image, please make sure that the model name of the selected image matches the development board.

b. **Version number**: For example, **3.x.x**, this version number will increase with the update of the mirror function. In addition, the last number of the version number of the Linux mirror of the development board is an even number.

c. **Types of Linux distributions**: **Ubuntu** and **Debian** are currently supported. Since Ubuntu is derived from Debian, there is not much difference in use between the two systems. However, the default configuration of some software and the use of commands are still slightly different. In addition, Ubuntu and Debian both have software repositories that are maintained and supported, and there are also slight differences in the supported and installable software packages. These require personal experience in order to have a deeper understanding. For more details, you can refer to the official documentation provided by Ubuntu and Debian.

d. **Distribution code**: used to distinguish different versions of a specific Linux distribution such as Ubuntu or Debian. Among them, **focal and jammy** are both Ubuntu distributions, focal means Ubuntu 20.04, and jammy means Ubuntu 22.04. The biggest difference between different versions is that the software in the software repositories maintained by the new version of Ubuntu system is much better than that of the old version of Ubuntu system. The new ones in it, such as Python and GCC compilation toolchains, etc. **buster and bullseye** are the specific version codes of Debian, buster means Debian10, bullseye means Debian11, and Debian11 is the latest stable version released by Debian.

e. **Server or desktop**: It is used to indicate whether the system has a desktop environment. If it is **server**, it means that the system does not have a desktop environment installed. The storage space and resources occupied by the image are relatively small, and the command line is mainly used to operate the control system. If it is **desktop**, it means that the XFCE4 desktop environment is installed by default in the system, and the storage space and resources occupied by the image are relatively large. You can connect the monitor, mouse and keyboard to operate the operating system through the interface. Of course, the desktop version of the system can also be operated through the command line like the server version of the system.

f. **Kernel version**: used to indicate the version number of the linux kernel, currently supports **linux4.4, linux5.10** and **linux5.18**.

3. 2. Linux kernel driver adaptation

Function	Linux4.4	Linux5.10	Linux5.18
USB2.0x2	OK	OK	OK
USB3.0x1	OK	OK	OK
USB Type-C 3.0	OK	NO	NO
Type-C（power supply）	OK	OK	OK
MiniPCIE	OK	OK	OK
gpio（26pin）	OK	OK	OK
spi/uart4（26pin）	OK	OK	OK
i2c8（26pin）	OK	OK	OK
i2c3（26pin）	OK	OK	OK
pwm（26pin）	OK	OK	OK
Debug serial port	OK	OK	OK
eMMC	OK	OK	OK
TF card boot	OK	OK	OK
HDMI video	OK	OK	OK
HDMI audio	OK	OK	OK

MIPI camera1	OK	NO	NO
MIPI Camera 2	OK	NO	NO
Lcd1	OK	OK	OK
Lcd2	OK	OK	NO
Gigabit Ethernet port	OK	OK	OK
Network port status light	OK	OK	OK
MIC	OK	OK	OK
Headphone playback	OK	OK	OK
headphone recording	OK	OK	OK
WIFI	OK	OK	OK
Bluetooth	OK	OK	OK
LED Light	OK	OK	OK
Type-C to HDMI display	OK	OK	OK
GPU	OK	OK	OK
hardware decoding	OK	OK	OK
Reset button	OK	OK	OK
WebGL hardware acceleration	OK	OK	OK
Dual camera display at the same time	OK	NO	NO
MIPI screen dual screen simultaneous display	OK	OK	NO
MIPI screen dual screen different display	OK	OK	NO
watchdog test	OK	OK	OK

3. 3. Description of the linux command format in this manual

1) All commands in this manual that need to be entered in the Linux system will be framed by the boxes below

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

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

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

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

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

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

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

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

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

   a. As shown below, **the black and 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), 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 that cannot be written in one line will be placed on the next line, as long as the black and bold parts are commands that need to be entered. When these commands are entered on a line, the "\" at the end of each line needs to be removed, which is not part of the command. In addition, there are spaces in different parts of the command, please don't miss them

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

3. 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 don't think that there is any fault, just press Enter after entering it.

When the password is incorrectly entered, or there is a problem with the ssh connection, please note that as long as the Linux image provided by Orange Pi is used, please do not suspect that the above password is wrong, but find other reasons.

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

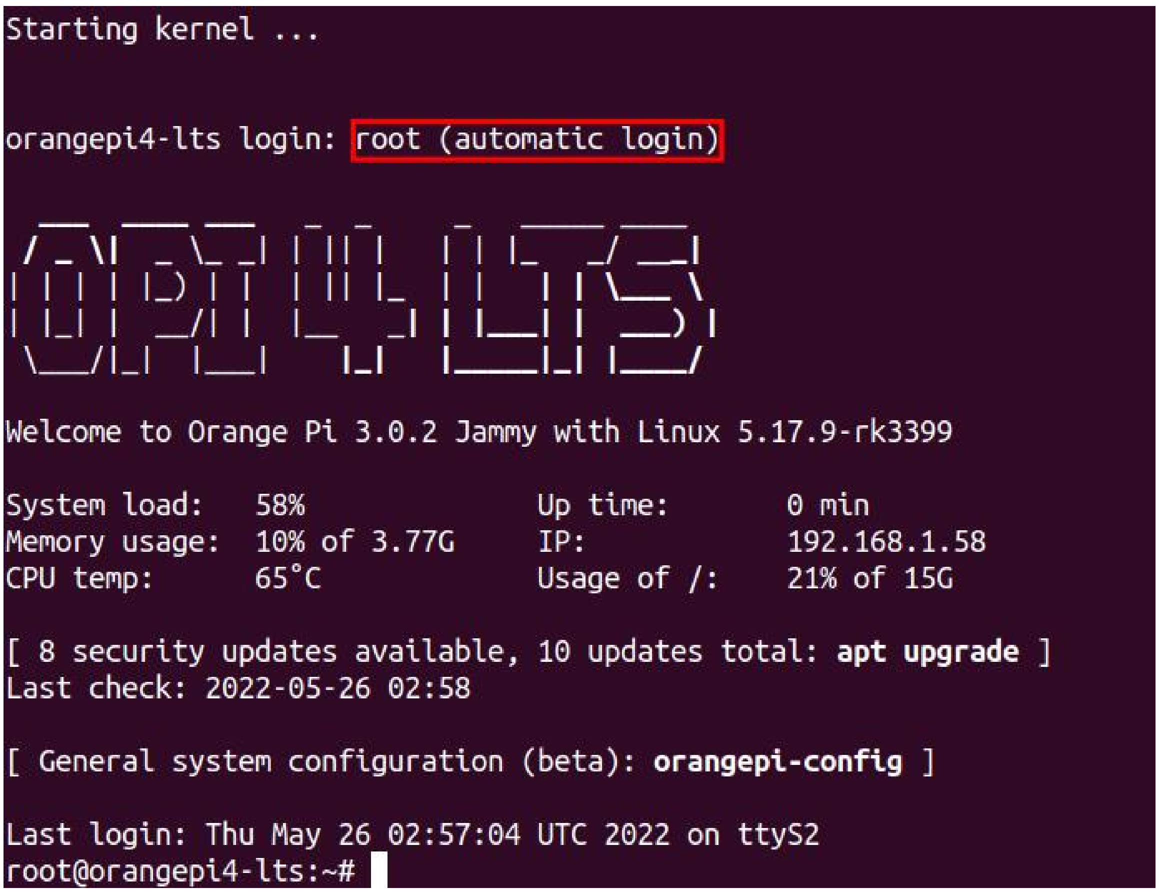
1) Method for root user to automatically log in to the terminal

a. First enter the following command to create a configuration file for terminal automatic login

```
root@orangepi:~# mkdir -p /etc/systemd/system/getty@.service.d
root@orangepi:~# mkdir -p /etc/systemd/system/serial-getty@.service.d
root@orangepi:~# cat <<-EOF > \
```

/etc/systemd/system/serial-getty@.service.d/override.conf
[Service]
ExecStartPre=/bin/sh -c 'exec /bin/sleep 10'
ExecStart=
ExecStart=-/sbin/agetty --noissue --autologin root %I \$TERM
Type=idle
EOF
root@orangepi:~# **cp /etc/systemd/system/serial-getty@.service.d/override.conf **
/etc/systemd/system/getty@.service.d/override.conf

b. Then restart the system to see that the terminal will automatically log in (no need to enter an account and password), the user used is **root**



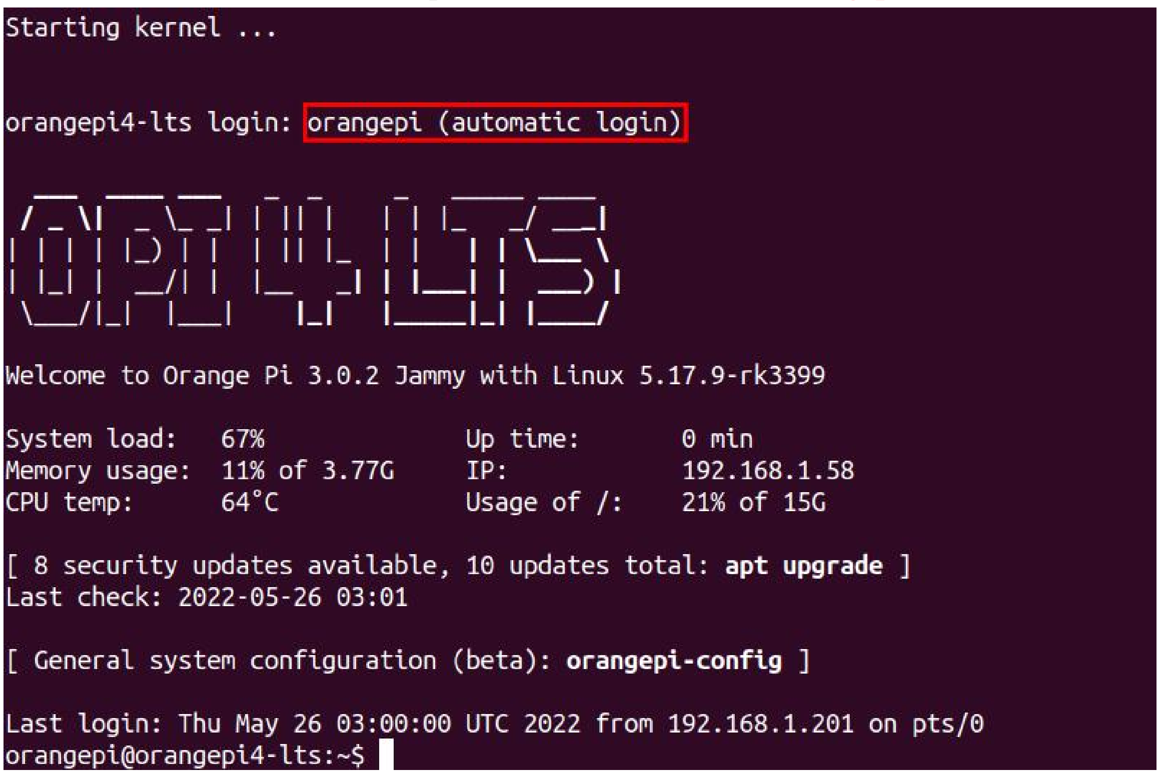
2) Method for orangepi users to automatically log in to the terminal

a. First enter the following command to create a configuration file for terminal automatic login

root@orangepi:~# **mkdir -p /etc/systemd/system/getty@.service.d**
root@orangepi:~# **mkdir -p /etc/systemd/system/serial-getty@.service.d**

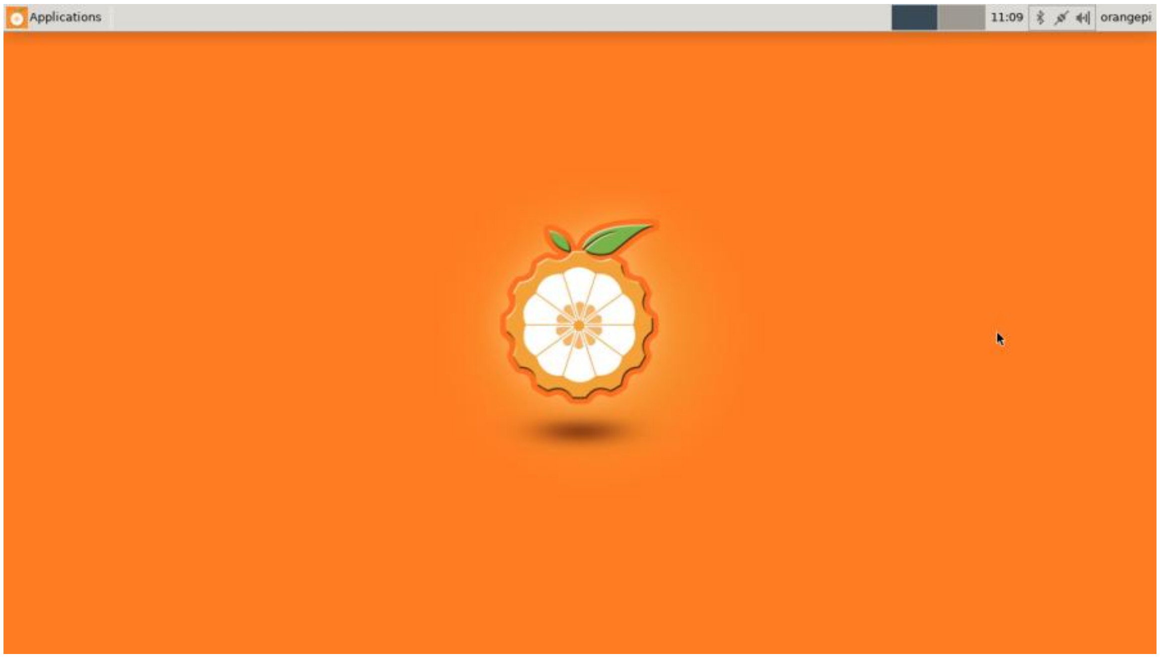
```
root@orangepi:~# cat <<-EOF >    \
/etc/systemd/system/serial-getty@.service.d/override.conf
[Service]
ExecStartPre=/bin/sh -c 'exec /bin/sleep 10'
ExecStart=
ExecStart=-/sbin/agetty --noissue --autologin orangepi %I \$TERM
Type=idle
EOF
root@orangepi:~# cp /etc/systemd/system/serial-getty@.service.d/override.conf    \
/etc/systemd/system/getty@.service.d/override.conf
```

b. Then restart the system to see that the terminal will automatically log in (no need to enter the account and password), the user used is **orangepi**



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

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



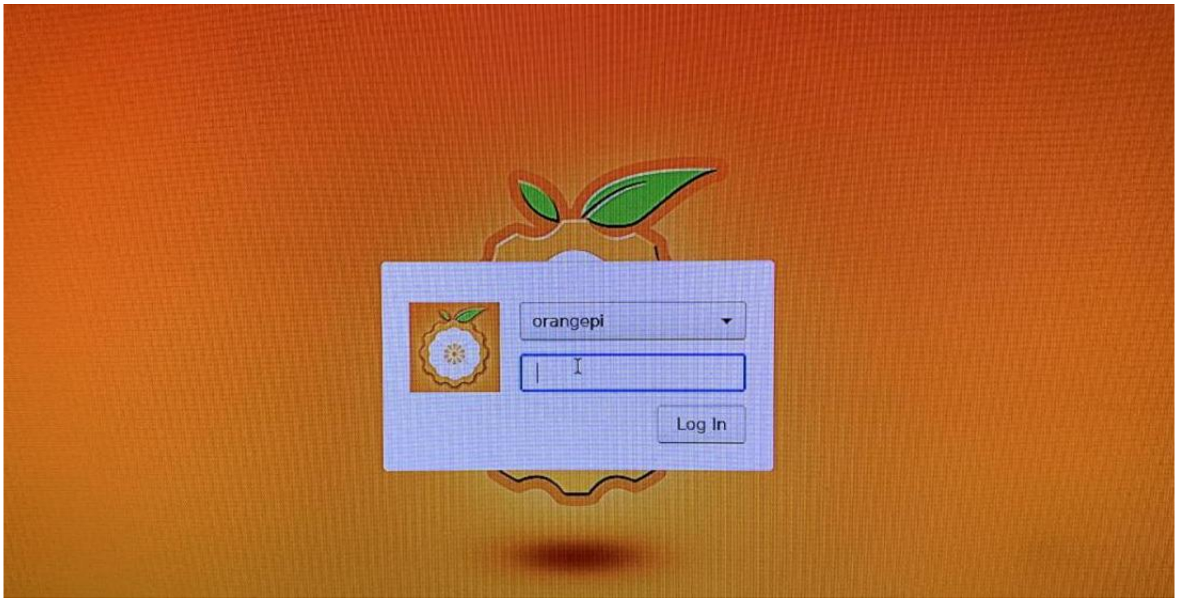
2) Modify the configuration in **/etc/lightdm/lightdm.conf.d/22-orangepi-autologin.conf** to prohibit the desktop version system from automatically logging in to the desktop. The modification command is as follows, or you can open the configuration file and modify it directly

```
orangepi@orangepi:~$ sudo sed -i    \
"s/autologin-user=orangepi/#autologin-user=orangepi/"    \
/etc/lightdm/lightdm.conf.d/22-orangepi-autologin.conf
```

3) After modification, the configuration of **/etc/lightdm/lightdm.conf.d/22-orangepi-autologin.conf** is as follows

```
orangepi@orangepi:~$ cat /etc/lightdm/lightdm.conf.d/22-orangepi-autologin.conf
[Seat:*]
#autologin-user=orangepi
autologin-user-timeout=0
user-session=xfce
```

4) Then restart the system, the login dialog box will appear, at this time, you need to enter the **password** to enter the system



3. 4. 4. Setting method for automatic login of root user in Linux desktop system

1) First set the automatic login user as **root** in **22-orangepi-autologin.conf**

```
orangepi@orangepi:~$ sudo vim /etc/lightdm/lightdm.conf.d/22-orangepi-autologin.conf
[Seat:*]
autologin-user=root
autologin-user-timeout=0
user-session=xfce
```

2) Then modify the **lightdm-autologin** configuration file, change root to anything, and remove the root user restriction

Note that Ubuntu22.04 does not need to set this step.

```
orangepi@orangepi:~$ sudo vim /etc/pam.d/lightdm-autologin
# Allow access without authentication
#auth        required pam_succeed_if.so user != root quiet_success
auth        required pam_succeed_if.so user != anything quiet_success
```

3) Then restart the system, it will automatically log in to the Linux system desktop with the root user

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

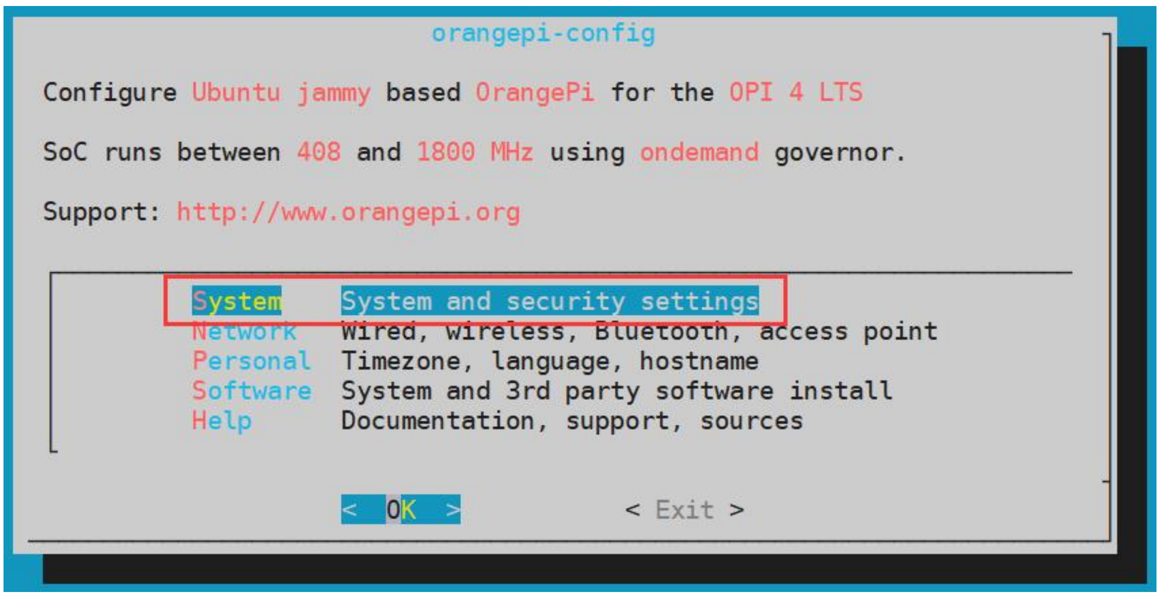
Also note that this is not a bug, since pulseaudio is not allowed to run under root.

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

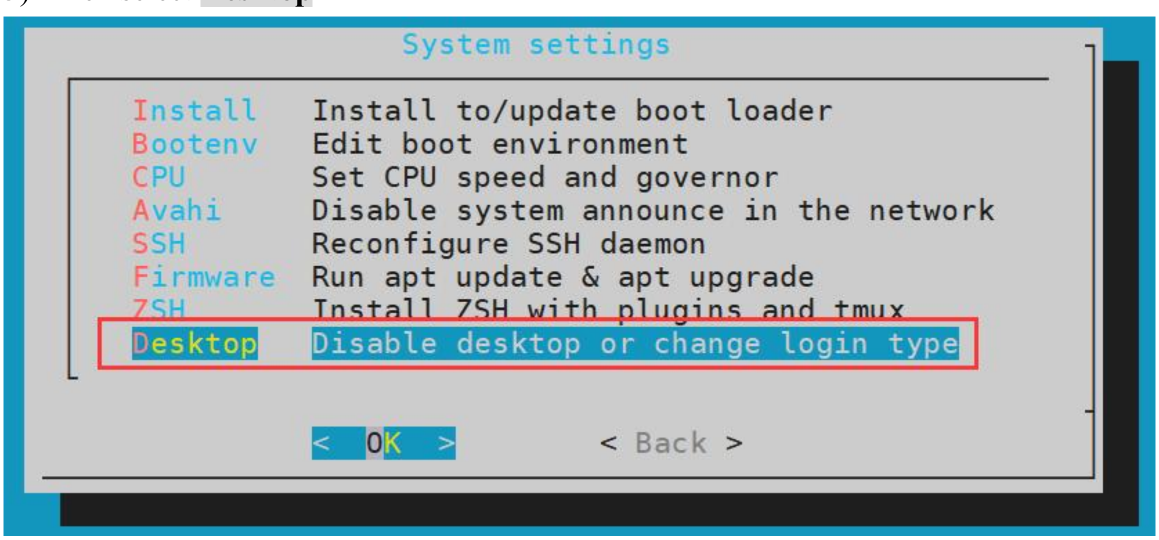
1) First enter the following command in the command line, **please remember to add sudo permissions**

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

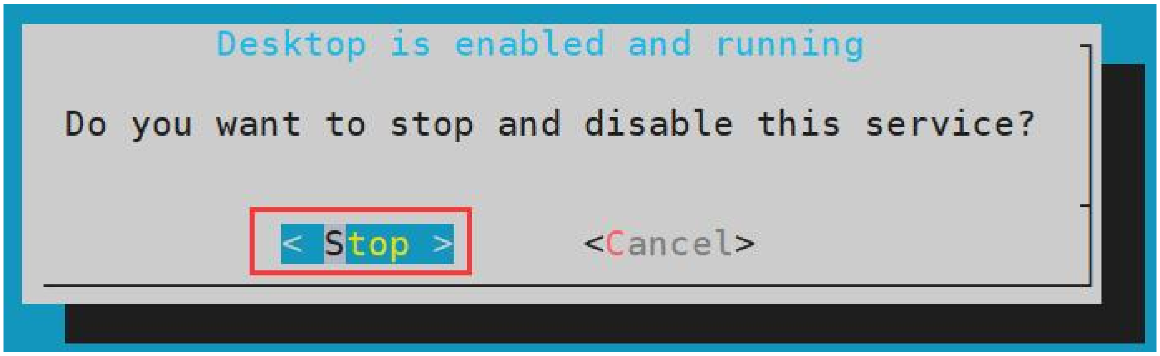
2) Then select **System**



3) Then select **Desktop**



4) Then select**<Stop>**



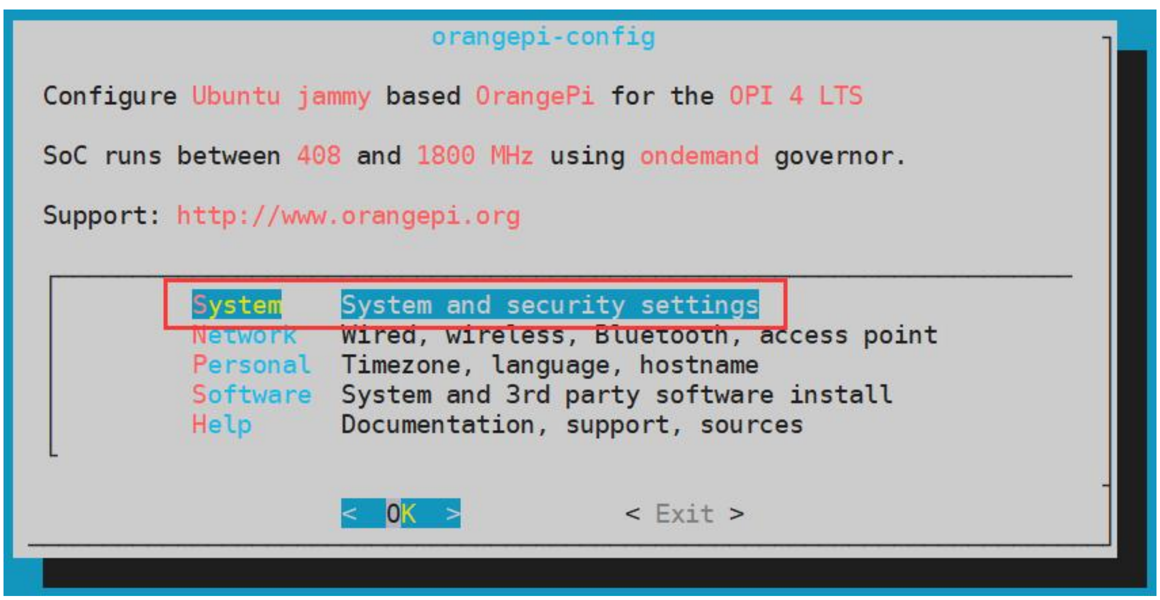
5) **Then restart the Linux system and you will find that the desktop will not be displayed**

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

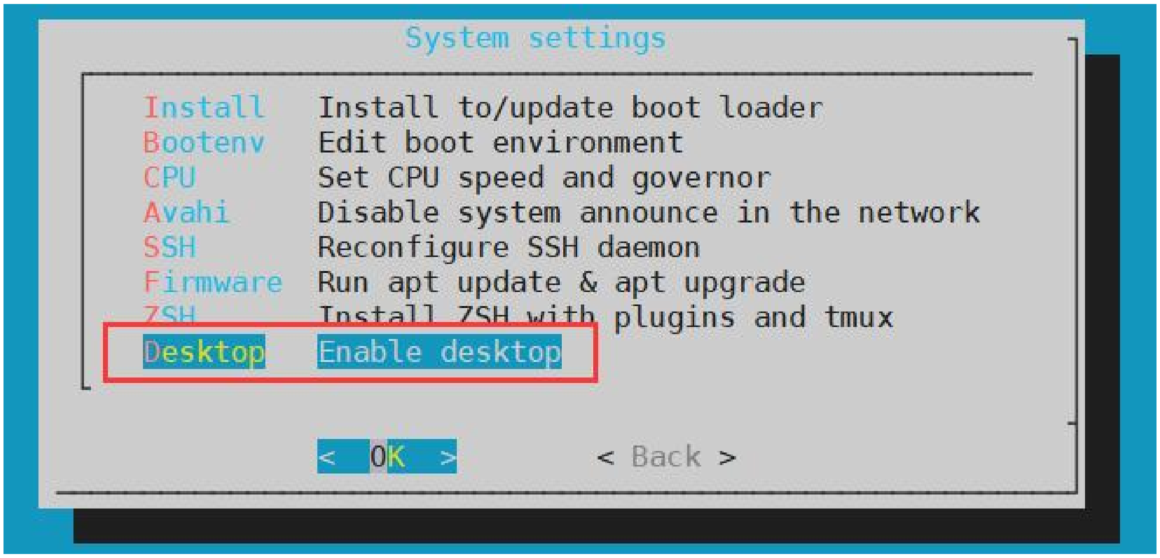
a. First, enter the following command in the command line, **please remember to add sudo permissions**

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

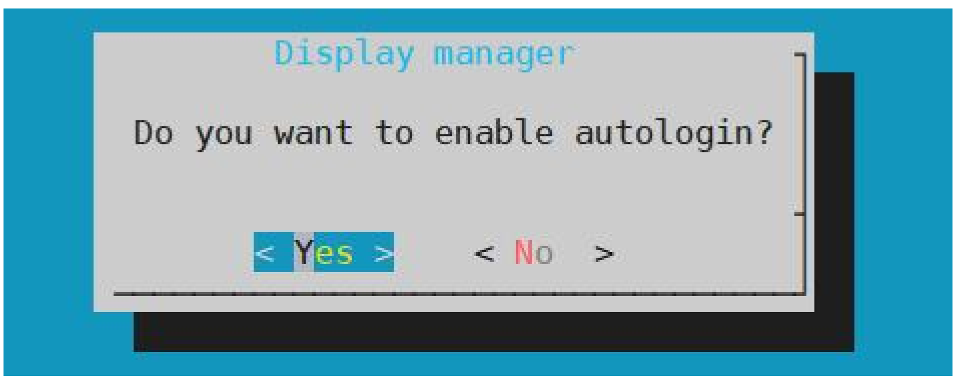
b. Then select **System**



c. Then select**Desktop Enable desktop**



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



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

3. 5. Onboard LED light display description

	green light	red light
u-boot startup phase	**off**	**on**
The kernel boots into the system	**flicker**	**on**

The red light is hardware controlled, so as long as the power is on, the red light will be on. The green light is controlled by software. After power on, wait for a

period of time. If the green light is not on, it means that the system has not started normally. At this time, you need to check whether the programming of the system is normal. If the green light starts to flash, it means that the Linux system should start normally.

3.6. Operation instructions for the capacity of the rootfs partition of the Linux system in the TF card

3.6.1. The first boot will automatically expand the capacity of the rootfs partition in the TF card

1) After burning the Linux image of the development board to the TF card, you can check the usage of the TF card capacity on the **Ubuntu computer**. The steps are as follows:

Note that if this step is not performed, it will not affect the automatic expansion of the Linux system of the development board. Here I just want to explain how to check the capacity of the TF card after burning the Linux image on the TF card.

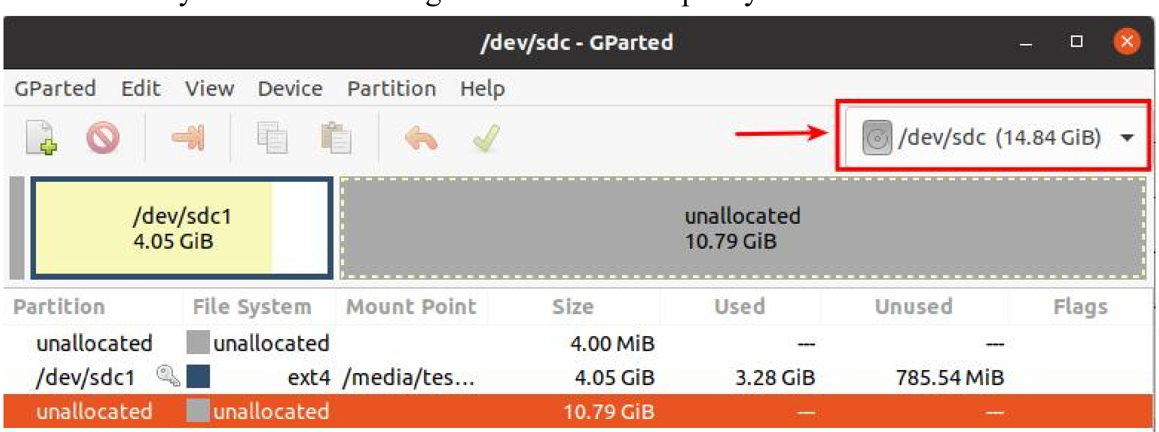
a. First install the software gparted in the Ubuntu computer

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

b. Then open gparted

```
test@test:~$ sudo gparted
```

c. After opening gparted, you can select the TF card in the upper right corner, and then you can see the usage of the TF card capacity



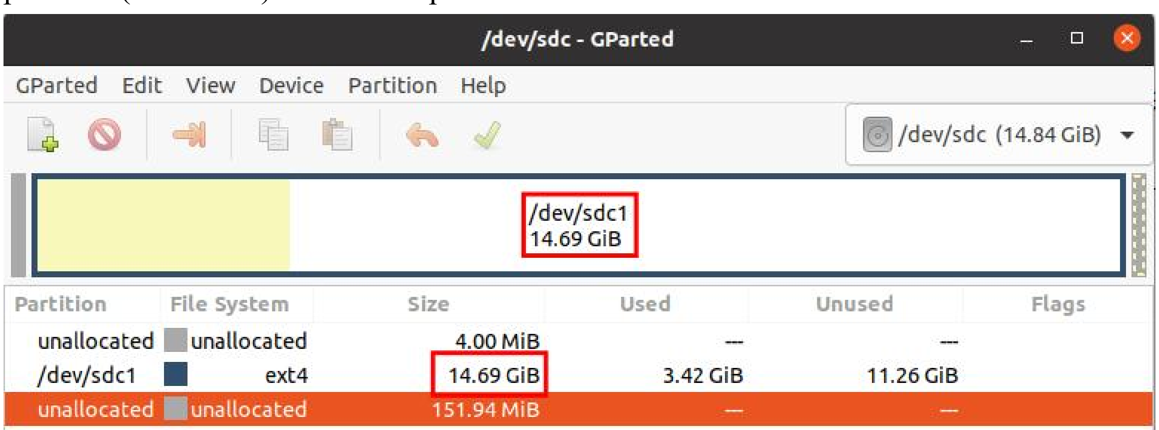
d. The above picture shows the situation of the TF card after burning the Linux desktop version system. It can be seen that although the total capacity of the TF card is 16GB (displayed as 14.84GiB in GParted), the rootfs partition (/dev/ sdc1) actually only allocated 4.05GiB, leaving 10.79GiB unallocated

2) Then you can insert the TF card with the programmed Linux system into the development board to start. When the TF card starts the Linux system for the first time, it will call the **orangepi-resize-filesystem** script through the systemd service **orangepi-resize-filesystem**.service to automatically perform The expansion of the rootfs partition, **so there is no need to manually expand**

3) After logging in to the system, you can use the **df -h** command to check the size of the rootfs. If it is consistent with the actual capacity of the TF card, it means that the automatic expansion is running correctly.

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            430M     0  430M   0% /dev
tmpfs           100M  5.6M   95M   6% /run
/dev/mmcblk0p1   15G  915M   14G   7% /
tmpfs           500M     0  500M   0% /dev/shm
```

4) After booting the Linux system for the first time, we can also remove the TF card from the development board and reinsert it into the **Ubuntu computer**, and then use gparted again to check the status of the TF card, as shown in the figure below, the rootfs partition (/dev/ sdc1) has been expanded to 14.69GiB



It should be noted that the Linux system has only one partition in ext4 format, and does not use a separate BOOT partition to store files such as kernel images, so there is no problem of expanding the BOOT partition.

3.6.2. The method of prohibiting the automatic expansion of the rootfs partition capacity in the TF card

1) First burn the linux image of the development board into the TF card on the **Ubuntu computer** (not Windows), **and then re-plug the TF card**

2) Then the Ubuntu computer will generally automatically mount the partition of the TF card. If the automatic mounting is normal, you can use the ls command to see the following output. The partition name of the TF card is not necessarily the same as the name shown in the following command.

```
test@test:~$ ls /media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db/
bin   boot   dev   etc   home   lib   lost+found   media   mnt   opt   proc   root   run
sbin   selinux   srv   sys   tmp   usr   var
```

3) Then switch the current user to the root user in the Ubuntu computer

```
test@test:~$ sudo -i
[sudo] test password：
root@test:~#
```

4) Then enter the root directory of the Linux system in the TF card and create a new file named **.no_rootfs_resize**

```
root@test:~# cd /media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db
root@test:/media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db# cd root
root@test:/media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db/root# touch .no_rootfs_resize
root@test:/media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db/root# ls .no_rootfs*
.no_rootfs_resize
```

5) Then you can uninstall the TF card, then pull out the TF card and insert it into the development board to start. When the linux system starts, when the file **.no_rootfs_resize** is detected in the **/root** directory, the rootfs will no longer be automatically expanded.

6) After prohibiting the automatic expansion of rootfs and entering the Linux system, you can see that the total capacity of the rootfs partition is only 4GB (the image of the desktop version is tested here), which is much smaller than the actual capacity of the TF card, indicating that the automatic expansion of rootfs is prohibited successfully.

```
orangepi@orangepi:~$ df -h
```

```
Filesystem      Size  Used Avail Use% Mounted on
udev            925M     0  925M   0% /dev
tmpfs           199M  3.2M  196M   2% /run
/dev/mmcblk0p1  4.0G  3.2G  686M  83% /
```

7) If you need to re-expand the capacity of the rootfs partition in the TF card, just execute the following command, and then restart the Linux system of the development board.

Note, please execute the following commands under the root user.

```
root@orangepi:~# rm /root/.no_rootfs_resize
root@orangepi:~# systemctl enable orangepi-resize-filesystem.service
root@orangepi:~# reboot
```

After restarting, enter the Linux system of the development board again, and you can see that the rootfs partition has been expanded to the actual capacity of the TF card.

```
root@orangepi:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            925M     0  925M   0% /dev
tmpfs           199M  3.2M  196M   2% /run
/dev/mmcblk0p1   15G  3.2G   12G  23% /
```

3.6.3. How to manually expand the capacity of the rootfs partition in the TF card

If the total capacity of the TF card is very large, such as 128GB, you do not want the rootfs partition of the Linux system to use all the capacity of the TF card, but only want to allocate a part of the capacity, such as 16GB, for the Linux system, and then the remaining capacity of the TF card can be used for other use. Then you can use the content described in this section to manually expand the capacity of the rootfs partition in TF.

1) First burn the linux image of the development board into the TF card on the **Ubuntu computer** (not Windows)**, and then re-plug the TF card**

2) Then the Ubuntu computer will generally automatically mount the partition of the TF card. If the automatic mounting is normal, you can use the **ls** command to see the following output. The partition name of the TF card is not necessarily the same as the

name shown in the following command.

```
test@test:~$ ls /media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db/
bin   boot   dev   etc   home   lib   lost+found   media   mnt   opt   proc   root   run
sbin   selinux   srv   sys   tmp   usr   var
```

3) Then switch the current user to the root user in the Ubuntu computer

```
test@test:~$ sudo -i
[sudo] test password:
root@test:~#
```

4) Then enter the root directory of the Linux system in the TF card and create a new file named **.no_rootfs_resize**

```
root@test:~# cd /media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db
root@test:/media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db# cd root
root@test:/media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db/root# touch .no_rootfs_resize
root@test:/media/test/27e62f92-8250-4ef1-83db-3d8f0c2e23db/root# ls .no_rootfs*
.no_rootfs_resize
```

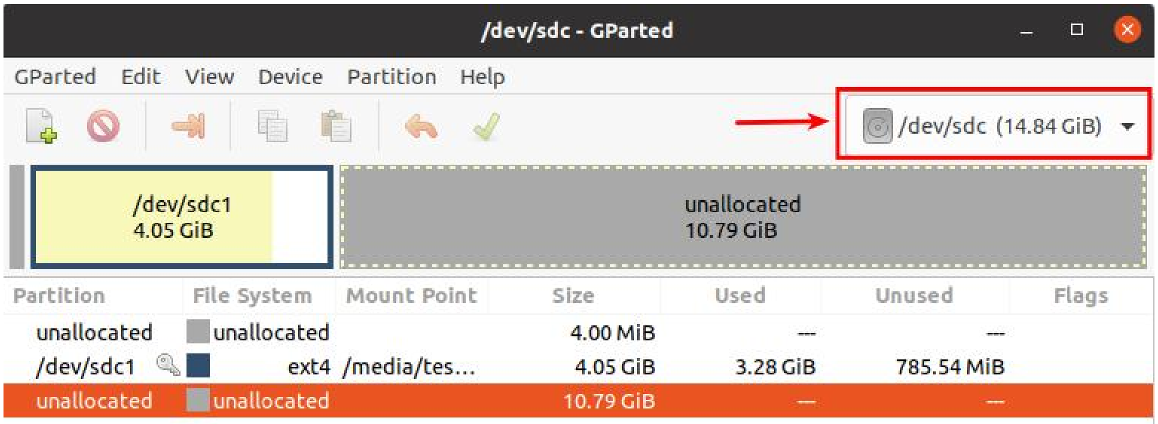
5) Then install the software gparted on the Ubuntu computer

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

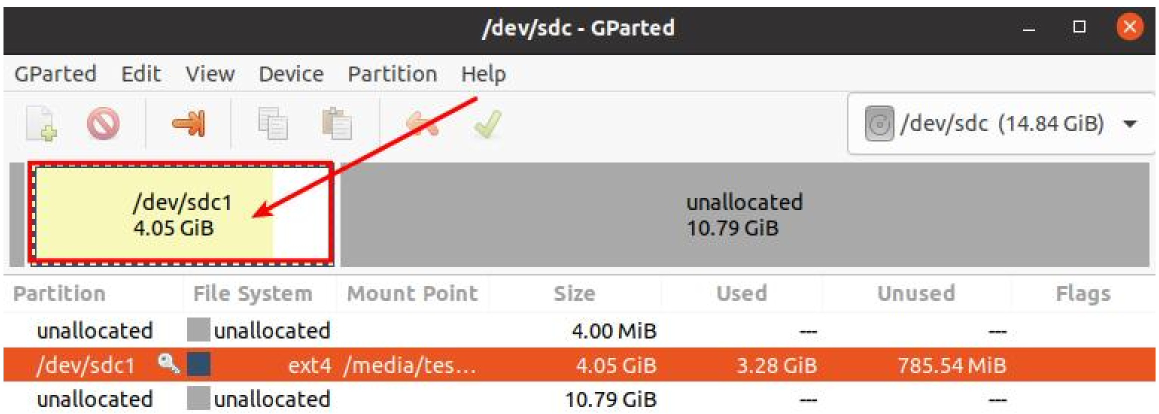
6) Then open gparted

```
test@test:~$ sudo gparted
```

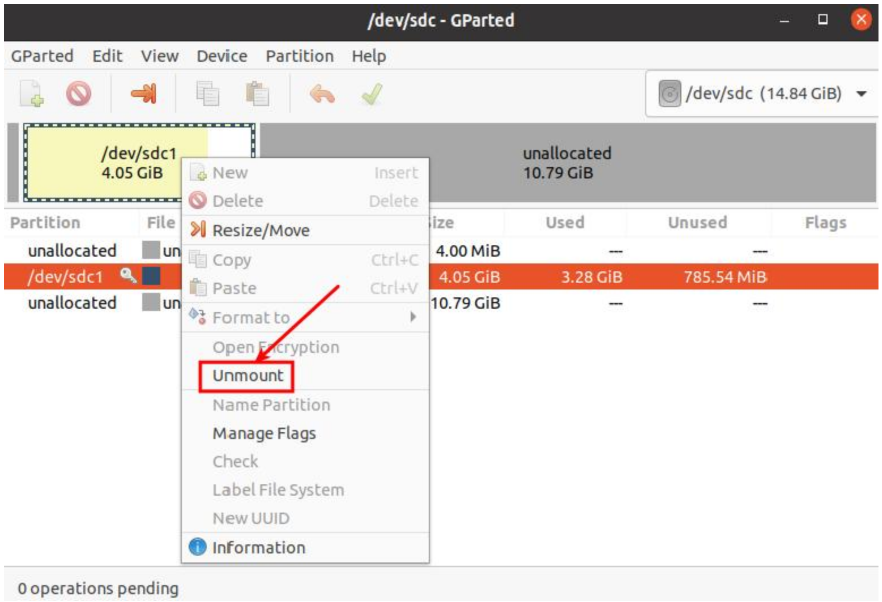
7) After opening gparted, you can select the TF card in the upper right corner, and then you can see the usage of the TF card capacity. The figure below shows the TF card after burning the Linux desktop version system. It can be seen that although the total capacity of the TF card is 16GB (displayed as 14.84GiB in GParted), the rootfs partition (/dev/sdc1) Only 4.05GiB are actually allocated, leaving 10.79GiB unallocated



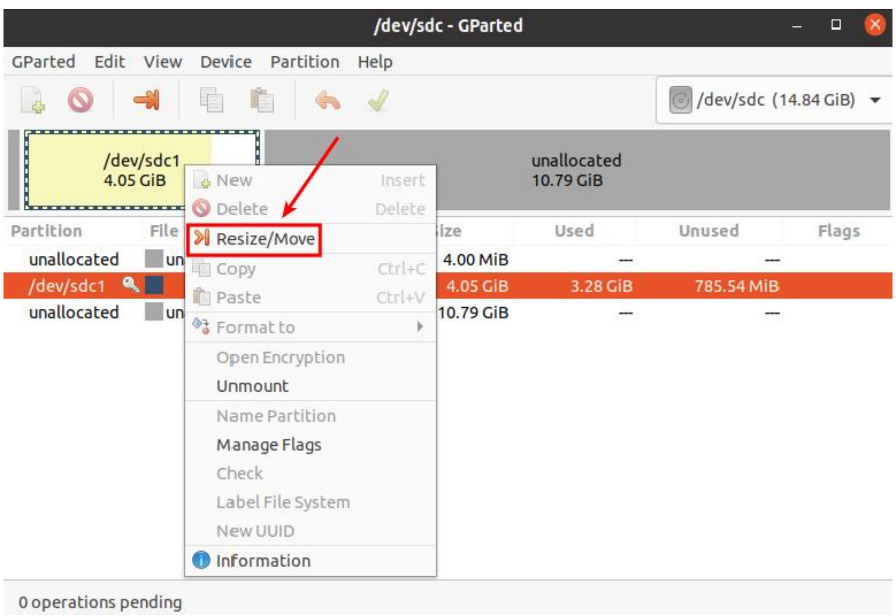
8) Then select the rootfs partition (/dev/sdc1)



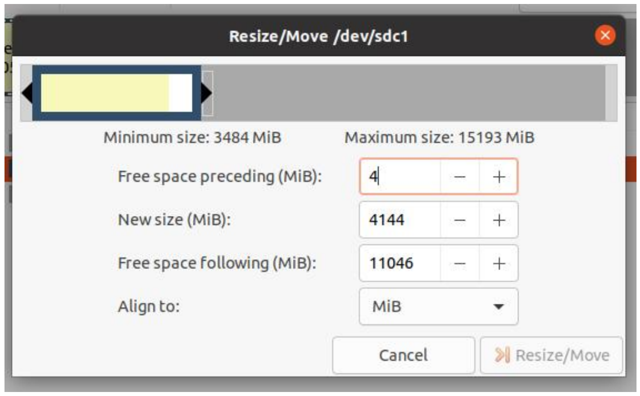
9) Then click the right mouse button to see the operation options shown in the figure below. If the TF card has been mounted, you need to Umount the rootfs partition of the TF card first.



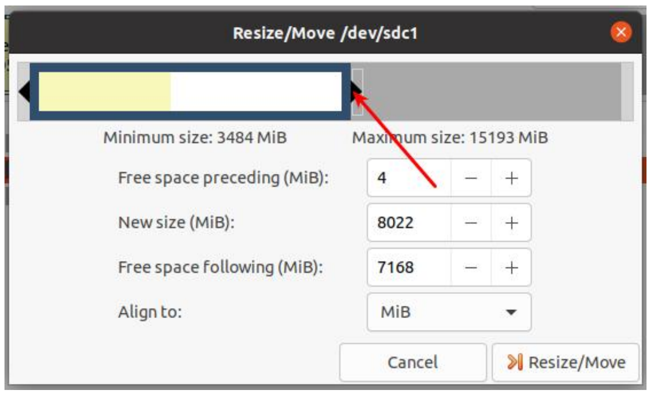
10) Then select the rootfs partition again, right-click, and select **Resize/Move** to start expanding the size of the rootfs partition



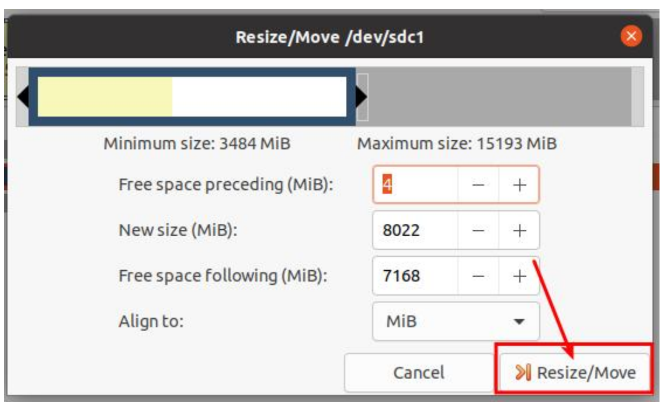
11) After the **Resize/Move** option is turned on, the following setting interface will pop up



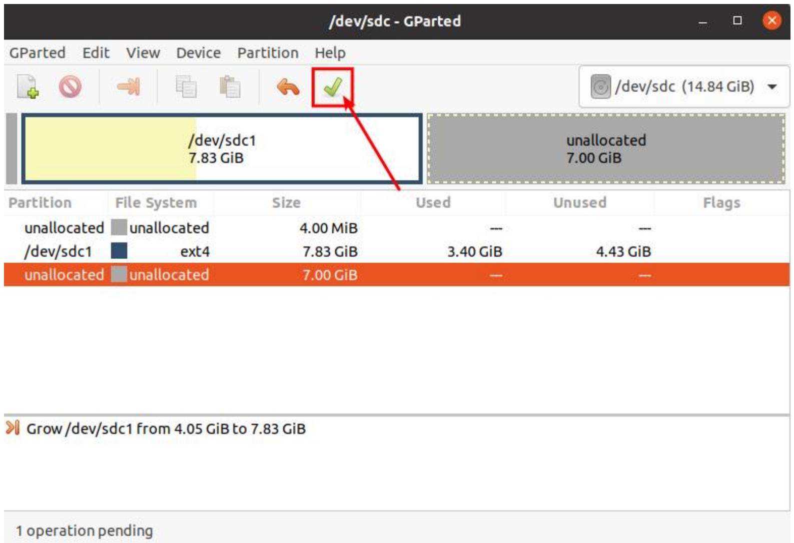
12) Then you can directly drag the position shown in the figure below to set the size of the capacity, or you can set the size of the rootfs partition by setting the number in **New sieze (MiB)**



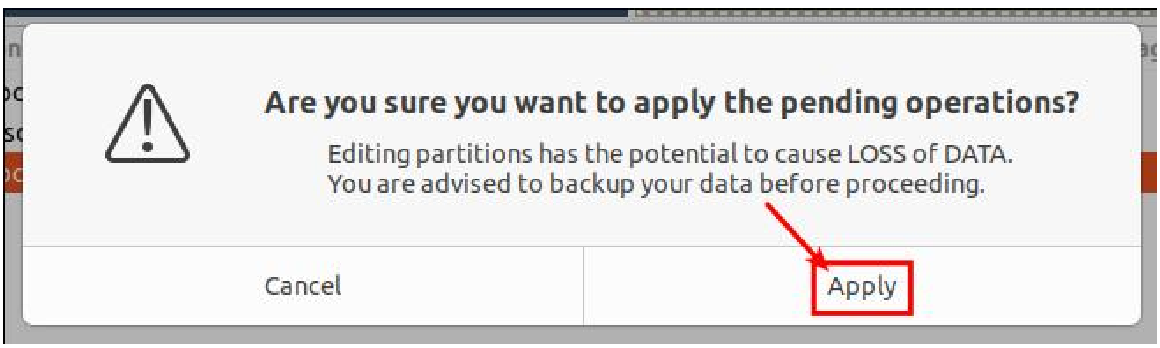
13) After setting the capacity, click **Resize/Move** in the lower right corner



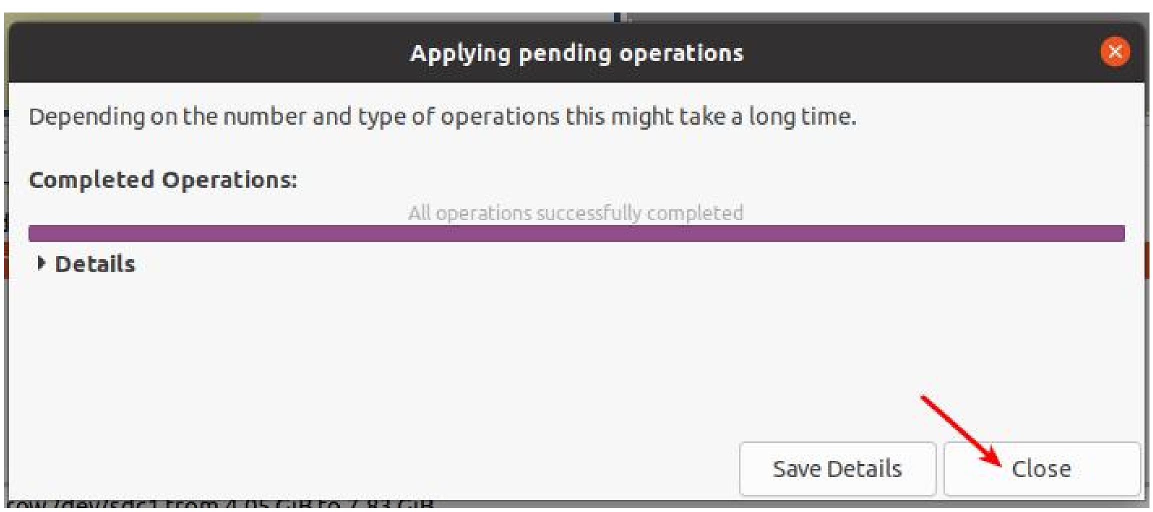
14) After the final confirmation, click the **green** √ as shown in the figure below



15) Then select **Apply**, it will officially start to expand the capacity of the rootfs partition



16) After the expansion is complete, click **Close** to close it



17) Then you can unplug the TF card and insert it into the development board to start it. After entering the Linux system of the development board, if you use the **df -h** command to see that the size of the rootfs partition is the same as the size set before, it means manual operation Expansion succeeded

```
root@orangepi:~# df -h
Filesystem          Size    Used Avail Use% Mounted on
udev                 925M       0   925M    0% /dev
tmpfs                199M    3.2M   196M    2% /run
/dev/mmcblk0p1    7.7G    3.2G    4.4G    42% /
```

3. 6. 4. Method to reduce the capacity of rootfs partition in TF card

After configuring the application program or other development environment in the Linux system of the TF card, if you want to backup the Linux system in the TF card, you can use the method in this section to reduce the size of the rootfs partition first, and then start the backup.

1) First insert the TF card you want to operate in the **Ubuntu computer** (not Windows)

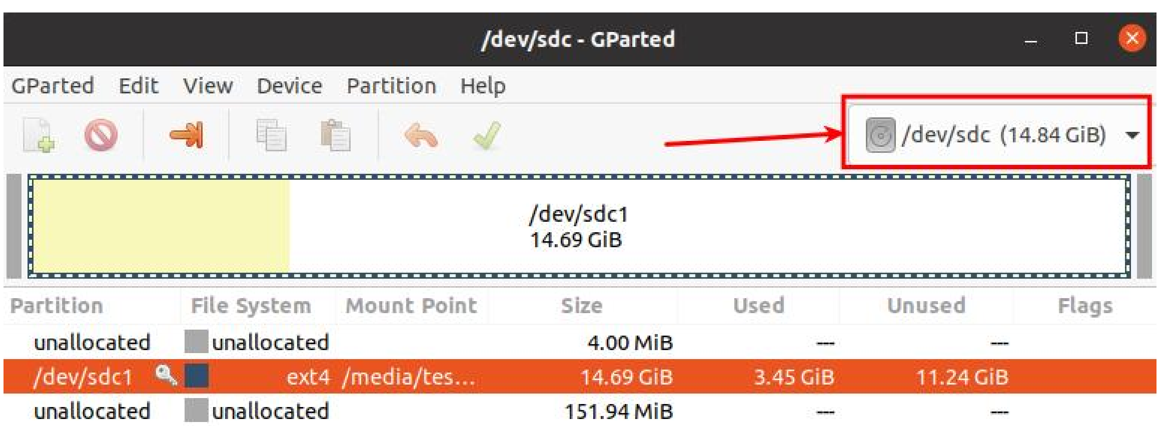
2) Then install the software gparted on the Ubuntu computer

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

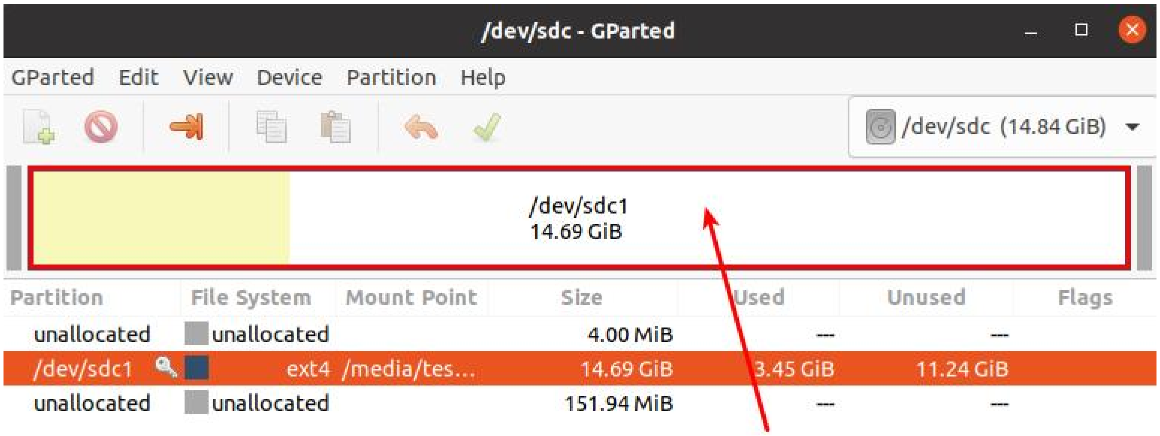
3) Then open gparted

```
test@test:~$ sudo gparted
```

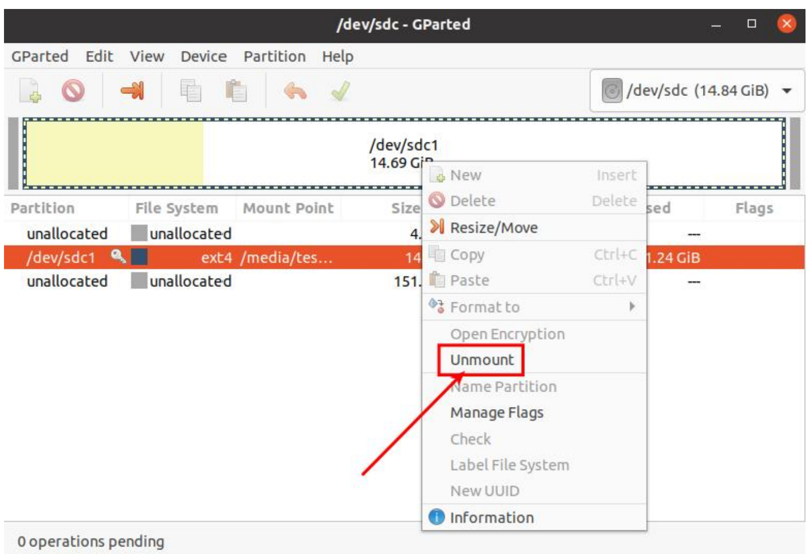
4) After opening gparted, you can select the TF card in the upper right corner, and then you can see the usage of the TF card capacity



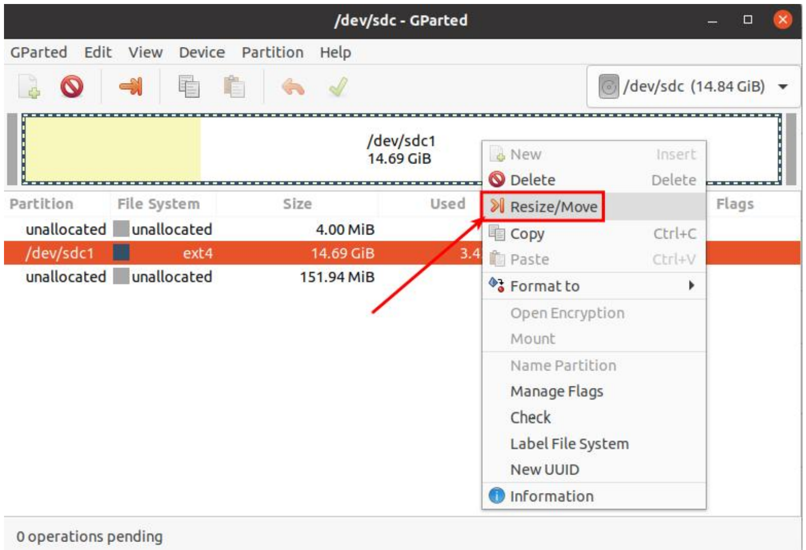
5) Then select the rootfs partition (/dev/sdc1)



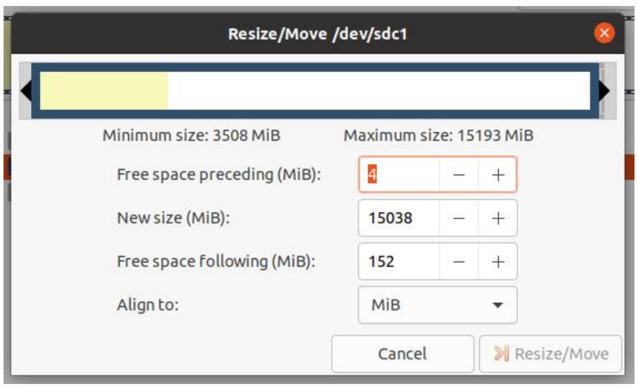
6) Click the right mouse button again to see the operation options shown in the figure below. If the TF card has been mounted, you need to Umount the rootfs partition of the TF card first.



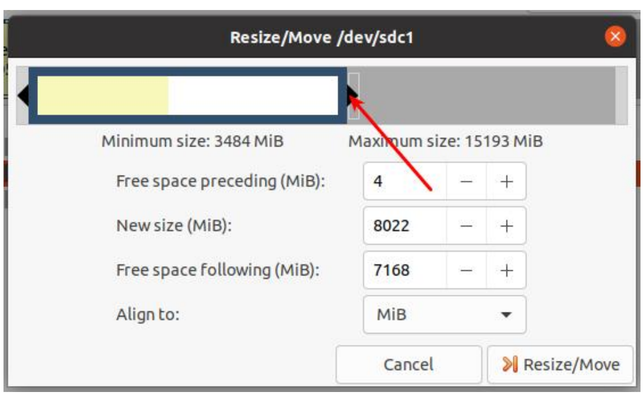
7) Then select the rootfs partition again, right-click, and select **Resize/Move** to start setting the size of the rootfs partition



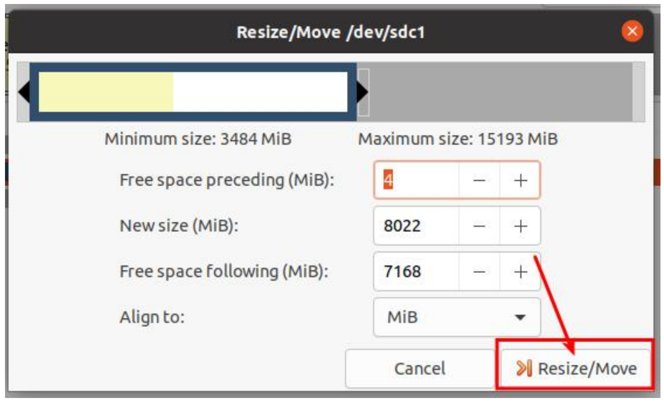
8) After the **Resize/Move** option is turned on, the following setting interface will pop up



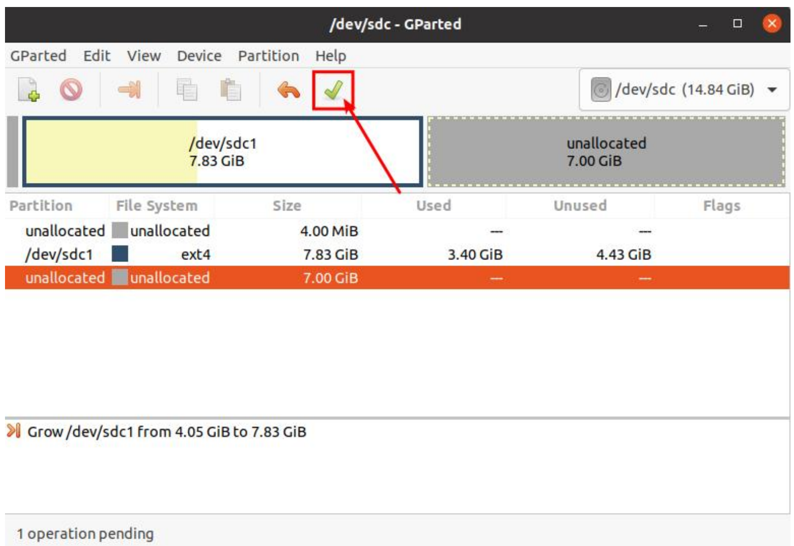
9) Then you can directly drag the position shown in the figure below to set the size of the capacity, or you can set the size of the rootfs partition by setting the number in **New sieze (MiB)**



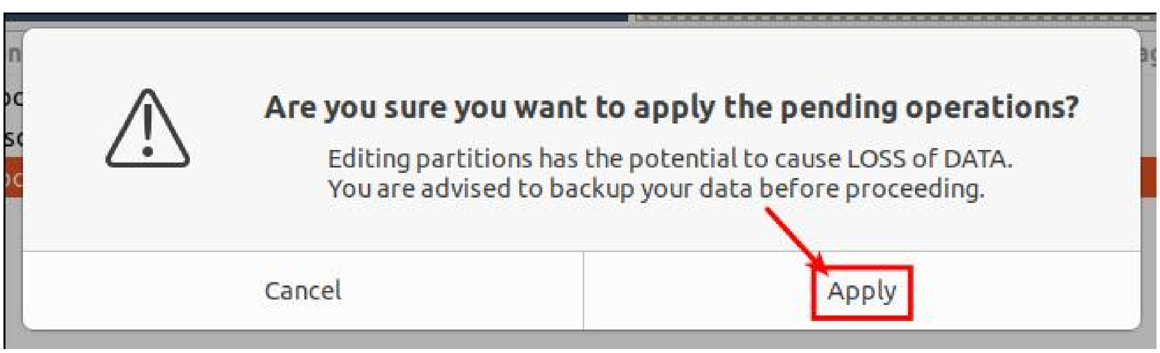
18) After setting the capacity, click **Resize/Move** in the lower right corner



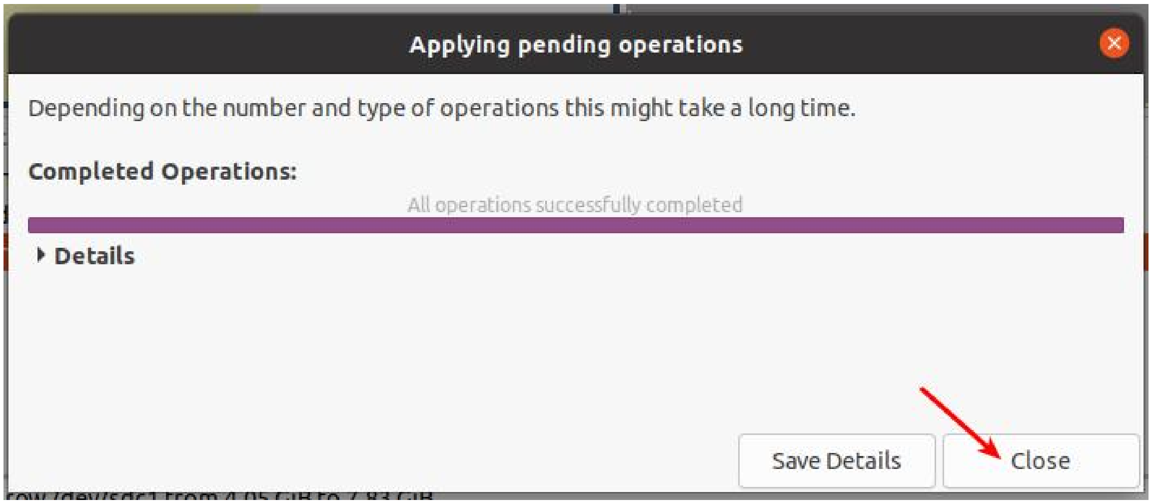
19) After the final confirmation, click the **green** √ as shown in the figure below



20) Then select **Apply**, it will officially start to expand the capacity of the rootfs partition



21) After the expansion is completed, click **Close** to close it



22) Then you can unplug the TF card, and then insert it into the development board to start. After entering the Linux system of the development board, if you use the **df -h** command to see that the size of the rootfs partition is the same as the size set earlier, it means that the size is reduced. capacity success

```
root@orangepi:~# df -h
Filesystem        Size   Used Avail Use% Mounted on
udev              925M      0  925M   0% /dev
tmpfs             199M   3.2M  196M   2% /run
/dev/mmcblk0p1    7.7G   3.2G  4.4G  42% /
```

3. 7. How to modify the linux log level (loglevel)

1) The loglevel of the linux system is set to 1 by default. When using the serial port to view the startup information, the kernel output log is as follows, basically all shielded

```
Starting kernel ...

Uncompressing Linux... done, booting the kernel.

Orange Pi 3.0.0 Buster ttyFIQ0

orangepi login:
```

2) When there is a problem with the Linux system startup, you can use the following

method to modify the value of loglevel, so as to print more log information to the serial port display, which is convenient for debugging. If the Linux system fails to start and cannot enter the system, you can insert the TF card into the Ubuntu PC through the card reader, and then directly modify the configuration of the Linux system in the TF card after mounting the TF card in the Ubuntu PC. Insert the TF card into the development board to start

```
root@orangepi4-lts:~# sed -i "s/verbosity=1/verbosity=7/" /boot/orangepiEnv.txt
root@orangepi4-lts:~# sed -i "s/console=both/console=serial/" /boot/orangepiEnv.txt
```

3) The above commands actually set the variables in **/boot/orangepiEnv.txt.** After setting, you can open **/boot/orangepiEnv.txt** to check.

```
root@orangepi4-lts:~# vim /boot/orangepiEnv.txt
verbosity=7
console=serial
```

4) Then restart the development board, the output information of the kernel will be printed to the serial port output

3. 8. Network connection test

3. 8. 1. Ethernet port test

1) First, insert the network cable into the Ethernet interface of the development board, and ensure that the network is unblocked

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

3) The command to view the IP address is as follows

```
orangepi@orangepi:~$ ip addr show eth0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 5e:ac:14:a5:93:b3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.16/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
        valid_lft 259174sec preferred_lft 259174sec
    inet6 240e:3b7:3240:c3a0:e269:8305:dc08:135e/64 scope global dynamic
```

```
noprefixroute
       valid_lft 259176sec preferred_lft 172776sec
    inet6 fe80::957d:bbbd:4928:3604/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

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

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

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

3. If there is no debugging serial port and no HDMI display, you can also view the IP address of the development board network port through the management interface of the router. However, this method often fails to see the IP address of the development board. If you can't see it, the debug method looks like this:

A）First check whether the Linux system has been started normally. If the green light of the development board is on or flashing, it is generally started normally. If the green light is not on or flashing, the system has not started normally;

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

C）Try another router (the router has encountered many problems, such as the router cannot assign an 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 an HDMI display or use the debug serial port to view the IP address。

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

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

```
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. 8. 2. WIFI connection test

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

3. 8. 2. 1. The server version image is connected to WIFI through the command

When the development board is not connected to the Ethernet, not connected to the HDMI display, and 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), it 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 to the network cable, **you can log in to the Linux system remotely 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 HDMI display terminal

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

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

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

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

a. **wifi_name** Need to be changed to the name of the WIFI hotspot you want to connect to

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

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

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

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

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

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

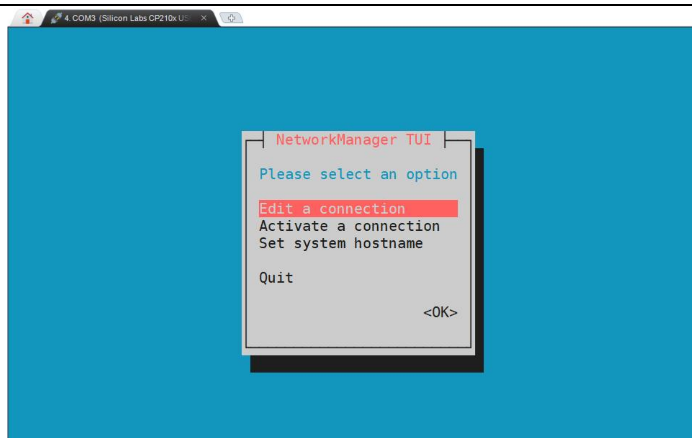
3.8.2.2. The server version image connects to WIFI through a graphical method

1) First log in to the linux system, there are the following three ways
   a. If the development board is connected to the network cable, you can log in to the **Linux system remotely 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 serial port software, and the graphical interface cannot be displayed using minicom)
   c. If the development board is connected to the HDMI display, you can log in to the linux system through the HDMI display terminal

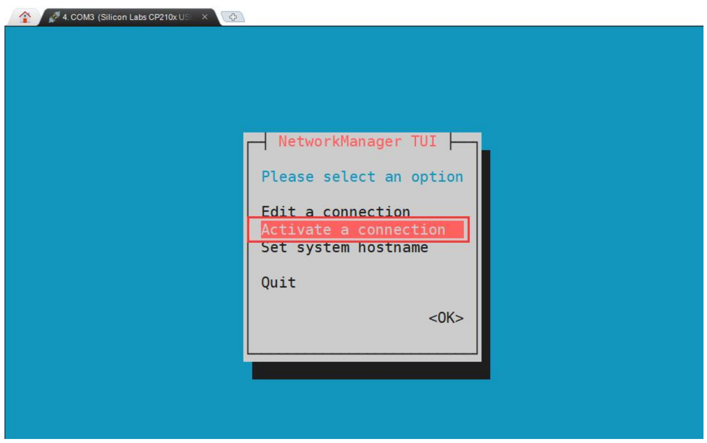
2) Then enter the nmtui command in the command line to open the wifi connection interface

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

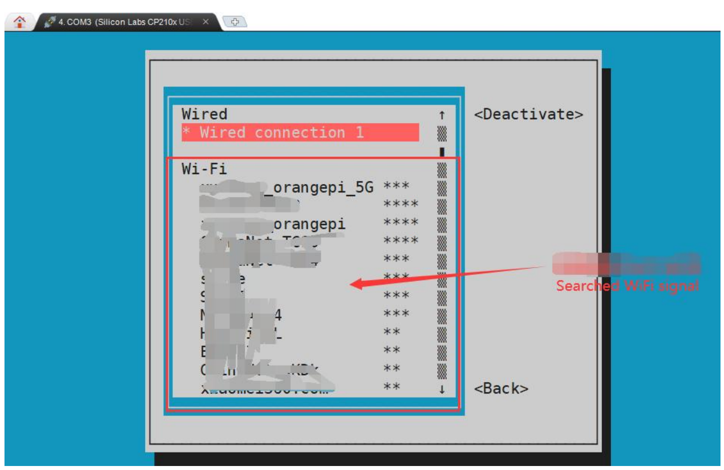
3) Enter the nmtui command to open the interface as shown below



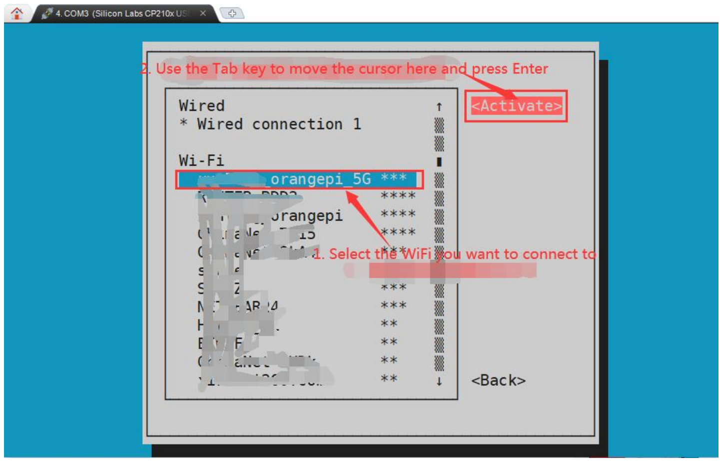
4) Select **Activate a connect** and press Enter



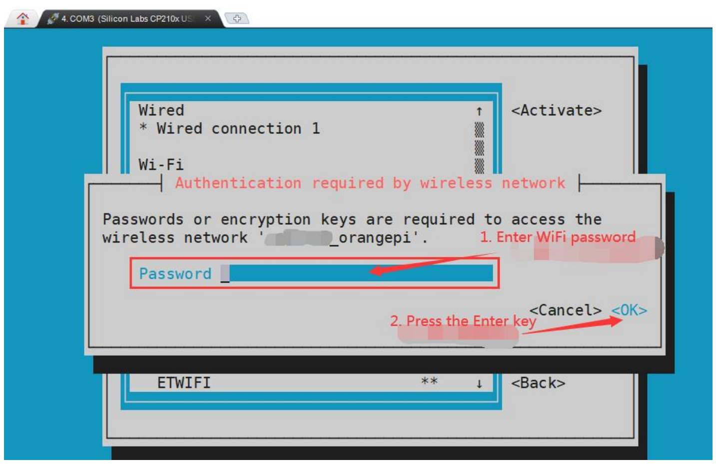
5) Then you can see all the searched WIFI hotspots



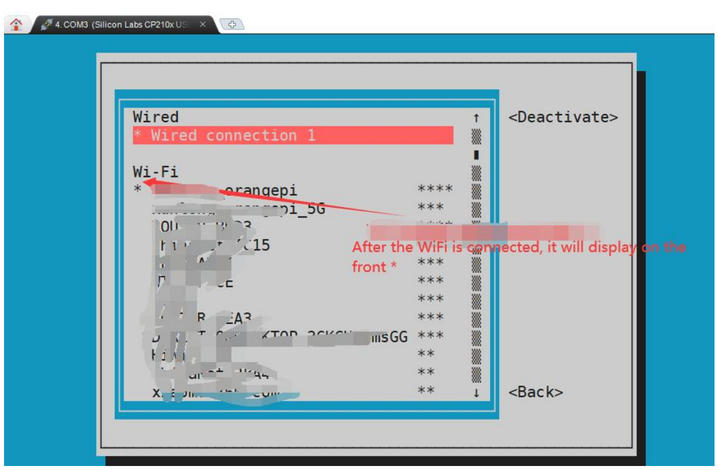
6) Select the WIFI hotspot you want to connect, then use the Tab key to position the cursor to **Activate** and press Enter



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



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



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

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

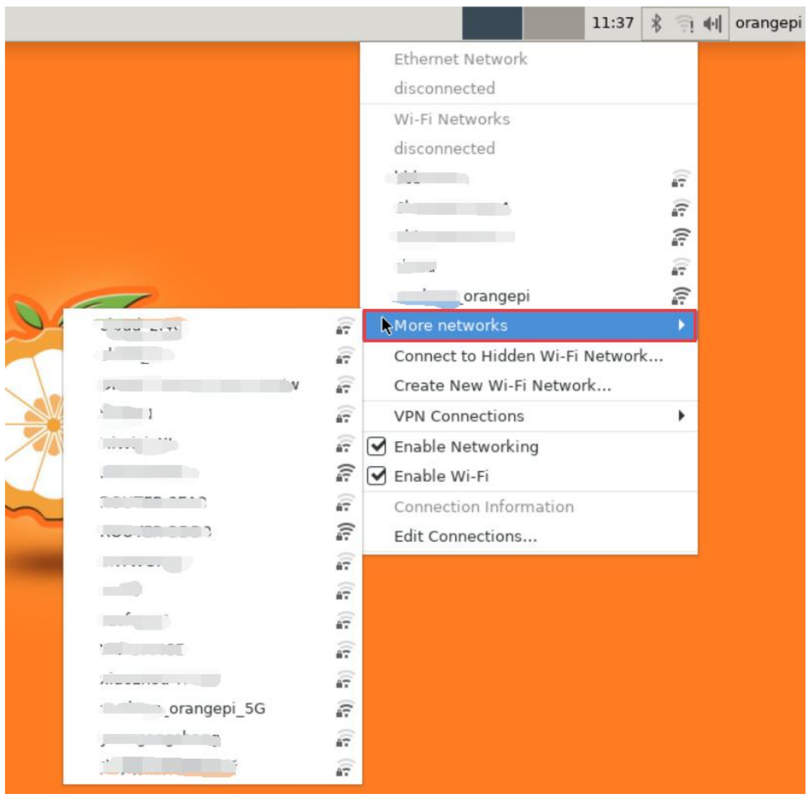
10) Use the **ping** command to test the connectivity of the wifi network. The **ping** command can be interrupted by the **Ctrl+C** shortcut key

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

3. 8. 2. 3. Test method for desktop image

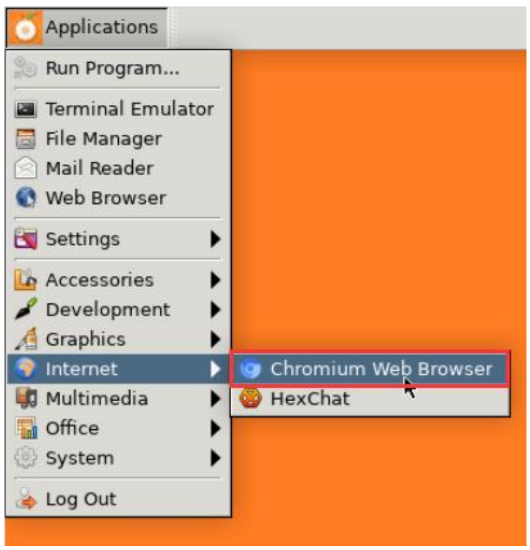
1) Click the network configuration icon in the upper right corner of the desktop (please do not connect the network cable when testing WIFI)

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

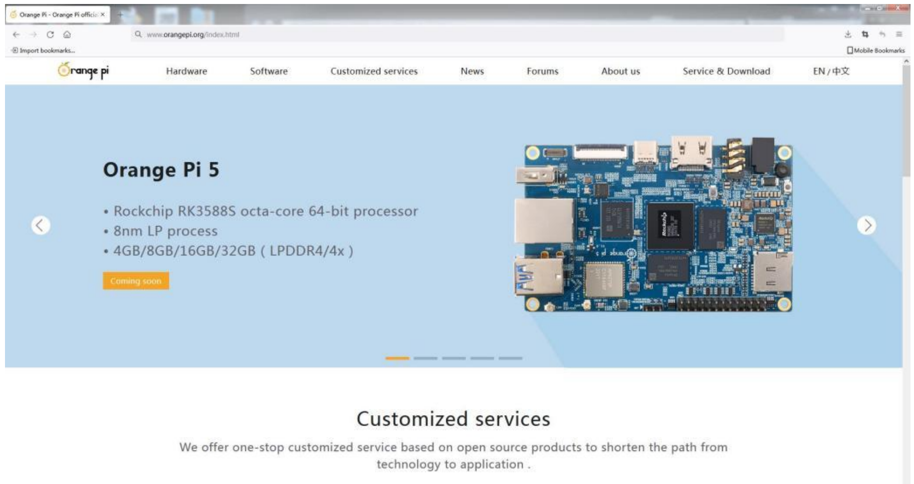


3) Then enter the password of the WIFI hotspot, and then click **Connect** to start connecting to the WIFI

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) After opening the browser, if you can see the page of the Orange Pi website, or you can open other pages, it means the WIFI connection is normal



3. 8. 3. How to use Hostapd to establish a WIFI hotspot

First, please make sure that the development board is connected to the network cable and can access the Internet normally. Because the process of setting up Hostapd to establish a WIFI hotspot needs to download some software packages from the Internet. If the development board cannot access the Internet through the wired network port, the installation will fail.

If the network cable is not connected, when other network devices (such as mobile phones or computers) are connected to the WIFI hotspot launched by the development board, they cannot access the external network normally (such as the mobile phone browser cannot open the web page).

After using Hostapd to establish a WIFI hotspot (note that after setting it up), if you do not need to access the external network, but only need to connect to the WIFI hotspot launched by the development board, then it is also possible not to connect the network cable.

In addition, before setting up Hostapd, please make sure that WIFI is not connected to the network, otherwise, it will prompt that WIFI is in use and cannot set up Hostapd normally.

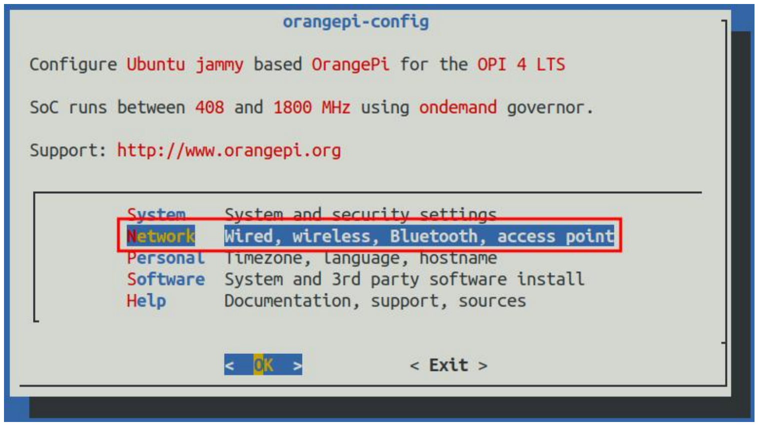
1) First update the software source index of the system

```
root@orangepi:~$ sudo apt-get update
```

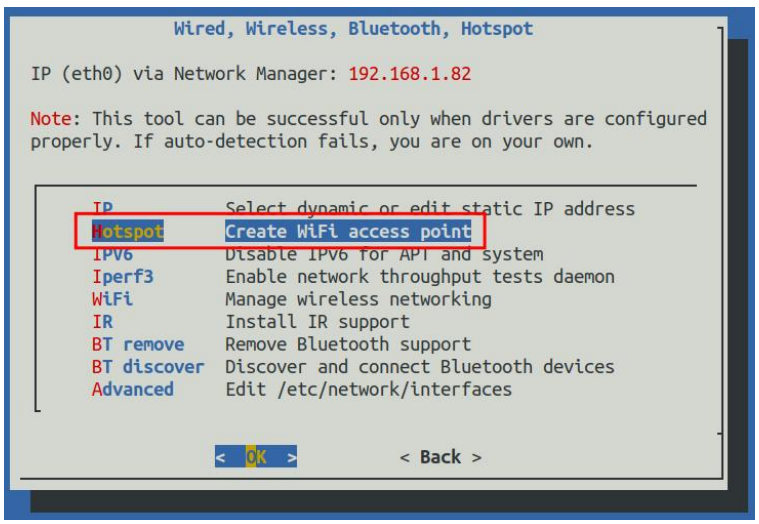
2) Then enter the **orangepi-config** command in the terminal, **remember to add sudo permissions**

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

3) The interface after orangepi-config is opened is shown in the figure below, select the **Network** option to enter the network-related settings interface



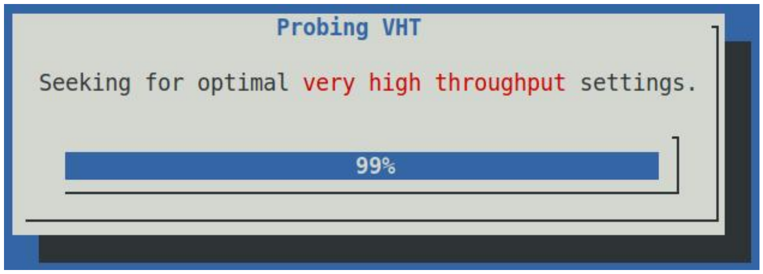
4) Then select **Hotspot Create WiFi access point** option to start setting up Hotspot



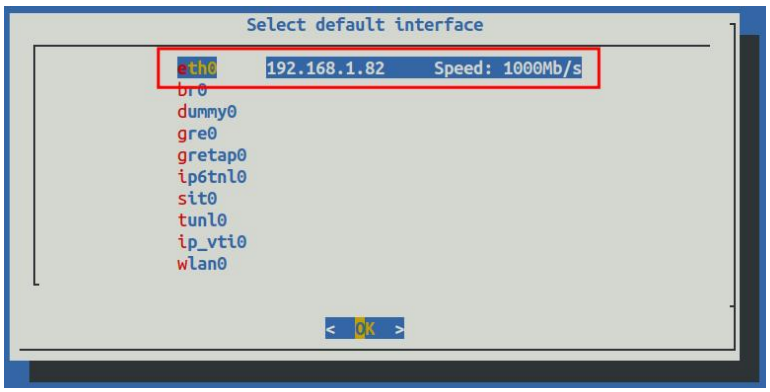
5) When the following selection box pops up, select wlan0

Note that this checkbox does not appear in Debian Buster.

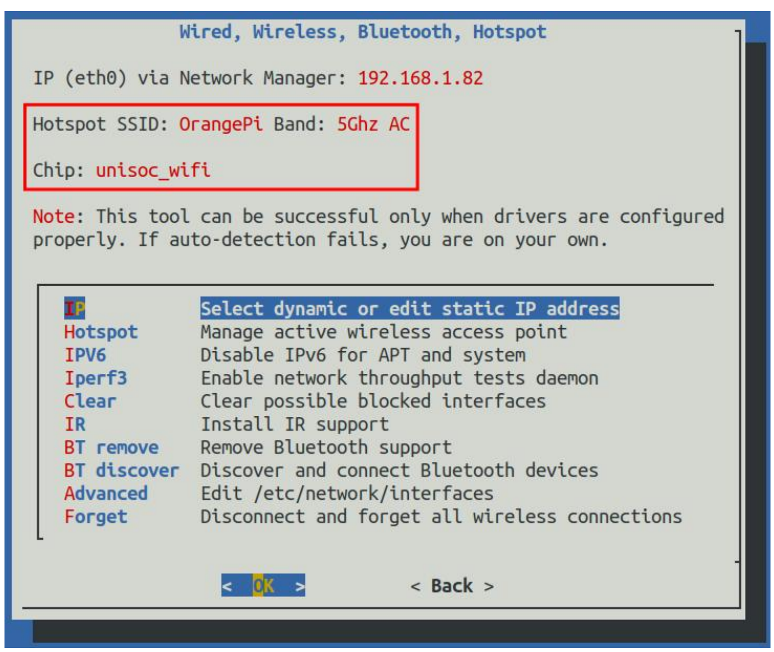
6) Then orangepi-config will start a series of settings, just wait patiently at this time



7) After waiting for a while, the following selection box will pop up, please select the first **eth0**



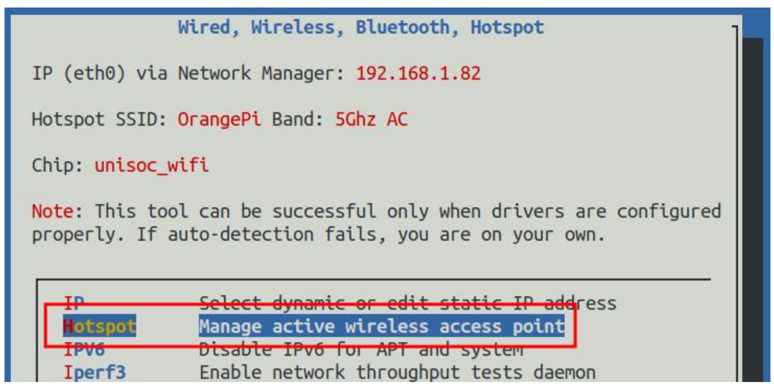
8) After all settings are completed, if orangepi-config displays the following interface, it means that Hostapd is set correctly



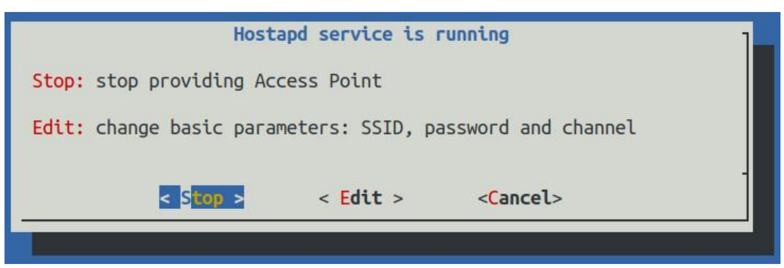
9) The name of the WIFI hotspot set by Hotspot is: **OrangePi** by default, and the password is: **12345678**. If everything is normal, the mobile phone or computer can search for the WIFI hotspot named OrangePi. Hostapd is set up correctly. The following figure is a schematic diagram of the WIFI hotspot emitted by the mobile phone connected to the development board:

10) After Hotspot is set up, open Hostapd in orangepi-config to configure it

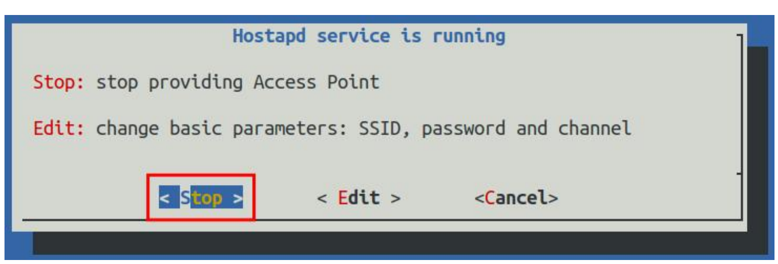
a. First select **Hotspot** in config-config



b. Then you can see the following selection interface

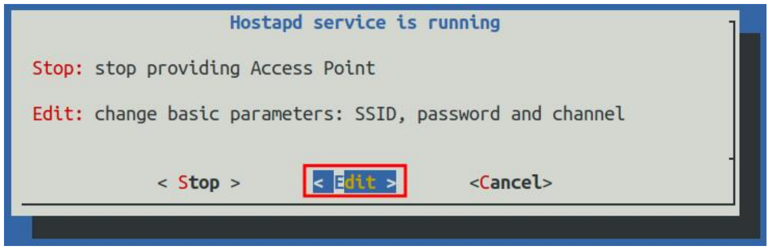


c. Select Stop to **stop** the Hostapd service

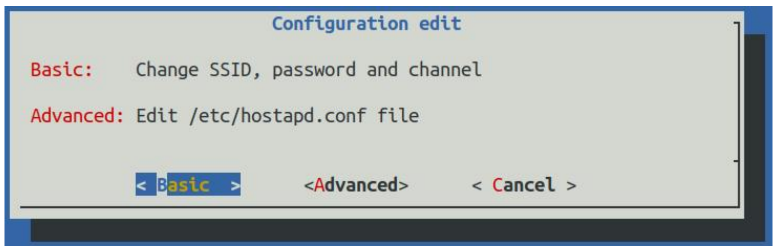


d. Select **Edit** to edit the configuration of Hostapd

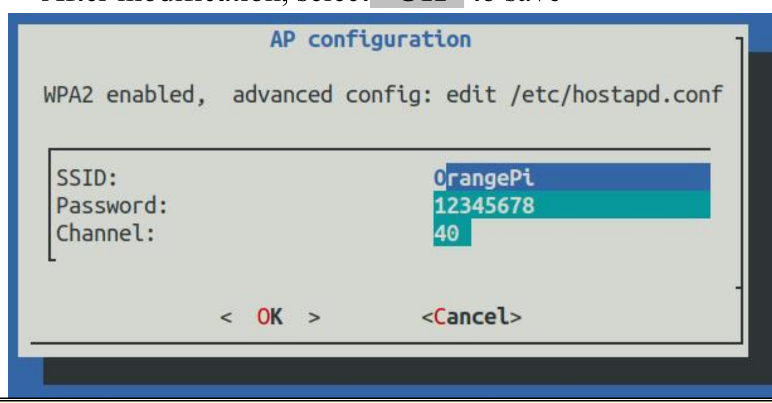
a) The **Edit** option location of Hostapd is shown in the figure below



b) The **Edit** option is opened as shown below

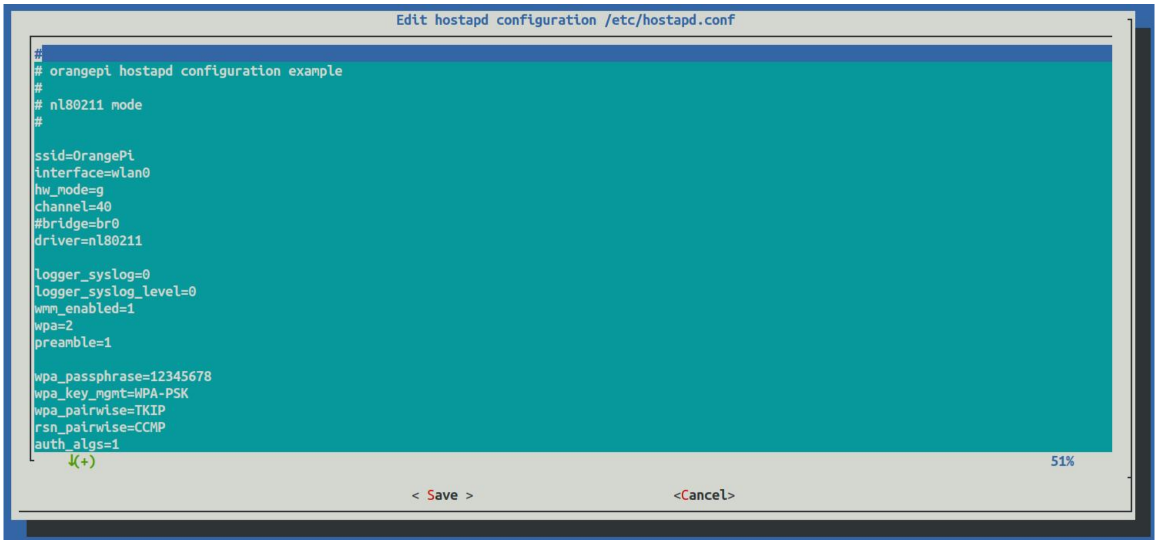


i. Select **Basic** to modify the name and password of the WIFI hotspot. After modification, select **<OK>** to save



Note: If you want to change the password, the changed password must not be less than 8 characters, otherwise the Hostapd service will not work properly.

ii. Select **Advanced** to directly modify the name and password of the WIFI hotspot and other configurations in the hostapd configuration file **/etc/hostapd.conf**. After modification, select **<Save>** to save it.



Note: If you want to change the password, the changed password must not be less than 8 characters, otherwise the Hostapd service will not work properly

3. 8. 4. How to set static IP address

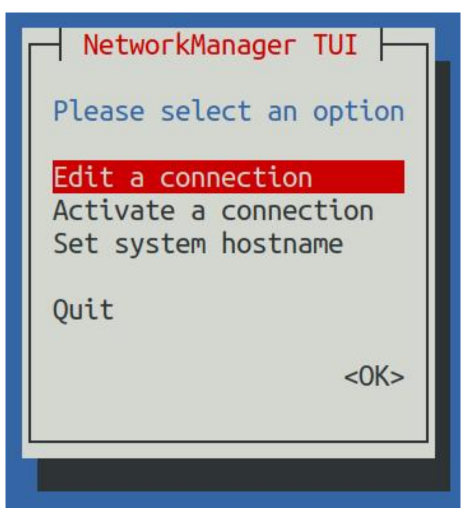
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file。

3. 8. 4. 1. Using the nmtui command to set a static IP address

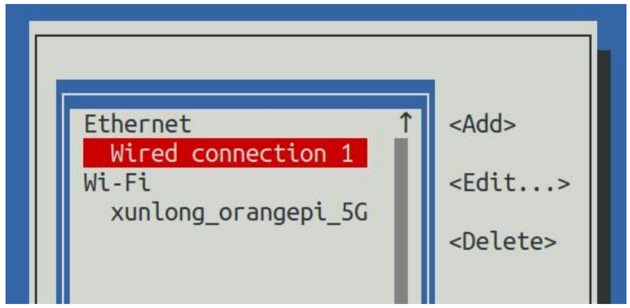
1) First run the nmtui command

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

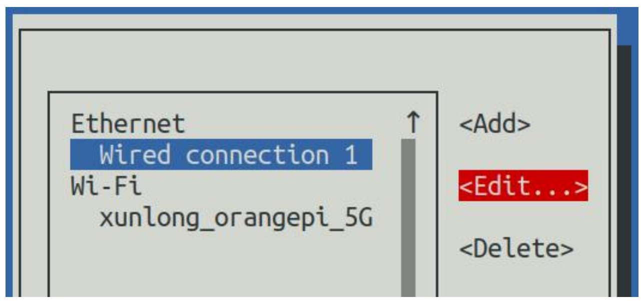
2) Then select **Edit a connection** and press enter



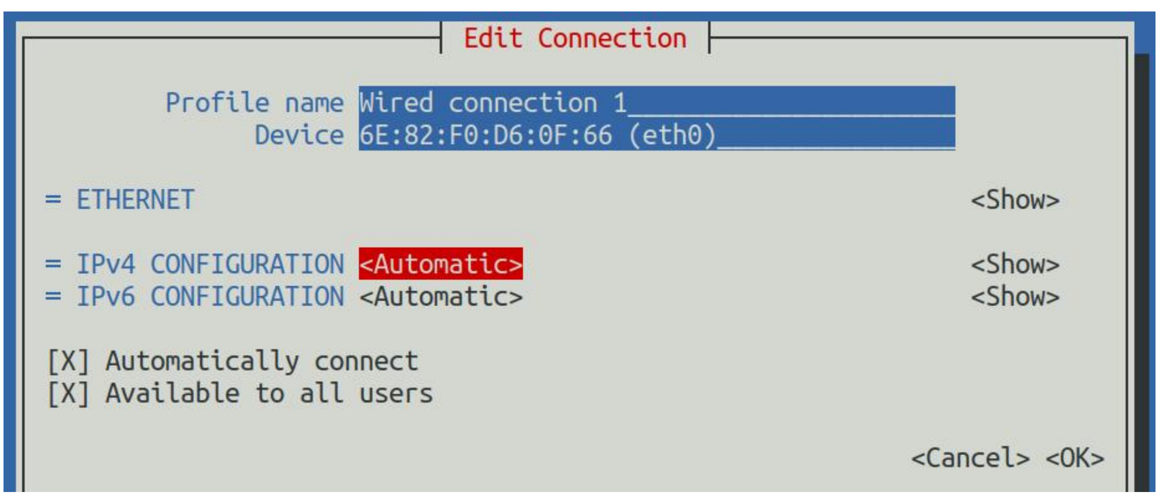
3) Then select the network interface that needs to set a static IP address, such as setting the static IP address of the **Ethernet interface** and select **Wired connection 1**.



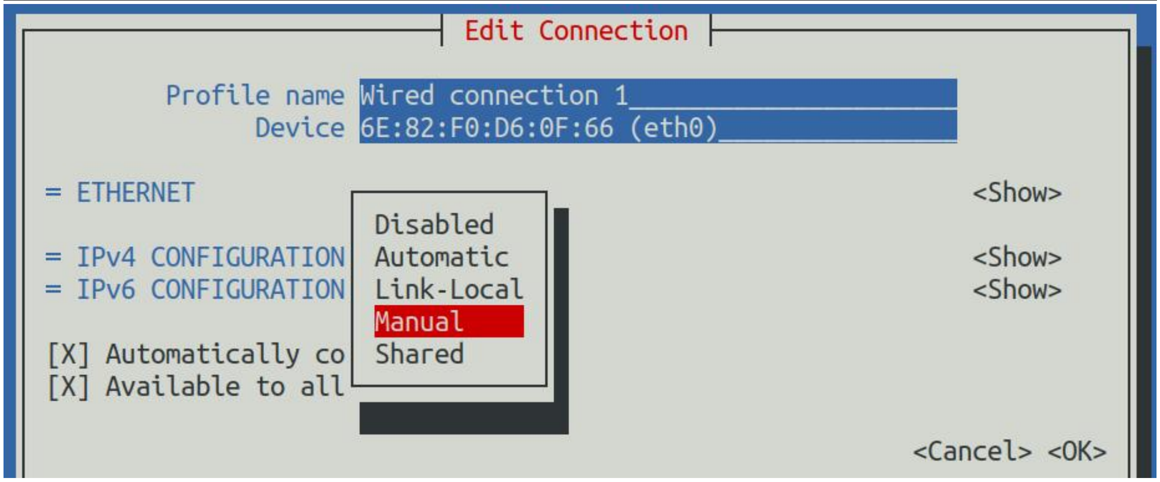
4) Then select **Edit** by **Tab** key and press Enter



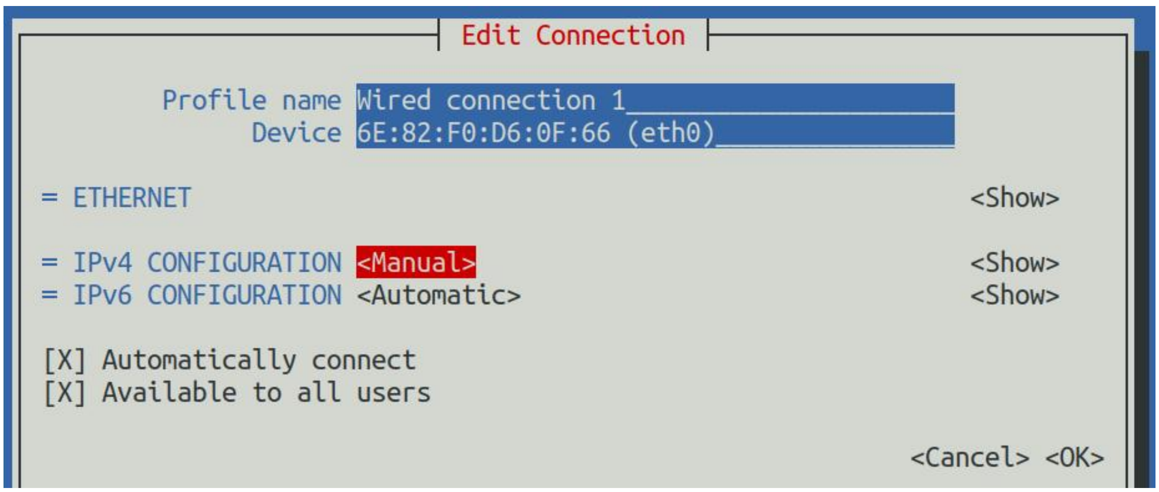
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4



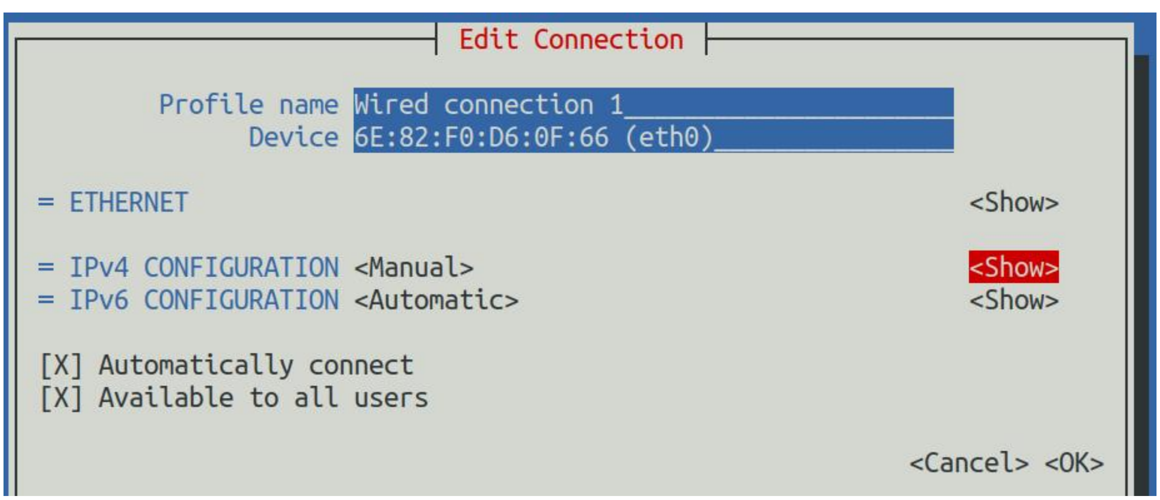
6) Then press Enter, use the up and down arrow keys to select **Manual**, then press Enter to confirm



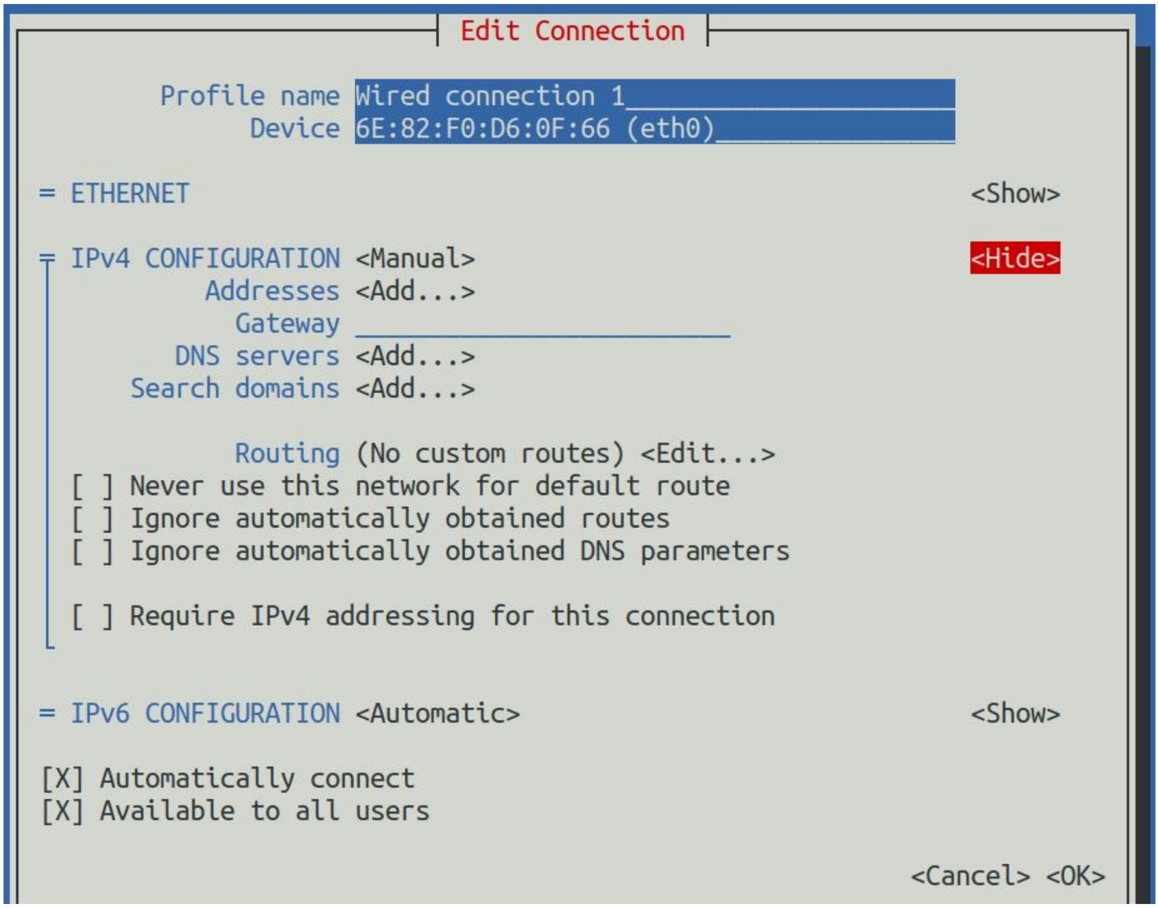
7) The display after selection is as shown below



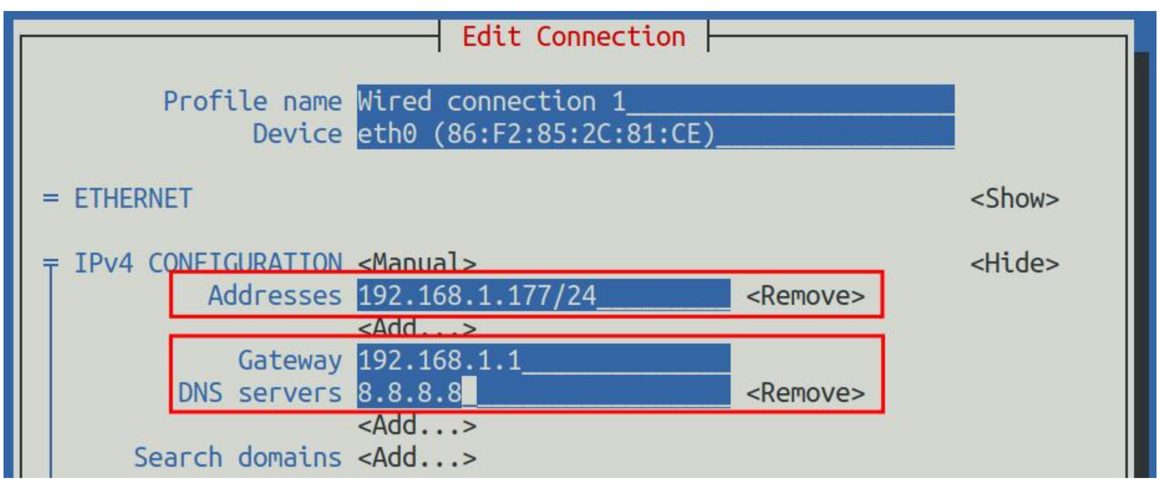
8) Then move the cursor to **<Show>** by Tab key



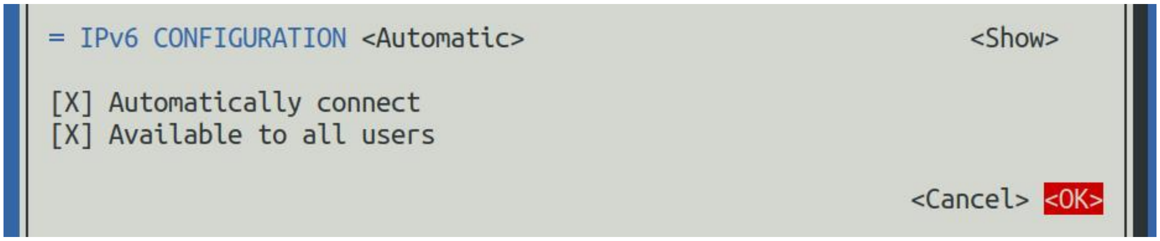
9) Then press Enter, the following setting interface will pop up after pressing Enter



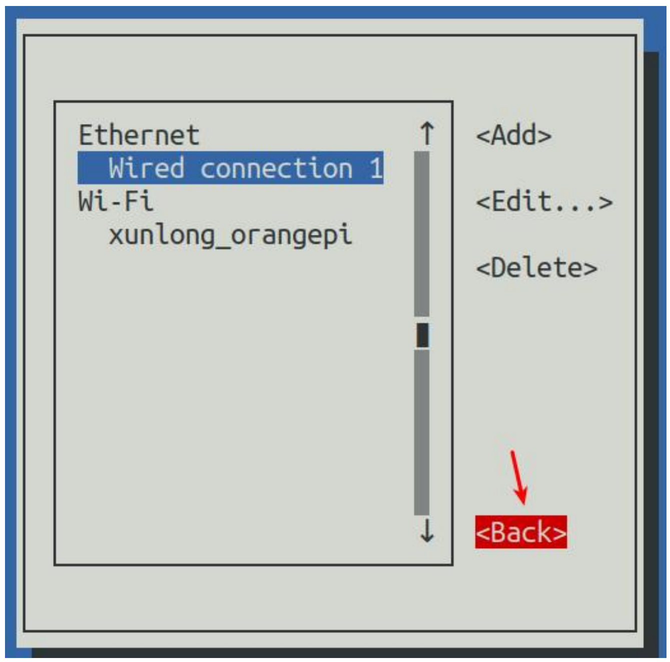
10) Then you can set the IP address (Addresses), gateway (Gateway) and DNS server address in the location shown in the figure below (there are many other setting options, please explore by yourself), **please set according to your specific needs, The value set in the image below is just an example**



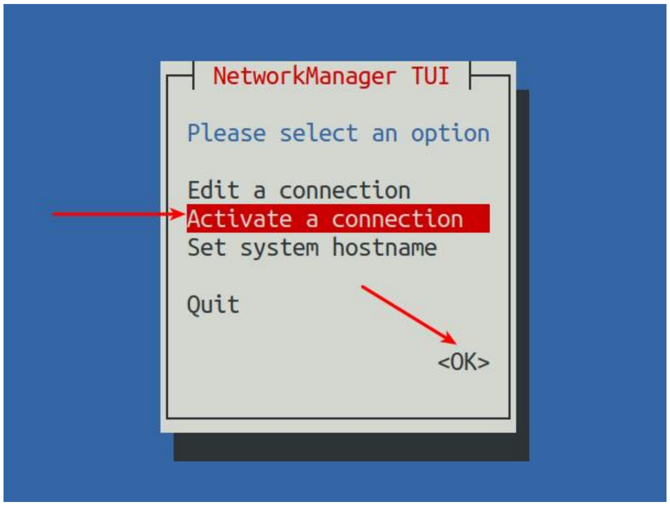
11) After setting, move the cursor to **<OK>** in the lower right corner, then press Enter to confirm



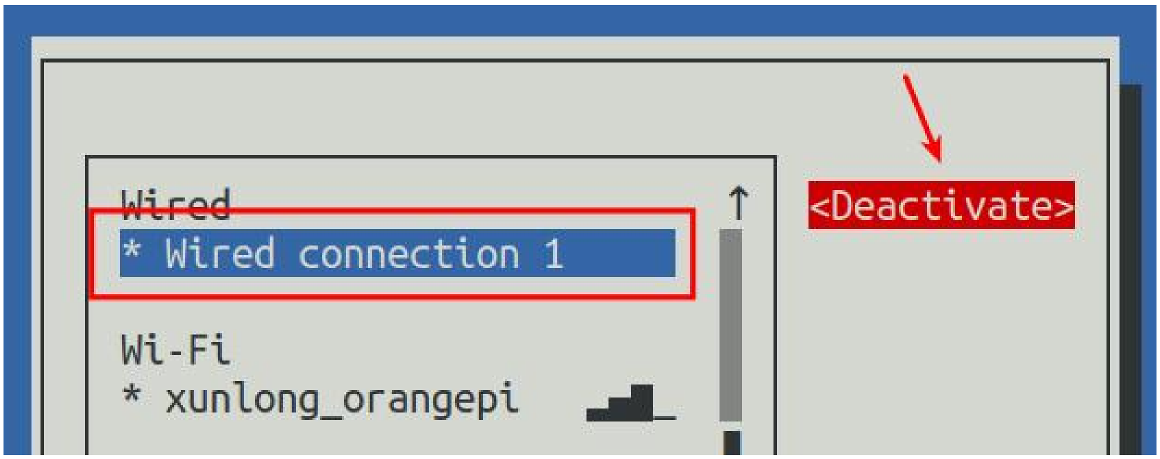
12) Then click **<Back>** to return to the previous selection interface



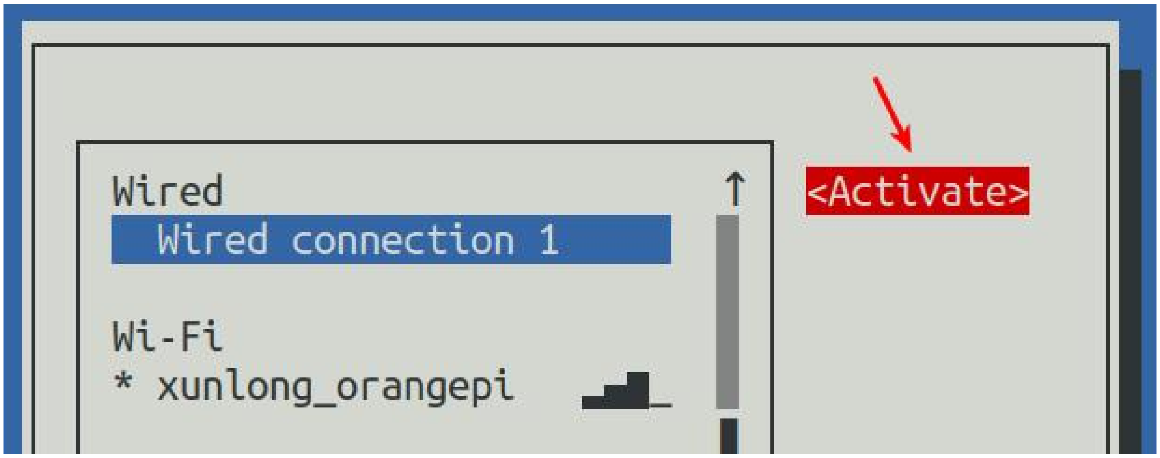
13) Then select **Activate a connection**, move the cursor to **<OK>**, and finally click Enter



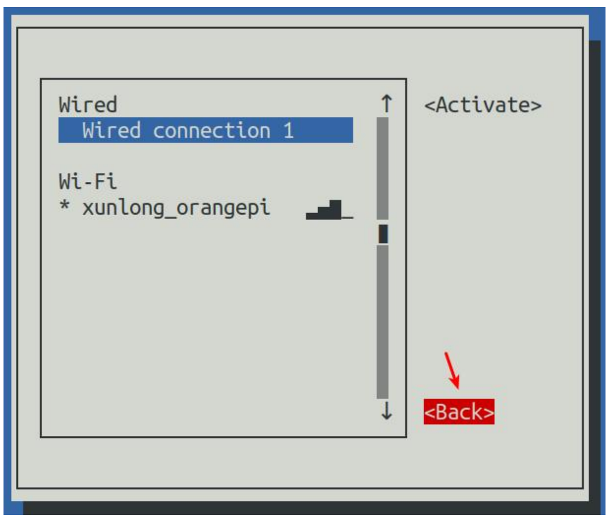
14) Then select the network interface to be set, such as **Wired connection 1**, then move the cursor to **<Deactivate>**, and press Enter to disable **Wired connection 1**

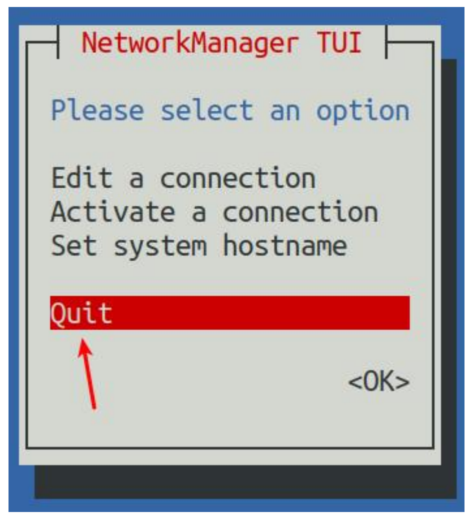


15) Then please do not move the cursor, and then press the Enter key to re-enable **Wired connection 1**, so that the static IP address set earlier will take effect



16) nmtui Then 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 become the static IP address set earlier

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

```
        valid_lft forever preferred_lft forever
```

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

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

3. 8. 4. 2. Using nmcli command to set 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 the WIFI, please connect the WIFI first**, and then start to set the static IP address

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

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

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

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

3) Then enter the following command,

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

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

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

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

4) Then restart the linux system

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

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

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

3. 8. 5. Set up the method of automatically connecting to the network when the Linux system starts up for the first time

The development board has an Ethernet port. If you want to remotely log in to the Linux system of the development board through the Ethernet port, you only need to plug a network cable that can access the Internet normally to the Ethernet port. After the Linux system is started, it will automatically pass DHCP to the Ethernet port. Assign an IP address, and then we can obtain the IP address of the Ethernet port through the HDMI screen, serial port or by viewing the background

of the router, and then we can log in to the Linux system remotely.

The development board also has wireless WIFI. If you want to remotely log in to the Linux system of the development board through WIFI, you need to remotely log in to the Linux system through the IP address of the Ethernet port through ssh and then connect to the WIFI through commands, or use commands on the HDMI screen or serial port. Connect to WIFI.

However, if there is no HDMI screen and serial port module, although there is a network cable, the IP address of the development board cannot be viewed through the router background. Or if there is no HDMI screen, serial port module and network cable, and only WIFI can be connected, you can use the method described in this section to automatically connect to WIFI and also set the static IP address of the WIFI or automatically set the static IP address of the Ethernet port.

To use the method in this section, you first need to prepare a Linux system machine. For example, a computer or virtual machine with Ubuntu system installed.

Why do you need a Linux system machine? Because the root file system of the Linux system of the development board burned in the TF card is in ext4 format, the Linux system machine can mount it normally, and then modify the configuration file.

If you want to modify it in the Windows system, you can use the Paragon ExtFS for Windows software. Since this software needs to be paid, and there is no similar free software that is easy to use, it will not be demonstrated here.

In addition, if there is any problem with Paragon ExtFS for Windows, please solve it by yourself, we will not answer any questions.

1) First burn the Linux image of the development board you want to use into the TF card, and then use the card reader to insert the TF card with the Linux image of the development board into the machine installed with the Linux system (such as the one with the Ubuntu system installed). Computer, the following is an example of Ubuntu computer to demonstrate)

2) When the TF card is inserted into the Ubuntu computer, the Ubuntu computer will generally automatically mount the partition of the Linux root file system in the TF card, which can be known from the following

commands:**/media/test/d17d5e4f-ae41-4554-a727-bb5c9c94134f** It is the path to mount the Linux root file system in the TF card (**the path name is based on what you actually see, not necessarily the same**)

```
test@test:~$ df -h | grep "media"
/dev/sdd1    1.4G    1.2G    167M    88%
/media/test/d17d5e4f-ae41-4554-a727-bb5c9c94134f
test@test:~$ ls /media/test/d17d5e4f-ae41-4554-a727-bb5c9c94134f
bin   boot   dev   etc   home   lib   lost+found   media   mnt   opt   proc   root   run
sbin   selinux   srv   sys   tmp   usr   var
```

3) Then enter the **/boot** directory of the Linux system burned in the TF card

```
test@test:~$ cd /media/test/d17d5e4f-ae41-4554-a727-bb5c9c94134f/boot
```

4) Then copy the **orangepi_first_run.tx**t.template as **orangepi_first_run.txt**. Through the orangepi_first_run.txt configuration file, you can set the development board to automatically connect to a WIFI hotspot when the Linux system starts for the first time, or you can set the WIFI or Ethernet port. static IP address

```
test@test:/media/test/d17d5e4f-ae41-4554-a727-bb5c9c94134f/boot$ sudo cp \
orangepi_first_run.txt.template orangepi_first_run.txt
```

5) You can open the orangepi_first_run.txt file through the following command, and then you can view and modify the content

```
test@test:/media/test/d17d5e4f-ae41-4554-a727-bb5c9c94134f/boot$ sudo vim \
orangepi_first_run.txt
```

6) Instructions for using variables in the orangepi_first_run.txt file
   a. **FR_general_delete_this_file_after_completion** variable is used to set whether to delete the orangepi_first_run.txt file after the first startup. The default value is 1, that is, delete. If it is set to 0, orangepi_first_run.txt will be renamed to orangepi_first_run.txt after the first startup. old, generally keep the default value
   b. **FR_net_change_defaults** variable is used to set whether to change the default network settings, this must be set to 1, otherwise all network settings will not take effect
   c. **FR_net_ethernet_enabled** variable is used to control whether to enable the configuration of the Ethernet port. If you need to set the static IP address of the

Ethernet port, please set it to 1

d. **FR_net_wifi_enabled** variable is used to control whether to enable the WIFI configuration. If you need to set the development board to automatically connect to the WIFI hotspot, you must set it to 1. Also, please note that if this variable is set to 1, the Ethernet port settings will be invalid. That is to say, WIFI and Ethernet ports cannot be set at the same time (why, because there is no need...)
e. **FR_net_wifi_ssid** variable is used to set the name of the WIFI hotspot you want to connect to
f. **FR_net_wifi_key** variable is used to set the password of the WIFI hotspot you want to connect to
g. **FR_net_use_static** variable is used to set whether to set the static IP address of the WIFI or Ethernet port
h. **FR_net_static_ip** variable is used to set the address of the static IP, please set it according to your actual situation
i. **FR_net_static_gateway** variable is used to set the gateway, please set it according to your actual situation

7) The following demonstrates several specific setting examples:
   a. For example, if you want the Linux system of the development board to automatically connect to the WIFI hotspot after the first startup, you can set it as follows:
      a) Set **FR_net_change_defaults** to **1**
      b) Set **FR_net_wifi_enabled** to **1**
      c) Set **FR_net_wifi_ssid** to the name of the WIFI hotspot you want to connect to
      d) Set **FR_net_wifi_key** as the password of the WIFI hotspot you want to connect to

   b. For example, after the Linux system that wants to develop the board is started for the first time, it will automatically connect to the WIFI hotspot, and set the IP address of the WIFI to a specific static IP address (so that when the Linux system starts, you can directly use the set static IP address to ssh remote Log in to the development board, you don't need to check the IP address of the development board through the router background), you can set it like this:
      a) Set **FR_net_change_defaults** to **1**
      b) Set **FR_net_wifi_enabled** to **1**

c) Set **FR_net_wifi_ssid** to the name of the WIFI hotspot you want to connect to
d) Set **FR_net_wifi_key** as the password of the WIFI hotspot you want to connect to
e) Set **FR_net_use_static** to **1**
f) Set **FR_net_static_ip** to the desired IP address
g) Set **FR_net_static_gateway** to the corresponding gateway address

c. For example, after the Linux system that wants to develop the board is started for the first time, the IP address of the Ethernet port is automatically set to the desired static IP address, which can be set as follows:
a) Set **FR_net_change_defaults** to **1**
b) Set **FR_net_ethernet_enable**d to **1**
c) Set **FR_net_use_static** to **1**
d) Set **FR_net_static_ip** to the desired IP address
e) Set **FR_net_static_gateway** to the corresponding gateway address

8) After modifying the orangepi_first_run.txt file, you can exit the /boot directory of the Linux system of the development board in the TF card, uninstall the TF card, and then insert the TF card into the development board to start

9) If a static IP address is not set, you still need to check the IP address through the router background. If a static IP address is set, you can ping the static IP address set on the computer. If you can ping, the system has been started normally, and The network has also been set correctly, and then you can use the set IP address to ssh to log in to the Linux system of the development board remotely.

After the Linux system of the development board is started for the first time, orangepi_first_run.txt will be deleted or renamed to orangepi_first_run.txt.old. At this time, even if the orangepi_first_run.txt configuration file is reset, and then restart the Linux system of the development board, orangepi_first_run. The configuration in txt will not take effect again, because this configuration will only work after the first boot after the Linux system is burned, please pay special attention to this.

3. 9. SSH remote login development board

By default, Linux systems enable ssh remote login and allow root users to log in to the system. Before ssh login, you need to ensure that the Ethernet or wifi network is connected, and then use the sudo ifconfig command or obtain the IP address of the development board by viewing the router

3. 9. 1. SSH remote login development board under Ubuntu

1) Obtain the IP address of the development board

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

test@test:~$ **ssh root@192.168.1.57**	(It needs to be replaced with the IP address of the development board)
root@192.168.1.57's password:	（Enter the password here, the default password is orangep）

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

If you are prompted to refuse the connection, as long as the image provided by Orange Pi is used, please do not doubt whether the password of orangepi is wrong, but find other reasons.

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

```
csy@ubuntu:~$ ssh root@192.168.1.57
root@192.168.1.57's password:
  ___  ____  ___   _ _     _   _____ ____
 / _ \|  _ \|_ _| | | |   | | |_   _/ ___|
| | | | |_) || |  | | |_  | |   | | \___ \
| |_| |  __/ | |  |__   _| | |___| |  ___) |
 \___/|_|   |___|    |_|   |_____|_| |____/

Welcome to Orange Pi 2.2.0 Bullseye with Linux 5.10.43

System load:   25%              Up time:       5 min     Local users:   2
Memory usage:  16% of 3.78G     IP:            192.168.1.57
CPU temp:      41°C             Usage of /:    22% of 15G

[ 0 security updates available, 26 updates total: apt upgrade ]
Last check: 2022-02-22 05:59

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

Last login: Tue Feb 22 06:03:37 2022 from 192.168.1.78
```

If ssh cannot log in to the linux system normally, please first check whether the IP address of the development board can be pinged. Can connect:

root@orangepi:~# **rm /etc/ssh/ssh_host_***

root@orangepi:~# **dpkg-reconfigure openssh-server**

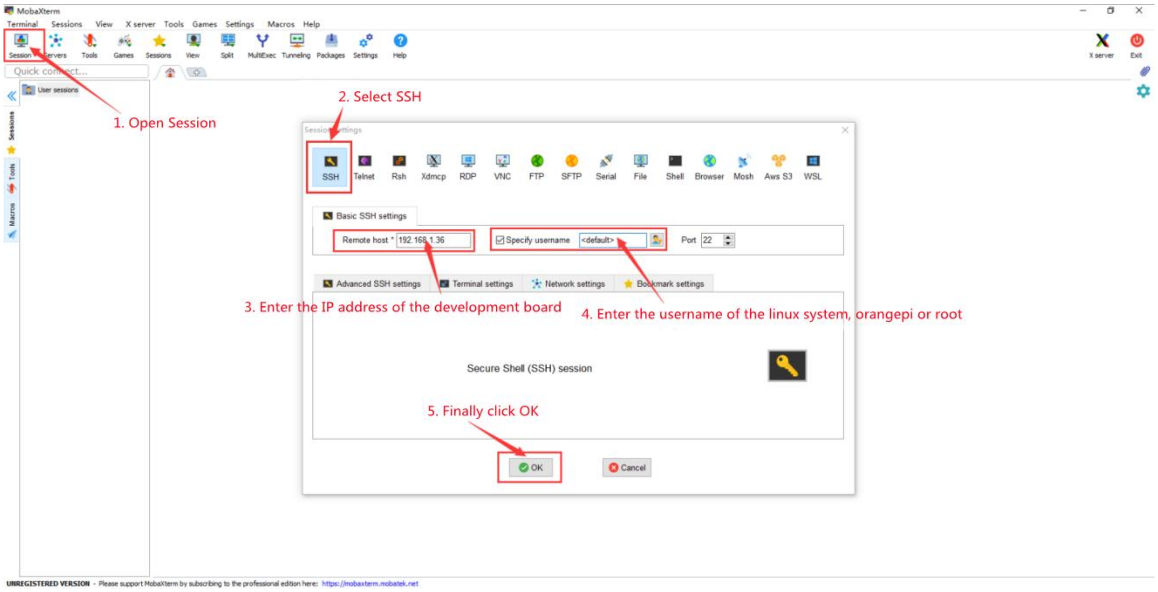
If it still doesn't work, please restart the system and try。

3. 9. 2. SSH remote login development board under Windows

1) First get the IP address of the development board

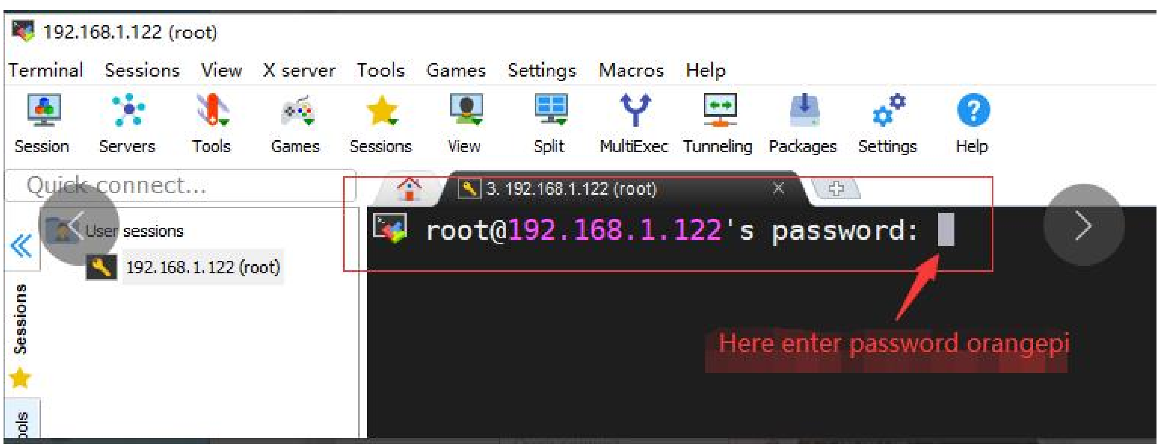
2) Under Windows, you can use MobaXterm to remotely log in to the development board, first create a new ssh session

a. Open **Session**
b. Then select **SSH** in **Session Setting**
c. Then enter the IP address of the development board in **Remote host**
d. Then enter the username **root** or **orangepi** of the linux system in **Specify username**
e. Finally click **OK**

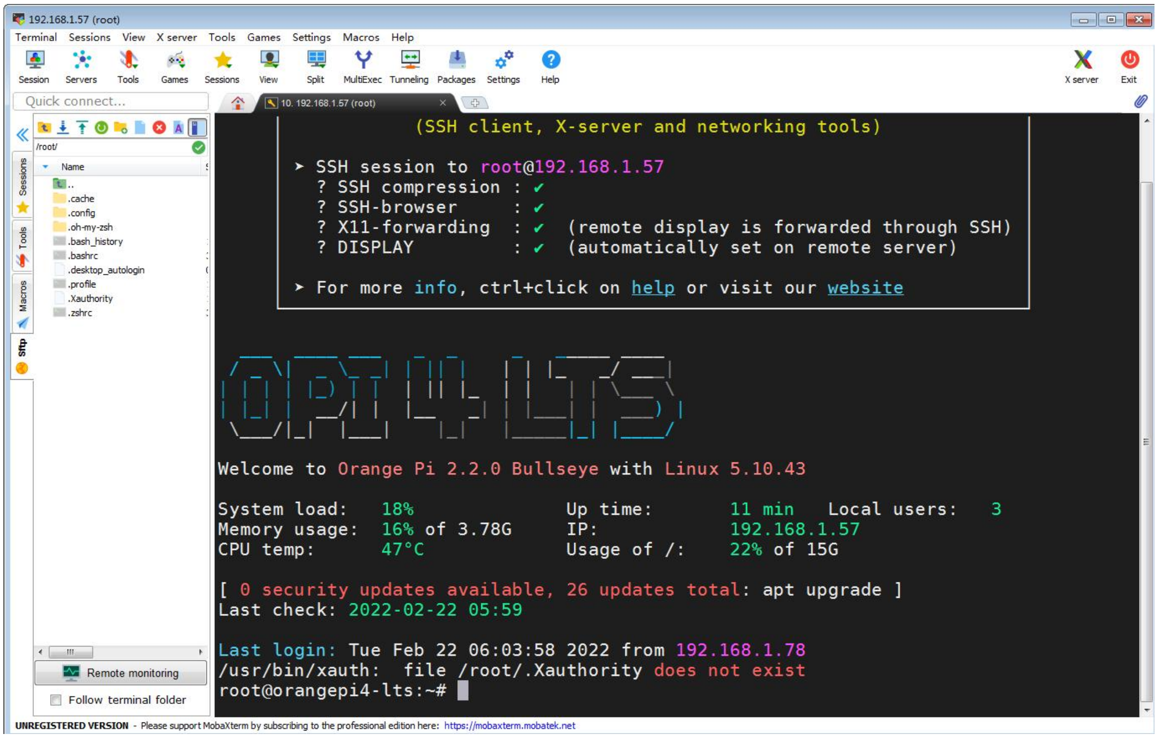


3) Then you will be prompted to enter a password. The default passwords for both root and orangepi users are orangepi

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



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



3. 10. HDMI related test items

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 display 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 laptop generally only has the output function, and does not have the function of HDMI in, which means that the HDMI output of other devices cannot be

displayed on the screen of the notebook.

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 does not display, please check whether the HDMI cable is plugged in tightly. After confirming that the connection is correct, you can try a different screen to see if there is any display.

3. 10. 2. HDMI to VGA display test

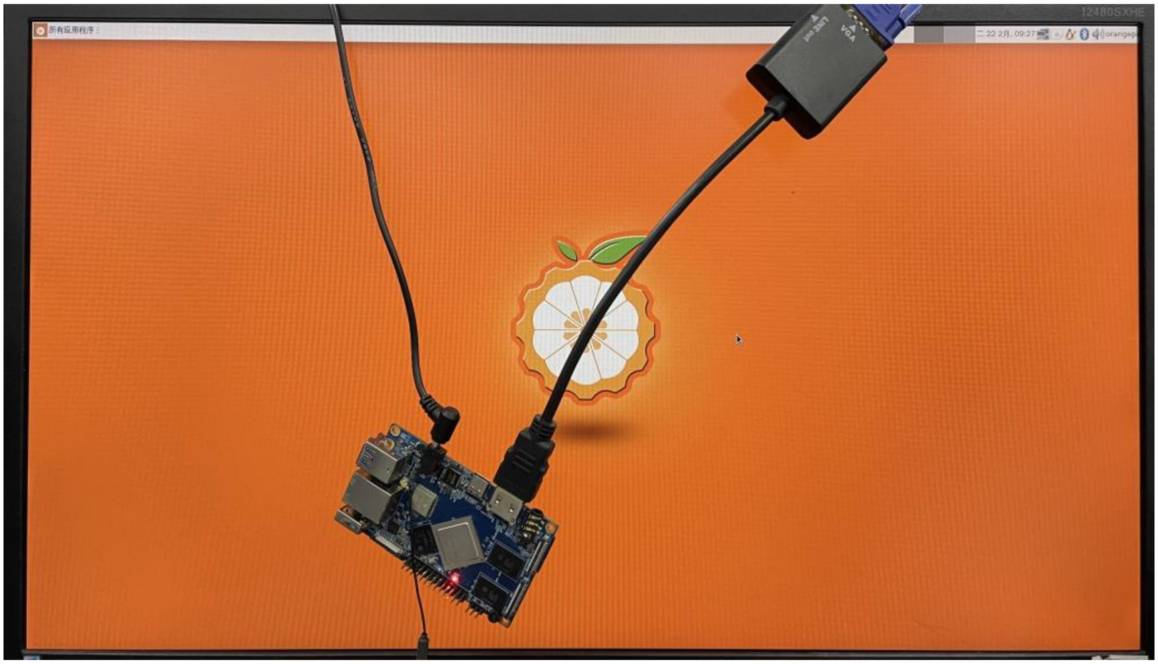
1) First you need to prepare the following accessories

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 do any settings, as long as the HDMI interface of the development board can display normally. So if there is a problem with the test, please check the HDMI to VGA converter, VGA cable and monitor for problems.

3.10.3. Type C to HDMI display test

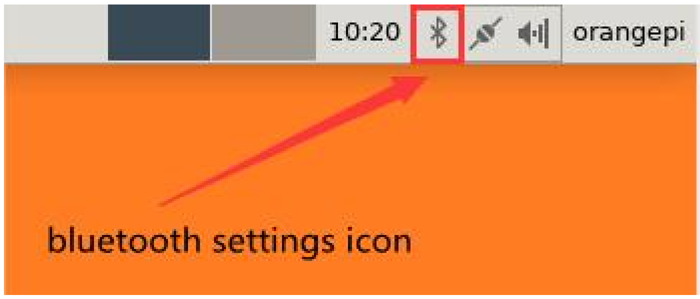
1) Use Type C to HDMI cable to connect Orange Pi development board and HDMI display

2) After starting the linux system, if the HDMI display has image output, it means that the Type C to HDMI display function is normal

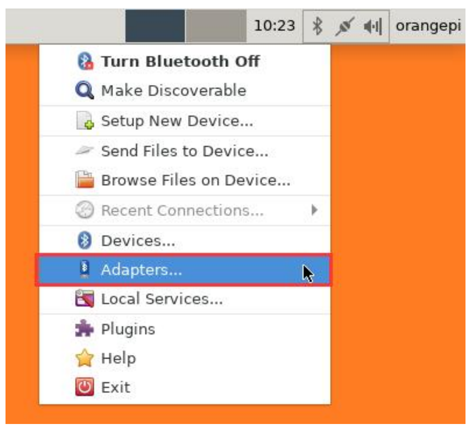
3. 11. How to use Bluetooth

3. 11. 1. Test method for desktop image

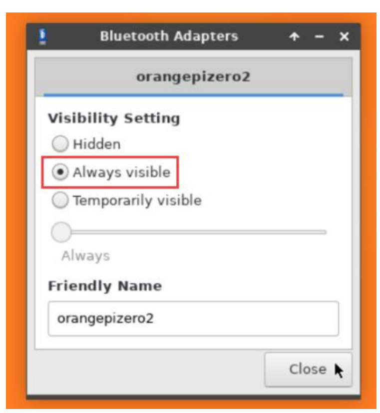
1) Click the Bluetooth icon in the upper right corner of the desktop



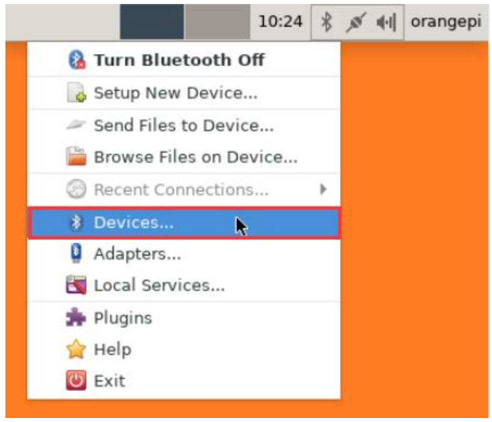
2) Then select adapter



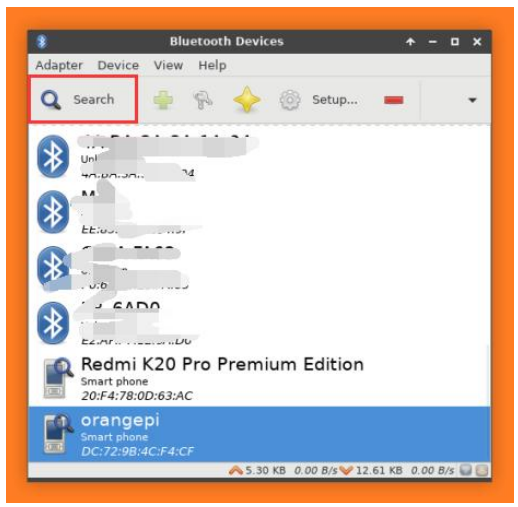
3) Set the **Visibility Setting** to **Always visible** in the Bluetooth adapter setting interface, and then click **close** to close



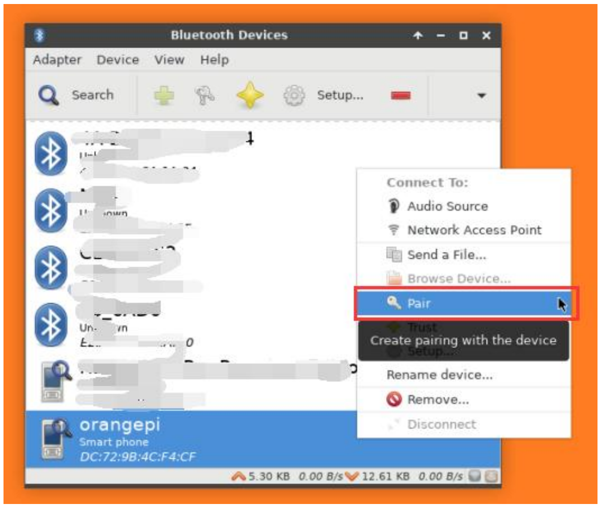
4) Then open the configuration interface of the Bluetooth device



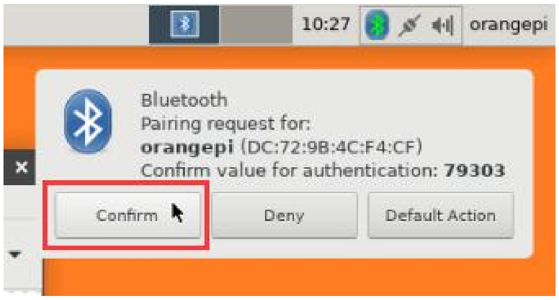
5) Click **Search** to start scanning for surrounding Bluetooth devices



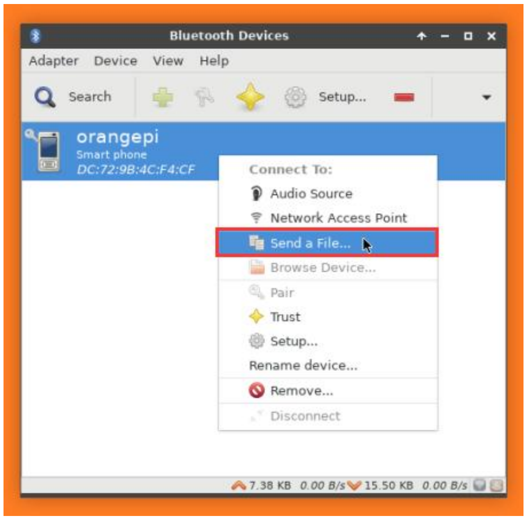
6) Then select the Bluetooth device you want to connect, and then click the right mouse button to pop up the operation interface of the Bluetooth device. Select **Pair** to start pairing. The demonstration here is pairing with an Android phone.



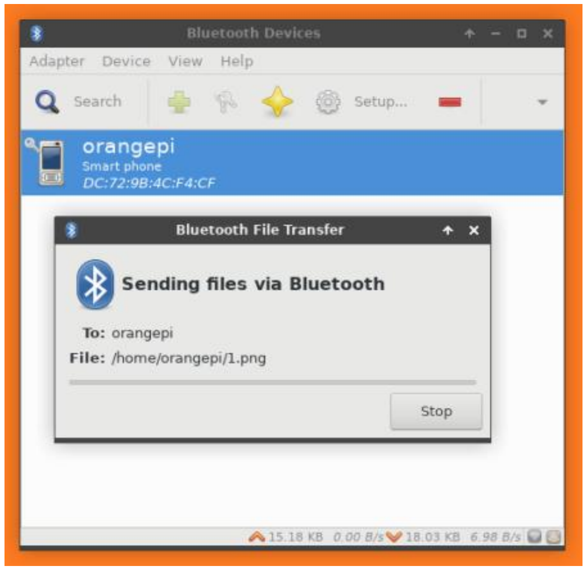
7) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop. Select **Confirm** to confirm. At this time, the mobile phone also needs to be confirmed.



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. 11. 2. How to use the server version image

1) First install bluez, the command is as follows

```
root@orangepi4-lts:~# apt-get update
root@orangepi4-lts:~# apt-get install -y bluez
```

2) Then you can use the **hciconfig** command to check whether there is a Bluetooth device node. If there is, it means that the Bluetooth initialization is normal.

```
root@orangepi4-lts:~# hciconfig -a
hci0:    Type: BR/EDR   Bus: UART
    BD Address: 43:45:C5:00:1F:AC   ACL MTU: 1021:8   SCO MTU: 64:1
    UP RUNNING PSCAN ISCAN
    RX bytes:897 acl:0 sco:0 events:65 errors:0
    TX bytes:4355 acl:0 sco:0 commands:65 errors:0
    Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
    Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
    Link policy: RSWITCH SNIFF
    Link mode: SLAVE ACCEPT
    Name: 'orangepi4-lts'
    Class: 0x1c0000
    Service Classes: Rendering, Capturing, Object Transfer
    Device Class: Miscellaneous,
    HCI Version:   (0x9)   Revision: 0x26
```

```
LMP Version:   (0x9)   Subversion: 0x6606
Manufacturer: Broadcom Corporation (15)
```

3) Scan for bluetooth devices using bluetoothctl

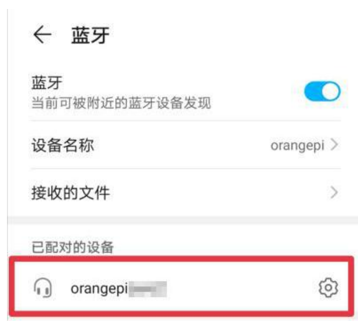
```
root@orangepi4-lts:~# bluetoothctl
[NEW] Controller 10:11:12:13:14:15 orangepi3 [default]
Agent registered
[bluetooth]# power on          #enable controller
Changing power on succeeded
[bluetooth]# discoverable on       #Make the controller discoverable
Changing discoverable on succeeded
[CHG] Controller 10:11:12:13:14:15 Discoverable: yes
[bluetooth]# pairable on      #Set the controller to be pairable
Changing pairable on succeeded
[bluetooth]# scan on           #Start scanning for nearby bluetooth devices
Discovery started
[CHG] Controller 10:11:12:13:14:15 Discovering: yes
[NEW] Device 76:60:79:29:B9:31 76-60-79-29-B9-31
[NEW] Device 9C:2E:A1:42:71:11 Xiaomi phone
[NEW] Device DC:72:9B:4C:F4:CF orangepi
[bluetooth]# scan off          # After scanning the Bluetooth device you want to connect, you can close the scan, and then write down the MAC address of the Bluetooth device. The Bluetooth device tested here is an Android phone, the Bluetooth name is orangepi, and the corresponding MAC address is DC:72:9B:4C :F4:CF
Discovery stopped
[CHG] Controller 10:11:12:13:14:15 Discovering: no
[CHG] Device DC:72:9B:4C:F4:CF RSSI is nil
```

4) After scanning the device you want to pair, you can pair it. The pairing needs to use the MAC address of the device

```
[bluetooth]# pair DC:72:9B:4C:F4:CF       #Use the scanned MAC address of the
                                            Bluetooth device for pairing
Attempting to pair with DC:72:9B:4C:F4:CF
[CHG] Device DC:72:9B:4C:F4:CF Connected: yes
```

```
Request confirmation
[leeb1m[agent] Confirm passkey 764475 (yes/no): yes    #Enter yes here, you also need
                                                        to confirm on the phone
[CHG] Device DC:72:9B:4C:F4:CF Modalias: bluetooth:v010Fp107Ed1436
[CHG] Device DC:72:9B:4C:F4:CF UUIDs: 0000046a-0000-1000-8000-00805f9b34fb
[CHG] Device DC:72:9B:4C:F4:CF ServicesResolved: yes
[CHG] Device DC:72:9B:4C:F4:CF Paired: yes
Pairing successful      #Prompt for successful pairing
[CHG] Device DC:72:9B:4C:F4:CF ServicesResolved: no
[CHG] Device DC:72:9B:4C:F4:CF Connected: no
```

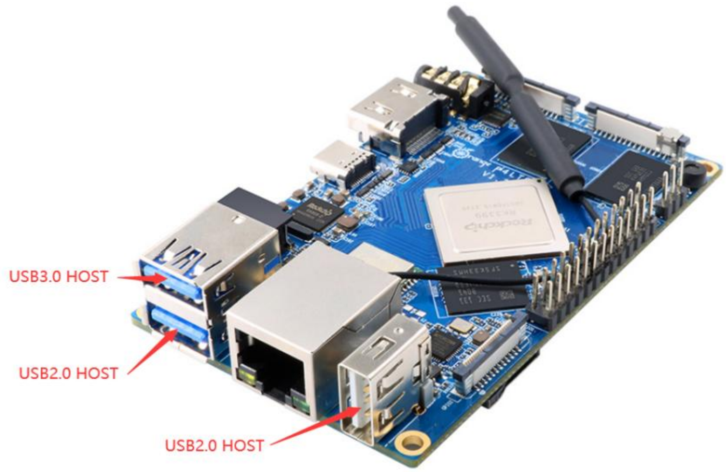
5) After the pairing is successful, the display of the Bluetooth interface of the mobile phone is as follows



3. 12. USB interface test

3. 12. 1. USB2.0 and USB3.0 interface description

1) Orange Pi 4 LTS has a total of 3 USB ports, including one USB3.0 and two USB2.0 ports, the corresponding positions are as follows：



Although the USB interface in the lower left corner is blue, it only has the function of USB2.0, please pay special attention to this.

3. 12. 2. Connect the mouse or keyboard to test

1) Insert the mouse or keyboard of the USB interface 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, the USB interface is in normal use (the mouse can only be used in the desktop version of the system)

3. 12. 3. 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 successfully recognized

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

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

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

```
test.txt
```

To mount a U disk in **exfat** format in Linux system, you can use the following command

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

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

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

3. 12. 4. USB camera test

1) First 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/video10**

```
root@orangepi:~# apt-get update
root@orangepi:~# apt-get install v4l-utils
root@orangepi:~# v4l2-ctl --list-devices
USB2.0 Camera RGB (usb-xhci-hcd.11.auto-1):
    /dev/video10
```

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

In addition, the serial numbers of videos are not necessarily all video10, please refer to what you actually see.

3) Use fswebcam to test the USB camera

   a. Install fswebcam

```
root@orangepi:~# apt-get update
root@orangepi:~# apt-get -y install fswebcam
```

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

      a) -d Option to specify the device node for the USB camera

      b) --no-banner watermark removal from photos

      c) -r Option to specify the resolution of the photo

d) -S option is set to skip previous frames

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

Note that the serial numbers of videos are not necessarily all video10, please refer to what you actually see.

```
root@orangepi:~# fswebcam -d /dev/video10 --no-banner -r 1280x720 -S \
5 ./image.jpg
```

c. If there is no HDMI display or LCD screen connected, after taking the picture, you can use the scp command to transfer the taken picture to the Ubuntu PC for mirror viewing

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

d. If an HDMI display or LCD screen is connected, you can view the captured pictures directly through the HDMI display or LCD screen

4) Use motion to test the USB camera

a. Install the camera test software motion

```
root@orangepi:~# apt-get update
root@orangepi:~# apt-get -y install motion
```

b. Modify the configuration of **/etc/default/motion** and change start_motion_daemon=no to start_motion_daemon=yes

Note that Ubuntu22.04 does not need to set this step.

```
root@orangepi:~# sed -i "s/start_motion_daemon=no/start_motion_daemon=yes/" \
/etc/default/motion
```

c. Modify the configuration of **/etc/motion/motion.conf**

```
root@orangepi:~# sed -i "s/stream_localhost on/stream_localhost off/" \
/etc/motion/motion.conf
```

d. In addition, make sure that the **videodevice** of **/etc/motion/motion.conf** is set to the device node corresponding to the USB camera

Note that the serial numbers of videos are not necessarily all video10, please refer to what you actually see.

```
orangepi@orangepi:~$ sudo vim /etc/motion/motion.conf
# Video device (e.g. /dev/video0) to be used for capturing.
videodevice /dev/video10
```

e. Then restart the motion service

```
root@orangepi4-lts:~# motion -b
```

f. Before using motion, please make sure that the Orange Pi development board can connect to the network normally, and then obtain the IP address of the development board through the ifconfig command

g. Then enter **[development board IP address: 8081]** in the Ubuntu PC or Windows PC or Firefox browser on the same local area network as the development board to see the video output by the camera

5) Use mjpg-streamer to test the USB camera

a. Download mjpg-streamer

a) Github download address:

```
orangepi@orangepi:~$ git clone https://github.com/jacksonliam/mjpg-streamer
```

b) The image download address of Gitee is:

```
orangepi@orangepi:~$ git clone https://gitee.com/leeboby/mjpg-streamer
```

b. Install dependent packages

a) Ubuntu system

```
orangepi@orangepi:~$ sudo apt-get install -y cmake libjpeg8-dev
```

b) Debian system

```
orangepi@orangepi:~$ sudo apt-get install -y cmake libjpeg62-turbo-dev
```

c. Compile and install mjpg-streamer

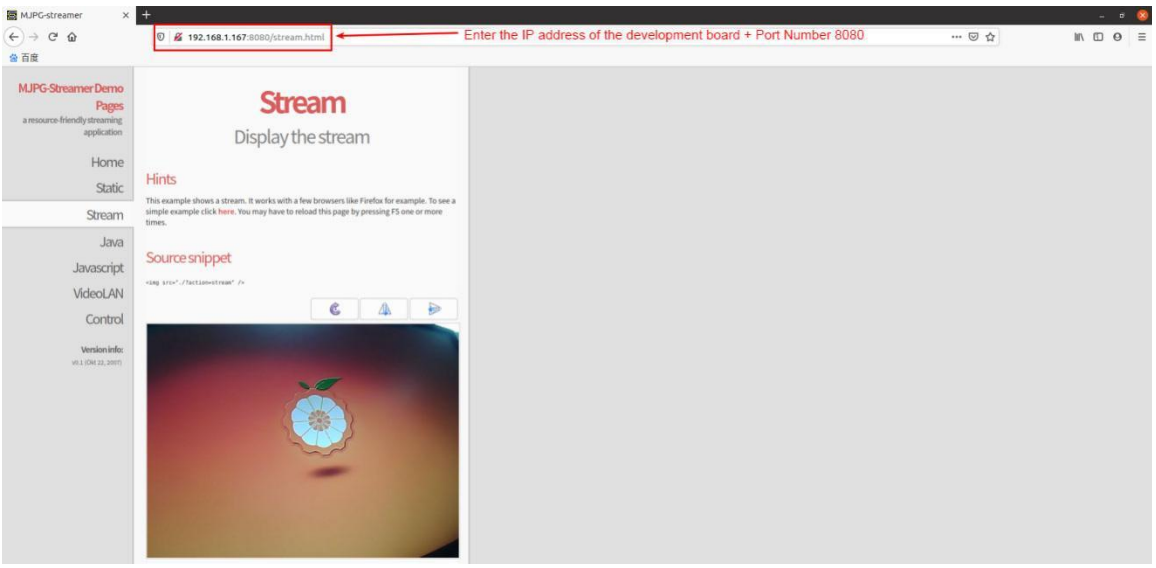
```
orangepi@orangepi:~$ cd mjpg-streamer/mjpg-streamer-experimental
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ make -j6
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ sudo make install
```

d. Then enter the following command to start mjpg_streamer

Note that the serial numbers of videos are not necessarily all video10, please refer to what you actually see.

```
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ export \
LD_LIBRARY_PATH=.
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ sudo \
./mjpg_streamer -i "./input_uvc.so -d /dev/video10 -u -f 30" \
-o "./output_http.so -w ./www"
```

e. Then enter **[IP address of the development board: 8080]** in the browser of the Ubuntu PC or Windows PC or mobile phone on the same LAN as the development board to see the video output by the camera.



f. It is recommended to use mjpg-streamer to test the USB camera, which is much smoother than motion, and does not feel any lag when using mjpg-streamer

3. 13. Audio Test

3. 13. 1. Audio test method of headphone jack

1) First insert headphones into the audio interface

2) Through the **aplay -l** command, you can view the sound card devices supported by the linux system, among which **rockchipes8316c** is the sound card device required for headphone playback

```
root@orangepi:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
```

```
xcb_connection_has_error() returned true
card 0: rockchipes8316c [rockchip-es8316c], device 0: ff890000.i2s-ES8316 HiFi ES8316 HiFi-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: rkhdmidpsound [rk-hdmi-dp-sound], device 0: HDMI-DP multicodec-0 []
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3) Then use the aplay command to play the audio file that comes with the Linux system, and the headset can hear the sound

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

3. 13. 2. HDMI audio playback test method

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

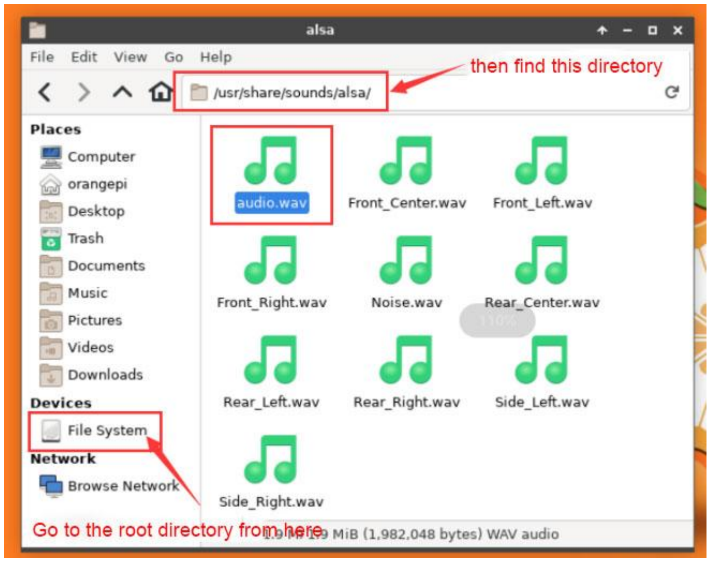
2) HDMI audio playback does not require other settings, just use the aplay command to play

```
root@orangepi4-lts:~# aplay -D hw:1,0 /usr/share/sounds/alsa/audio.wav
Playing WAVE '/usr/share/sounds/alsa/audio.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

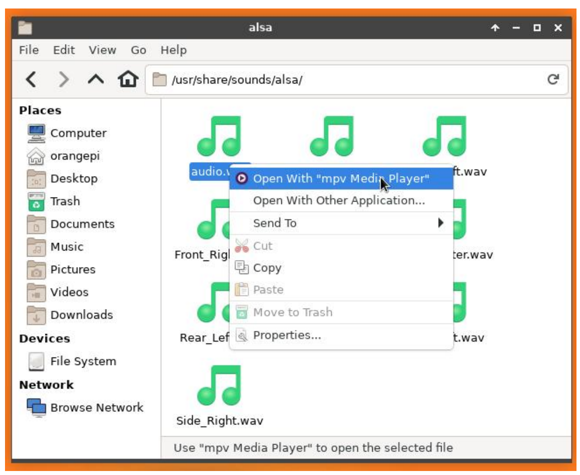
3. 13. 3. Test the audio method in the desktop system

1) First open the file manager

2) Then find the following file (if the audio file does not exist in the system, you can upload an audio file to the system yourself)

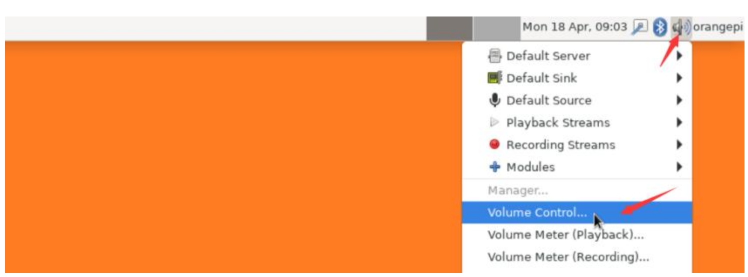


3) Then select the audio.wav file, right-click and choose to open with mpv or celluloid to start playing. If there is no pre-installed player in the image, you can use the apt-get install command to install one, and you can install whichever you want.

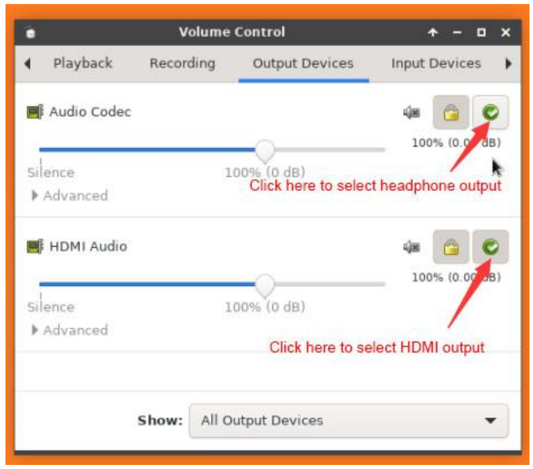


4) How to switch between HDMI playback and headphone playback

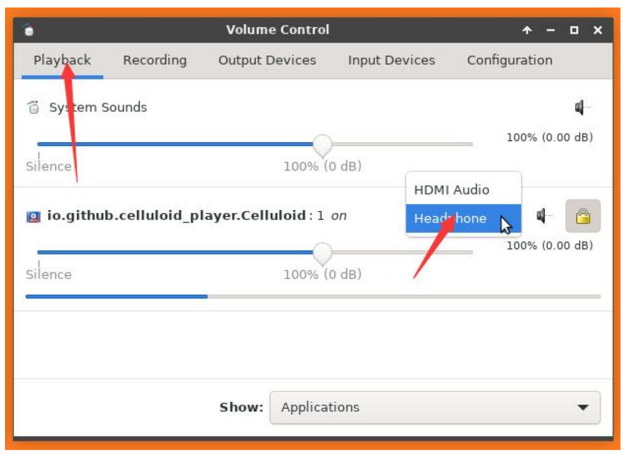
a. First open the volume control interface



b. Select **Output Devices**, then you can select the output audio device



c. In addition, when playing audio, the audio setting options of the **playback** software will be displayed in Playback, as shown in the figure below, you can also set which audio device to play to here.



3. 13. 4. Onboard MIC recording test method

1) First make sure the system has opened the recording channel below, and adjust the recording volume to the maximum

```
root@orangepi:~# amixer cset name='Differential Mux' lin2-rin2
root@orangepi:~# amixer cset name="ADC Capture Volume" 192
```

2) The recording command is as follows

```
root@orangepi:~# arecord -D hw:0,0 -d 5 -f cd -t wav test.wav
Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

3) After the recording is completed, a recording file named test.wav will be generated in the current path. Use the aplay command to play test.wav to check whether there is sound output. If there is sound output, the recording is normal.

```
root@orangepi:~# ls -lh
total 862K
-rw-r--r-- 1 root root 862K Feb   5 04:24 test.wav
root@orangepi4-lts:~# aplay -D hw:0,0 test.wav
```

3. 13. 5. Headphone recording test method

1) First plug in the headphones with the recording function

2) Then use amixer to open the recording channel below, and adjust the recording volume to the maximum

```
root@orangepi:~# amixer cset name='Differential Mux' lin1-rin1
root@orangepi:~# amixer cset name="ADC Capture Volume" 192
```

3) The recording command is as follows

```
root@orangepi4-lts:~# arecord -D hw:0,0 -d 5 -f cd -t wav test.wav
Recording WAVE 'test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
```

4) After the recording is completed, a recording file named test.wav will be generated in the current path. Use the aplay command to play test.wav to check whether there is sound output. If there is sound output, the recording is normal.

```
root@orangepi4-lts:~# ls -lh
total 862K
-rw-r--r-- 1 root root 862K Feb   5 04:24 test.wav
root@orangepi4-lts:~# aplay -D hw:0,0 test.wav
```

3. 14. Temperature sensor

The displayed temperature value needs to be divided by 1000, the unit is Celsius.

1) RK3399/RK3399-T has a total of 2 temperature sensors, the command to check the temperature is as follows

a. sensor0：CPU

```
root@orangepi4-lts:~# cat /sys/class/thermal/thermal_zone0/type
soc-thermal
root@orangepi4-lts:~# cat /sys/class/thermal/thermal_zone0/temp
```

```
48125
```

b. sensor1：GPU

```
root@orangepi4-lts:~# cat /sys/class/thermal/thermal_zone1/type
gpu-thermal
root@orangepi4-lts:~# cat /sys/class/thermal/thermal_zone1/temp
49375
```

3. 15. How to use Mini PCIE

3. 15. 1. Connect SATA hard disk through mini PCIE interface

1) For the method of hardware wiring, please refer to the **instructions for connecting mini PCIE to SATA hard disk**

2) After the wiring is completed, connect the DC power supply to the development board. After the system starts, execute the lspci command in the terminal to see the pci devices recognized by the Linux system.

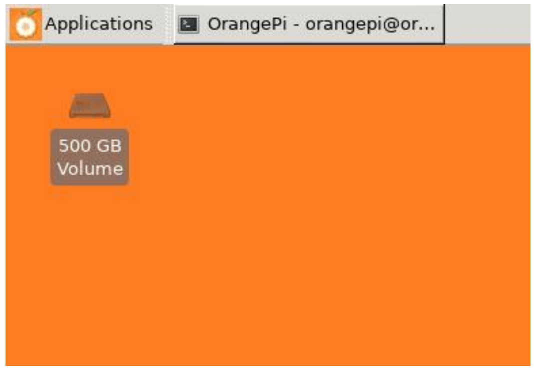
```
root@orangepi:~# lspci
00:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd RK3399 PCI Express Root Port
01:00.0 SATA controller: ASMedia Technology Inc. ASM1062 Serial ATA Controller (rev 02)
```

And the information about the hard disk can also be seen in the output information of the dmesg command

```
[   14.535120] scsi 0:0:0:0: Direct-Access     ATA      WDC WD5000LPCX-2 1A01 PQ: 0 ANSI: 5
[   14.536785] sd 0:0:0:0: [sda] 976773168 512-byte logical blocks: (500 GB/466 GiB)
[   14.536808] sd 0:0:0:0: [sda] 4096-byte physical blocks
[   14.537238] sd 0:0:0:0: [sda] Write Protect is off
[   14.537262] sd 0:0:0:0: [sda] Mode Sense: 00 3a 00 00
[   14.537411] sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
[   14.590267]  sda: sda1
[   14.592221] sd 0:0:0:0: [sda] Attached SCSI disk
```

3) You can also see that the hard disk has been mounted on the desktop of the Linux

system

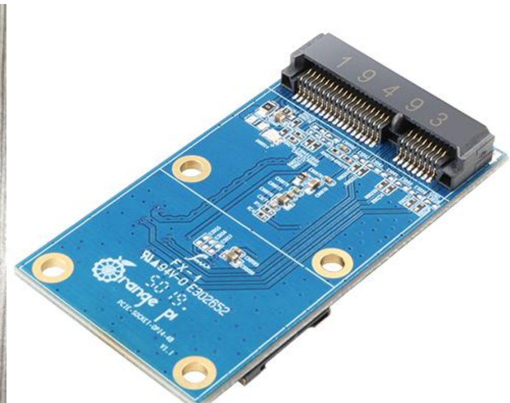


3.15.2. Connect the Gigabit Ethernet card through the mini PCIE interface

1) First prepare the required accessories

a. Realtek 8111E mini PCIE to Gigabit Ethernet module

b. 24pin reverse cable and mini PCIE adapter board



c. Fast or Gigabit Ethernet cable

2) Connect the 24pin reverse cable to the mini PCIE adapter board as shown in the figure below

3) Then connect the mini PCIE to Gigabit LAN module to the mini PCIE adapter board

Put a piece of insulating white paper between the gigabit network card module and the mini PCIE adapter board to avoid short circuit caused by direct contact

between the gigabit network card module and the mini PCIE adapter board. After connection, use a rope to fix it.

4) Then connect the mini PCIE adapter board to the Orange Pi 4 LTS development board through the 24pin reverse cable

5) Finally, connect the network cable to the network port of the mini PCIE to Gigabit network card module

6) Power on the Orange Pi 4 LTS. After the system starts, enter the lspci command in the command line terminal. If there is the following output information, it means that the

mini PCIE to Gigabit network card has been recognized.

```
root@orangepi:~# lspci
00:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd RK3399 PCI Express Root Port
01:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller (rev 06)
```

Use the ifconfig command to view the network device name and IP address corresponding to the mini PCIE to Gigabit network card

```
root@orangepi:~# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.63  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::bafe:2ff8:1a59:c3eb  prefixlen 64  scopeid 0x20<link>
        ether 00:e0:4c:68:0c:3c  txqueuelen 1000  (Ethernet)
        RX packets 98  bytes 7680 (7.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 31  bytes 3066 (2.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

7) The command to test network connectivity is as follows

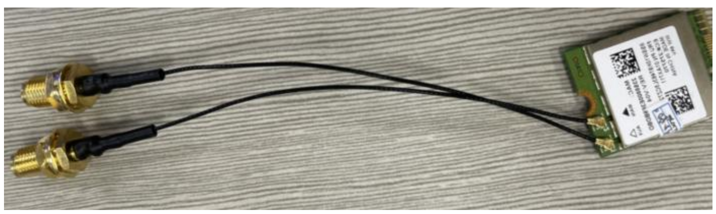
```
root@orangepi:~# ping www.baidu.com -I enp1s0
PING www.a.shifen.com (14.215.177.38) from 192.168.1.63 enp1s0: 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=56 time=8.49 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=56 time=8.81 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=56 time=7.99 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=56 time=8.87 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 8ms
rtt min/avg/max/mdev = 7.988/8.540/8.874/0.362 ms
```

3. 15. 3. Connect wireless network card through mini PCIE interface

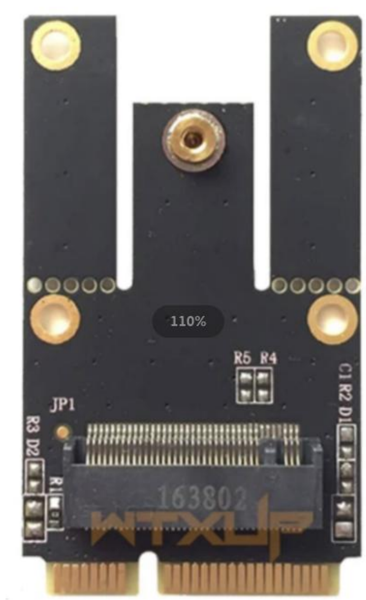
1) First prepare the required accessories

   a. Realtek RTL8822BE wireless network card, the size is 2230, the interface specification is NGFF M2

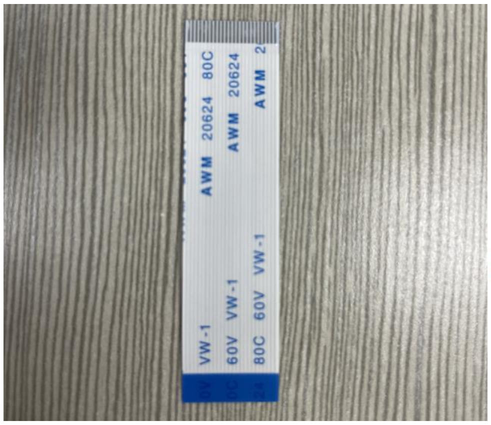
b. Supporting antenna, IPX4 interface, there are two kinds of built-in and external, the picture below shows the external antenna

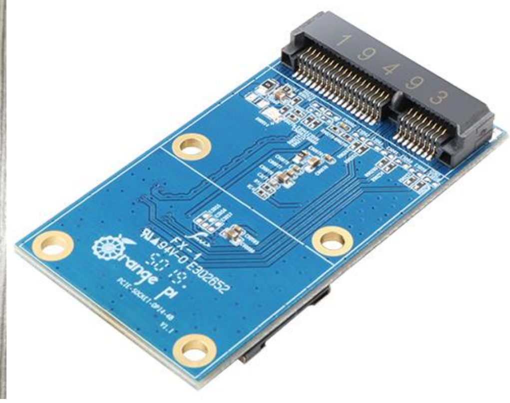


c. NGFF to mini PCIE adapter card



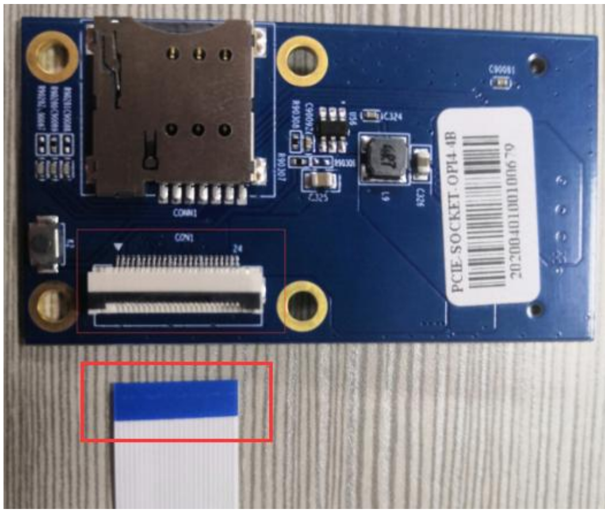
d. 24pin reverse cable and mini PCIE adapter board





2) Connect the RTL8822BE wireless network card to the NGFF to mini PCIE adapter card as shown below

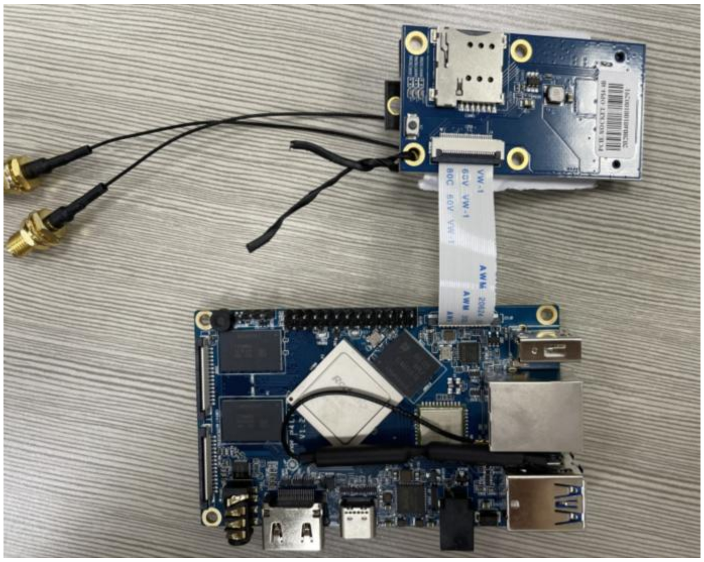
3) Then connect the 24pin reverse row wiring to the mini PCIE adapter board as shown below



4) Then connect the NGFF to mini PCIE adapter card to the mini PCIE adapter board

Put a piece of insulating white paper between the wireless network card module and the mini PCIE adapter board to avoid short circuit caused by direct contact between the wireless network card module and the mini PCIE adapter board, and can be fixed with a rope after connection.

8) Finally, connect the mini PCIE adapter board to the Orange Pi 4 LTS development board through the 24pin reverse cable



9) Power on the Orange Pi 4 LTS development board. After the system starts, enter the lspci command in the command line terminal. If there is the following output information, it means that the RTL8822BE wireless network card is recognized

```
root@orangepi4-lts:~# lspci
00:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd RK3399 PCI Express Root Port
01:00.0 Unassigned class [ff00]: Realtek Semiconductor Co., Ltd. RTL8822BE 802.11a/b/g/n/ac WiFi adapter
```

Use the ifconfig command to view the network device name corresponding to the RTL8822BE wireless network card

```
root@orangepi4-lts:~# ifconfig wlp1s0
wlp1s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 5a:00:e3:f9:bd:bd  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 18  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

10) Then you can test the WIFI according to the instructions in the **WIFI connection**

test section

3. 15. 4. Connect SSD through mini PCIE interface

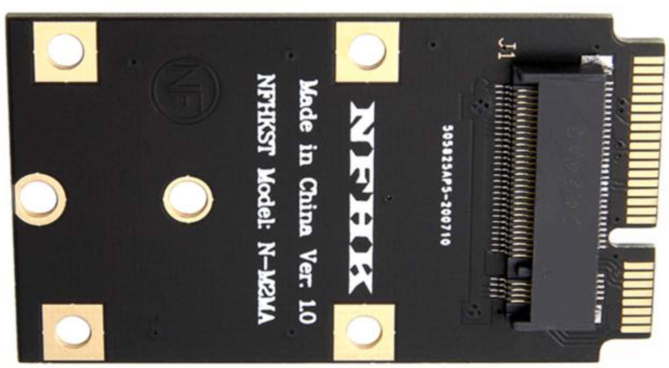
1) First prepare the required accessories

a. Kingston SSD, the size is 2280, the interface specification is NGFF M2, and the protocol is NvMe

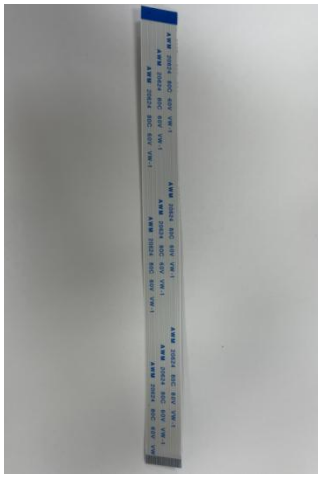
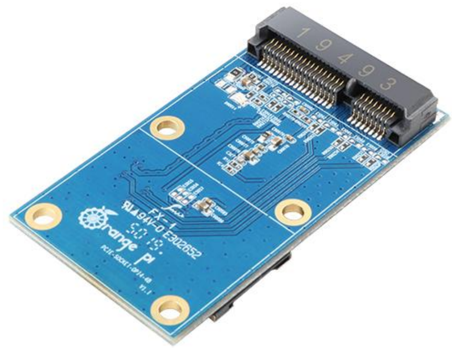
When purchasing, be sure to choose an M.2 NGFF SSD that supports the NVMe protocol. The corresponding M.2 interface is of the M key type. SSDs that meet this requirement should be able to support it. If it is an M.2 NGFF SSD of the SATA protocol, it is not supported. Yes, you can check this link for information on the M.2 interface:

https://www.delock.de/infothek/M.2/M.2_e.html

b. mini PCIE to NVMe M.2 NGFF riser card

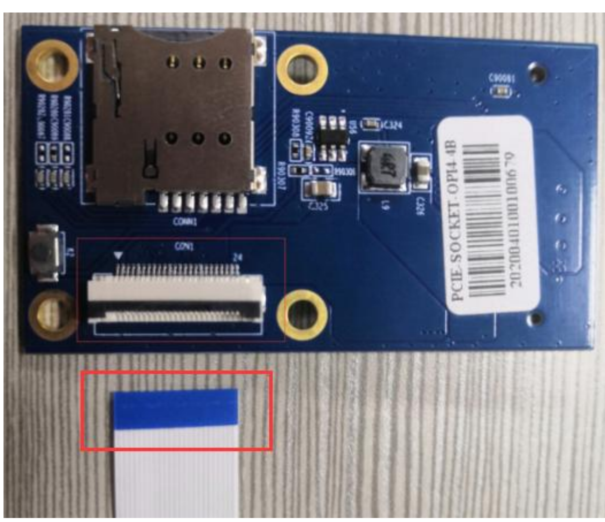


e. 24pin reverse cable and mini PCIE adapter board

2) Then connect the SSD to the mini PCIE to NVMe M.2 NGFF riser card as shown in the figure below, put a piece of insulating white paper in the middle to avoid short circuit caused by direct contact between the SSD and the riser card, and fix it with a rubber band

3) Then connect the 24pin reverse cable to the mini PCIE adapter board as shown below

4) Then connect the mini PCIE to NVMe M.2 NGFF adapter card to the mini PCIE adapter board

5) Then connect the mini PCIE adapter board to the Orange Pi 4 LTS development board through a 24pin reverse cable

6) Power on the Orange Pi 4 LTS development board. After the system starts, enter the **lspci** command in the command line terminal. If the following output information is displayed, it means that the **Kingston SSD** has been recognized

```
root@orangepi4-lts:~# lspci
```

```
00:00.0 PCI bridge: Fuzhou Rockchip Electronics Co., Ltd RK3399 PCI Express Root Port
01:00.0 Non-Volatile memory controller: Kingston Technology Company, Inc. Device 2262 (rev 03)
```

7) And the nvme device can also be seen in the output information of the dmesg command

```
root@orangepi4-lts:~# dmesg | grep nvme
[    2.922491] nvme nvme0: pci function 0000:01:00.0
[    2.922563] nvme 0000:01:00.0: enabling device (0000 -> 0002)
[    2.937629] nvme nvme0: missing or invalid SUBNQN field.
[    2.944972] nvme nvme0: 6/0/0 default/read/poll queues
```

8) In addition, the device node of **Kingston SSD** will be generated under /dev/

```
root@orangepi4-lts:~# ls /dev/nvme0*
/dev/nvme0    /dev/nvme0n1
```

9) Enter the following command to view the capacity of the SSD

```
root@orangepi4-lts:~# fdisk -l /dev/nvme0n1
Disk /dev/nvme0n1: 232.89 GiB, 250059350016 bytes, 488397168 sectors
Disk model: KINGSTON SKC2500M8250G
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

10) The method of testing SSD read and write rate is as follows:

a. First format the SSD as ext4 format

```
root@orangepi4-lts:~# mkfs.ext4 /dev/nvme0n1
mke2fs 1.46.2 (28-Feb-2021)
/dev/nvme0n1 contains a ext4 file system
        created on Mon Mar 21 07:34:41 2022
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 61049646 4k blocks and 15269888 inodes
Filesystem UUID: ef089041-afa0-4ec6-acba-d32282952f80
Superblock backups stored on blocks:
```

```
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
```

b. Then mount the SSD to the /mnt directory

```
root@orangepi4-lts:~# mount /dev/nvme0n1 /mnt
```

c. Test the write rate of 4k data blocks

```
root@orangepi4-lts:~# dd if=/dev/zero of=/mnt/test bs=4k count=1024k
1048576+0 records in
1048576+0 records out
4294967296 bytes (4.3 GB, 4.0 GiB) copied, 19.9681 s, 215 MB/s
```

d. Test the read rate of 4k data blocks

```
root@orangepi4-lts:~# dd if=/mnt/test of=/dev/null bs=4k
1048576+0 records in
1048576+0 records out
4294967296 bytes (4.3 GB, 4.0 GiB) copied, 20.4867 s, 210 MB/s
```

3. 16. GPU test description

3. 16. 1. Linux4.4 and Linux5.10 Debian system GPU test instructions

Note that the Linux4.4 Debian10 and Linux5.10 Debian11 systems use the kernel driver and closed-source user space Mali library provided by RK. Currently, only Debian10 and Debian11 desktop systems support this function, and Ubuntu20.04 does not support it temporarily.

1) First open the command line terminal on the desktop of the Debian system, then run **glmark2-es2**, and choose one of the following commands:

Note that the command below is to be run on the desktop.

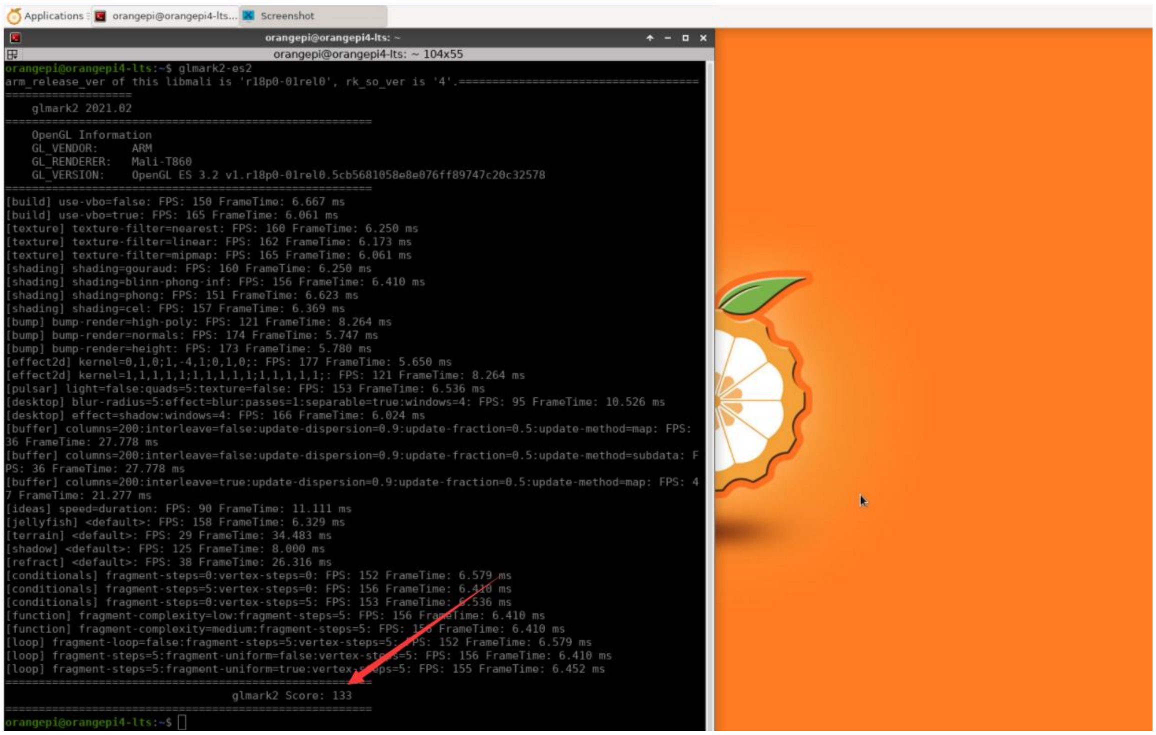
```
orangepi@orangepi:~$ glmark2-es2
orangepi@orangepi:~$ glmark2-es2 --off-screen
```

2) **glmark2-es2** is a benchmark tool for OpenGL (ES) 2.0. Using glmark2 can test the performance of GPU OpenGL ES 2.0

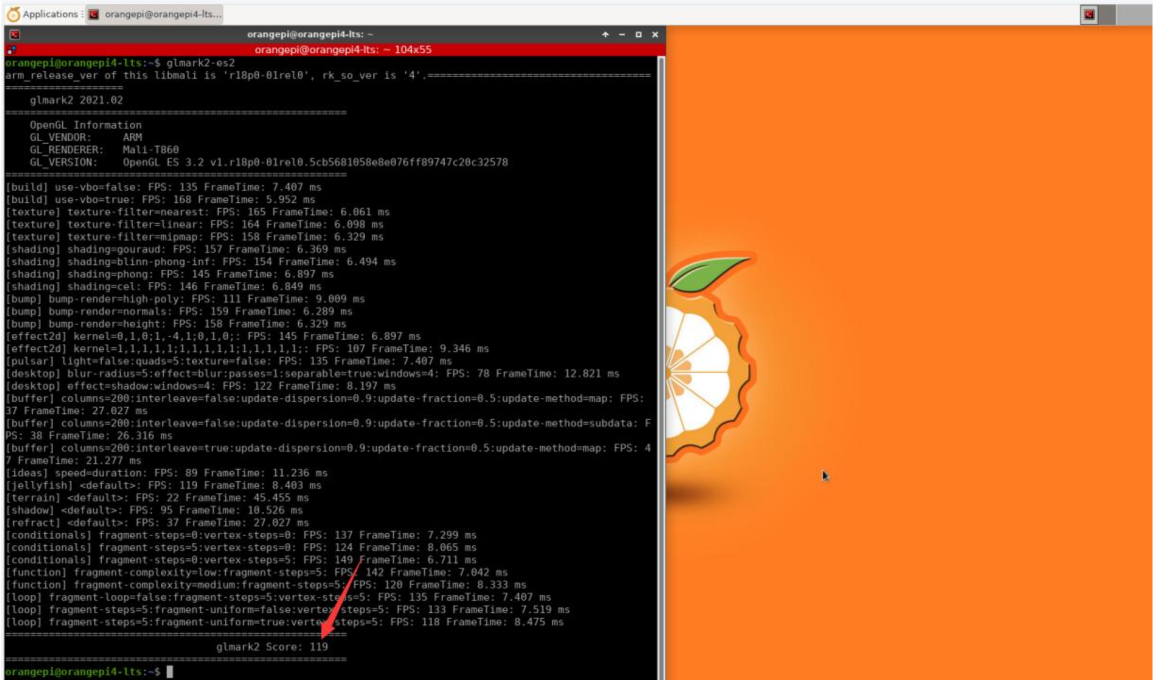
Note that the following results are tested in Debian11, please test Debian10 yourself, no screenshots are given here。

a. The test scores for the a.**glmark2-es2** command are as follows

a) Test results with fan cooling

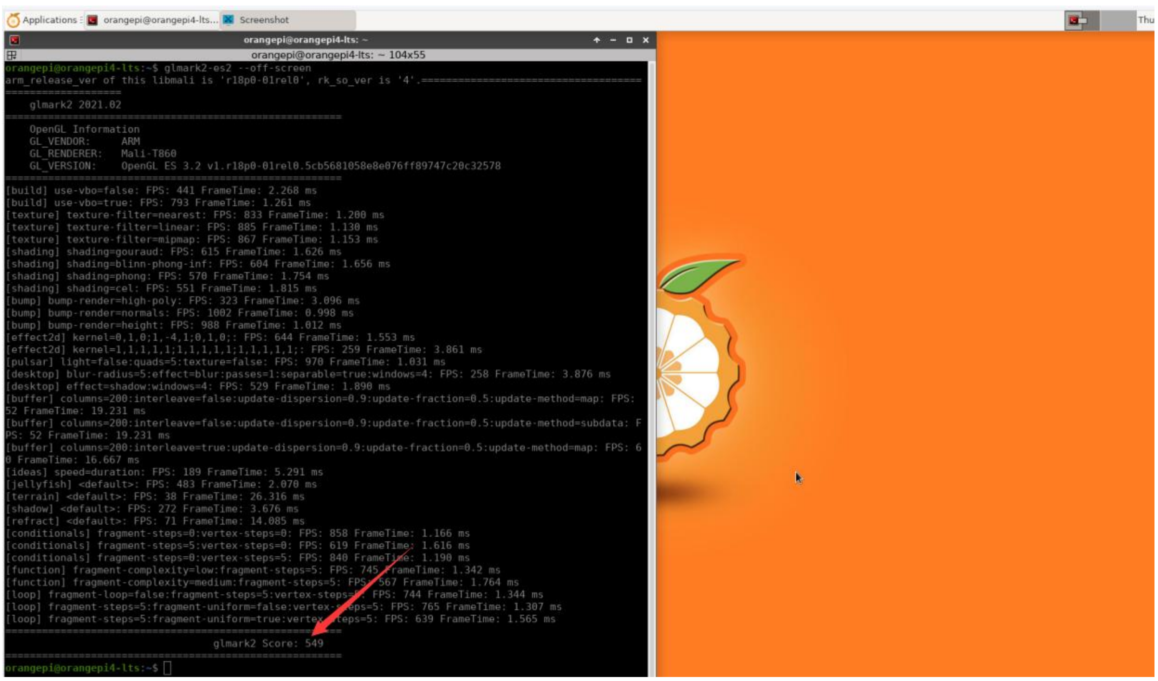


b) Test results without heat dissipation

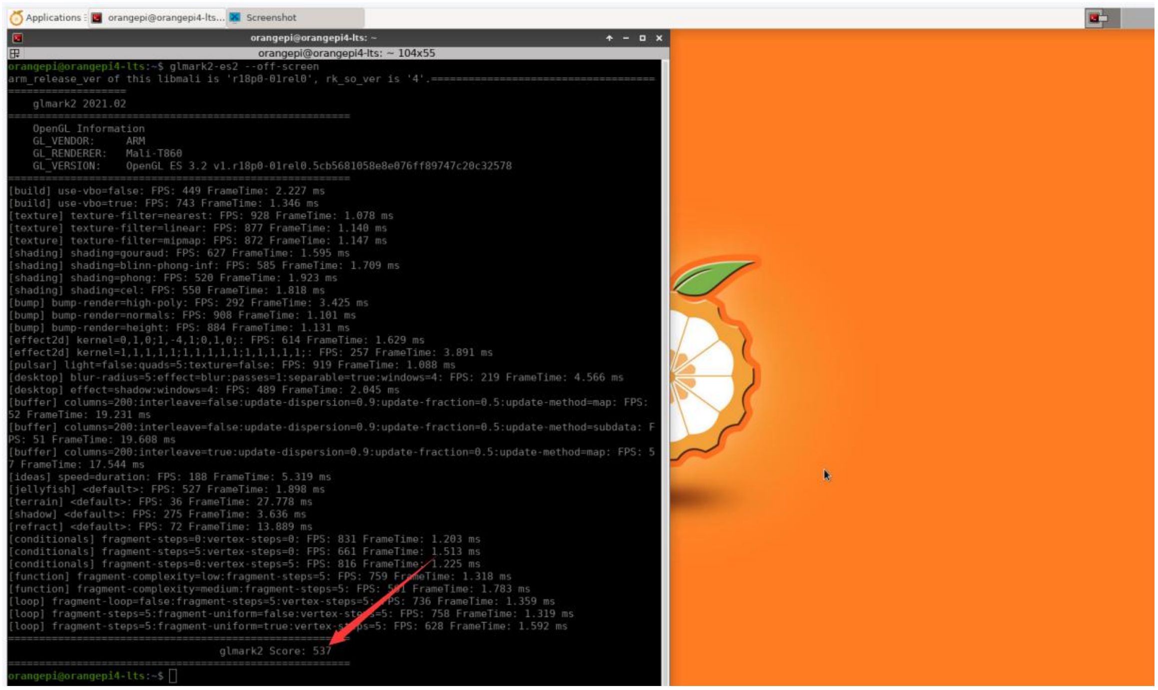


b. The test scores for the **glmark2-es2 --off-screen** command are as follows

a) Test results with fan cooling

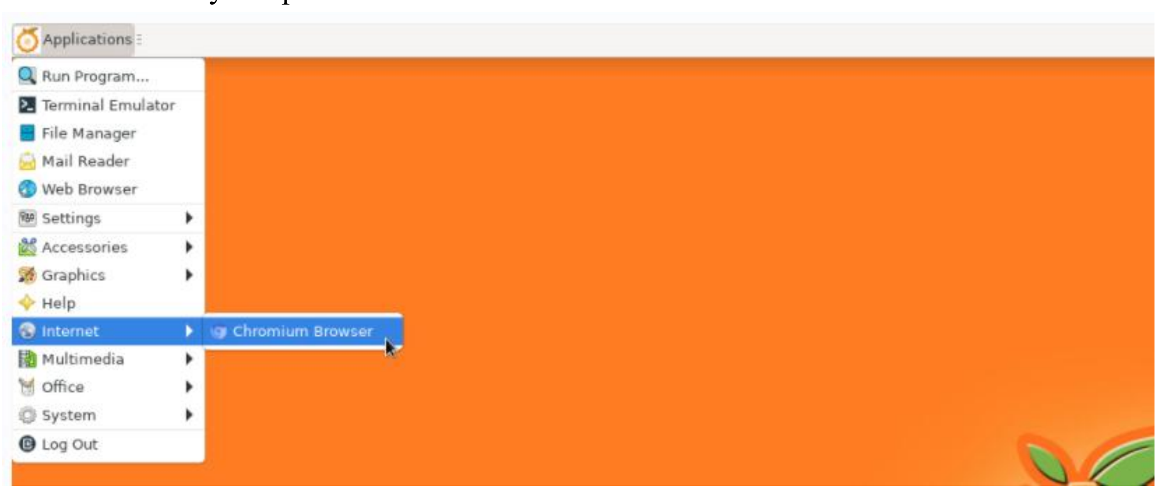


b) Test results without heat dissipation

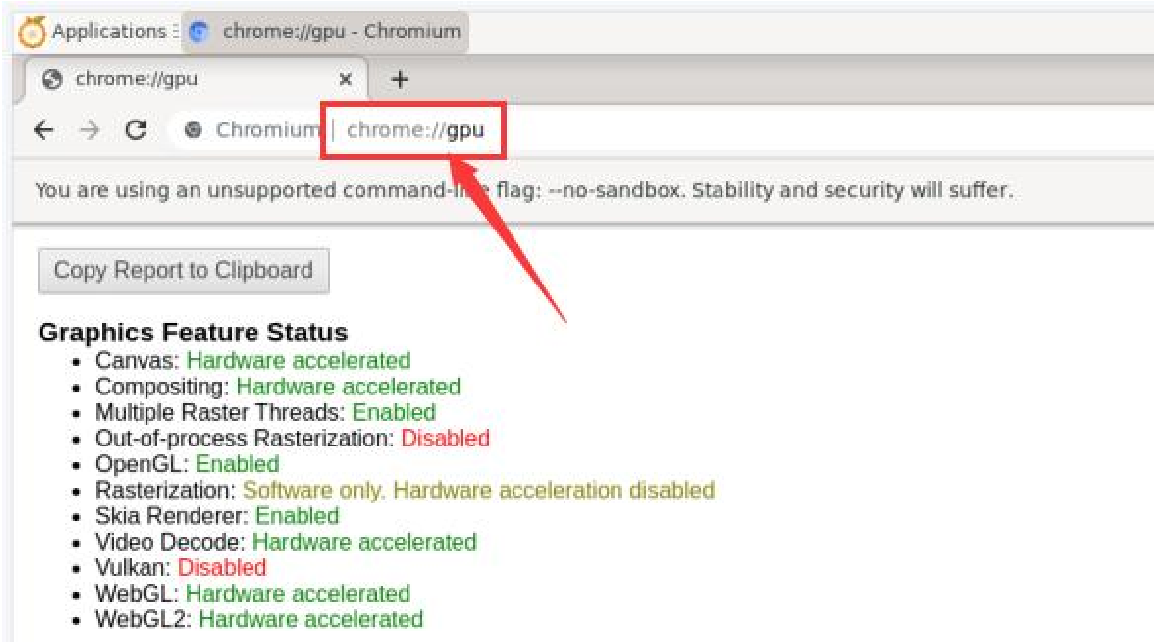


3) You can also open the **Chromium** browser and enter **chrome://gpu** in the address bar to view the GPU support in the Chromium browser

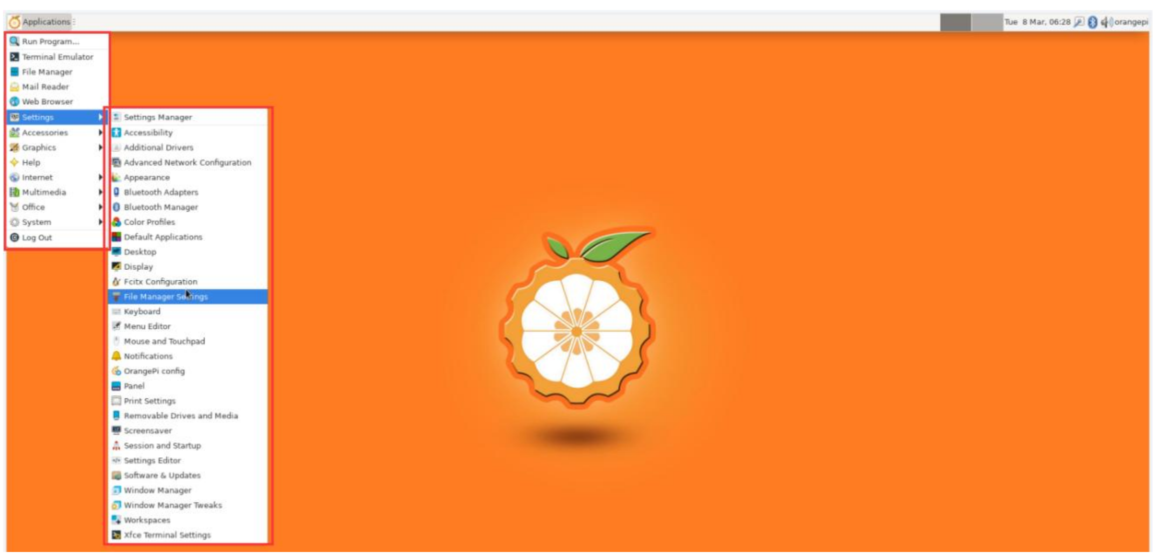
a. The way to open the browser is as follows



b. The GPU support in the Chromium browser is as follows



4) It should be noted that after **Linux5.10 Debian11** turns on GPU hardware acceleration, as shown in the figure below, when dragging with the mouse in the application list in the upper left corner, the speed will be slower than when GPU hardware acceleration is not turned on. This is already know the problem



5) If you do not need GPU hardware acceleration, you can modify the following configuration file to turn off GPU hardware acceleration (**after turning off GPU and VPU, both GPU and VPU cannot be used, including MPV hardware decoding and Chromium hardware acceleration**), and then restart the system, it will not work I have

the above carton problem.

```
root@orangepi4-lts:~# cat /etc/X11/xorg.conf.d/02-modesetting.conf
Section "Device"
        Identifier    "Rockchip Graphics"
        Driver          "modesetting"

### Use Rockchip RGA 2D HW accel
#       Option          "AccelMethod"       "exa"

### Use GPU HW accel
        Option              "AccelMethod"          "none"    #Set to none to turn off GPU acceleration
```

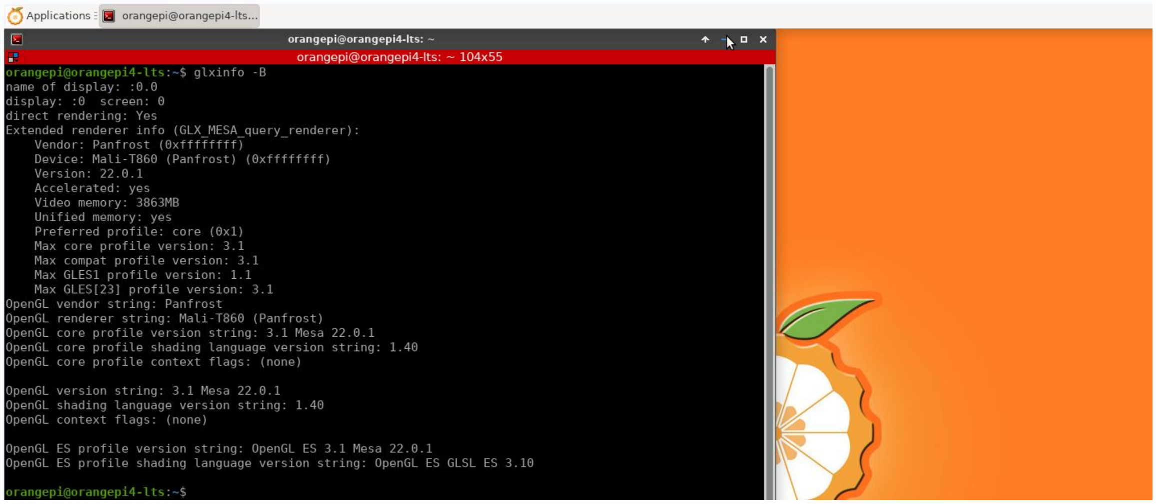
3.16.2. Ubuntu 22.04 Linux5.18 system GPU test instructions

Note that Linux 5.18 uses the open source Panfrost kernel GPU driver and the open source Mesa library, and does not use RK's set of closed source things. First of all, thanks to these excellent developers.

1) First, please connect the HDMI display. All the following commands are operated in the desktop displayed by HDMI, so please do not use ssh remote login or use serial port to log in to the Linux system. If you don't have an HDMI display, you can use NoMachine to remotely log into the desktop of your Linux system.

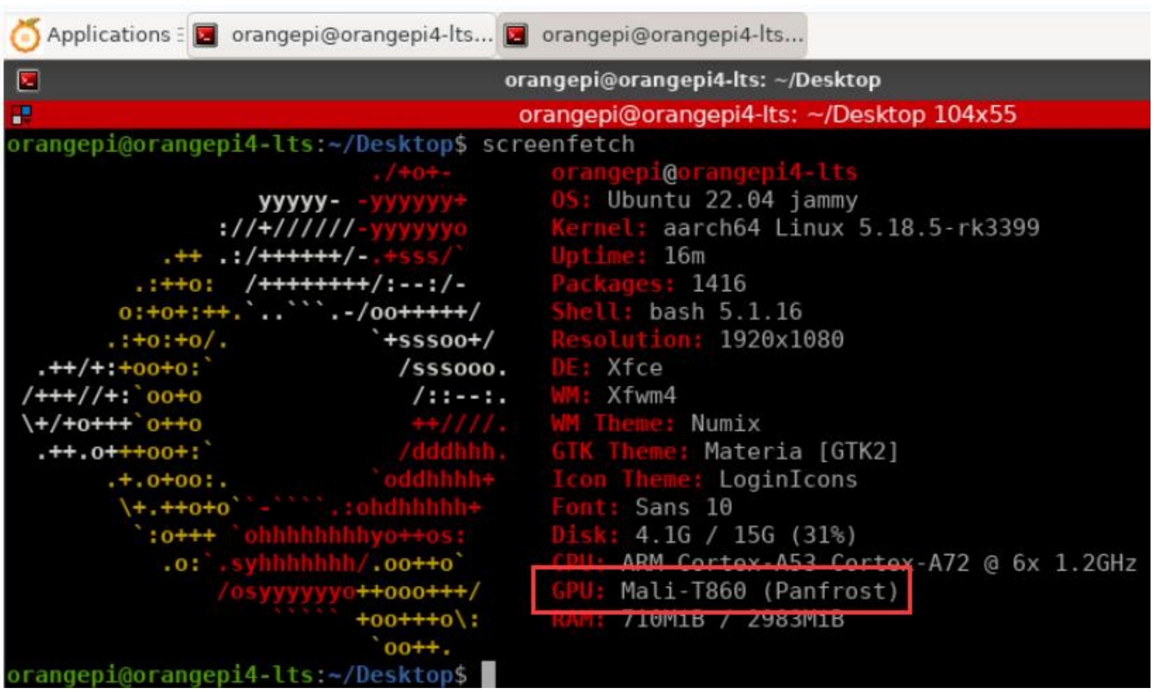
2) After entering the desktop, first open a terminal, and then use the **glxinfo -B** command to see that the GPU driver used is **Mali-T860 (Panfrost)** instead of **llvmpipe**.

```
orangepi@orangepi:~$ glxinfo -B
```



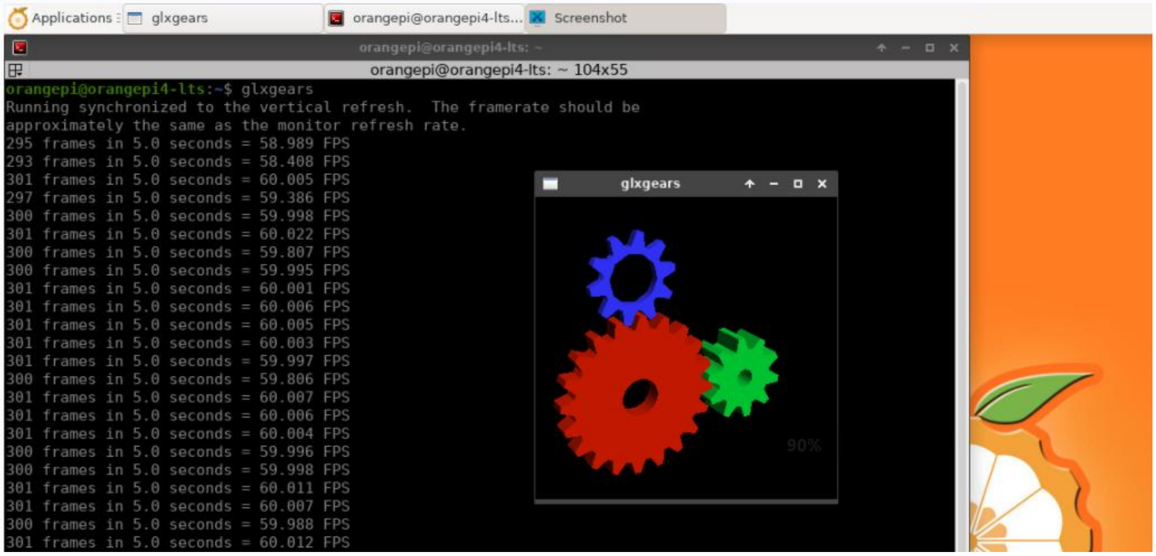
3) Using the screenfetch command, you can also see that the GPU driver is using **Mali-T860 (Panfrost)**。

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



4) The **Glx-Gears** test looks like this:

```
orangepi@orangepi:~$ glxgears
```

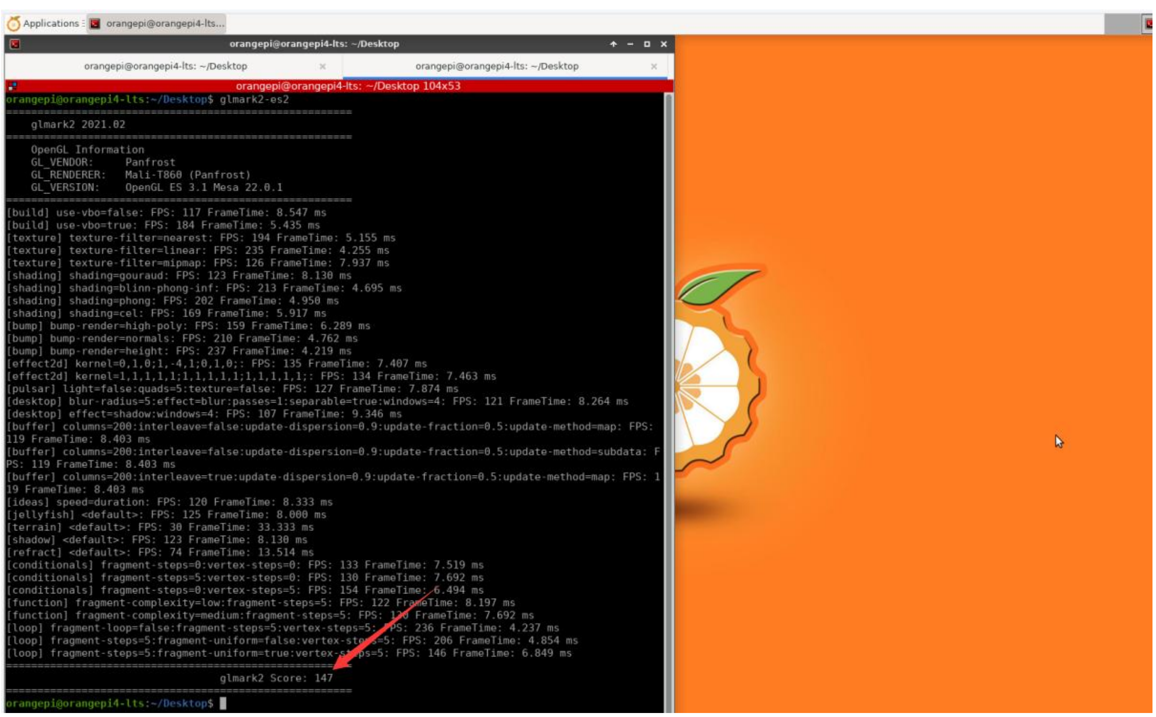


5) **glmark2-es2** is a benchmark tool for OpenGL (ES) 2.0, you can use glmark2 to test the performance of GPU OpenGL ES 2.0。

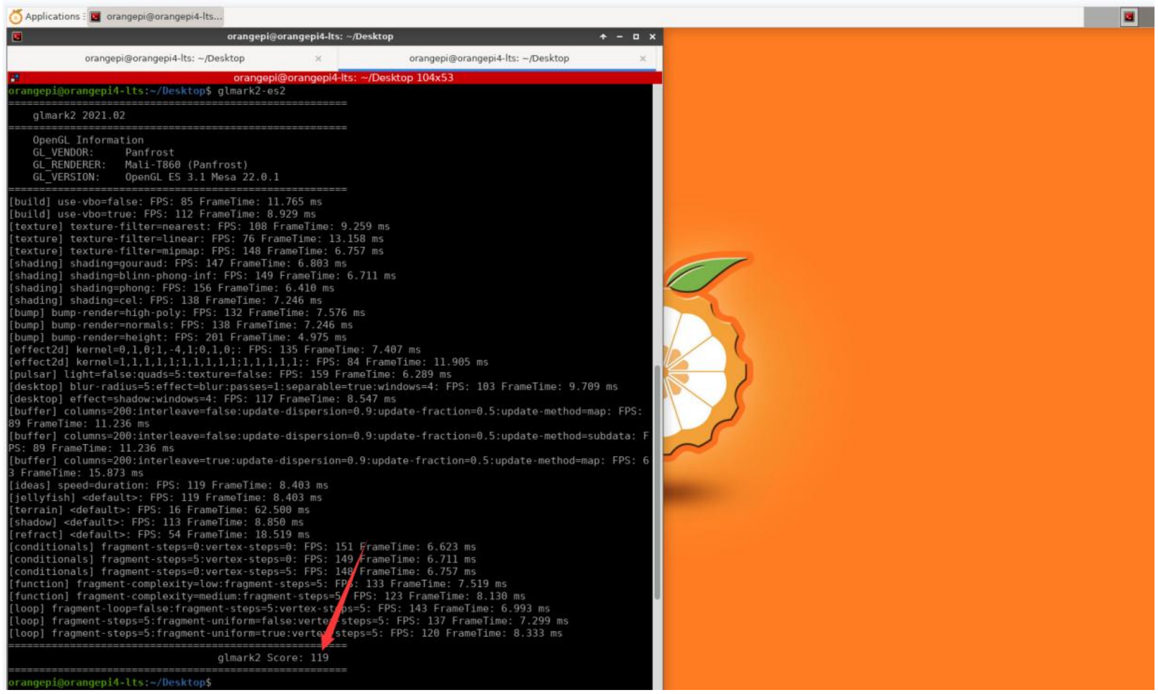
```
orangepi@orangepi:~$ sudo apt install -y glmark2-es2
orangepi@orangepi:~$ glmark2-es2
orangepi@orangepi:~$ glmark2-es2 --off-screen
```

a. The test scores for the **glmark2-es2** command are as follows

a) Test results with fan cooling

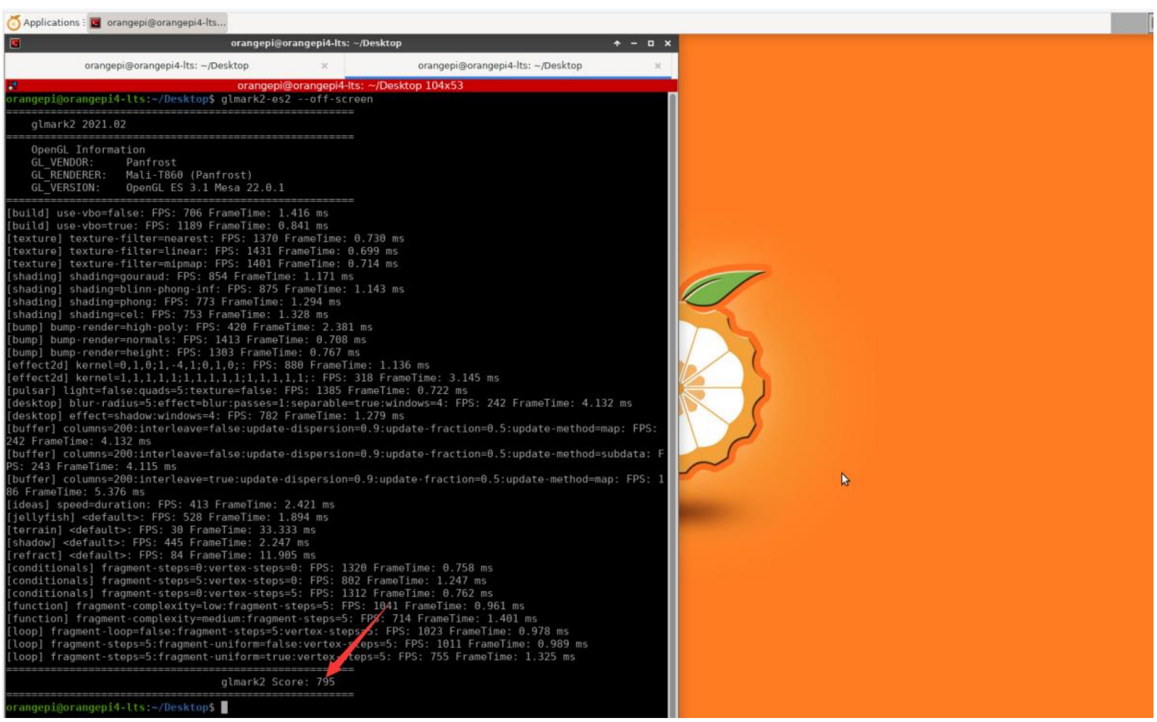


b) Test results without heat dissipation

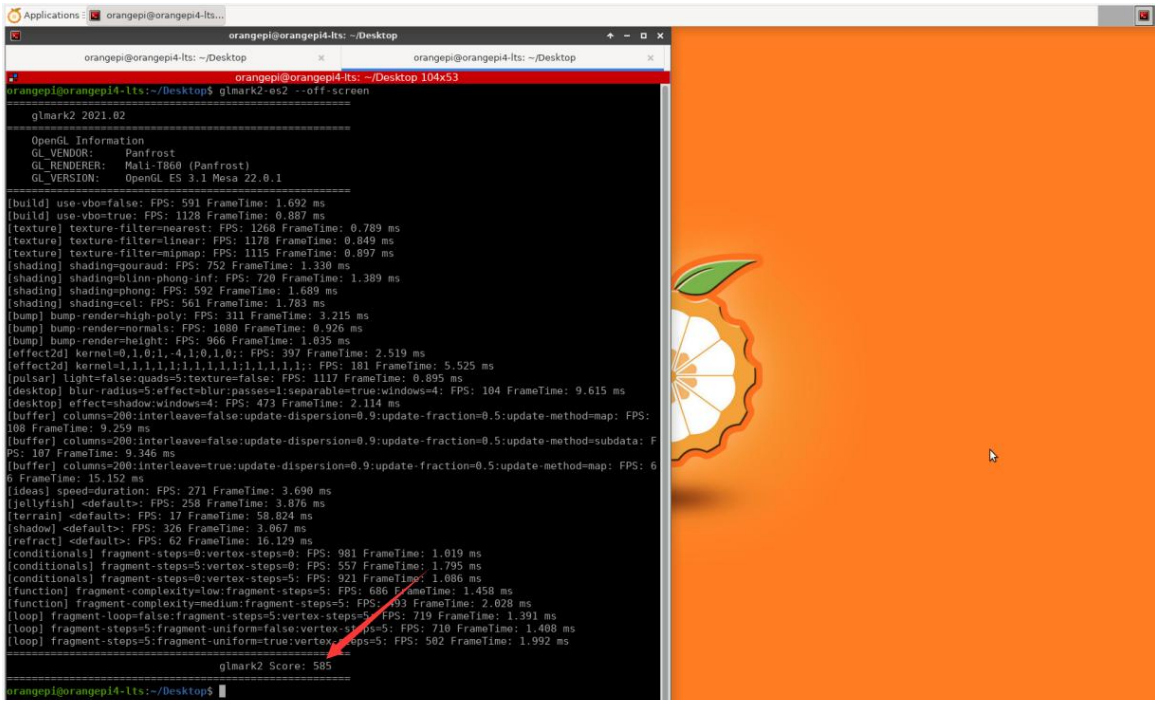


b. The test scores for the **glmark2-es2 --off-screen** command are as follows

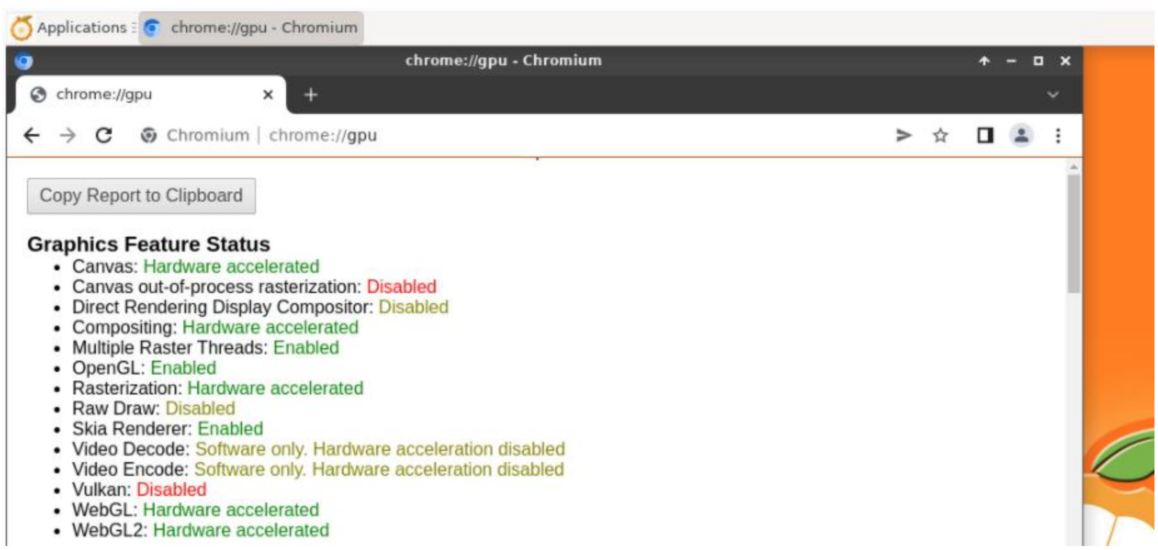
a) Test results with fan cooling



b) Test results without heat dissipation



6) Then open the Chromium browser and enter **chrome://gpu** in the address bar to view the GPU support in the Chromium browser



3. 17. MPV hardware decoding playback video test

Note that only Linux4.4 Debian10 and Linux5.10 Debian11 desktop version systems support this function, and Linux5.10 Ubuntu20.04 and Linux5.18

Ubuntu22.04 do not.

1) The Linux4.4 Debian10 and Linux5.10 Debian11 desktop systems have integrated the MPV player provided by RK, which can call the **rkmpp** decoding plug-in. The currently tested and supported decoding formats are **H264, H265, VP9**

2) The video file for testing can be downloaded by opening the link below

a. The video download link in H264 format is

https://test-videos.co.uk/vids/jellyfish/mp4/h264/1080/Jellyfish_1080_10s_30MB.mp4
http://bbb3d.renderfarming.net/download.html

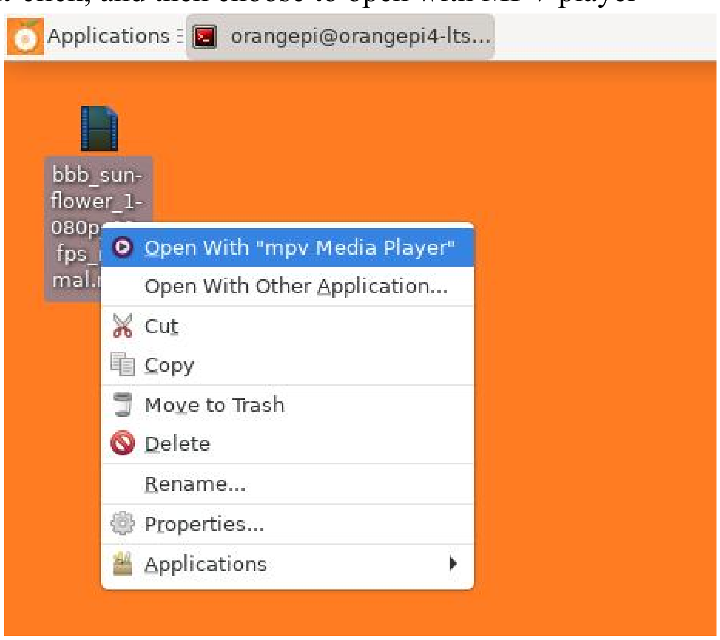
b. The video download link in H265 format is

https://test-videos.co.uk/vids/jellyfish/mp4/h265/1080/Jellyfish_1080_10s_30MB.mp4

c. The video download link in VP9 format is

https://test-videos.co.uk/vids/jellyfish/webm/vp9/1080/Jellyfish_1080_10s_30MB.webm

3) Then copy the video file for testing to the desktop, then select the video file to be played, then right-click, and then choose to open with MPV player



4) Then the video will start to play. At this time, you can open the terminal and enter the top command to check the CPU usage. If the CPU usage is relatively low, it means that the video is being hard-decoded by using the VPU.

5) At this time, enter the following command in the terminal to turn on the debugging switch of the VPU driver. If you can see the following printing information, it means that the VPU is being used for video decoding

a. The command for Linux4.4 is as follows

```
root@orangepi4-lts:~# echo 0x100 > \
 /sys/module/rk_vcodec/parameters/debug
```

b. The command for Linux5.10 is as follows

```
root@orangepi4-lts:~# echo 0x100 > \
 /sys/module/rk_vcodec/parameters/mpp_dev_debug
```

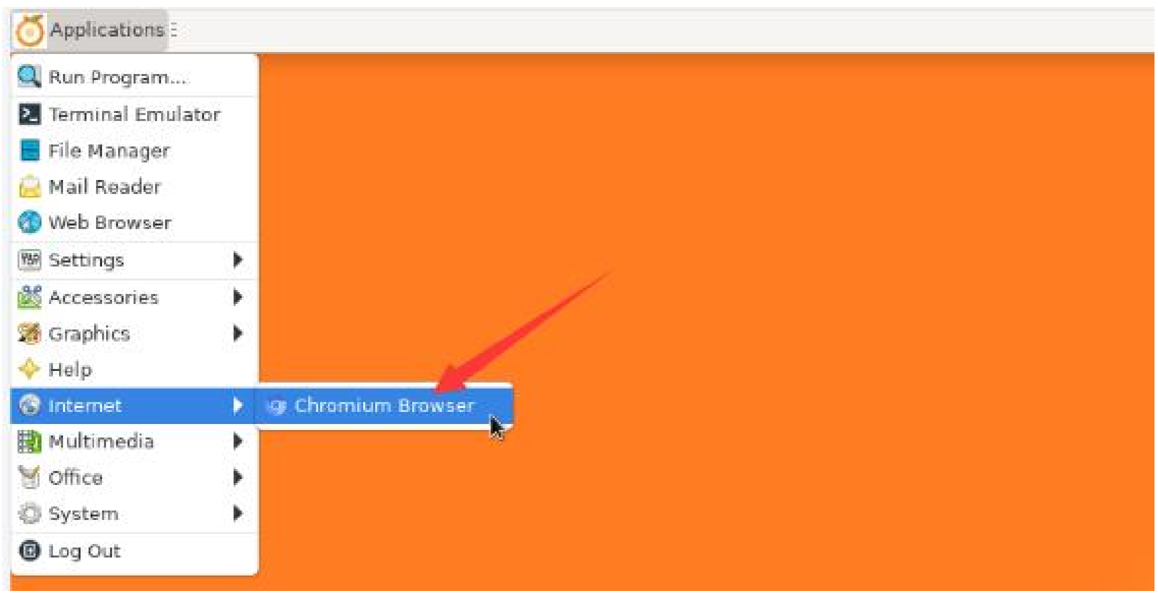
```
root@orangepi4-lts:~# dmesg -c
root@orangepi4-lts:~# dmesg
[ 1229.295635] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 5262 us
[ 1229.309028] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 3210 us
[ 1229.320295] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 5366 us
[ 1229.326177] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 3438 us
[ 1229.332428] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 3708 us
[ 1229.342385] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 7921 us
[ 1229.354728] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 6564 us
[ 1229.364151] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 7135 us
[ 1229.375270] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 7242 us
```

6) Use the left and right arrow keys to control the video playback progress

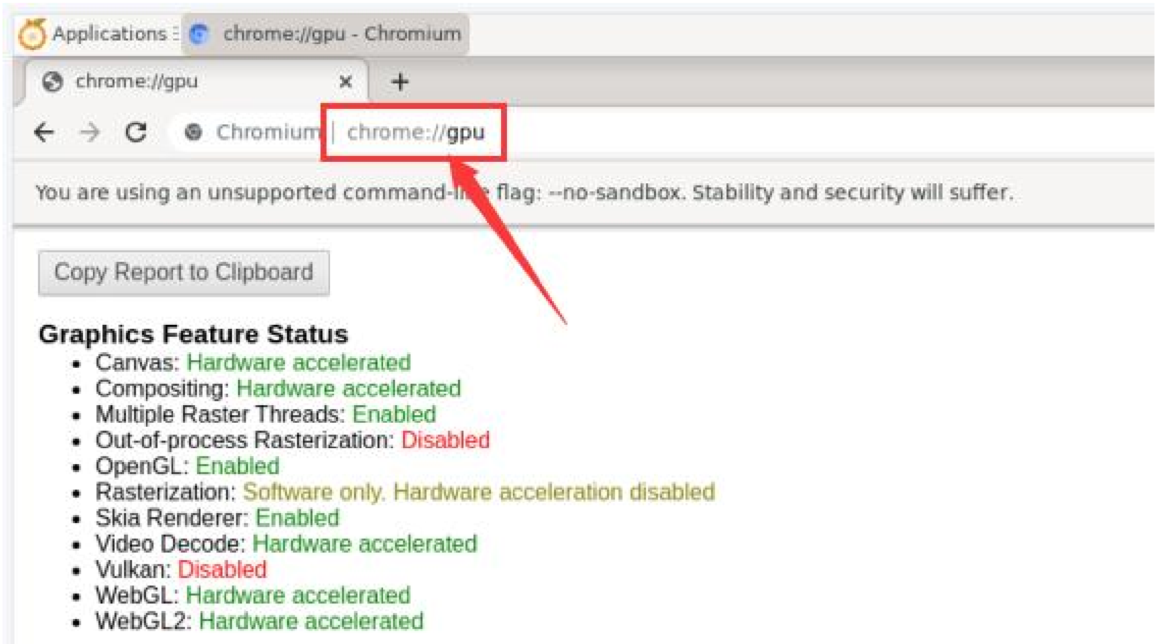
3. 18. Chromium browser hardware decoding video playback test

Note that only Linux4.4 Debian10 and Linux5.10 Debian11 desktop systems support hardware acceleration for video decoding in Chromium browsers, other systems do not. In addition, the installation package of the Chromium browser is provided by RK. Since the code of RK's adaptation to the Chromium browser is not open, we cannot update it. If you use other methods (such as apt-get install) to update the Chromium browser, this function will not work.

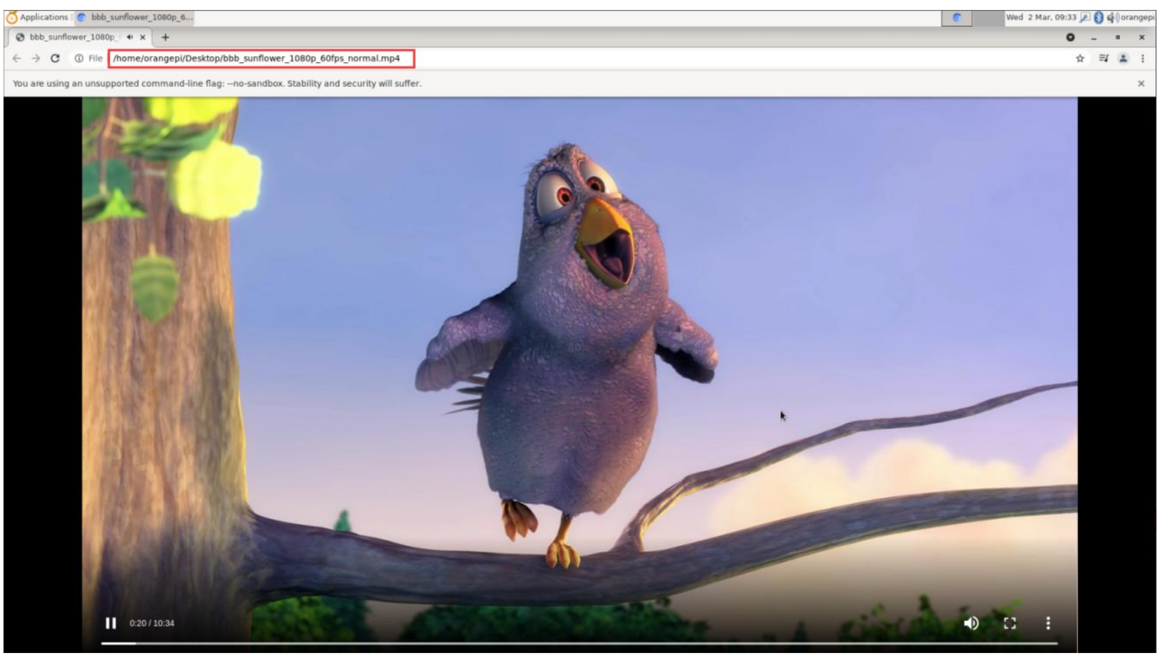
1) First open the chromium browser



2) Then enter **chrome://gpu** in the address bar of the browser to view the support of hardware acceleration. You can see from the figure below that the **Video Decode** feature has been **Enabled**



3) Enter the path of the local video file in the address bar of the chromium browser. After pressing Enter, the video will start to play. If there is no test video locally, you can also choose to access the online video.



4) At this time, enter the following command in the terminal to turn on the debugging switch of the VPU driver. If you can see the following print information, it means that the VPU is being used for video decoding

a. The command for Linux4.4 is as follows

```
root@orangepi4-lts:~# echo 0x100 >   \
 /sys/module/rk_vcodec/parameters/debug
```

b. The command for Linux5.10 is as follows

```
root@orangepi4-lts:~# echo 0x100 >   \
 /sys/module/rk_vcodec/parameters/mpp_dev_debug
```

```
root@orangepi4-lts:~# dmesg -c
root@orangepi4-lts:~# dmesg
[ 1229.295635] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 5262 us
[ 1229.309028] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 3210 us
[ 1229.320295] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 5366 us
[ 1229.326177] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 3438 us
[ 1229.332428] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 3708 us
[ 1229.342385] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 7921 us
[ 1229.354728] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 6564 us
[ 1229.364151] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 7135 us
[ 1229.375270] rk_vcodec: ff660000.rkvdec: pid: 4561, session: 00000000c9c1ef4e, time: 7242 us
```

3.19. Ubuntu22.04 Linux5.18 Kodi hard solution play video instructions

Note that only the image of the Ubuntu22.04 Linux5.18 kernel supports hard decoding and playback of video through Koid, so please make sure that the image used is the following two versions of the image:

Orangepi4-lts_3.0.x_ubuntu_jammy_server_linux5.18.5.7z

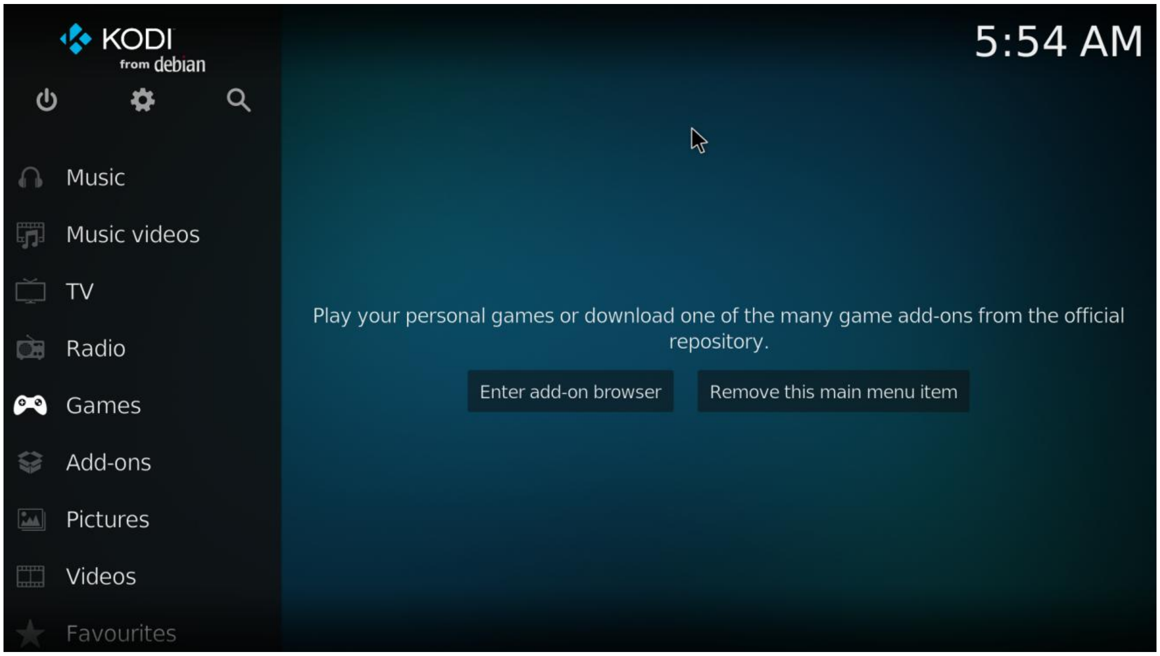
Orangepi4-lts_3.0.x_ubuntu_jammy_desktop_xfce_linux5.18.5.7z

First, connect the development board to the HDMI monitor, keyboard and mouse, and then perform the following operations on the HDMI monitor.

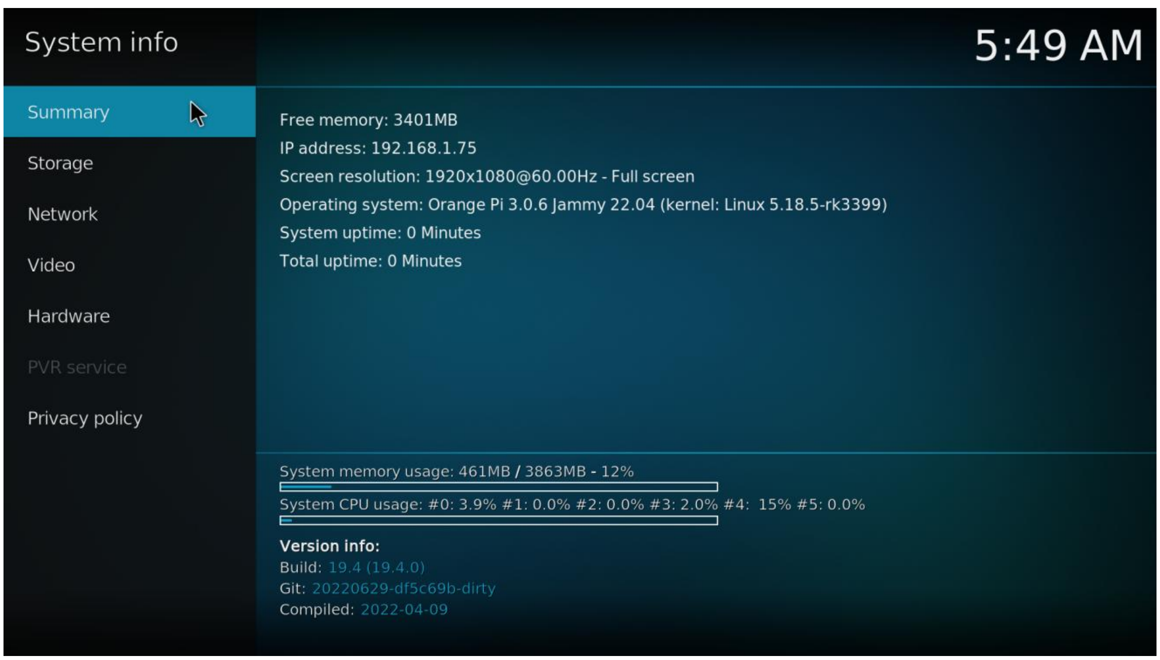
1) If you are using an image of the desktop version (with desktop in the image name), first switch the HDMI display interface from the desktop to the command line terminal by pressing **Ctrl+Alt+F2** on the keyboard, or follow the Linux desktop version system. **The instructions in the Methods for Disabling the Desktop section** close the desktop. The currently adapted Kodi cannot hard-decode and play videos on X11. If you are using an image of the server version (with server in the image name), you can log in directly.

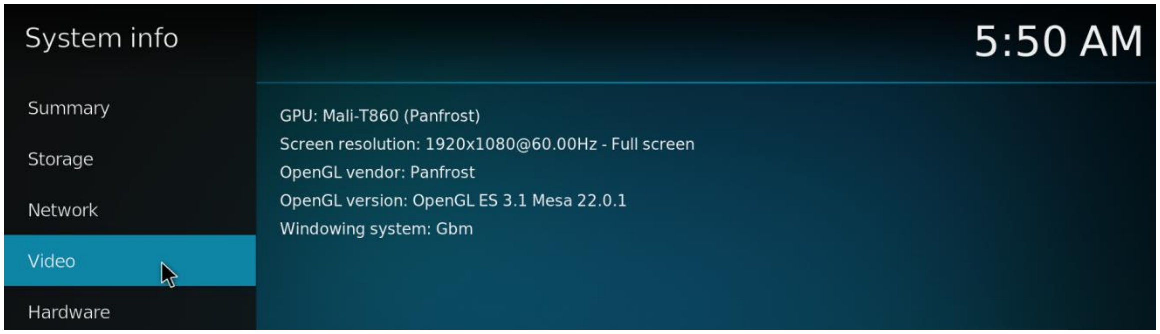
2) After logging in to the system, run the following command to open kodi:

```
orangepi@orangepi:~$ kodi
```



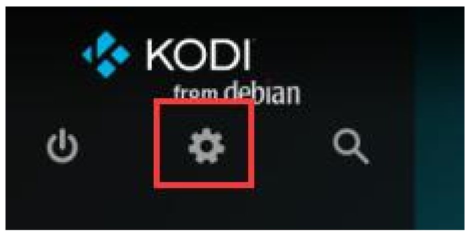
3) In the settings, you can see the basic information of the system as shown below





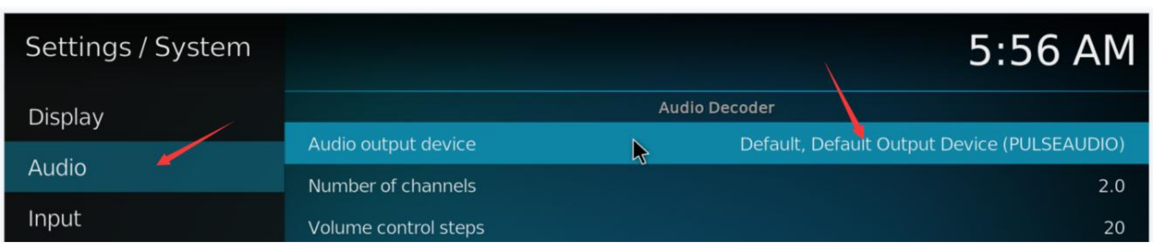
4) Here's how to switch audio output devices in Kodi：

a. Click the Settings button

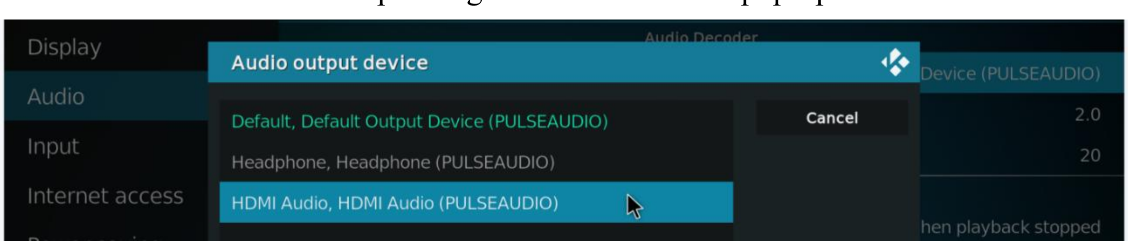


b. Then select **System**

c. Then select **Audio**, and then click the **Audio output device** column

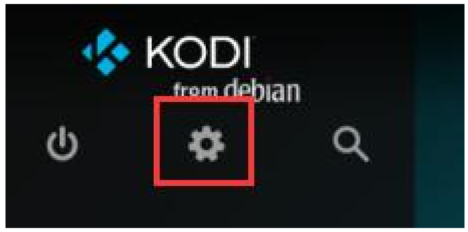


d. Then select the corresponding audio device in the pop-up selection box.



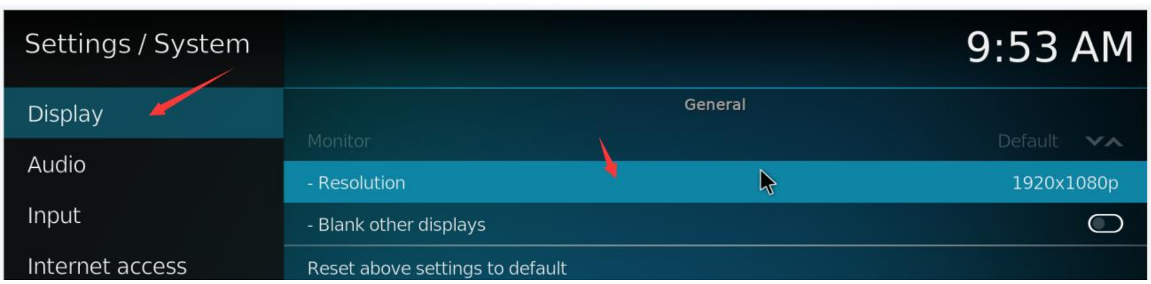
5) The method to set HDMI resolution is as follows:

a. Click the Settings button

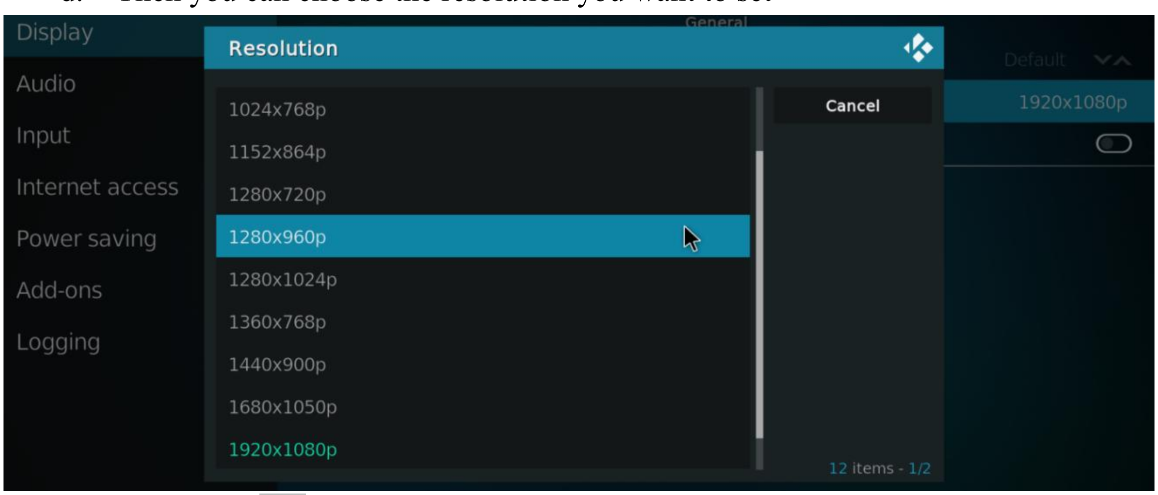


b. Then select **System**

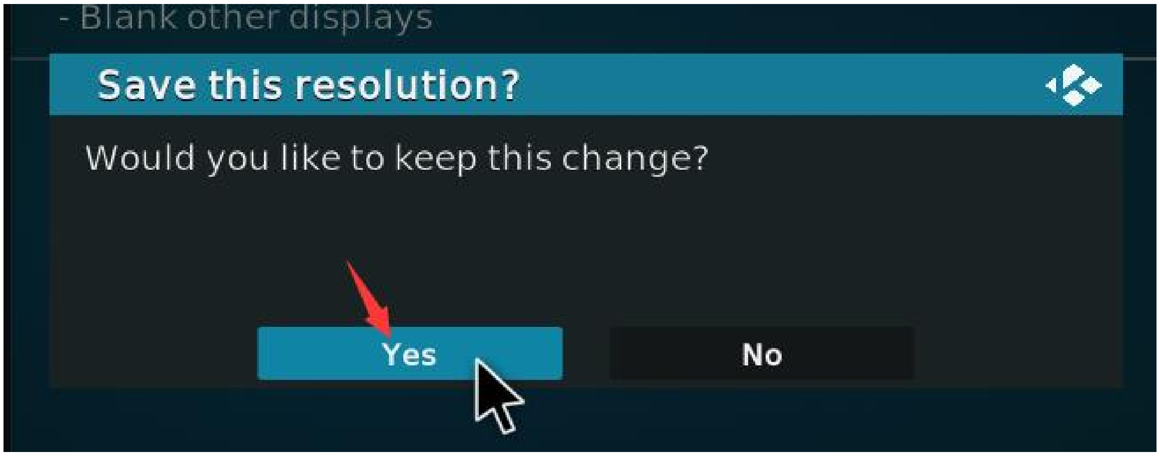
c. Then select **Display**, then click **Resolution**



d. Then you can choose the resolution you want to set



e. Then click **Yes** to save the settings



6) The method to play the video is as follows:

   a. Currently tested supported video formats are **H264, H265, VP8, VP9** and **MPEG2**

   b. The tested video can be downloaded in Google cloud disk, from which you can see more detailed information of the test video. The Google cloud disk link is as follows：

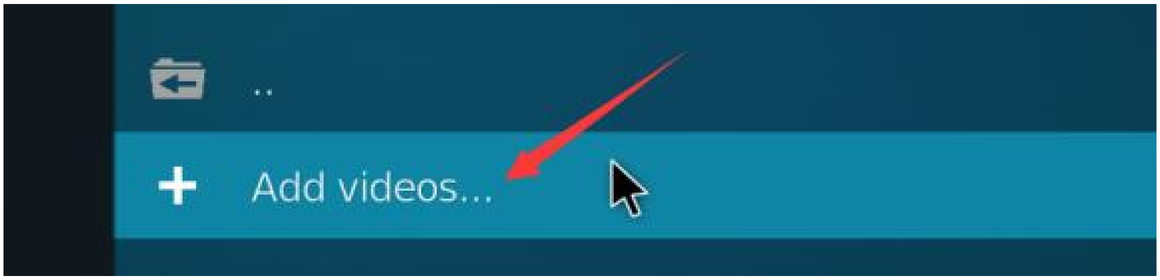
Link:https://drive.google.com/drive/folders/1KtVFs4o4VQbyhMIkMkmiMW6CWyTG-tKU?usp=sharing

   c. After downloading the **test_video** folder, please upload it to the **/home/orangepi** directory of the Linux system of the development board. The upload command in Ubuntu computer is as follows：

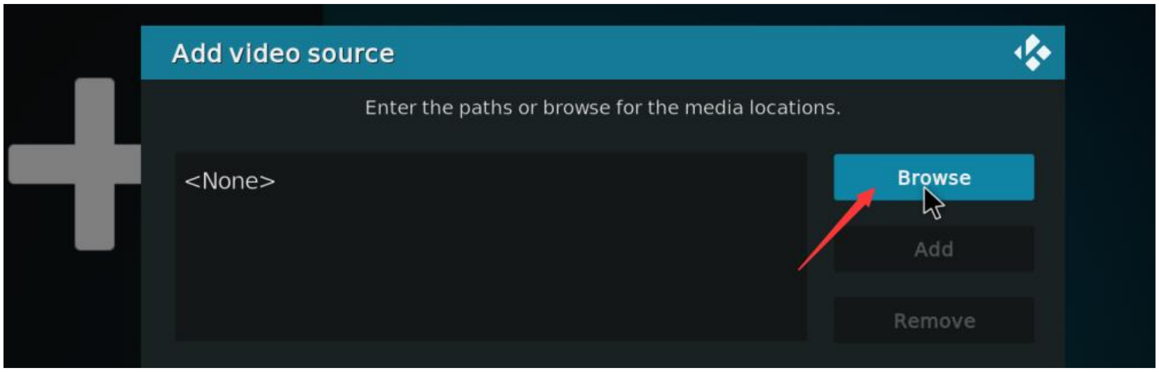
test@test:~$ **scp -r test_video orangepi@192.168.1.xx:/home/orangepi/**

   d. Then go back to the main Kodi interface and select **Movies**

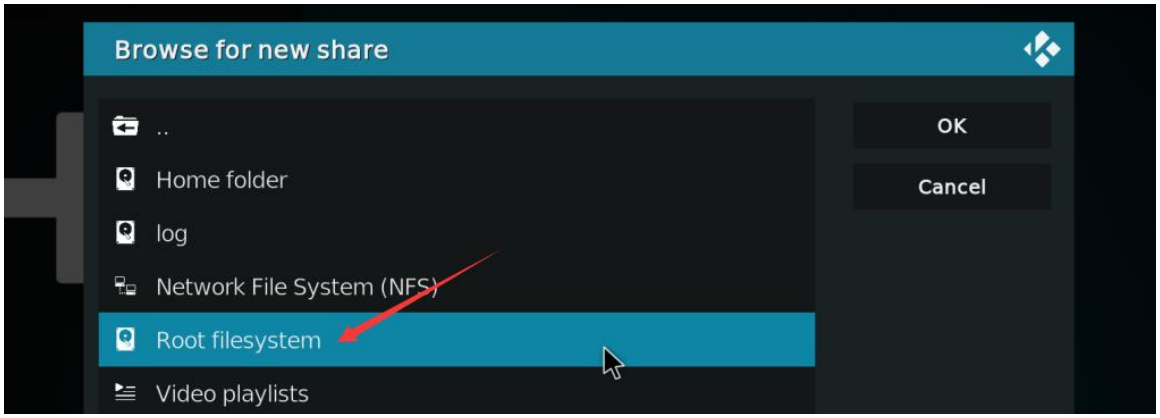
e. Then click **Add videos**



f. Then click **Browse**

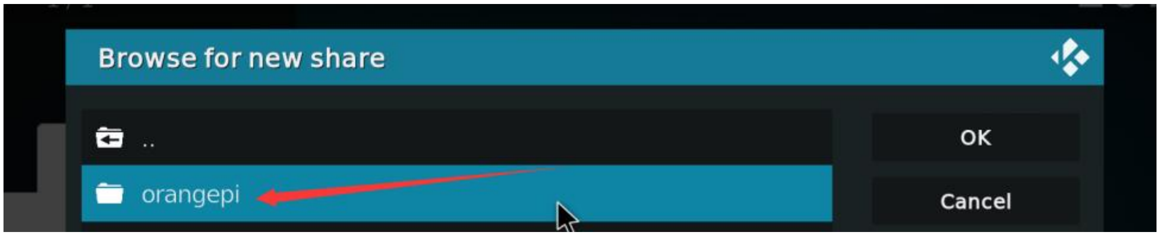


g. Then click **Root filesystem**



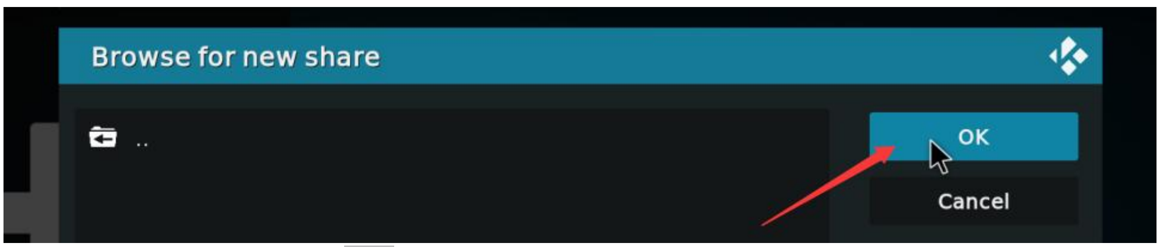
h. Then click **home** directory

i. Then select the **orangepi** directory

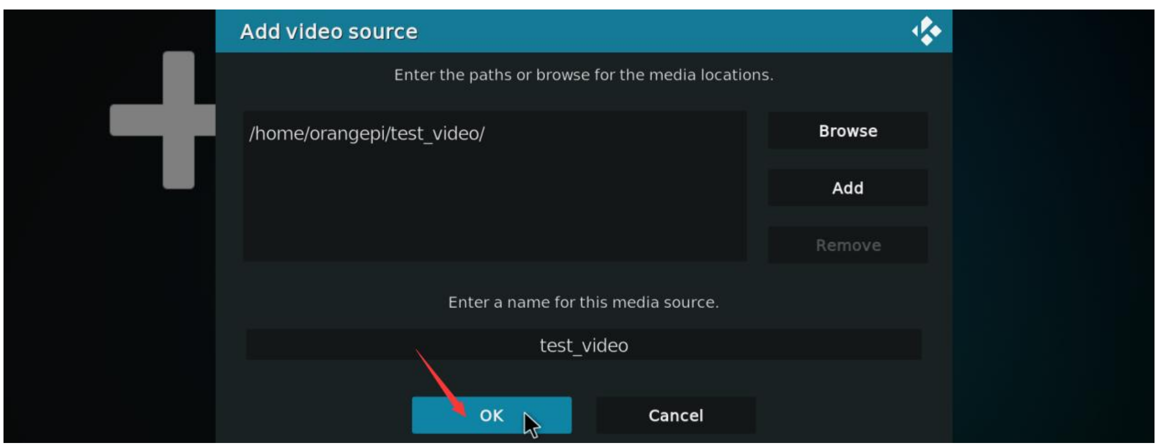


j. Then select the **test_video** test video folder just uploaded

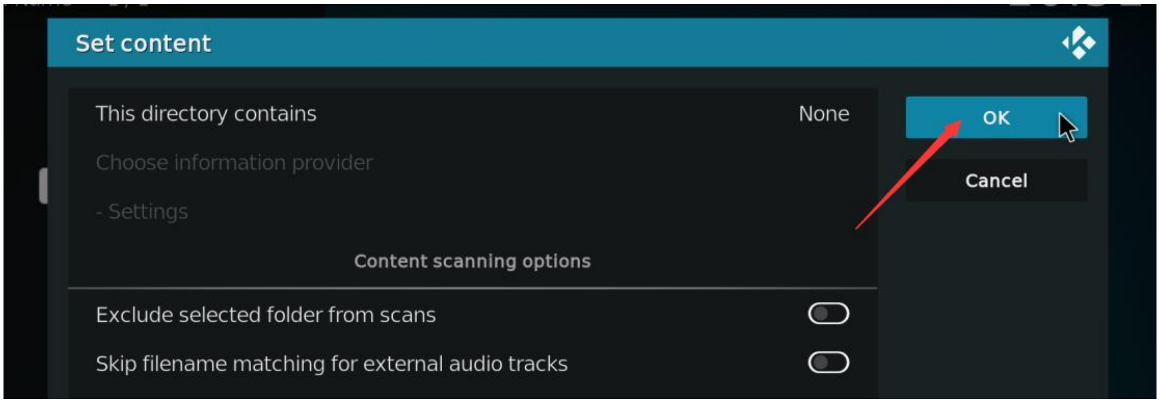
k. Then click **OK**



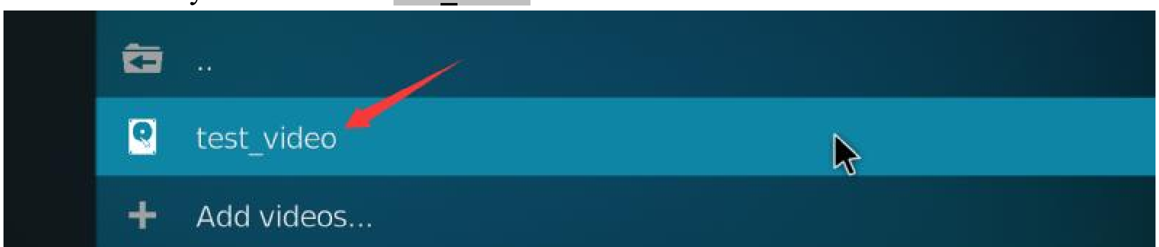
l. Continue to click **OK**



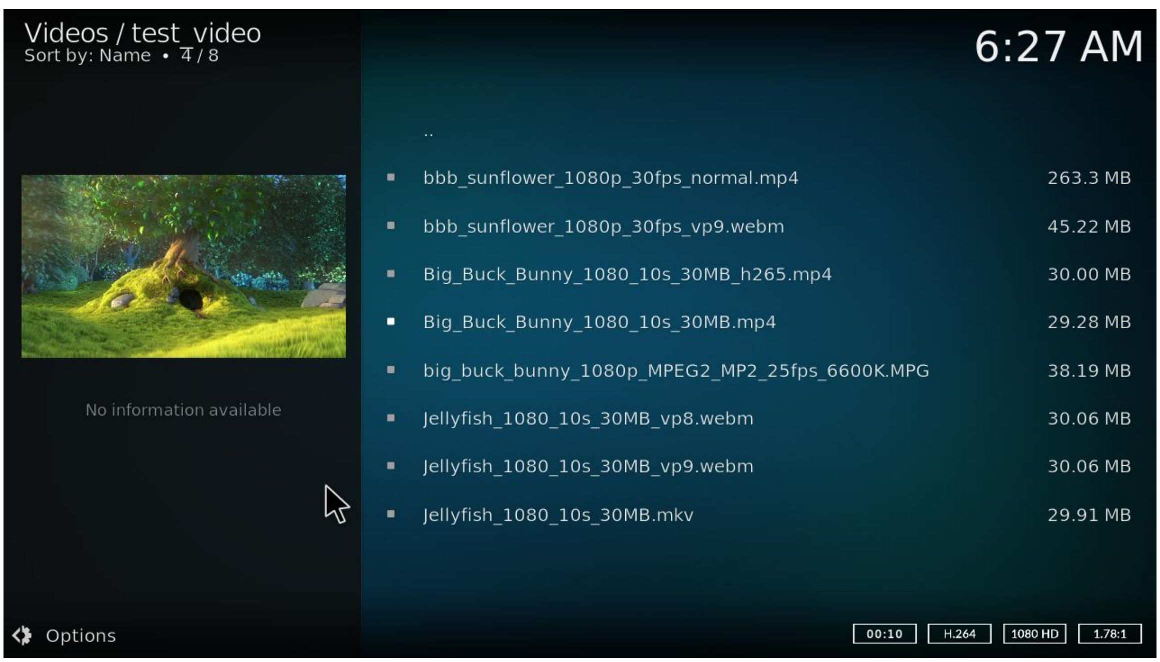
m. Continue to click **OK**



n. Then you can see the **test_video** folder



o. Click the **test_video** playback folder again to view the videos playing in it



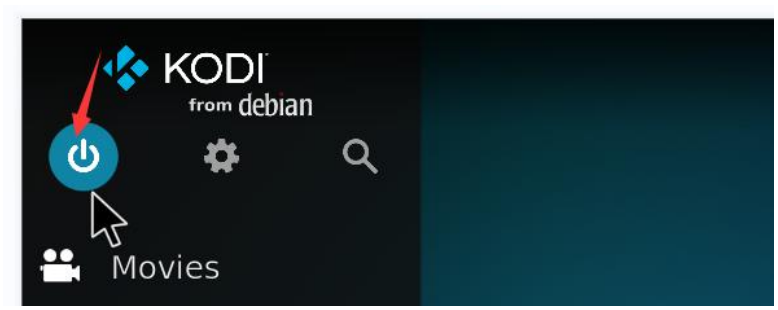
p. When playing a video, you can open a terminal window and use the top command to check the CPU usage. In actual testing, the CPU usage is generally only **30%~70%** when playing the video. If there is no hard solution, the CPU usage of playing video will reach **300%~600%**.

```
orangepi@orangepi:~$ top
```

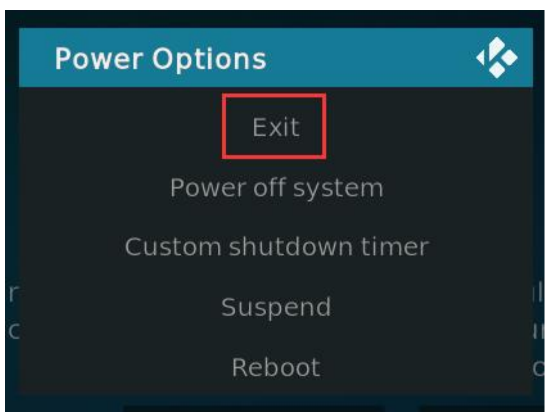
```
top - 06:29:39 up 41 min,  4 users,  load average: 1.51, 1.46, 1.36
Tasks: 222 total,   3 running, 219 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.5 us,  2.1 sy,  0.0 ni, 94.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  3863.3 total,  2443.8 free,   535.4 used,   884.0 buff/cache
MiB Swap:  1931.6 total,  1931.6 free,     0.0 used.  3134.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2775 orangepi  20   0 3066268 219236 131704 R  30.8   5.5   9:37.44 kodi.bin
   3771 orangepi  20   0   15804   3624   2844 R   2.0   0.1   0:00.37 top
   1910 orangepi  20   0 1230704  19892  15476 R   1.3   0.5   0:11.12 pulseaudio
     14 root      20   0       0      0      0 I   0.3   0.0   0:03.08 rcu_preempt
    599 root      -2   0       0      0      0 S   0.3   0.0   0:01.95 pan_js
    600 root      -2   0       0      0      0 S   0.3   0.0   0:00.79 pan_js
```

7) Here's how to quit Kodi：

a. First click on the power button in the upper left corner of the main Kodi interface。



b. Then click **Exit** to exit Kodi

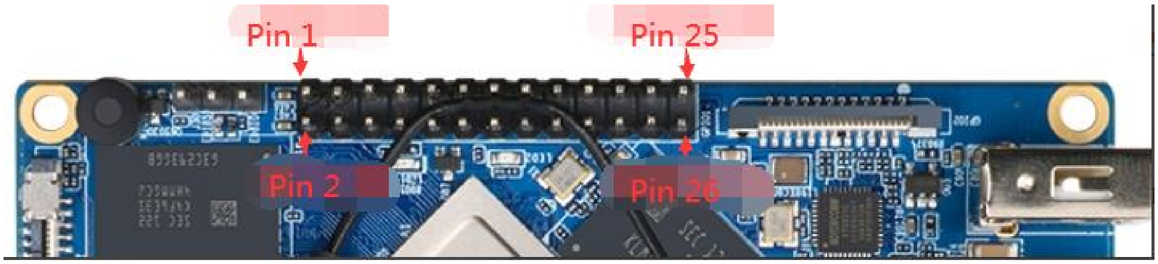


c. If you are using the desktop version system, you can switch to the command line through the **Ctrl+Alt+F2** key combination before, and you can also switch back to the desktop through the **Ctrl+Alt+F7** key combination.

3. 20. 26pin interface pin description

1) Please refer to the figure below for the order of the 26 pin interface pins of the Orange

Pi 4 LTS development board



2) The functions of the 40 pin interface pins of the Orange Pi 4 LTS development board are shown in the table below

GPIO No.	GPIO	Function	pin
		3.3V	1
52	GPIO1_C4	I2C8_SDA	3
53	GPIO1_C5	I2C8_SCL	5
150	GPIO4_C6	PWM1	7
		GND	9
33	GPIO1_A1	GPIO1_A1	11
35	GPIO1_A3	GPIO1_A3	13
92	GPIO2_D4	GPIO2_D4	15
		3.3V	17
40	GPIO1_B0	SPI1_TXD	19
39	GPIO1_A7	SPI1_RXD	21
41	GPIO1_B1	SPI1_CLK	23

pin	Function	GPIO	GPIO No.
2	5V		
4	5V		
6	GND		
8	I2C3_SCL	GPIO4_C1	145
10	I2C3_SDA	GPIO4_C0	144
12	GPIO1_C2	GPIO1_C2	50
14	GND		
16	GPIO1_C6	GPIO1_C6	54
18	GPIO1_C7	GPIO1_C7	55
20	GND		
22	GPIO1_D0	GPIO1_D0	56
24	SPI1_CS	GPIO1_B2	42

		GND	**25**	**26**	**GPIO4_C5**	**GPIO4_C5**	**149**

3) There are a total of 17 GPIO ports in the 26pin interface. Except the voltage of GPIO2_D4 is **1.8V,** the voltage of all other GPIO ports is **3.0v**

3. 21. How to install wiringOP

1) Download the code of wiringOP

```
root@orangepi4-lts:~# apt-get update
root@orangepi4-lts:~# apt-get -y install git
root@orangepi4-lts:~# git clone https://github.com/orangepi-xunlong/wiringOP
```

2) Compile wiringOP

```
root@orangepi4-lts:~# cd wiringOP
root@orangepi4-lts:~/wiringOP# ./build clean
root@orangepi4-lts:~/wiringOP# ./build
```

3) The output of the test gpio readall command is as follows, in which the physical pins 1 to 26 are in one-to-one correspondence with the 26Pin pins on the development board

```
root@orangepi4-lts:~# gpio readall
 +------+-----+----------+------+---+OPi 4 LTS +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3V |      |   |  1 || 2  |   |      | 5V       |     |      |
 |   64 |   0 | I2C8_SDA | ALT2 | 1 |  3 || 4  |   |      | 5V       |     |      |
 |   65 |   1 | I2C8_SCL | ALT2 | 1 |  5 || 6  |   |      | GND      |     |      |
 |  150 |   2 |     PWM1 |   IN | 0 |  7 || 8  | 1 | ALT2 | I2C3_SCL | 3   | 145  |
 |      |     |      GND |      |   |  9 || 10 | 1 | ALT2 | I2C3_SDA | 4   | 144  |
 |   33 |   5 | GPIO1_A1 |   IN | 0 | 11 || 12 | 1 | IN   | GPIO1_C2 | 6   | 50   |
 |   35 |   7 | GPIO1_A3 |  OUT | 1 | 13 || 14 |   |      | GND      |     |      |
 |   92 |   8 | GPIO2_D4 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO1_C6 | 9   | 54   |
 |      |     |     3.3V |      |   | 17 || 18 | 0 | IN   | GPIO1_C7 | 10  | 55   |
 |   40 |  11 | SPI1_TXD | ALT3 | 0 | 19 || 20 |   |      | GND      |     |      |
 |   39 |  12 | SPI1_RXD | ALT3 | 1 | 21 || 22 | 0 | IN   | GPIO1_D0 | 13  | 56   |
 |   41 |  14 | SPI1_CLK | ALT3 | 1 | 23 || 24 | 1 | ALT3 | SPI1_CS  | 15  | 42   |
 |      |     |      GND |      |   | 25 || 26 | 0 | IN   | GPIO4_C5 | 16  | 149  |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+OPi 4 LTS +---+---+--+----------+-----+------+
```

3.22. 26pin interface GPIO, I2C, UART, SPI, PWM test

wiringOP has been adapted to the Orange Pi development board, and wiringOP can be used to test the functions of GPIO, I2C, UART and SPI.

Before starting the test, make sure you have compiled and installed wiringOP by referring to the section Installing wiringOP.

3.22.1. 26pin GPIO port test

1) The following uses pin No. 11—the corresponding GPIO is GPIO1_A1—the corresponding wPi serial number is 5—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepi4-lts:~# gpio readall
 +------+-----+----------+------+---+OPi 4 LTS +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3V |      |   |  1 || 2  |   |      | 5V       |     |      |
 |   64 |   0 | I2C8_SDA | ALT2 | 1 |  3 || 4  |   |      | 5V       |     |      |
 |   65 |   1 | I2C8_SCL | ALT2 | 1 |  5 || 6  |   |      | GND      |     |      |
 |  150 |   2 |     PWM1 |   IN | 0 |  7 || 8  | 1 | ALT2 | I2C3_SCL | 3   | 145  |
 |      |     |      GND |      |   |  9 || 10 | 1 | ALT2 | I2C3_SDA | 4   | 144  |
 |   33 |   5 | GPIO1_A1 |   IN | 0 | 11 || 12 | 1 | IN   | GPIO1_C2 | 6   | 50   |
 |   35 |   7 | GPIO1_A3 |  OUT | 1 | 13 || 14 |   |      | GND      |     |      |
 |   92 |   8 | GPIO2_D4 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO1_C6 | 9   | 54   |
 |      |     |     3.3V |      |   | 17 || 18 | 0 | IN   | GPIO1_C7 | 10  | 55   |
```

2) First set the GPIO port as output mode, where the third parameter requires the serial number of the wPi corresponding to the input pin

```
root@orangepi4-lts:~/wiringOP# gpio mode 5 out
```

Use gpio readall to see that the mode of pin 11 has changed to out

```
root@orangepi4-lts:~# gpio readall
 +------+-----+----------+------+---+OPi 4 LTS +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3V |      |   |  1 || 2  |   |      | 5V       |     |      |
 |   64 |   0 | I2C8_SDA | ALT2 | 1 |  3 || 4  |   |      | 5V       |     |      |
 |   65 |   1 | I2C8_SCL | ALT2 | 1 |  5 || 6  |   |      | GND      |     |      |
 |  150 |   2 |     PWM1 |   IN | 0 |  7 || 8  | 1 | ALT2 | I2C3_SCL | 3   | 145  |
 |      |     |      GND |      |   |  9 || 10 | 1 | ALT2 | I2C3_SDA | 4   | 144  |
 |   33 |   5 | GPIO1_A1 |  OUT | 0 | 11 || 12 | 1 | IN   | GPIO1_C2 | 6   | 50   |
 |   35 |   7 | GPIO1_A3 |  OUT | 1 | 13 || 14 |   |      | GND      |     |      |
 |   92 |   8 | GPIO2_D4 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO1_C6 | 9   | 54   |
 |      |     |     3.3V |      |   | 17 || 18 | 0 | IN   | GPIO1_C7 | 10  | 55   |
```

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

successfully set.

root@orangepi4-lts:~/wiringOP# **gpio write 5 0**

Use gpio readall to see that the value of pin 11 (V) has become 0

```
root@orangepi4-lts:~# gpio readall
 +------+-----+----------+------+---+OPi 4 LTS +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3V |      |   |  1 || 2  |   |      | 5V       |     |      |
 |   64 |   0 | I2C8_SDA | ALT2 | 1 |  3 || 4  |   |      | 5V       |     |      |
 |   65 |   1 | I2C8_SCL | ALT2 | 1 |  5 || 6  |   |      | GND      |     |      |
 |  150 |   2 |     PWM1 |   IN | 0 |  7 || 8  | 1 | ALT2 | I2C3_SCL | 3   | 145  |
 |      |     |      GND |      |   |  9 || 10 | 1 | ALT2 | I2C3_SDA | 4   | 144  |
 |   33 |   5 | GPIO1_A1 |  OUT | 0 | 11 || 12 | 1 | IN   | GPIO1_C2 | 6   | 50   |
 |   35 |   7 | GPIO1_A3 |  OUT | 1 | 13 || 14 |   |      | GND      |     |      |
 |   92 |   8 | GPIO2_D4 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO1_C6 | 9   | 54   |
 |      |     |     3.3V |      |   | 17 || 18 | 0 | IN   | GPIO1_C7 | 10  | 55   |
```

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 successfully set.

root@orangepi4-lts:~/wiringOP# **gpio write 5 1**

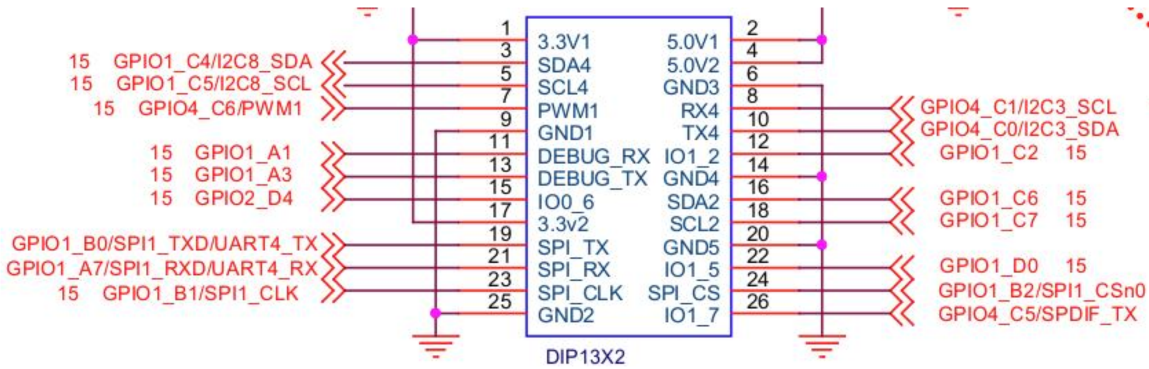
Use gpio readall to see that the value (V) of pin 11 has changed to 1

```
root@orangepi4-lts:~# gpio readall
 +------+-----+----------+------+---+OPi 4 LTS +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3V |      |   |  1 || 2  |   |      | 5V       |     |      |
 |   64 |   0 | I2C8_SDA | ALT2 | 1 |  3 || 4  |   |      | 5V       |     |      |
 |   65 |   1 | I2C8_SCL | ALT2 | 1 |  5 || 6  |   |      | GND      |     |      |
 |  150 |   2 |     PWM1 |   IN | 0 |  7 || 8  | 1 | ALT2 | I2C3_SCL | 3   | 145  |
 |      |     |      GND |      |   |  9 || 10 | 1 | ALT2 | I2C3_SDA | 4   | 144  |
 |   33 |   5 | GPIO1_A1 |  OUT | 1 | 11 || 12 | 1 | IN   | GPIO1_C2 | 6   | 50   |
 |   35 |   7 | GPIO1_A3 |  OUT | 1 | 13 || 14 |   |      | GND      |     |      |
 |   92 |   8 | GPIO2_D4 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO1_C6 | 9   | 54   |
 |      |     |     3.3V |      |   | 17 || 18 | 0 | IN   | GPIO1_C7 | 10  | 55   |
```

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. 22. 2. 26pin SPI test

1) It can be seen from the schematic diagram of 26pin that the spi available for the development board is spi1



2) First check whether there is a **spidev1.0** device node in the Linux system. If it exists, it means that SPI1 has been set and can be used directly

```
root@orangepi4-lts:~# ls /dev/spi*
/dev/spidev1.0
```

3) Compile the spidev_test test program in the examples of wiringOP

```
root@orangepi4-lts:~/wiringOP/examples# make spidev_test
[CC] spidev_test.c
[link]
```

4) Do not short the txd and rxd pins of SPI1 first, the output result of running spidev_test is as follows, you can see that the data of TX and RX are inconsistent

```
root@orangepi4-lts:~/wiringOP/examples# ./spidev_test -v -D /dev/spidev1.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF  | ................................
```

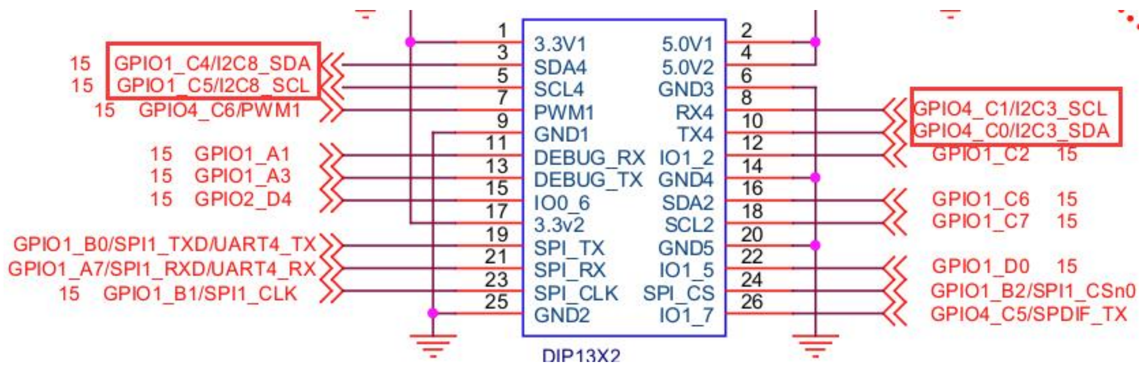
5) Then short-circuit the two pins of SPI1's txd (pin 19 in the 40pin interface) and rxd (pin 21 in the 40pin interface) and then run the output of spidev_test as follows, you can see the sent and received the same data

```
root@orangepi4-lts:~/wiringOP/examples# ./spidev_test -v -D /dev/spidev1.0
spi mode: 0x0
```

```
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    | ......@....▒..................▒.
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    | ......@....▒..................▒.
```

3. 22. 3. 26pin I2C test

1) It can be seen from the schematic diagram of 26pin that the available i2c for the development board are i2c3 and i2c8



2) After the system starts, you can see the following multiple i2c device nodes under **/dev**

a. i2c3 in 26pin corresponds to /dev/i2c-3

b. i2c8 in 26pin corresponds to /dev/i2c-8

```
root@orangepi4-lts:~# ls /dev/i2c*
/dev/i2c-0    /dev/i2c-1    /dev/i2c-2    /dev/i2c-3    /dev/i2c-4    /dev/i2c-7    /dev/i2c-8
/dev/i2c-9
```

3) Then start testing i2c, first install i2c-tools

```
root@orangepi4-lts:~# apt-get update
root@orangepi4-lts:~# apt-get -y install i2c-tools
```

4) Then connect an i2c device to the i2c8 pin of the 26pin connector (i2c3 test is the same as i2c8, just connect the device to the i2c3 pin, the following takes i2c8 as an example)

	i2c8	i2c3

Sda pin	Corresponds to pin 3	Corresponds to pin 10
Sck pin	Corresponds to pin 5	Corresponds to pin 8
Vcc pin	Corresponds to pin 1	Corresponds to pin 1
Gnd pin	Corresponds to pin 6	Corresponds to pin 6

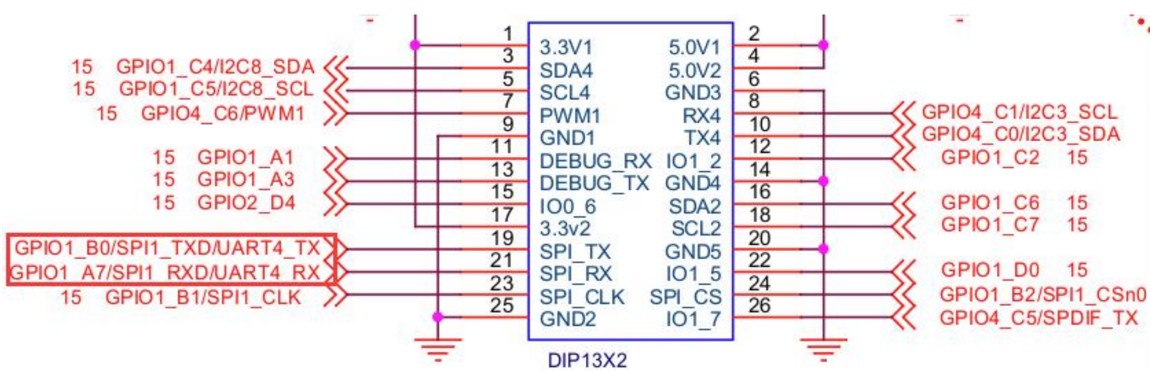
5) Then use the **i2cdetect -y 8** command. If the address of the connected i2c device can be detected, it means that the i2c can be used normally

```
root@orangepi4-lts:~# i2cdetect -y 8
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- 38 -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

3. 22. 4. 26pin UART test

Note that SPI1 and UART4 of Orgne Pi 4 LTS multiplex the same pins, so when using UART4, SPI1 cannot be used, and the configuration of SPI1 needs to be turned off.

1) It can be seen from the schematic diagram of 26pin that the uart available for the development board is uart4



2) The SPI1 and UART4 of the Oragne Pi 4 LTS multiplex the same pins. In the dts, the UART4 is turned off by default, and the SPI1 is turned on. If you want to use the function of UART4, you can use DT overlay to open the configuration of uart4 and turn off the configuration of SPI1 in the Linux system

a. For Linux4.4 system, the method is as follows

a) A script named **orangepi-add-overlay** is pre-installed in the linux system. Through this script, we can use DT overlay to dynamically modify the configuration in dts. First write the rockchip-uart4.dts file, the content is as follows

```
root@orangepi4-lts:~# vim rockchip-uart4.dts
/dts-v1/;
/plugin/;

/ {
        compatible = "rockchip,rk3399";

        fragment@0 {
                target = <&spi1>;
                __overlay__ {
                        status = "disabled";
                };
        };

        fragment@1 {
                target = <&uart4>;
                __overlay__ {
                        status = "okay";
                };
        };
};
```

b) Then you can use **orangepi-add-overlay** to compile rockchip-uart4.dts into rockchip-uart4.dtbo

```
root@orangepi:~# orangepi-add-overlay rockchip-uart4.dts
Compiling the overlay
Copying the compiled overlay file to /boot/overlay-user/
Reboot is required to apply the changes
```

c) Then restart the Linux system, you can see the following printing information in the log output from the serial port, indicating that rockchip-uart4.dtbo is loaded successfully

```
384 bytes read in 6 ms (62.5 KiB/s)
Applying user provided DT overlay rockchip-uart4.dtbo
```

d) After entering the Linux system, you can see the device node ttyS4 under /dev

```
root@orangepi4-lts:~# ls /dev/ttyS*
/dev/ttyS0    /dev/ttyS4
```

b. For Linux5.1x system, the method is as follows

a) First set the overlays variable in **/boot/orangepiEnv.txt** to open the uart4 configuration

```
root@orangepi4-lts:~# vim /boot/orangepiEnv.txt
overlays=uart4
```

b) Then restart the system. When starting, you can see the following print information in the startup log of u-boot, indicating that the configuration of uart4 is loaded successfully

```
Applying kernel provided DT overlay rockchip-uart4.dtbo
2698 bytes read in 8 ms (329.1 KiB/s)
```

c) After entering the Linux system, you can see the device node ttyS4 under **/dev**

```
root@orangepi4-lts:~# ls /dev/ttyS*
/dev/ttyS2    /dev/ttyS4
```

3) Then start to test the uart interface, first use the DuPont line to short-circuit the rx and tx of the uart4 interface to be tested

	uart4
tx Pin	Corresponds to pin 19
rx Pin	Corresponds to pin 21

4) Use the gpio command in wiringOP 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

```
root@orangepi4-lts:~# gpio serial /dev/ttyS4

Out:    0:   ->     0
Out:    1:   ->     1
Out:    2:   ->     2
```

```
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5
Out:    6:   ->    6
Out:    7:   ->    7
Out:    8:   ->    8
Out:    9:   ->    9^C
```

5) You can also use the serialTest.c program in wiringOP to test the loopback function of the serial port. The specific steps are as follows:

a. First modify the name of the serial device node opened by the serial test program serialTest in wiringOP to **/dev/ttyS4**

```
root@orangepi:~/wiringOP/examples# vim serialTest.c
```

```
int main ()
{
  int fd ;
  int count ;
  unsigned int nextTime ;

  if ((fd = serialOpen ("/dev/ttyS4", 115200)) < 0)
  {
    fprintf (stderr, "Unable to open serial device: %s\n", strerror (errno)) ;
    return 1 ;
  }
```

b. Then compile the serial test program serialTest in wiringOP

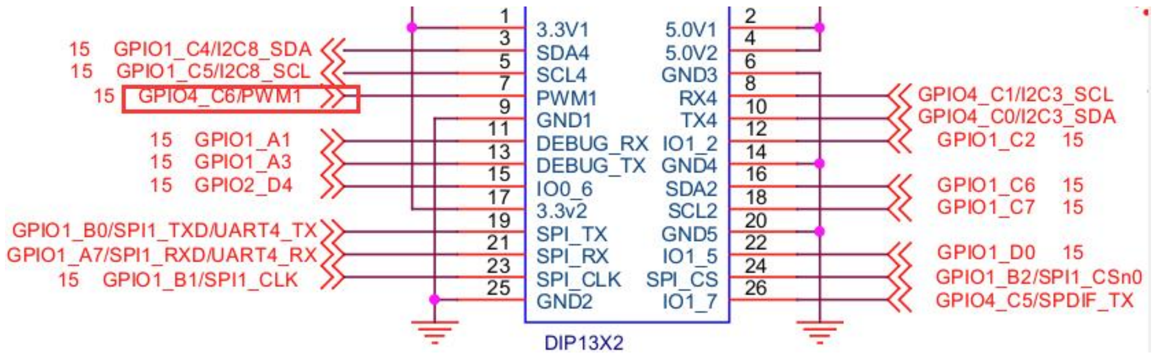
```
root@orangepi:~/wiringOP/examples# make serialTest
[CC] serialTest.c
[link]
root@orangepi:~/wiringOP/examples#
```

c. Finally run serialTest, if you can see the following print, it means the serial communication is normal

```
root@orangepi:~/wiringOP/examples# ./serialTest

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

3. 22. 5. 26pin PWM test

1) The 7th pin of 26pin is PWM1, the official image has turned on PWM1 by default, and PWM1 can be used without other configuration



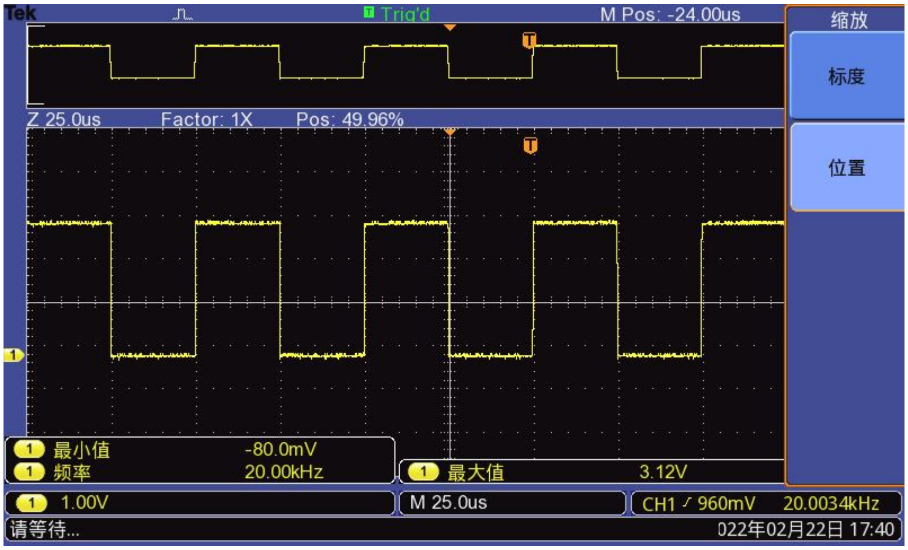
2) After the pwm driver is loaded successfully, the pwmchip1 directory will be generated under **/sys/class/pwm/**, write 0 to the export file, the pwm timer will be opened, and a pwm0 directory will be generated. On the contrary, writing 0 to the unexport file will turn off the pwm timer, and the pwm0 directory will be deleted. This directory has the following files:

file name	effect
enable	**Write 1 to enable pwm, write 0 to disable pwm**
polarity	**There are two parameter options, normal and inversed, indicating that the output pin level is inverted.**
duty_cycle	**The unit is nanoseconds. In normal mode, it means the duration of high level. In inversed mode, it means the duration of low level.**
period	**The unit is nanoseconds, indicating the duration of the pwm wave**

3) Example of use: let pwm1 output a square wave with a duty cycle of 50% and a period of 50 microseconds

```
root@orangepi:~# cd /sys/class/pwm/pwmchip1
root@orangepi:/sys/class/pwm/pwmchip1# echo 0 > export
root@orangepi:/sys/class/pwm/pwmchip1# echo 50000 > pwm0/period
root@orangepi:/sys/class/pwm/pwmchip1# echo 25000 > pwm0/duty_cycle
root@orangepi:/sys/class/pwm/pwmchip1# echo 1 > pwm0/enable
```

4) On the oscilloscope, you can see that pwm1 outputs the following waveforms



3. 23. How to install and use wiringOP-Python

wiringOP-Python is a library of wiringOP's Python language version, which is used to operate hardware resources such as GPIO, I2C, SPI, and UART of the development board in Python programs.

3. 23. 1. Installation method of wiringOP-Python

1) First install the dependency package

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the source code of wiringOP-Python

Note that the following git clone --recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on wiringOP. Please make sure that the download process does not report errors due to network problems.

```
root@orangepi:~# git clone \
--recursive https://github.com/orangepi-xunlong/wiringOP-Python

Cloning into 'wiringOP-Python'...
remote: Enumerating objects: 581, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (14/14), done.
```

```
remote: Total 581 (delta 17), reused 22 (delta 15), pack-reused 552
Receiving objects: 100% (581/581), 303.79 KiB | 1.78 MiB/s, done.
Resolving deltas: 100% (339/339), done.
Submodule 'wiringOP' (https://github.com/orangepi-xunlong/wiringOP.git) registered for
path 'wiringOP'
Cloning into '/home/orangepi/wiringOP-Python/wiringOP'...
remote: Enumerating objects: 626, done.
remote: Counting objects: 100% (155/155), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 626 (delta 129), reused 136 (delta 116), pack-reused 471
Receiving objects: 100% (626/626), 365.55 KiB | 597.00 KiB/s, done.
Resolving deltas: 100% (409/409), done.
Submodule path 'wiringOP': checked out
'0a7284942375ff68a9940c44234b6d6ec5f7aa59'
```

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 the help information is output, it means that wiringOP-Python is installed successfully. Press the **q** key to exit the help information interface.

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

NAME
    wiringpi

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

```
    # the SWIG interface file instead.
```

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

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

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

b. Then import the python module of wiringpi

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

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

```
>>> help(wiringpi)
Help on module wiringpi:

NAME
    wiringpi

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

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

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


>>>
```

3. 23. 2. 26pin GPIO port test

Like wiringOP, wiringOP-Python can also determine which GPIO pin to operate by specifying the wPi number. Because there is no command to view the wPi number in wiringOP-Python, you can only check the board wPi number and physical properties through the gpio command in wiringOP. Correspondence of pins.

```
root@orangepi4-lts:~# gpio readall
 +------+-----+----------+------+---+-OPi 4 LTS +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3V |      |   |  1 || 2  |   |      | 5V       |     |      |
 |   64 |   0 | I2C8_SDA | ALT2 | 1 |  3 || 4  |   |      | 5V       |     |      |
 |   65 |   1 | I2C8_SCL | ALT2 | 1 |  5 || 6  |   |      | GND      |     |      |
 |  150 |   2 |     PWM1 |   IN | 0 |  7 || 8  | 1 | ALT2 | I2C3_SCL | 3   | 145  |
 |      |     |      GND |      |   |  9 || 10 | 1 | ALT2 | I2C3_SDA | 4   | 144  |
 |   33 |   5 | GPIO1_A1 |   IN | 0 | 11 || 12 | 1 | IN   | GPIO1_C2 | 6   | 50   |
 |   35 |   7 | GPIO1_A3 |  OUT | 1 | 13 || 14 |   |      | GND      |     |      |
 |   92 |   8 | GPIO2_D4 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO1_C6 | 9   | 54   |
 |      |     |     3.3V |      |   | 17 || 18 | 0 | IN   | GPIO1_C7 | 10  | 55   |
 |   40 |  11 | SPI1_TXD | ALT3 | 0 | 19 || 20 |   |      | GND      |     |      |
 |   39 |  12 | SPI1_RXD | ALT3 | 1 | 21 || 22 | 0 | IN   | GPIO1_D0 | 13  | 56   |
 |   41 |  14 | SPI1_CLK | ALT3 | 1 | 23 || 24 | 1 | ALT3 | SPI1_CS  | 15  | 42   |
 |      |     |      GND |      |   | 25 || 26 | 0 | IN   | GPIO4_C5 | 16  | 149  |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+-OPi 4 LTS +---+---+--+----------+-----+------+
```

1) The following uses pin No. 11—the corresponding GPIO is GPIO1_A1—the corresponding wPi serial number is 5—as an example to demonstrate how to set the high and low levels of the GPIO port

```
root@orangepi4-lts:~# gpio readall
 +------+-----+----------+------+---+-OPi 4 LTS +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3V |      |   |  1 || 2  |   |      | 5V       |     |      |
 |   64 |   0 | I2C8_SDA | ALT2 | 1 |  3 || 4  |   |      | 5V       |     |      |
 |   65 |   1 | I2C8_SCL | ALT2 | 1 |  5 || 6  |   |      | GND      |     |      |
 |  150 |   2 |     PWM1 |   IN | 0 |  7 || 8  | 1 | ALT2 | I2C3_SCL | 3   | 145  |
 |      |     |      GND |      |   |  9 || 10 | 1 | ALT2 | I2C3_SDA | 4   | 144  |
 |   33 |   5 | GPIO1_A1 |  OUT | 0 | 11 || 12 | 1 | IN   | GPIO1_C2 | 6   | 50   |
 |   35 |   7 | GPIO1_A3 |  OUT | 1 | 13 || 14 |   |      | GND      |     |      |
 |   92 |   8 | GPIO2_D4 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO1_C6 | 9   | 54   |
 |      |     |     3.3V |      |   | 17 || 18 | 0 | IN   | GPIO1_C7 | 10  | 55   |
```

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

a. First set the GPIO port as 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(5, GPIO.OUTPUT) ;  "
```

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

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(5, GPIO.LOW)"
```

c. Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is successfully set.

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(5, GPIO.HIGH)"
```

3) The steps to test in the command line of python3 are as follows:

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

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

b. Then import the python module of wiringpi

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

c. Then set the GPIO port as 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(5, 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 successfully set.

```
>>> wiringpi.digitalWrite(5, 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 successfully set.

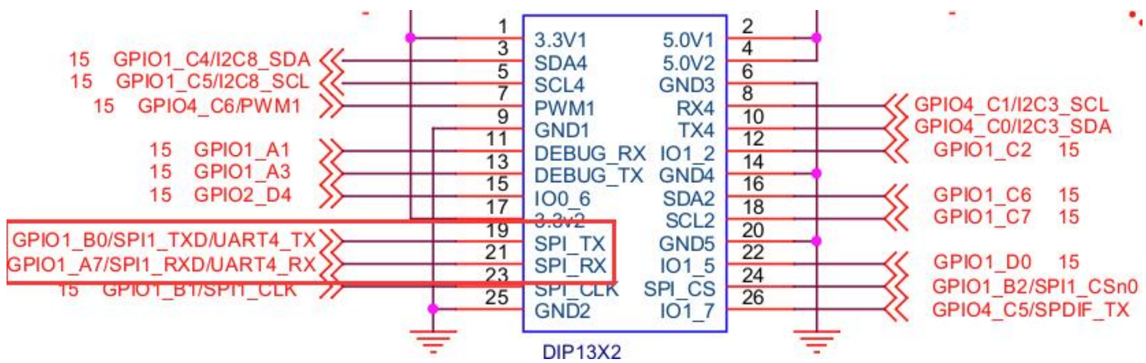
```
>>> wiringpi.digitalWrite(5, GPIO.HIGH)
```

4) The method of wiringOP-Python to set the GPIO high and low levels in the 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. 23. 3. 26pin SPI test

1) It can be seen from the schematic diagram of 26pin that the spi available for the development board is spi1



2) First check whether there is a device node of **spidev1.0** in the Linux system. If it exists, it means that SPI1 has been set and can be used directly

```
root@orangepi4-lts:~# ls /dev/spi*
/dev/spidev1.0
```

3) Then you can use the **spidev_test.py** program in the examples to test the loopback function of the SPI. The **spidev_test.py** program needs to specify the following two parameters:

a. **--channel**: Specify the channel number of the SPI

b. **--port**: Specify the port number of the SPI

4) Do not short the two pins of SPI1, mosi and miso. The output result of running spidev_test.py is as follows. It can be seen that the data of TX and RX are inconsistent

```
root@orangepi:~/wiringOP-Python# cd examples
```

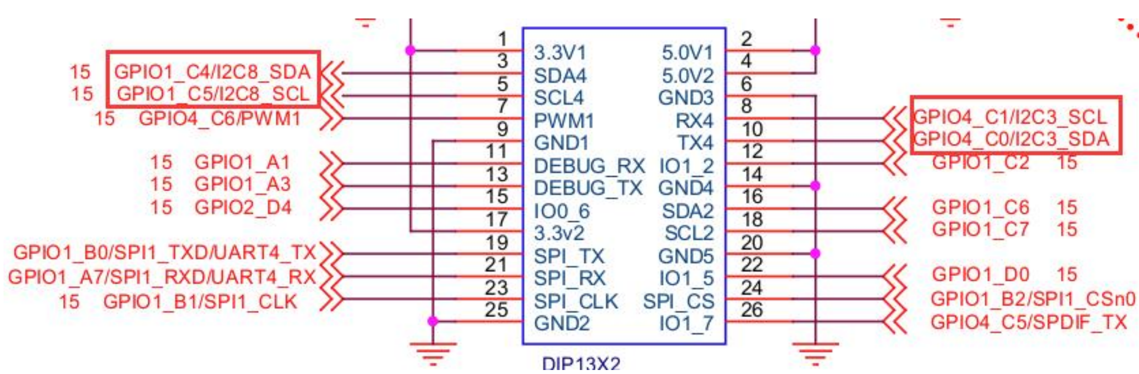
```
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 1 --port 0
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev1.0
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@..........|
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF    |................|
```

5) Then use the DuPont line to short-circuit the two pins of SPI1's txd (pin 19 in the 26pin interface) and rxd (pin 21 in the 26pin interface) and then run the output of spidev_test.py as follows, you can see The data sent and received are the same, indicating that the SPI1 loopback test is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 1 --port 0
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev1.0
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@..........|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@..........|
```

3. 23. 4. 26pin I2C test

1) It can be seen from the schematic diagram of 26pin that the available i2c for the development board are i2c3 and i2c8



2) After the system starts, you can see the following multiple i2c device nodes under **/dev**

a. i2c3 in 26pin corresponds to /dev/i2c-3

b. i2c8 in 26pin corresponds to /dev/i2c-8

```
root@orangepi4-lts:~# ls /dev/i2c*
/dev/i2c-0  /dev/i2c-1  /dev/i2c-2  /dev/i2c-3  /dev/i2c-4  /dev/i2c-7  /dev/i2c-8
/dev/i2c-9
```

3) Then start testing i2c, first install i2c-tools

```
root@orangepi4-lts:~# sudo apt update
root@orangepi4-lts:~# sudo apt -y install i2c-tools
```

4) Then connect an i2c device to the i2c8 pin of the 26pin connector, here is an example of the ds1307 RTC module

RTC module pins	The corresponding pin of the development board 26pin
5V	pin 2
GND	pin 6
SDA	pin 3
SCL	pin 5

5) Then use the **i2cdetect -y 8** command if the address of the connected i2c device can be detected, it means that the i2c device is connected correctly

```
root@orangepi4-lts:~# i2cdetect -y 8
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

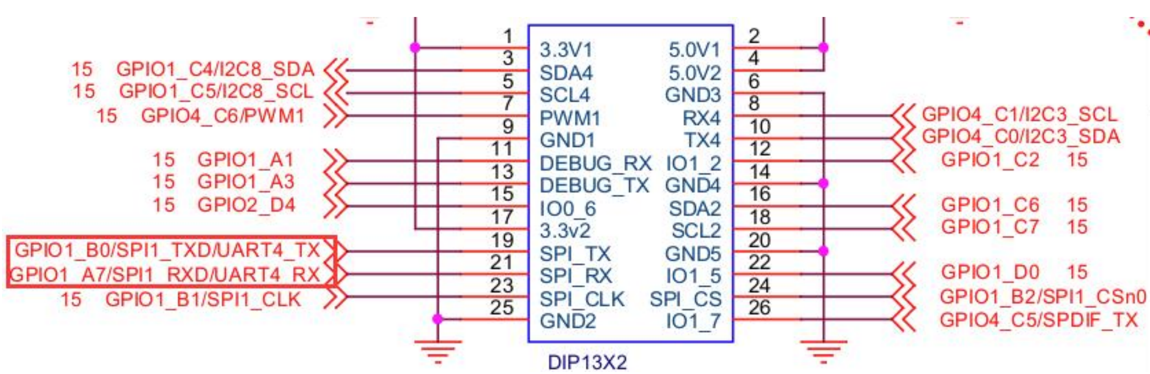
6) Then you can run the **ds1307.py** test program in the **examples** to read the time of the RTC

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device \
"/dev/i2c-8"
Thu 2022-06-16   04:35:46
Thu 2022-06-16   04:35:47
Thu 2022-06-16   04:35:48
^C
exit
```

3. 23. 5. 26pin UART test

Note that SPI1 and UART4 of Orgne Pi 4 LTS multiplex the same pins, so when using UART4, SPI1 cannot be used, and the configuration of SPI1 needs to be turned off.

1) It can be seen from the schematic diagram of 26pin that the uart available for the development board is uart4



2) The SPI1 and UART4 of the Oragne Pi 4 LTS multiplex the same pins. In the dts, the

UART4 is turned off by default, and the SPI1 is turned on. If you want to use the function of UART4, you can use DT overlay to open the configuration of uart4 and turn off the configuration of SPI1 in the Linux system

a. For Linux4.4 system, the method is as follows

a) A script named **orangepi-add-overlay** is pre-installed in the linux system. Through this script, we can use DT overlay to dynamically modify the configuration in dts. First write the rockchip-uart4.dts file, the content is as follows

```
root@orangepi4-lts:~# vim rockchip-uart4.dts
/dts-v1/;
/plugin/;

/ {
        compatible = "rockchip,rk3399";

        fragment@0 {
                target = <&spi1>;
                __overlay__ {
                        status = "disabled";
                };
        };

        fragment@1 {
                target = <&uart4>;
                __overlay__ {
                        status = "okay";
                };
        };
};
```

b) Then you can use **orangepi-add-overlay** to compile rockchip-uart4.dts into rockchip-uart4.dtbo

```
root@orangepi:~# orangepi-add-overlay rockchip-uart4.dts
Compiling the overlay
Copying the compiled overlay file to /boot/overlay-user/
Reboot is required to apply the changes
```

c) Then restart the Linux system, you can see the following printing information in the log output from the serial port, indicating that rockchip-uart4.dtbo is loaded successfully

```
384 bytes read in 6 ms (62.5 KiB/s)
Applying user provided DT overlay rockchip-uart4.dtbo
```

d) After entering the Linux system, you can see the device node ttyS4 under /dev

```
root@orangepi4-lts:~# ls /dev/ttyS*
/dev/ttyS0    /dev/ttyS4
```

b. For Linux5.10 system, the method is as follows

a) First set the overlays variable in **/boot/orangepiEnv.txt** to open the uart4 configuration

```
root@orangepi4-lts:~# cat /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
overlay_prefix=rockchip
fdtfile=rockchip/rk3399-orangepi-4-lts.dtb
rootdev=UUID=c51e6614-42cf-473c-9134-46a72667eb9c
rootfstype=ext4
overlays=uart4
```

b) Then restart the system. When starting, you can see the following print information in the startup log of u-boot, indicating that the configuration of uart4 is loaded successfully

```
Applying kernel provided DT overlay rockchip-uart4.dtbo
2698 bytes read in 8 ms (329.1 KiB/s)
```

c) After entering the Linux system, you can see the device node ttyS4 under **/dev**

```
root@orangepi4-lts:~# ls /dev/ttyS*
/dev/ttyS2    /dev/ttyS4
```

3) Then start to test the uart interface, first use the DuPont line to short-circuit the rx and tx of the uart4 interface to be tested

	Uart4
tx pin	Corresponds to pin 19
rx pin	Corresponds to pin 21

4) Finally, you can run the **serialTest.py** program in the examples to test the loopback function of the serial port. If you can see the following print, it means that the serial port loopback test is normal

```
root@orangepi:~/wiringOP-Python# cd examples
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. 24. How to use 0.96-inch OLED module with I2C interface

1) The 0.96-inch OLED module of Orange Pi is shown in the figure below, and its 7-bit i2c slave address is 0x3c

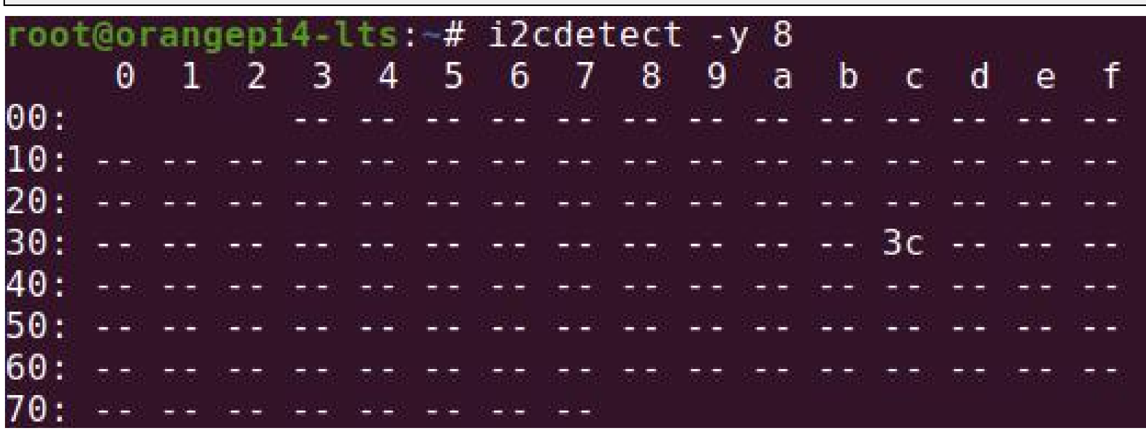
2) First, connect the 0.96-inch OLED module to the 26pin interface of the Orange Pi development board through the DuPont cable. The wiring method is as follows (the following takes i2c8 as an example, i2c3 only needs to change the scl to pin 8, and the sda to the 8th pin. to pin 10)

OLED module pins	describe	Development board 26pin interface i2c8	Development board 26pin interface i2c3

		pin	pin
GND	power ground	pin 6	pin 6
VCC	5V	pin 4	pin 4
SCL	I2C clock line	pin 5	pin 8
SDA	I2C data line	pin 3	pin 10
RST	Connect to 3.3V	pin 1	pin 1
DC	Connect to GND	pin 9	pin 9
CS	Connect to GND	pin 25	pin 25

3) After connecting the OLED module to the development board, first use the i2c-tools tool to check whether the address of the OLED module can be scanned

root@orangepi4-lts:~# **apt update**
root@orangepi4-lts:~# **apt install i2c-tools**
root@orangepi4-lts:~# **i2cdetect -y 8**



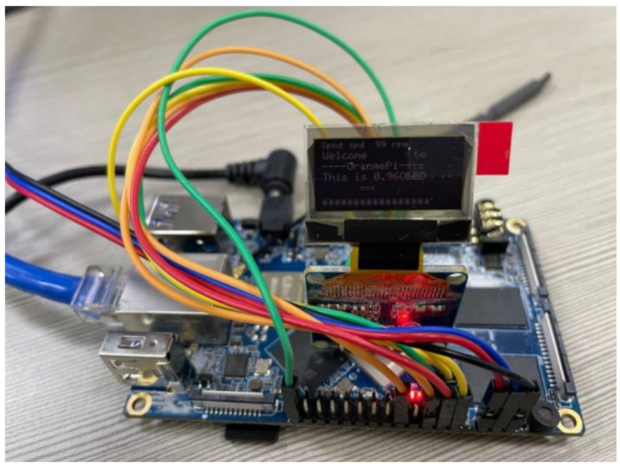
```
root@orangepi4-lts:~# i2cdetect -y 8
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- 3c -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

4) Then you can use the oled_demo in wiringOP to test the OLED module. The test steps

are as follows

```
root@orangepi4-lts:~# git clone https://github.com/orangepi-xunlong/wiringOP
root@orangepi4-lts:~# cd wiringOP
root@orangepi4-lts:~/wiringOP# ./build clean && ./build
root@orangepi4-lts:~/wiringOP# cd examples
root@orangepi4-lts:~/wiringOP/examples# make oled_demo
root@orangepi4-lts:~/wiringOP/examples# ./oled_demo /dev/i2c-8
---------start--------
----------end---------
```

5) After running oled_demo, you can see the following output on the OLED screen



3. 25. Hardware watchdog test

1) Download the code of wiringOP

```
root@orangepi4-lts:~# apt-get update
root@orangepi4-lts:~# apt-get -y install git
root@orangepi4-lts:~# git clone https://github.com/orangepi-xunlong/wiringOP
```

2) Compile the watchdog test program

```
root@orangepi4-lts:~# cd wiringOP/examples/
root@orangepi4-lts:~/wiringOP/examples# gcc watchdog.c -o watchdog
```

3) Run the watchdog test program

a. The second parameter 10 represents the counting time of the watchdog. If the dog is not fed within this time, the system will restart

b. We can feed the dog by pressing any key on the keyboard (except ESC), after feeding the dog, the program will print a line of keep alive to indicate that the dog was fed successfully

```
root@orangepi4-lts:~/wiringOP/examples# ./watchdog 10
open success
options is 33152,identity is Synopsys DesignWare Watchdog
put_usr return,if 0,success:0
The old reset time is: 21
return ENOTTY,if -1,success:-1
return ENOTTY,if -1,success:-1
put_user return,if 0,success:0
put_usr return,if 0,success:0
keep alive
keep alive
```

3. 26. How to check the serial number of rk3399 chip

1) The command to check the serial number of RK3399 and RK3399-T chips is as follows, the serial number of each chip is different and unique, so the serial number can be used to distinguish multiple development boards

```
orangepi@orangepi:~$ cat /proc/cpuinfo | grep "Serial"
Serial          : 6311556468e5db96
```

3. 27. How to program linux image to eMMC

Note that only the Orange Pi 4 LTS development board with the eMMC chip model can burn the image into the eMMC. If you purchase the Orange Pi 4 LTS development board without the eMMC chip, you cannot burn the image into the eMMC.

Note that the development board can be started through TF card or eMMC. The priority of TF card is higher than that of eMMC. That is to say, if a TF card is inserted into the development board, and there is a system in the TF card, the

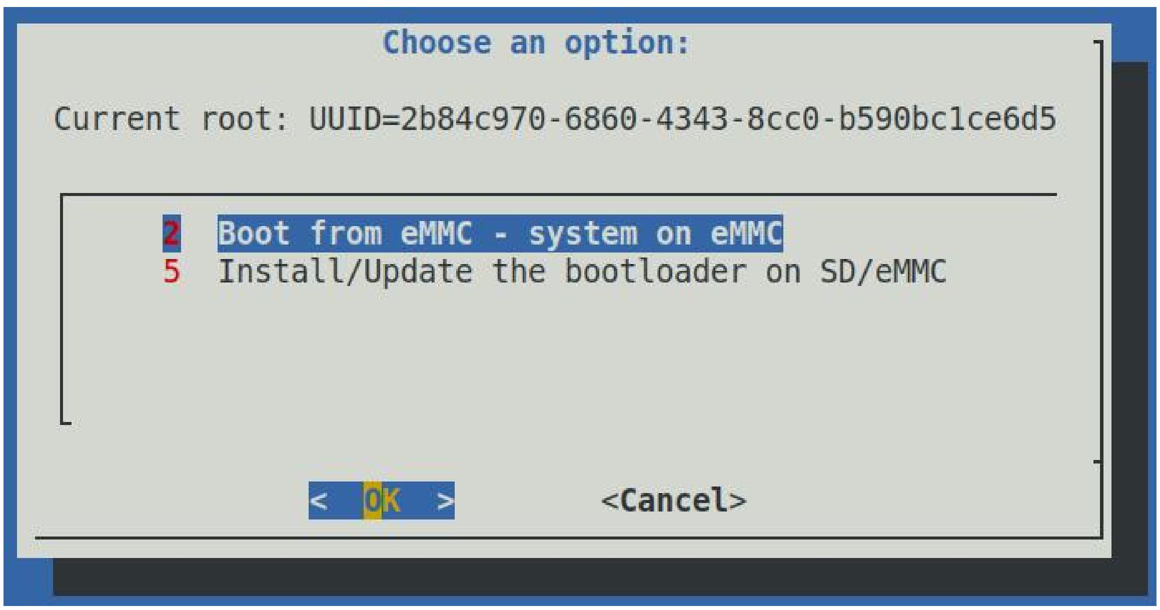
system in the TF card will be activated by default, but the system in eMMC will not be activated.

1) Burning the linux image to eMMC needs to be done with the help of a TF card, first burn the linux image to the TF card, and then start the development board to enter the linux system

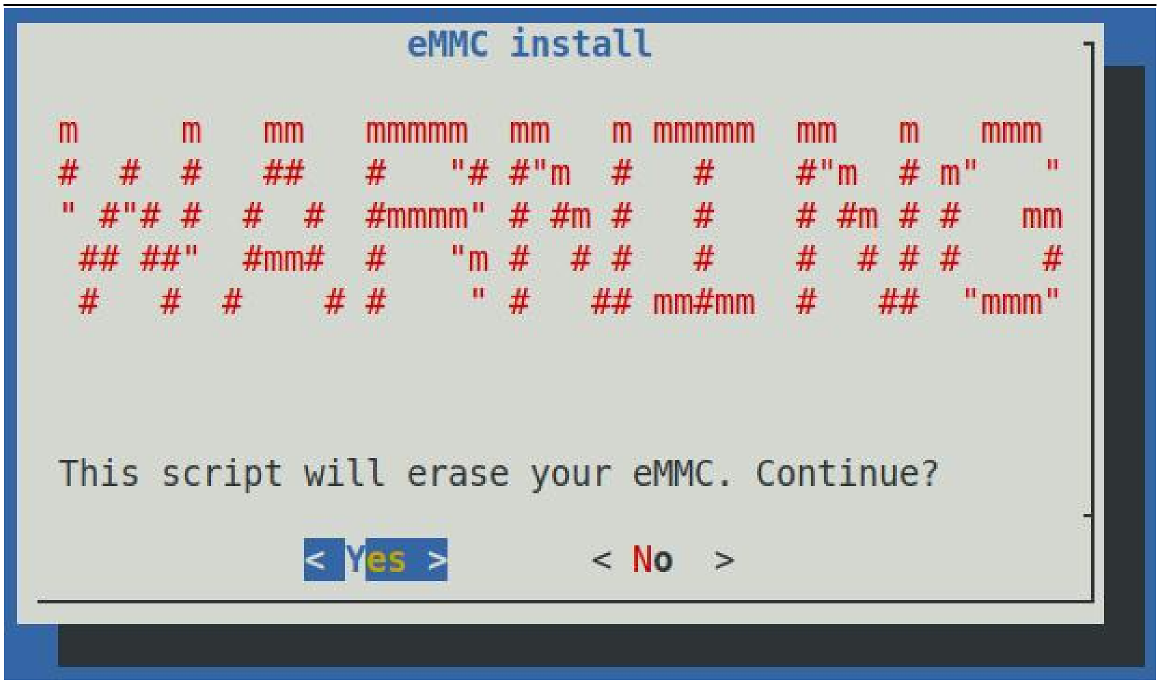
2) Then run the **nand-sata-install** script, **remember to add sudo permissions**

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

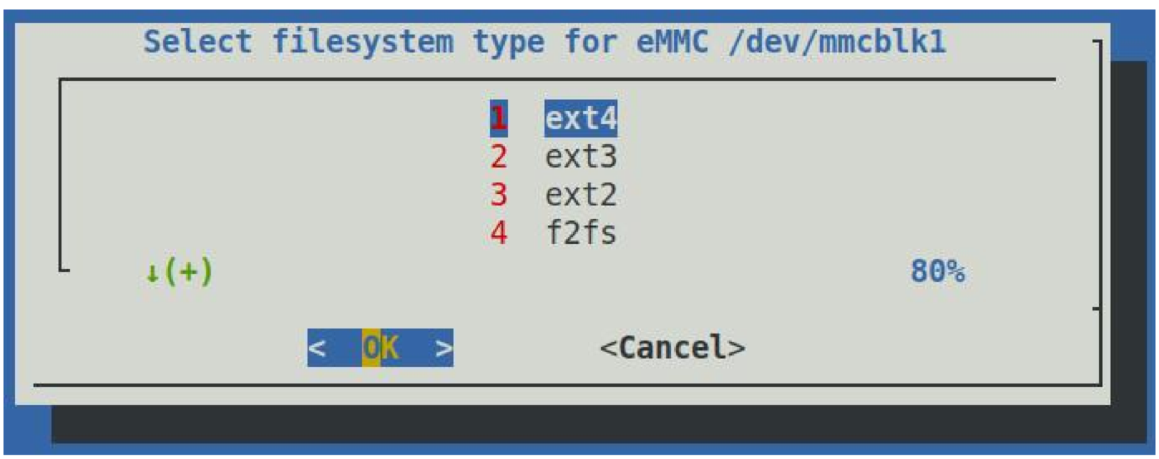
3) Then select **2 Boot from eMMC - system on eMMC**

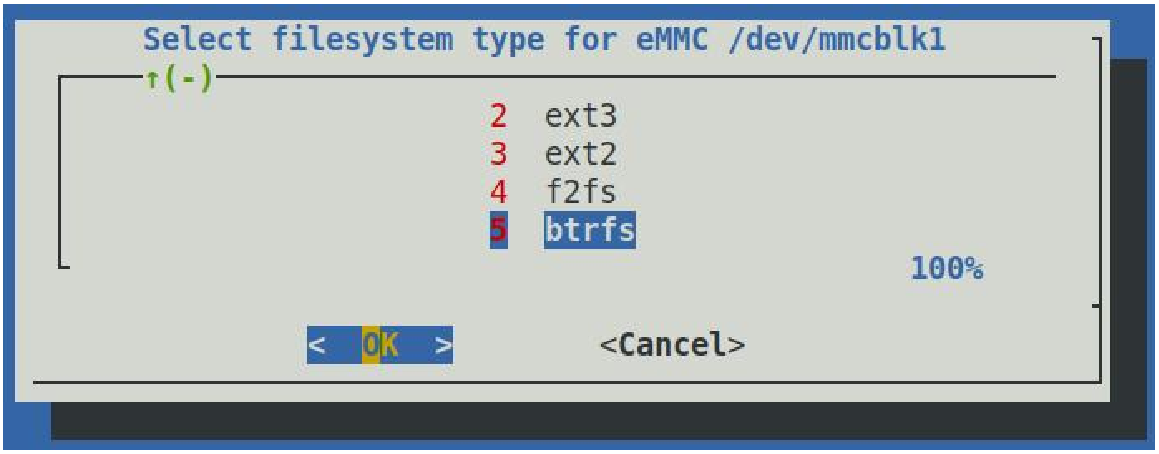


4) Then a warning will pop up, the script will erase all data on the eMMC, select **<Yes>** to continue

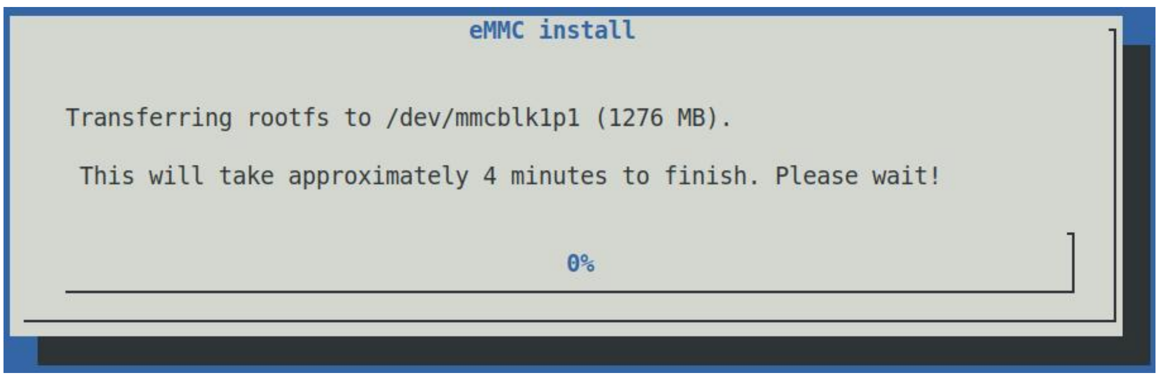


5) Then you will be prompted to select the type of file system, which supports ext2/3/4, f2fs and btrfs five file systems

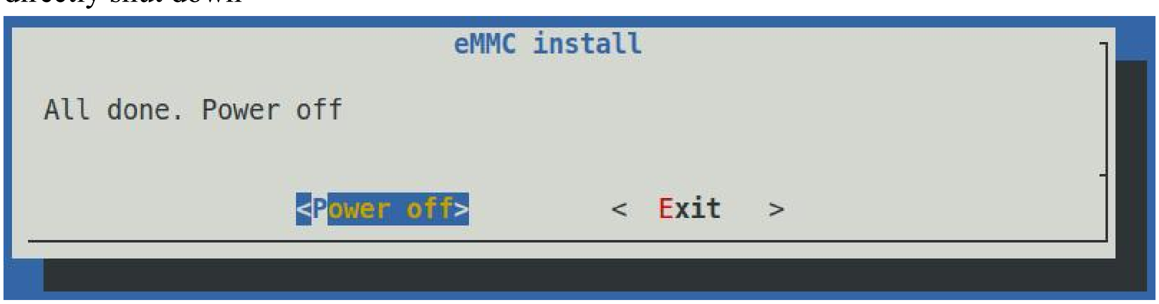




6) Then it will start to format eMMC. After formatting eMMC, it will start to burn the linux image into eMMC



7) After burning, the following options will be prompted, you can select **<Power off>** to directly shut down



8) Then pull out the TF card, and then power on again, it will start the linux system in eMMC

If there is a problem with startup after burning, you can use the following

command to clear the eMMC first, and then try burning again:

1. First insert the TF card, then start the system, and then use the ls /dev/mmcblk* command to view all the mmc device nodes in the system, eMMC generally has two boot partitions, there is no TF card, through /dev/mmcblk1boot1 You can know that the block device of eMMC is /dev/mmcblk1 (not all systems must be mmcblk1, please refer to what you actually see).

```
orangepi@orangepi:~$ ls /dev/mmcblk*
/dev/mmcblk1    /dev/mmcblk1boot0    /dev/mmcblk1boot1    /dev/mmcblk1p1
/dev/mmcblk1rpmb    /dev/mmcblk2    /dev/mmcblk2p1
```

2. Now that we know that the block device of eMMC is /dev/mmcblk1, we can use the following command to clear eMMC:

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

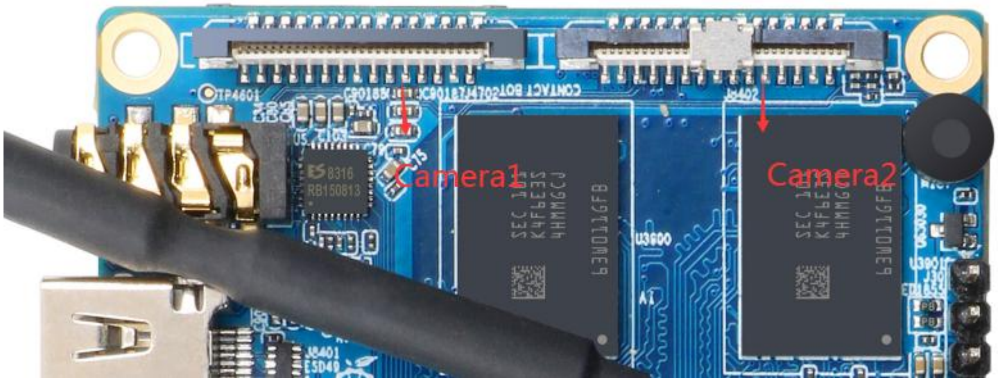
The above command of= specifies the device node of eMMC, and count=1000 will clear the first 1GB of eMMC space. If you think it is not enough, you can increase this value.

3. 28. How to use the OV13850 camera on Linux 4.4

Note that only the Debian10 Linux4.4 desktop version system supports this function, and other systems do not currently support it. So if you want to test the OV13850 camera, please burn the Debian10 Linux4.4 desktop system first, and then test it.

3. 28. 1. Camera connection instructions

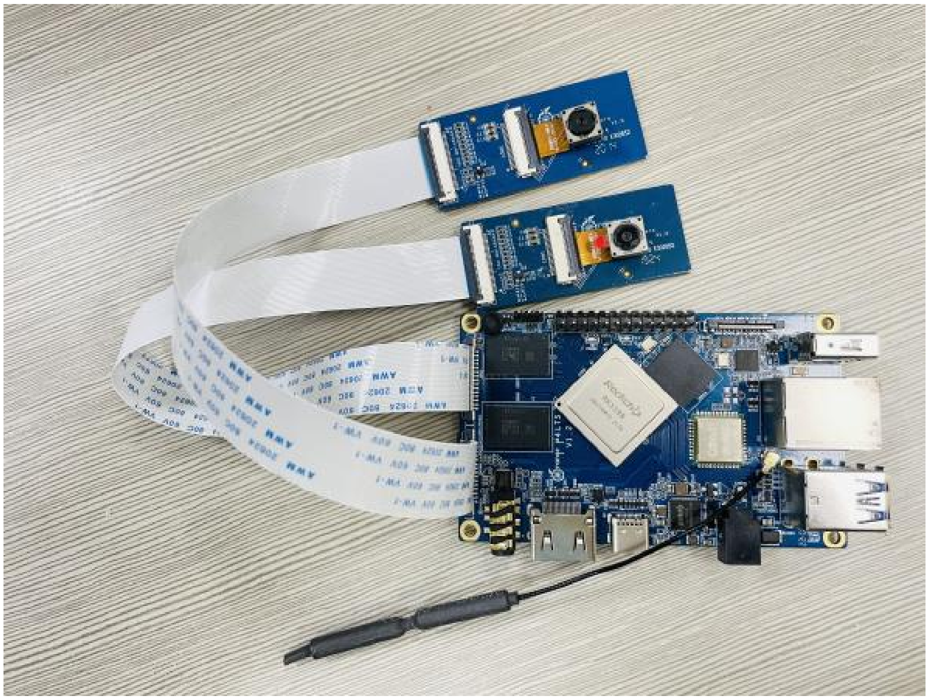
1) The Orange Pi 4 LTS has two Camera ports, both of which only support the OV13850 camera by default. These two Camera interfaces can use one of the interfaces alone, or they can use two Camera interfaces to connect two cameras at the same time. The positions of the Camera1 and Camera2 interfaces are shown in the following figure:



2) The OV13850 camera kit includes an OV13850 camera, an adapter board and a cable

3) First insert the OV13850 camera into the adapter board, and then insert the cable into another card slot of the adapter board

4) Then insert the other end of the cable into the Camera interface of the development board. The development board can be connected to two cameras at the same time, or it can be connected to a single camera. After connecting the camera, start the Linux system (please do not plug in the camera after powering on)



5) After starting the system, execute the following commands. If the following information appears, the camera is working normally. If there is no such information, please check whether the camera is connected correctly.

```
root@orangepi4-lts:~# dmesg | grep Async
[    1.623685] rkisp1: Async subdev notifier completed
```

3. 28. 2. How to open a single OV13850 camera

1) In the Debian system, the camera can be called through the pre-installed **test_camera-gst.sh** script. The parameter description of the **test_camera-gst.sh** script is as follows:

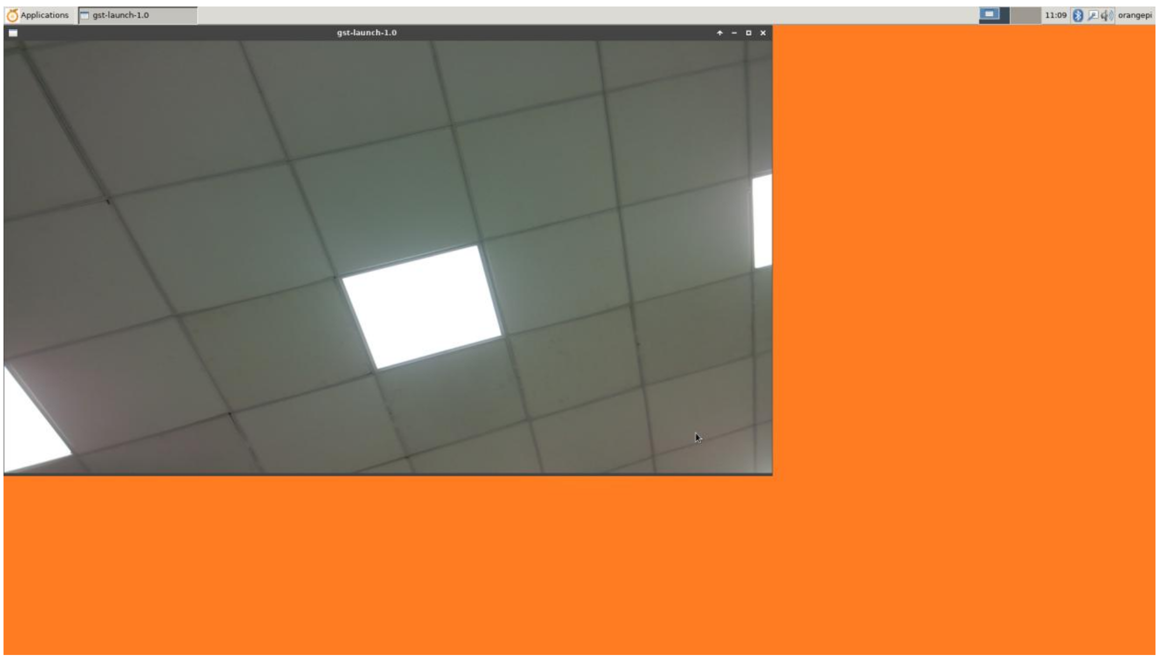
parameter	Function
--index or -i	Select the serial number of the camera to be used. The optional values are 0 and 1. When connecting two cameras at the same time, specify 1 to indicate that the second camera is to be operated.
--action or -a	Specifies the action to be executed by the command. The optional parameters are: preview, photo and video, corresponding to preview, photo and video respectively
--output or -o	Specify the output file name, which is used to specify the output file name when taking pictures and videos
--verbose or -v	When specified as yes, the full command line that invokes the

	gst-launch-1.0 command will be output

2) Here's how to view the camera preview image:
   a. First run the test_camera-gst.sh script

```
root@orangepi4-lts:~# test_camera-gst.sh
Setting pipeline to PAUSED ...
media get entity by name: lens is null
Pipeline is live and does not need PREROLL ...
```

   b. Then a live camera window will open on the desktop



   c. If no parameters are specified for this command, the default action is to preview, which will open the camera recognized by the system. If two OV13850 cameras are connected, camera1 (the camera interface near the headphone holder) will be opened first.

3) The command to use the camera to take pictures is as follows, and the file name of the taken photo is **test.jpg**

```
root@orangepi4-lts-lts:~# test_camera-gst.sh -a photo -o test.jpg
gst-launch-1.0: no process found
Setting pipeline to PAUSED ...
media get entity by name: lens is null
Pipeline is live and does not need PREROLL ...
```

```
root@orangepi4-lts:~# ls
test.jpg
```

4) The command to use the camera to record is as follows, after running the script, it will start recording a 17s video

```
root@orangepi4-lts-lts:~# test_camera-gst.sh --action video --output test.ts
gst-launch-1.0: no process found
Setting pipeline to PAUSED ...
mpi: mpp version: Without VCS info
mpp_rt: NOT found ion allocator
...
root@orangepi4-lts:~# ls
test.ts
```

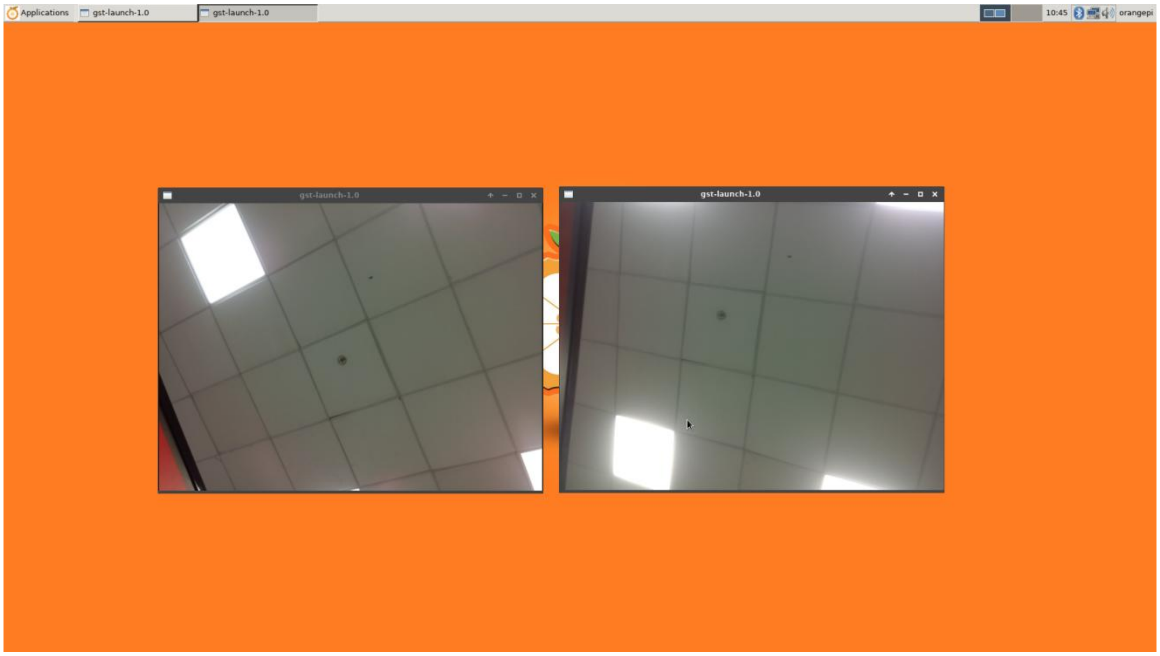
5) Then use a video player such as MPV to play the video you just recorded

3. 28. 3. How to open two OV13850 cameras at the same time

1) Run the **test_camera-dual.sh** script to open two OV13850 cameras

```
root@orangepi4-lts:~# test_camera-dual.sh
Start MIPI CSI Camera Preview!
Setting pipeline to PAUSED ...
Setting pipeline to PAUSED ...
```

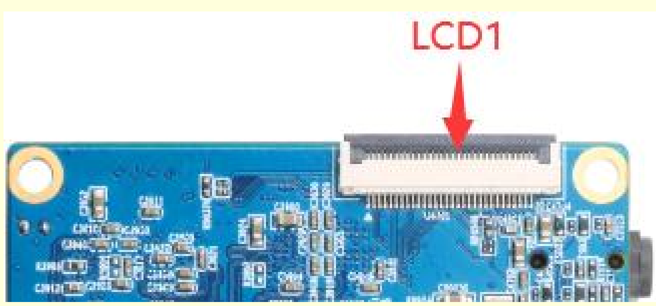
2) The effect is as shown in the figure below, the real-time windows of the two cameras will be opened on the desktop respectively

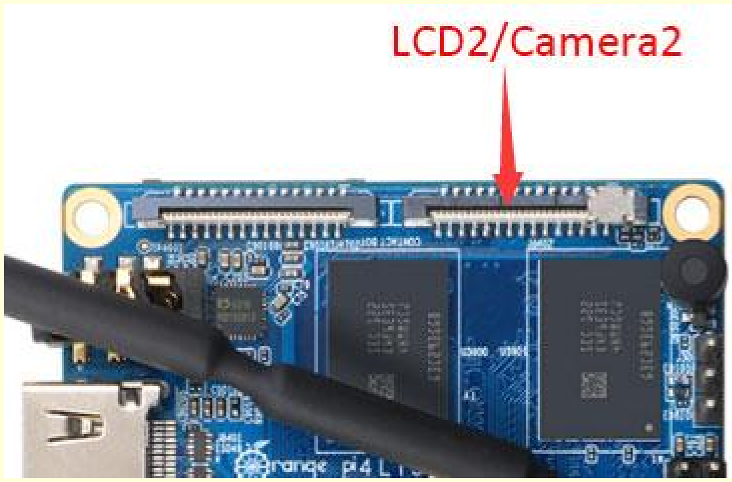


3. 29. 10.1 inch MIPI LCD screen usage

The default display output of the Linux image we provide supports HDMI and Type-C to HDMI, and does not support MIPI LCD screens. If you want to use the MIPI LCD screen, you first need to close the HDMI and Type-C to HDMI configuration, and open the MIPI LCD screen configuration, and then you can use the MIPI LCD screen.

Orange Pi 4 LTS has two MIPI LCD screen interfaces, the corresponding positions of LCD1 and LCD2 are shown in the following figure:





As can be seen from the above figure, the interface on the right supports both MIPI LCD screen and MIPI camera. This function is optional, so when using LCD2, you need to ensure that the configuration of Camera2 has been turned off.

The development board has two MIPI LCD interfaces, we can either use one of the MIPI LCD interfaces alone, or use two MIPI LCD interfaces for dual-screen display at the same time.

At present, the MIPI LCD interface supports two types of 10.1-inch MIPI screens. The model information can be seen on the back of the screen, as shown in the following figure:

The one on the left is the MIPI screen with model AFJ101BA2131 currently on sale, and the one on the right is the discontinued MIPI screen with model SL101PN27D1665

3. 29. 1. Instructions for LCD1 interface

1) LCD1 is turned off by default in the dts of the Linux system. First, you need to open the configuration of LCD1 in the dts.

a. For Linux 4.4 kernel, the method is as follows

a) A script named **orangepi-add-overlay** is pre-installed in the linux system. Through this script, we can use DT overlay to dynamically modify the configuration in dts. First write the rockchip-lcd1.dts file. If the LCD1 interface is connected to a MIPI screen with **model AFJ101BA2131**, the content of the file is as follows

```
root@orangepi4-lts:~# vim rockchip-lcd1.dts
/dts-v1/;
/plugin/;
```

```
/ {
        compatible = "rockchip,rk3399";

        fragment@0 {
                target = <&dsi>;
                __overlay__ {
                        status = "okay";
                };
        };

        fragment@1 {
                target = <&gt9xx>;
                __overlay__ {
                        status = "okay";
                };
        };

        fragment@2 {
                target = <&afj101_panel>;
                __overlay__ {
                        status = "okay";
                };
        };
};
```

b) If the LCD1 interface is connected to a MIPI screen with **model SL101PN27D1665**, the file content is as follows

```
root@orangepi4-lts:~# vim rockchip-lcd1.dts
/dts-v1/;
/plugin/;

/ {
        compatible = "rockchip,rk3399";

        fragment@0 {
                target = <&dsi>;
                __overlay__ {
```

```
                        status = "okay";
                };
        };

        fragment@1 {
                target = <&gt9xx>;
                __overlay__ {
                        status = "okay";
                };
        };

        fragment@2 {
                target = <&sl101_panel>;
                __overlay__ {
                        status = "okay";
                };
        };
};
```

c) Then you can use **orangepi-add-overlay** to compile rockchip-lcd1.dts into rockchip-lcd1.dtbo

```
root@orangepi:~# orangepi-add-overlay rockchip-lcd1.dts
Compiling the overlay
Copying the compiled overlay file to /boot/overlay-user/
Reboot is required to apply the changes
```

d) Then restart the Linux system, you can see the following printing information in the log output from the serial port, indicating that rockchip-lcd1.dtbo is loaded successfully

```
379 bytes read in 6 ms (61.5 KiB/s)
Applying user provided DT overlay rockchip-lcd1.dtbo
```

e) After entering the Linux system, enter the following command to see the relevant information of the kernel output

```
root@orangepi4-lts:~# dmesg |grep mipi
[       1.478148] dw-mipi-dsi ff960000.dsi: final DSI-Link bandwidth: 444 x 4 Mbps
```

b. For Linux 5.10 and Linux5.18 kernel, the method is as follows

a) First, set the overlays variable in /boot/orangepiEnv.txt to open the configuration of the LCD1 interface. If the LCD1 interface is connected to a MIPI screen with **model AFJ101BA2131**, the overlays variable is configured as follows

```
root@orangepi4-lts:~# vim /boot/orangepiEnv.txt
overlays=lcd1-afj101
```

b) If the LCD1 interface is connected to a MIPI screen with **model SL101PN27D1665**, the overlays variable is configured as follows

```
root@orangepi4-lts:~# vim /boot/orangepiEnv.txt
overlays=lcd1-sl101
```

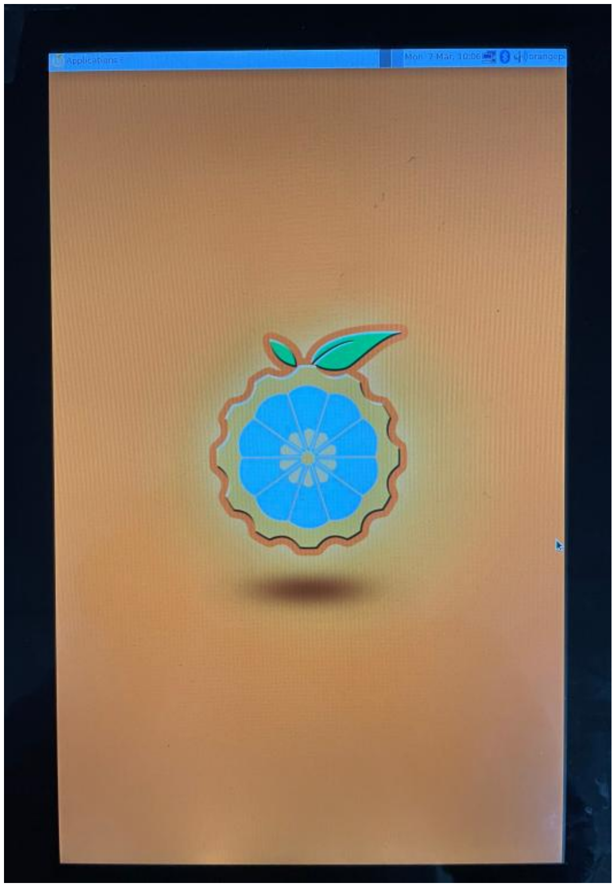
c) Then restart the system. When starting, if you can see the following printing information in the startup log of u-boot, it means that the configuration of lcd1 is loaded successfully

```
379 bytes read in 9 ms (41 KiB/s)
Applying kernel provided DT overlay rockchip-lcd1-afj101.dtbo
```

d) After entering the Linux system, enter the following command to see the relevant information of the kernel output

```
root@orangepi4-lts:~# dmesg |grep mipi
[       8.622148] dw-mipi-dsi-rockchip ff960000.dsi:
[drm:dw_mipi_dsi_bridge_mode_set] final DSI-Link bandwidth: 444 x 4 Mbps
```

2) Then disconnect the power supply of the development board, and then connect the MIPI screen by **referring to the usage of the 10.1-inch MIPI screen**, plug in the power supply to start the system, and you can see the system interface on the screen



3. 29. 2. Instructions for the use of LCD2 interface

It should be noted that to use the LCD2 interface, you need to open the configuration of LCD1 and LCD2 at the same time, LCD2 can display normally, open the configuration of LCD2 alone, and test LCD2 is not displayed.

Linux5.18 does not support LCD2 yet.

1) In the dts of the Linux system, LCD2 is turned off by default. First, you need to open the configuration of LCD2 in dts.

a. For Linux 4.4 kernel, the method is as follows

a) A script named **orangepi-add-overlay** is pre-installed in the linux system. Through this script, we can use DT overlay to dynamically modify the configuration in dts. First write the rockchip-lcd2.dts file. If the LCD2 interface is connected to a MIPI screen with **model AFJ101BA2131**, the content of the file is as follows

```
root@orangepi4-lts:~# cat rockchip-lcd2.dts
/dts-v1/;
```

```
/plugin/;

/ {
        compatible = "rockchip,rk3399";
        fragment@0 {
                target = <&dsi1>;
                __overlay__ {
                        status = "okay";
                };
        };
        fragment@1 {
                target = <&gt9xx_1>;
                __overlay__ {
                        status = "okay";
                };
        };
        fragment@2 {
                target = <&rkisp1_1>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@3 {
                target = <&isp1_mmu>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@4 {
                target = <&mipi_dphy_tx1rx1>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@5 {
                target = <&ov13850_1>;
```

```
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@6 {
                target = <&cdn_dp>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@7 {
                target = <&hdmi>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@8 {
                target = <&afj101_panel1>;
                __overlay__ {
                        status = "okay";
                };
        };
};
```

b) If the LCD2 interface is connected to a MIPI screen with **model SL101PN27D1665**, the file content is as follows

```
root@orangepi4-lts:~# cat rockchip-lcd2.dts
/dts-v1/;
/plugin/;

/ {
        compatible = "rockchip,rk3399";
        fragment@0 {
                target = <&dsi1>;
                __overlay__ {
                        status = "okay";
                };
```

```
        };
        fragment@1 {
                target = <&gt9xx_1>;
                __overlay__ {
                        status = "okay";
                };
        };
        fragment@2 {
                target = <&rkisp1_1>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@3 {
                target = <&isp1_mmu>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@4 {
                target = <&mipi_dphy_tx1rx1>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@5 {
                target = <&ov13850_1>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@6 {
                target = <&cdn_dp>;
                __overlay__ {
                        status = "disabled";
                };
```

```
        };
        fragment@7 {
                target = <&hdmi>;
                __overlay__ {
                        status = "disabled";
                };
        };
        fragment@8 {
                target = <&sl101_panel1>;
                __overlay__ {
                        status = "okay";
                };
        };
};
```

c) Then you can use **orangepi-add-overlay** to compile rockchip-lcd2.dts into rockchip-lcd2.dtbo

```
root@orangepi:~# orangepi-add-overlay rockchip-lcd2.dts
Compiling the overlay
Copying the compiled overlay file to /boot/overlay-user/
Reboot is required to apply the changes
```

d) Then restart the Linux system, you can see the following printing information in the log output from the serial port, indicating that rockchip-lcd2.dtbo is loaded successfully

```
1135 bytes read in 6 ms (184.6 KiB/s)
Applying user provided DT overlay rockchip-lcd2.dtbo
```

e) After entering the Linux system, enter the following command to see the relevant information of the kernel output

```
root@orangepi4-lts:~# dmesg |grep mipi
root@orangepi4-lts:~#   dmesg |grep mipi
[    2.714240] dw-mipi-dsi ff960000.dsi: final DSI-Link bandwidth: 444 x 4 Mbps
[    3.104433] dw-mipi-dsi ff968000.dsi: final DSI-Link bandwidth: 444 x 4 Mbps
```

b. For Linux5.10 system, the method is as follows

a) You can set the overlays variable in /boot/orangepiEnv.txt to open the configuration of LCD2. If the LCD1 and LCD2 interfaces are connected to

MIPI screens with **model AFJ101BA2131**, the overlays variable configuration is as follows

```
root@orangepi4-lts:~# cat /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
overlay_prefix=rockchip
fdtfile=rockchip/rk3399-orangepi-4-lts.dtb
rootdev=UUID=c51e6614-42cf-473c-9134-46a72667eb9c
rootfstype=ext4
overlays=lcd1-afj101 lcd2-afj101
```

b) If the LCD1 and LCD2 interfaces are connected to MIPI screens with **model SL101PN27D1665**, the overlays variable configuration is as follows

```
root@orangepi4-lts:~# cat /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
overlay_prefix=rockchip
fdtfile=rockchip/rk3399-orangepi-4-lts.dtb
rootdev=UUID=c51e6614-42cf-473c-9134-46a72667eb9c
rootfstype=ext4
overlays=lcd1-sl101 lcd2-sl101
```

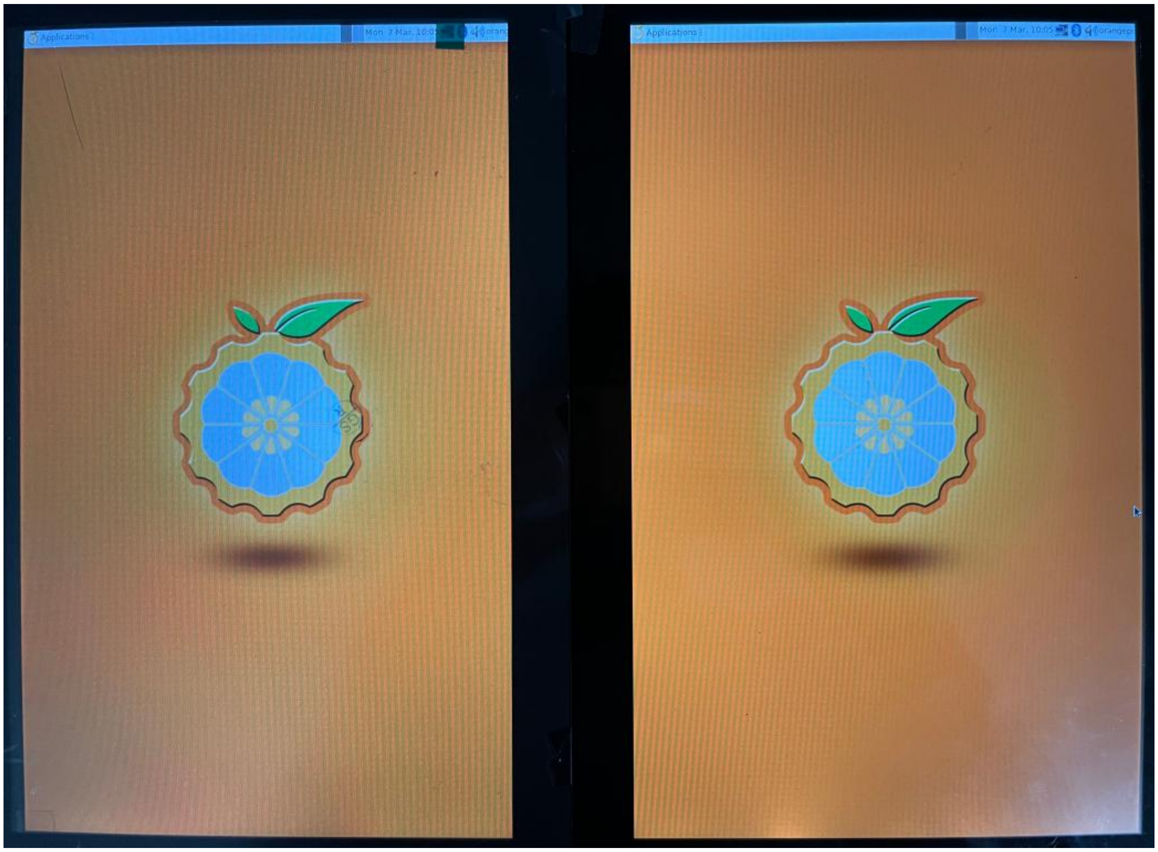
c) Then restart the system. When starting, you can see the following printing information in the startup log of u-boot, indicating that the configurations of lcd1 and lcd2 are loaded successfully

```
379 bytes read in 9 ms (41 KiB/s)
Applying kernel provided DT overlay rockchip-lcd1-afj101.dtbo
626 bytes read in 9 ms (67.4 KiB/s)
Applying kernel provided DT overlay rockchip-lcd2-afj101.dtbo
```

d) After entering the Linux system, enter the following command to see the relevant information of the kernel output

```
root@orangepi4-lts:~# dmesg |grep mipi
[       8.124926] dw-mipi-dsi-rockchip ff960000.dsi:
[drm:dw_mipi_dsi_bridge_mode_set] final DSI-Link bandwidth: 444 x 4 Mbps
[       8.168243] dw-mipi-dsi-rockchip ff968000.dsi:
[drm:dw_mipi_dsi_bridge_mode_set] final DSI-Link bandwidth: 444 x 4 Mbps
```

2) Then disconnect the power supply of the development board, and then connect the MIPI screen by **referring to the usage of the 10.1-inch MIPI screen**, plug in the power supply to start the system, if two 10.1-inch screens are connected at the same time, you can see the system on the two 10.1-inch screens interface



3. 30. How to set dual-screen simultaneous display

1) The Linux image on the official website supports HDMI output and TypeC-DP output by default. You can use the following commands to set dual-screen simultaneous display

```
root@orangepi4-lts:~#
su orangepi -c "DISPLAY=:0 xrandr --output HDMI-1 --same-as DP-1"
```

2) If you use MIPI LCD output and connect two 10.1-inch MIPI screens, you can use the following commands to set

```
root@orangepi4-lts:~#
su orangepi -c "DISPLAY=:0 xrandr --output DSI-1 --same-as DSI-2"
```

If the system restarts after setting, please confirm whether the power supply is sufficient.

3. 31. Set up Chinese environment and install Chinese input method

Note that before installing the Chinese input method, please make sure that the Linux system used by the development board is the desktop version system.

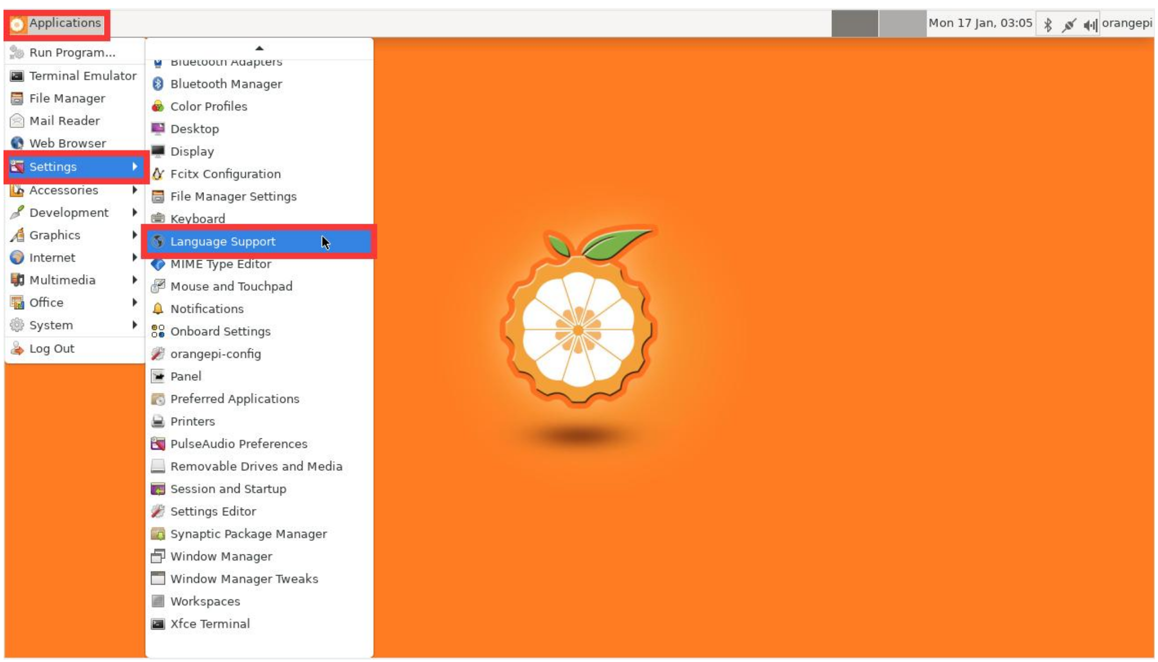
3. 31. 1. Ubuntu system installation method

1) First update the software source of the system

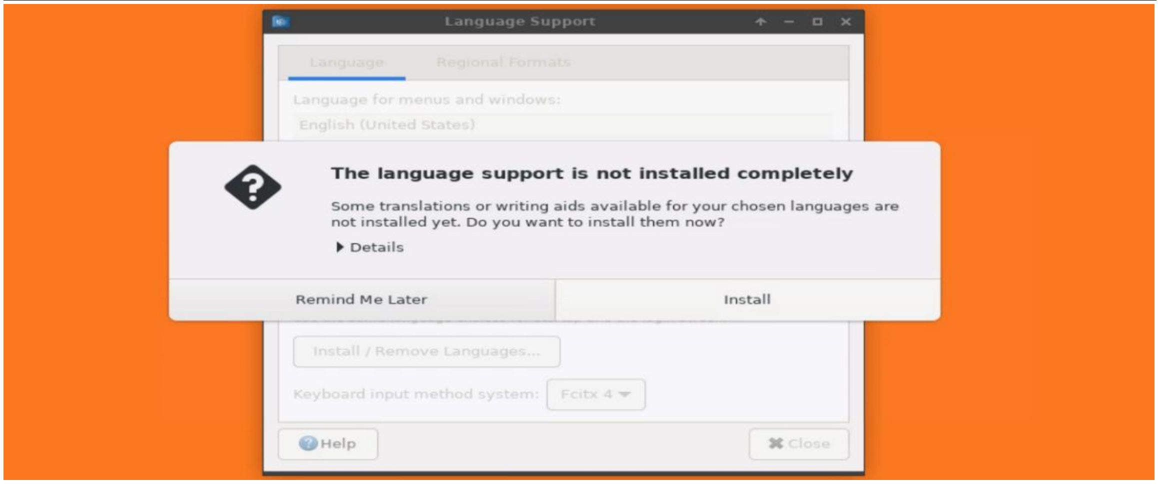
Note that an error may be reported during the installation process without executing the following command, so please do not ignore this step。

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

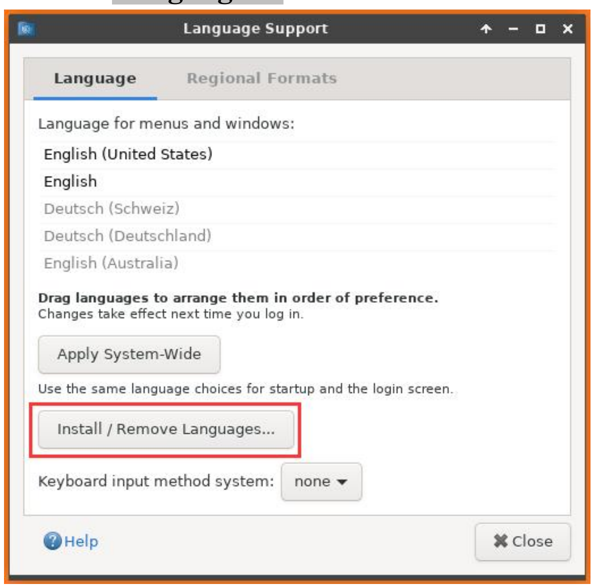
2) Then open **Language Support**



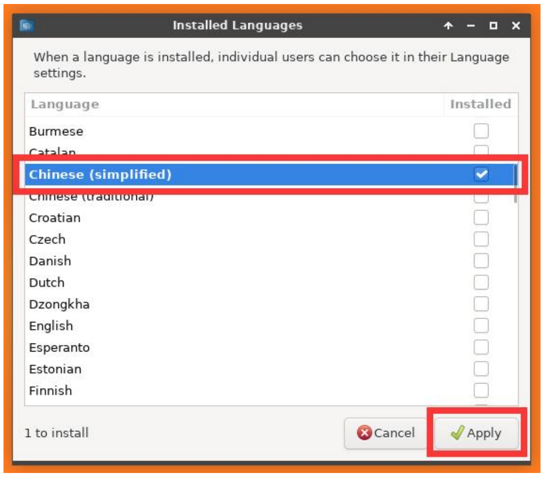
3) If the following information is prompted, click **Install** to repair it. Some systems will not have the following prompt, just skip it.



4) Then open Install/Remove **Languages...**

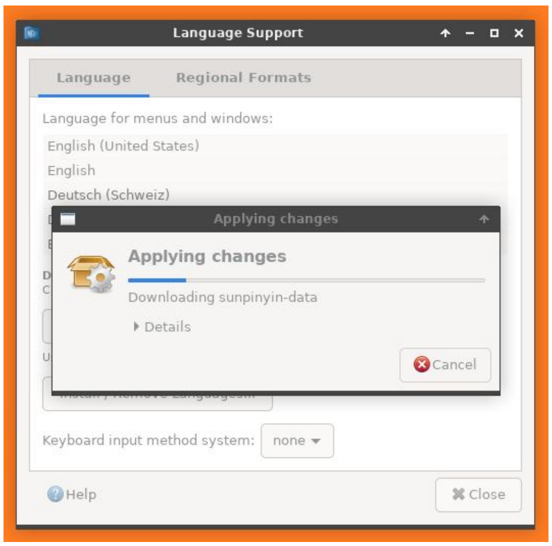


5) Then find **Chinese (simplified)**, click the box on the right to select it, and then click **Apply** in the lower right corner



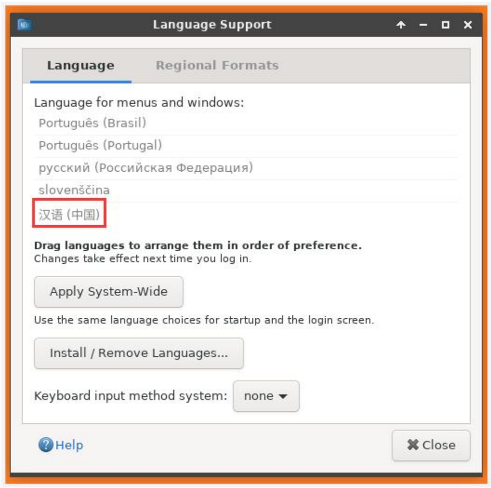
6) Then enter the password of the Linux system in the pop-up password input interface, the default is **orangepi**

7) Then it will start to install the required software packages. At this time, wait patiently for the installation to complete.

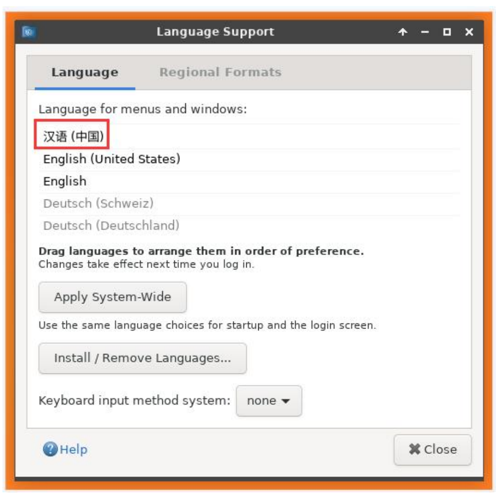


If the software installation fails, it is generally because the apt update command was not executed at the beginning. If the apt update command is executed and the prompt fails, it is generally because an error occurred when the apt update command was executed, but it was not executed successfully.

8) After the installation is complete, you can see the **Chinese (China)** option

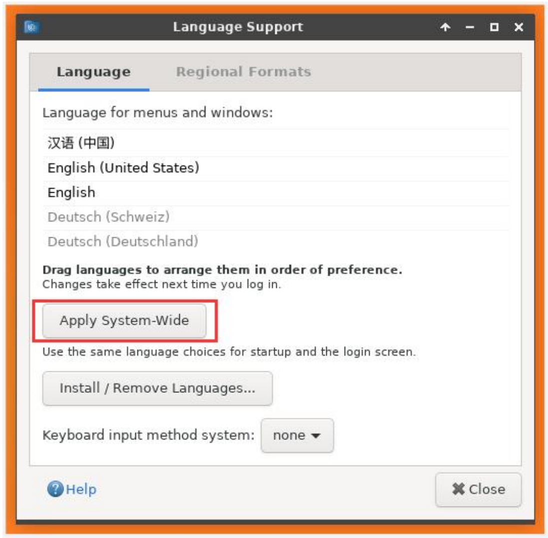


9) Then please use the left mouse button to select **Chinese (China)** and hold it down, then drag it up to the first position, the display after dragging is as shown below

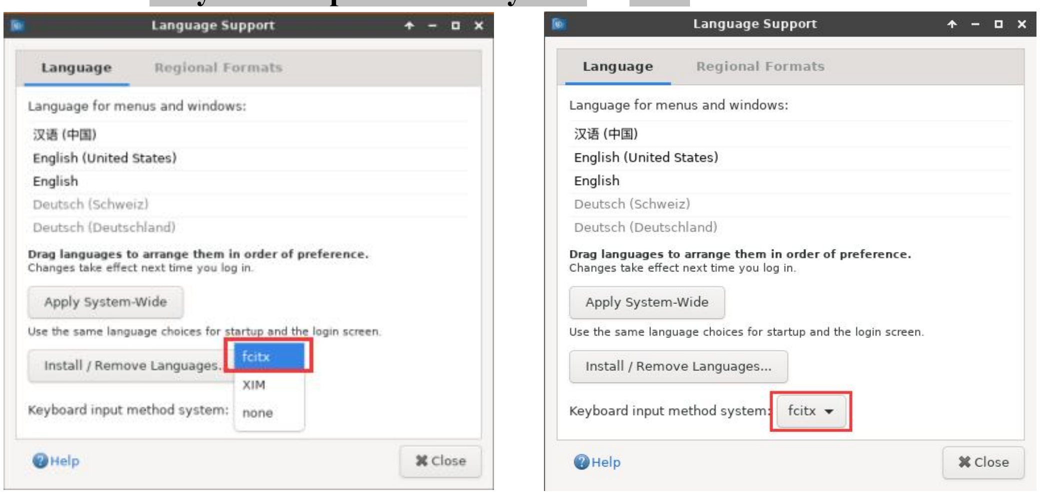


Note that this step is not very easy to drag, please be patient and try a few more times.

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



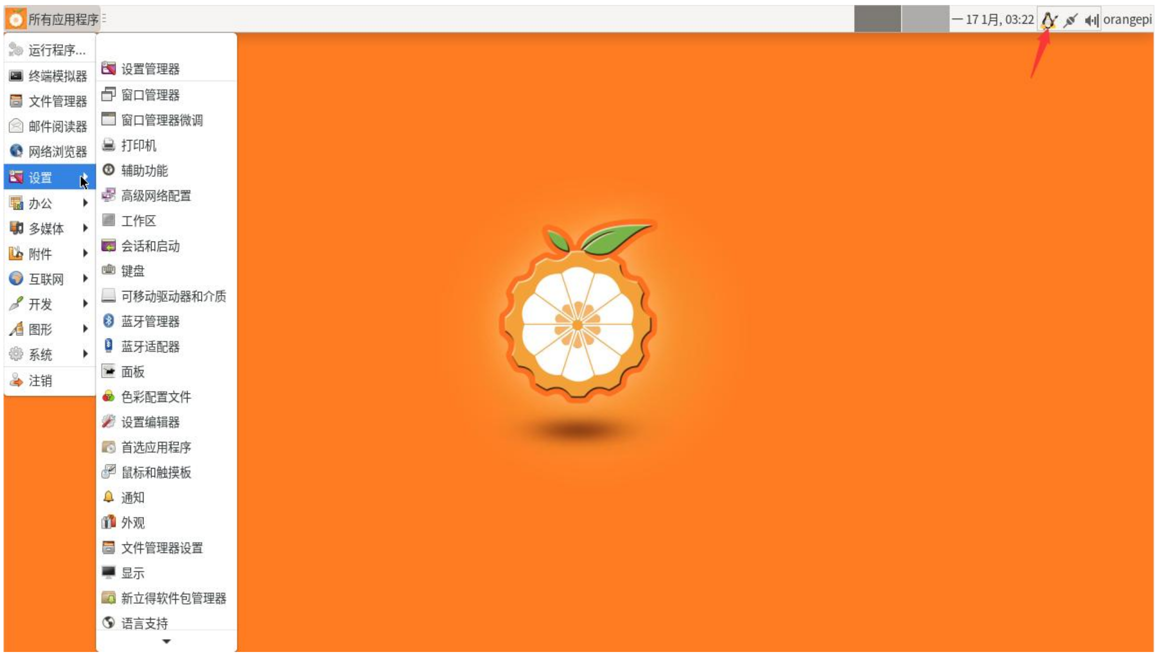
11) Then select **Keyboard input method system** as **fcitx**



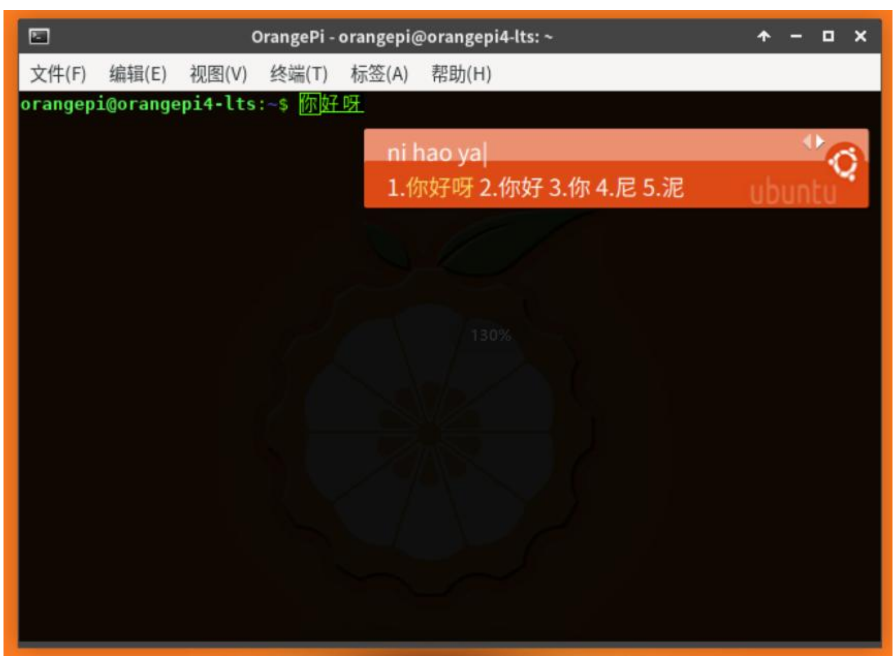
12) **Then restart the Linux system to make the configuration take effect**

13) After re-entering the system, you can see that the desktop is displayed in Chinese, and you can also see a penguin in the upper right corner of the desktop

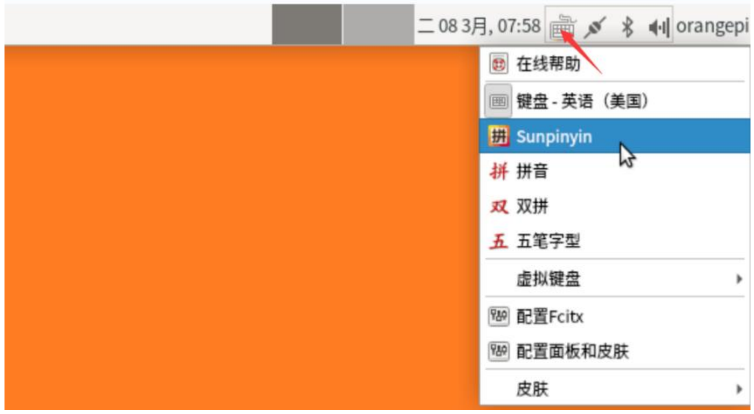
Note that Ubuntu 22.04 shows a black keyboard icon in the upper right corner.



14) Then we can open the terminal to test the Chinese input method. After opening the terminal, the default is the English input method. We can switch to the Chinese input method through the **Ctrl+Space** shortcut key, and then we can enter Chinese.



15) In addition, you can place the mouse on the penguin in the upper right corner of the desktop, and then click the **right mouse button** to view all input methods supported by the system, or select the input method to be used



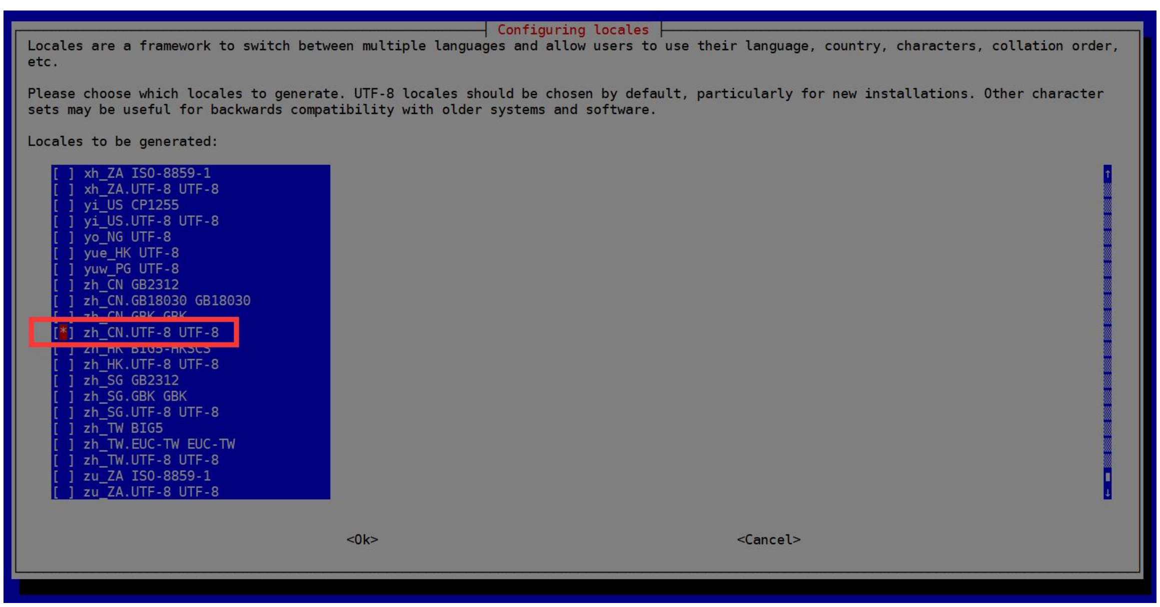
3. 31. 2. Installation method of Debian system

1) First set the default **locale** to Chinese

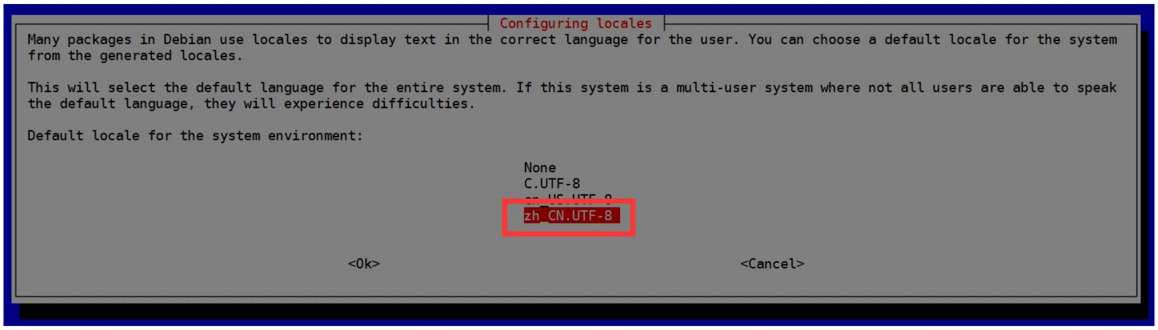
a. Enter the following command to start configuring **locale**

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

b. Then select **zh_CN.UTF-8 UTF-8** in the pop-up interface (move up and down through the up and down direction keys on the keyboard, select through the space bar, and finally use the Tab key to move the cursor to **<OK>**, then press Enter key)



c. Then set the default **locale** to **zh_CN.UTF-8**



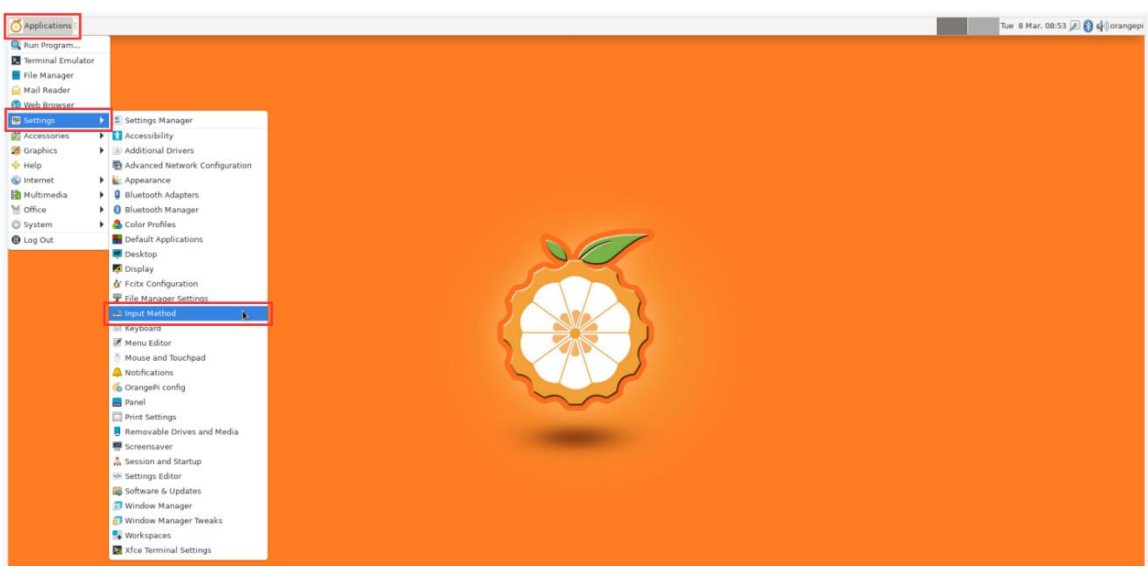
d. After exiting the interface, the **locale** setting will start, and the output displayed on the command line is as follows

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
  en_US.UTF-8... done
  zh_CN.UTF-8... done
Generation complete.
```

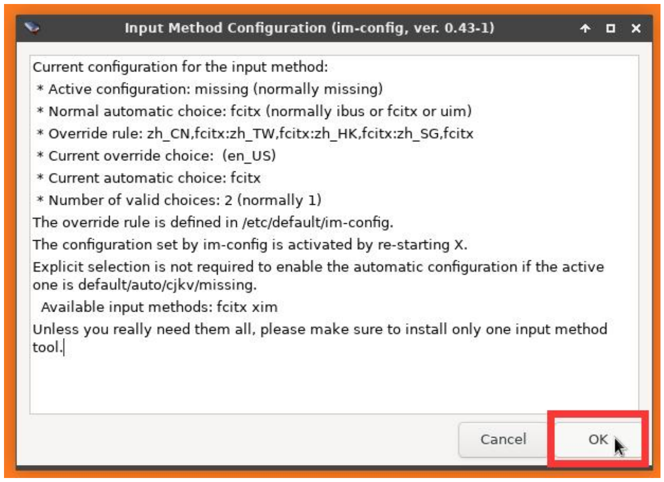
2) Then install the following packages

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y fonts-arphic-bsmi00lp   \
fonts-arphic-gbsn00lp fonts-arphic-gkai00mp fcitx fcitx-table*   \
fcitx-frontend-gtk* fcitx-frontend-qt* fcitx-config-gtk* im-config   \
fcitx-googlepinyin fcitx-ui* zenity geany
```

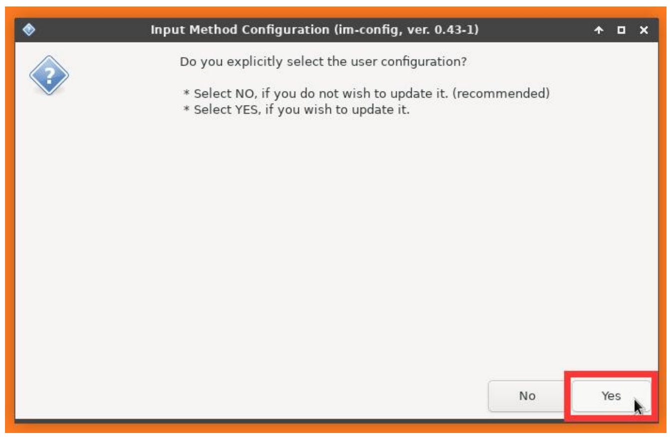
3) Then open **Input Method**



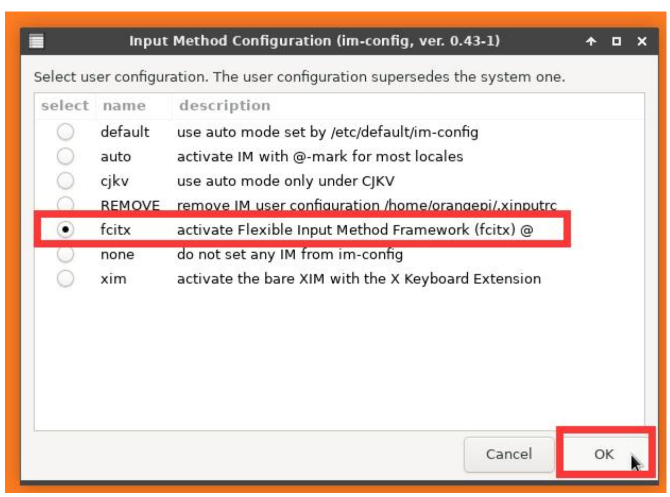
4) Then select **OK**



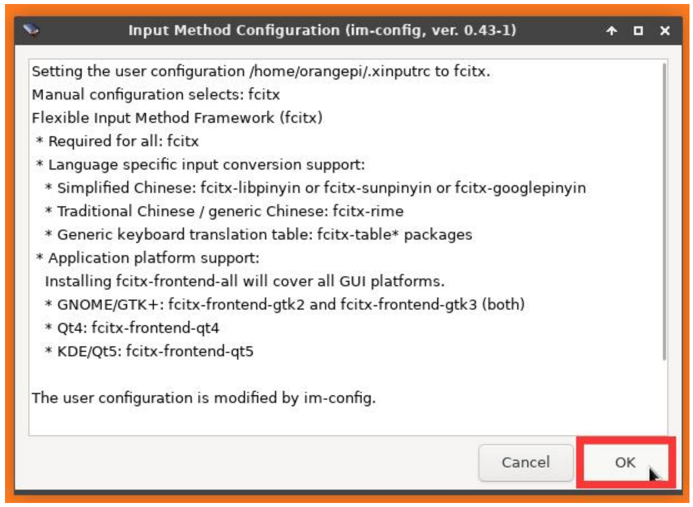
5) Then select **Yes**



6) Then select **fcitx**

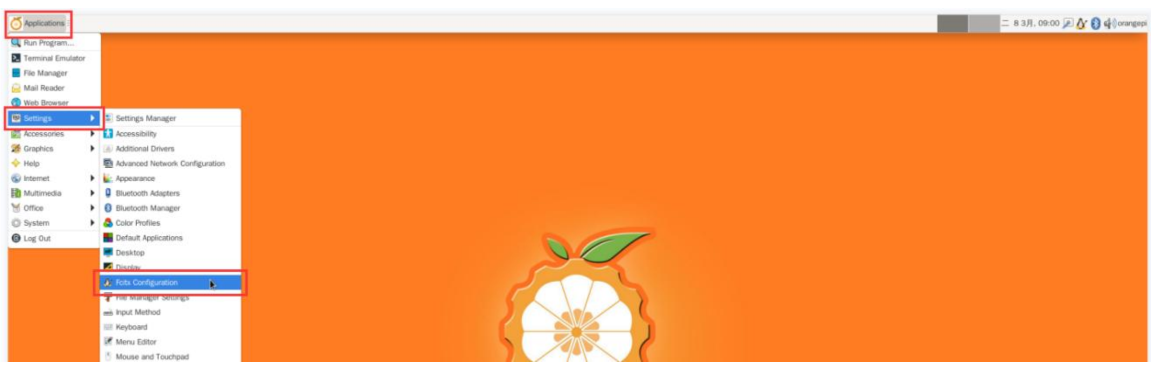


7) Then select **OK**

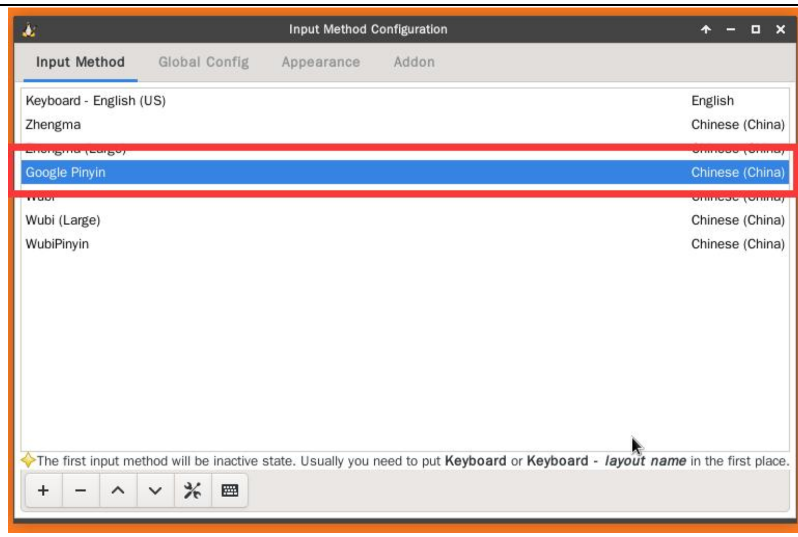


8) **Then restart the Linux system for the configuration to take effect**

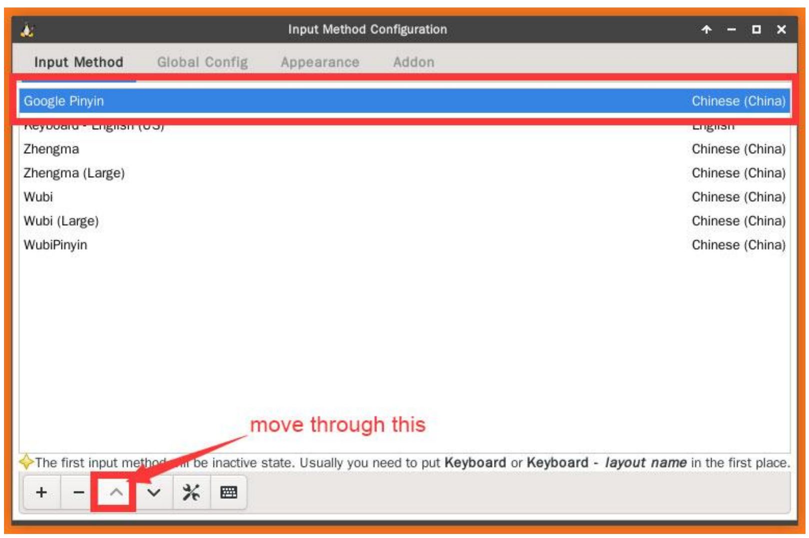
9) Then open **Fcitx configuration**



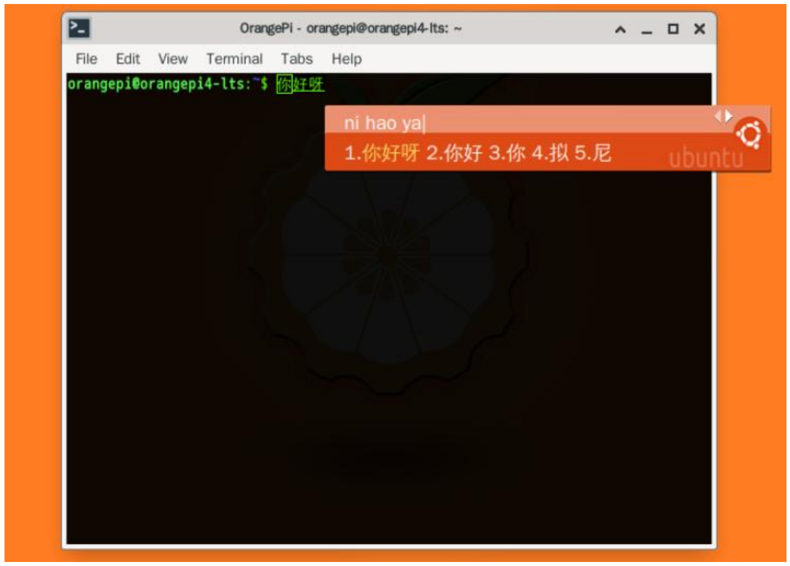
10) Then select **Google Pinyin**



11) Then bring **Google Pinyin** to the front, and then close the configuration interface in the upper right corner



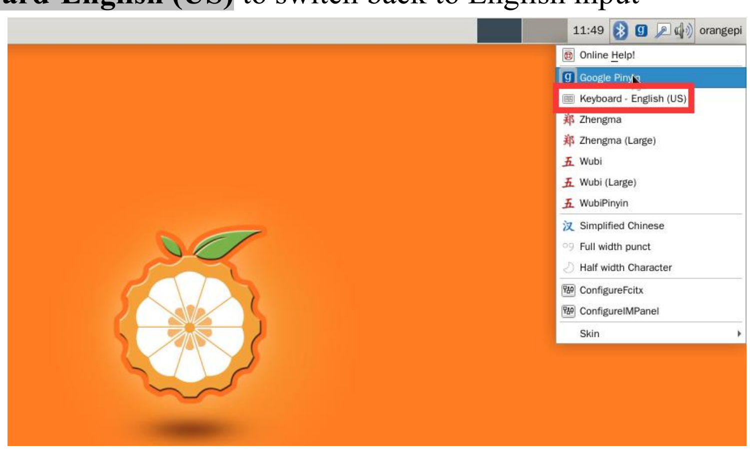
12) Then open the terminal to test the Chinese input method, as shown in the figure below



13) You can switch back to English input in the upper right corner of the desktop

a. First place the mouse cursor on the penguin position as shown in the figure below

b. Then click the **right mouse button** to see the following options, and then select **Keyboard-English (US)** to switch back to English input

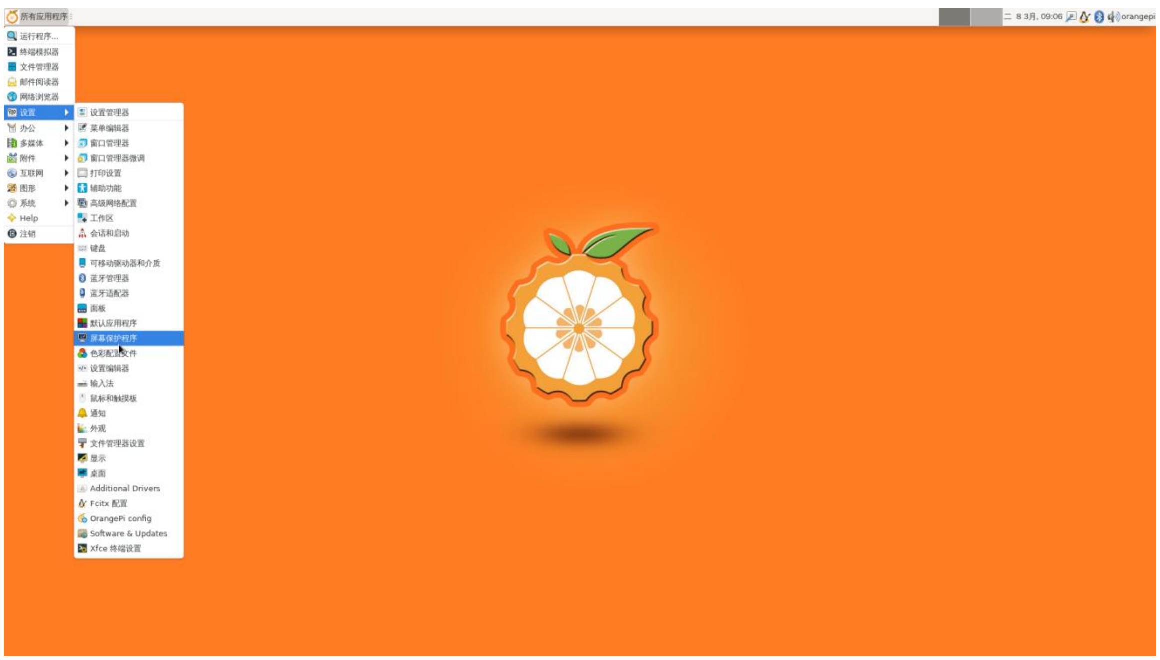


14) At this point, you can switch between Chinese and English input methods through the **Ctrl+Space** shortcut.

15) If you need the entire system to be displayed in Chinese, you can set the variables in **/etc/default/locale** to **zh_CN.UTF-8**

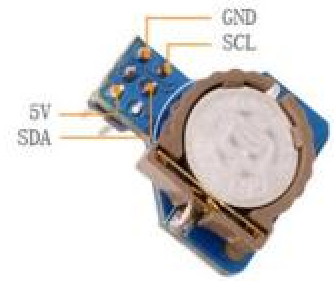
```
root@orangepi:~# cat /etc/default/locale
#  File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** to see that the desktop is displayed in Chinese



3. 32. How to use the orange pi DS1307 RTC clock module

1) The orange pi DS1307 RTC clock module is shown in the figure below. It uses the i2c interface to communicate with the development board, and the i2c device address is 0x68. The RTC module is not equipped with a battery by default, and a button battery needs to be prepared before use



2) First connect the RTC module to the 40pin of the development board, the wiring method is as follows

Pins of the RTC module	The corresponding pin of the development board 26pin
5V	pin 2
GND	pin 6
SDA	pin 3
SCL	pin 5

3) After connecting the RTC module, first use the i2cdetect command to check whether the device address of the RTC module can be detected

root@orangepi:~# **apt update**
root@orangepi:~# **apt -y install i2c-tools**
root@orangepi:~# **i2cdetect -y 8**

```
root@orangepi4-lts:~# i2cdetect -y 8
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

4) Because the kernel has opened the ds1037 driver by default, the function can be tested directly. Execute the following command to add an rtc device and view the generated rtc device, where rtc0 is the onboard rtc and rtc1 is the newly added external rtc

root@orangepi:~# **echo "ds1307 0x68" > /sys/class/i2c-adapter/i2c-8/new_device**

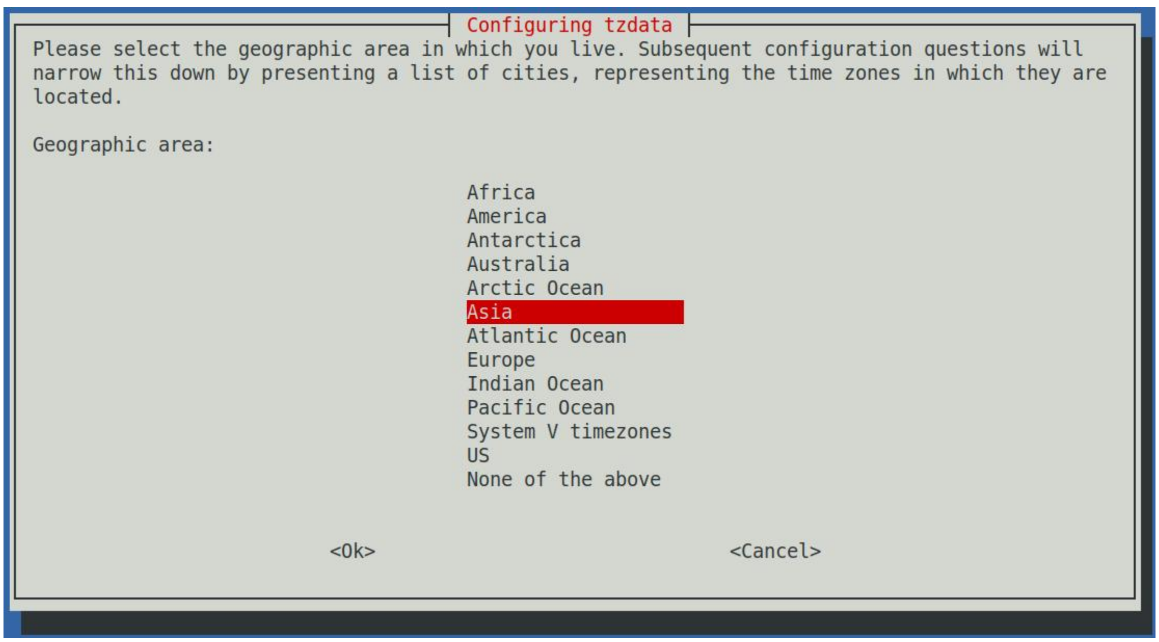
```
root@orangepi:~# ls /dev/rtc*
/dev/rtc    /dev/rtc0    /dev/rtc1
```

5) When the linux system starts, if the development board is connected to the network, the linux system will automatically synchronize the system time to the correct time through the network. The default time of the linux system is UTC. In China, the time zone needs to be changed to **Asia/Shanghai**. The time obtained by using the data command is correct, the method is as follows

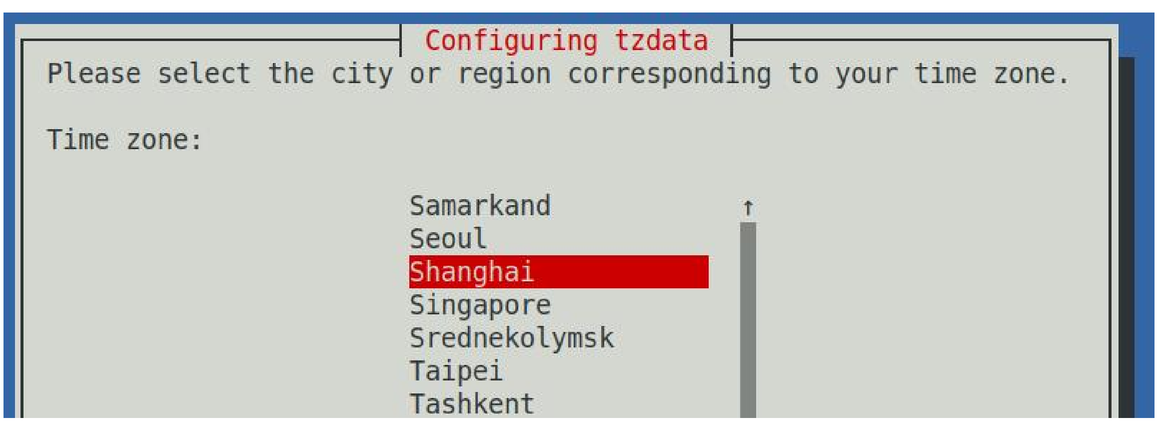
a. Execute the following command

```
root@orangepi:~# dpkg-reconfigure tzdata
```

b. Then select the geographic area as **Asia**



c. Then select the time zone as **Shanghai**



d. After the configuration is completed, use the date command to view the time and it will be normal

```
root@orangepi:~# date
```

6) If the current time of the system is incorrect, please connect to the network first, and then use the following command to synchronize the time. The reason why the system time is set correctly is to prepare for the synchronization of the time of the RTC module later.

```
root@orangepi:~# apt -y update
root@orangepi:~# apt install ntpdate
root@orangepi:~# ntpdate 0.cn.pool.ntp.org
```

7) The command to view the current time of the RTC module is as follows, because the rtc specified by hwclock by default is the rtc0 device, so you need to use the -f option to specify the rtc1 device

```
root@orangepi:~# hwclock -r -f /dev/rtc1
```

8) The time read by the RTC module for the first time is definitely wrong. The current time of the system can be synchronized to the RTC module through the following command. Before synchronization, it is necessary to ensure that the current time of the system is correct

```
root@orangepi:~# date                   #First make sure the current system time is correct
root@orangepi:~# hwclock -w -f /dev/rtc1   #Then write the system time to the RTC module
root@orangepi:~# hwclock -r -f /dev/rtc1   #Finally read the time of the RTC module to confirm that the settings are correct
```

9) If it is confirmed that the time in the RTC module is correct, then you can unplug the power supply, then turn on the power, and execute the following command to synchronize the time in the RTC module to the system

```
root@orangepi:~#
echo "ds1307 0x68" > /sys/class/i2c-adapter/i2c-8/new_device    #add rtc device
root@orangepi:~# hwclock -s -f /dev/rtc1    #Then synchronize the RTC module time to the system
root@orangepi:~# date                       #Finally, use the date command to check whether the system time is correct
```

10) The above operation is to manually synchronize the time of the RTC module to the system. If you need to automatically synchronize the system time at startup, you need to set the startup script as follows to automatically synchronize the system time

a. Create the rc-local.service file

```
root@orangepi:~# sudo vi /etc/systemd/system/rc-local.service
```

Copy the following content into the rc-local.service file

```
[Unit]
Description=/etc/rc.local Compatibility
ConditionPathExists=/etc/rc.local

[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99

[Install]
WantedBy=multi-user.target
```

b. Create the rc.local file

```
root@orangepi:~# sudo vi /etc/rc.local
```

Copy the following content into the rc-local.service file

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
```

```
# By default this script does nothing.

echo "ds1307 0x68" > /sys/class/i2c-adapter/i2c-8/new_device
hwclock -s -f /dev/rtc1

exit 0
```

c. Add permissions to rc.local

```
root@orangepi:~# sudo chmod +x /etc/rc.local
```

d. Enable service

```
root@orangepi:~# sudo systemctl enable rc-local
```

e. Start the service and check the status

```
root@orangepi:~# sudo systemctl start rc-local.service
root@orangepi:~# sudo systemctl status rc-local.service
```

11) At this point, you can disconnect all network connections of the development board, wait for a few minutes, restart the system, and then check the system time to find that even if there is no network, the system time is correct

3. 33. Installation method of pagoda Linux panel

1) Pagoda Linux panel is a server management software that improves operation and maintenance efficiency. It supports more than 100 server management functions such as one-click LAMP/LNMP/cluster/monitoring/website/FTP/database/JAVA (excerpted from **Pagoda's official website**)

2) The recommended order for Pagoda Linux system compatibility is

```
Debian10/11 > Ubuntu 20.04
```

3) Log in to the linux system and enter the following command to start the installation of the pagoda

```
orangepi@orangepi:~$ wget -O install.sh
http://download.bt.cn/install/install-ubuntu_6.0.sh && sudo bash install.sh
```

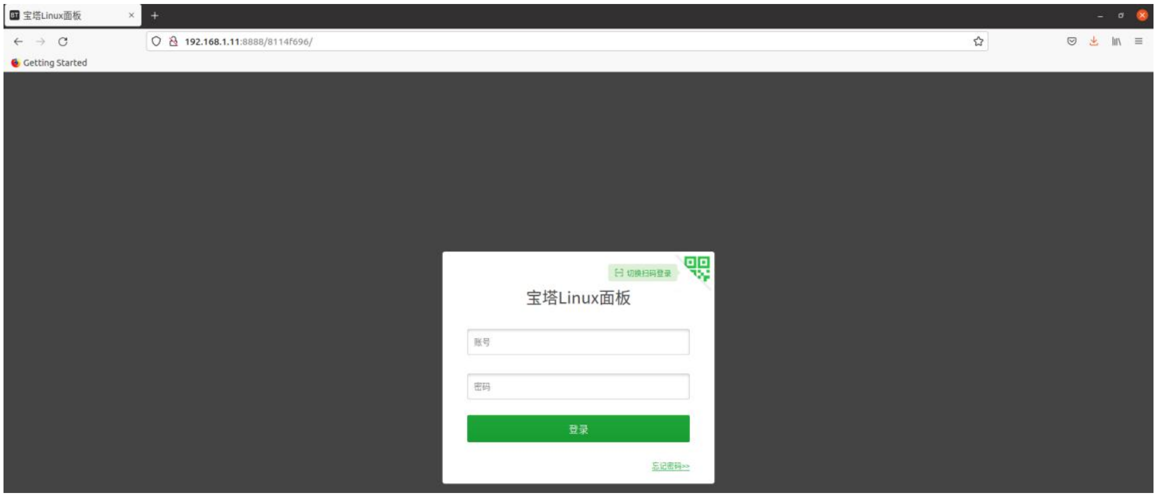
4) Then the pagoda installer will remind whether to install **Bt-Panel** to the **/www** folder, then enter **y**

```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| The WebPanel URL will be http://SERVER_IP:8888 when installed.
+----------------------------------------------------------------------

Do you want to install Bt-Panel to the /www directory now?(y/n): y
```

5) Then all you have to do is wait patiently. When you see the following print information output from the terminal, it means that the pagoda has been installed. The whole installation process takes about 44 minutes, and there may be some differences depending on the network speed.

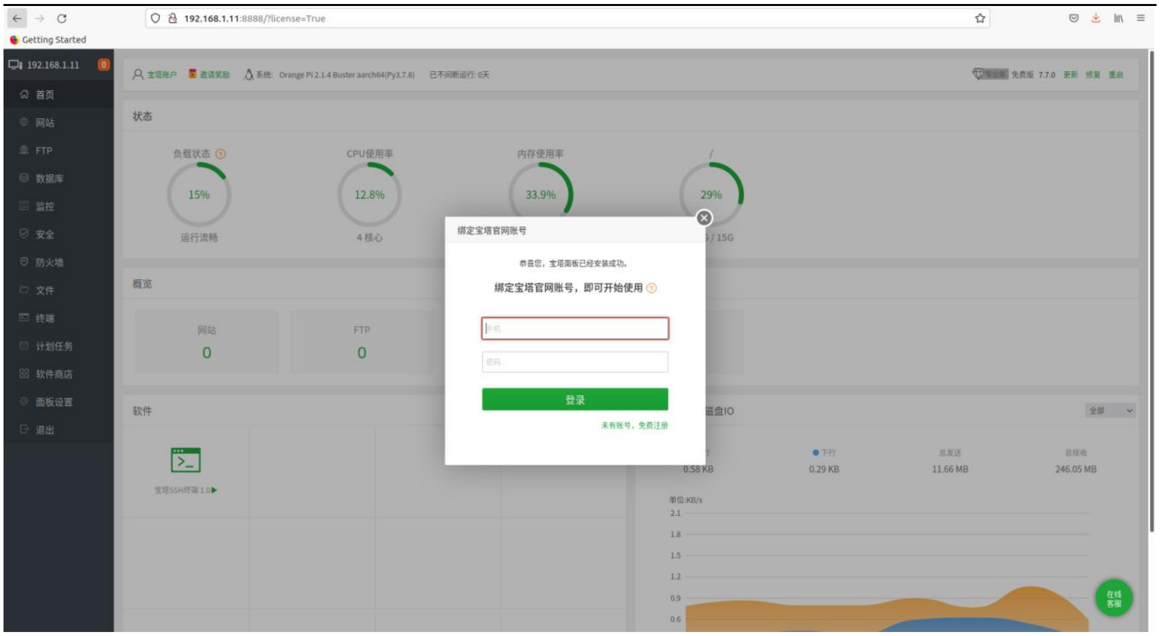
```
==================================================================
Congratulations! Installed successfully!
==================================================================
外网面板地址: http://113.116.156.118:8888/f8467e2b
内网面板地址: http://192.168.1.56:8888/f8467e2b
username: spqwngso
password: 72762417
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: 44 Minute!
```

6) At this time, enter the **panel address** shown above in the browser to open the login interface of the pagoda Linux panel, and then enter the **username** and **password** shown in the above figure in the corresponding position to log in to the pagoda

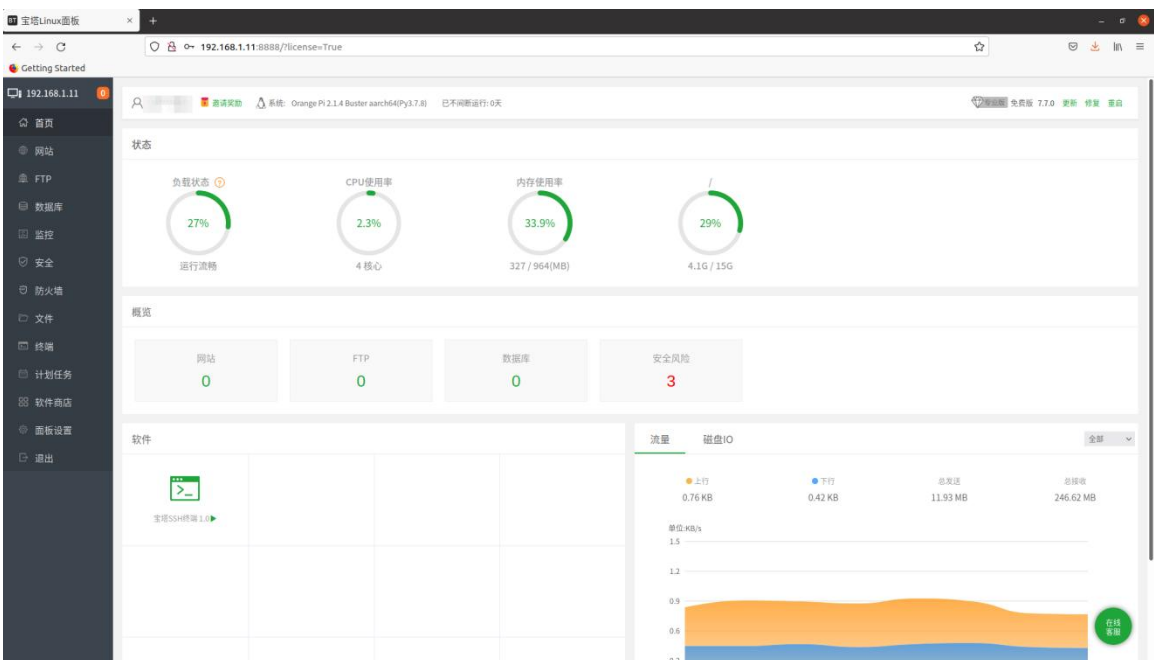


7) After successfully logging in to the pagoda, the following welcome interface will pop up. First, please read the user instructions in the middle and drag it to the bottom, then you can select "I have agreed and read the "User Agreement", and then click "Enter the panel" You can enter the pagoda

8) After entering the pagoda, you will first be prompted to bind an account on the official website of the pagoda. If you do not have an account, you can go to the official website of the pagoda (**https://www.bt.cn**) to register one

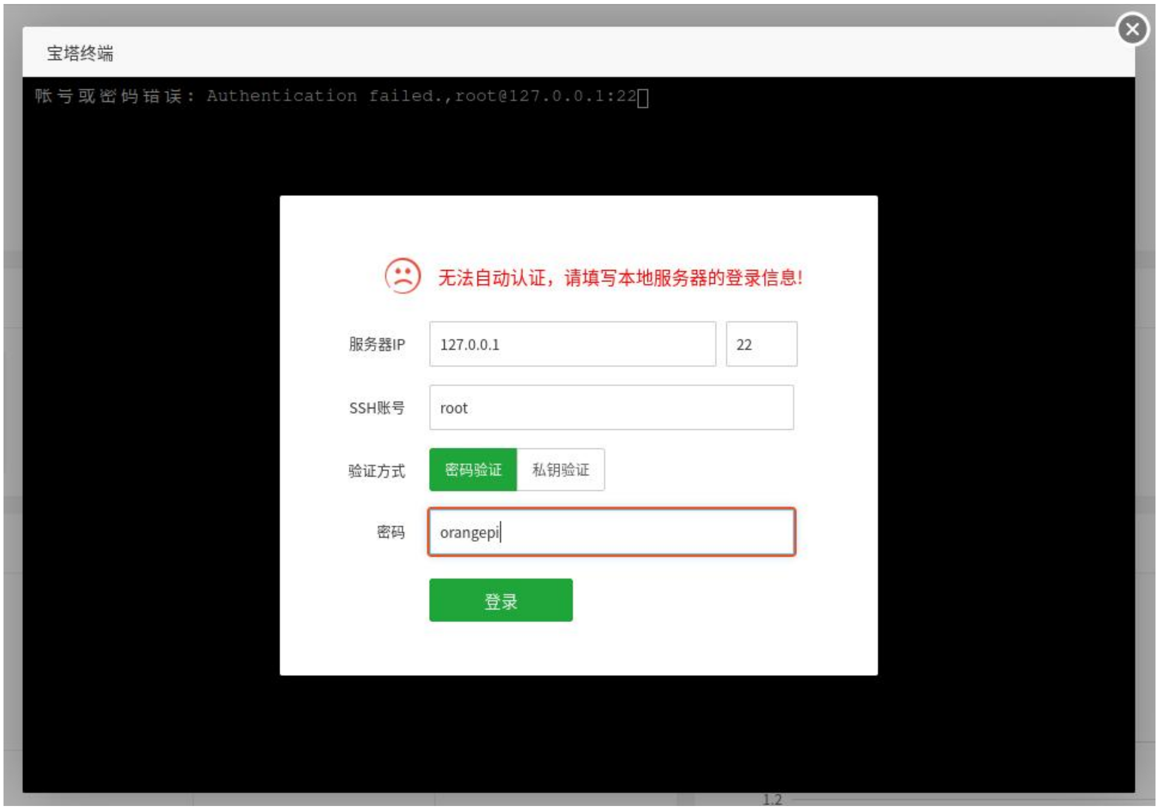


9) The final displayed interface is shown in the figure below. You can intuitively see some status information of the Linux system of the development board, such as load status, CPU usage, memory usage, and storage space usage, etc.

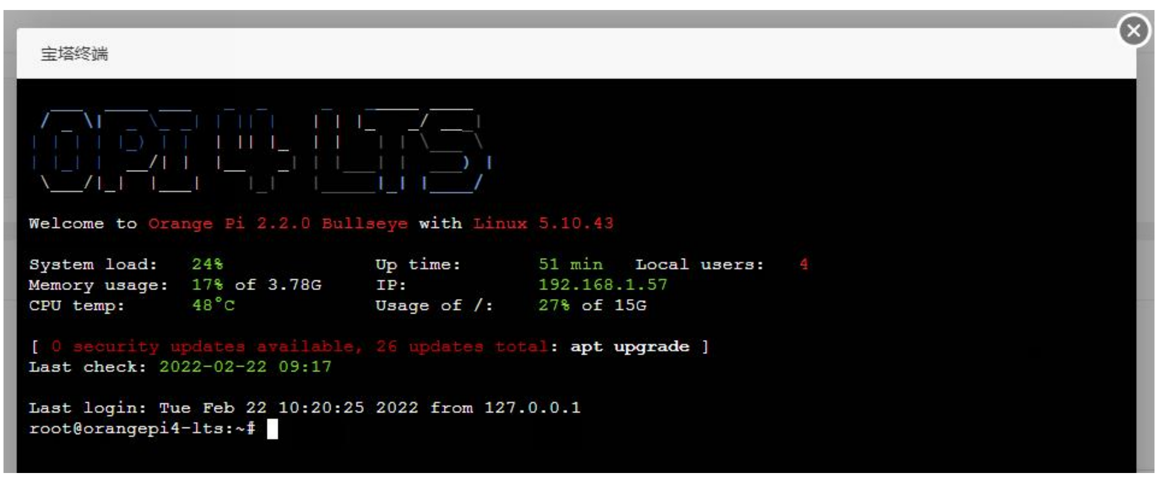


10) Test the SSH terminal login of the pagoda

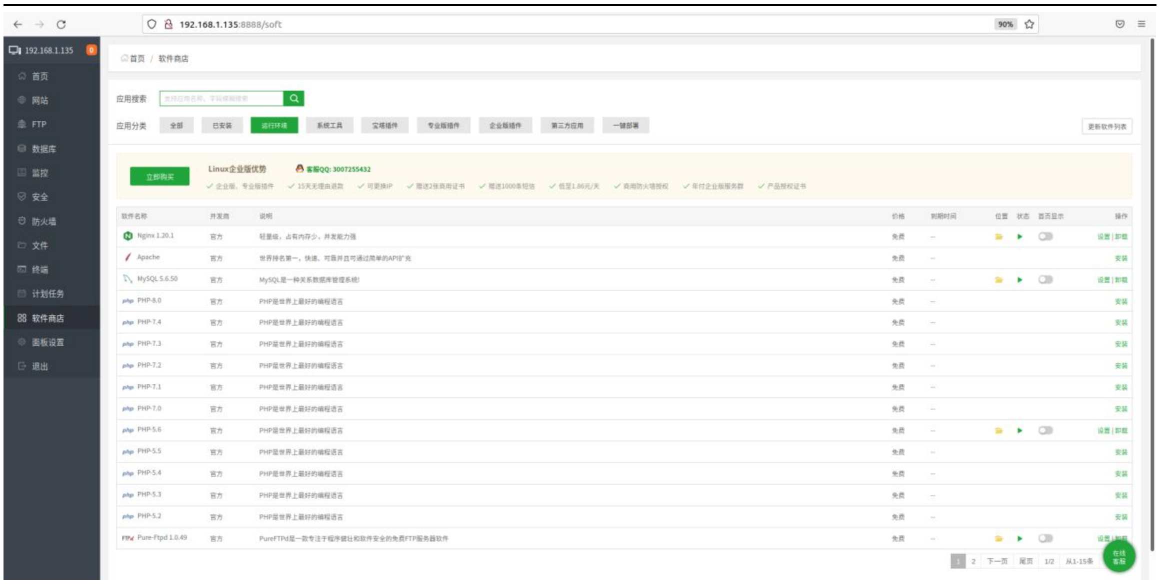
a. After opening the SSH terminal of the pagoda, you will first be prompted to enter the password of the development board system. At this time, enter **orangepi** in the password box (the default password, if there is any modification, please fill in the modified one)



b. The display after successful login is as shown below



11) Software such as Apache, MySQL and PHP can be installed in the software store of the pagoda, and various applications can also be deployed with one click. Please explore these functions by yourself, and I will not demonstrate them one by one here.



12) Pagoda command line tool test

```
root@orangepi4-lts:~# bt
===============宝塔面板命令行==================
(1) 重启面板服务           (8) 改面板端口
(2) 停止面板服务           (9) 清除面板缓存
(3) 启动面板服务           (10) 清除登录限制
(4) 重载面板服务
(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)...
===============================================
==================================================================
BT-Panel default info!
==================================================================
外网面板地址: http://116.30.192.202:8888/0d375687
内网面板地址: http://192.168.1.57:8888/0d375687
*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: zv2hscnh
password: 71125e2f
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
==================================================================
root@orangepi4-lts:~#
```

13) For more functions of the pagoda, you can refer to the following information to

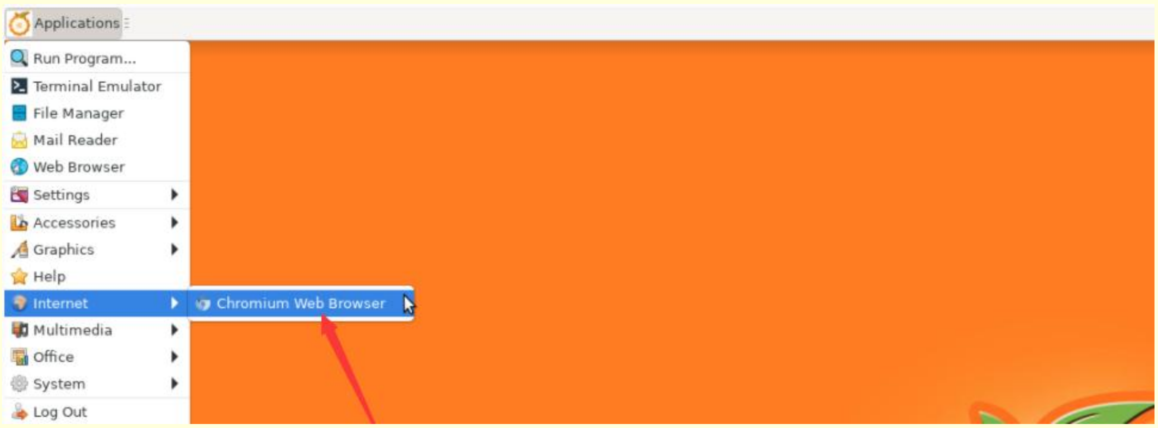
explore by yourself

manual：http://docs.bt.cn
Forum address：https://www.bt.cn/bbs
GitHub link：https://github.com/aaPanel/BaoTa

3. 34. Ubuntu22.04 method of installing browser

3. 34. 1. Ubuntu22.04 Chromium browser installation method

If you are using the latest version of the Ubuntu 22.04 image, the deb version of the Chromium browser is pre-installed. Before installation, please check whether the Chromium browser already exists in the following location. If so, you don't need to install it again, you can use it directly.



1) The software repository of Ubuntu 22.04 only supports the snap version of the Chromium browser. The installation command is as follows：

```
orangepi@orangepi:~$ sudo snap install chromium
```

2) If you don't want to install the snap version of the Chromium browser, you can also install the deb version of the Chromium browser by the following method：

a. The storage address of the Chromium browser installation package of the a.deb version is as follows

https://launchpad.net/~saiarcot895/+archive/ubuntu/chromium-beta
https://ppa.launchpadcontent.net/saiarcot895/chromium-beta/ubuntu

b. First add the PPA source of the Chromium browser to the system source

```
orangepi@orangepi:~$ sudo add-apt-repository ppa:saiarcot895/chromium-beta
```

c. When you see the following prompt, please press Enter to confirm

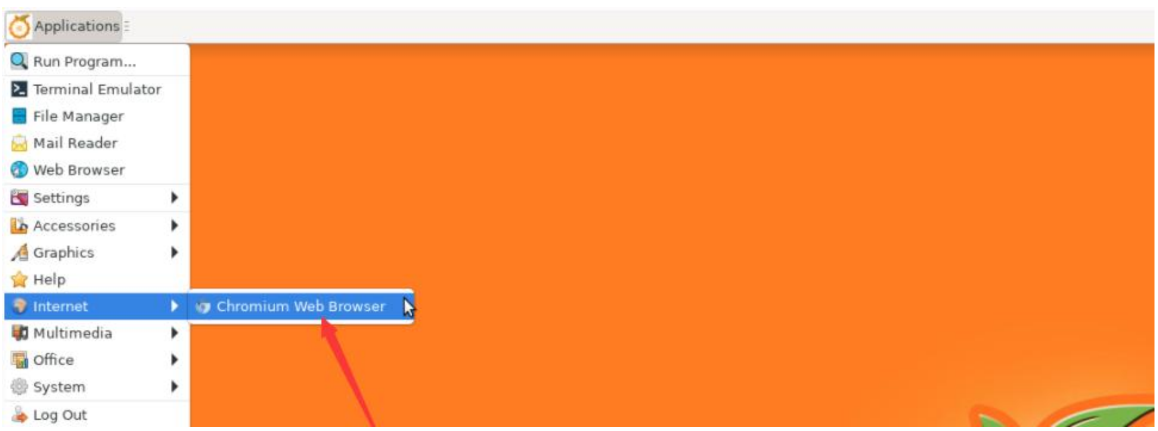
```
=== Packaging ===

Packaging for this PPA is at https://github.com/saiarcot895/chromium-ubuntu-build. A separate branch is created for each upstream branch number.
More info: https://launchpad.net/~saiarcot895/+archive/ubuntu/chromium-beta
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
```

d. Then use the following command to install the deb version of the Chromium browser

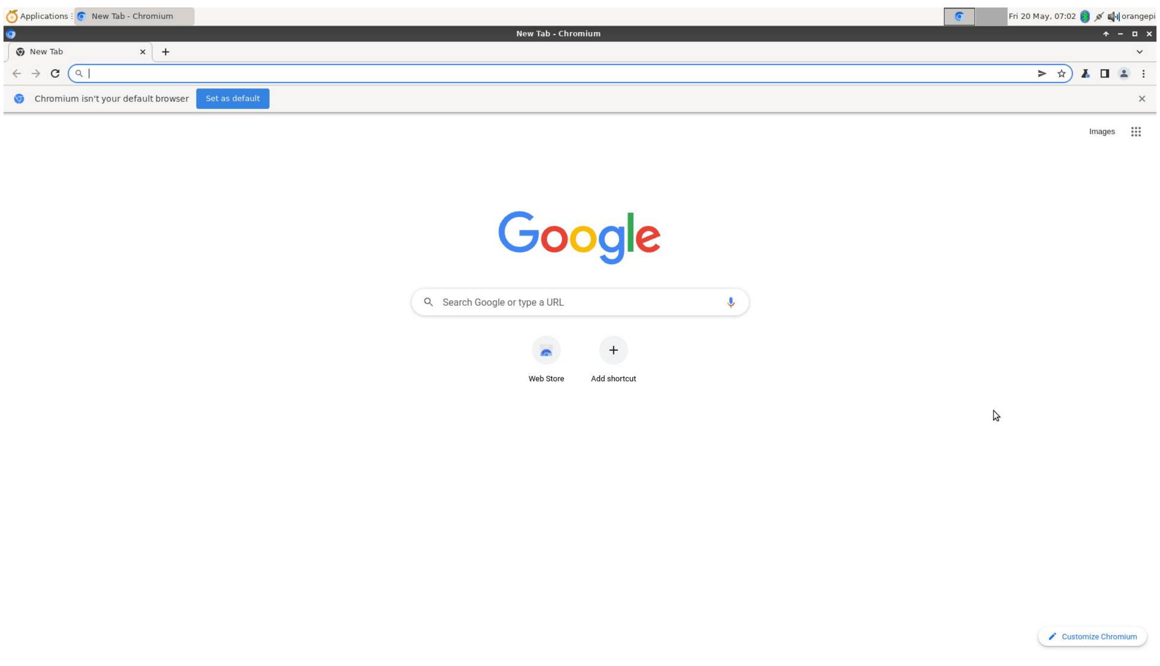
```
orangepi@orangepi:~$ sudo apt install -y chromium-browser
```

It should be noted that installing the Chromium browser in this way in China is generally very slow, and it is necessary to ensure that the network environment is better.

e. After installation, you can see the shortcut of Chromium browser in the application



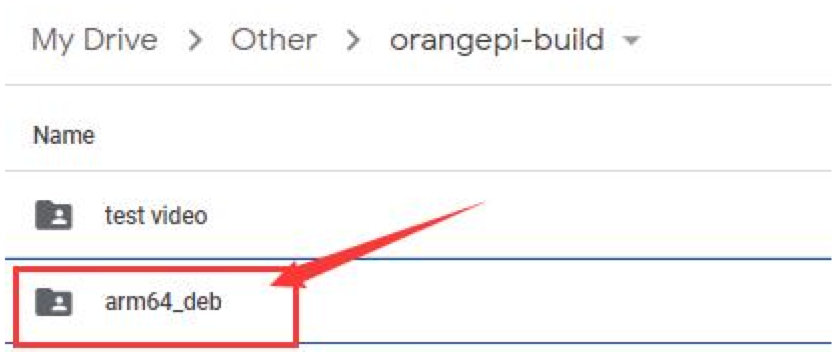
f. The display after opening the Chromium browser is as follows

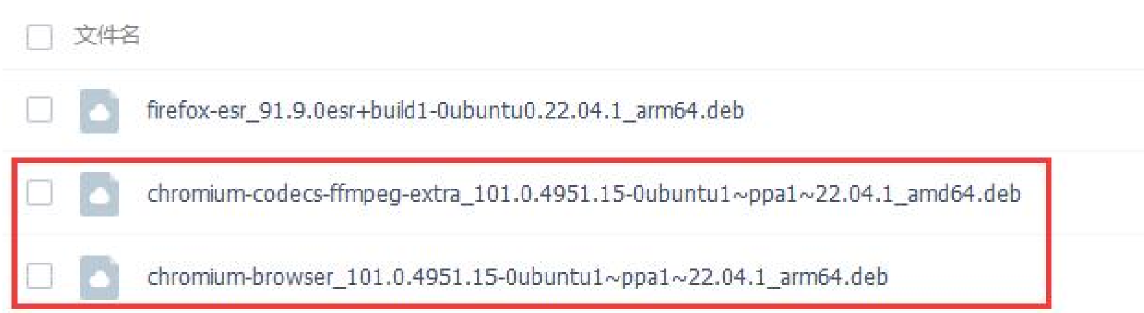


3) If the Chromium browser cannot be installed by the above method, you can directly download the corresponding deb package, and then upload it to the Ubuntu system of the development board for direct installation

a. The deb package required by the Chromium browser can be downloaded from the link below

Link:https://drive.google.com/drive/folders/1c7o6fBjTZYoervmc7oTtGemL5oww6tz0?usp=sharing



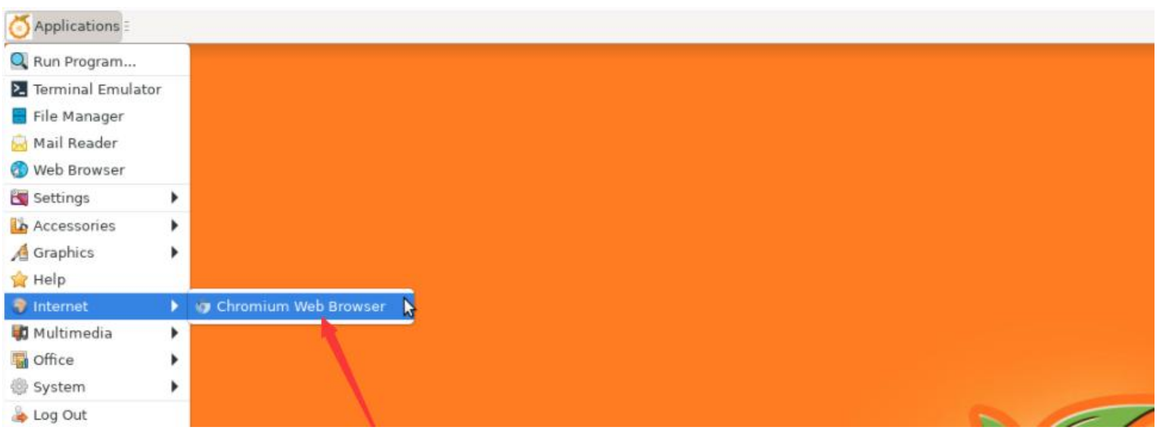


b. After downloading, upload it to the Ubuntu system of the development board

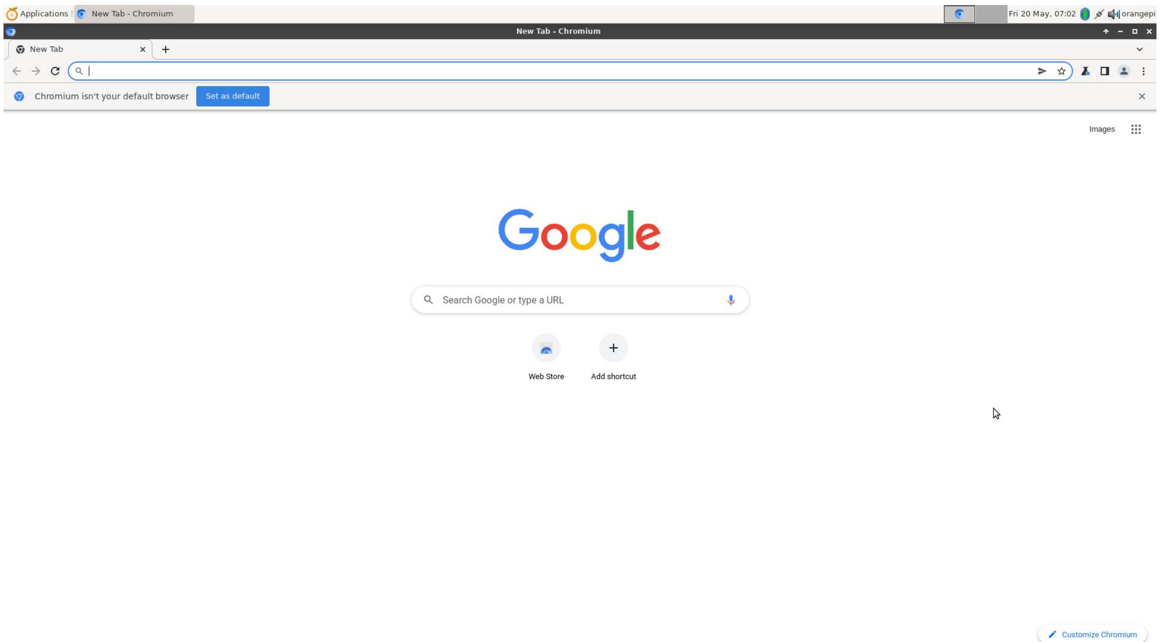
c. Then use the following command to install the deb package of the Chromium browser

```
orangepi@orangepi:~$ sudo dpkg -i    \
chromium-codecs-ffmpeg-extra_101.0.4951.15-0ubuntu1~ppa1~22.04.1_arm64.deb
orangepi@orangepi:~$ sudo dpkg -i    \
chromium-browser_101.0.4951.15-0ubuntu1~ppa1~22.04.1_arm64.deb
```

d. After installation, you can see the shortcut of the Chromium browser in the application



e. The display after opening the Chromium browser is as follows



3. 34. 2. Installation method of Ubuntu22.04 Firefox browser

1) Ubuntu22.04 only supports the snap version of the Firefox browser by default. The installation command is:

```
orangepi@orangepi:~$ sudo apt install firefox
```

2) The firefox-esr version browser can be installed by the following methods:

a. First add firefox-esr version PPA source

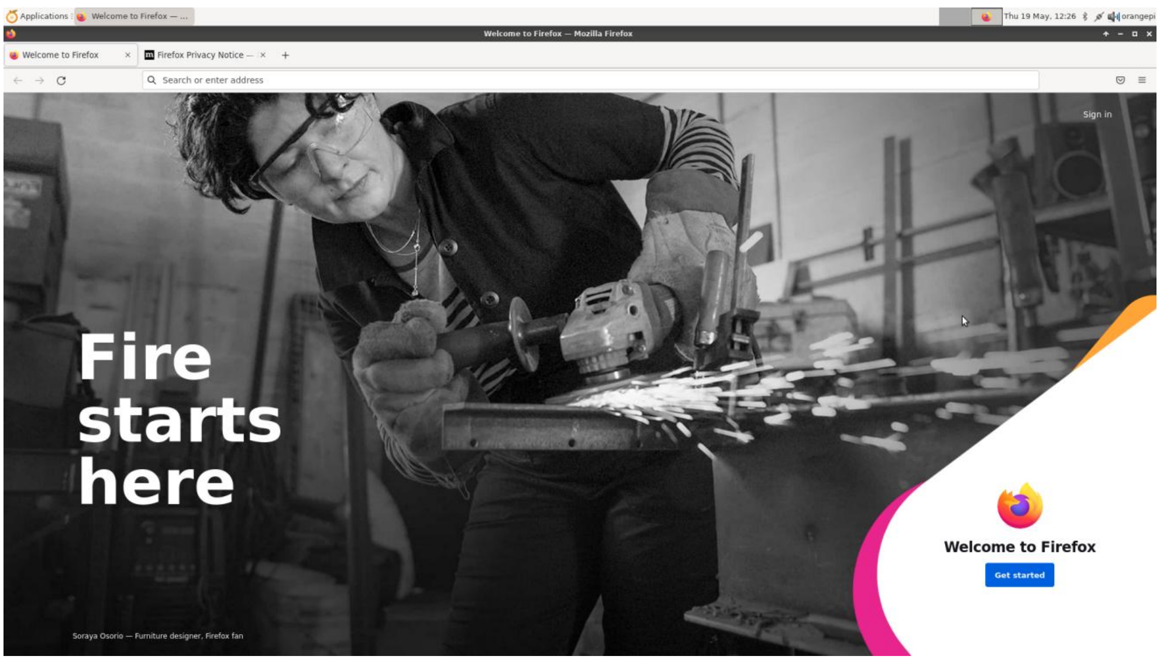
```
orangepi@orangepi:~$ sudo add-apt-repository ppa:mozillateam/ppa
PPA publishes dbgsym, you may need to include 'main/debug' component
Repository: 'deb https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/ jammy main'
Description:
Mozilla Team's Firefox 91 ESR and Thunderbird 91 or 78 stable builds
More info: https://launchpad.net/~mozillateam/+archive/ubuntu/ppa
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.        <--- Please enter here to confirm
```

b. Then install firefox-esr using the command below

```
orangepi@orangepi:~$ sudo apt install firefox-esr
```

c. After installation, you can see the shortcut of firefox-esr browser in the application

d. The display after opening the firefox-esr browser is as follows



3. 35. Partial programming language test supported by Linux system

3. 35. 1. Debian Bullseye System

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

a. The version of a.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. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

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

2) Debian Bullseye has Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

b. Write the **hello_world.py** program in the Python language

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

c. The result of running **hello_world.py** is as follows

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

3) Debian Bullseye does not install Java compilation tools and runtime environment by default

a. You can install openjdk with the following command, the latest version in

Debian Bullseye is openjdk-17

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

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

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

c. Write the Java version 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. 35. 2. Debian Buster System

4) Debian Buster is installed with gcc compilation toolchain by default, which can directly compile C language programs in the Linux system of the development board

a. The version of a.gcc is shown below

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 8.3.0-6) 8.3.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
	printf("Hello World!\n");

```

```
        return 0;
}
```

c. Then compile and run **hello_world.c**

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

5) Debian Buster is installed with Python2 and Python3 by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
orangepi@orangepi:~$ python3
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

b. Write the **hello_world.py** program in the Python language

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

c. The result of running **hello_world.py** is as follows

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

6) Debian Buster does not install Java compilation tools and runtime environment by default

a. You can install openjdk with the following command, the default version in Debian Buster is openjdk-11

```
orangepi@orangepi:~$ sudo apt install -y openjdk-11-jdk
```

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

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

```
openjdk 11.0.13 2021-10-19
OpenJDK Runtime Environment (build 11.0.13+8-post-Debian-1deb10u1)
OpenJDK 64-Bit Server VM (build 11.0.13+8-post-Debian-1deb10u1, mixed mode)
```

c. Write the Java version of **hello_world.java**

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

d. Then compile and run **hello_world.java**

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

3. 35. 3. Ubuntu Jammy system

1) Ubuntu Jammy is installed with gcc compilation toolchain by default, which can directly compile C language programs in the Linux system of the development board

a. The version of a.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. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
```

```
}
```

c. Then compile and run **hello_world.c**

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

2) Ubuntu Jammy has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.4 (main, Apr  2 2022, 09:04:19) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

b. Write the **hello_world.py** program in the Python language

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

c. The result of running **hello_world.py** is as follows

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

3) Ubuntu Jammy does not install Java compilation tools and runtime environment by default

a. You can use the following command to install openjdk-18

```
orangepi@orangepi:~$ sudo apt install -y openjdk-18-jdk
```

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

```
orangepi@orangepi:~$ java --version
openjdk 18-ea 2022-03-22
OpenJDK Runtime Environment (build 18-ea+36-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 18-ea+36-Ubuntu-1, mixed mode, sharing)
```

c. Write the Java version 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. 35. 4. Ubuntu Focal system

4) Ubuntu Focal is installed with gcc compilation toolchain by default, which can directly compile C language programs in the Linux system of the development board

a. The version of a.gcc is shown below

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**

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

5) Ubuntu Focal has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
```

```
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

b. Write the **hello_world.py** program in the 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!
```

6) Ubuntu Focal does not install Java compilation tools and runtime environment by default

a. You can install openjdk-17 using the following command

```
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 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!
```

3.36. The method of remotely logging in to the Linux system desktop

Compared with VNC, it is more recommended to use NoMachine to remotely log in to the Linux system desktop

3.36.1. Remote login using NoMachine

Please make sure that the Ubuntu or Debian system installed on the development board is the desktop version. In addition, NoMachine also provides detailed usage documentation. It is strongly recommended to read through this documentation to familiarize yourself with the usage of NoMachine. The documentation links are as follows:

https://knowledgebase.nomachine.com/DT10R00166

NoMachine supports Windows, Mac, Linux, iOS and Android platforms, so we can remotely log in to control the Orange Pi development board through NoMachine on a variety of devices. The following demonstrates how to remotely log in to the Linux system desktop of the Orange Pi development board through NoMachine in Windows. For installation methods on other platforms, please refer to the official documentation of NoMachine.

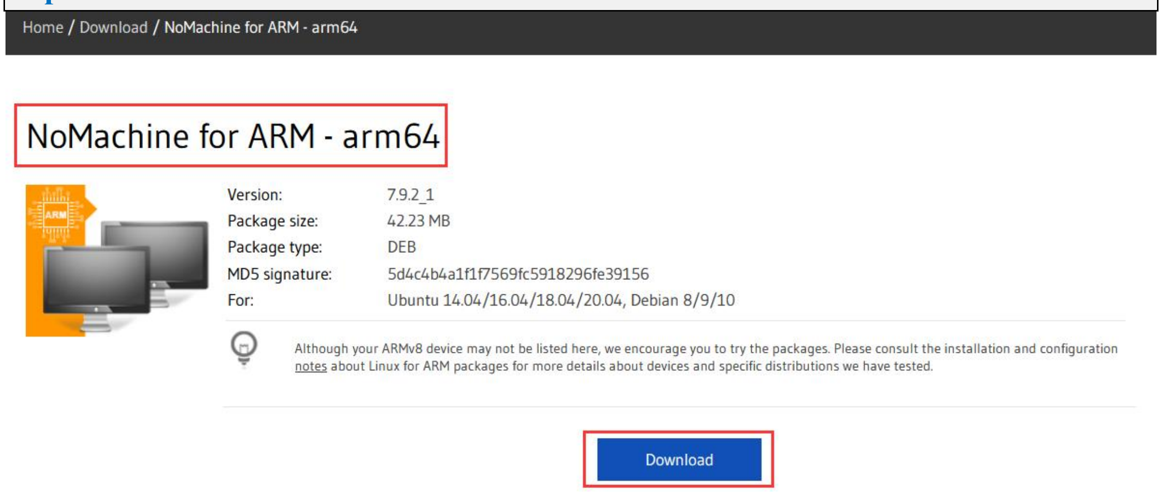
Before operation, please make sure that the Windwos computer and the development board are in the same local area network, and can log in to the Ubuntu or Debian system of the development board normally through ssh.

1) First download the installation package of the Linux arm64 deb version of the NoMachine software, and then install it into the Linux system of the development board

a. Since both RK3399 and RK3399-T are SOCs of ARMv8 architecture, the system we use is Ubuntu or Debian, so here we need to download the **NoMachine for ARM ARMv8 DEB** installation package. The download link is as follows:

Note that this download link may change, please look for the Armv8/Arm64 version of the deb package.

https://www.nomachine.com/download/download&id=116&s=ARM



b. In addition, the **NoMachine** installation package can also be downloaded in the official tool

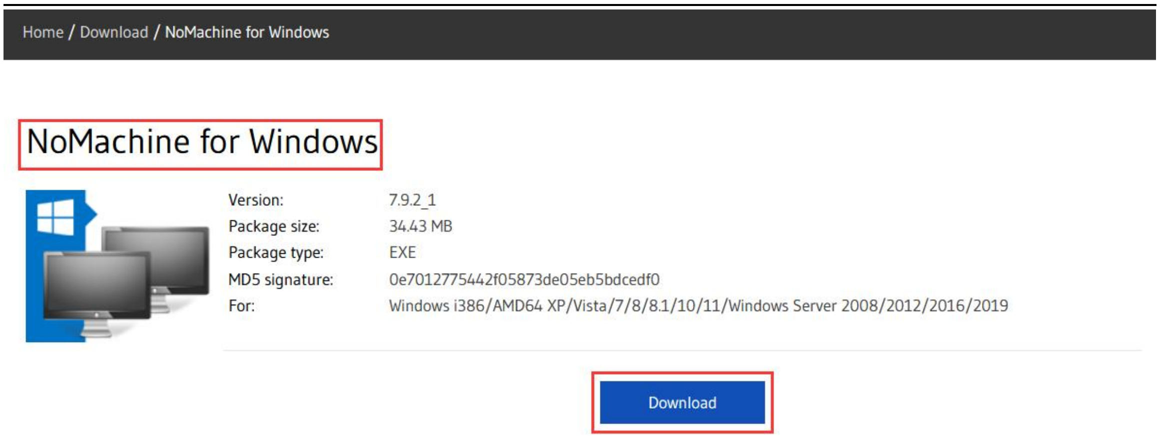
c. Then upload the downloaded **nomachine_7.9.2_1_arm64.deb** to the Linux system of the development board

d. Then use the following command to install **NoMachine** in the Linux system of the development board

```
orangepi@orangepi:~$ sudo dpkg -i nomachine_7.9.2_1_arm64.deb
[sudo] password for orangepi:
Selecting previously unselected package nomachine.
(Reading database ... 182635 files and directories currently installed.)
Preparing to unpack nomachine_7.9.2_1_arm64.deb ...
Unpacking nomachine (7.9.2-1) ...
Setting up nomachine (7.9.2-1) ...
NX> 700 Starting install at: Sun Apr 17 10:52:07 2022.
NX> 700 Installing: nxclient version: 7.9.2.
NX> 700 Using installation profile: Debian.
NX> 700 Install log is: /usr/NX/var/log/nxinstall.log.
NX> 700 Compiling the USB module.
NX> 700 Installing: nxplayer version: 7.9.2.
NX> 700 Using installation profile: Debian.
NX> 700 Install log is: /usr/NX/var/log/nxinstall.log.
NX> 700 To connect the remote printer to the local desktop,
NX> 700 the user account must be a member of the CUPS System Group: lpadmin.
NX> 700 Installing: nxnode version: 7.9.2.
NX> 700 Using installation profile: Debian.
NX> 700 Install log is: /usr/NX/var/log/nxinstall.log.
NX> 700 Creating configuration in: /usr/NX/etc/node.cfg.
NX> 700 Installing: nxserver version: 7.9.2.
NX> 700 Using installation profile: Debian.
NX> 700 Install log is: /usr/NX/var/log/nxinstall.log.
NX> 700 Creating configuration in: /usr/NX/etc/server.cfg.
NX> 700 Install completed at: Sun Apr 17 10:53:00 2022.
NX> 700 NoMachine was configured to run the following services:
NX> 700 NX service on port: 4000
```

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

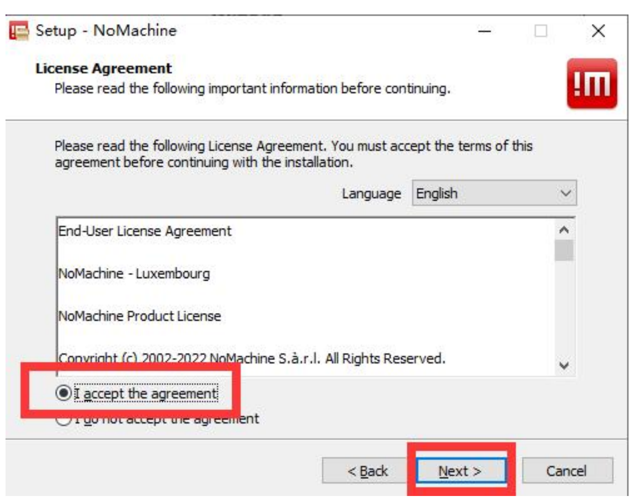
https://www.nomachine.com/download/download&id=8



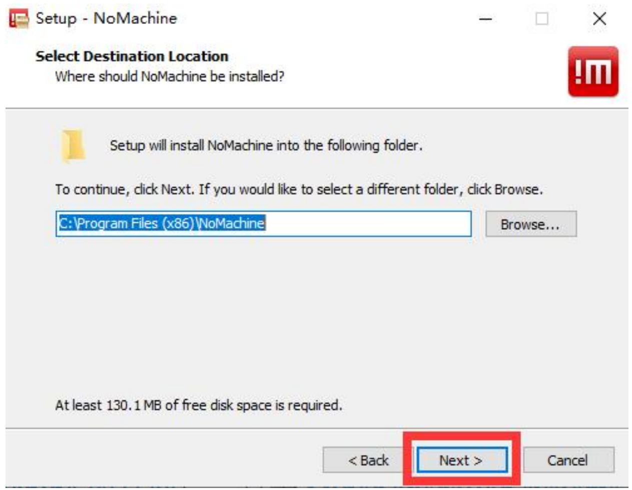
3) Then install NoMachine in Windows

a. Double-click NoMachine in Windows to start the installation of NoMachine, and then select **Next**

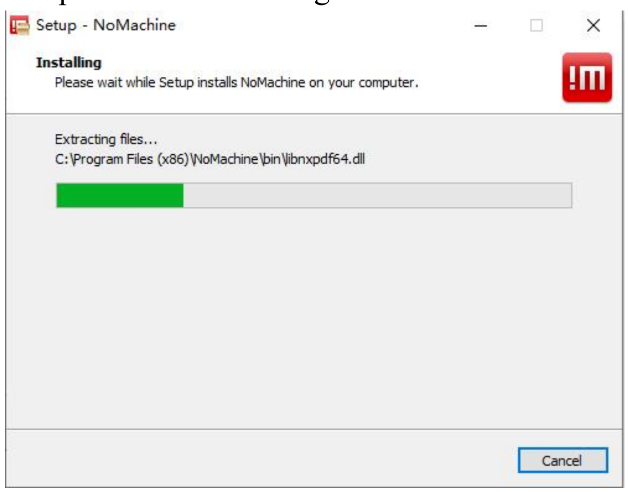
b. Then select **I accept the agreement**, and then select **Next**



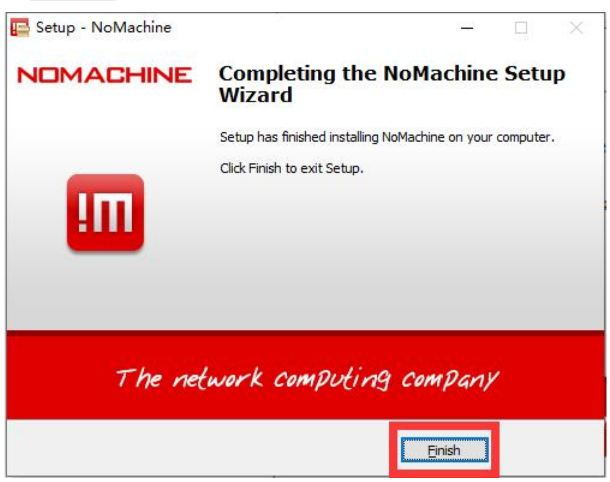
c. Then click **Next**



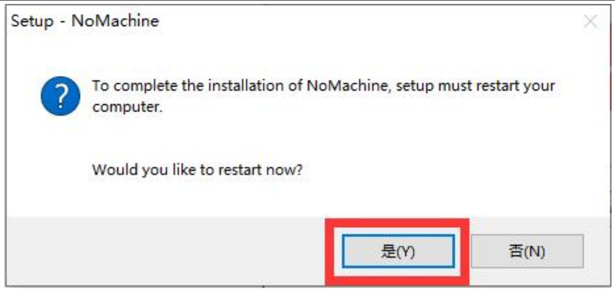
d. The installation process will then begin



e. After the installation is complete, the display is as shown in the figure below, and then click **Finish**.

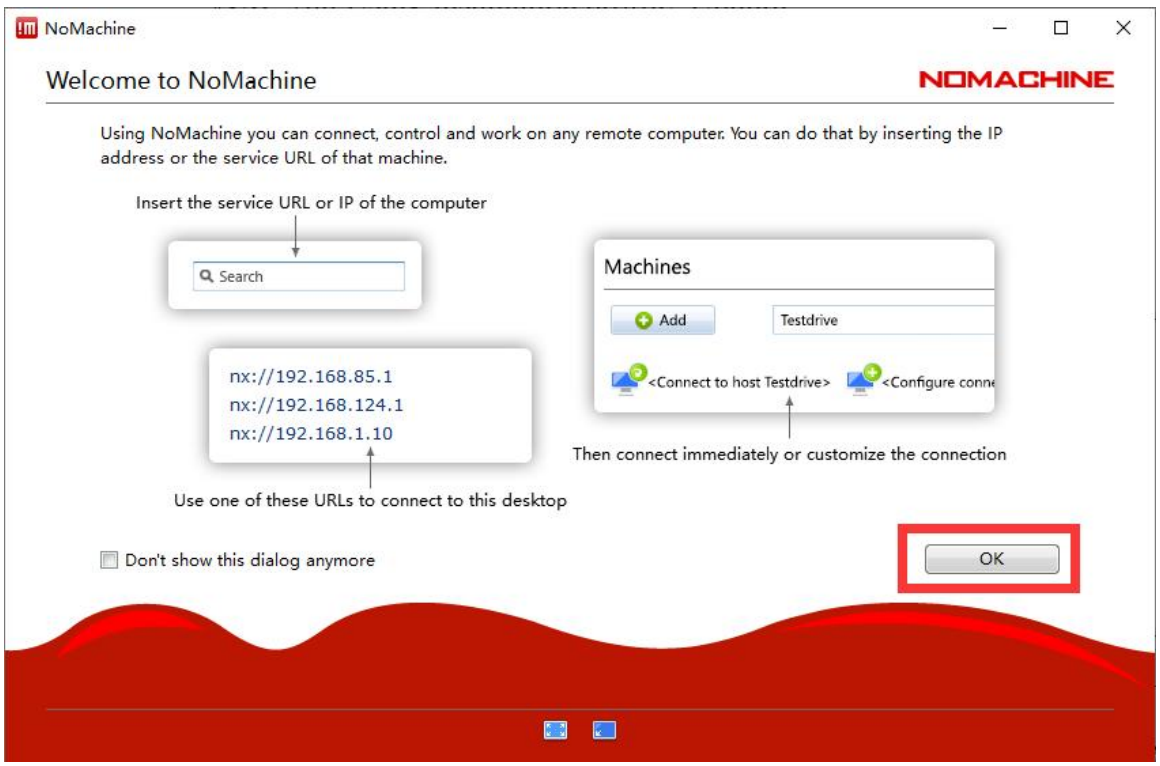


f. Then NoMachine will prompt you to restart to complete the installation, here we choose Yes (Y) to restart the computer

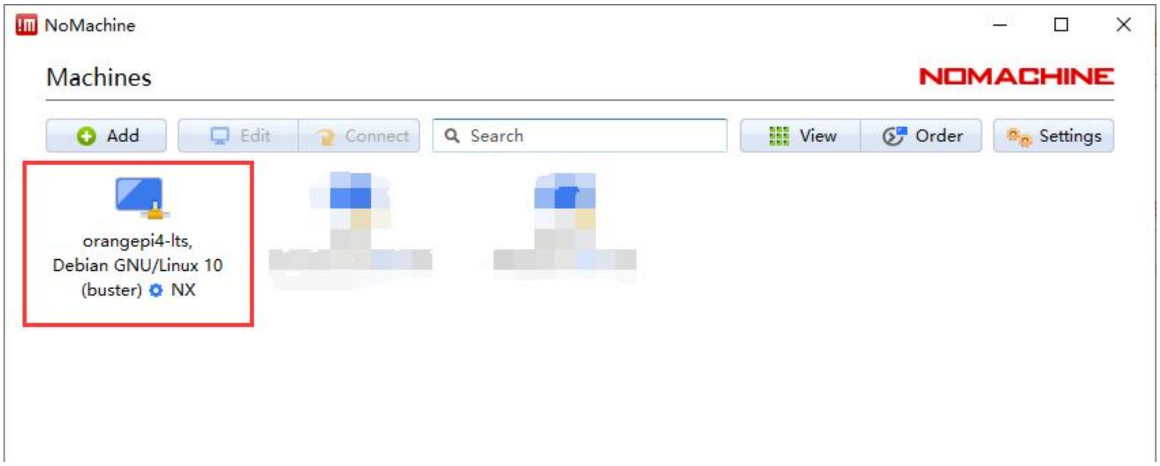


4) Then open **NoMachine** in Window double click

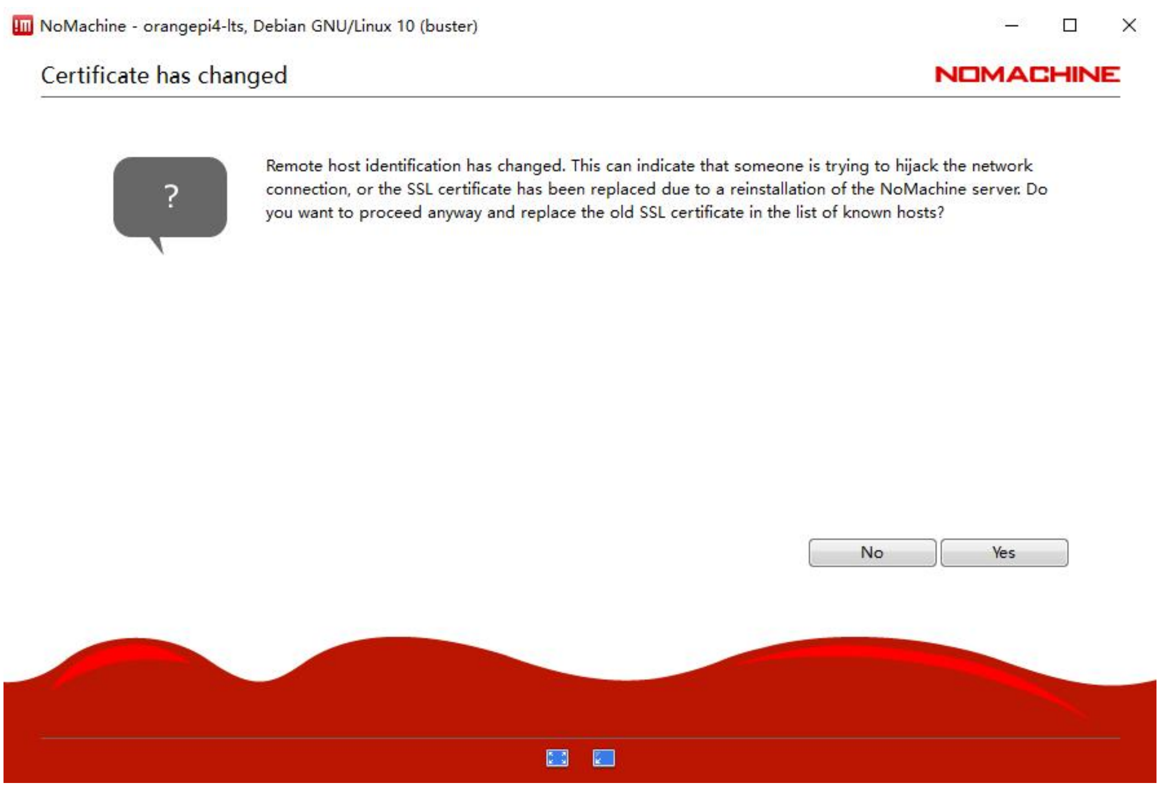
5) Then click **OK**



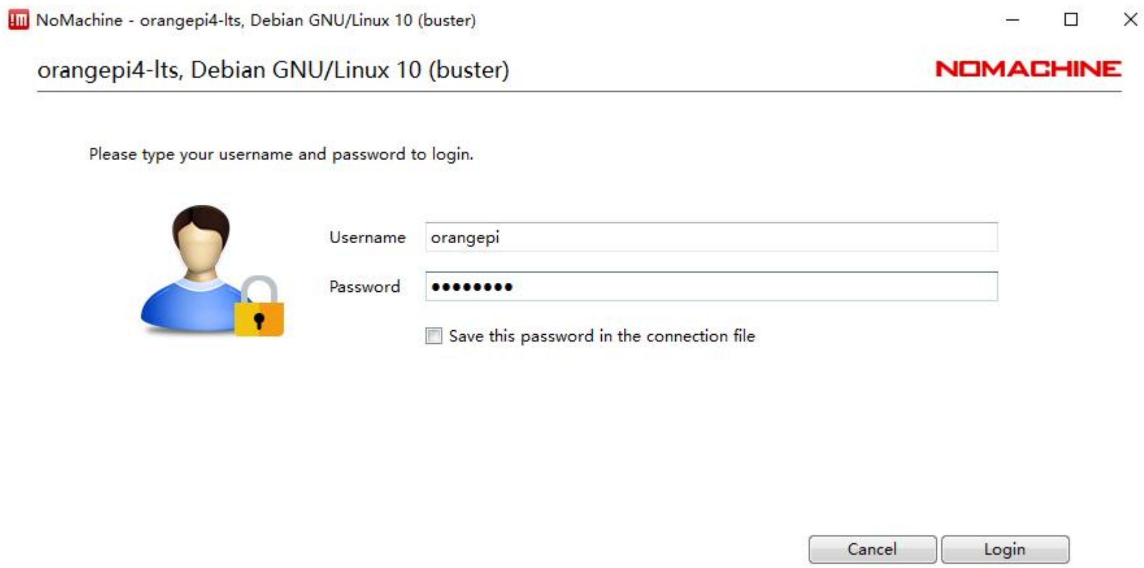
6) After NoMachine is started, it will automatically scan other devices with NoMachine installed in the local area network. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices, and then click the position shown in the red box in the figure below. Start to log in to the Linux system desktop of the development board



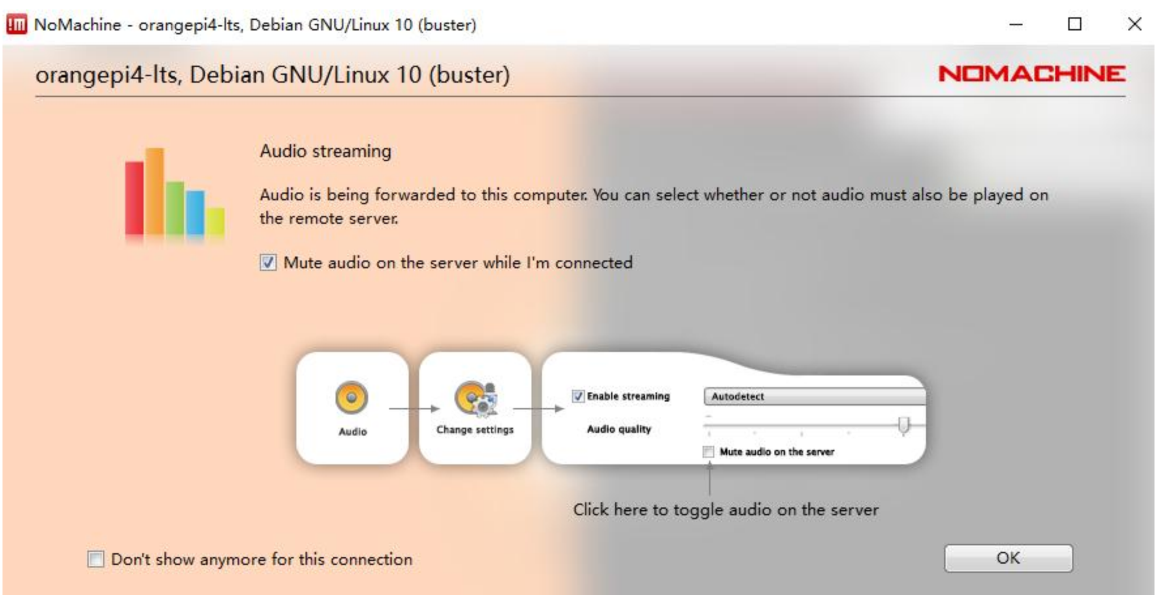
7) Then click **Yes**



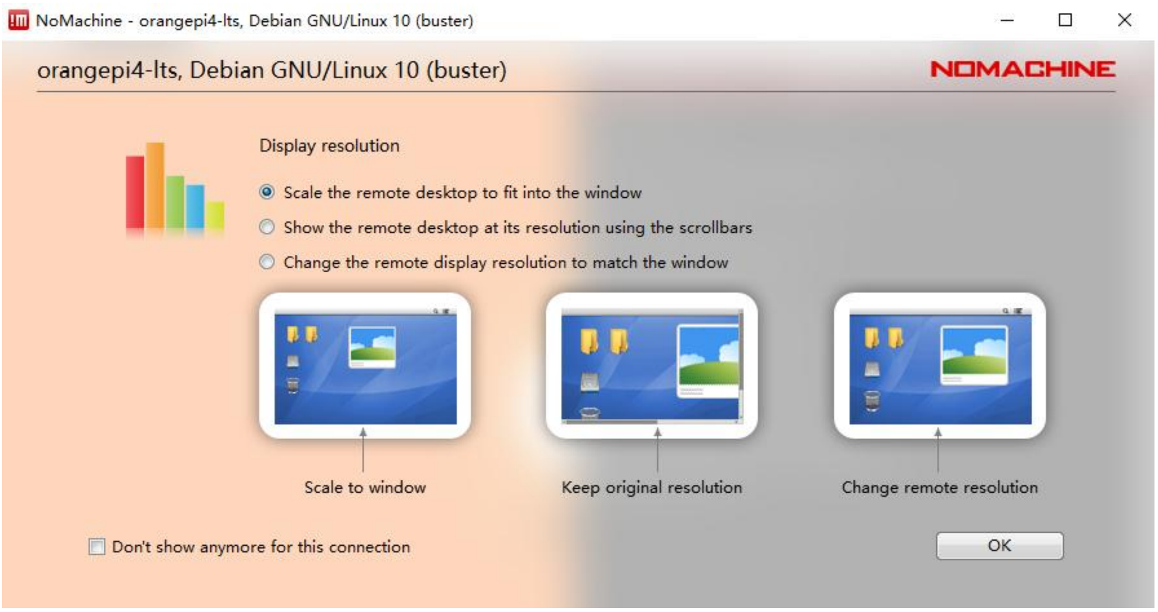
8) Then enter the user name **orangepi** and password **orangepi** of the Linux system of the development board in the corresponding position in the figure below, and then click **Login** to start logging in

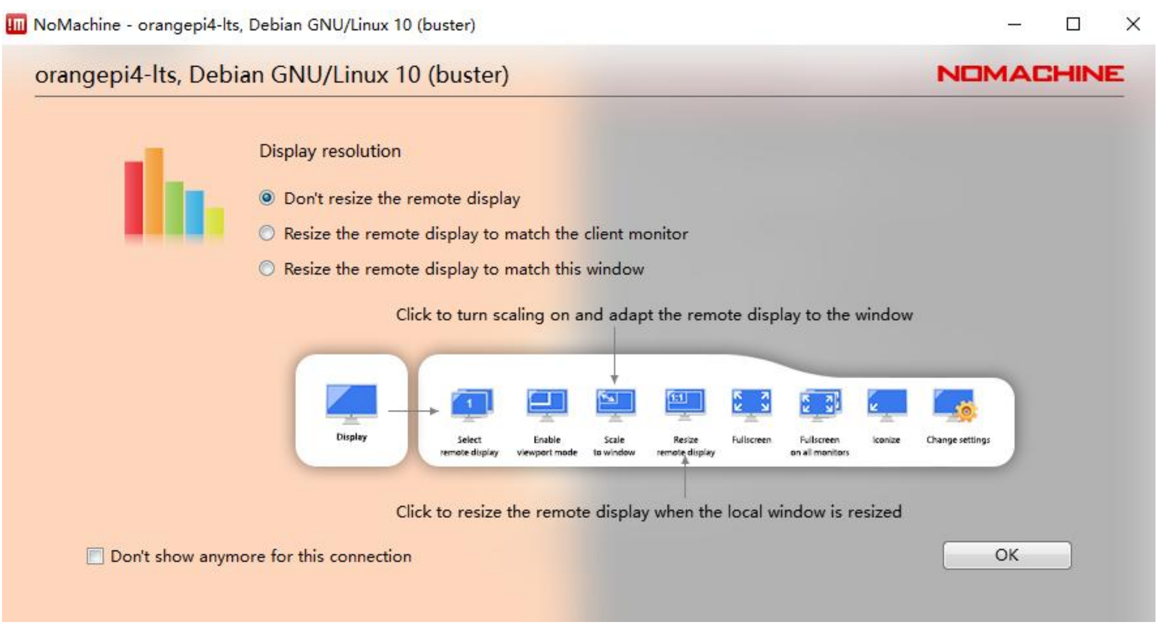


9) Then click OK



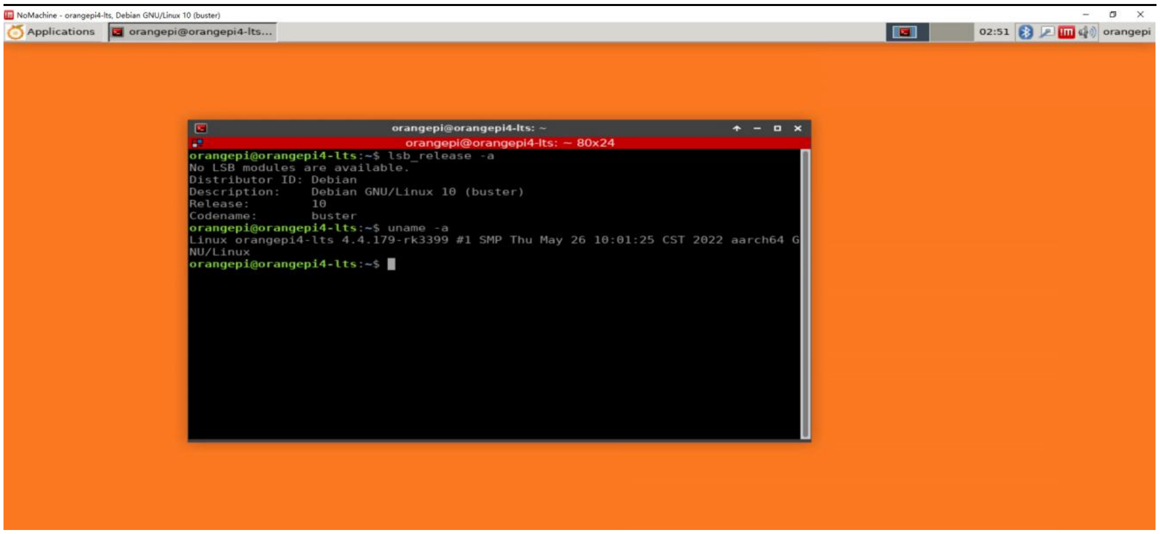
10) Then you can set the display resolution, select it according to your needs, and then click OK





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

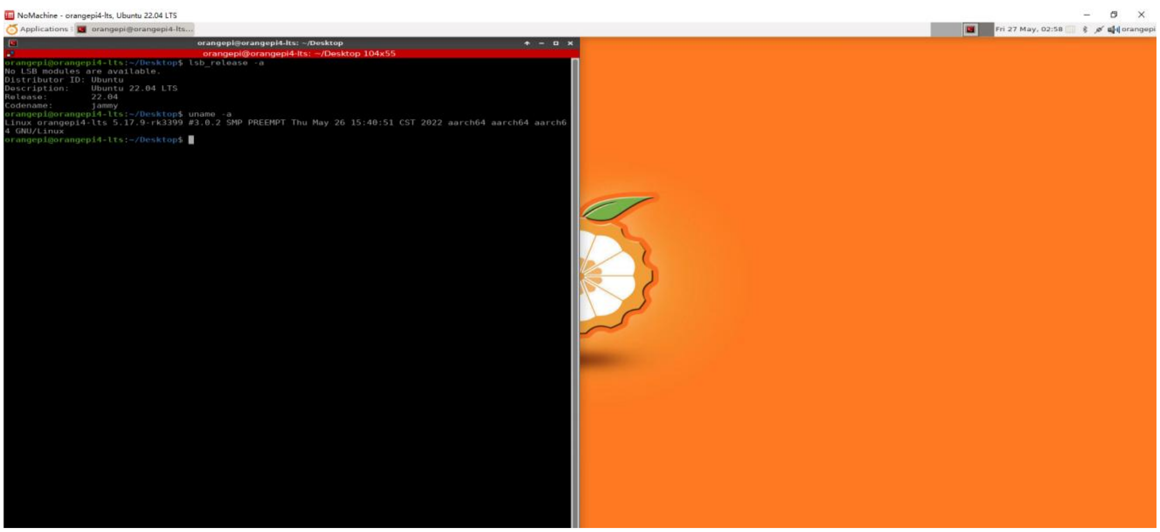
a. Debian Buster



b. **Debian Bullseye Linux5.10 The system is currently unavailable for testing**

c. **Ubuntu Focal Linux5.10 The system is currently unavailable for testing**

d. Ubuntu Jammy



3. 37. How to install Docker

The official installation documentation link provided by Docker is as follows:

Debian system: https://docs.docker.com/engine/install/debian/
Ubuntu system: https://docs.docker.com/engine/install/ubuntu/

1) The old version of the Docker installation package is called docker, docker.io or docker-engine. If these packages are installed, you need to uninstall them first. The

command is as follows:

```
orangepi@orangepi:~$ sudo apt-get remove -y docker docker-engine docker.io \
containerd runc
```

2) Then add the official docker software repository

a. The commands used by the Debian system are as follows

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get install -y ca-certificates curl gnupg lsb-release
orangepi@orangepi:~$ curl -fsSL https://download.docker.com/linux/debian/gpg | \
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
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
```

b. The command used by the Ubuntu system is as follows

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get install -y ca-certificates curl gnupg lsb-release
orangepi@orangepi:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
orangepi@orangepi:~$ echo "deb [arch=$(dpkg --print-architecture) \
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

3) Then install Docker Engine

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y docker-ce docker-ce-cli containerd.io
```

Note: If an error is reported after Debian Buster is installed, please enter the following command to solve it:

```
orangepi@orangepi:~$ echo 1 | update-alternatives --config iptables > /dev/null
orangepi@orangepi:~$ sudo systemctl restart docker
```

4) Then you can add the current user to the docker user group, so that you can run docker

commands without sudo

```
orangepi@orangepi:~$ sudo usermod -aG docker $USER
```

Note: You need to log out and log in again for the system to take effect, or restart the system.

5) Verify the status of docker

```
orangepi@orangepi:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-08-24 10:29:22 UTC; 26min ago
     Docs: https://docs.docker.com
 Main PID: 3145 (dockerd)
    Tasks: 15
   CGroup: /system.slice/docker.service
           └─3145 /usr/bin/dockerd -H fd://
--containerd=/run/containerd/containerd.sock
```

6) You can use the following command to test docker, if you can run hello-world, it means that docker can be used normally

```
orangepi@orangepi:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
256ab8fe8778: Pull complete
Digest:
sha256:7f0a9f93b4aa3022c3a4c147a449ef11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

7) Method of setting docker warehouse as domestic source

a. Create the **/etc/docker/daemon.json** file and add the following configuration to it

```
orangepi@orangepi:~$ sudo vim /etc/docker/daemon.json
```

```
{
  "registry-mirrors": [
     "https://docker.mirrors.ustc.edu.cn"
  ]
}
```

b. Then enter the following command to restart the docker service (or restart the system)

```
orangepi@orangepi:~$ sudo systemctl restart docker
```

3. 38. Python related instructions

3. 38. 1. How to compile and install Python source code

If the Python version in the Ubuntu or Debian system software repository does not meet the development requirements, and you want to use the latest version of Python, you can use the following method to download the Python source package to compile and install the latest version of Python.

The following demonstration is to compile and install the latest version of Python 3.9. If you want to compile and install other versions of Python, the method is the same (you need to download the source code corresponding to the Python you want to install).

1) First install the dependencies needed to compile Python

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt install -y build-essential zlib1g-dev    \
libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libsqlite3-dev    \
libreadline-dev libffi-dev curl libbz2-dev
```

2) Then download the latest version of Python3.9 source code and unzip it

```
orangepi@orangepi:~$ wget    \
https://www.python.org/ftp/python/3.9.10/Python-3.9.10.tgz
orangepi@orangepi:~$ tar xvf Python-3.9.10.tgz
```

3) Then run the configure command

```
orangepi@orangepi:~$ cd Python-3.9.10
orangepi@orangepi:~$ ./configure --enable-optimizations
```

4) Then compile and install Python3.9, the compilation time will take about half an hour

```
orangepi@orangepi:~$ make -j4
orangepi@orangepi:~$ sudo make altinstall
```

5) After installation, you can use the following command to check the version number of the Python you just installed

```
orangepi@orangepi:~$ python3.9 --version
Python 3.9.10
```

6) Then update pip

```
orangepi@orangepi:~$ /usr/local/bin/python3.9 -m pip install --upgrade pip
```

3. 38. 2. How to replace pip source in Python

The default source used by Linux system pip is the official source of Python, but the speed of accessing the official source of Python in China is very slow, and the installation of Python packages often fails due to network reasons. So when using pip to install the Python library, please remember to replace the pip source.

1) First install **python3-pip**

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3-pip
```

2) The method of permanently replacing the pip source under Linux

a. First create a new **~/.pip** directory, then add the **pip.conf** configuration file, and set the source of pip to Tsinghua source in it

```
orangepi@orangepi:~$ mkdir -p ~/.pip
orangepi@orangepi:~$ cat <<EOF > ~/.pip/pip.conf
[global]
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
EOF
```

b. Then using pip3 to install the Python library will be very fast

3) The method of temporarily replacing the pip source under Linux, where

<packagename> needs to be replaced with a specific package name

```
orangepi@orangepi:~$ pip3 install <packagename> -i \
https://pypi.tuna.tsinghua.edu.cn/simple --trusted-host pypi.tuna.tsinghua.edu.cn
```

3. 39. Installation method of OpenCV

3. 39. 1. Using apt to install OpenCV

1) The installation command is as follows

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get install -y libopencv-dev python3-opencv
```

2) Then use the following command to print the version number of OpenCV and the output is normal, indicating that the installation of OpenCV is successful

a. The version of OpenCV in Ubuntu22.04 is as follows:

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.5.4
```

b. The version of OpenCV in Ubuntu20.04 is as follows:

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.2.0
```

c. The version of OpenCV in Debian10 is as follows:

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
3.2.0
```

d. The version of OpenCV in Debian11 is as follows:

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.5.1
```

3. 40. How to install Home Assistant

Note that only the method of installing Home Assistant in Ubuntu or Debian system is provided here. For detailed usage of Home Assistant, please refer to the official documentation or corresponding books.

3. 40. 1. Install via docker

1) Please install docker first, and make sure that docker can run normally. For the installation steps of Docker, please refer to the instructions in the section How to Install

Docker.

2) Then you can search for the docker image of Home Assistant

```
orangepi@orangepi:~$ docker search homeassistant
```

3) Then use the following command to download the docker image of Home Assistant to the local, the image size is about 1GB, the download time will be longer, please wait patiently for the download to complete

```
orangepi@orangepi:~$ docker pull homeassistant/home-assistant
Using default tag: latest
latest: Pulling from homeassistant/home-assistant
be307f383ecc: Downloading
5fbc4c07ac88: Download complete
...... (omit part of the output)
3cc6a1510c9f: Pull complete
7a4e4d5b979f: Pull complete
Digest:
sha256:81d381f5008c082a37da97d8b08dd8b358dae7ecf49e62ce3ef1eeaefc4381bb
Status: Downloaded newer image for homeassistant/home-assistant:latest
docker.io/homeassistant/home-assistant:latest
```

If the network connected to the development board is relatively fast, but the download of the docker image is particularly slow, please check if you forgot to configure the download address of the docker image as a domestic source. The configuration method is described in the section on how to install Docker.

4) Then you can use the following command to view the docker image of Home Assistant just downloaded

```
orangepi@orangepi:~$ docker images homeassistant/home-assistant
REPOSITORY                     TAG      IMAGE ID       CREATED        SIZE
homeassistant/home-assistant   latest   bfa0ab9e1cf5   2 months ago   1.17GB
```

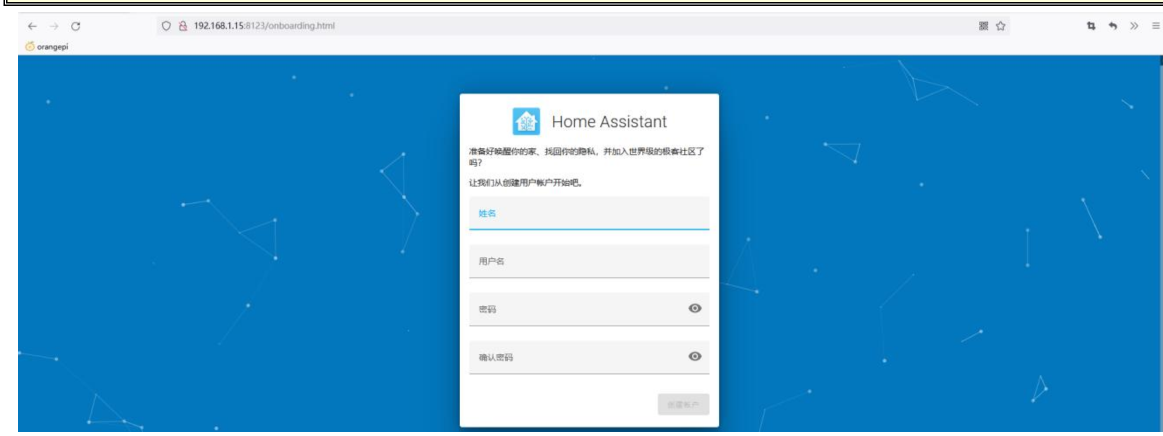
5) At this point, you can run the docker container of Home Assistant

```
orangepi@orangepi:~$ docker run -d \
  --name homeassistant \
```

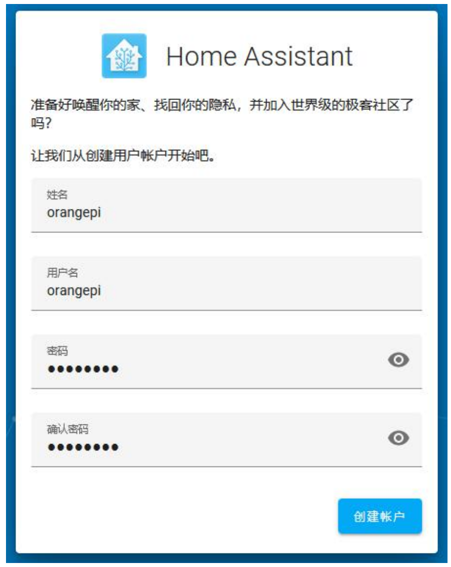
```
--privileged \
--restart=unless-stopped \
-e TZ=Asia/Shanghai \
-v /home/orangepi/home-assistant:/config \
--network=host \
homeassistant/home-assistant:latest
```

6) Then enter **[IP address of the development board: 8123]** in the browser to see the interface of Home Assistant

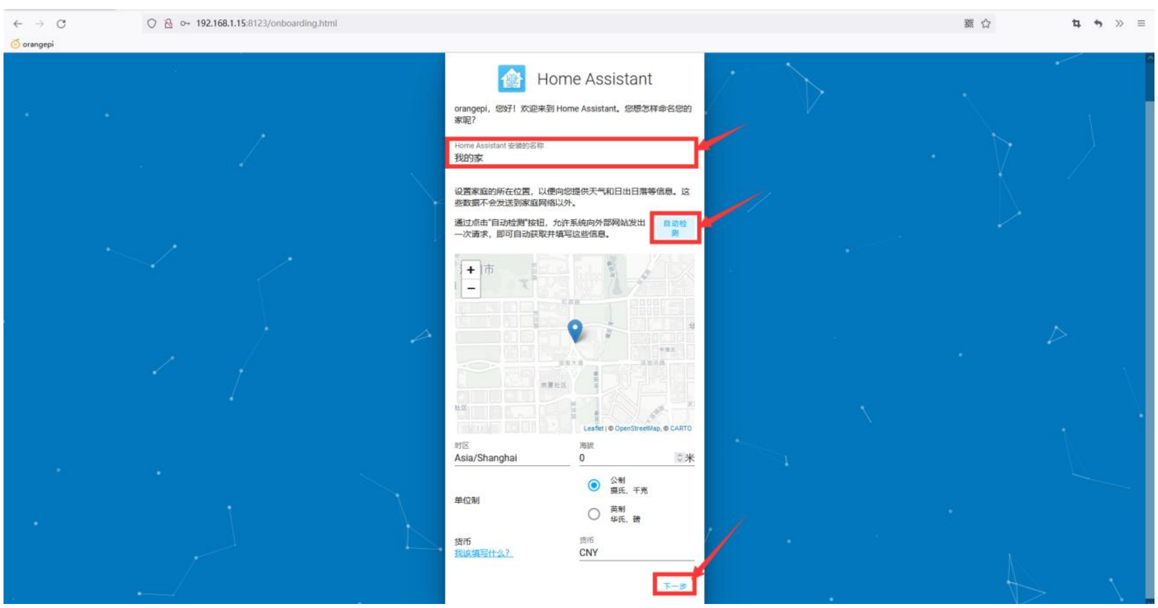
The startup of the Home Assistant container takes a while, if the interface below does not appear normally, please wait a few seconds before refreshing. If the following interface is not displayed after waiting for more than a minute, it means that there is a problem with the Home Assistant installation. At this time, you need to check whether there is a problem with the previous installation and setting process.



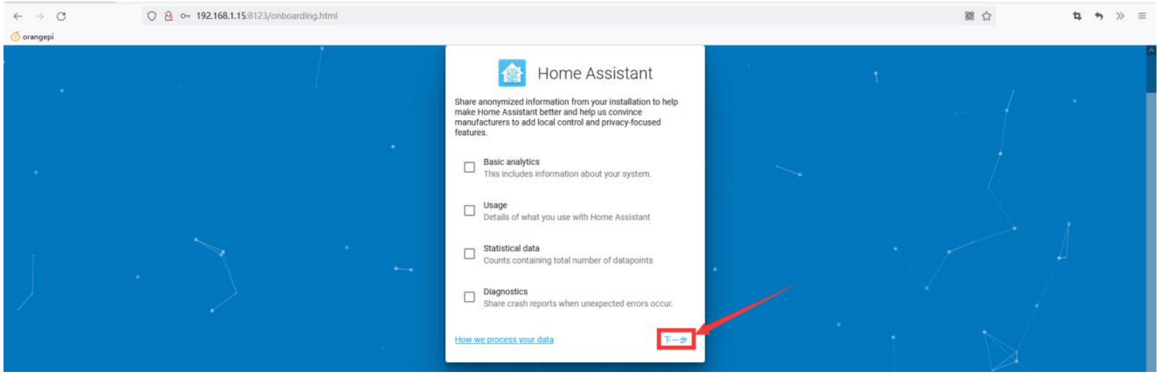
7) Then enter your **name**, **username** and password and click **Create Account**



8) Then follow the interface prompts to set according to your own preferences, and then click Next

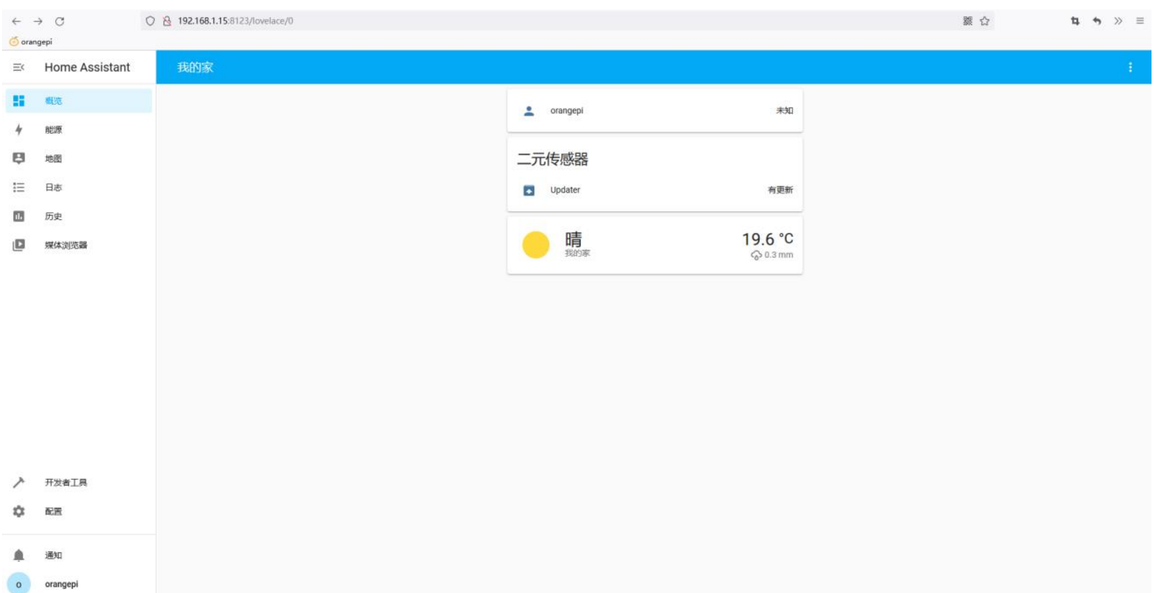


9) Then click Next



10) Then click Finish

11) The main interface finally displayed by Home Assistant is shown in the figure below



12) Ways to stop the Home Assistant container

a. The command to view the docker container is as follows

```
orangepi@orangepi:~$ docker ps -a
```

b. The command to stop the Home Assistant container is as follows

```
orangepi@orangepi:~$ docker stop homeassistant
```

c. The command to delete the Home Assistant container is as follows

```
orangepi@orangepi:~$ docker rm homeassistant
```

3. 40. 2. Install via python

Before installation, please change the source of pip to a domestic source to speed up the installation of Python packages. For the configuration method, see the description in the section How to replace pip source in Python.

1) First install the dependency package

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3 python3-dev python3-venv
python3-pip libffi-dev libssl-dev libjpeg-dev zlib1g-dev autoconf build-essential
libopenjp2-7 libtiff5 libturbojpeg0-dev tzdata
```

2) Then you need to compile and install Python 3.9. For the method, please refer to the section on compiling and installing Python source code.

The default Python version of Debian Bullseye is Python3.9, so there is no need to compile and install.

The default Python version of Ubuntu Jammy is Python3.10, so there is no need to compile and install.

3) Then create a Python virtual environment

```
orangepi@orangepi:~$ sudo mkdir /srv/homeassistant
orangepi@orangepi:~$ sudo chown orangepi:orangepi /srv/homeassistant
orangepi@orangepi:~$ cd /srv/homeassistant
orangepi@orangepi:~$ python3.9 -m venv .
orangepi@orangepi:~$ source bin/activate
(homeassistant) orangepi@orangepi:/srv/homeassistant$
```

The fourth command Ubuntu Jammy needs to be modified to python3.10 -m venv .

4) Then install the required Python packages

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ python3 -m pip install wheel
```

5) Then you can install Home Assistant Core

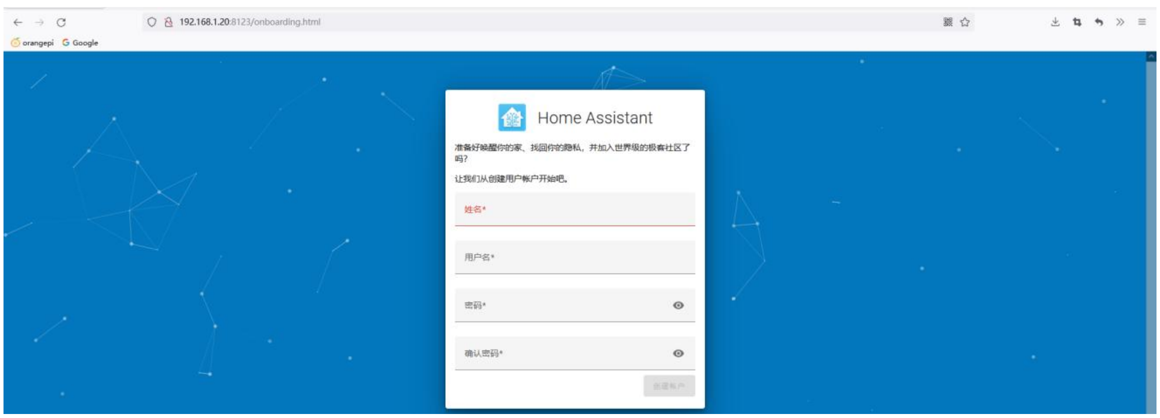
```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ pip3 install homeassistant
```

6) Then enter the following command to run Home Assistant Core

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ hass
```

7) Then enter **[IP address of the development board: 8123]** in the browser to see the interface of Home Assistant

When you run the hass command for the first time, it will download, install and cache some necessary libraries and dependencies. This process may take a few minutes. Note that the interface of Home Assistant cannot be seen in the browser at this time, please wait for a while before refreshing.



3. 41. Installation method of Tensorflow

Note that before installing Tensorflow, please make sure that the Linux system used is Debian Buster. The installation method of Tensorflow demonstrated in this section cannot be guaranteed to work normally on other versions of Linux systems.

3. 41. 1. The method of using script to automatically install Tensorflow

1) First download and install the tensorflow installation script provided by Orange Pi

```
orangepi@orangepi:~$ wget \
https://gitee.com/leeboby/tensorflow/raw/master/install_tensorflow.sh
```

2) Then run the **install_tensorflow.sh** script to start installing tensorflow

```
orangepi@orangepi:~$ sudo bash install_tensorflow.sh
```

3) After the tensorflow installation is completed, the version number of tensorflow will be automatically tested and printed. If you can see the following output at the end, it means that the tensorflow installation is successful

```
##### Start Test Tensorflow #####
Tensorflow version is : 2.4.0
###### End Test Tensorflow ######
```

3. 41. 2. Steps to manually install Tensorflow

1) First use the following command to set the source of pip to Tsinghua source to speed up the download speed of the Python package

```
orangepi@orangepi:~$ mkdir -p ~/.pip
orangepi@orangepi:~$ cat <<EOF > ~/.pip/pip.conf
[global]
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
EOF
```

2) Then install the dependency package

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y python3-pip gfortran   \
libopenblas-dev liblapack-dev libatlas-base-dev libblas-dev   \
libhdf5-dev hdf5-tools libhdf5-dev zlib1g-dev zip libjpeg62-turbo-dev   \
python3-dev pkg-config python3-setuptools python3-wheel
```

3) Then download the **whl** package related to tensorflow

```
orangepi@orangepi:~$ git clone --depth=1 https://gitee.com/leeboby/tensorflow.git
```

4) Then enter the **tensorflow** directory to install the whl package that tensorflow depends on

```
orangepi@orangepi:~$ cd tensorflow
orangepi@orangepi:~/tensorflow$ pip3 install   \
tensorflow/grpcio-1.32.0-cp37-cp37m-linux_aarch64.whl
orangepi@orangepi:~/tensorflow$ pip3 install   \
```

```
tensorflow/numpy-1.19.5-cp37-cp37m-linux_aarch64.whl
orangepi@orangepi:~/tensorflow$ pip3 install \
tensorflow/h5py-2.10.0-cp37-cp37m-linux_aarch64.whl
```

5) Then you can use the following command to install tensorflow

```
orangepi@orangepi:~/tensorflow$ pip3 install \
tensorflow-2.4.0-cp37-none-linux_aarch64.whl
```

6) After installing tensorflow, you can use the following command to print the version number of tensorflow. If the version number **2.4.0** of tensorflow can be printed out normally, it means that the installation of tensorflow is successful

```
orangepi@orangepi:~/tensorflow$ python3 -c \
"import tensorflow; print(tensorflow.__version__)"
2.4.0
```

7) References

```
https://github.com/lhelontra/tensorflow-on-arm
https://tf.kmtea.eu/whl/stable.html
https://www.tensorflow.org
https://repo.rock-chips.com/pypi/simple
```

3. 42. Installation method of OpenMediaVault

OpenMediaVault is a Debian-based NAS operating system.

It can be known from the table below:
Debian10 can only install OpenMediaVault 5.x version;
Debian11 can only install OpenMediaVault 6.x version.

Table 1: openmediavault historical releases

Version	Codename	Base Distro	Status	Date Released
0.2	Ix	Debian 6	EOL	Oct 2011
0.3	Omnious	Debian 6	EOL	Jul 2012
0.4	Fedaykin	Debian 6	EOL	Sep 2012
0.5	Sardoukar	Debian 6	EOL	Aug 2013
1.0	Kralizec	Debian 7	EOL	Sept 2014
2.0	Stoneburner	Debian 7	EOL	Jun 2015
3.0	Erasmus	Debian 8	EOL	Jun 2016
4.0	Arrakis	Debian 9	EOL	Apr 2018
5.0	Usul	Debian 10	Stable	Mar 2020
6.0	Shaitan	Debian 11	In Development	est. Q2/2022

So before installation, please select the version of OpenMediaVault you want to install, and then make sure that the Debian system used by the development board is the matching system.

In addition, OpenMediaVault officially recommends using the server version of the system, so please do not use the desktop version of the system to install OpenMediaVault.

Can I install openmediavault on top a running Debian system? Yes, but it is recommended that the current running OS not to have a desktop environment installed.

3. 42. 1. Install OpenMediaVault 5.x on Debian 10

Note that Debian10 can only install OpenMediaVault 5.x.

1) The official documentation of OpenMediaVault is as follows:

   a. Documentation for version 5.x (**stable version of OpenMediaVault**)

https://openmediavault.readthedocs.io/en/5.x/

2) The official documentation for installing OpenMediaVault in Debian10 is as follows:

https://openmediavault.readthedocs.io/en/5.x/installation/on_debian.html

3) First install the keyring of OpenMediaVault, note that the following commands are executed under the root user

```
root@orangepi:~# apt-get install -y gnupg
root@orangepi:~# wget -O   \
"/etc/apt/trusted.gpg.d/openmediavault-archive-keyring.asc"   \
https://packages.openmediavault.org/public/archive.key
root@orangepi:~# apt-key add   \
"/etc/apt/trusted.gpg.d/openmediavault-archive-keyring.asc"
```

4) Then add the package repository of OpenMediaVault, pay attention to switch to the **root** user and enter the following command, the black font part is a complete command, please copy it directly

```
root@orangepi:~# cat <<EOF > /etc/apt/sources.list.d/openmediavault.list
deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/public usul main
deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/packages usul main
## Uncomment the following line to add software from the proposed repository.
# deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/public usul-proposed main
# deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/packages usul-proposed main
## This software is not part of OpenMediaVault, but is offered by third-party
## developers as a service to OpenMediaVault users.
# deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/public usul partner
# deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/packages usul partner
EOF
```

The Tsinghua source is used above. For related instructions, please refer to the following link

https://mirrors.tuna.tsinghua.edu.cn/help/openmediavault/

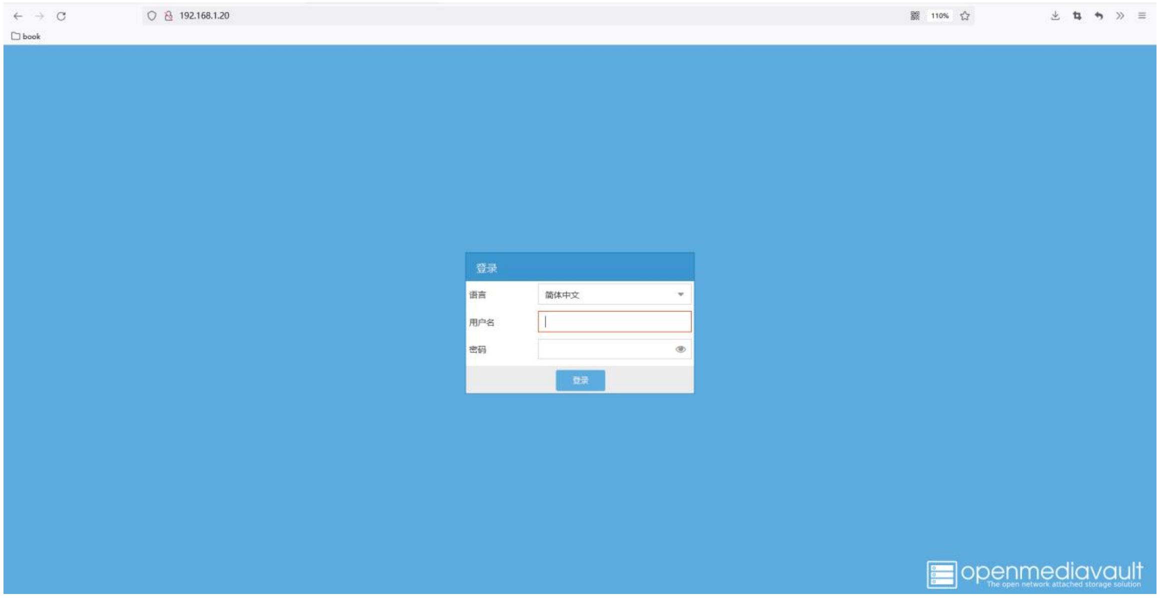
5) Then use the following command to install OpenMediaVault

```
root@orangepi:~# export LANG=C.UTF-8
root@orangepi:~# export DEBIAN_FRONTEND=noninteractive
root@orangepi:~# export APT_LISTCHANGES_FRONTEND=none
root@orangepi:~# apt-get update
root@orangepi:~# apt-get --yes --auto-remove --show-upgraded \
--allow-downgrades --allow-change-held-packages \
--no-install-recommends \
--option DPkg::Options::="--force-confdef" \
--option DPkg::Options::="--force-confold" \
install openmediavault-keyring openmediavault
```

6) Then run the following command. After the operation is completed, enter the IP address of the development board in the browser to open the login page of OpenMediaVault

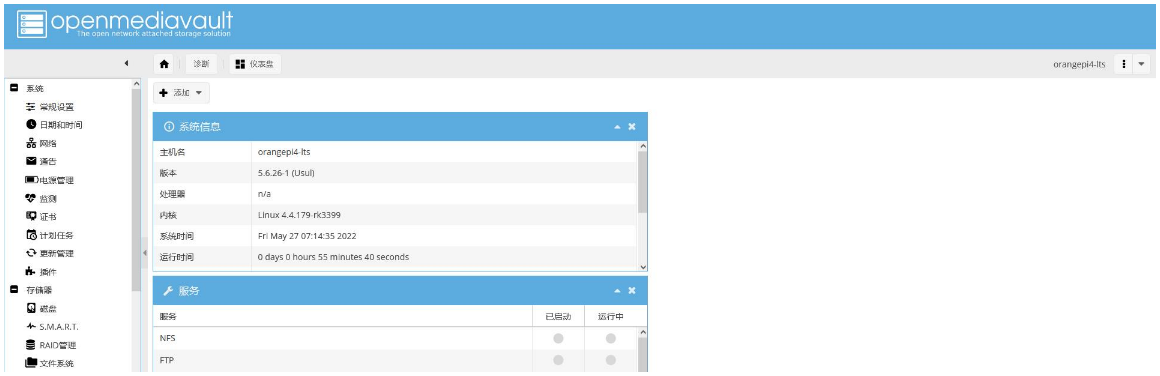
```
root@orangepi:~# omv-confdbadm populate
```

7) The login interface of OpenMediaVault is as follows



8) Then enter the default username **admin** and password **openmediavault**

9) The main interface displayed by OpenMediaVault login is as follows



3. 42. 2. Install OpenMediaVault 6.x on Debian 11

Note that **Debian11** can only install OpenMediaVault 6.x.

1) The official documentation of OpenMediaVault is as follows:

https://openmediavault.readthedocs.io/en/latest/

2) The official documentation for installing OpenMediaVault in Debian looks like this:

https://openmediavault.readthedocs.io/en/latest/installation/on_debian.html

3) First install the keyring of OpenMediaVault, note that the following commands are executed under the **root** user

```
root@orangepi:~# apt-get install -y gnupg
root@orangepi:~# wget -O    \
"/etc/apt/trusted.gpg.d/openmediavault-archive-keyring.asc"    \
https://packages.openmediavault.org/public/archive.key
root@orangepi:~# apt-key add    \
"/etc/apt/trusted.gpg.d/openmediavault-archive-keyring.asc"
```

4) Then add the package repository of OpenMediaVault, pay attention to switch to the **root** user and enter the following command, the black font part is a complete command, please copy it directly

```
root@orangepi:~# cat <<EOF >> /etc/apt/sources.list.d/openmediavault.list
deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/public shaitan main
deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/packages shaitan main
## Uncomment the following line to add software from the proposed repository.
# deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/public shaitan-proposed main
# deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/packages shaitan-proposed main
```

```
## This software is not part of OpenMediaVault, but is offered by third-party
## developers as a service to OpenMediaVault users.
# https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/public shaitan partner
# deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/packages shaitan partner
EOF
```

The Tsinghua source is used above. For related instructions, please refer to the following link

https://mirrors.tuna.tsinghua.edu.cn/help/openmediavault/

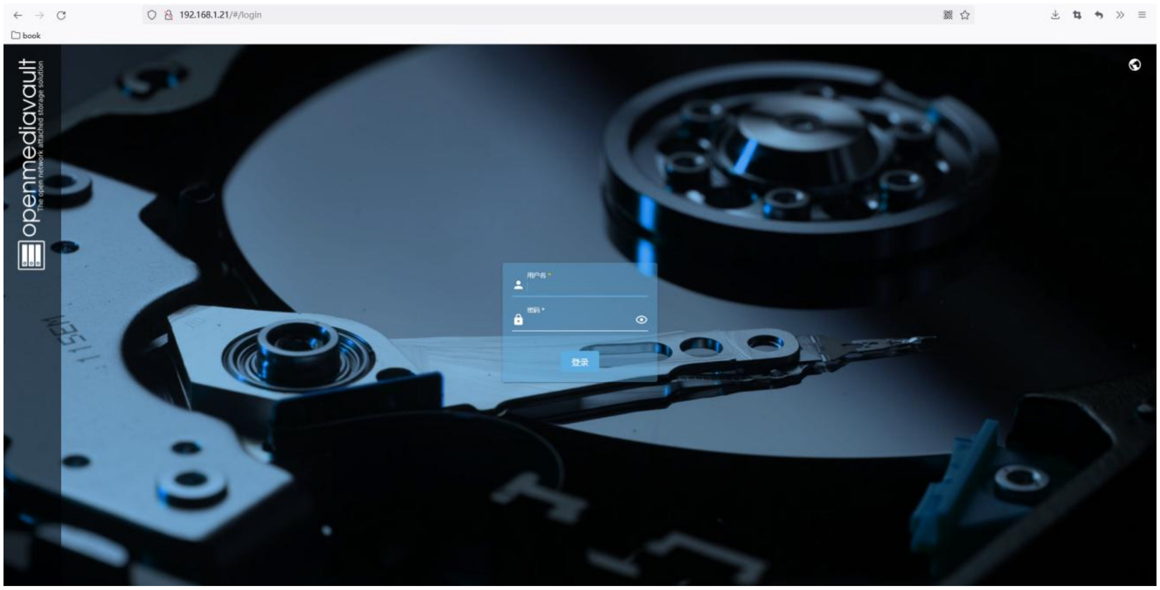
5) Then use the following command to install OpenMediaVault

```
root@orangepi:~# export LANG=C.UTF-8
root@orangepi:~# export DEBIAN_FRONTEND=noninteractive
root@orangepi:~# export APT_LISTCHANGES_FRONTEND=none
root@orangepi:~# apt-get update
root@orangepi:~# apt-get --yes --auto-remove --show-upgraded \
--allow-downgrades --allow-change-held-packages \
--no-install-recommends \
--option DPkg::Options::="--force-confdef" \
--option DPkg::Options::="--force-confold" \
install openmediavault-keyring openmediavault
```

6) Then run the following command. After the operation is complete, enter the IP address of the development board in the browser to open the login page of OpenMediaVault

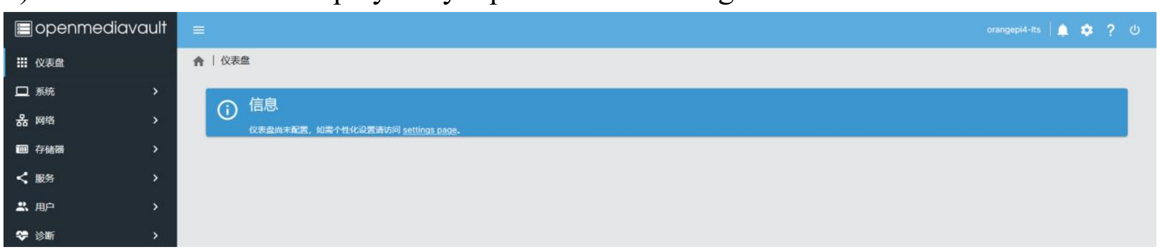
```
root@orangepi:~# omv-confdbadm populate
```

7) The login interface of OpenMediaVault is as follows

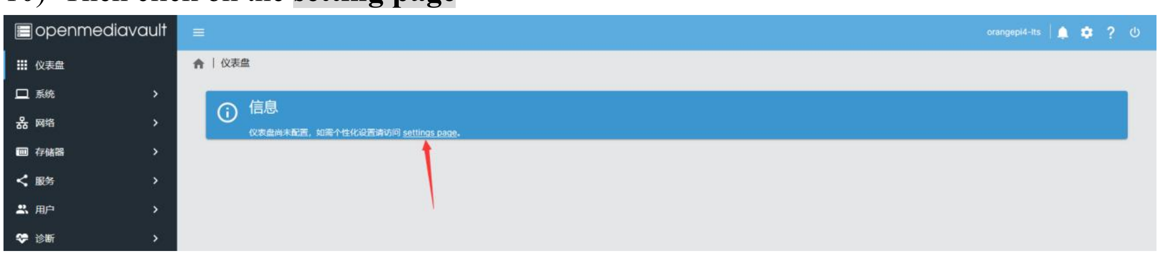


8) Then enter the default username **admin** and password **openmediavault**

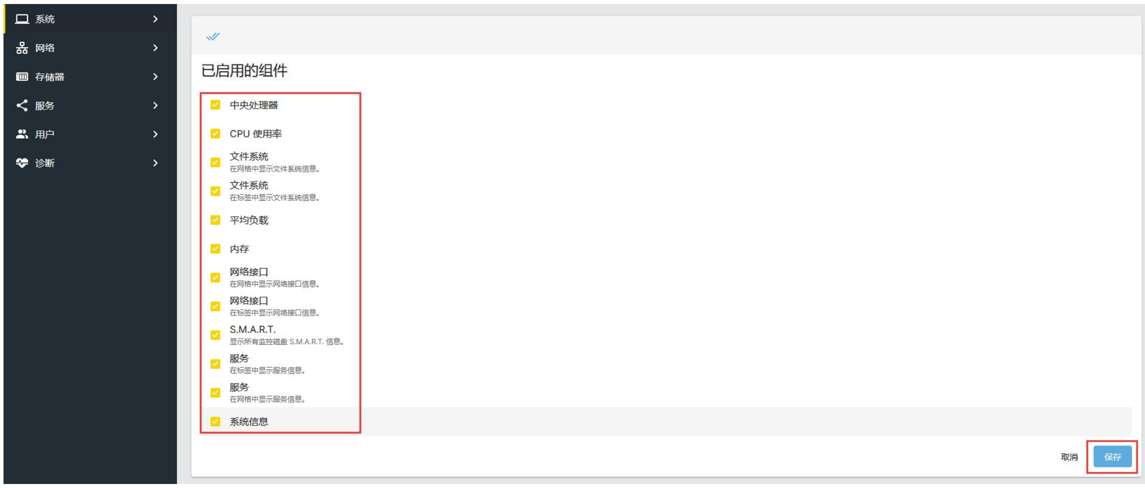
9) The main interface displayed by OpenMediaVault login is as follows



10) Then click on the **setting page**



11) Then select all these components, and then click the save button to **save**



12) Then you can see the system information displayed on the dashboard

13) If the interface displayed by the CPU is not normal, you can open the **diagnosis -> performance statistics -> CPU**, and then click the refresh button in the upper right corner to refresh the

14) How to install OMV plugin

a. First open the following URL

https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/openmediavault-plugin-developers/pool/main/o/openmediavault-omvextrasorg/

b. Then record the file name of the latest plugin package

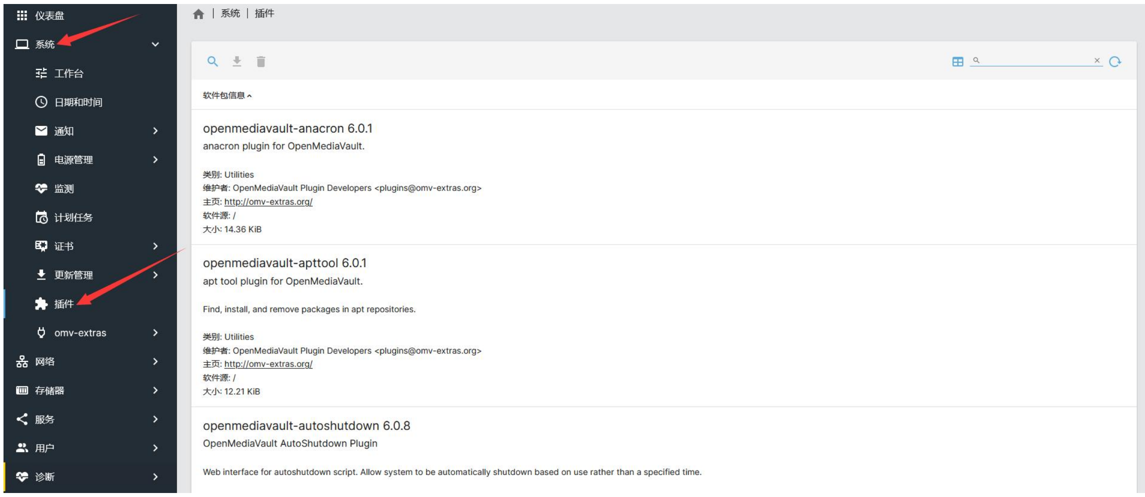
c. Then use the following command to download the plug-in package shown above in the Linux system of the development board (**if the following command cannot be downloaded, the name of the plug-in package may have changed, please replace it with the latest name**)

```
orangepi@orangepi:~$ wget \
https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/openmediavault-plugin-developers/pool/main/o/openmediavault-omvextrasorg/openmediavault-omvextrasorg_6.0.8_all.deb
```

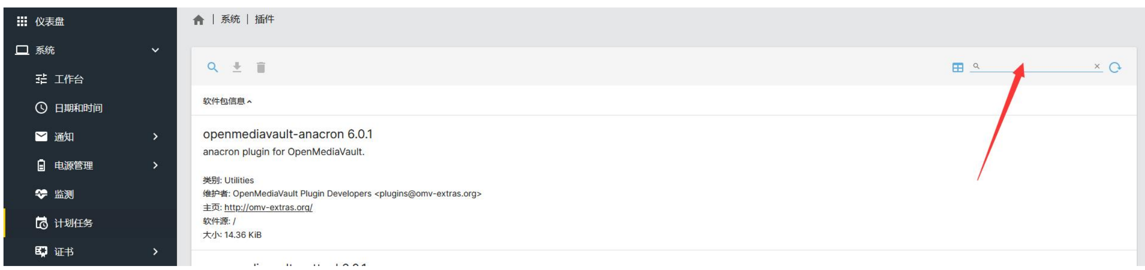
d. Then install the deb package just downloaded in the Linux system of the development board

```
orangepi@orangepi:~$ sudo dpkg -i openmediavault-omvextrasorg_6.0.8_all.deb
```

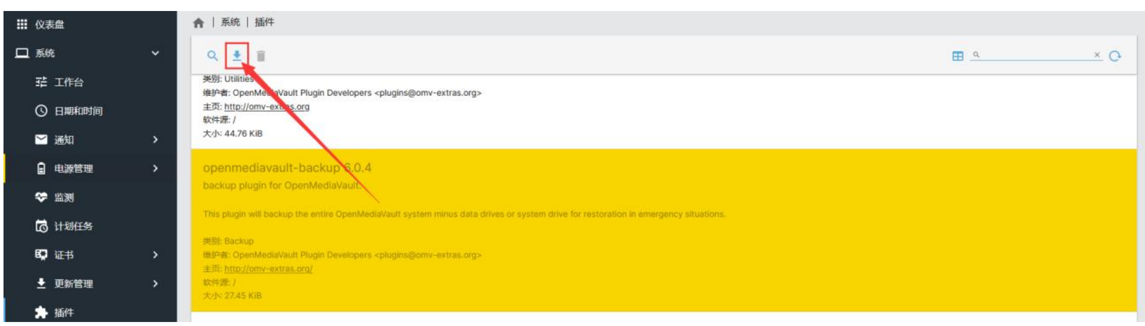
e. Then click System -> Plugins to see the following interface



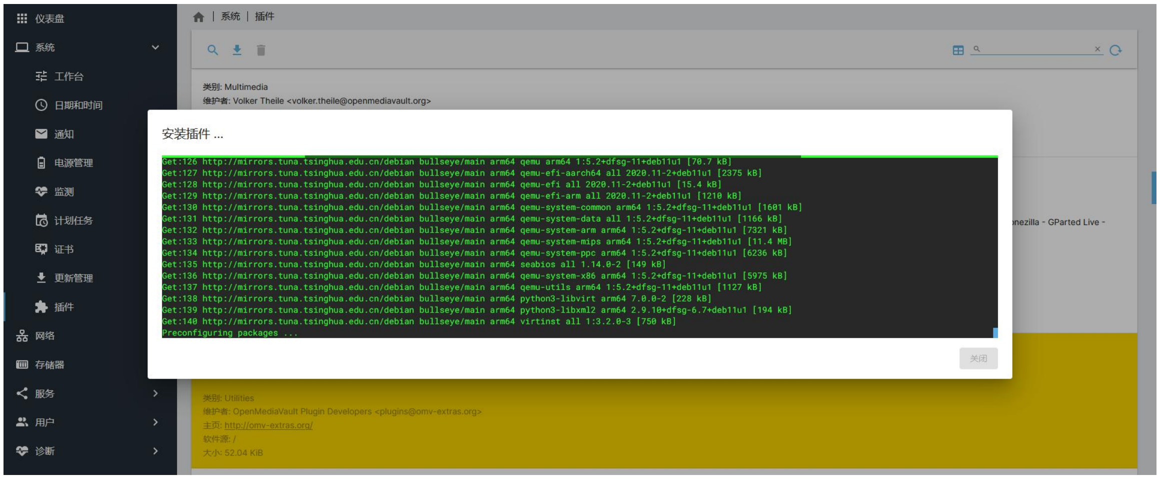
f. Then you can select or search for the plugin you want to install



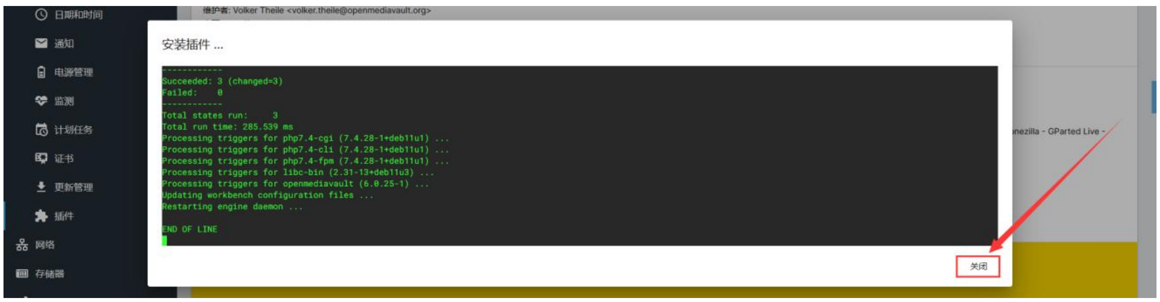
g. Then click the button shown in the figure below to start installing the selected plug-in



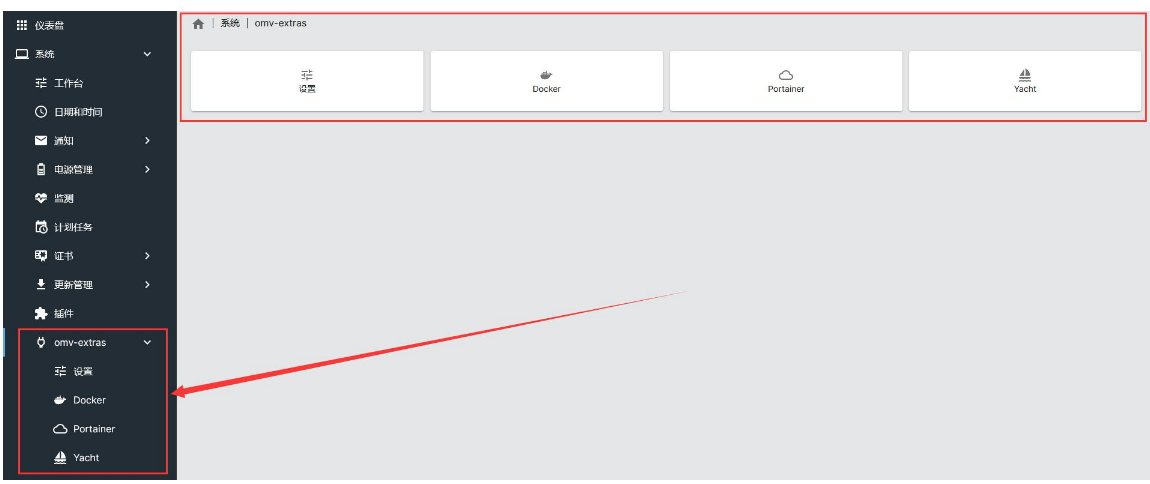
h. The plugin installation process is as follows



i. After the installation is complete, click Close



15) After the plugin package is installed, there will be an **omv-extras** option on the right side of the web interface. **Docker**, **Portainer** and **Yacht** can be installed in omv-extras



16) Before installing docker, please replace the software source of docker. First, use the following command to open **omvextras.list**, and then replace the content in the blue font part. Finally, please remember to use **sudo apt-get update** to update the package index cache of the Linux system

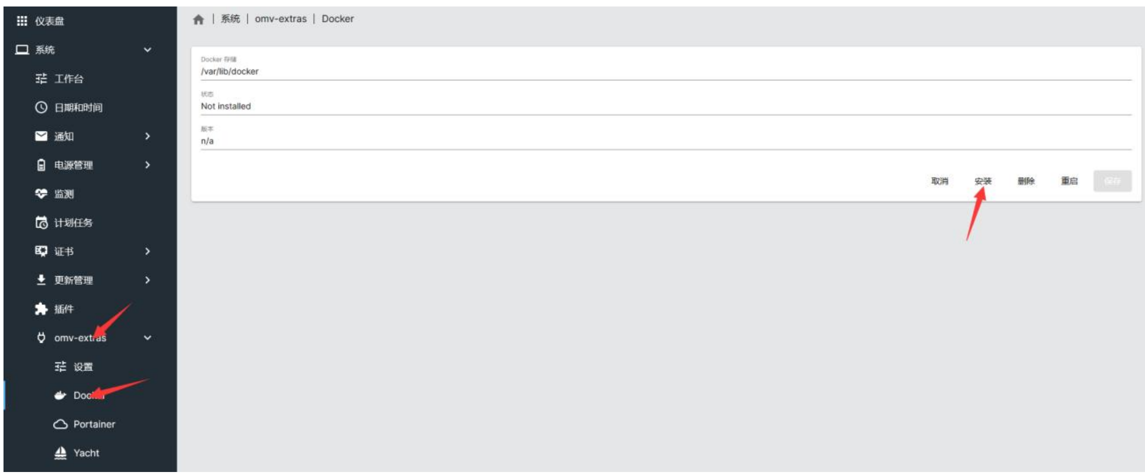
```
orangepi@orangepi:~$ sudo vim /etc/apt/sources.list.d/omvextras.list
deb https://mirrors.tuna.tsinghua.edu.cn/OpenMediaVault/openmediavault-plugin-developers shaitan main
deb [arch=arm64] https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/debian bullseye stable
orangepi@orangepi:~$ sudo apt-get update
```

17) The way to install Docker in OMV is as follows

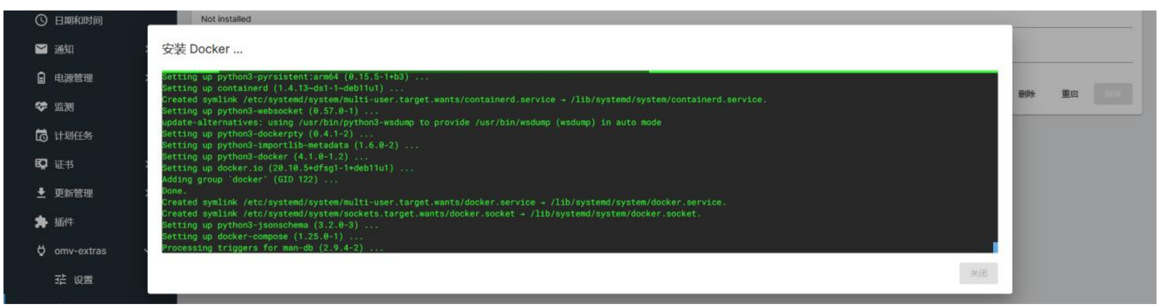
a. First install **apparmor**

```
orangepi@orangepi:~$ sudo apt install -y apparmor
```

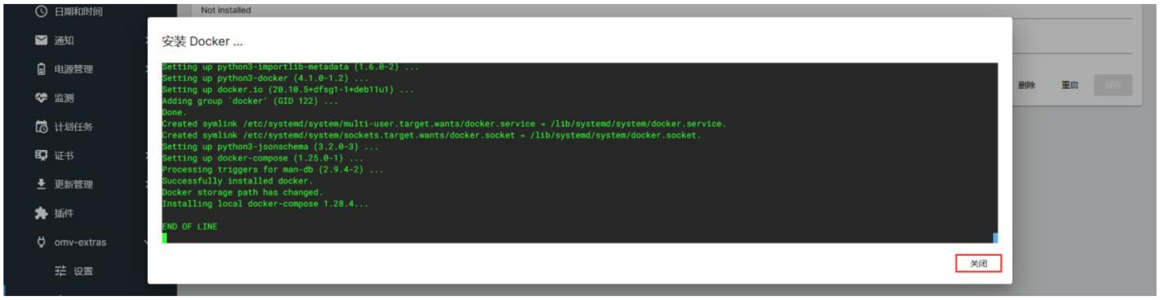
b. Then open the control interface of **Docker**, and then click the **install** button to start installing Docker



c. The display output of the Docker installation process is shown below



d. The display after the Docker installation is completed is as follows, and then click **Close**.

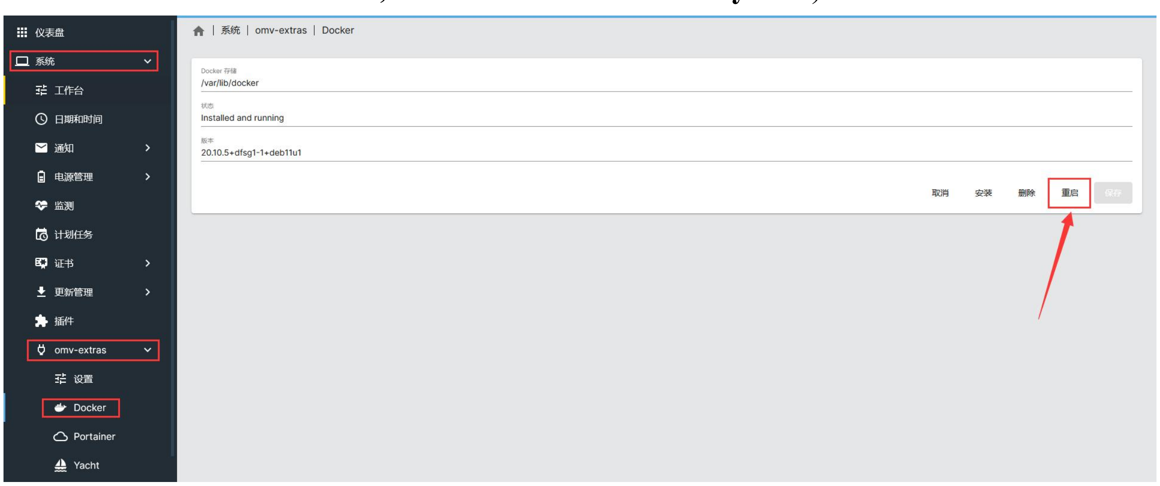


e. Finally, you need to set the address of the Docker warehouse to the domestic address to speed up the download speed of the Docker container. The method looks like this:

a) First open the **/etc/docker/daemon.json** file, and then add the configuration in the red font part below (note that **"data-root": "/var/lib/docker"** should be followed by a ,)

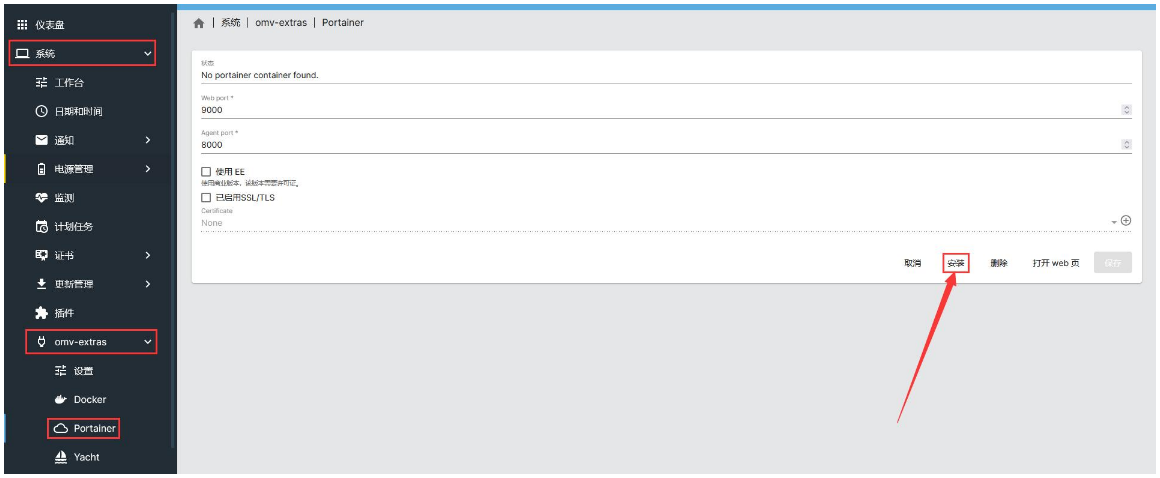
```
orangepi@orangepi:~$ sudo vim /etc/docker/daemon.json
{
  "data-root": "/var/lib/docker",
  "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]
}
```

b) Then click the **restart** button on the Docker control interface to restart the Docker service for the configuration to take effect (**if an error is reported, first check whether the above configuration is correct, and then try a few more times, or restart the Debian11 system**)



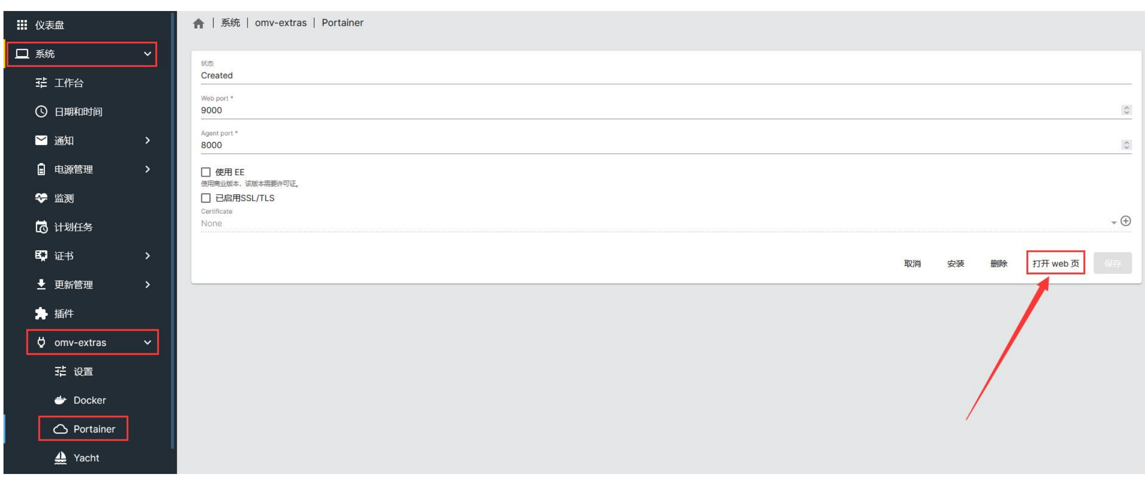
18) **Portainer** is a docker visual management tool. The steps to install Portainer are:

a. First open Portainer's control interface, and then click the **install** button to start installing Portainer

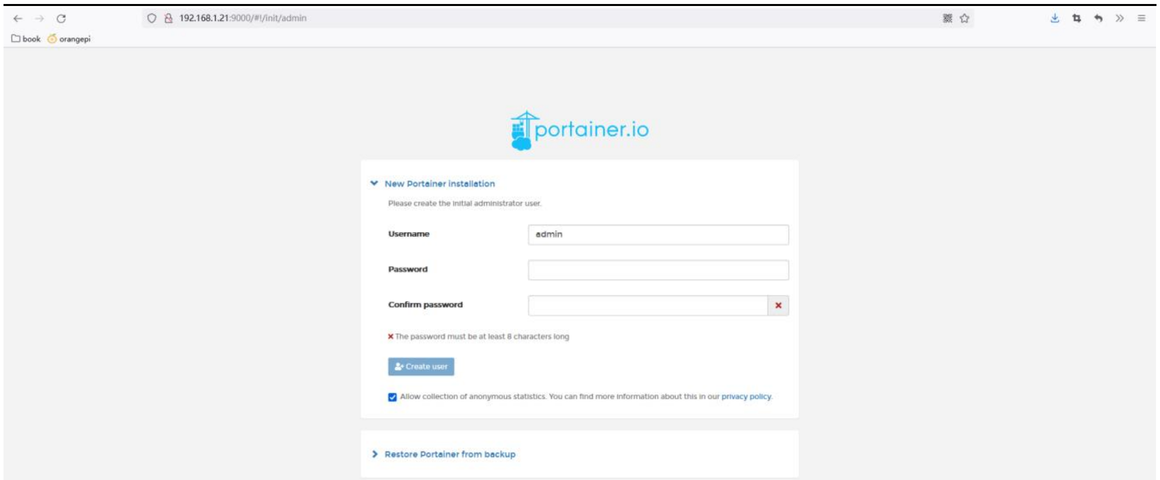


b. After the installation of Portainer is completed, the display is as shown in the figure below, and then click the close button to **close** it.

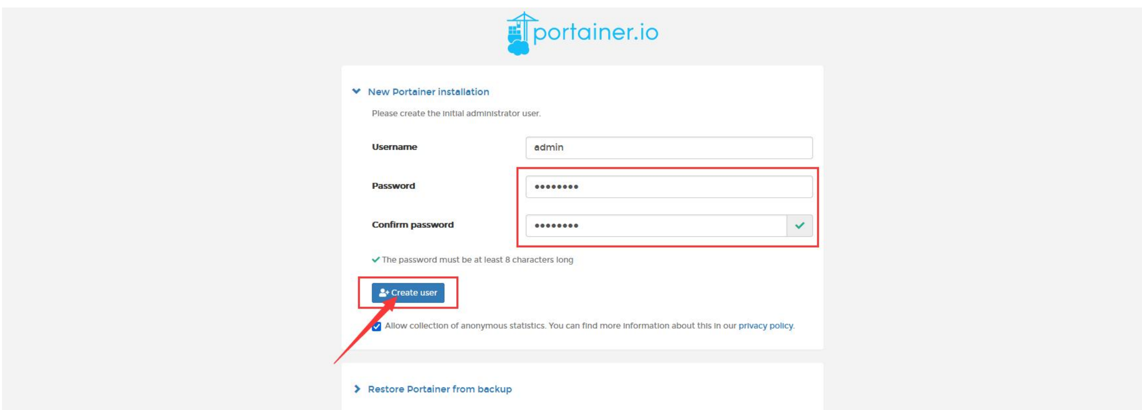
c. Then open Portainer's control interface, and then click to open the web page to open Portainer's web control interface



d. The display of d.Portainer's web control interface after opening is as follows



e. Then set the password of Portainer, and then click **Create user** to enter the web control interface of Portainer

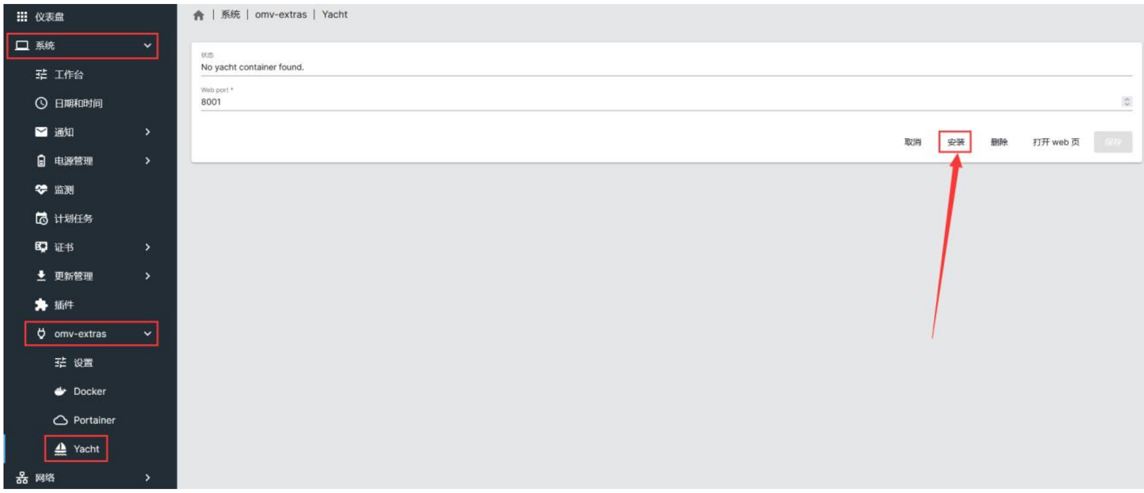


f. The main interface of Portainer after login is shown as below

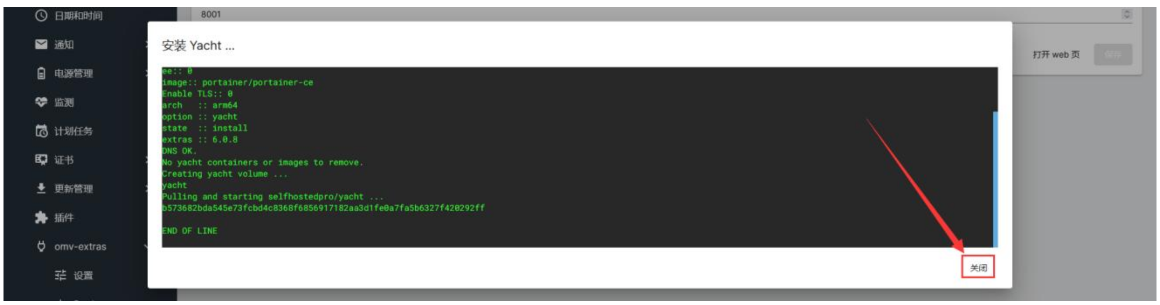
g. Please study the usage of g.Portainer by yourself

19) **Yacht** and Portainer have similar functions. They are both docker visual management tools. The installation steps are as follows:

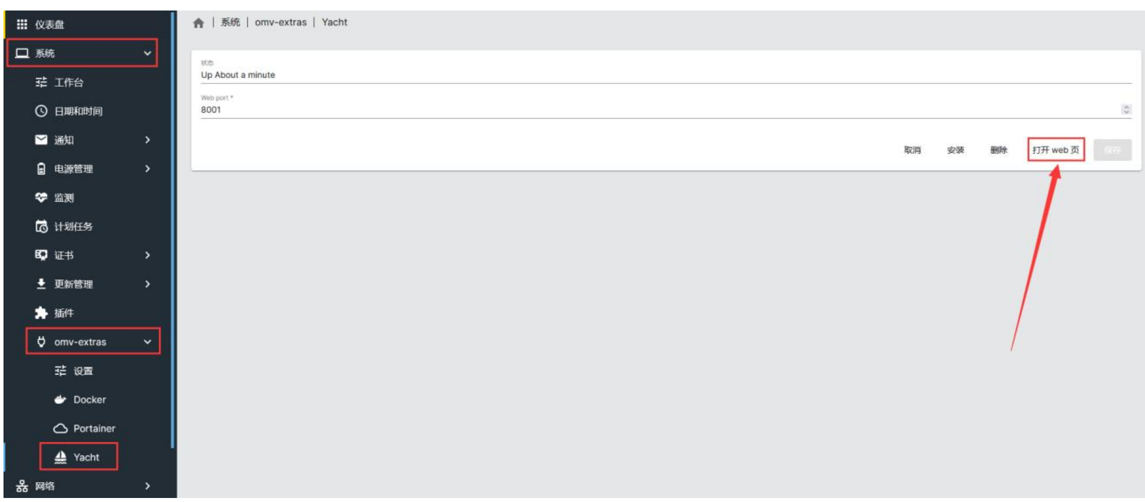
a. First open the Yacht control interface in OMV, and then click the install button to start installing Yacht



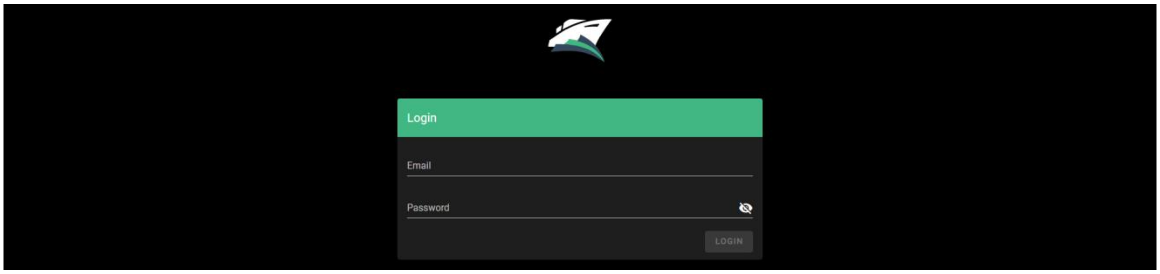
b. The display after Yacht is installed is shown in the figure below, and then click the close button to **close** it



c. Then open Yacht's control interface in OMV, and then click to **open the web page** to open Yacht's web control interface

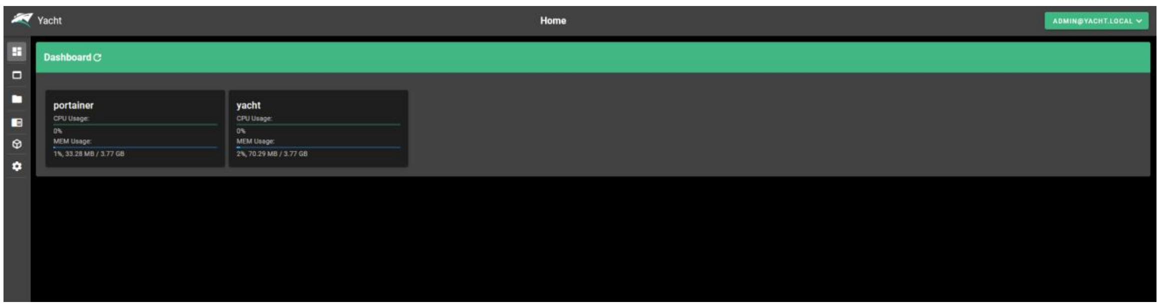


d. The display of Yacht's web control interface after opening is as follows



e. Then enter Yacht's default account **admin@yacht.local** in the **Email** column, and enter the default password **pass** in the **Password** column, and then click **LOGIN** to enter the Yacht web control interface

f. The main interface after Yacht login is shown as below

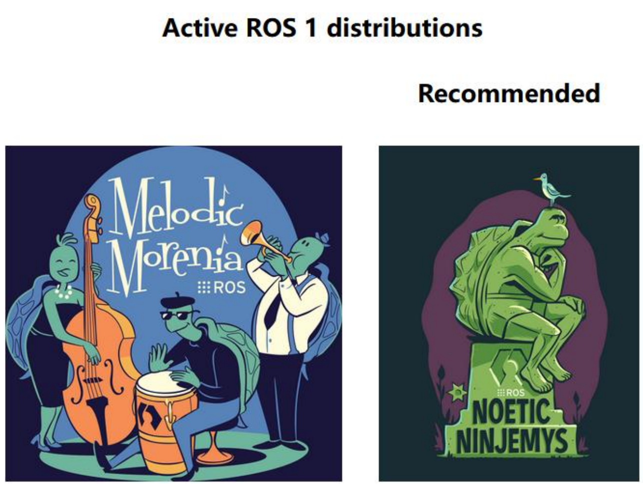


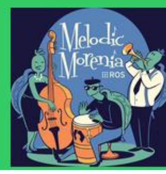
g. Please do your own research on how to use Yacht

3. 43. ROS installation method

3. 43. 1. How to install ROS 1 Noetic

1) The current active version of ROS 1 is as follows, the recommended version is **Noetic Ninjemys**



Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

```
http://docs.ros.org
https://wiki.ros.org/Distributions
```

2) The link to the official installation documentation for ROS 1 **Noetic Ninjemys** is as follows:

a. Ubuntu

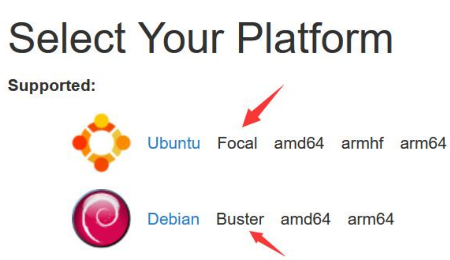
```
http://wiki.ros.org/noetic/Installation/Ubuntu
```

b. Debian

```
http://wiki.ros.org/noetic/Installation/Debian
```

3) **Ubuntu20.04** is recommended for Ubuntu Linux in the official installation document of ROS**Noetic Ninjemys, so please make sure that the system used by the development board is Ubuntu 20.04. If you want to use the Debian system, please test it yourself, and I will not demonstrate it here.**

```
http://wiki.ros.org/noetic/Installation
```



4) First add the software source of ros

```
orangepi@orangepi:~$ sudo sh -c 'echo    \
"deb http://mirrors.ustc.edu.cn/ros/ubuntu $(lsb_release -sc) main"    \
```

```
> /etc/apt/sources.list.d/ros-latest.list'
```

5) Then set Keys

```
orangepi@orangepi:~$ sudo apt-key adv    \
--keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key \
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

Executing: /tmp/apt-key-gpghome.0zbZ9ucMdb/gpg.1.sh --keyserver
hkp://keyserver.ubuntu.com:80 --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
gpg: key F42ED6FBAB17C654: public key "Open Robotics <info@osrfoundation.org>"
imported
gpg: Total number processed: 1
gpg:                    imported: 1
```

6) Then update the apt repository cache

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

7) Then install ros

```
orangepi@orangepi:~$ sudo apt install -y ros-noetic-desktop-full
```

8) Before running the commands in ros, you need to set the environment variables first

```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
```

9) If you don't want to manually set the environment variables before using the commands in ros, you can add the commands for setting the environment variables to **~/.bashrc**, so that every time you open a new terminal, the environment variables of ros will be automatically set

```
orangepi@orangepi:~$ echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
orangepi@orangepi:~$ source ~/.bashrc
```

10) Now everything needed to run the core ros package has been installed. If you want to create and manage your own ros workspace, you also need to install some other tools. For example, rosinstall is a common command-line tool that lets you download the source tree of some ros packages. Run the following command to install this tool and other

dependencies needed to build the ros package

```
orangepi@orangepi:~$ sudo apt install -y python3-rosdep python3-rosinstall    \
python3-rosinstall-generator python3-wstool build-essential
```

11) Before using the ROS tool, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code

Note that running the following command needs to ensure that the development board can access github normally, otherwise an error will be reported due to network problems.

```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

        rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
```

```
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

12) How to verify if ROS is installed correctly

a. First run **roscore**

```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ roscore
... logging to
/home/orangepi/.ros/log/132132c4-c873-11ec-9b13-5099013e1a05/roslaunch-orangepi-1
8381.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://orangepi:44425/
ros_comm version 1.15.14


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.14

NODES

auto-starting new master
process[master]: started with pid [18389]
ROS_MASTER_URI=http://orangepi:11311/
```

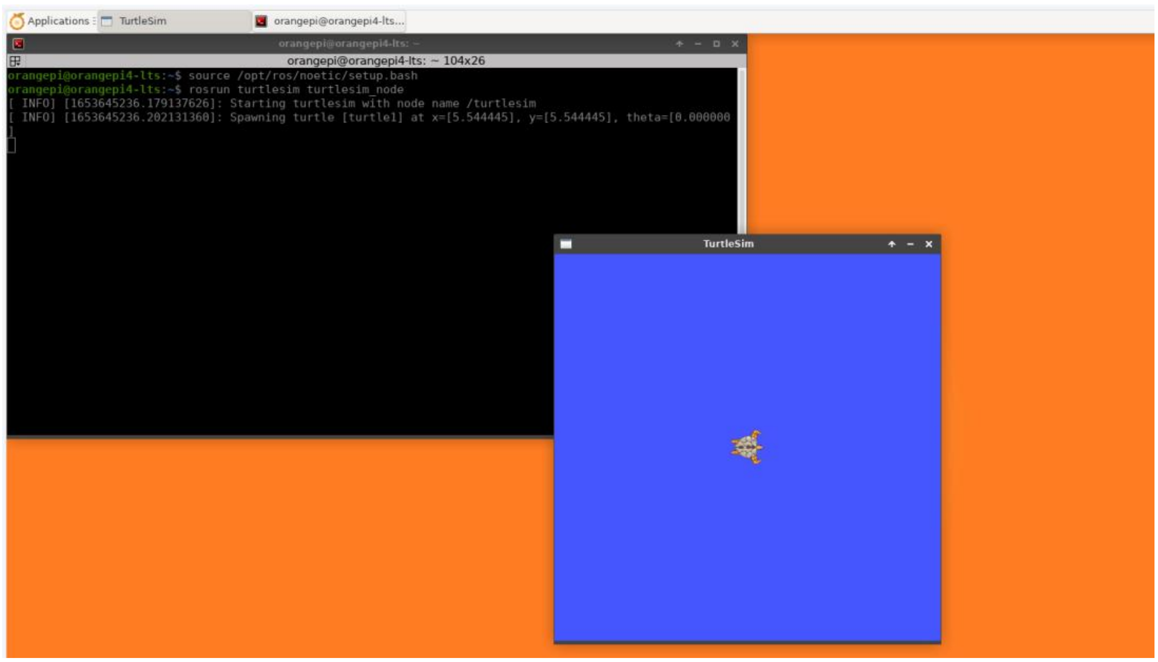
```
setting /run_id to 132132c4-c873-11ec-9b13-5099013e1a05
process[rosout-1]: started with pid [18399]
started core service [/rosout]
```

b. Then start a small turtle routine to test whether ROS can be used normally

a) First open a command line terminal window in the desktop

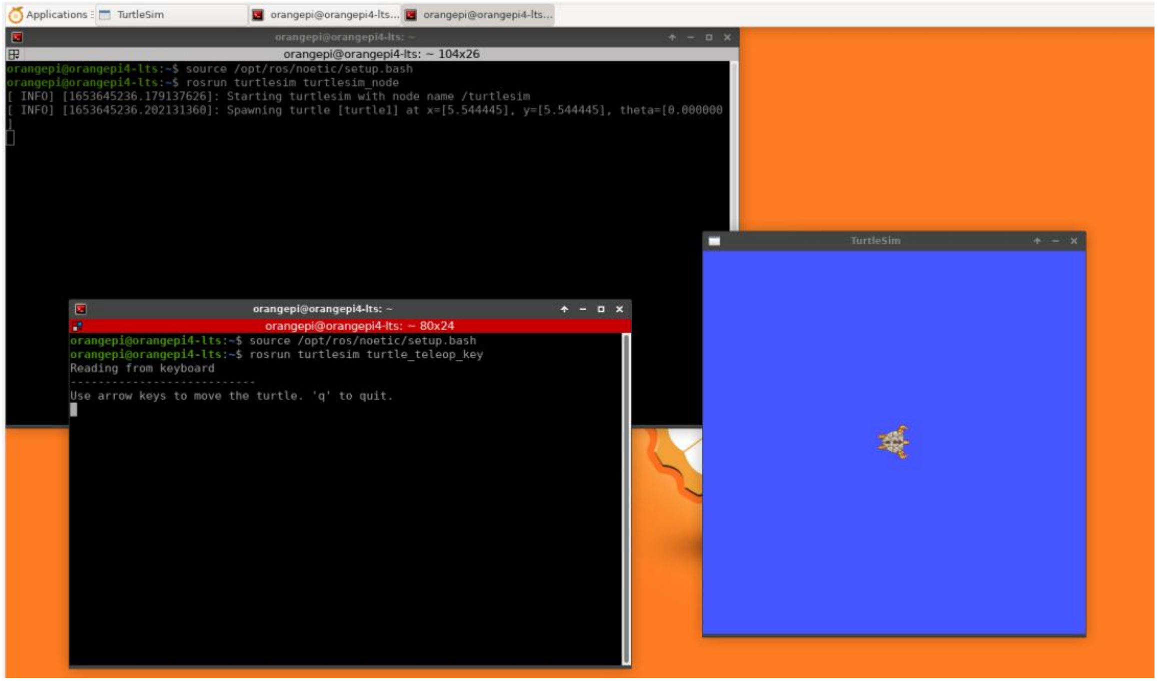
b) Then enter the following command, a small turtle as shown in the figure below will pop up

```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ rosrun turtlesim turtlesim_node
```

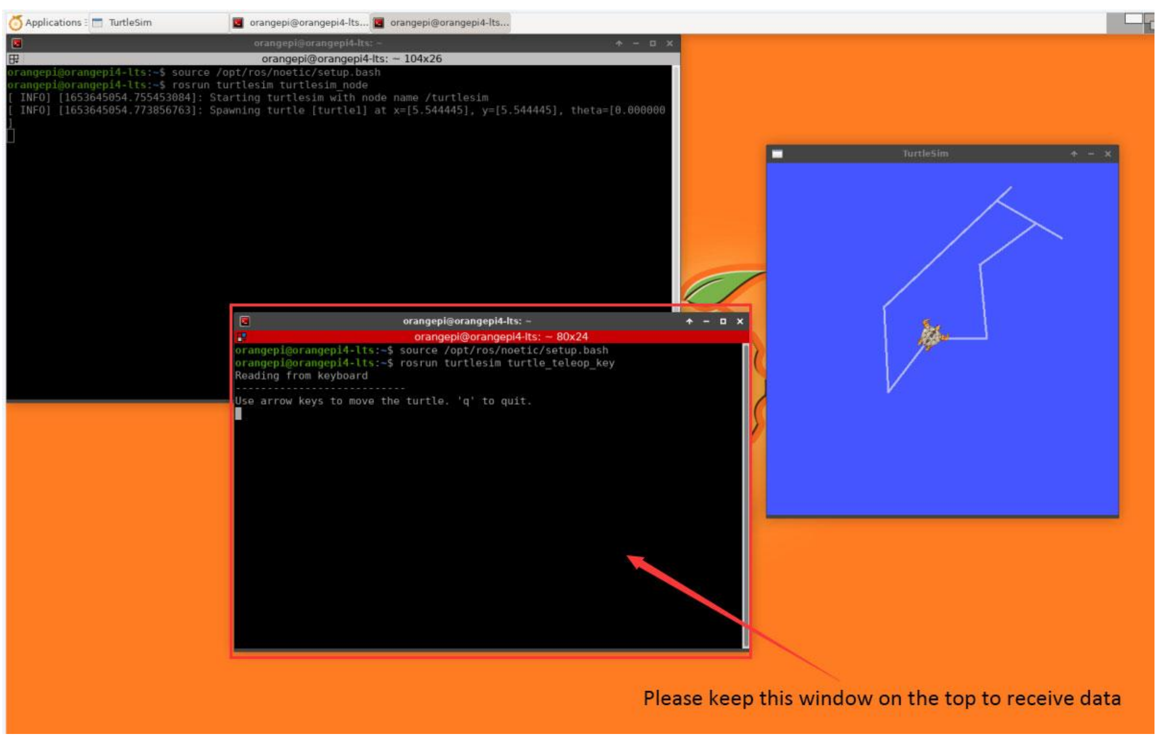


c) Then open a terminal window and enter the following command to run the turtle control program

```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ rosrun turtlesim turtle_teleop_key
```



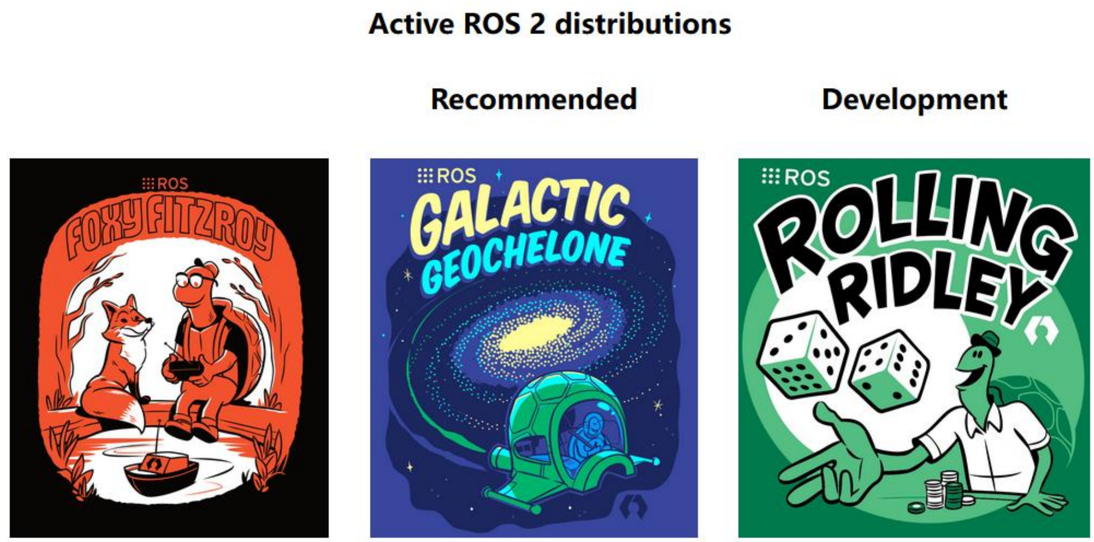
d) Then please keep the terminal window of the turtle control program you just opened at the top. At this time, you can control the small turtle to move up, down, left and right by pressing the direction keys on the keyboard.



3. 43. 2. How to install ROS 2 Galactic

1) The current active versions of ROS 2 are as follows, the recommended version is

Galactic Geochelone



Distro	Release date	Logo	EOL date
Humble Hawksbill	May 23rd, 2022		May 2027
Galactic Geochelone	May 23rd, 2021		November 2022
Foxy Fitzroy	June 5th, 2020		May 2023

```
http://docs.ros.org
http://docs.ros.org/en/galactic/Releases.html
```

2) The link to the official installation documentation of ROS 2 **Galactic Geochelone** is as follows:

```
docs.ros.org/en/galactic/Installation.html
http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html
```

3) Ubuntu 20.04 is recommended for Ubuntu Linux in the official installation document of ROS 2 **Galactic Geochelone**, so please make sure that the system used by the development board is **Ubuntu20.04**. There are several ways to install ROS 2. The

following demonstrates how to install ROS 2 **Galactic Geochelone** through **Debian packages**.

4) First add the key

```
orangepi@orangepi:~$ sudo apt-key adv --keyserver \
'hkp://keyserver.ubuntu.com:80' \
--recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

[sudo] password for orangepi:
Executing: /tmp/apt-key-gpghome.gNBSKx6Ums/gpg.1.sh --keyserver
hkp://keyserver.ubuntu.com:80 --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
gpg: key F42ED6FBAB17C654: public key "Open Robotics <info@osrfoundation.org>"
imported
gpg: Total number processed: 1
gpg:               imported: 1
```

5) Then add the repository source of ROS 2 to the Ubuntu system

```
orangepi@orangepi:~$ echo "deb [arch=$(dpkg --print-architecture)] \
http://mirrors.ustc.edu.cn/ros2/ubuntu $(source /etc/os-release && echo \
$UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list \
> /dev/null
```

6) Then update the apt repository cache

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

7) Then you can install ROS 2 related packages. The following commands will install ROS, RViz, demos, tutorials

```
orangepi@orangepi:~$ sudo apt install -y ros-galactic-desktop
```

8) Before running the **ros2** command, you need to set the environment variable first

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 -h
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.
```

9) You can use the following method to test whether ROS 2 is successfully installed
   a. Open a terminal first, then use the following two commands to run a **C++ talker** that keeps sending Hello World

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 run demo_nodes_cpp talker
[INFO] [1649599956.390893735] [talker]: Publishing: 'Hello World: 1'
[INFO] [1649599957.390812753] [talker]: Publishing: 'Hello World: 2'
[INFO] [1649599958.390890143] [talker]: Publishing: 'Hello World: 3'
......
```

   b. Then open a terminal and use the following two commands to run a **Python**

listener. If you can receive the Hello World sent above, it means that both the C++ and Python APIs of RSO 2 can work normally

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 run demo_nodes_py listener
[INFO] [1649600109.504678962] [listener]: I heard: [Hello World: 154]
[INFO] [1649600110.393777793] [listener]: I heard: [Hello World: 155]
[INFO] [1649600111.393769143] [listener]: I heard: [Hello World: 156]
......
```

10) For the usage of ROS, please refer to the documentation of ROS 2

http://docs.ros.org/en/galactic/Tutorials.html

11) Run the following command to uninstall ROS 2

```
orangepi@orangepi:~$ sudo apt remove ~nros-galactic-* && sudo apt autoremove
orangepi@orangepi:~$ sudo rm /etc/apt/sources.list.d/ros2.list
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt autoremove
orangepi@orangepi:~$ sudo apt upgrade
```

3. 44. Installation method of Pi-hole

Note that this section only provides the installation method of Pi-hole, please refer to the official documentation of Pi-hole for the usage method:

https://pi-hole.net

https://docs.pi-hole.net

Note that Pi-hole does not support Ubuntu22.04 yet, so please do not use this system to install Pi-hole.

1) First download Pi-hole's installation script, then run

a. Use Orange Pi to provide pi-hole script installation **(recommended)**

Note that the pi-hole installation script provided by Orange Pi does not modify any pi-hole functions, but only solves the problem of download and installation

failure caused by inability to access Github normally.

a) The first method: first download the repository of pi-hole, and then run the installation script

```
orangepi@orangepi:~$ git clone --depth 1 https://gitee.com/leeboby/pi-hole.git \
Pi-hole
orangepi@orangepi:~$ cd Pi-hole/automated\ install/
orangepi@orangepi:~$ sudo bash basic-install.sh
```

b) The second method: first download the installation script, then run

```
orangepi@orangepi:~$ wget -O basic-install.sh \
https://gitee.com/leeboby/pi-hole/raw/master/automated%20install/basic-install.sh
orangepi@orangepi:~$ sudo bash basic-install.sh
```

b. Install using pi-hole official script **(not recommended if it will not solve network problems)**

a) The first method: use the following command to download the installation script of pi-hole and run it directly

```
orangepi@orangepi:~$ curl -sSL https://install.pi-hole.net | bash
```

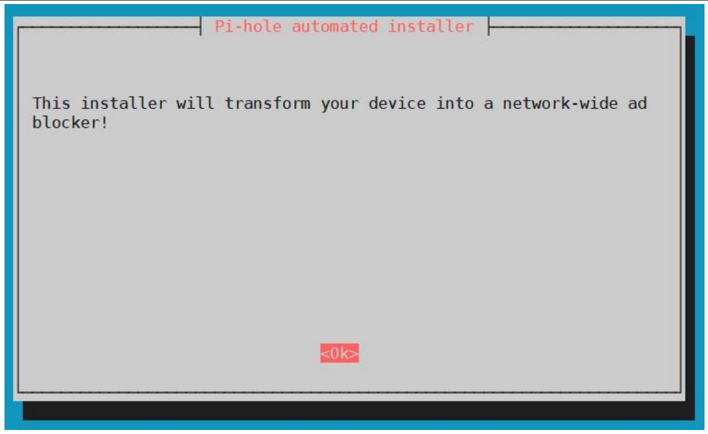
b) The second method: first download the repository of pi-hole, and then run the installation script

```
orangepi@orangepi:~$ git clone --depth 1 https://github.com/pi-hole/pi-hole.git \
Pi-hole
orangepi@orangepi:~$ cd Pi-hole/automated\ install/
orangepi@orangepi:~$ sudo bash basic-install.sh
```

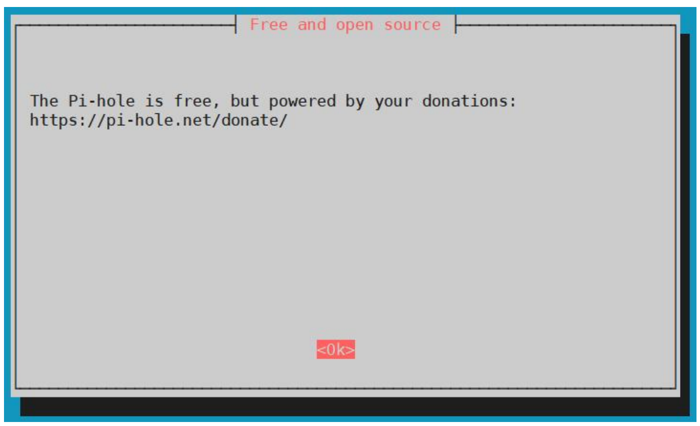
c) The third method: first download the installation script, then run

```
orangepi@orangepi:~$ wget -O basic-install.sh https://install.pi-hole.net
orangepi@orangepi:~$ sudo bash basic-install.sh
```

2) Then press enter

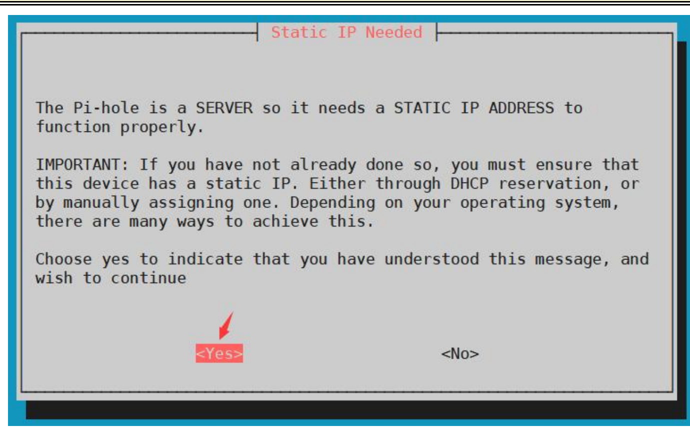


3) Continue to press Enter



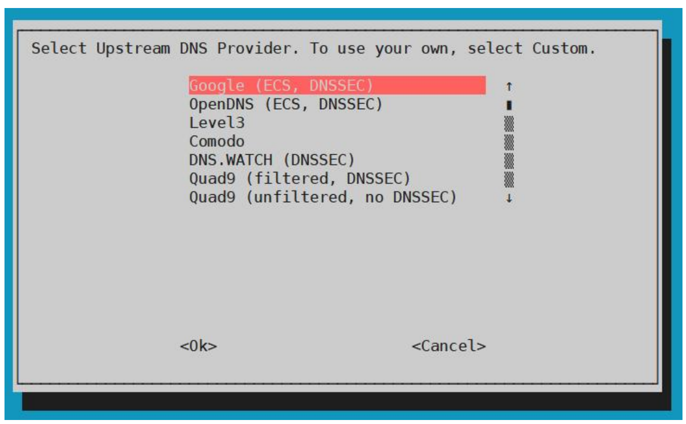
4) Then Pi-hole will prompt you to set a static IP address, please select **Yes** and press Enter

Here, you can not set the static IP address first, and then set it after the installation is completed. For the setting method of the static IP address, please refer to the description in the section on the method of setting the static IP address.

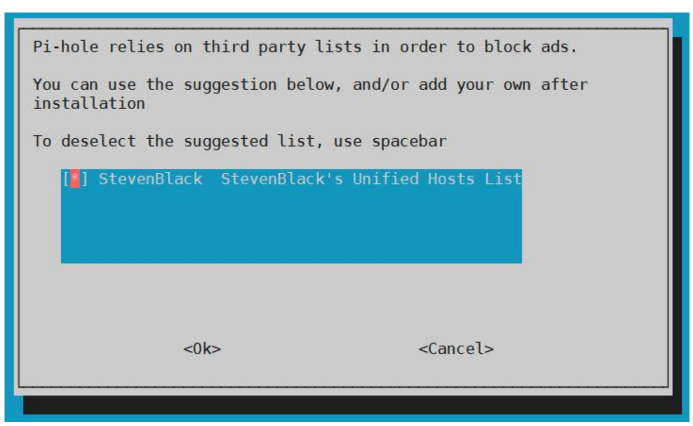


5) Then select the network interface. If you use a network cable, select eth0. If you want to use WIFI, select wlan0. After selecting, press Enter.

6) Then select the DNS provider, generally select Google.

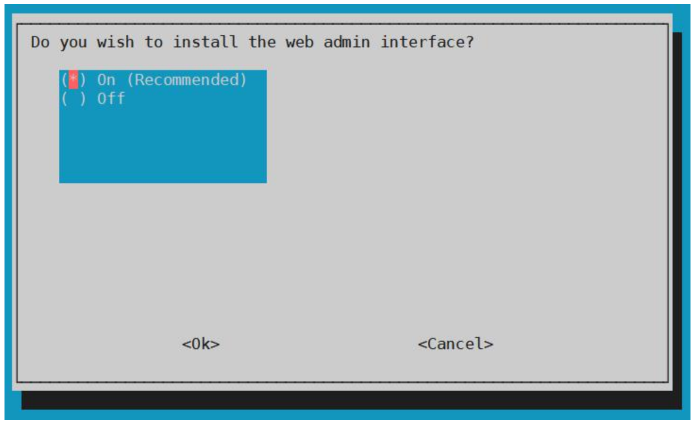


7) Then select the rule list and press Enter

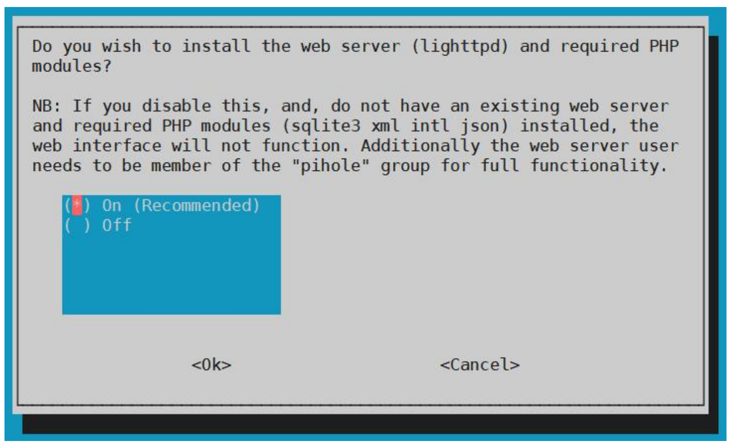


8) Then choose whether to install the web management interface, and press Enter to

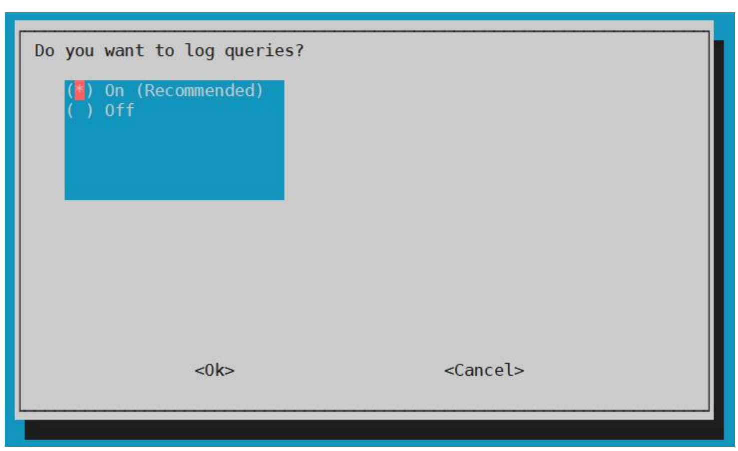
select the installation



9) Then press Enter to choose to install the web server and the required PHP modules



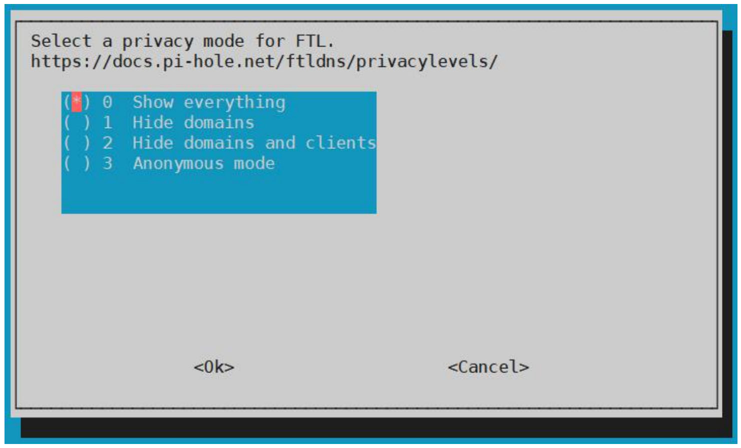
10) Then press Enter to select Open Log



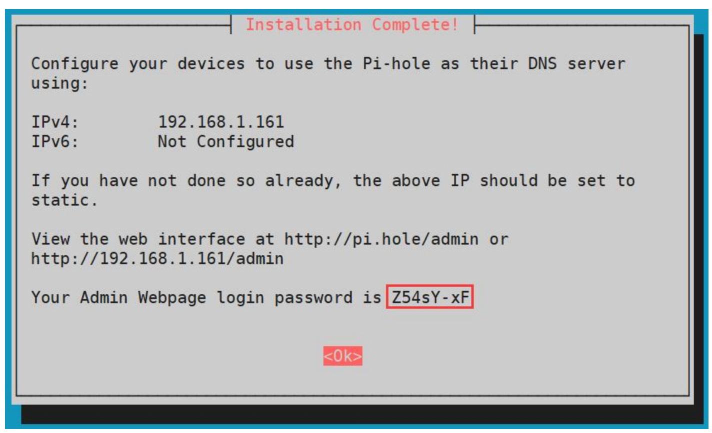
11) Then select the privacy mode, and select the default **0 Show everything** during

installation. Anyway, it can be modified after the installation is completed. Please check the following link for the difference between the different options of the privacy mode:

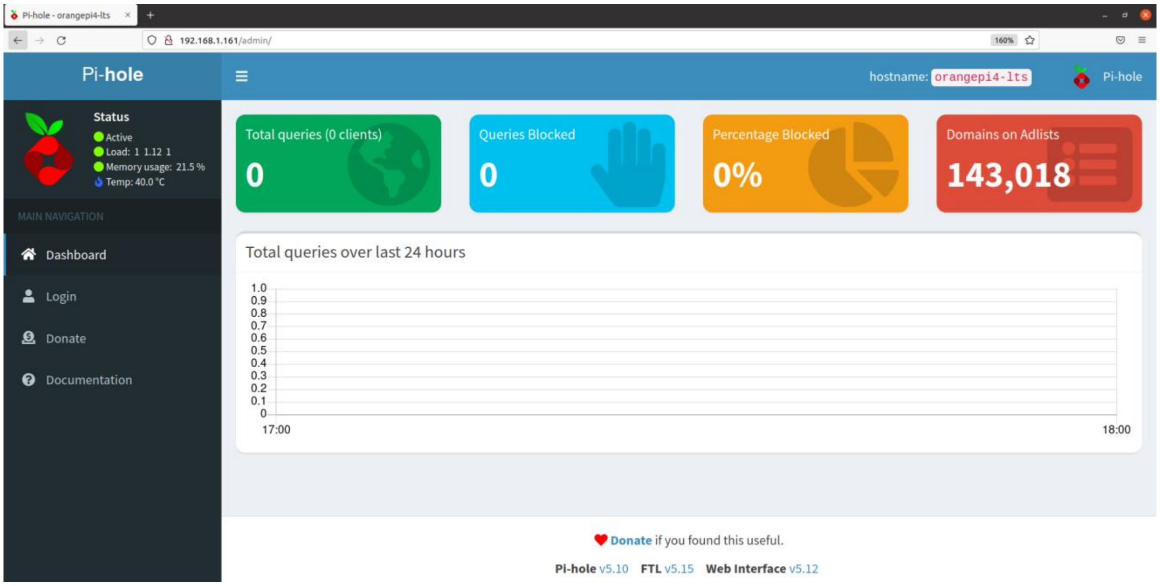
https://docs.pi-hole.net/ftldns/privacylevels/



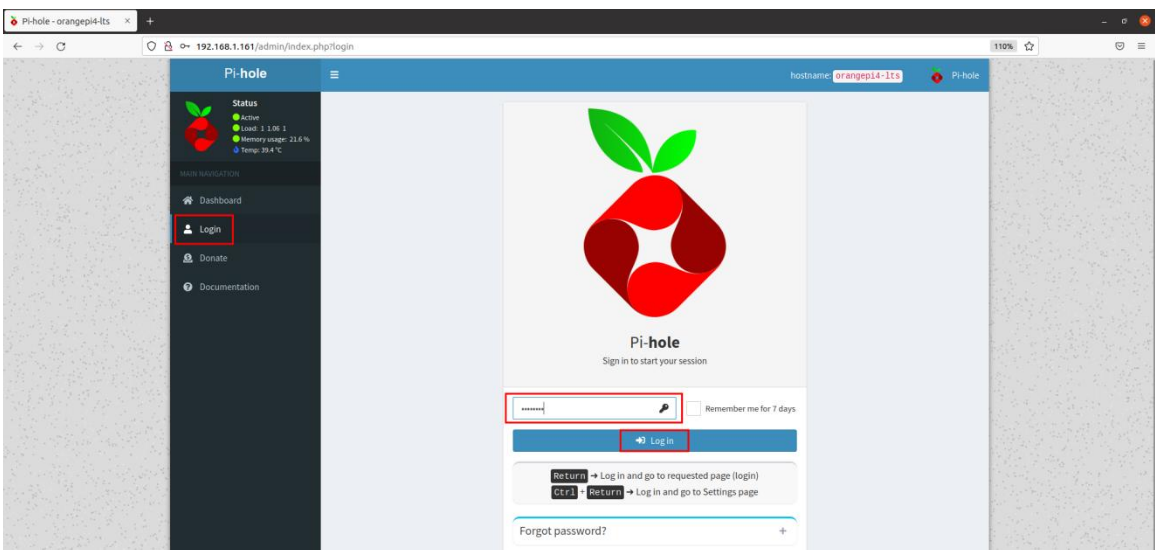
12) Then wait for the pi-hole installation to complete, and finally the following information will be prompted. The key point is to remember the web interface login address and login password



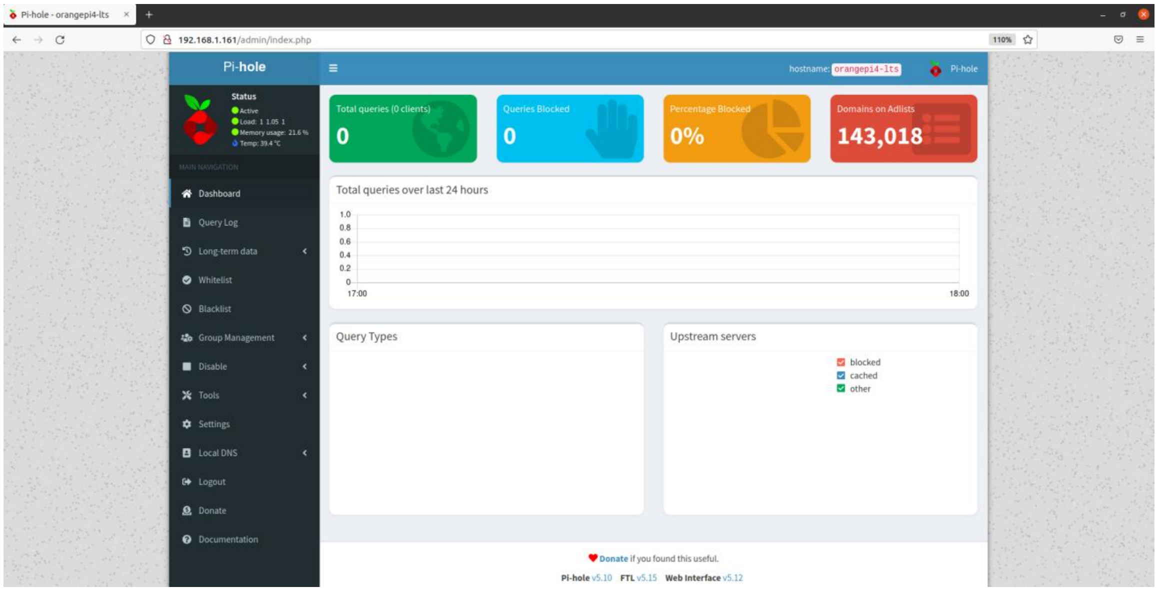
13) Then enter the **IP address /admin of the development board** in the browser to see the web management interface of pi-hole



14) Click **Login** on the left, then enter the password shown above and click **Log in** below to log in to pi-hole



15) The interface after logging in is shown as follows, you can see that there are many more options than before logging in



3.45. Tencent ncnn high-performance neural network forward computing framework test

If you don't know what ncnn is, you can read the introduction of ncnn in the README on github.

The ncnn github repository address is: https://github.com/Tencent/ncnn

ncnn

license BSD-3-Clause | downloads 89k | codecov 93% | code quality: c/c++ A+

ncnn is a high-performance neural network inference computing framework optimized for mobile platforms. ncnn is deeply considerate about deployment and uses on mobile phones from the beginning of design. ncnn does not have third party dependencies. It is cross-platform, and runs faster than all known open source frameworks on mobile phone cpu. Developers can easily deploy deep learning algorithm models to the mobile platform by using efficient ncnn implementation, create intelligent APPs, and bring the artificial intelligence to your fingertips. ncnn is currently being used in many Tencent applications, such as QQ, Qzone, WeChat, Pitu and so on.

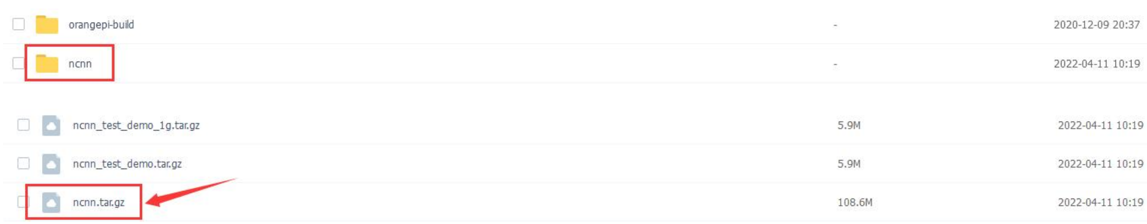
ncnn 是一个为手机端极致优化的高性能神经网络前向计算框架。ncnn 从设计之初深刻考虑手机端的部署和使用。无第三方依赖，跨平台，手机端 cpu 的速度快于目前所有已知的开源框架。基于 ncnn，开发者能够将深度学习算法轻松移植到手机端高效执行，开发出人工智能 APP，将 AI 带到你的指尖。ncnn 目前已在腾讯多款应用中使用，如 QQ，Qzone，微信，天天P图等。

1) Tencent ncnn source code download command is as follows

   a. The first method: Download the **ncnn.tar.gz** compressed package provided in the Orang Pi Google cloud disk

      a) From the Google cloud disk link below, you can download the **ncnn.tar.gz** source code compressed package. Go to the ncnn folder and you can see

Link:
https://drive.google.com/drive/folders/1JAj7xT1ViZKH51BZ-dUqM9bBySQILpZ1?usp=sharing



      b) After downloading the **ncnn.tar.gz** compressed package, first upload **ncnn.tar.gz** to the Linux system of the development board

      c) Then extract **ncnn.tar.gz** using the following command

```
orangepi@orangepi:~$ tar zxf ncnn.tar.gz
orangepi@orangepi:~$ ls
ncnn    ncnn.tar.gz
```

   b. The second method: use the git command to download the source code directly, but if the problem of accessing github from the development board is not solved, it is difficult to download successfully. If there is no problem accessing github, it is recommended to use this method, because this method can ensure that the code is up-to-date.

```
orangepi@orangepi:~$ git clone https://github.com/Tencent/ncnn.git
orangepi@orangepi:~$ cd ncnn
orangepi@orangepi:~/ncnn$ git submodule update --init
```

2) Then install the dependency package

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y build-essential git cmake    \
libprotobuf-dev protobuf-compiler libopencv-dev
```

3) Then start compiling, the ncnn compilation command is as follows

```
orangepi@orangepi:~$ cd ncnn
```

```
orangepi@orangepi:~/ncnn$ mkdir build
orangepi@orangepi:~/ncnn$ cd build
orangepi@orangepi:~/ncnn/build$ cmake \
-DCMAKE_TOOLCHAIN_FILE=../toolchains/aarch64-linux-gnu.toolchain.cmake \
-DNCNN_SIMPLEOCV=ON -DNCNN_BUILD_EXAMPLES=ON ..
orangepi@orangepi:~/ncnn/build$ make -j$(nproc)
[  0%] Built target ncnn-generate-spirv
[  1%] Generating source absval_arm_arm82.h
[  1%] Generating source convolution1d_arm_arm82.cpp
......
[ 99%] Linking CXX executable squeezenetssd
[ 99%] Built target squeezenetssd
[100%] Linking CXX executable shufflenetv2
[100%] Built target shufflenetv2
```

In the case of fan cooling, it takes about 8 minutes to compile ncnn directly on the development board, please wait patiently for the compilation to complete.

4) There are some test examples in ncnn, such as **squeezenet** test commands and results are as follows

```
orangepi@orangepi:~/ncnn/build$ cd ../examples
orangepi@orangepi:~/ncnn/examples$ ../build/examples/squeezenet \
../images/256-ncnn.png
532 = 0.165950
920 = 0.094098
716 = 0.062193
```

5) **benchncnn** can be used to test the reasoning performance of the neural network. The test method is as follows

a. The **benchncnn** executable file generated by compilation is in the following path. **Note that the execution path of the following command is the top-level directory of the ncnn source code**

```
orangepi@orangepi:~/ncnn$ ls build/benchmark/
benchncnn    CMakeFiles    cmake_install.cmake    Makefile
```

b. First, you need to copy **benchncnn** to the **benchmark** directory

```
orangepi@orangepi:~/ncnn$ cp build/benchmark/benchncnn benchmark/
```

c. The usage of **benchncnn** is as follows

./benchncnn [loop count] [num threads] [powersave] [gpu device] [cooling down]

Parameter

param	options	default
loop count	1~N	4
num threads	1~N	max_cpu_count
powersave	0=all cores, 1=little cores only, 2=big cores only	0
gpu device	-1=cpu-only, 0=gpu0, 1=gpu1 ...	-1
cooling down	0=disable, 1=enable	1

https://github.com/Tencent/ncnn/blob/9d0c36358cec2d1da471574064d4abd8787b45a8/benchmark/README.md

d. **benchncnn** using cpu test results are as follows

```
orangepi@orangepi:~/ncnn$ cd benchmark
orangepi@orangepi:~/ncnn/benchmark$ ./benchncnn 4 $(nproc) 0 -1
```

a) Debian Bullseye Linux5.10 desktop system test results

```
orangepi@orangepi4-lts:~/ncnn/benchmark$  ./benchncnn 4 $(nproc) 0 -1
loop_count = 4
num_threads = 6
powersave = 0
gpu_device = -1
cooling_down = 1
          squeezenet  min =   47.10  max =   49.54  avg =   48.17
     squeezenet_int8  min =   40.82  max =   55.36  avg =   45.99
           mobilenet  min =   59.23  max =   62.68  avg =   60.47
      mobilenet_int8  min =   40.55  max =   42.73  avg =   41.73
        mobilenet_v2  min =   57.35  max =   59.48  avg =   58.52
        mobilenet_v3  min =   43.07  max =   44.24  avg =   43.62
          shufflenet  min =   29.31  max =   32.46  avg =   30.76
       shufflenet_v2  min =   28.45  max =   29.52  avg =   28.86
             mnasnet  min =   47.08  max =   49.16  avg =   48.48
     proxylessnasnet  min =   56.76  max =   58.79  avg =   57.58
     efficientnet_b0  min =   85.52  max =   88.25  avg =   86.89
   efficientnetv2_b0  min =   94.84  max =  149.72  avg =  110.00
        regnety_400m  min =   75.44  max =   77.46  avg =   76.09
           blazeface  min =   17.04  max =   19.82  avg =   18.11
           googlenet  min =  128.53  max =  179.40  avg =  142.45
      googlenet_int8  min =  110.20  max =  114.53  avg =  111.82
            resnet18  min =  117.81  max =  123.54  avg =  120.93
       resnet18_int8  min =   80.30  max =  113.18  avg =   89.89
             alexnet  min =  114.82  max =  116.61  avg =  115.65
               vgg16  min =  645.31  max =  672.31  avg =  659.03
          vgg16_int8  min =  373.61  max =  393.09  avg =  380.40
            resnet50  min =  271.34  max =  325.47  avg =  285.54
       resnet50_int8  min =  214.00  max =  260.09  avg =  228.50
      squeezenet_ssd  min =  146.58  max =  148.52  avg =  147.51
 squeezenet_ssd_int8  min =  104.55  max =  113.50  avg =  109.26
       mobilenet_ssd  min =  126.63  max =  160.13  avg =  135.55
  mobilenet_ssd_int8  min =   83.87  max =   86.44  avg =   85.03
      mobilenet_yolo  min =  279.70  max =  293.56  avg =  284.54
  mobilenetv2_yolov3  min =  176.45  max =  238.33  avg =  196.00
         yolov4-tiny  min =  235.39  max =  261.13  avg =  244.49
           nanodet_m  min =   68.32  max =   81.87  avg =   74.00
    yolo-fastest-1.1  min =   40.27  max =   41.27  avg =   40.74
      yolo-fastestv2  min =   32.04  max =   35.44  avg =   33.70
orangepi@orangepi4-lts:~/ncnn/benchmark$
```

6) NanoDet is an ultra-fast and lightweight mobile Anchor-free object detection model. The test method is as follows

a. The compiled **nanodet** executable file is in the following path. **Note that the execution path of the following command is the upper level directory of the ncnn source code.**

```
orangepi@orangepi:~$ ls ncnn/build/examples/nanodet
```

```
ncnn/build/examples/nanodet
```

b. First create a new **nanodet_demo** folder

```
orangepi@orangepi:~$ mkdir nanodet_demo
```

c. Then copy the compiled **nanodet** executable program to the **nanodet_demo** folder

```
orangepi@orangepi:~$ cp ncnn/build/examples/nanodet nanodet_demo/
```

d. Then you need to download the **nanodet** model file and upload it to the **nanodet_demo** folder

a) The download address of the **nanodet** model file is as follows

```
https://github.com/nihui/ncnn-assets/tree/master/models
```

b) Open the link above, find the two files **nanodet_m.bin** and **nanodet_m.param**, download them, and upload them to the **nanodet_demo** folder of the Linux system of the development board

nanodet_m.bin	Add files via upload	16 months ago
nanodet_m.param	Add files via upload	16 months ago

c) At this point, there should be the following three files in the **nanodet_demo** folder

```
orangepi@orangepi:~$ cd nanodet_demo
orangepi@orangepi:~/nanodet_demo$ ls
nanodet    nanodet_m.bin    nanodet_m.param
```

e. Then you need to put the pictures you want to detect in the **nanodet_demo** folder, such as the picture below with many cars (you can use your mobile phone to take a few pictures of traffic or animals)

```
orangepi@orangepi:~/nanodet_demo$ ls
car.jpg    nanodet    nanodet_m.bin    nanodet_m.param
```

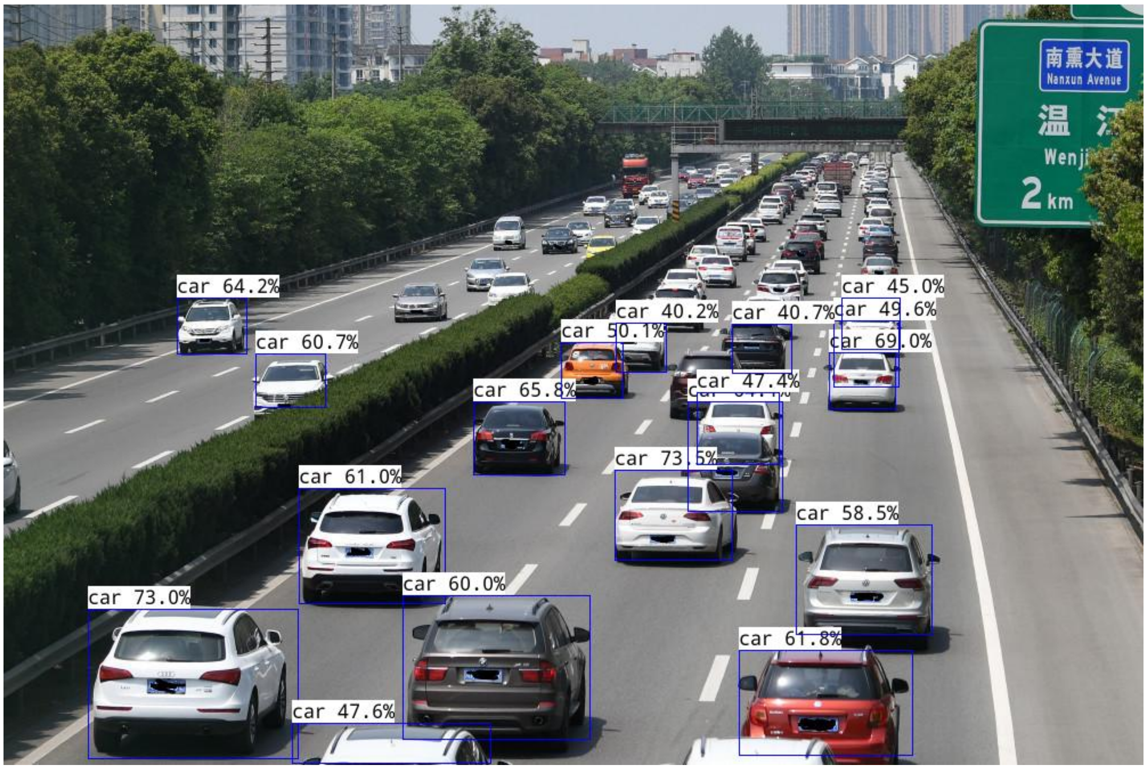
f. Then run the following command to use **nanodet** for target detection, please replace **car.jpg** with the name of your image

```
orangepi@orangepi:~/nanodet_demo$ ./nanodet car.jpg
2 = 0.73488 at 536.36 408.79 103.68 x 79.63
2 = 0.73003 at 74.47 530.85 184.61 x 131.70
2 = 0.68989 at 724.94 305.76 58.30 x 49.73
2 = 0.65828 at 412.10 348.38 80.33 x 64.65
2 = 0.64167 at 152.09 257.67 61.52 x 49.91
2 = 0.64124 at 600.63 348.06 83.82 x 96.93
2 = 0.61759 at 645.80 566.98 152.59 x 92.63
2 = 0.61004 at 259.78 424.55 128.62 x 101.88
2 = 0.60663 at 221.76 306.18 61.54 x 47.73
2 = 0.60043 at 350.11 518.50 164.37 x 126.59
2 = 0.58546 at 695.82 456.88 119.12 x 96.87
2 = 0.50075 at 489.94 296.66 54.39 x 49.01
2 = 0.49616 at 728.32 277.73 57.70 x 58.59
2 = 0.47553 at 253.21 630.12 174.18 x 35.88
2 = 0.47408 at 608.36 340.09 72.34 x 63.40
2 = 0.44989 at 735.15 257.97 51.55 x 45.79
2 = 0.40665 at 639.82 280.51 52.43 x 43.13
2 = 0.40169 at 537.50 279.67 44.98 x 43.54
imshow save image to image.png
waitKey stub
```

g. The result of the detection will be saved in a picture named **image.png**

```
orangepi@orangepi:~/nanodet_demo$ ls
car.jpg   image.png   nanodet   nanodet_m.bin   nanodet_m.param
```

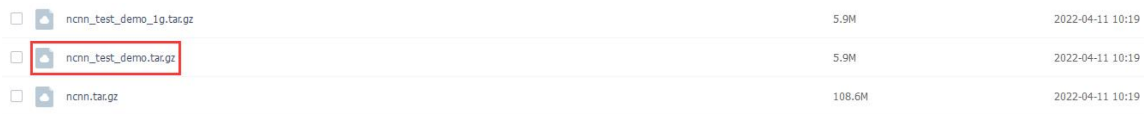
h. If you are using a desktop version of Linux system, you can directly open **image.png** to view it. If you are using a server version of Linux system, you can copy **image.png** to your computer for viewing. The content of **image.png** is shown in the figure below. You can see that the upper left corner of the recognized object will display the type of object and the percentage of reliability



7) In order to facilitate the testing of **benchncnn** and **nanodet**, I have compiled an executable file containing only **benchncnn** and **nanodet** and the model files required for the test, packaged into a **ncnn_test_demo.tar.gz** compressed package and placed it on Google cloud disk, no need to download Compile the source code of **ncnn**, use this executable program to start the test directly

a. You can download the **ncnn_test_demo.tar.gz** compressed package from the Google cloud disk link below. Go to the ncnn folder and you can see

Link:
https://drive.google.com/drive/folders/1JAj7xT1ViZKH51BZ-dUqM9bBySQILpZ1?usp=sharing



b. After downloading the **ncnn_test_demo.tar.gz** compressed package, first upload the **ncnn_test_demo.tar.gz** compressed package to the Linux system of the development board

c. Then extract **ncnn_test_demo.tar.gz** using the following command

```
orangepi@orangepi:~$ tar ncnn_test_demo.tar.gz
```

d. After decompression, enter the ncnn_test_demo directory and you can see that it contains two subfolders, **benchncnn_demo** and **nanonodet_demo**, which are used to test **benchncnn** and **nanodet** respectively

```
orangepi@orangepi:~$ cd ncnn_test_demo
orangepi@orangepi:~/ncnn_test_demo$ ls
benchncnn_demo    nanodet_demo
```

e. Enter the **benchncnn_demo** folder, and then run the command **./benchncnn 4 $(nproc) 0 -1** to directly test the inference performance of the neural network

```
orangepi@orangepi:~/ncnn_test_demo$ cd benchncnn_demo
orangepi@orangepi:~/ncnn_test_demo/benchncnn_demo$ ./benchncnn 4 $(nproc) 0 -1
```

f. Enter the **nanodet_demo** folder, and then run the **./nanodet car.jpg** command to directly use nanodet to detect objects in the **car.jpg** image. You can also put the image you want to detect in the **nanodet_demo** folder, and then Use **nanodet** to detect

```
orangepi@orangepi:~/ncnn_test_demo$ cd nanodet_demo
orangepi@orangepi:~/ncnn_test_demo/nanodet_demo$ ./nanodet car.jpg
```

Note that the content demonstrated in this section is mainly to prove that ncnn can be compiled and run normally on Orange Pi's development board and system. If there is any problem with the content demonstrated in this section, you can provide feedback and provide technical support (such as ncnn source code download failure, ncnn There are problems with compilation, problems with benchncnn and nanodet tests), but other things beyond this section cannot provide technical support, please study by yourself.

3. 46. Installation and testing method of face_recognition face recognition library

Note that the content in this section is tested on the desktop version of Linux system, so please make sure that the system used by the development board is the desktop version system.

In addition, the following installation tests are carried out under the orangepi user, please keep the environment consistent.

The address of the face_recognition source code repository is:
https://github.com/ageitgey/face_recognition

The documentation for the Chinese version of face_recognition is:
https://github.com/ageitgey/face_recognition/blob/master/README_Simplified_Chinese.md

3. 46. 1. Automatic installation of face_recognition using script

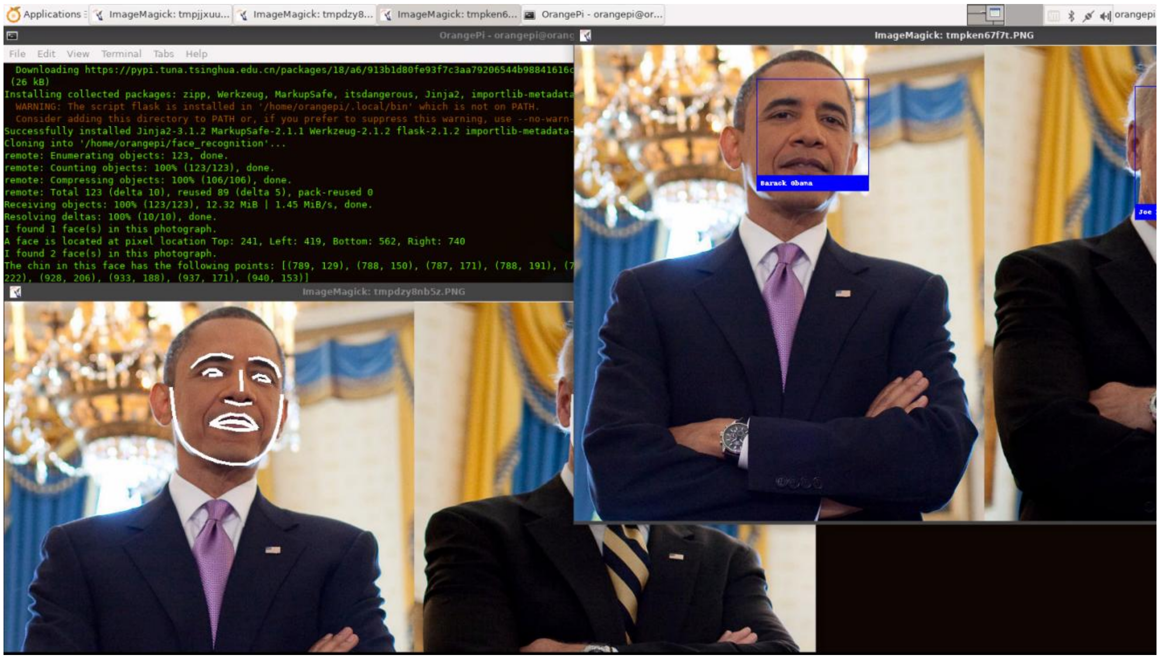
1) First open a terminal in the desktop and download **face_recognition_install.sh**

```
orangepi@orangepi:~/Desktop$ wget \
https://gitee.com/leeboby/face_recognition_install/raw/master/face_recognition_install.sh
```

2) Then execute the following command to start installing **face_recognition**

```
orangepi@orangepi:~/Desktop$ bash face_recognition_install.sh
```

3) After face_recognition is installed, it will automatically download the source code of face_recognition, and then automatically run some examples in face_recognition. If you can see the following pictures pop up on the desktop at the end, it means that the face_recognition installation test is successful



3. 46. 2. Manual installation of face_recognition

1) First create a new **~/.pip** directory, then add the **pip.conf** configuration file, and set the pip mirror source to Tsinghua source in it. The commands to be executed are as follows:

```
orangepi@orangepi:~$ mkdir -p ~/.pip
orangepi@orangepi:~$ cat <<EOF > ~/.pip/pip.conf
[global]
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
EOF
```

2) Then install the dependency package

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y python3-pip libopencv-dev \
python3-opencv imagemagick python3-scipy python3-setuptools python3-wheel \
python3-dev cmake python3-testresources
```

3) Then update pip3

```
orangepi@orangepi:~$ python3 -m pip install -U pip setuptools wheel
```

4) Before installing **face_recognition**, you first need to install the **dlib** library. Since the

dlib library is slow to compile and install on the development board, I saved a compiled dlib whl file on **gitee**, and it can be installed directly after downloading. The download address of the dlib whl file is as follows:

```
https://gitee.com/leeboby/python_whl
```

a. First download the python_whl repository to the Linux system of the development board

```
orangepi@orangepi:~$ git clone --depth=1 https://gitee.com/leeboby/python_whl
```

b. In the python_whl folder, you can see that there are multiple versions of the dlib installation package. The Linux systems corresponding to different versions of dlib are as follows:

Ubuntu20.04	**dlib-19.24.0-cp38-cp38-linux_aarch64.whl**
Ubuntu22.04	**dlib-19.24.0-cp310-cp310-linux_aarch64.whl**
Debian10	**dlib-19.24.0-cp37-cp37m-linux_aarch64.whl**
Debian11	**dlib-19.24.0-cp39-cp39-linux_aarch64.whl**

c. Then you can start to install dlib, the command is as follows

a) Ubuntu20.04

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install
dlib-19.24.0-cp38-cp38-linux_aarch64.whl
```

b) Ubuntu22.04

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install
dlib-19.24.0-cp310-cp310-linux_aarch64.whl
```

c) Debian10

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install
dlib-19.24.0-cp37-cp37m-linux_aarch64.whl
```

d) Debian11

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install
dlib-19.24.0-cp39-cp39-linux_aarch64.whl
```

d. After installation, if the version number of dlib can be printed normally by using the following command, it means that dlib is installed correctly

```
orangepi@orangepi:~/python_whl$ python3 -c "import dlib; print(dlib.__version__)"
```

```
19.24.0
```

5) then install **face_recognition_models-0.3.0-py2.py3-none-any.whl**

```
orangepi@orangepi:~/python_whl$ python3 -m pip install \
face_recognition_models-0.3.0-py2.py3-none-any.whl
```

6) then install **face_recognition**

```
orangepi@orangepi:~$ python3 -m pip install face_recognition
```

7) Then you **need to reopen a terminal** to find and run the two commands **face_detection** and **face_recognition**

a. face_recognition command is used to identify whose face is in a single image or a folder of images

b. face_detection command is used to locate the position of the face in a single image or a folder of images

```
orangepi@orangepi:~$ which face_detection
/usr/local/bin/face_detection
orangepi@orangepi:~$ which face_recognition
/usr/local/bin/face_recognition
```

Or run the following commands in the terminal, you can find the above two commands without reopening the terminal

```
orangepi@orangepi:~$ export PATH=/home/orangepi/.local/bin:$PATH
```

3. 46. 3. Test method of face_recognition

Note that the following operations are demonstrated on the desktop, so please connect the HDMI display first, or use NoMachine/VNC to log in to the Linux desktop to test.

1) There are some sample codes in the source code of **face_recognition**, which we can use for testing directly. The download address of the source code of face_recognition is as follows:

a. GitHub official download address

```
orangepi@orangepi:~$ git clone https://github.com/ageitgey/face_recognition.git
```

b. Gitee image download address

```
orangepi@orangepi:~$ git clone https://gitee.com/leeboby/face_recognition.git
```

2) The path of the face_recognition sample code is as follows

```
face_recognition/examples
```

3) The link to the Chinese documentation of face_recognition is as follows, please read it carefully before using face_recognition

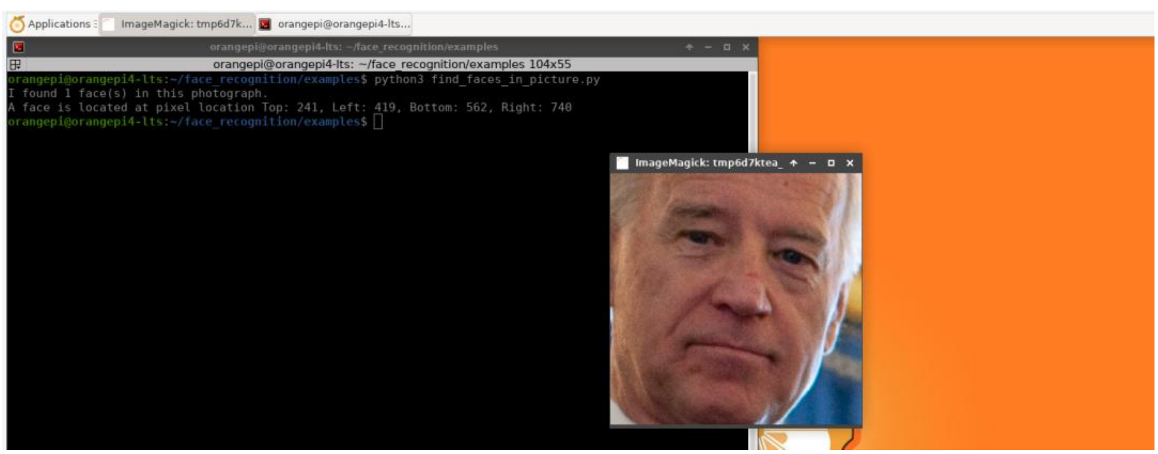
https://github.com/ageitgey/face_recognition/blob/master/README_Simplified_Chinese.md

4) **find_faces_in_picture.py** is used to locate the position of the face in the picture. The test steps are as follows

a. Open a terminal on the desktop, enter the **face_recognition/examples** directory, and execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 find_faces_in_picture.py
I found 1 face(s) in this photograph.
A face is located at pixel location Top: 241, Left: 419, Bottom: 562, Right: 740
```

b. Wait for a while and the following picture will pop up, this is the face located in the test picture



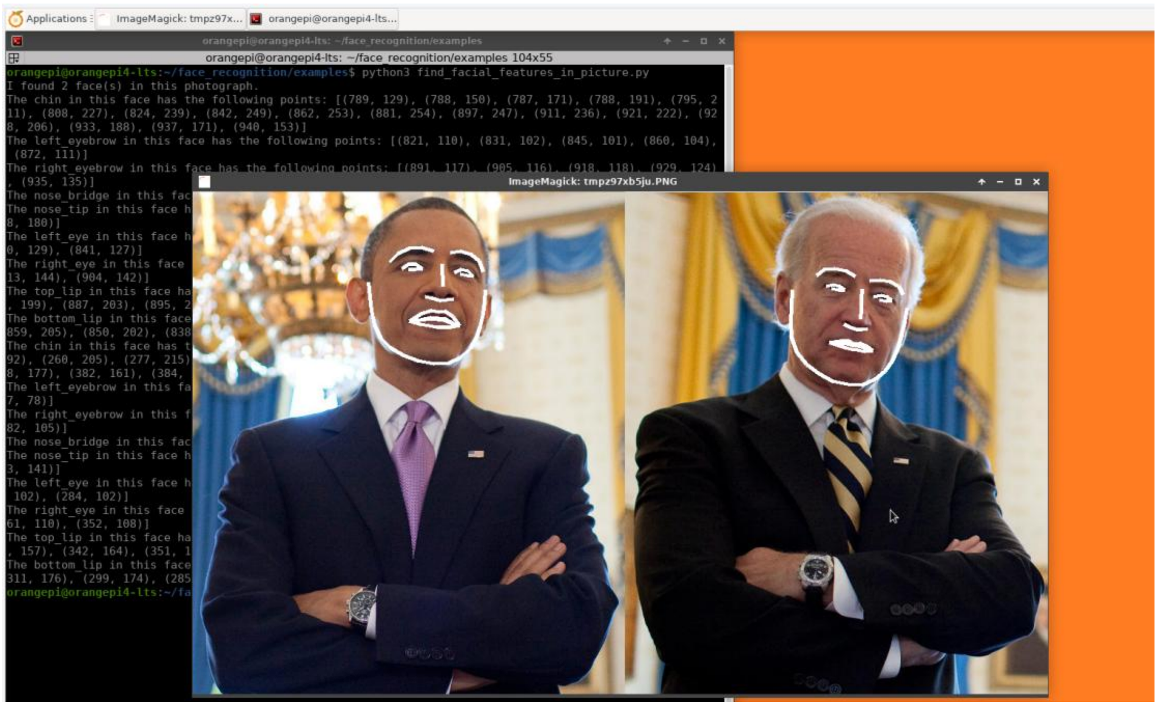
5) **find_facial_features_in_picture.py** is used to identify the key points of the face in a single picture. The test steps are as follows

a. Open a terminal on the desktop, enter the **face_recognition/examples** directory, and execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
```

```
orangepi@orangepi:~/face_recognition/examples$ python3 \
find_facial_features_in_picture.py
```

b. After waiting for a while, the picture below will pop up, and you can see that the outlines of the faces are marked

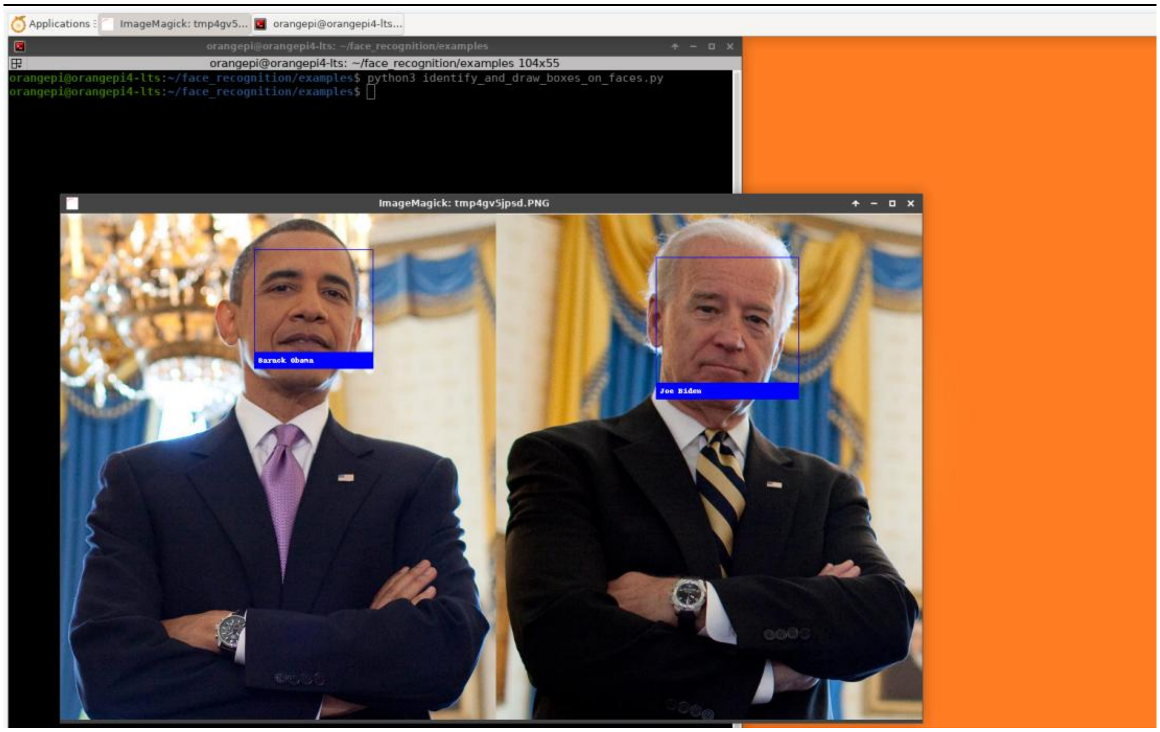


6) **identify_and_draw_boxes_on_faces.py** is used to identify faces and use box labels. The test steps are as follows

a. Open a terminal on the desktop, enter the **face_recognition/examples** directory, and execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 \
identify_and_draw_boxes_on_faces.py
```

b. After waiting for a while, the following picture will pop up. You can see that the faces in the picture are marked with boxes, and the names of the characters are displayed correctly.



7) **face_distance.py** is used to compare whether two faces belong to the same person at different precisions. First open a terminal, then enter the **face_recognition/examples** directory, and then execute the following command to see the output of the test

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 face_distance.py
The test image has a distance of 0.35 from known image #0
- With a normal cutoff of 0.6, would the test image match the known image? True
- With a very strict cutoff of 0.5, would the test image match the known image? True

The test image has a distance of 0.82 from known image #1
- With a normal cutoff of 0.6, would the test image match the known image? False
- With a very strict cutoff of 0.5, would the test image match the known image?
  False
```

8) **recognize_faces_in_pictures.py** is used to identify who the faces in unknown pictures are. First open a terminal, then enter the **face_recognition/examples** directory, and then execute the following command, wait for one end to see the test results

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 \
recognize_faces_in_pictures.py
```

```
Is the unknown face a picture of Biden? False
Is the unknown face a picture of Obama? True
Is the unknown face a new person that we've never seen before? False
```

9) **facerec_from_webcam_faster.py** is used to identify faces in USB cameras (**MIPI cameras are currently not supported**). The test steps are as follows:

a. First, please insert the USB camera into the USB interface of the development board, and then use the **v4l2-ctl** (note that the l in v4l2 is a lowercase letter l, not the number 1) command to check the serial number of the device node of the USB camera

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y v4l-utils
orangepi@orangepi:~$ v4l2-ctl --list-devices
USB2.0 UVC PC Camera: USB2.0 UV (usb-fe380000.usb-1):
        /dev/video0
        /dev/video1
        /dev/media0
```

b. Then open a terminal on the desktop, enter the **face_recognition/examples** directory, and first modify the device serial number of the camera used in **facerec_from_webcam_faster.py**. For example, through the v4l2-ctl --list-devices command above, you can see that the USB camera is **/dev/video0**. Since the default is 0, there is no need to modify it. If the device interface of the USB camera you see is not **/dev/video0**, then Need to modify 0 in **cv2.VideoCapture(0)** to the corresponding serial number

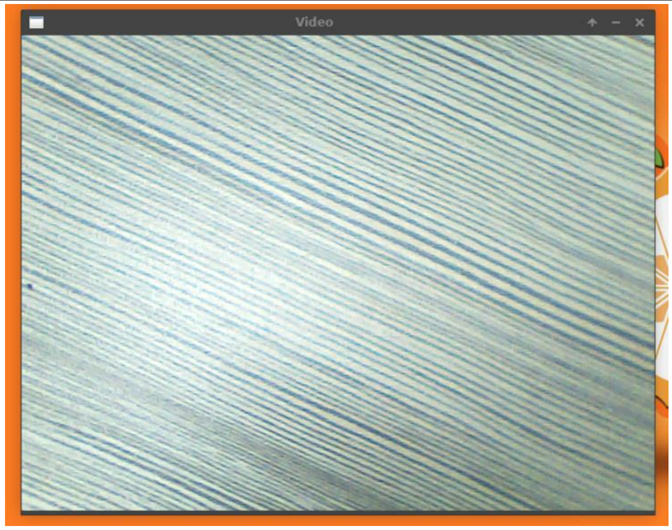
Debian10 needs to modify **cv2.VideoCapture(0)** to **cv2.VideoCapture("/dev/video10")**.

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ vim \
facerec_from_webcam_faster.py
video_capture = cv2.VideoCapture(0)
```

c. Then execute the following command to run **facerec_from_webcam_faster.py**

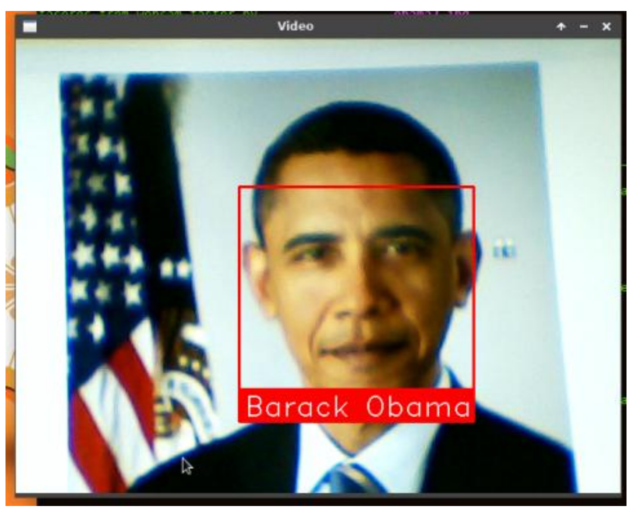
```
orangepi@orangepi:~/face_recognition/examples$ python3 \
facerec_from_webcam_faster.py
```

d. Wait for a while and the display screen of the camera will pop up



e. At this point, you can point the camera at yourself. When the camera detects a face, it will use a box to frame the detected face. **Note that when detecting faces, the picture displayed by the camera will be relatively stuck, please do not move too fast**

f. You can also open a picture of Obama, and then use the camera to aim at the opened picture, you can see that not only the face can be marked, but also the name of the detected face can be displayed correctly. **Note that when detecting faces, the picture displayed by the camera will be relatively stuck, please do not move too fast**



10) **web_service_example.py** is a very simple case of using a web service to upload a picture to run face recognition. The back-end server will identify whether the picture is Obama, and output the recognition result as a json key-value pair. The test steps are as follows :

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and execute the following command (**if it is face_recognition that is automatically installed using a script, then you don't need to install flask**)

```
orangepi@orangepi:~$ python3 -m pip install flask
orangepi@orangepi:~$ cd face_recognition/examples
root@orangepi:~/face_recognition/examples$ python3 web_service_example.py
 * Serving Flask app 'web_service_example' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses (0.0.0.0)
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://127.0.0.1:5001
 * Running on http://192.168.1.79:5001 (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 500-161-390
```

b. Then run the following command to return the result of image recognition (note that the execution path of the following command is **face_recognition/examples**)

```
orangepi@orangepi:~/face_recognition/examples$ curl -XPOST -F \
"file=@obama2.jpg" http://127.0.0.1:5001
{
  "face_found_in_image": true,
  "is_picture_of_obama": true
}
```

c. We can also copy the picture **face_recognition/examples/obama2.jpg** to other Linux computers, of course, we can also prepare a picture named **obama2.jpg** by ourselves, and then use the following commands to remotely remote from the Linux computer Identify the face through the service running on the development board (**note that the IP address in the command needs to be replaced with the IP address of the development board, and the file name after the file needs to be replaced with the name of the image you want to test**)

```
test@test:~$ curl -XPOST -F "file=@obama2.jpg" http://192.168.1.79:5001
{
  "face_found_in_image": true,
  "is_picture_of_obama": true
}
```

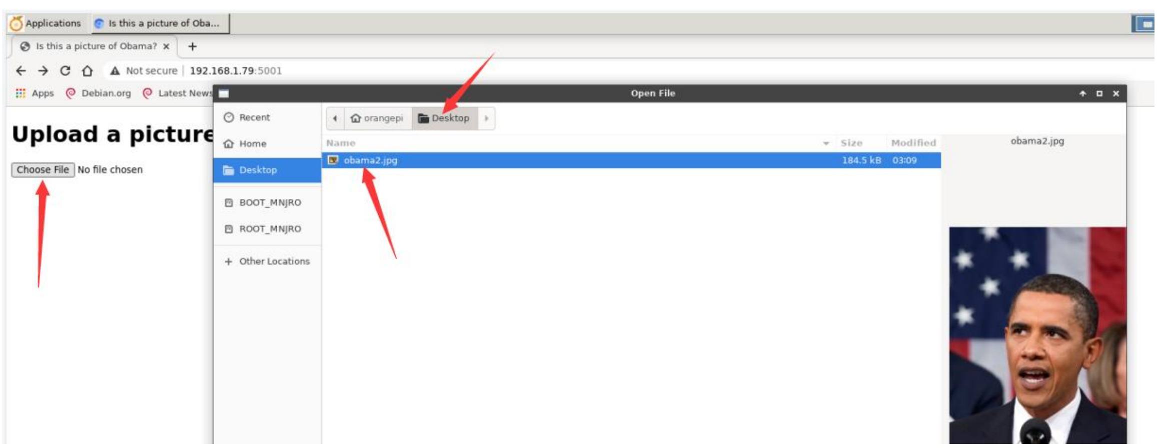
d. The method of using the browser test is as follows:

a) First open the browser, then enter **the IP address of the development board: 5001** in the address bar of the browser, and then you can see the following page

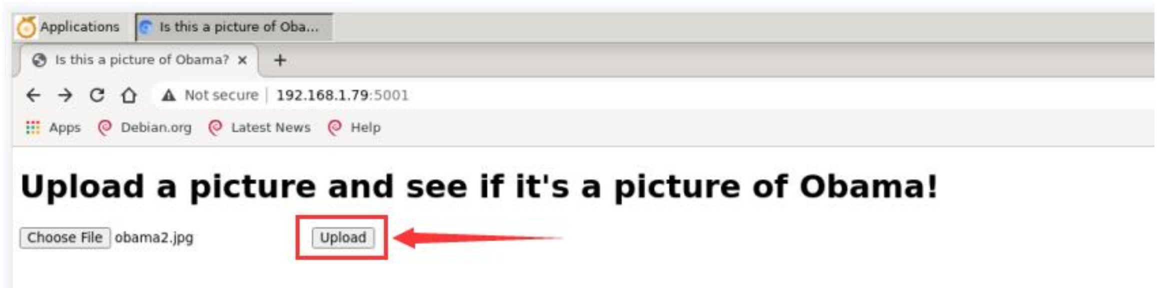
b) Then copy obama2.jpg to the desktop

```
orangepi@orangepi:~/face_recognition/examples$ cp obama2.jpg \
/home/orangepi/Desktop/
```

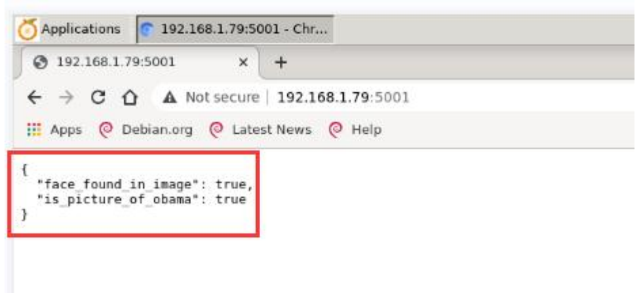
c) Then select the image you just copied in the browser



d) Then click **Upload** to upload the image you just selected for face recognition



e) After a period of time, the test results will be displayed



11) **face_detection** command test example

a. The face_detection command-line tool can locate faces (output pixel coordinates) in a single image or in a folder of images. Use **face_detection --help** to view the help information of the face_detection command

```
orangepi@orangepi:~$ face_detection --help
Usage: face_detection [OPTIONS] IMAGE_TO_CHECK

Options:
  --cpus INTEGER    number of CPU cores to use in parallel. -1 means "use all in
                    system"
  --model TEXT      Which face detection model to use. Options are "hog" or
                    "cnn".
  --help            Show this message and exit.
```

b. An example of detecting a single image is as follows:

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ face_detection obama2.jpg
obama2.jpg,302,474,611,164
```

c. An example of using multiple cores to detect multiple images in parallel is as follows:

a) First go to the **face_recognition/examples** folder

b) Then create a new test folder

c) Then copy the jpg images to the test folder

d) Then use all cpus to run **face_detection** in parallel to check the pictures in the test folder, where **--cpus -1** means use all cpus

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ mkdir test
orangepi@orangepi:~/face_recognition/examples$ cp *.jpg test
orangepi@orangepi:~/face_recognition/examples$ face_detection --cpus -1 test
test/obama-240p.jpg,29,261,101,189
test/obama_small.jpg,65,215,169,112
test/obama2.jpg,302,474,611,164
test/two_people.jpg,62,394,211,244
test/two_people.jpg,95,941,244,792
test/obama.jpg,136,624,394,366
test/obama-480p.jpg,65,507,189,383
test/obama-720p.jpg,94,751,273,572
test/obama-1080p.jpg,136,1140,394,882
test/biden.jpg,233,749,542,439
```

12) **face_recognition** command test example

a. **face_recognition** command line tool can recognize whose face is in a single image or a folder of images. Use **face_recognition --help** to view help information for the face_recognition command

```
orangepi@orangepi:~$ face_recognition --help
Usage: face_recognition [OPTIONS] KNOWN_PEOPLE_FOLDER
IMAGE_TO_CHECK

Options:
  --cpus INTEGER              number of CPU cores to use in parallel (can speed
                              up processing lots of images). -1 means "use all in
                              system"
  --tolerance FLOAT          Tolerance for face comparisons. Default is 0.6.
                              Lower this if you get multiple matches for the same
                              person.
  --show-distance BOOLEAN    Output face distance. Useful for tweaking tolerance
                              setting.
```

```
--help                              Show this message and exit.
```

b. First create a new face picture folder **known_people** with a known name, then copy two pictures to **known_people**, and then copy **obama2.jpg** as **unknown.jpg**, which is the picture we want to identify

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ mkdir known_people
orangepi@orangepi:~/face_recognition/examples$ cp biden.jpg obama.jpg \
known_people
orangepi@orangepi:~/face_recognition/examples$ cp obama2.jpg unkown.jpg
```

c. Then you can use the following command to identify the name of the character in the **unkown.jpg** picture, you can see that the unknown.jpg picture is recognized as obama

```
orangepi@orangepi:~/face_recognition/examples$ face_recognition known_people \
unkown.jpg
unkown.jpg,obama
```

d. If we identify an irrelevant picture, unknown_person will be displayed

```
root@orangepi:~/face_recognition/examples$ face_recognition known_people \
alex-lacamoire.png
alex-lacamoire.png,unknown_person
```

e. We can also create a new test folder, and then put multiple pictures in it, and then we can use all the CPUs to recognize all the pictures in parallel

```
orangepi@orangepi:~/face_recognition/examples$ mkdir test
orangepi@orangepi:~/face_recognition/examples$ cp *.jpg *.png test
orangepi@orangepi:~/face_recognition/examples$ face_recognition --cpus -1 \
known_people test
test/obama-240p.jpg,obama
test/alex-lacamoire.png,unknown_person
test/obama_small.jpg,obama
test/unkown.jpg,obama
test/obama2.jpg,obama
test/lin-manuel-miranda.png,unknown_person
test/two_people.jpg,biden
test/two_people.jpg,obama
test/obama-720p.jpg,obama
test/obama.jpg,obama
```

```
test/obama-480p.jpg,obama
test/biden.jpg,biden
test/obama-1080p.jpg,obama
```

3. 47. QT installation method

3. 47. 1. How to install QT5

1) The command to install QT5 is as follows

   a. The installation commands for **Debian10** and **Ubuntu20.04** are:

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get -y install qt5-default \
qttools5-dev-tools qtbase5-doc-html qt5-assistant qt5-doc
```

   b. The installation commands for **Debian11** and **Ubuntu22.04** are:

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get -y install qttools5-dev-tools qtbase5-doc-html \
qt5-assistant qt5-doc qt5-qmake qt5-qmake-bin
```

2) After installation, use the following command to view the version number of QT

   a. The default version of Ubuntu 20.04 is:

```
orangepi@orangepi:~$ qmake -v
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/aarch64-linux-gnu
```

   b. The default version of Ubuntu 22.04 is:

```
orangepi@orangepi:~$ qmake -v
QMake version 3.1
Using Qt version 5.15.3 in /usr/lib/aarch64-linux-gnu
```

   c. The default version for Debian 10 is:

```
orangepi@orangepi:~$ qmake -v
QMake version 3.1
Using Qt version 5.11.3 in /usr/lib/aarch64-linux-gnu
```

   d. The default version for Debian 11 is:

```
orangepi@orangepi:~$ qmake -v
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

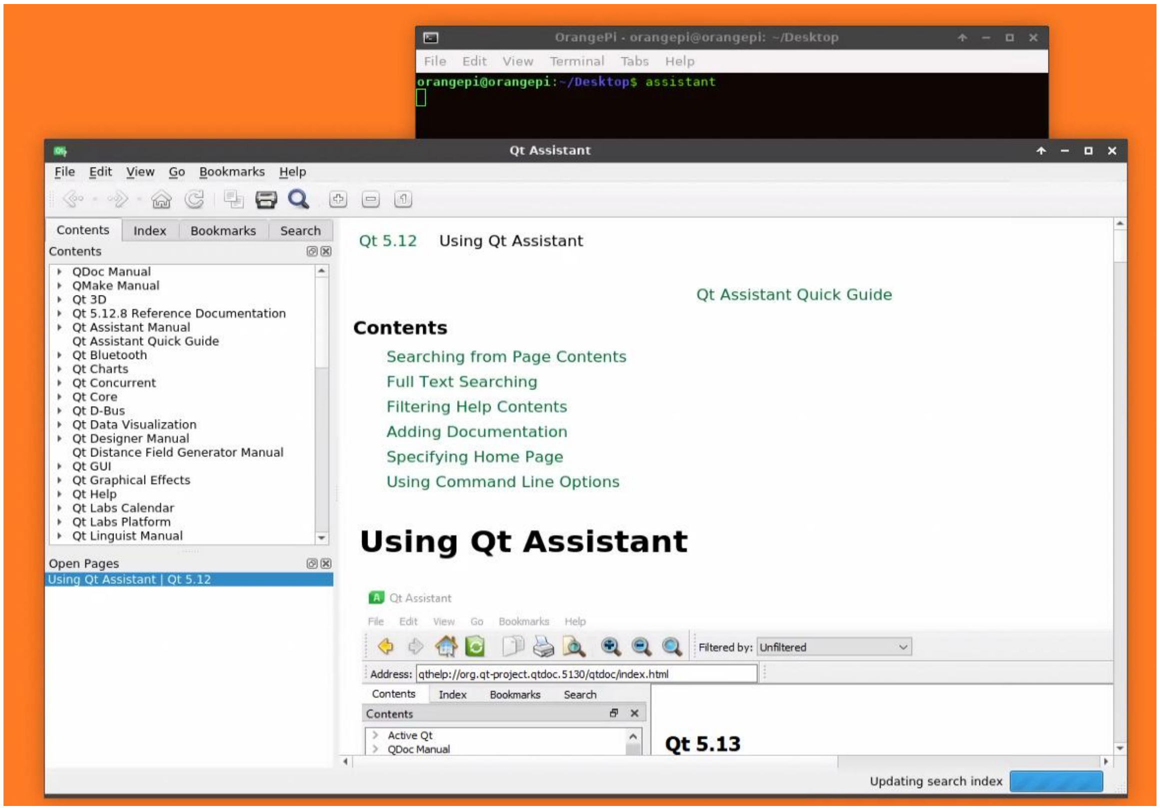
3) The command to view the QT installation path is as follows (Ubuntu 20.04 as an example)

```
orangepi@orangepi:~$ qmake -qt5 -query
QT_SYSROOT:
QT_INSTALL_PREFIX:/usr
QT_INSTALL_ARCHDATA:/usr/lib/aarch64-linux-gnu/qt5
QT_INSTALL_DATA:/usr/share/qt5
QT_INSTALL_DOCS:/usr/share/qt5/doc
QT_INSTALL_HEADERS:/usr/include/aarch64-linux-gnu/qt5
QT_INSTALL_LIBS:/usr/lib/aarch64-linux-gnu
QT_INSTALL_LIBEXECS:/usr/lib/aarch64-linux-gnu/qt5/libexec
QT_INSTALL_BINS:/usr/lib/qt5/bin
QT_INSTALL_TESTS:/usr/tests
QT_INSTALL_PLUGINS:/usr/lib/aarch64-linux-gnu/qt5/plugins
QT_INSTALL_IMPORTS:/usr/lib/aarch64-linux-gnu/qt5/imports
QT_INSTALL_QML:/usr/lib/aarch64-linux-gnu/qt5/qml
QT_INSTALL_TRANSLATIONS:/usr/share/qt5/translations
QT_INSTALL_CONFIGURATION:/etc/xdg
QT_INSTALL_EXAMPLES:/usr/lib/aarch64-linux-gnu/qt5/examples
QT_INSTALL_DEMOS:/usr/lib/aarch64-linux-gnu/qt5/examples
QT_HOST_PREFIX:/usr
QT_HOST_DATA:/usr/lib/aarch64-linux-gnu/qt5
QT_HOST_BINS:/usr/lib/qt5/bin
QT_HOST_LIBS:/usr/lib/aarch64-linux-gnu
QMAKE_SPEC:linux-g++
QMAKE_XSPEC:linux-g++
QMAKE_VERSION:3.1
QT_VERSION:5.12.8
```

4) Use the following command to open QT assistant

```
orangepi@orangepi:~$ assistant
```

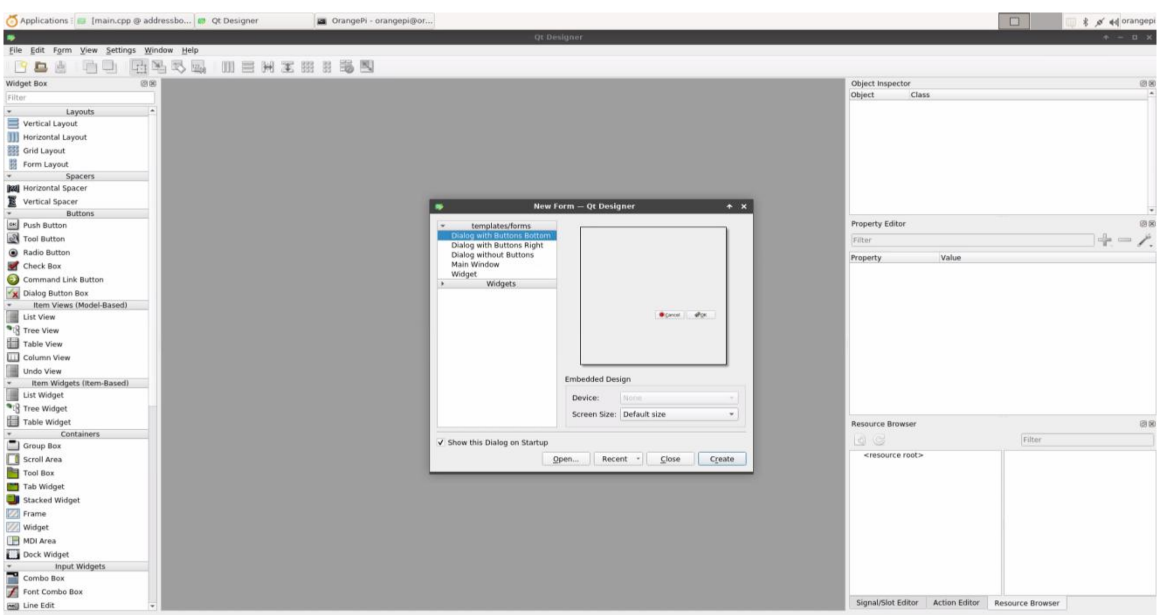
The interface after QT assistant is opened is as follows



5) Use the following command to open QT designer

orangepi@orangepi:~$ **designer**

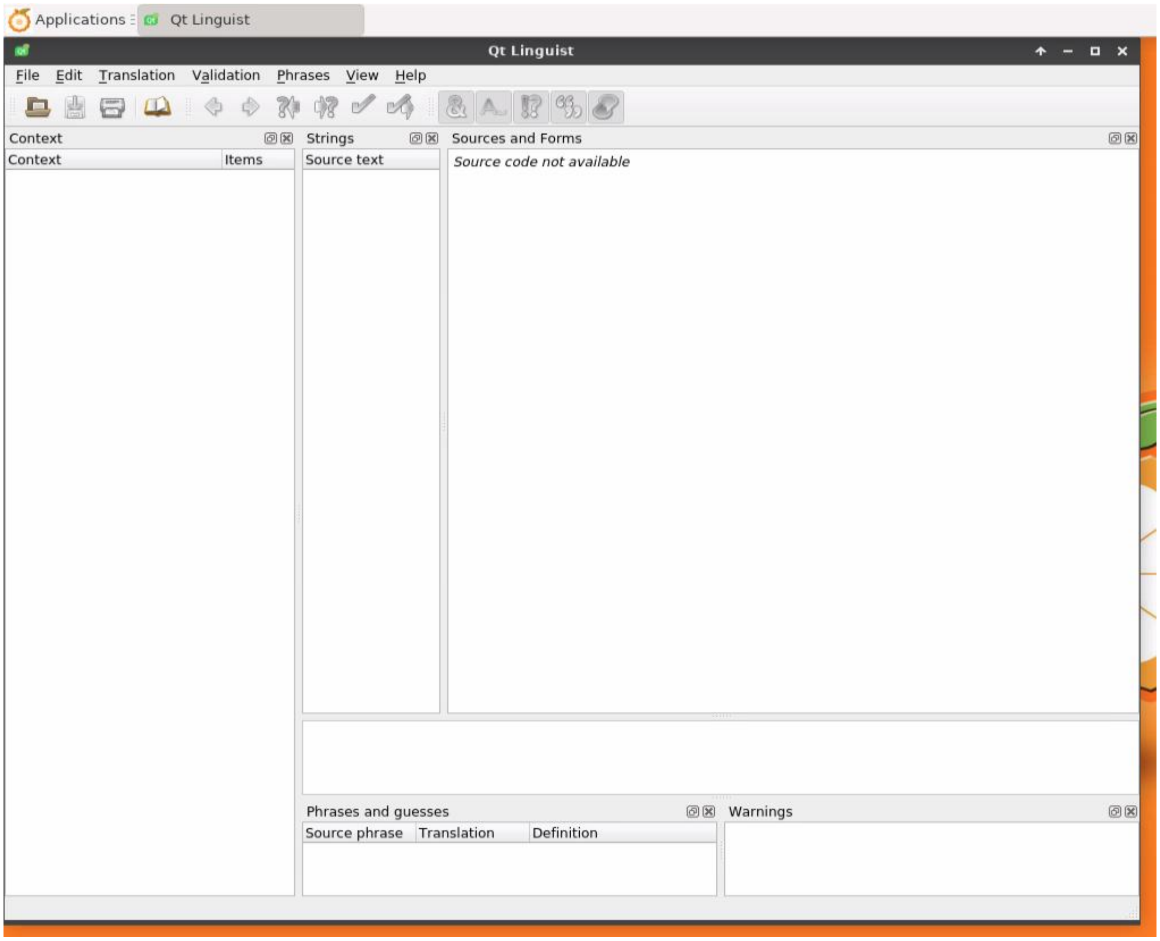
The interface after QT designer is opened is as follows



6) Use the following command to open QT linguist

```
orangepi@orangepi:~$ linguist
```

The interface after QT linguist is opened is as follows

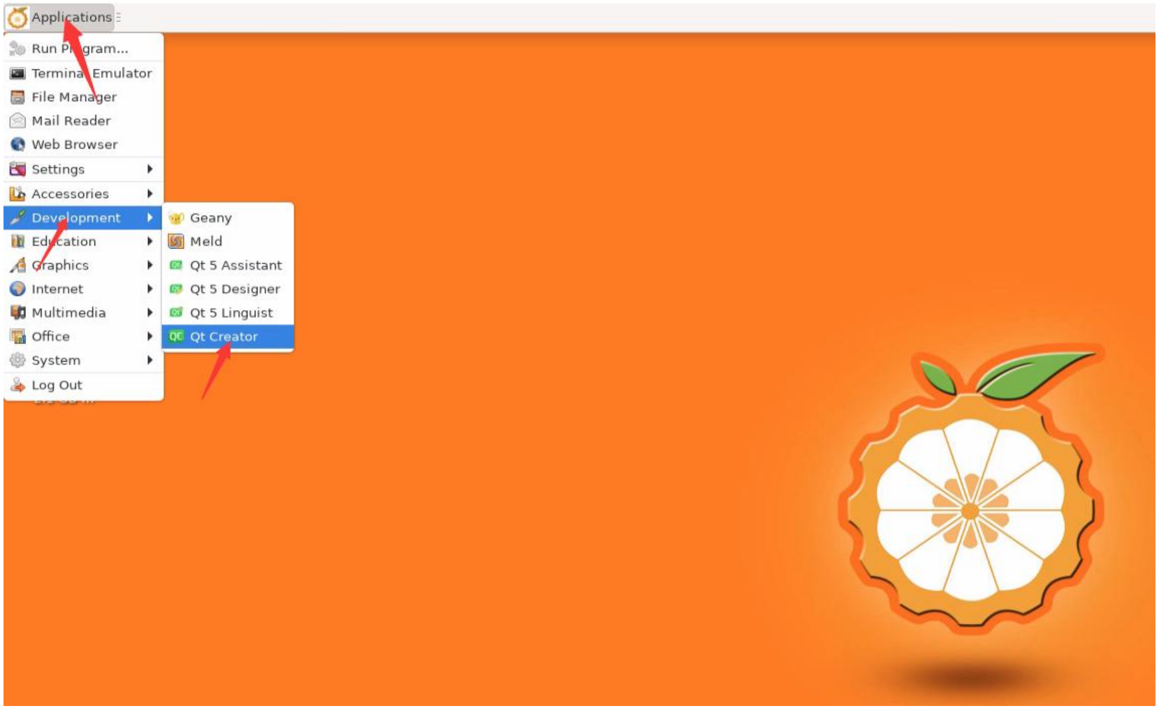


3. 47. 2. How to install QT Creator

1) The command to install QT Creator is as follows

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt-get -y install qtcreator qmlscene gdb    \
qtdeclarative5-dev qtbase5-examples
```

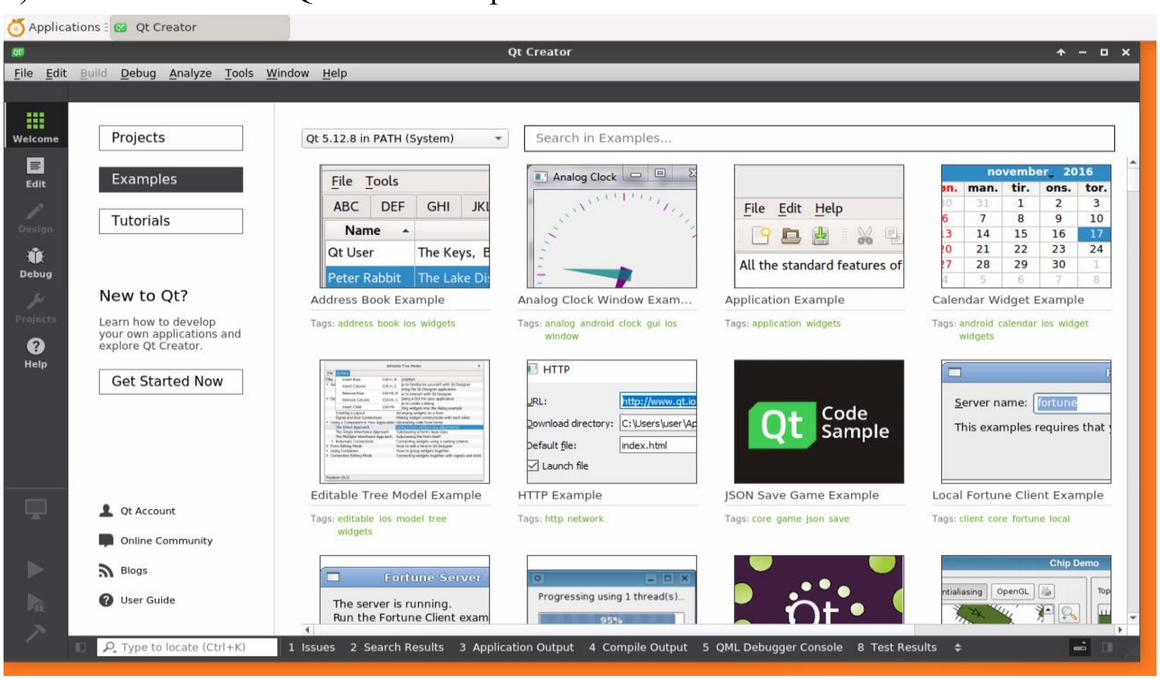
2) After QT Creator is installed, you can see the startup icon of QT Creator in **Applications**



You can also open QT Creator with the following command

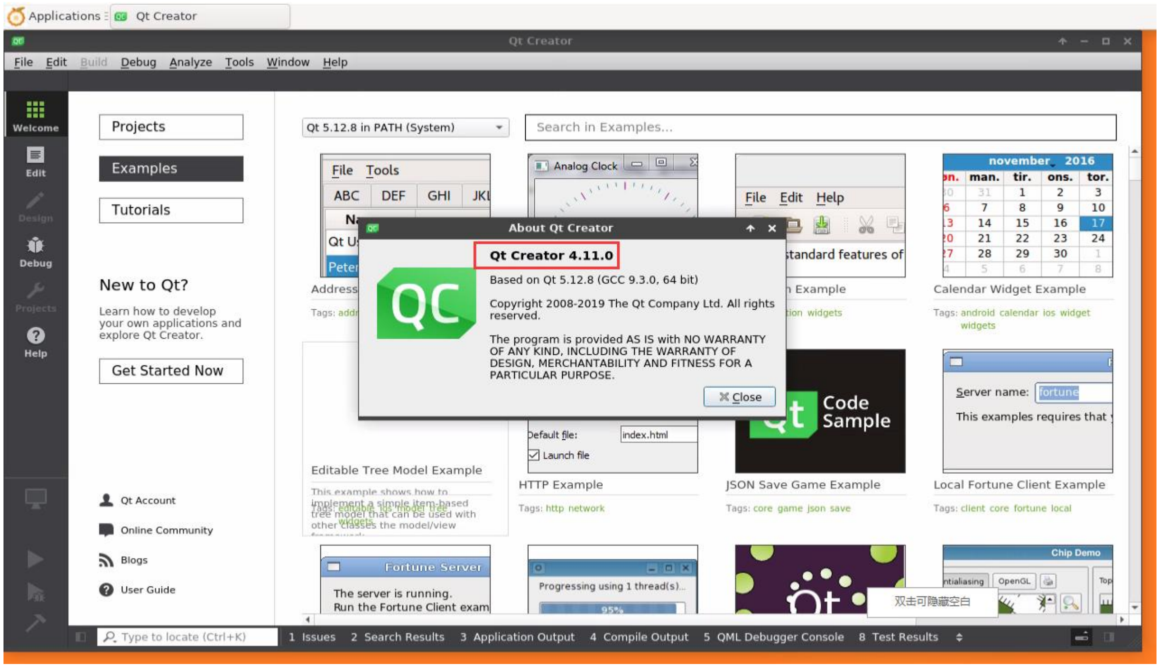
```
orangepi@orangepi:~$ qtcreator
```

3) The interface after QT Creator is opened is as follows



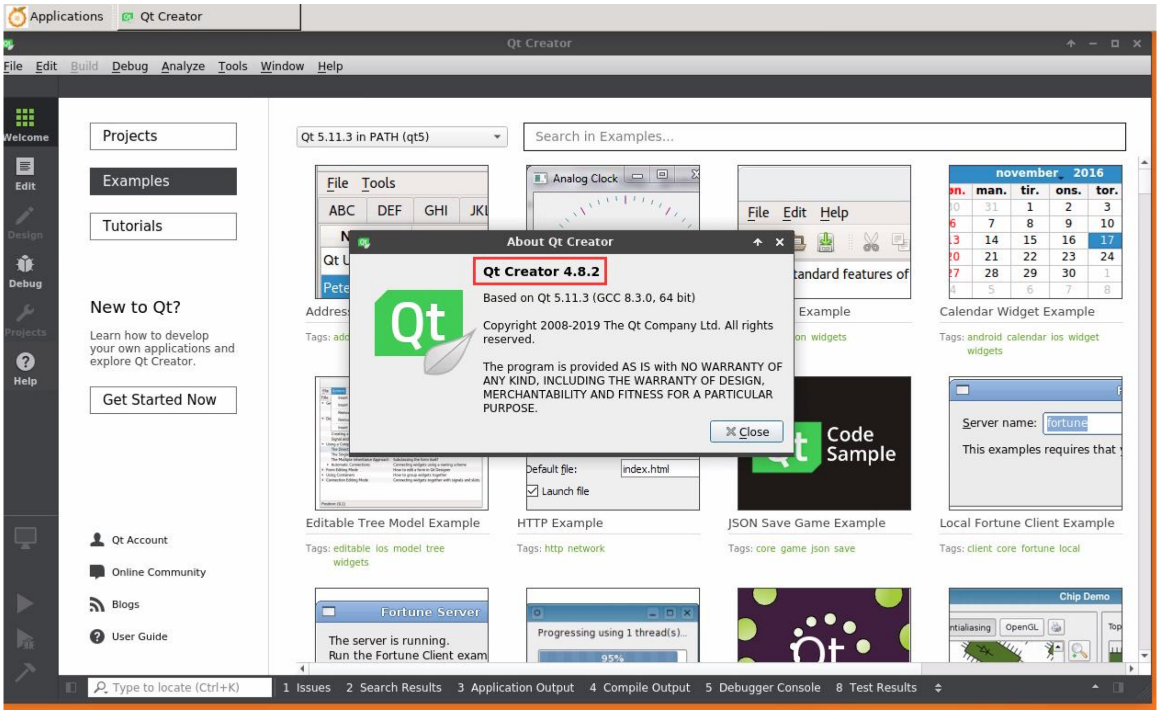
4) The version of QT Creator is as follows

a. The default version of QT Creator in **Ubuntu20.04** is as follows

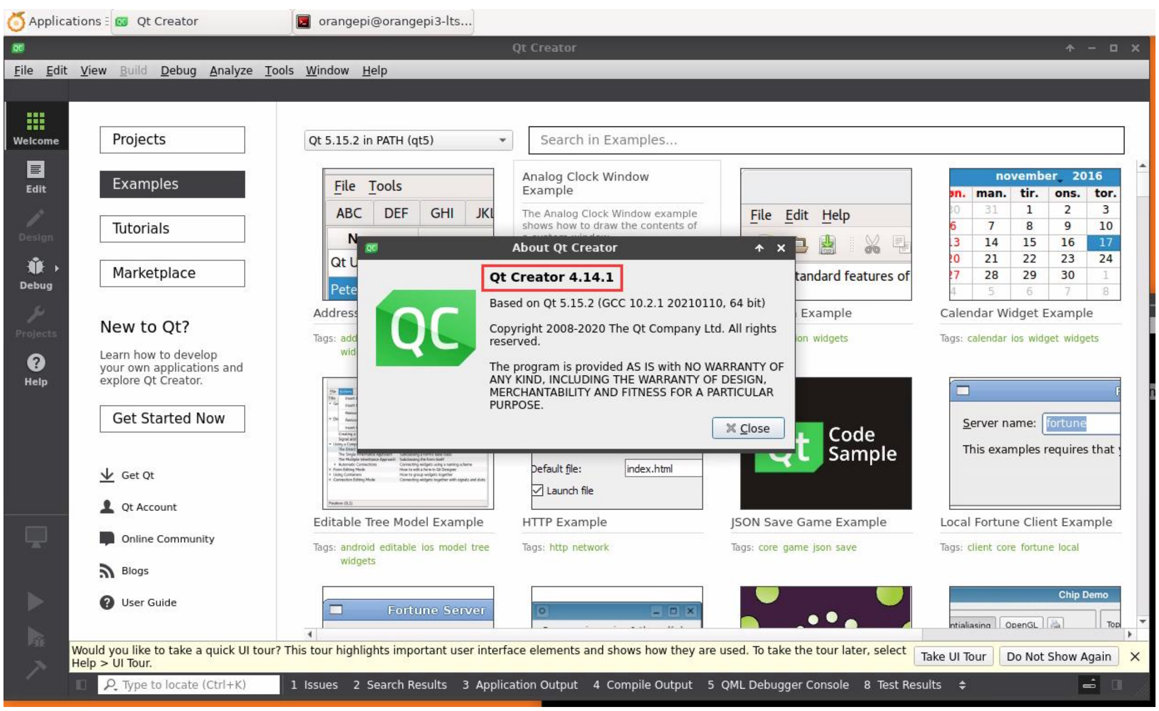


b. The default version of QT Creator in **Ubuntu22.04** is as follows

c. The default version of QT Creator in **Debian10** is as follows



d. The default version of QT Creator in **Debian11** is as follows



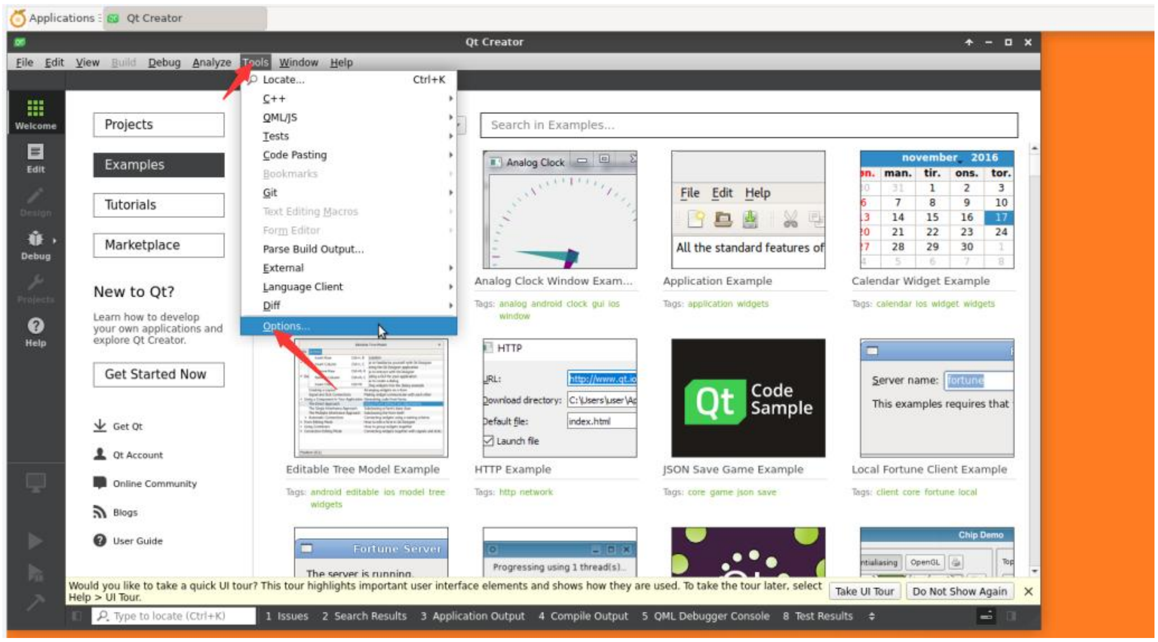
5) Let's test the sample code that comes with QT. Before the test, you need to modify the owner and group of the **examples** folder to the **orangepi** user

```
orangepi@orangepi:~$ sudo chown orangepi:orangepi \
```

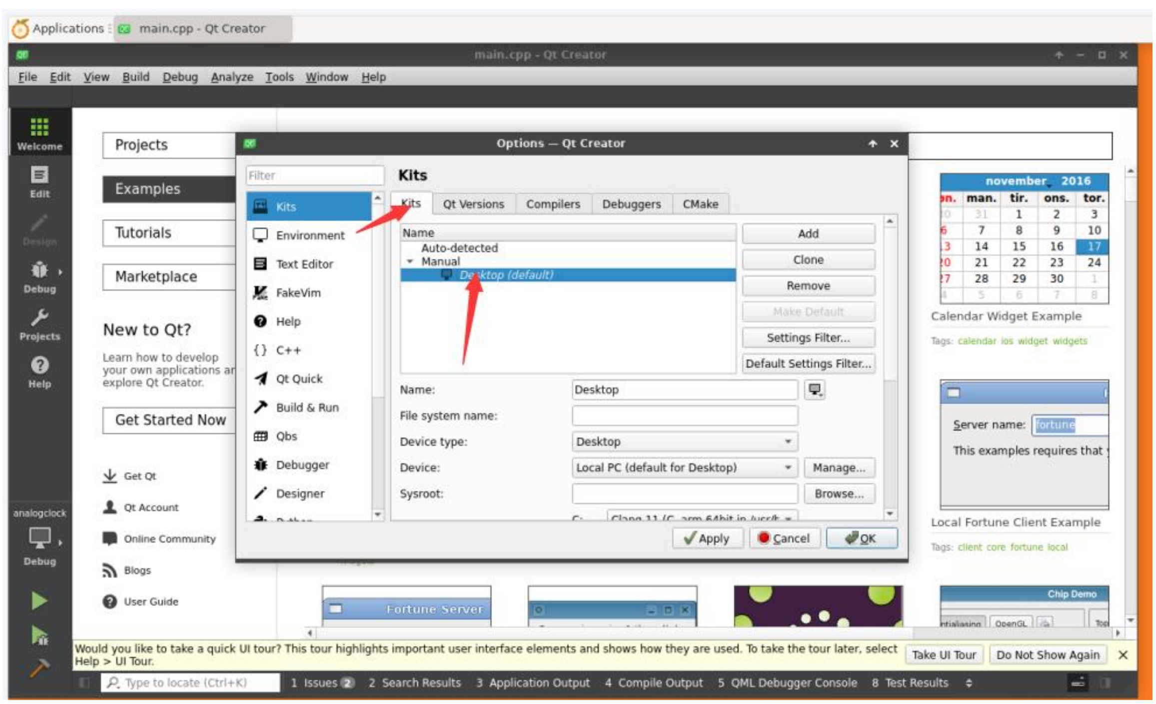
```
/usr/lib/aarch64-linux-gnu/qt5/examples -R
```

6) Also make sure that the compiler used for C and C++ is GCC

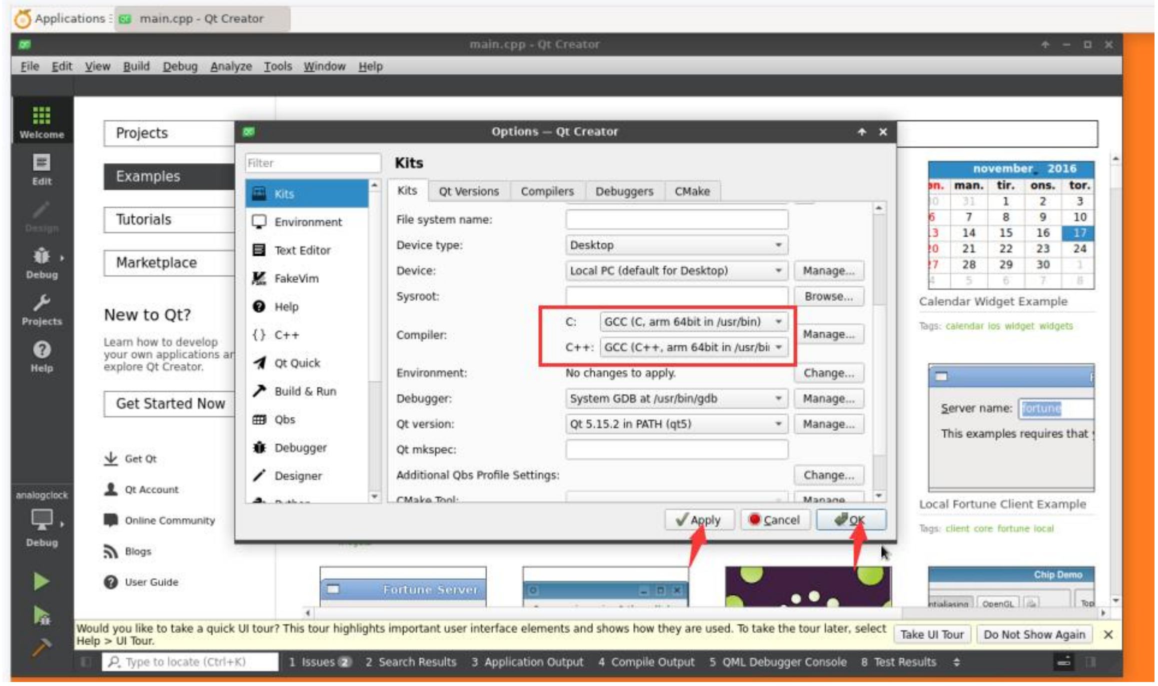
   a. first open **Tools->Options...**



   b. then select **Kits->Desktop(default)**

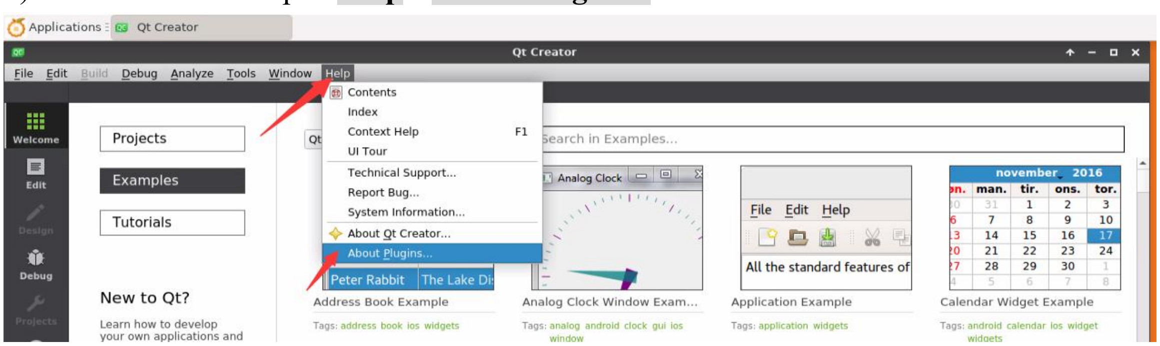


   c. Then scroll down, find the **Compile** option, and then select the C and C++ compiler as GCC

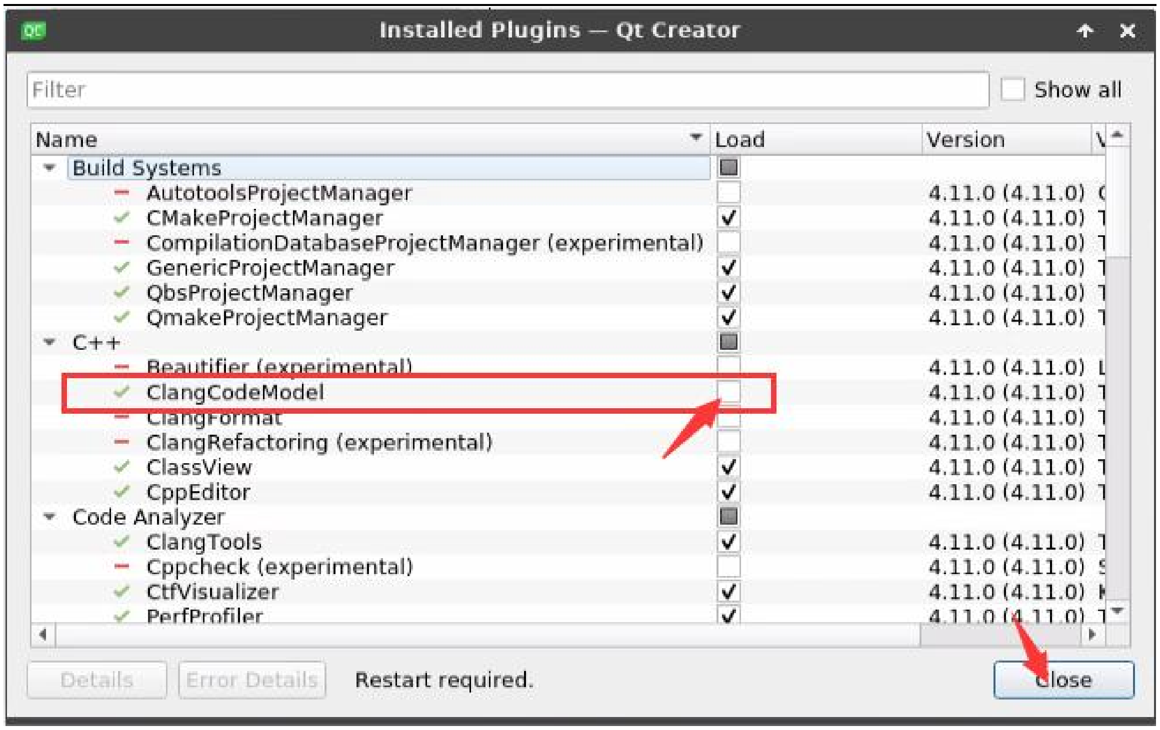


d. Then click **Apply**, and then click **OK** to confirm

7) You also need to open **Help->About Plugins...**

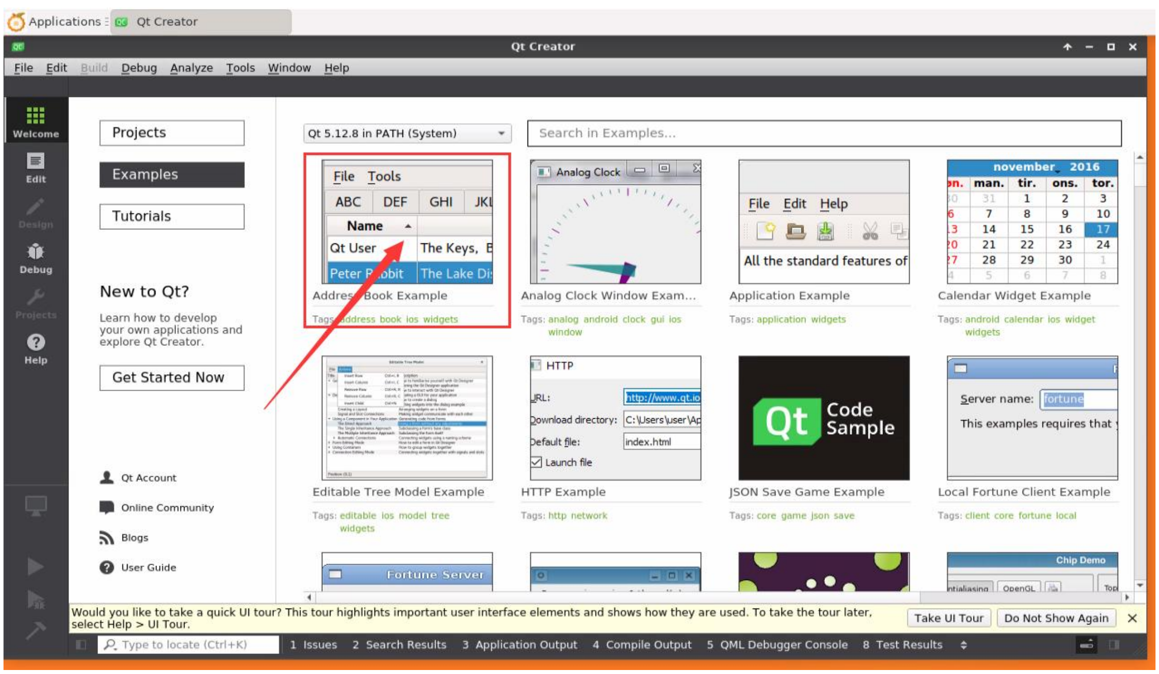


Then remove the tick of **ClangCodeModel**

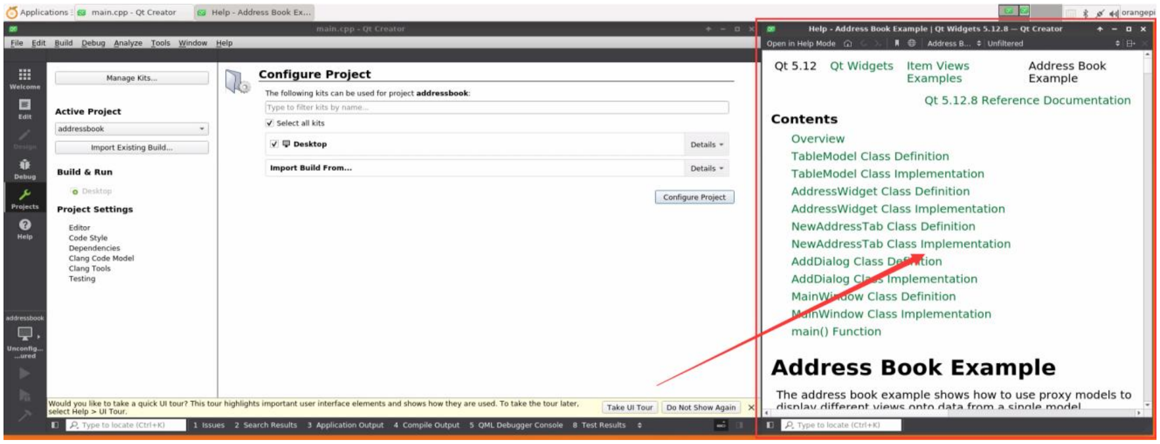


8) **After setting, you need to restart QT Creator**

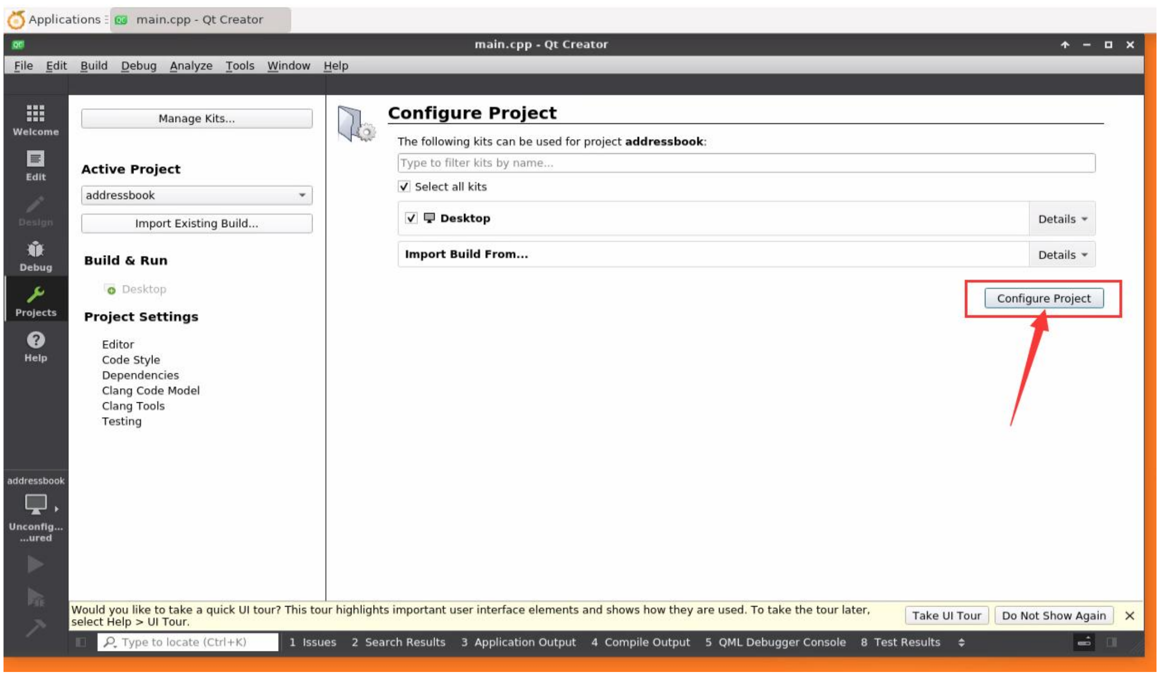
9) Then you can open a sample code



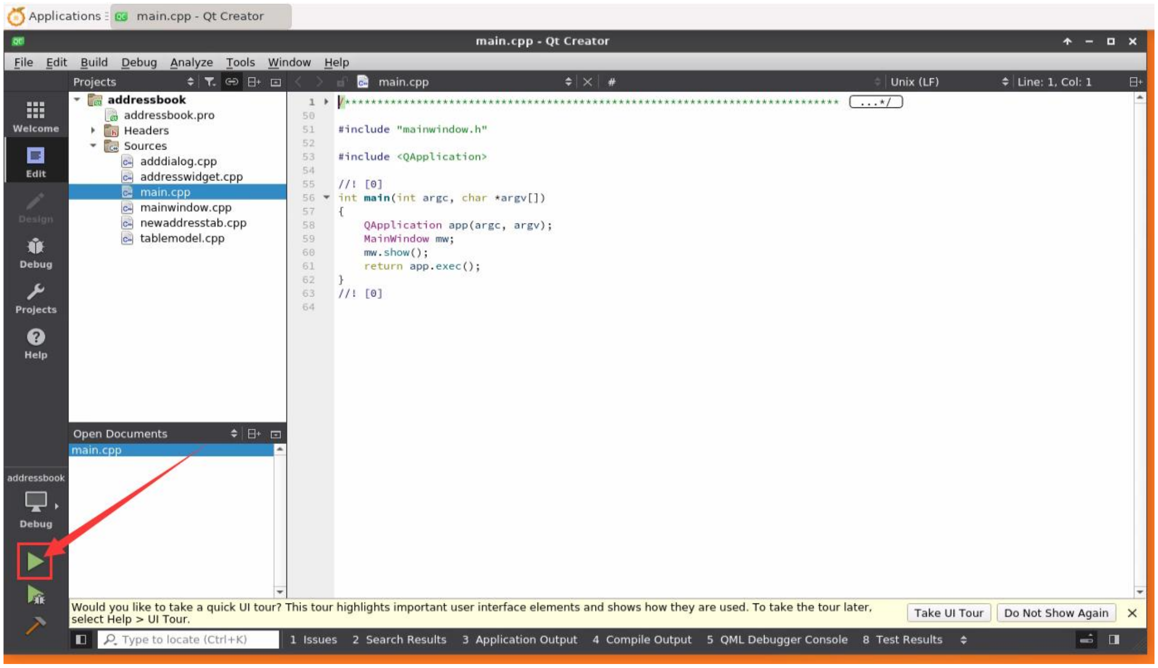
10) After clicking the sample code, the corresponding documentation will be automatically opened, and you can read the instructions carefully.



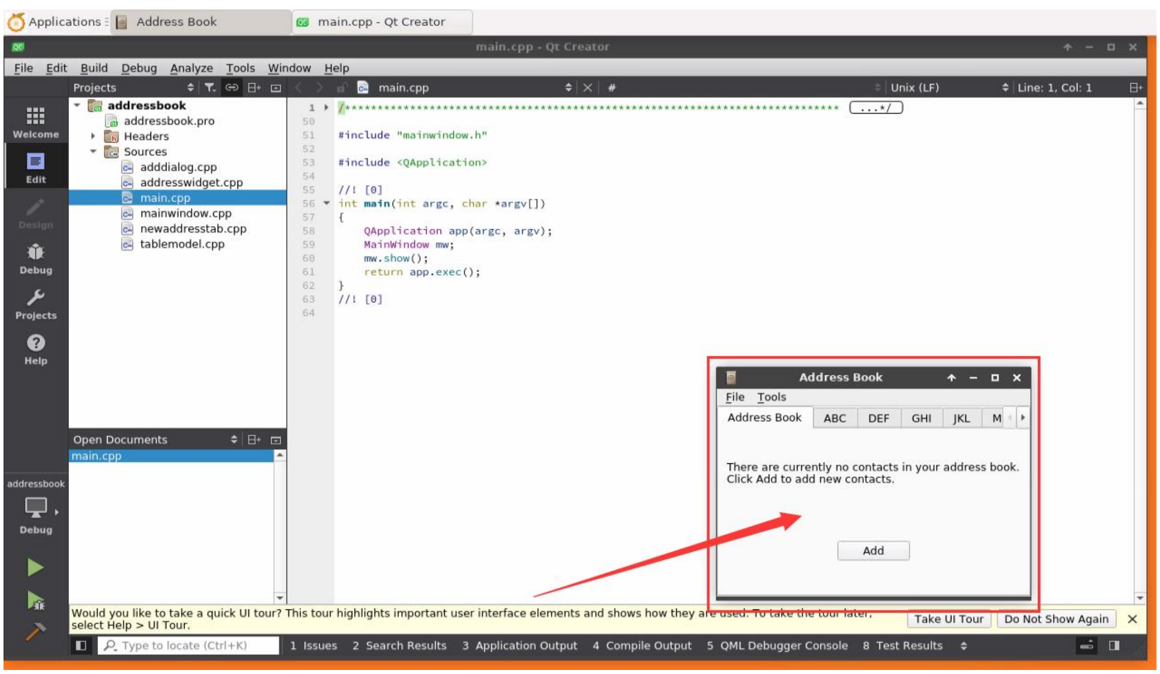
11) Then click on **Configure Project**



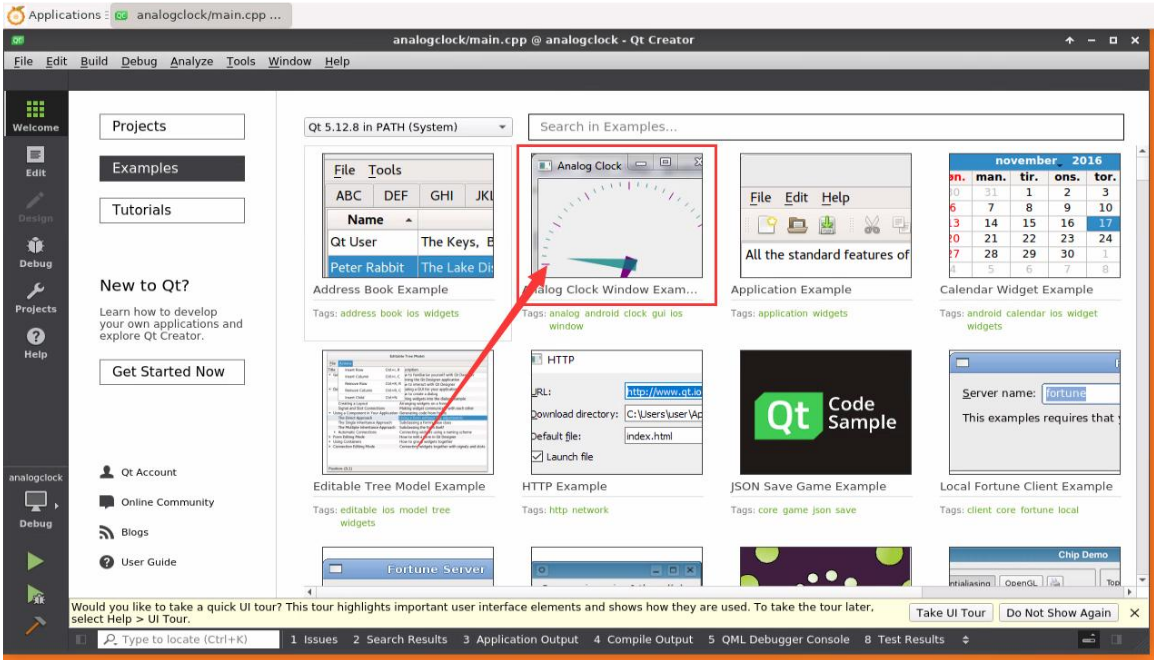
12) Then click the green triangle in the lower left corner to compile and run the sample code

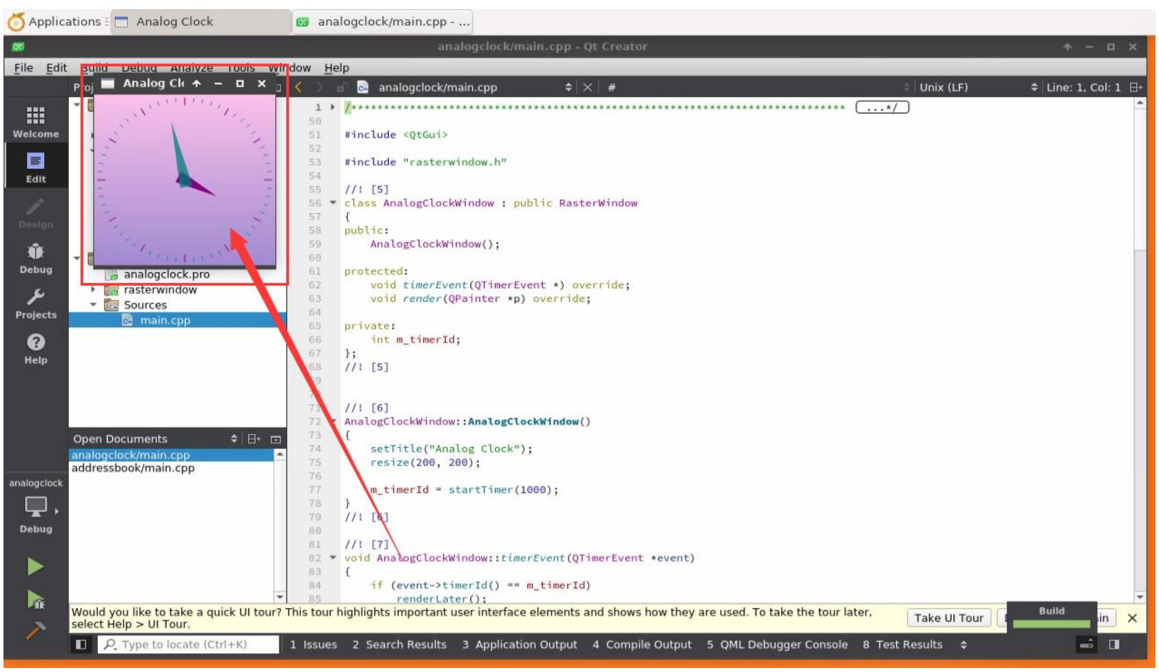


13) After waiting for a period of time, the interface shown in the figure below will pop up, which means that QT can compile and run normally



14) The running result of the analogclock sample code is as follows





15) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu
https://download.qt.io/archive/qtcreator
https://download.qt.io/archive/qt

3. 48. Reset and shutdown methods

1) During the operation of the Linux system, if the power is directly unplugged, some data may be lost in the file system. It is recommended to use the poweroff command to shut down the Linux system of the development board before powering off, and then unplug the power.

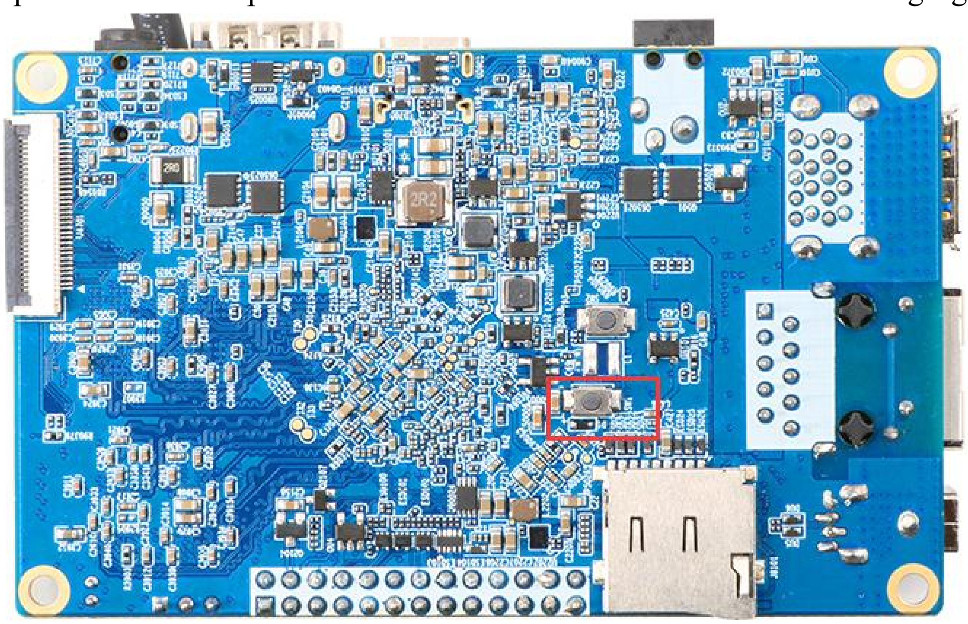
```
orangepi@orangepi:~$ sudo poweroff
```

2) **After turning off the development board, you need to re-plug the power supply to turn it on.**

3) Use the reboot command to restart the Linux system on the development board.

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

4) You can also short press the power button on the development board to reset the development board. The position of the reset button is shown in the following figure:



4. Instructions for use of Android system

4. 1. Supported Android Versions

Android version	kernel version
Android 8.1	linux4.4

4. 2. Android 8.1 function adaptation

Function	status
HDMI video	OK
HDMI audio	OK
USB2.0 x 2	OK
USB3.0 x 1	OK
TypeC USB3.0	OK
TF card boot	OK
EMMC start	OK
network card	OK
WIFI	OK
Bluetooth	OK
Bluetooth earphone	OK
headphone recording	OK
Headphone playback	OK
microphone recording	OK
LED lights	OK
Temperature Sensor	OK
USB camera	OK
GPU	OK
Video codec	OK
Reset button	OK
upgrade key	OK

ADB debugging	**OK**
OV13850 camera	**OK**
10.1 inch MIPI screen	**OK**
mini-PCIE	**OK**
TypeC to HDMI	**OK**

4. 3. On-board LED light display description

	green light	**red light**
u-boot startup phase	**off**	**on**
The kernel boots into the system	**flash**	**on**

4. 4. How to use ADB

4. 4. 1. Use the data cable to connect adb debugging

1) First prepare a good quality Type-C data cable. It is recommended to use the white port Type-C 2.0 data cable shown on the left. The purple port fast charging cable shown on the right cannot be used on the OrangePi 4 LTS development board. Normal use

2) First, you need to use the Type-C data cable to connect the development board to the USB interface of the computer (please use the DC power supply to power the development board at the same time)

3) Install adb tool on Ubuntu PC

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

4) Check to identify the ADB device

```
test@test:~$ adb devices
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

5) Then you can log in to the android system through adb shell on the Ubuntu PC

```
test@test:~$ adb shell
rk3399_mid:/ $
```

6) If you need to change the Android system files, you need to close the security verification, execute the following command

```
test@test:~$ adb root
test@test:~$ adb disable-verity
```

7) Then execute the command to restart the system

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

8) Remount the Android system

```
test@test:~$ adb root
test@test:~$ adb remount
```

9) Then you can transfer files to the Android system

```
test@test:~$ adb push example.txt /system/
```

4. 4. 2. Using network connection adb debugging

Using network adb does not require the USB Typc C interface data cable to connect the computer and the development board, but communicates 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, and then to use

1) Make sure that the **service.adb.tcp.port** of the Android system is set to the port number 5555

```
rk3399_mid:/ # 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

```
rk3399_mid:/ # setprop service.adb.tcp.port 5555
rk3399_mid:/ # stop adbd
rk3399_mid:/ # 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
```

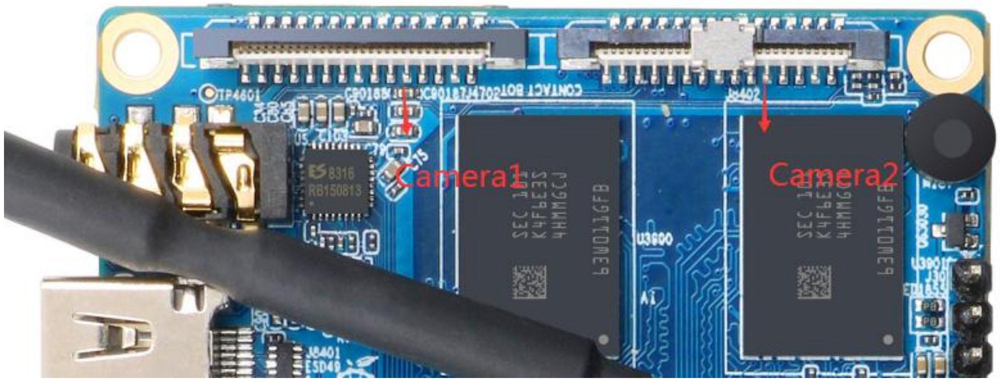
5) Then you can log in to the android system through adb shell on the Ubuntu PC

```
test@test:~$ adb shell
rk3399_mid:/ #
```

4. 5. How to use OV13850 camera

1) OrangePi 4 LTS has two Camera ports and both support OV13850 camera. The two Canera interfaces can use one of the interfaces alone, or you can use the two Camera

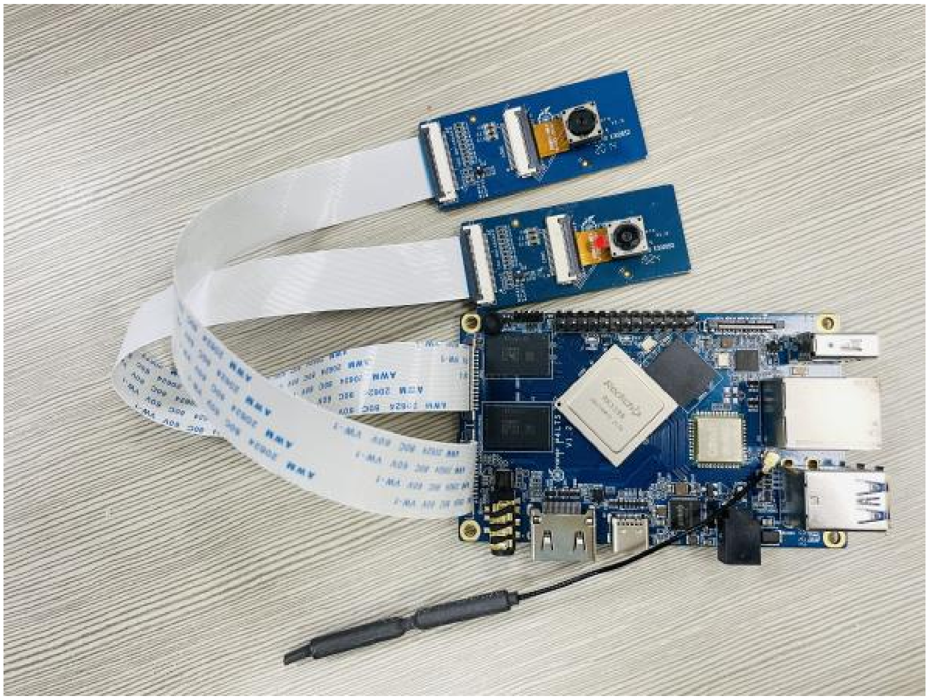
interfaces to connect two cameras at the same time. After connecting two cameras, one is front and one is rear



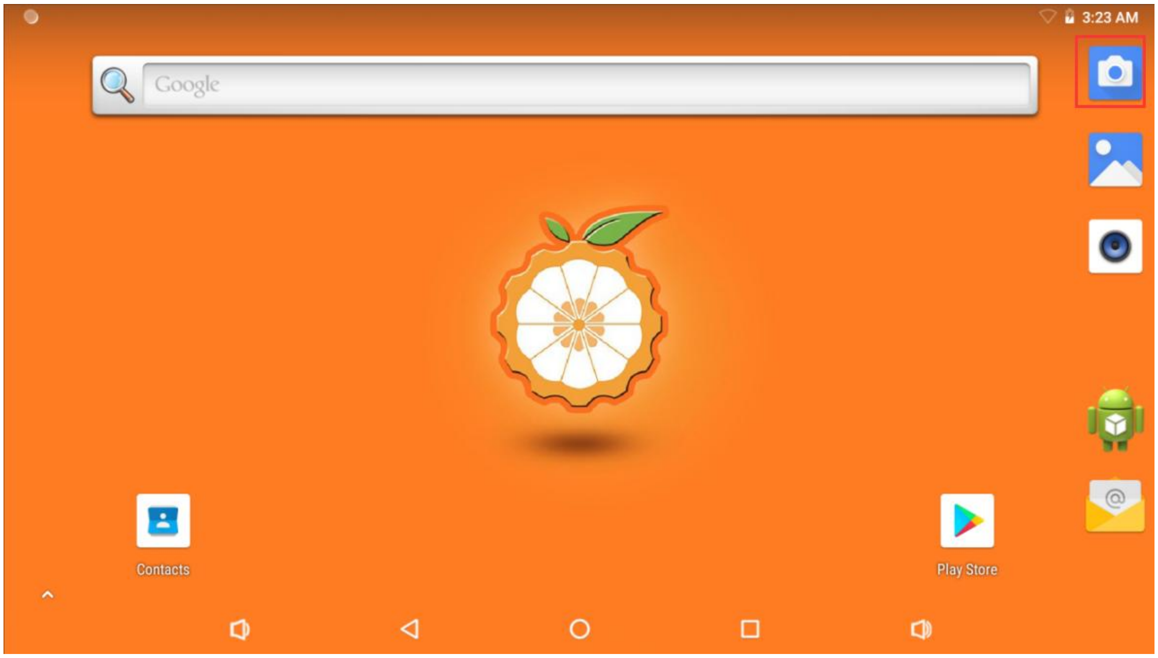
2) The OV13850 camera kit includes an OV13850 camera, an adapter board and a cable

3) First insert the OV13850 camera into the adapter board, and then insert the cable into another card slot of the adapter board

4) Then insert the other end of the cable into the Camera camera interface of the development board. The interface can be connected to two cameras at the same time, or one camera can be connected separately. After connecting the camera, start the Android system (**do not plug in the camera after power on**)

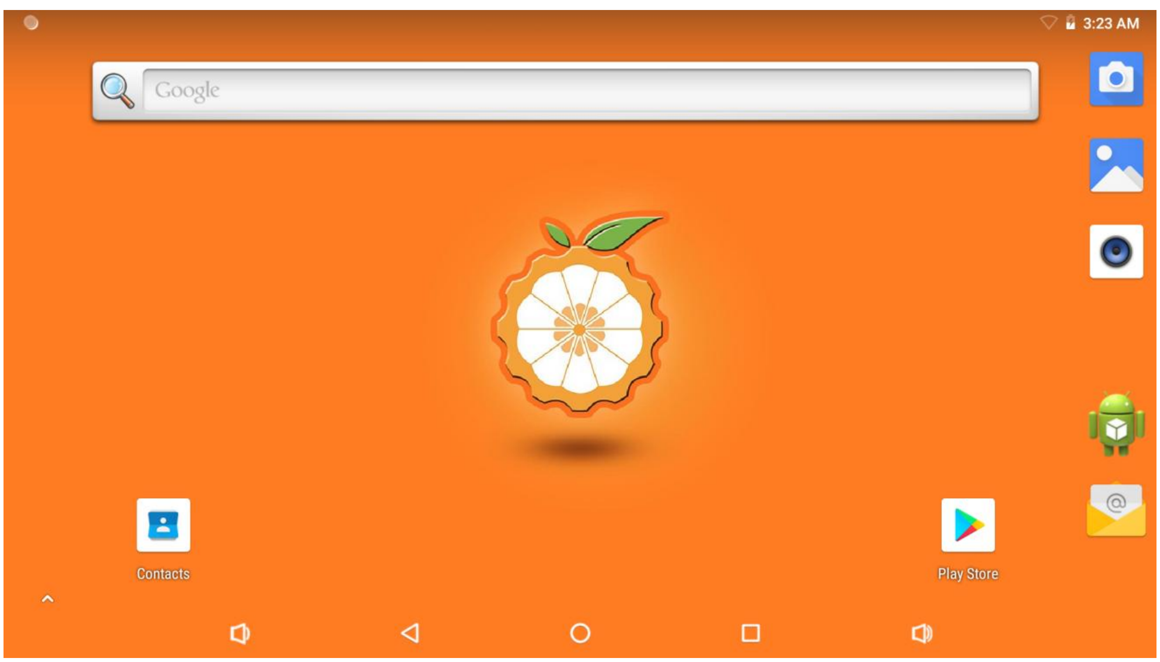


5) After the Android system is started, open the camera APP and you can see the output of the OV13850 camera. The location of the camera APP is shown in the following figure:



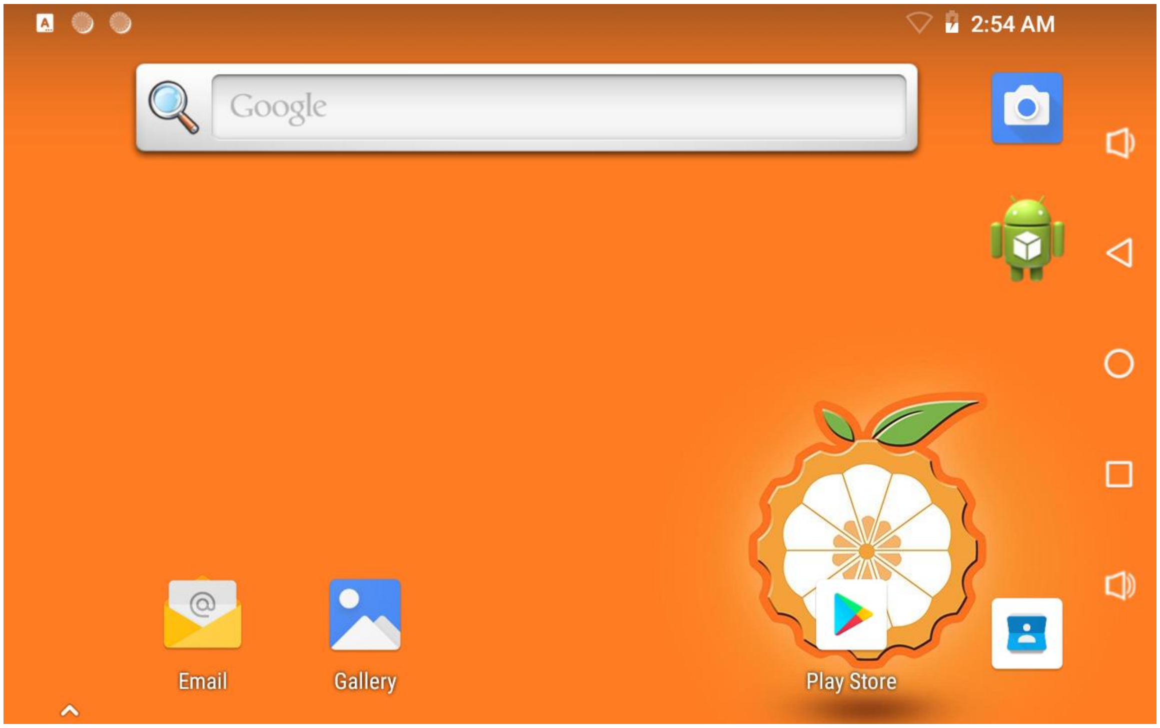
4. 6. How to use the HDMI interface

1) Connect the development board to the HDMI display or TV through the HDMI to HDMI cable. After the Android system starts, the interface is as shown below



2) If the image with the word "LCD" is burned, after the system is started, the interface

will be as shown below



4. 7. The method of displaying the system interface through the TypeC interface

1） Prepare a Type-C to HDMI cable, and connect the development board to an HDMI display or TV through the Type-C interface for display

4. 8. 10.1 inch MIPI screen usage

At present, the MIPI LCD interface supports two types of 10.1-inch MIPI screens. The model information can be seen on the back of the screen, as shown in the following figure:

The one on the left is the MIPI screen with model AFJ101BA2131 currently on sale, and the one on the right is the discontinued MIPI screen with model SL101PN27D1665

If you are using a MIPI screen with model AFJ101BA2131, please make sure that the image used is the image of the following two versions:

OrangePi_4-lts_Android8.1_LCD_AFJ101_800x1280_v1.x.tar.gz

OrangePi_4-lts_SD_Android8.1_LCD_AFJ101_800x1280_v1.x.tar.gz

If you are using a MIPI screen with model number SL101PN27D1665, please make sure that the image used is the image of the following two versions:

Orangepi4-lts_Android8.1_LCD_v1.1.tar.gz

Orangepi4-lts_SD_Android8.1_LCD_v1.0.tar.gz

1) 10.1 inch MIPI screen delivery list, including a touch screen, a MIPI LCD screen, a 31pin to 40pin cable, a 12pin touch screen cable, a 30pin MIPI cable, and a transfer board

2) Connect the 12pin touch screen cable and 30pin MIPI cable to the adapter board as shown in the figure below. Note that the 12pin touch screen cable direction is the blue bar facing down.

3) With the touch screen facing down, stack the MIPI LCD screen on the touch screen as shown below

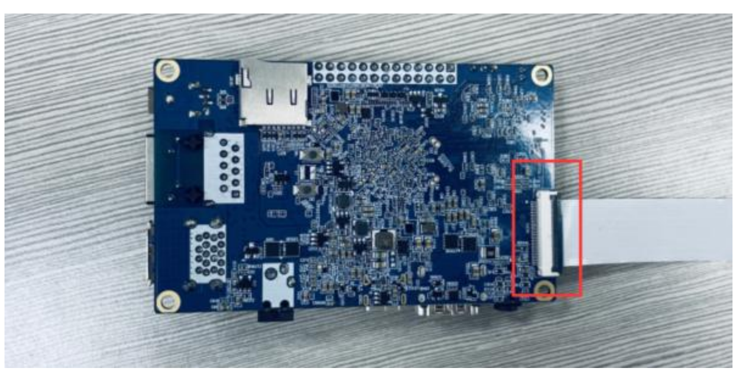
4) Place the connected adapter board on the MIPI LCD screen as shown in the figure below

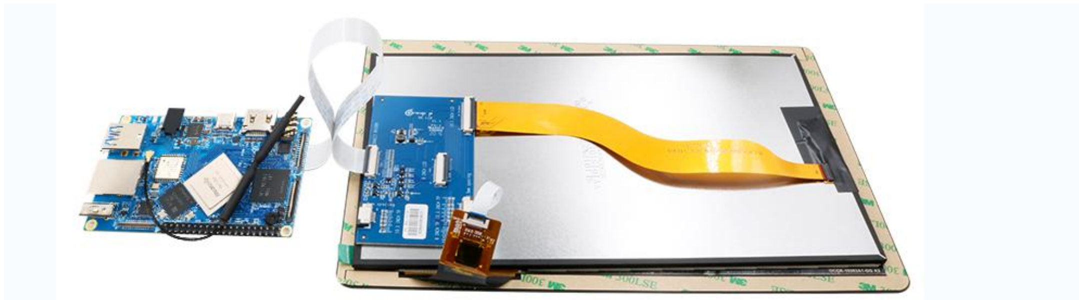
5) Then connect the MIPI LCD screen and the adapter board through the 31pin to 40pin

cable

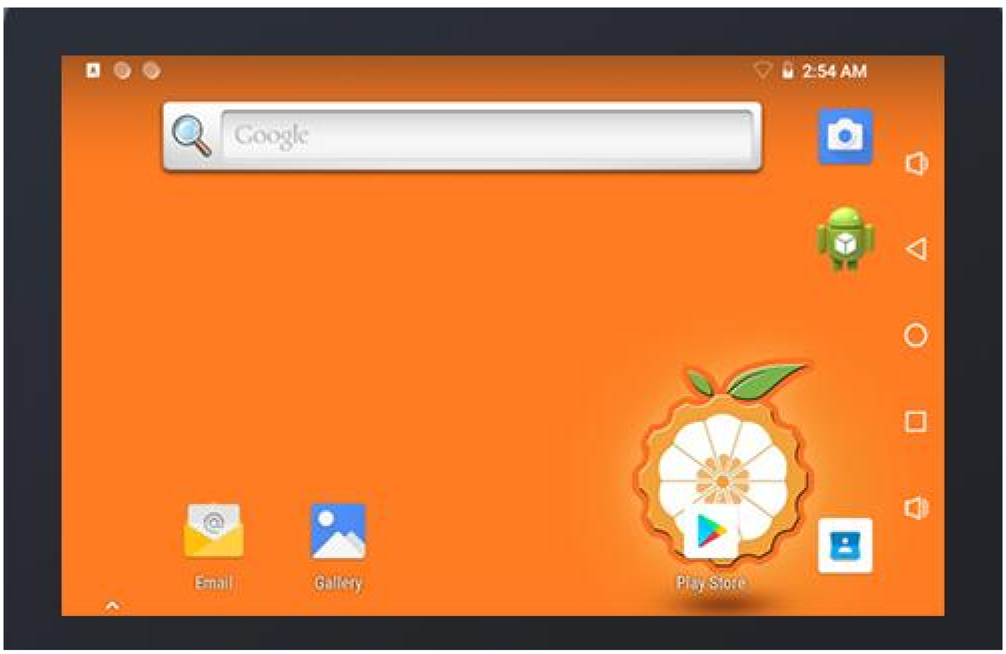
6) Then connect the touch screen and the adapter board through the 12pin touch screen cable

7) Connect the adapter board to the LCD1 interface of the Orange Pi 4 LTS through a 30pin MIPI cable





8) Then insert the DC power supply into the development board. After the system is started, the interface is as shown in the figure below.



4. 9. How to use the USB camera

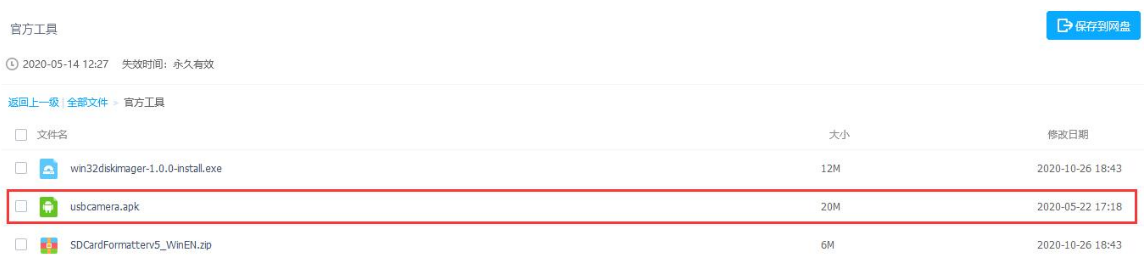
1) First insert the USB camera into the USB interface of the development board. If the USB camera is recognized normally, the corresponding video device node will be generated under /dev

```
rk3399_mid:/ $ ls /dev/video*
/dev/video0
rk3399_mid:/ $ ls /sys/class/video4linux/ -lh
total 0
lrwxrwxrwx 1 root root 0 2020-09-30 03:29 video0 \
```

```
-> ../../devices/platform/usb@fe900000/fe900000.d0
```

2) Then make sure the adb connection between the Ubuntu PC and the development board is normal

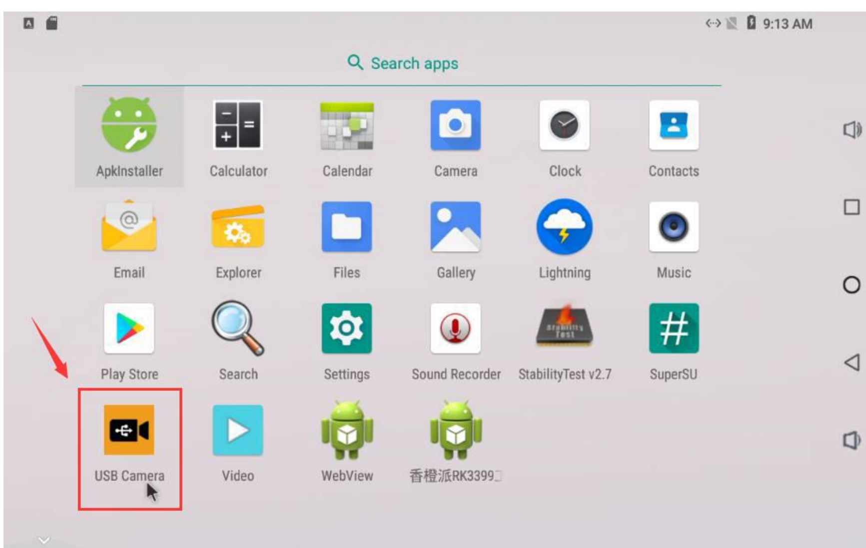
3) Download the USB camera test APP in the **official tool** on the data download page of Orange Pi 4 LTS



4) Then use the adb command to install the USB camera test APP to the Android system, of course, you can also use the U disk copy method to install

```
test@test:~$ adb install usbcamera.apk
```

5) After installation, you can see the startup icon of the USB camera in the Android App list



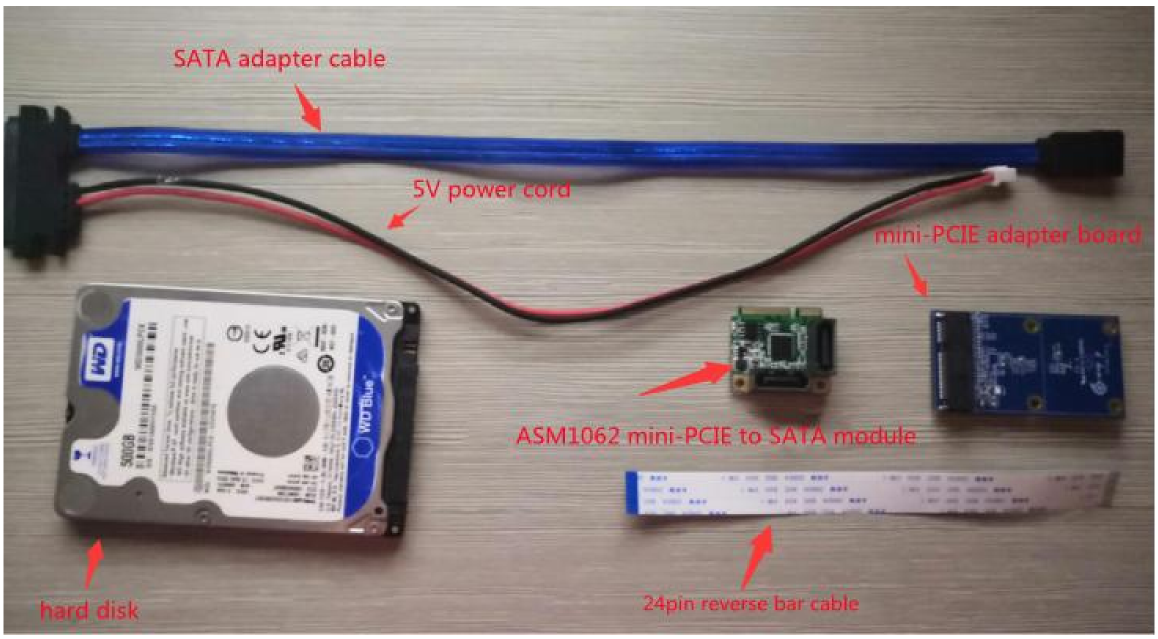
6) Then double-click to open the USB camera APP to see the output video of the USB camera

4. 10. How to use Mini PCIE

Note that the Android system currently only supports the function of connecting mini PCIE to SATA hard disk, and other functions are not adapted

4. 10. 1. Instructions for connecting Mini PCIE to SATA hard disk

1) Prepare the required accessories, 24pin reverse cable, mini PCIE adapter board, ASM1062 mini PCIE to SATA module, hard disk, SATA adapter cable, 5V power cable, the accessories picture is as shown below



2) Connect the 24pin reverse cable to the mini PCIE adapter board as shown in the figure below. Note that the blue bar of the cable is facing outward.

3) Connect the mini PCIE adapter board to the 24pin interface of the Orange Pi 4 LTS development board

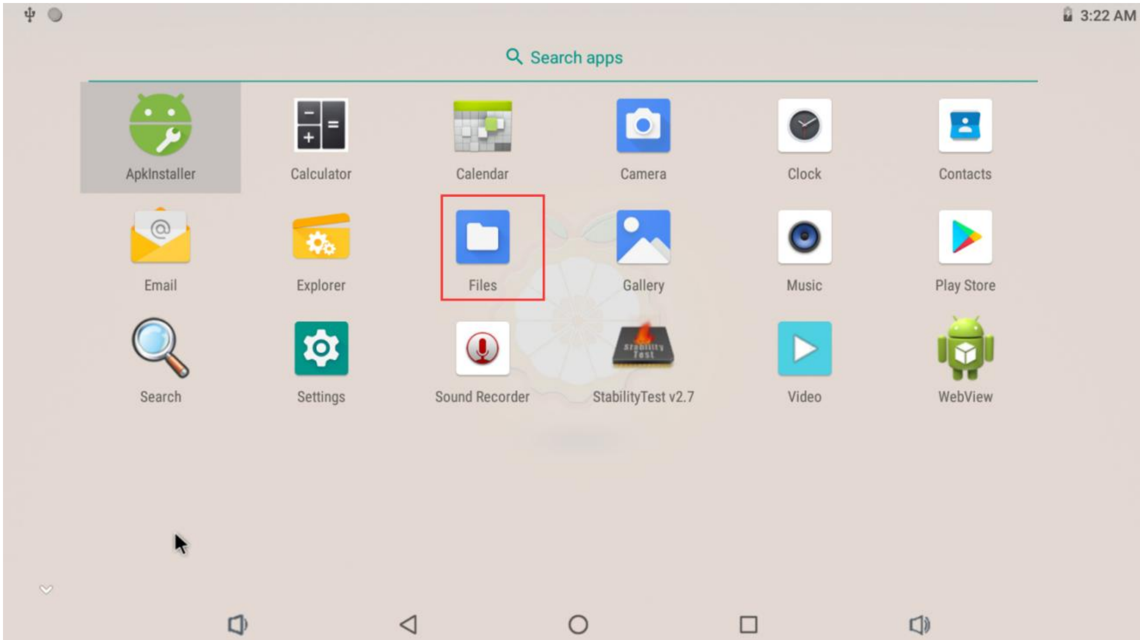
4) Connect the ASM1062 mini PCIE to SATA module to the mini PCIE adapter board

5) Connect the hard disk to the interface of the mini PCIE to SATA module through the SATA cable

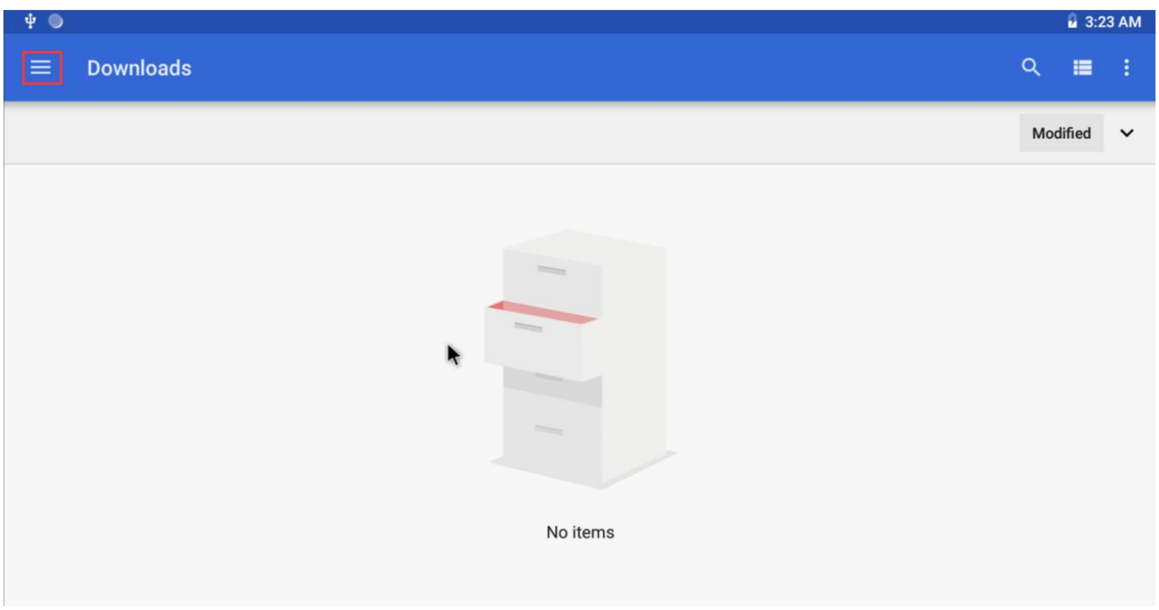
6) Connect the power cable of the SATA adapter cable to the 5V power supply. After the connection is completed, the development board is connected to the DC power supply and powered on. ASM1062 mini PCIE to SATA module LED light flashes, indicating successful connection

This function does not support hot swapping and must be connected before powering on

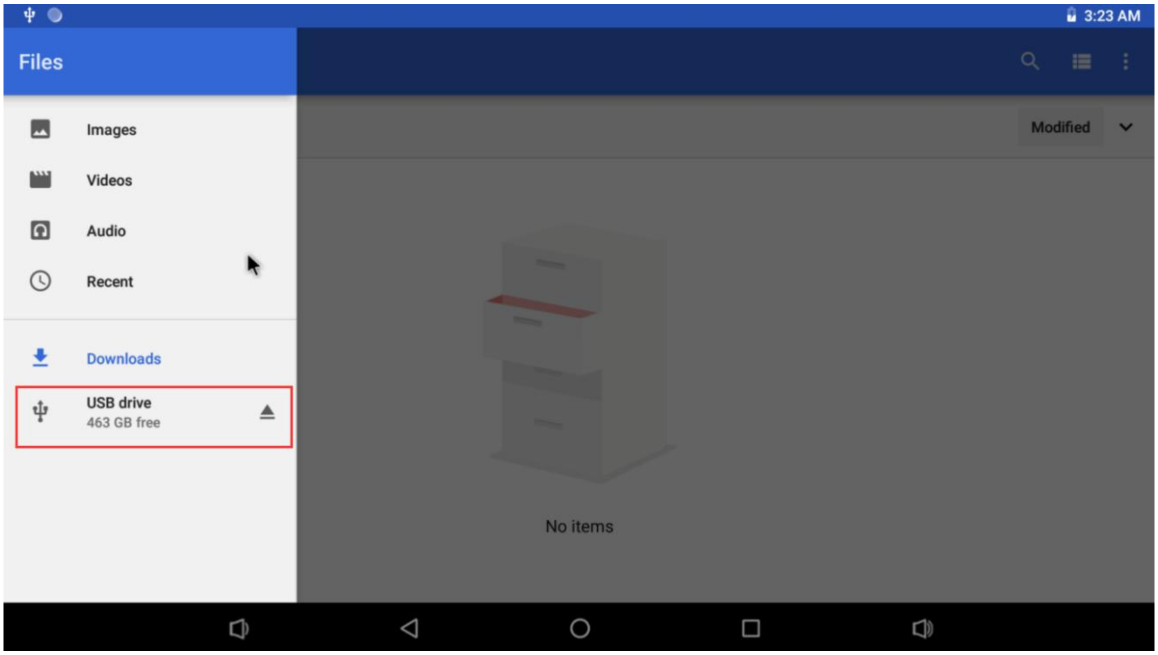
7) Power on and start the development board. After the system is started, open the file manager app



8) After opening, the file manager interface is as shown below, and then click the position of the red box in the upper left corner



9) At this point, you can see that the 500G hard disk is recognized


3:23 AM
Files
Images
Videos
Audio
Recent
Downloads
USB drive
463 GB free
Modified
No items

5. Instructions for using the Linux SDK

5. 1. Compilation system requirements

1) The Linux SDK, namely **orangepi-build**, only supports running on a computer with **Ubuntu 22.04** installed, so before downloading orangepi-build, please make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on the computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please change the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:       Ubuntu 22.04 LTS
Release:            22.04
Codename:           jammy
```

2) If the computer is installed with Windows system and there is no computer with Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in the Windows system. But please note, do not compile orangepi-build on the WSL virtual machine, because orangepi-build has not been tested in the WSL virtual machine, so it cannot be guaranteed that orangepi-build can be used normally in WSL, and please do not use the Linux system of the development board. Use orangepi-build in .

3) The installation image download address of Ubuntu 22.04 **amd64** version is:

```
https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso
```

4) After installing Ubuntu 22.04 in the computer or virtual machine, please set the software source of Ubuntu 22.04 to Tsinghua source, otherwise it is easy to make mistakes due to network reasons when installing the software later.

a. For the method of replacing Tsinghua source, please refer to the description of this webpage.

```
https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/
```

b. Note that the Ubuntu version needs to be switched to 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 cat /etc/apt/sources.list.bak
test@test:~$ sudo vim /etc/apt/sources.list
# The source image is commented by default to improve the speed of apt update, you can uncomment it yourself if necessary
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# Pre-release software sources, not recommended to enable
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

d. After the replacement, you need to update the package information and ensure that no errors are reported

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

e. In addition, since the source code such as the kernel and U-boot are stored on GitHub, it is very important to ensure that the computer can download the code from GitHub normally when compiling the image.

5. 2. Get the source code of linux sdk

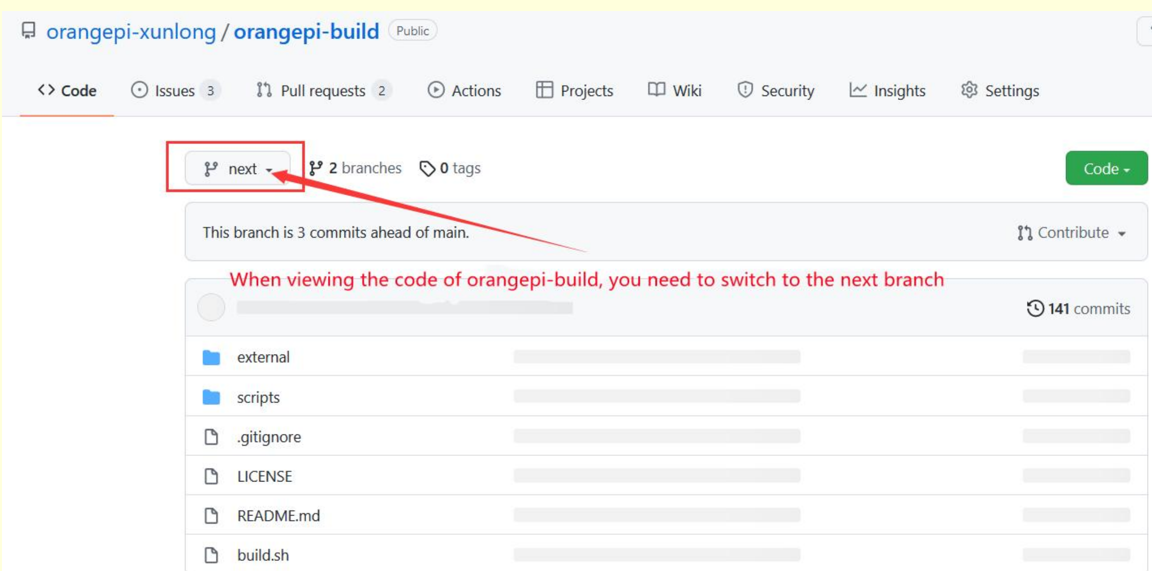
Note that Orange Pi 4 LTS must use the source code of the next branch of orangepi-build, and the source code of the main branch is not adapted to Orange Pi 4 LTS.

5. 2. 1. Download orangepi-build from github

1) The linux sdk actually refers to the code of orangepi-build. Orangepi-build is modified based on the armbian build compilation system. Using orangepi-build, multiple versions of linux images can be compiled. First download the code of orangepi-build. Currently, the RK3399 series development boards in the Linux SDK already support the legacy branch and the current branch.

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install git
test@test:~$ git clone https://github.com/orangepi-xunlong/orangepi-build.git -b next
```

Note that Orange Pi 4 LTS must use the source code of the next branch of orangepi-build. The above git clone command specifies that the branch of the orangepi-build source code is next. The source code of the main branch is not adapted to Orange Pi 4 LTS.



Downloading the code of orangepi-build through the git clone command does not require entering the username and password of the github account (the same is true for downloading other codes in this manual). If the Ubuntu PC prompts the user who needs to enter the github account after entering the git clone command The name and password are usually the wrong address of the orangepi-build repository behind git clone. Please check the spelling of the command carefully, instead of thinking that we forgot to provide the username and password of the github account here.

2) The u-boot and linux kernel versions currently used by the RK3399 series development boards are as follows

branch	u-boot version	linux kernel version
legacy	u-boot 2020.10	linux4.4
current	u-boot 2020.10	linux5.10
next	u-boot 2020.10	linux5.18

3) Orangepi-build will contain the following files and folders after downloading

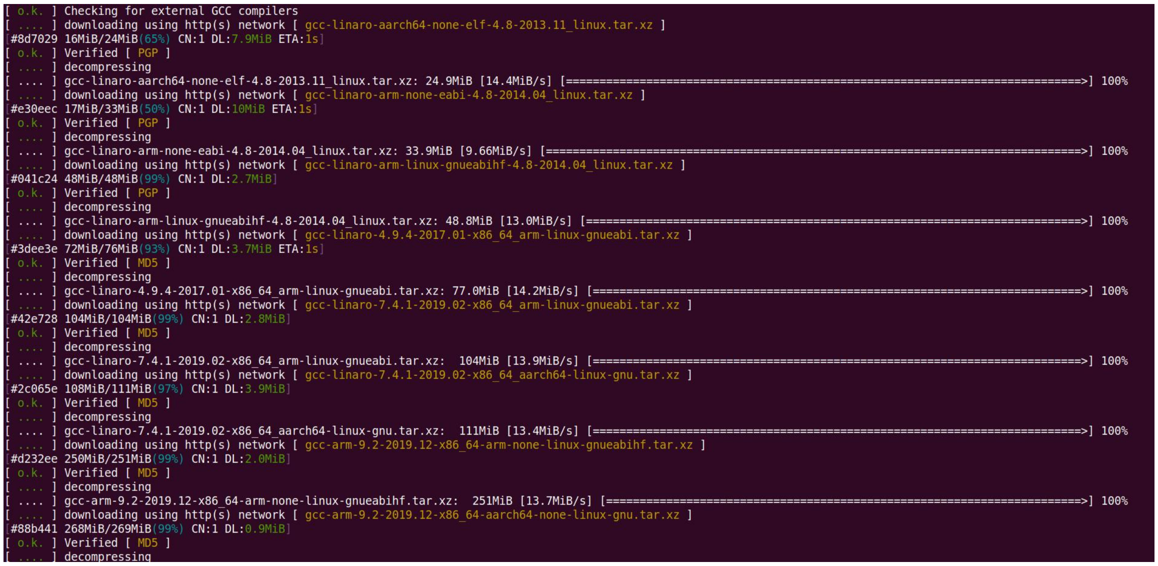
a. **build.sh**: Compile startup script
b. **external**: Contains configuration files, specific scripts and source code of some programs needed to compile the image, etc.
c. **LICENSE**: GPL 2 license file
d. **README.md**: orangepi-build documentation
e. **scripts**: Generic script for compiling linux images

```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

If you download the code of orangepi-build from github, you may find that orangepi-build does not contain the source code of u-boot and linux kernel after downloading, nor does it require cross-compilation tools to compile u-boot and linux kernel chain, this is normal, because these things are stored in other separate github repositories or some servers (the addresses will be detailed below). orangepi-build will specify the address of u-boot, linux kernel and cross-compilation toolchain in the script and configuration file. When running orangepi-build, when it finds that these things are not available locally, it will automatically go to the corresponding place to download.

5. 2. 2. Download the cross-compilation toolchain

1) When orangepi-build runs for the first time, it will automatically download the cross-compilation toolchain and put it in the **toolchains** folder. After running the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists. , if it does not exist, it will restart the download, if it exists, it will be used directly, and the download will not be repeated



2) The mirror website of the cross-compilation tool chain in China is the open source software mirror site of Tsinghua University

https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/

3) After the **toolchains** is downloaded, it will contain multiple versions of the cross-compilation toolchain

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

4) The cross-compilation toolchain used to compile the RK3399 Linux kernel source code is

a. linux4.4

```
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
```

b. linux5.10

```
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

c. linux5.18

```
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

5) The cross-compilation toolchain used to compile RK3399 u-boot source code is

a. v2020.10

```
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

5. 2. 3. Orangepi-build complete directory structure description

1) After the orangepi-build repository is downloaded, it does not contain the source code of the linux kernel, u-boot and the cross-compilation tool chain. The source code of the linux kernel and u-boot are stored in a separate git repository

a. The git repository where the linux kernel source code is stored is as follows, pay attention to switch the branch of the linux-orangepi repository to

a) linux4.4

https://github.com/orangepi-xunlong/linux-orangepi/tree/**orange-pi-4.4-rockchip64**

b) Linux5.10

https://github.com/orangepi-xunlong/linux-orangepi/tree/**orange-pi-5.10-rk3399**

c) Linux5.18

https://github.com/orangepi-xunlong/linux-orangepi/tree/**orange-pi-5.18**

b. The git repository where the u-boot source code is stored is as follows. Note that the branch of the u-boot-orangepi repository is switched to

v2020.10-rockchip64

https://github.com/orangepi-xunlong/u-boot-orangepi/tree/**v2020.10-rockchip64**

If you are not familiar with orangepi-build and do not know the detailed process of compiling the linux kernel and u-boot, please do not download and use the above linux kernel and u-boot source code for compilation operation, because the compilation script and configuration file of orangepi-build Some adjustments and optimizations will be made to u-boot and linux. If you do not use orangepi-build to compile u-boot and linux, you may encounter problems of compilation failure or

failure to start.

2) When orangepi-build runs for the first time, it will download the cross-compilation toolchain, u-boot and linux kernel source code. After successfully compiling a linux image, the files and folders that can be seen in orangepi-build are:

c. **build.sh**: Compile startup script
d. **external**: Contains configuration files, scripts for specific functions, and source code of some programs needed to compile the image. The rootfs compressed package cached during the process of compiling the image is also stored in external
e. **kernel**: The source code of the Linux kernel is stored. The folder named **orange-pi-4.4-rockchip64** stores the kernel source code of the legacy branch of the RK3399 series development board. The folder named **orange-pi-5.10-rk3399** stores the RK3399 series. The kernel source code of the current branch of the development board, the folder named **orange-pi-5.18** stores the kernel source code of the next branch of the RK3399 series development board (if only the linux image of the legacy branch is compiled, then only the legacy branch can be seen If only the linux image of the current branch is compiled, then only the kernel source code of the current branch can be seen; if only the linux image of the next branch is compiled, then only the kernel source code of the next branch can be seen). Please do not modify the name of the folder manually. If modified, the kernel source code will be downloaded again when the compilation system is running.
f. **LICENSE**: GPL 2 license file
g. **README.md**: orangepi-build documentation
h. **output**: Store the compiled u-boot, linux and other deb packages, compilation logs, and compiled images and other files
i. **scripts**: Generic script for compiling linux images
j. **toolchains**: Store the cross-compilation toolchain
k. **u-boot**: The source code of u-boot is stored. The folder named **v2020.10-rockchip64** stores the u-boot source code used by the current branch, legacy branch and next branch of the RK3399 series development board. Please name the folder of the u-boot source code. Do not modify it manually. If modified, the u-boot source code will be downloaded again when the compilation system is running.

l. **userpatches**: Store the configuration files needed to compile the script

test@test:~/orangepi-build$ **ls**
build.sh external kernel LICENSE output README.md scripts toolchains u-boot userpatches

5. 2. 4. Download orangepi-build from Google Cloud Disk

1) If the speed of downloading orangepi-build from github is very slow, you can also download the orangepi-build compressed package from Google cloud disk. The download link of Google cloud disk is

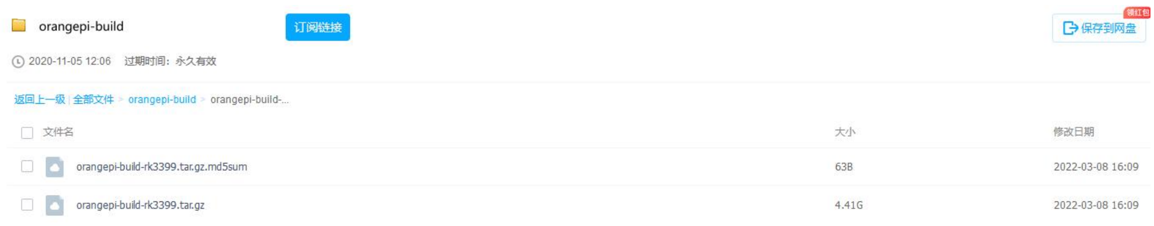
Link:
https://drive.google.com/drive/folders/1Ai3GVaSk7Dd9IVb9PJYuQc-BRRGFzKrH?usp=sharing

Note that after opening the link of Google cloud disk, you can find the folder named orangepi-build-rk3399 and download it. You don't need to worry about the role of other files, it doesn't matter here. Please don't worry about why the above screenshot is different from the length displayed in the actual Google cloud disk, because there is no need to cut a bunch of irrelevant things to take up space.

2) There are two files in the orangepi-build-rk3399 folder of Google cloud disk

a. **orangepi-build-rk3399.tar.gz** is the compressed package of orangepi-build source code
b. **orangepi-build-rk3399.tar.gz.md5sum** is the MD5 checksum file of the compressed package of orangepi-build source code
c. After downloading, please first check whether the MD5 checksum of the orangepi-build-rk3399.tar.gz compressed package is correct, which can prevent problems with the downloaded compressed package. If it is not correct, please download it again and check the verification and whether the correct command is

test@test:~$ **md5sum -c orangepi-build-rk3399.tar.gz.md5sum**
orangepi-build-rk3399.tar.gz: **success**



3) Then you can use the **tar -zxf** command to decompress orangepi-build-rk3399.tar.gz

```
test@test:~$ tar -zxf orangepi-build-rk3399.tar.gz
test@test:~$ cd orangepi-build/
test@test:~/orangepi-build$ ls
build.sh  external  kernel  LICENSE  README.md  scripts  toolchains
u-boot  userpatches
```

4) Before using orangepi-build to compile the system, please synchronize orangepi-build with the github server to ensure that the code is in the latest state

```
test@test:~/orangepi-build$ git pull
```

5) The **orangepi-build-rk3399.tar.gz** compressed package on the Google cloud disk not only contains the code of the orangepi-build compilation system, but also caches the source code of the cross-compilation toolchain, u-boot and linux kernel. Therefore, in the process of compiling the image, we will not go to the github server to download the source code of u-boot and linux kernel and the cross-compilation toolchain from scratch, which can save a lot of time. When the orangepi-build compilation system starts to run, the source code of u-boot and linux kernel will be automatically synchronized from github by default to ensure that the code is in the latest state, so there is no need to manually synchronize the source code of u-boot and linux kernel

6) The decompressed kernel folder of **orangepi-build-rk3399.tar.gz** on Google cloud disk will contain multiple versions of kernel source code. Please do not modify the names of these kernel source code folders, otherwise the compilation system will not be able to find them. Go to the kernel source code and then download the kernel source code from github again. The kernel source code used by the RK3399 series development board is as follows:

orange-pi-4.4-rockchip64	**RK3399 series development board legacy branch use**
orange-pi-5.10-rk3399	**RK3399 series development board current branch**

	use
orange-pi-5.18	**RK3399 series development board next branch use**

7) The decompressed u-boot folder of **orangepi-build-rk3399.tar.gz** on Google cloud disk will contain the u-boot source code, please do not modify the name of the u-boot source code folder, otherwise it will cause the compilation system to find If the u-boot source code is not available, download the u-boot source code from github again. The u-boot source code used by the RK3399 series development board is:

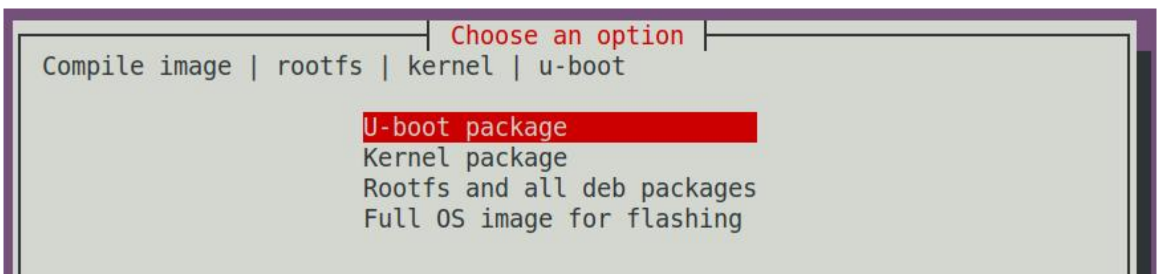
v2020.10-rockchip64	**RK3399 series development board current and legacy and next branch use**

5. 3. Compile u-boot

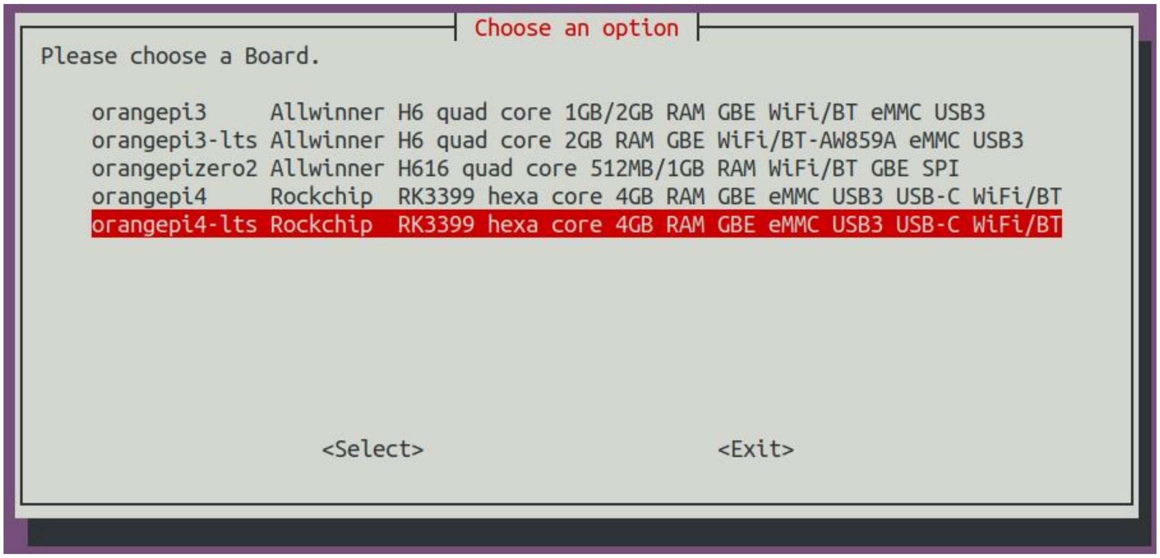
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

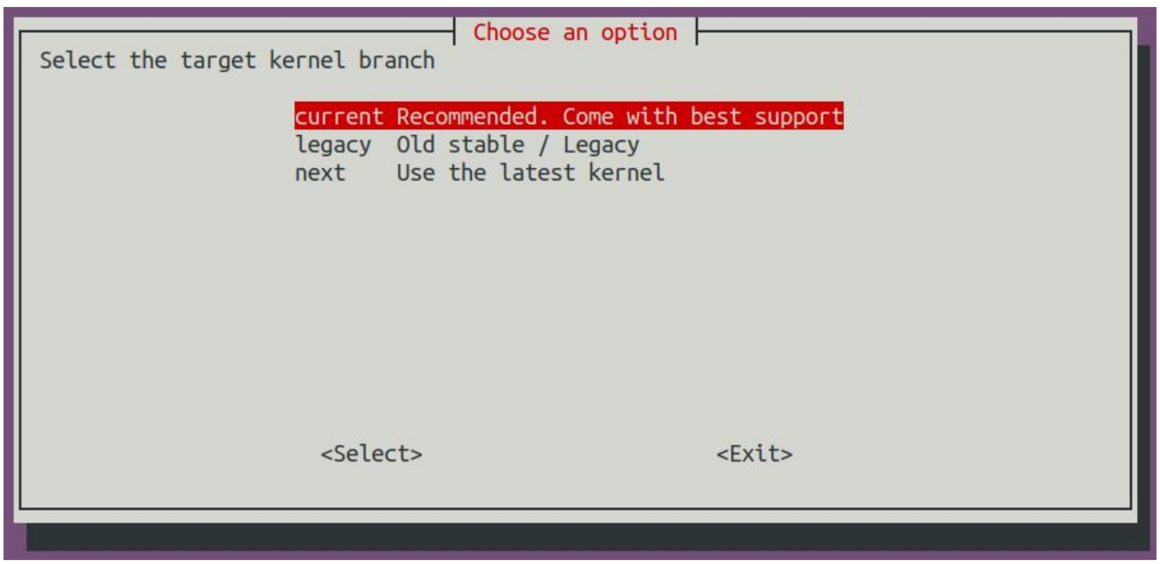
2) Select **U-boot package** and press Enter



3) Then select the model of the development board



4) Then select branch



5) Then it will start to compile u-boot, and some of the information prompted during compilation are as follows

a. u-boot source code version

[o.k.] Compiling u-boot [**v2020.10**]

b. The version of the cross-compile toolchain

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 9.2.1**]

c. The path to the generated u-boot deb package

[o.k.] Target directory [**orangepi-build/output/debs/u-boot**]

d. The package name of the u-boot deb package generated by compilation

```
[ o.k. ] File name [ linux-u-boot-current-orangepi4-lts_3.0.0_arm64.deb ]
```

e. compile time

```
[ o.k. ] Runtime [ 1 min ]
```

f. Repeat the command to compile u-boot, use the following command to start compiling u-boot directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepi4-lts
BRANCH=current BUILD_OPT=u-boot KERNEL_CONFIGURE=yes ]
```

6) View the compiled u-boot deb package

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-current-orangepi4-lts_3.0.0_arm64.deb
```

7) The files contained in the generated u-boot deb package are as follows

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ $ dpkg -x \
linux-u-boot-current-orangepi4-lts_3.0.0_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-current-orangepi4-lts_3.0.0_arm64.deb  usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-current-orangepi4-lts_3.0.0_arm64
    │   ├── idbloader.bin
    │   ├── trust.bin
    │   └── uboot.img
    └── u-boot
        ├── LICENSE
        ├── orangepi-4-rk3399_defconfig
        └── platform_install.sh

3 directories, 6 files
```

8) The orangepi-bulid compilation system will first synchronize the u-boot source code with the u-boot source code of the github server when compiling the u-boot source code, so if you want to modify the u-boot source code, you first need to turn off the download and update function of the source code (**You need to compile u-boot once before closing this function, otherwise you will be prompted that the source code of u-boot cannot be found**), otherwise the modifications 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"
```

9) When debugging the u-boot code, you can use the following method to update the u-boot in the linux image for testing

a. Upload the compiled u-boot deb package to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-current-orangepi4-lts_3.0.0_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the installed deb package of u-boot

```
root@orangepi:~# apt purge -y linux-u-boot-orangepi4-lts-current
```

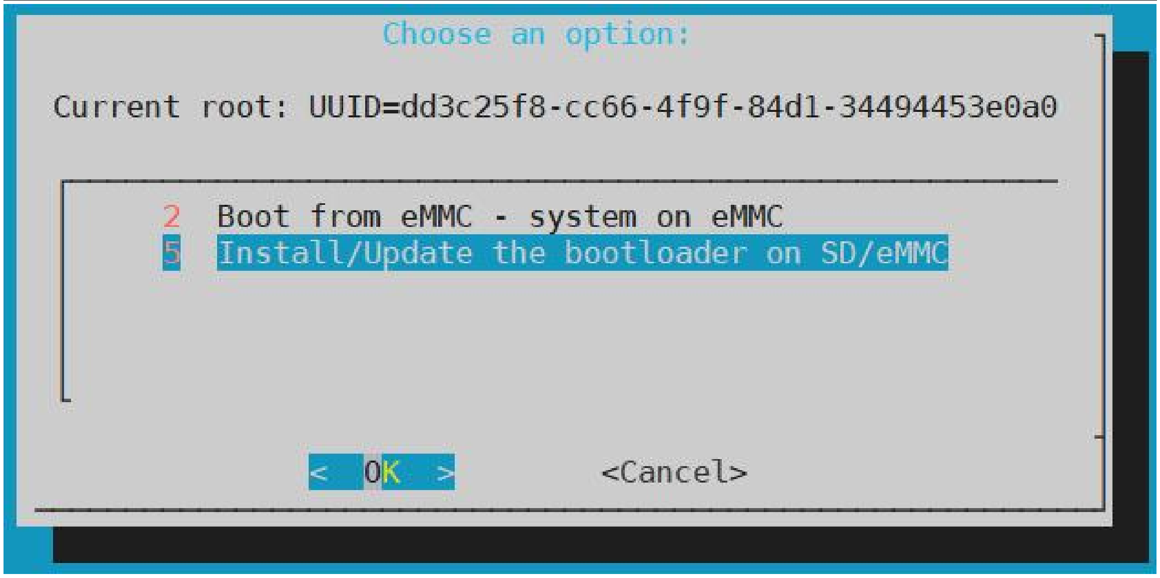
c. Install the new u-boot deb package just uploaded

```
root@orangepi:~# dpkg -i linux-u-boot-current-orangepi4-lts_3.0.0_arm64.deb
```

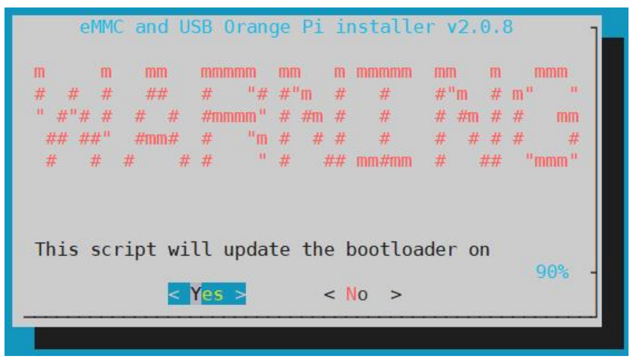
d. Then run the nand-sata-install script

```
root@orangepi:~# nand-sata-install
```

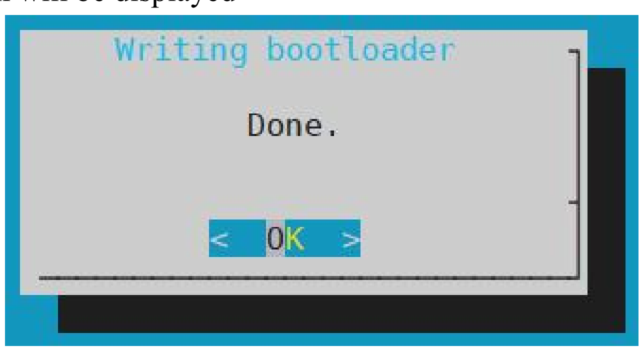
e. Then select **5 Install/Update the bootloader on SD/eMMC**



f. After pressing the Enter key, a Warring will pop up first



g. Press the Enter key again to start updating u-boot. After the update, the following information will be displayed



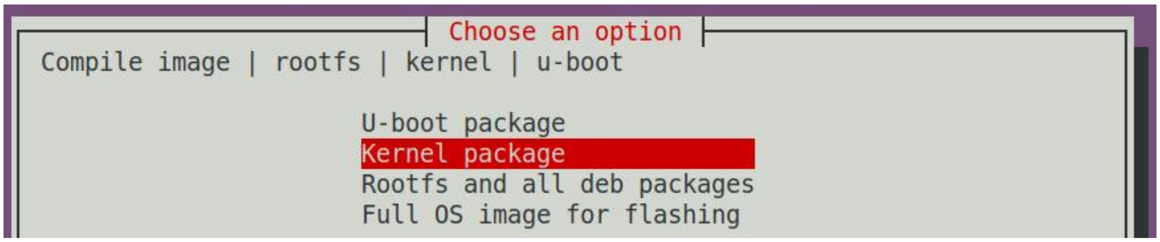
h. Then you can restart the development board to test whether the modification of u-boot takes effect

5. 4. Compile the linux kernel

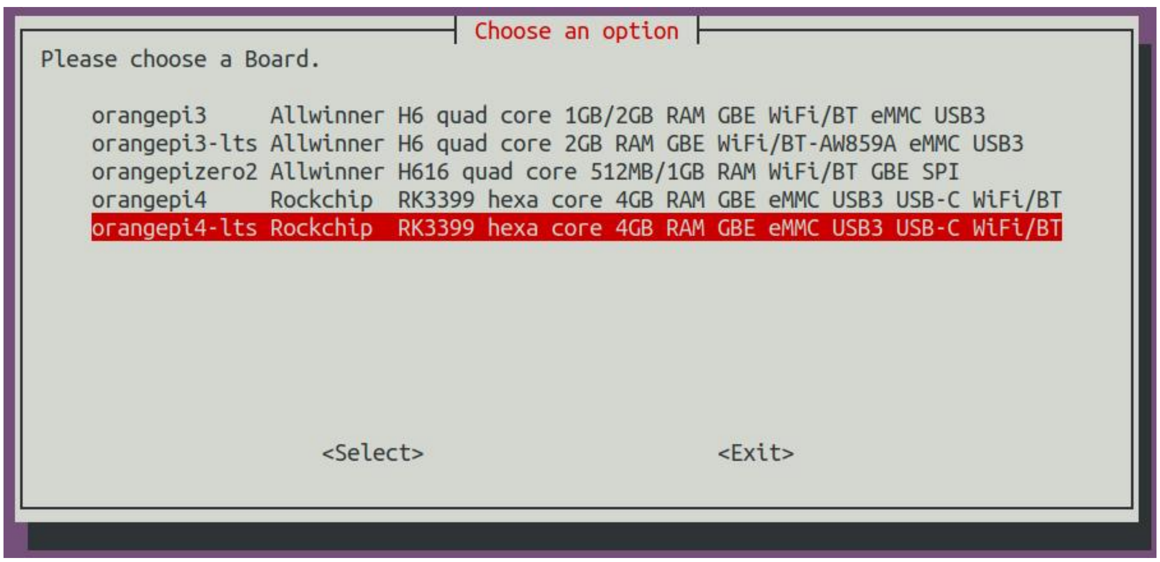
1) Run the build.sh script, remember to add sudo permissions

test@test:~/orangepi-build$ **sudo ./build.sh**

2) Select **Kernel package**, then press Enter



3) Then select the model of the development board

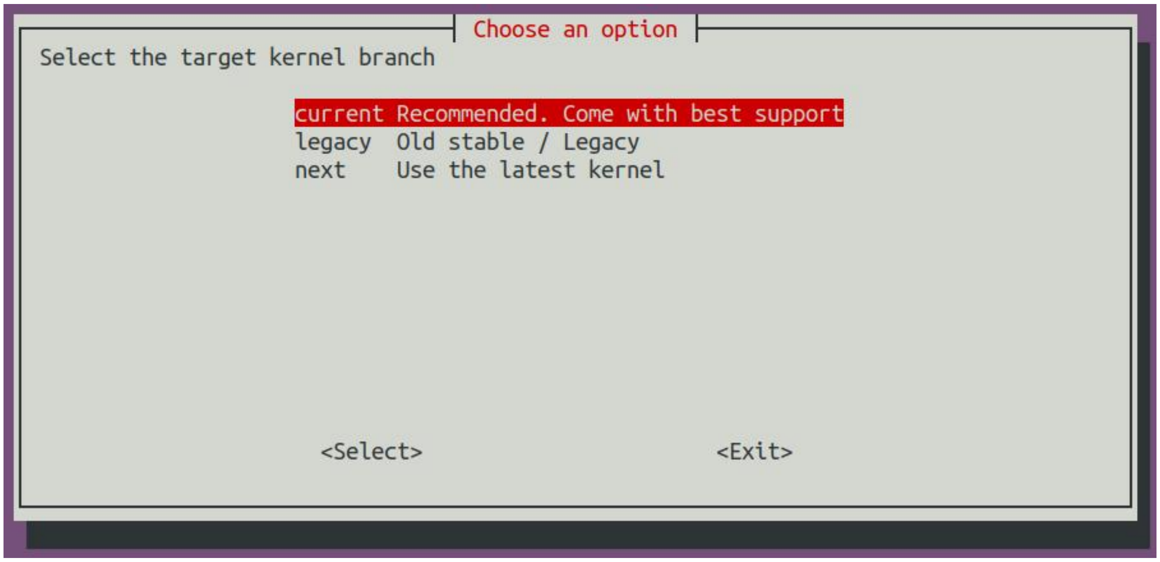


4) Then select branch

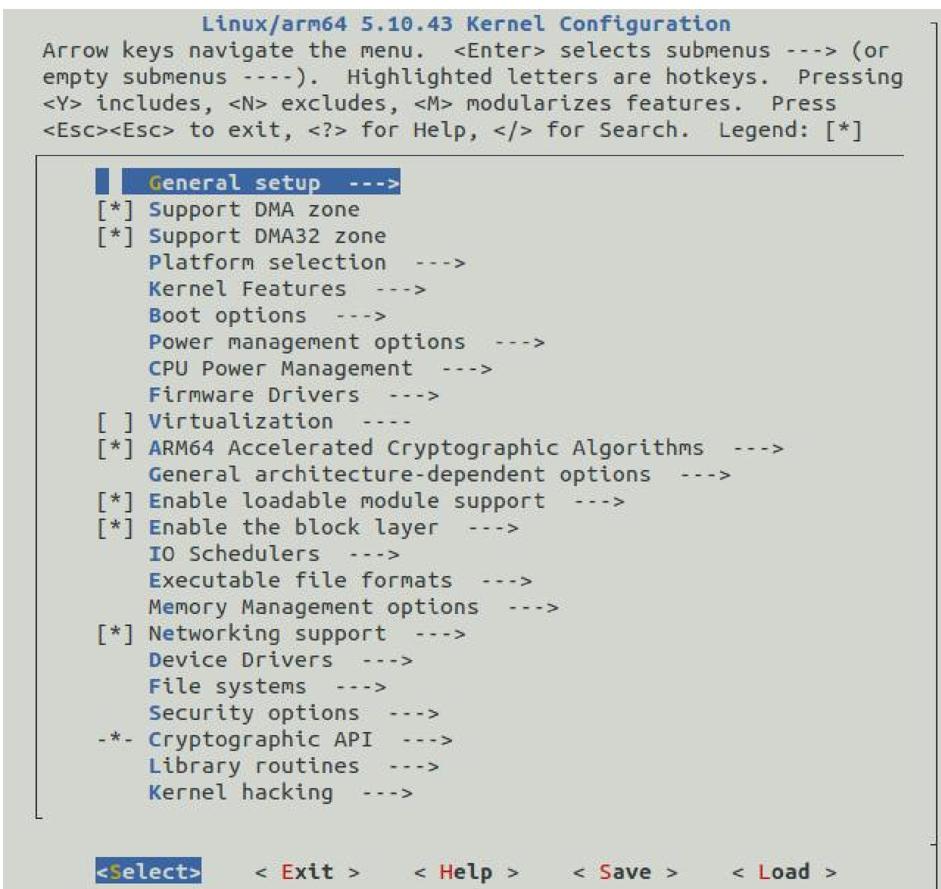
a. current will compile linux 5.10

b. legacy will compile linux4.4

c. next will compile linux5.18



5) Then the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration. If you do not need to modify the kernel configuration, you can simply exit. After exiting, the kernel source code will be compiled.



a. If you do not need to modify the configuration options of the kernel, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily shield the configuration interface of the pop-up kernel

test@test:~/orangepi-build$ **sudo ./build.sh KERNEL_CONFIGURE=no**

b. You can also set **KERNEL_CONFIGURE=no** in the orangepi-build/userpatches/config-default.conf configuration file to permanently disable this feature

c. If the following error is displayed when compiling the kernel, this is because the terminal interface of Ubuntu PC is too small, so the interface of make menuconfig cannot be displayed. Please adjust the terminal of Ubuntu PC to the maximum, and then re-run the build.sh script

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

6) Part of the information prompted when compiling the kernel source code is explained as follows

a. The version of the linux kernel source code

[o.k.] Compiling current kernel [**5.10.43**]

b. The version of the cross-compilation toolchain used

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 9.2.1**]

c. The configuration file used by the kernel by default and the path where it is stored

[o.k.] Using kernel config file [**config/kernel/linux-5.10-rk3399.config**]

d. If **KERNEL_CONFIGURE=yes**, the final configuration file .config used by the kernel will be copied to **output/config**. If the kernel configuration is not modified, the final configuration file is the same as the default configuration file

[o.k.] Exporting new kernel config [**output/config/linux-5.10-rk3399.config**]

e. The path to the generated kernel-related deb package

```
[ o.k. ] Target directory [ output/debs/ ]
```

f. The package name of the kernel image deb package generated by compilation

```
[ o.k. ] File name [ linux-image-current-rk3399_3.0.0_arm64.deb ]
```

g. Compilation time used

```
[ o.k. ] Runtime [ 5 min ]
```

h. Finally, the compilation command to repeat the compilation of the last selected kernel will be displayed. Use the following command to directly start compiling the kernel source code without selecting through the graphical interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi4-lts
BRANCH=current BUILD_OPT=kernel KERNEL_CONFIGURE=yes ]
```

7) View the compiled and generated kernel-related deb packages

a. **linux-dtb-current-rk3399_3.0.0_arm64.deb** Not used yet, don't worry about it

b. **linux-headers-current-rk3399_3.0.0_arm64.deb** Include kernel header files

c. **linux-image-current-rk3399_3.0.0_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-current-rk3399_3.0.0_arm64.deb
output/debs/linux-image-current-rk3399_3.0.0_arm64.deb
output/debs/linux-headers-current-rk3399_3.0.0_arm64.deb
```

8) The files contained in the generated linux-image deb package are as follows

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi_build/output/debs$ mkdir test
test@test:~/orangepi_build/output/debs$ cp \
linux-image-current-rk3399_3.0.0_arm64.deb test/
test@test:~/orangepi_build/output/debs$ cd test
test@test:~/orangepi_build/output/debs/test$ dpkg -x \
linux-image-current-rk3399_3.0.0_arm64.deb .
test@test:~/orangepi_build/output/debs/test$ ls
boot   etc   lib   linux-image-current-rk3399_3.0.0_arm64.deb   usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi-build/output/debs/test$ tree -L 2
```

```
.
├── boot
│   ├── config-5.10.43                    //The configuration file used to compile the kernel source code
│   ├── System.map-5.10.43
│   └── vmlinuz-5.10.43                   //Compile the generated kernel image file
├── etc
│   └── kernel
├── lib
│   └── modules                           //Compile the generated kernel module
├── linux-image-current-rk3399_3.0.0_arm64.deb
└── usr
    ├── lib
    └── share

8 directories, 4 files
```

9) The orangepi-bulid compilation system will first synchronize the Linux kernel source code with the Linux kernel source code of the github server when compiling the linux kernel source code, so if you want to modify the linux kernel source code, you first need to turn off the update function of the source code (**you need to compile it once This function can only be turned off after the linux kernel source code, otherwise it will prompt that the source code of the linux kernel cannot be found**), otherwise the modifications made will be restored, as follows:

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If the kernel has been modified, the following methods can be used to update the kernel and kernel modules of the Linux system on the development board

a. Upload the compiled deb package of the linux kernel to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-current-rk3399_3.0.0_arm64.deb root@192.168.1.207:/root
```

b. Then log in to the development board and uninstall the deb package of the installed linux kernel

```
root@orangepi:~# apt purge -y linux-image-current-rk3399
```

c. Install the deb package of the new linux kernel just uploaded

```
root@orangepi:~# dpkg -i linux-image-current-rk3399_3.0.0_arm64.deb
```

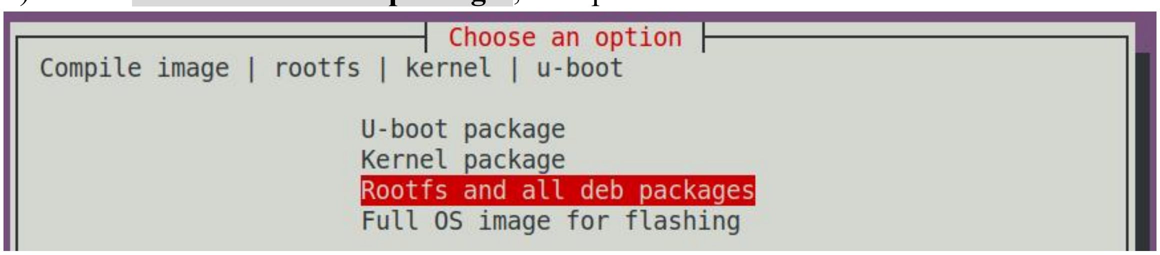
d. Then restart the development board, and then check whether the kernel-related modifications have taken effect

5. 5. Compile rootfs

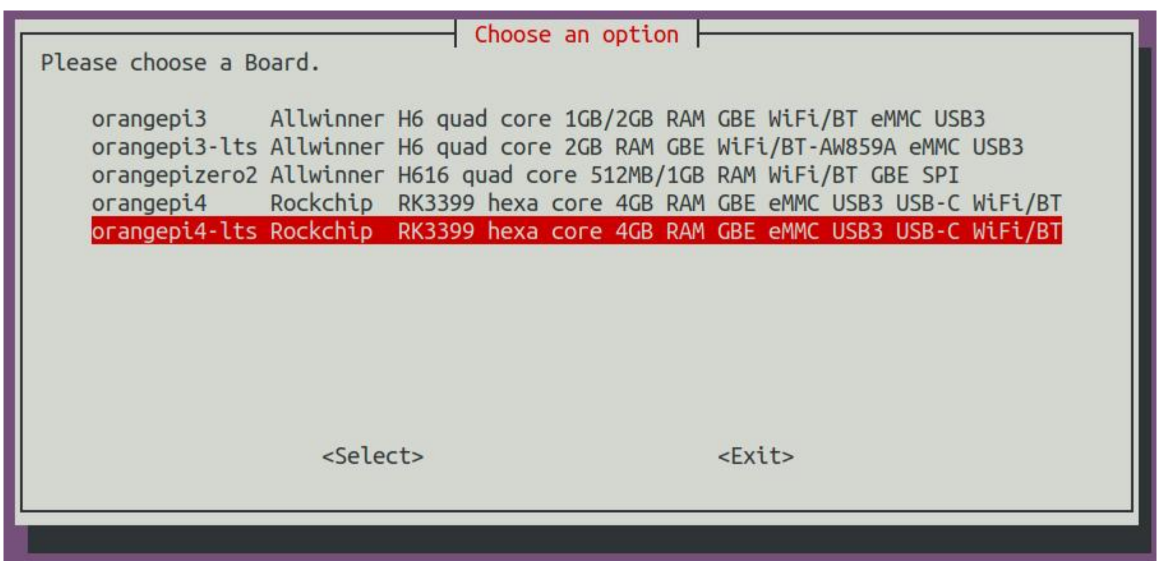
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

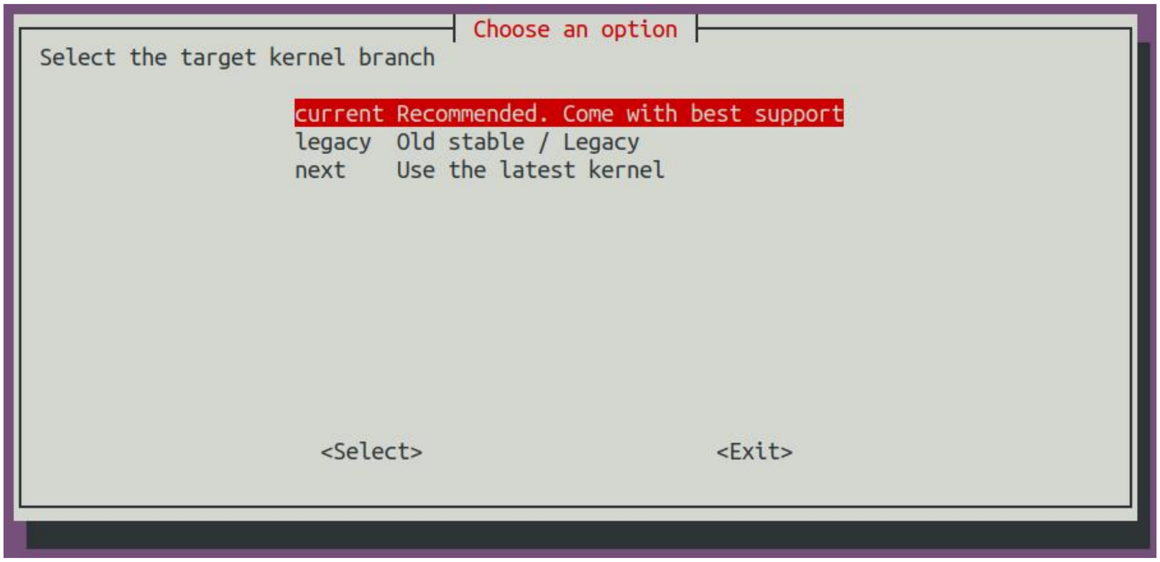
2) Select **Rootfs and all deb packages**, then press Enter



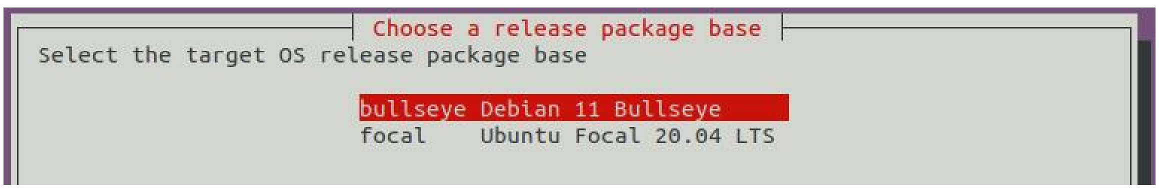
3) Then select the model of the development board



4) Then select branch



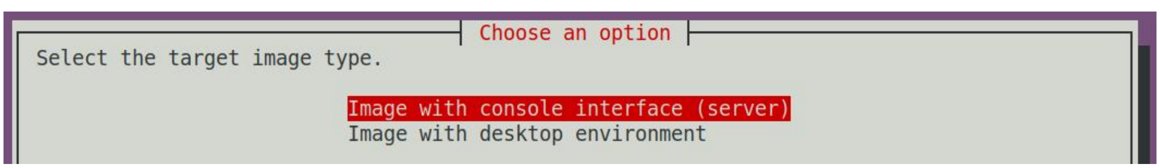
5) Then select the type of rootfs (the rootfs type supported by different branches will be different)



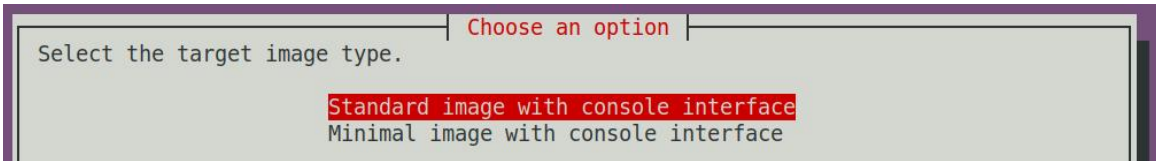
6) Then select the type of mirror

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small in size

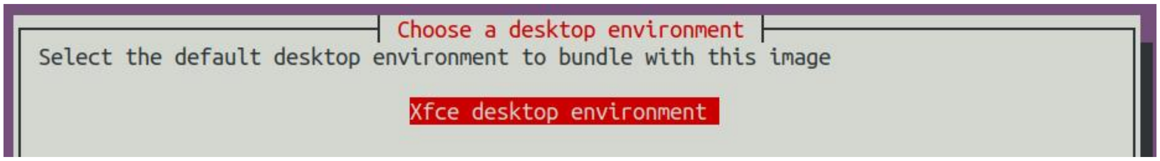
b. **Image with desktop environment** Indicates a mirror image with a desktop, which is relatively large in size

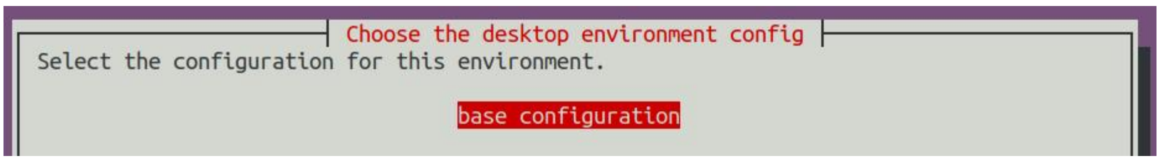


7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The software pre-installed in the Minimal version will be much less than the Standard version.



8) If you are compiling the desktop version of the image, you also need to select the type of desktop environment, but only XFCE is currently supported, so just press Enter.





9) You can then select additional packages that need to be installed. For example, if you need to install a browser, you can choose **browsers**. Which packages are included in each selection can be seen in the code of orangepi-build, you can also modify these configurations to add the packages you want to install

1. First enter **external/config/desktop** to see the desktop configuration folders of different linux distributions. Note that not all the Orange Pi development boards that can be seen in the code are supported and tested.

test@test:~/orangepi-build$ **cd external/config/desktop**
test@test:~/orangepi-build/external/config/desktop$ **ls**
bionic bookworm **bullseye** **buster** **focal** jammy README.md sid

2. Then select the type of distribution you want to view or modify, and enter the corresponding directory, such as **bullseys**

test@test:~/orangepi-build/external/config/desktop$ **cd bullseye**
test@test:~/orangepi-build/external/config/desktop/bullseye$ **ls**
appgroups environments

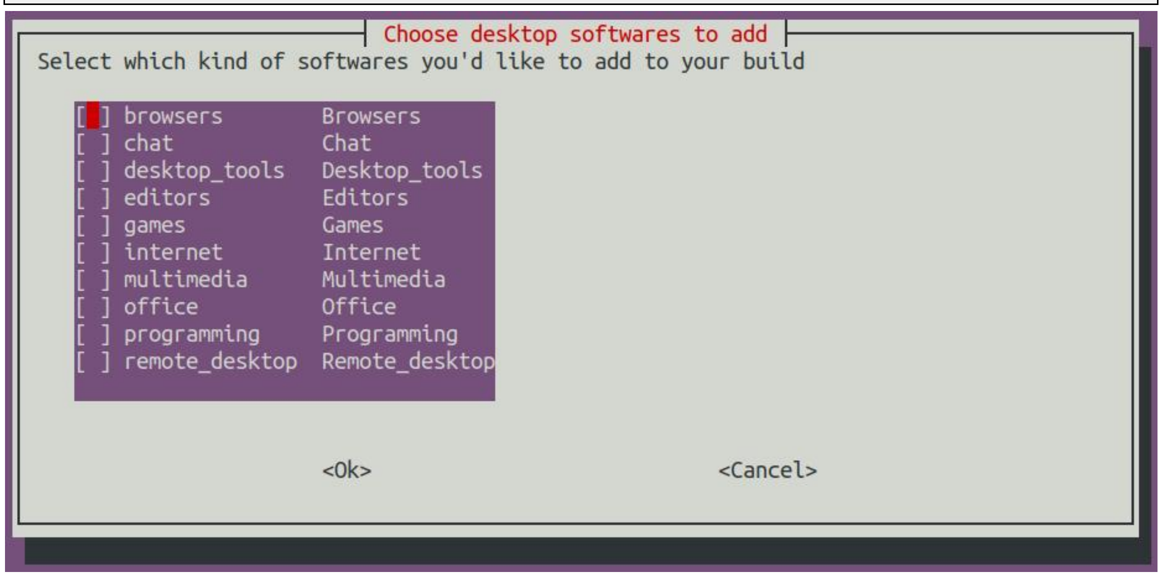
3. Then enter the **appgroups** directory to see all app groups

test@test:~/orangepi-build/external/config/desktop/bullseye$ **cd appgroups**
test@test:~/orangepi-build/external/config/desktop/bullseye/appgroups$ **ls**

browsers chat desktop_tools editors games internet multimedia office programming remote_desktop

4. Open the **packages** file under different groups to view the software contained in the group

test@test:~/orangepi-build/external/config/desktop/bullseye/appgroups$ **cat programming/packages**
geany
thonny
test@test:~/orangepi-build/external/config/desktop/bullseye/appgroups$ **cat office/packages**
ibreoffice



10) Then it will start to compile rootfs, and some of the information prompted during compilation are as follows

a. type of rootfs

[o.k.] local not found [Creating new rootfs cache for **bullseye**]

b. The storage path of the rootfs compressed package generated by compilation

[o.k.] Target directory [**external/cache/rootfs**]

c. The name of the rootfs compressed package generated by compilation

[o.k.] File name [**bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4**]

d. Compilation time

```
[ o.k. ] Runtime [ 13 min ]
```

e. Repeat the command to compile rootfs, use the following command to start compiling rootfs directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh    BOARD=orangepi4-lts
BRANCH=current BUILD_OPT=rootfs RELEASE=bullseye
BUILD_MINIMAL=no    BUILD_DESKTOP=no
KERNEL_CONFIGURE=yes    ]
```

11) View the rootfs compressed package generated by compilation

a. bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4 is the compressed package of rootfs, the meaning of each field of the name is
   a) **bullseye** indicates the type of linux distribution of rootfs
   b) **xfce** indicates that the rootfs is of the desktop version, and if it is **cli**, it indicates the server version
   c) **arm64** represents the architecture type of rootfs
   d) **25250ec7002de9e81a41de169f1f89721** is the MD5 hash value generated by the package names of all packages installed by rootfs. As long as the list of packages installed by rootfs is not modified, this value will not change, and the compilation script will use this MD5 hash value to Determine if you need to recompile rootfs

b. bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list lists the package names of all 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
```

12) If the required rootfs already exists under **external/cache/rootfs**, then compiling the rootfs again will skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has There is a rootfs available for cache, if there is one, use it directly, which can save a lot of download and compilation time

5. 6. Compile the linux image

1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Full OS image for flashing** and 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.

    orangepi3      Allwinner H6 quad core 1GB/2GB RAM GBE WiFi/BT eMMC USB3
    orangepi3-lts  Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
    orangepizero2  Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
    orangepi4      Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
    orangepi4-lts  Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT

              <Select>                    <Exit>
```

4) Then select branch

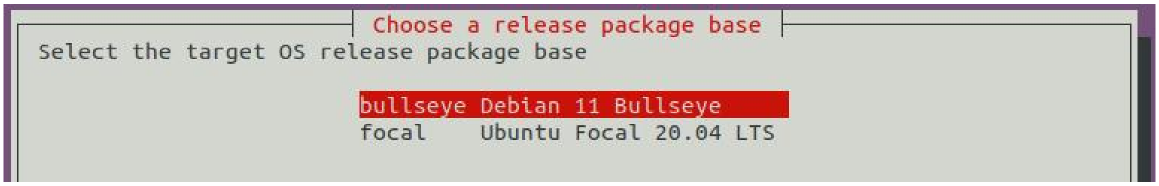
a. current will compile linux 5.10

b. legacy will compile linux4.4

d. next will compile linux5.18

```
Choose an option
Select the target kernel branch

        current Recommended. Come with best support
        legacy  Old stable / Legacy
        next    Use the latest kernel
```

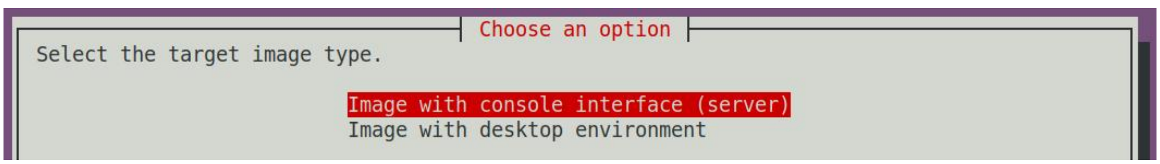
5) Then select the type of rootfs (the rootfs type supported by different branches will be different)



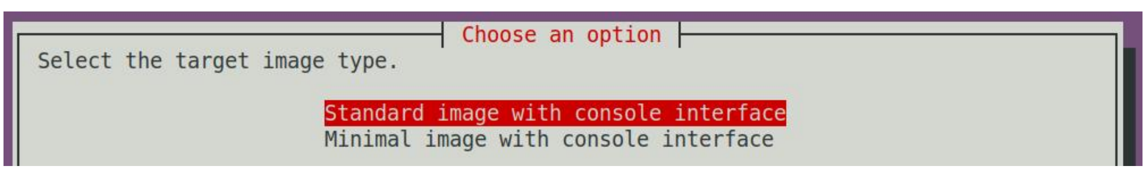
6) Then select the type of mirror

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small in size

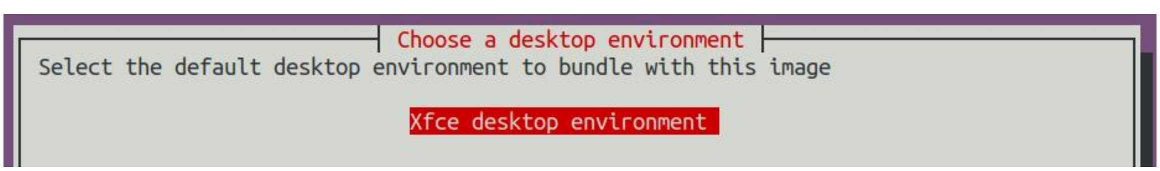
b. **Image with desktop environment** Indicates a mirror image with a desktop, which is relatively large in size

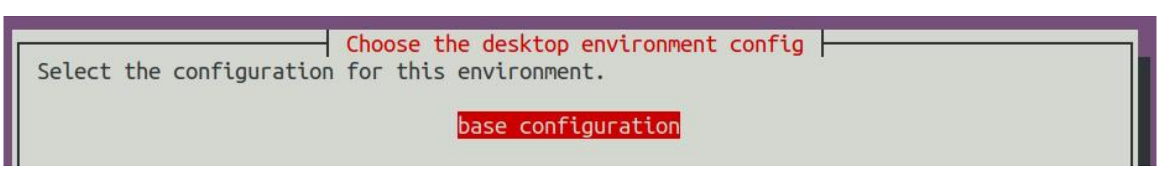


7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The software pre-installed in the Minimal version will be much less than the Standard version.



8) If you are compiling the desktop version of the image, you also need to select the type of desktop environment, but only XFCE is currently supported, so just press Enter.





9) You can then select additional packages that need to be installed. For example, if you need to install a browser, you can choose **browsers**. Which packages are included in each selection can be seen in the code of orangepi-build, you can also modify these configurations to add the packages you want to install

1. First enter **external/config/desktop** to see the desktop configuration folders of different linux distributions. Note that not all the Orange Pi development boards that can be seen in the code are supported and tested.

```
test@test:~/orangepi-build$ cd external/config/desktop
test@test:~/orangepi-build/external/config/desktop$ ls
bionic   bookworm   bullseye   buster   focal   jammy   README.md   sid
```

2. Then select the type of distribution you want to view or modify, and enter the corresponding directory, such as **bullseys**

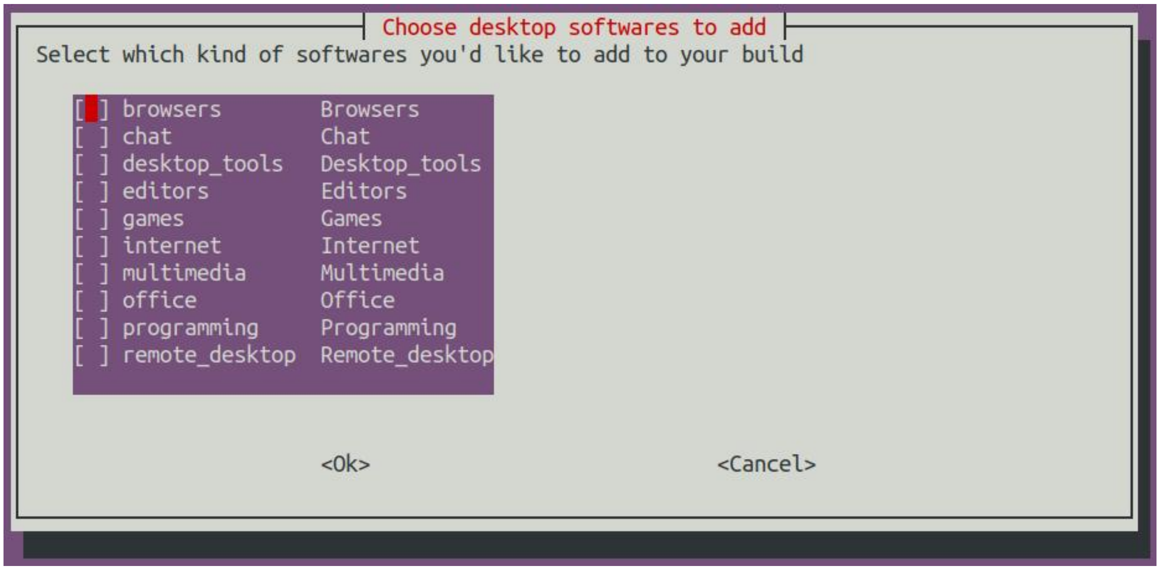
```
test@test:~/orangepi-build/external/config/desktop$ cd bullseye
test@test:~/orangepi-build/external/config/desktop/bullseye$ ls
appgroups   environments
```

3. Then enter the **appgroups** directory to see all app groups

```
test@test:~/orangepi-build/external/config/desktop/bullseye$ cd appgroups
test@test:~/orangepi-build/external/config/desktop/bullseye/appgroups$ ls
browsers   chat   desktop_tools   editors   games   internet   multimedia   office
programming   remote_desktop
```

4. Open the **packages** file under different groups to view the software contained in the group

```
test@test:~/orangepi-build/external/config/desktop/bullseye/appgroups$ cat
programming/packages
geany
thonny
test@test:~/orangepi-build/external/config/desktop/bullseye/appgroups$ cat
office/packages
ibreoffice
```



10) Then it will start to compile the linux image. The general process of compilation is as follows

a. Initialize the compilation environment of the Ubuntu PC and install the software packages required for the compilation process
b. Download the source code of u-boot and linux kernel (if cached, only update the code)
c. Compile u-boot source code and generate u-boot deb package
d. Compile the linux source code to generate linux-related deb packages
e. Make a deb package of linux firmware
f. Make the deb package of the orangepi-config tool
g. Make board-level supported deb packages
h. If you are compiling the desktop version of the image, you will also create a desktop-related deb package
i. Check whether the rootfs has been cached, if there is no cache, then recreate the rootfs, if it has been cached, directly decompress and use
j. Install the deb package generated earlier into rootfs
k. k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.
l. Then make an image file and format the partition, the default type is ext4
m. Copy the configured rootfs to the mirrored partition
n. then update initramfs
o. Finally, write the bin file of u-boot into the image through the dd command

11) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43/Orangepi4-lts_3.0.0_debian_bullseye_desktop_xfce_linux5.10.43.img ]
```

b. Compilation time used

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the command to compile the image, use the following command to start compiling the image directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepi4-lts BRANCH=current BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

6. Instructions for using Android SDK

The compilation of Android SDK is carried out on a PC with Ubuntu 14.04 installed. Other versions of Ubuntu systems may have some differences. The mirror download address of Ubuntu14.04 amd64 version is as follows

https://repo.huaweicloud.com/ubuntu-releases/14.04/ubuntu-14.04.6-desktop-amd64.iso

6. 1. Download the source code of Android SDK

1) First download the compressed package of Android SDK from Google Cloud Disk

RK3399-Android8.1.tar.gz01	4G	2022-03-08 16:48
RK3399-Android8.1.tar.gz02	4G	2022-03-08 16:48
RK3399-Android8.1.tar.gz00	4G	2022-03-08 16:47
RK3399-Android8.1.tar.gz03	3.91G	2022-03-08 16:41
RK3399-Android8.1.tar.gz.md5sum	244B	2022-03-08 15:43

2) After downloading the compressed package of Android SDK, please check whether the MD5 checksum is correct. If it is not correct, please download the source code again.

```
test@test:~$ md5sum -c RK3399-Android8.1.tar.gz.md5sum
RK3399-Android8.1.tar.gz00: Sure
RK3399-Android8.1.tar.gz01: Sure
RK3399-Android8.1.tar.gz02: Sure
RK3399-Android8.1.tar.gz03: Sure
```

3) Then you need to combine multiple compressed files into one, and then decompress

```
test@test:~$ cat RK3399-Android8.1.tar.gz* > RK3399-Android8.1.tar.gz
test@test:~$ tar -xvf RK3399-Android8.1.tar.gz
```

6. 2. Build Android Compilation Environment

1) Install JDK

```
test@test:~$ sudo add-apt-repository ppa:openjdk-r/ppa
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install openjdk-8-jdk
```

2) Configure JAVA environment variables

a. First determine the installation path of java, generally

```
test@test:~$ ls /usr/lib/jvm/java-8-openjdk-amd64
ASSEMBLY_EXCEPTION  bin  docs  include  jre  lib  man  src.zip
THIRD_PARTY_README
```

b. Then use the following command to export java environment variables

```
test@test:~$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
test@test:~$ export PATH=$JAVA_HOME/bin:$PATH
test@test:~$ export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar
```

3) Install platform support software

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install 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 u-boot-tools
```

6. 3. Compile Android image

6. 3. 1. Compiling u-boot

1) The compilation method of u-boot is as follows

```
test@test:~$ cd RK3399-Android8.1/
test@test:~/RK3399-Android8.1$ ./make.sh -B
```

2) After the compilation is successful, the output content is as follows

```
out:trust.img
merge success(trust.img)
load addr is 0x200000!
pack input u-boot.bin
pack file size: 682652
crc = 0xc21153c6
pack uboot.img success!
```

6. 3. 2. Compile the kernel

1) The compilation method of the kernel is as follows

```
test@test:~/RK3399-Android8.1$ ./make.sh -K
```

2) After the compilation is successful, the output content is as follows

```
scripts/kconfig/conf    --silentoldconfig Kconfig
   CHK        include/config/kernel.release
   CHK        include/generated/uapi/linux/version.h
......
```

```
make[2]: "include/generated/vdso-offsets.h" is up to date.
  Building modules, stage 2.
  MODPOST 13 modules
Pack to resource.img successed!
  Image:   resource.img (with rk3399-orangepi-4-lts.dtb logo.bmp logo_kernel.bmp) is ready
  Image:   boot.img (with Image resource.img) is ready
```

6. 3. 3. Compile android

1) The compilation method of android is as follows

```
test@test:~/RK3399-Android8.1$ ./make.sh -A
```

2) After the compilation is successful, the output content is as follows

```
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
......
[100% 129/129] Install system fs image: out/target/product/rk3399_mid/system.img
out/target/product/rk3399_mid/system.img+out/target/product/rk3399_mid/obj/PACKAGING/recovery_patch_intermediates/recovery_from_boot.p maxsize=2740531200 blocksize=135168 total=1198447603 reserve=27709440

#### build completed successfully (03:53 (mm:ss)) ####
```

6. 3. 4. Packaging the full image

1) The android image packaging method is as follows

```
test@test:~/RK3399-Android8.1$ ./make.sh -M -u
```

2) After the compilation is successful, the output content is as follows

```
create uboot.img...done.
create trust.img...done.
create loader...done.
......
Make firmware OK!
```

```
------ OK ------
********RKImageMaker ver 1.63********
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.
```

3) After the compilation is completed, the generated image file will be placed under **rockdev/Image-rk3399_mid/**. Where update.img is the Android firmware that can be burned and run

```
test@test:~/RK3399-Android8.1$ cd rockdev/Image-rk3399_mid/
test@test:~/RK3399-Android8.1$ ls update*
update.img
```