OrangePi Zero 3
User Manual

Catalog

1. Basic features of OrangePi Zero 3

1. 1. What is OrangePi Zero 3

OrangePi is an open source single-board card computer, a new generation of arm64 development board, which can run Android TV 12, Ubuntu and Debian and other operating systems. OrangePi Zero 3 uses Allwinner H618 system-on-a-chip and has 1GB or 1.5GB or 2GB or 4GB LPDDR4 memory.

1. 2. Purpose of OrangePi Zero 3

We can use it to achieve:

- A small Linux desktop computer
- A small Linux web server
- Install the Klipper host computer to control the 3D printer
- Android TV box

Of course, there are more functions. Relying on a powerful ecosystem and a variety of expansion accessories, OrangePi can help users easily realize the delivery from idea to prototype to mass production. It is a maker, dreamer, hobby The ideal creative platform for readers.

1. 3. Who is OrangePi Zero 3 designed for?

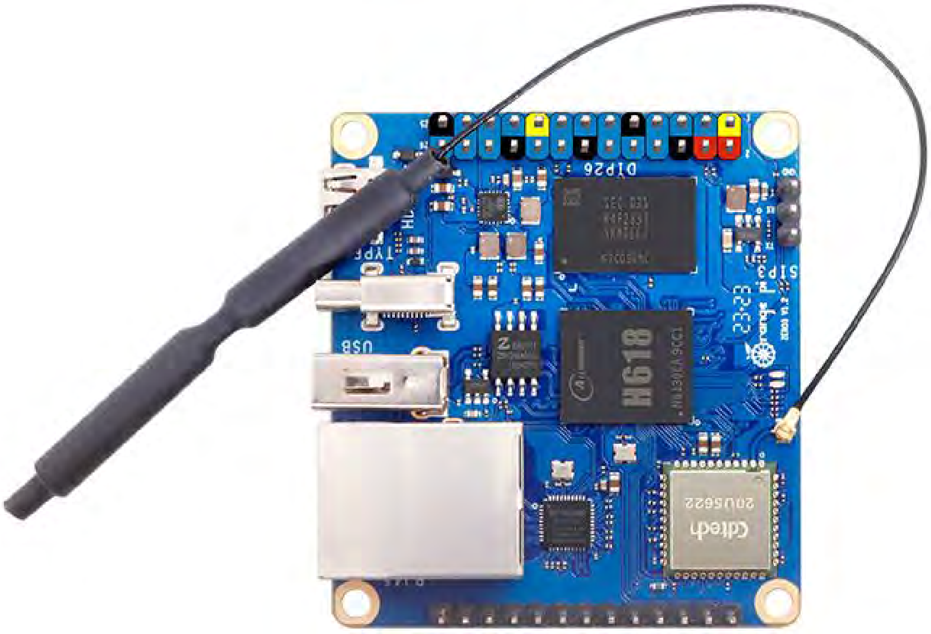
The OrangePi development board is not just a consumer product, but is designed for anyone who wants to use technology to create and innovate. It's a simple, fun, and useful tool you can use to shape the world around you.

1. 4. Hardware Features of OrangePi Zero 3

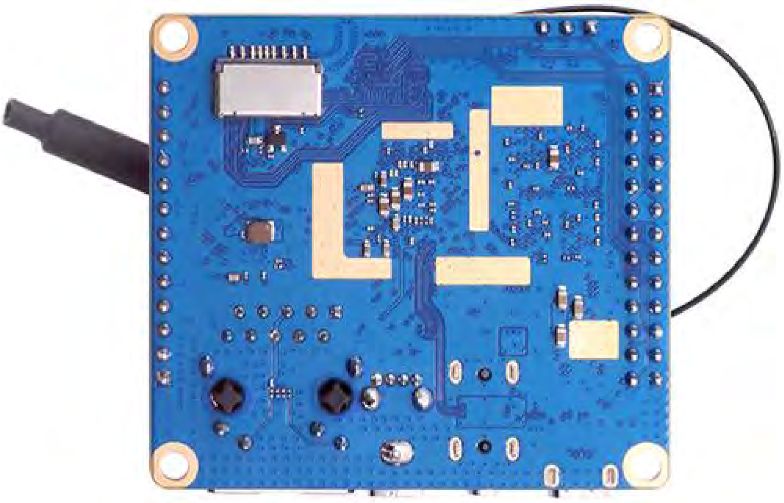
Introduction to hardware features	
CPU	Allwinner H618 quad-core 64-bit 1.5GHz high-performance Cortex-A53 processor
GPU	Mali G31 MP2 Supports OpenGL ES 1.0/2.0/3.2、OpenCL 2.0
Memory	1GB/1.5GB/2GB/4GB LPDDR4 (shared with GPU)
Onboard storage	micro SD card slot, 16MB SPI Flash
Ethernet	Support 10/100M/1000M Ethernet
WIFI+Bluetooth	• CDW-20U5622 chip, support IEEE 802.11 a/b/g/n/ac, BT5.0
Video Output	• Micro HDMI 2.0a • TV CVBS output, support PAL/NTSC (via 13pin expansion board)
Audio output	• Micro HDMI output • 3.5mm audio port (via 13pin expansion board)
Power Supply	USB Type C interface input
USB 2.0 port	3* USB 2.0 HOST (two of them are through 13pin expansion board)
26pin connector	With I2Cx1, SPIx1, UARTx1 and multiple GPIO ports
13pin connector	With USB 2.0 HOSTx2, TV-OUT, LINE OUT, IR-RX, and 3 GPIO ports
Debug serial port	UART-TX, UART-RX and GND
LED light	Power light and status light
Infrared receiver	Support infrared remote control (via 13pin expansion board)
Supported OS	Android12 TV, Ubuntu, Debian, etc.
Introduction of Appearance Specifications	
Product Size	85mm×56mm
Weight	30g

1. 5. Top view and bottom view of OrangePi Zero 3

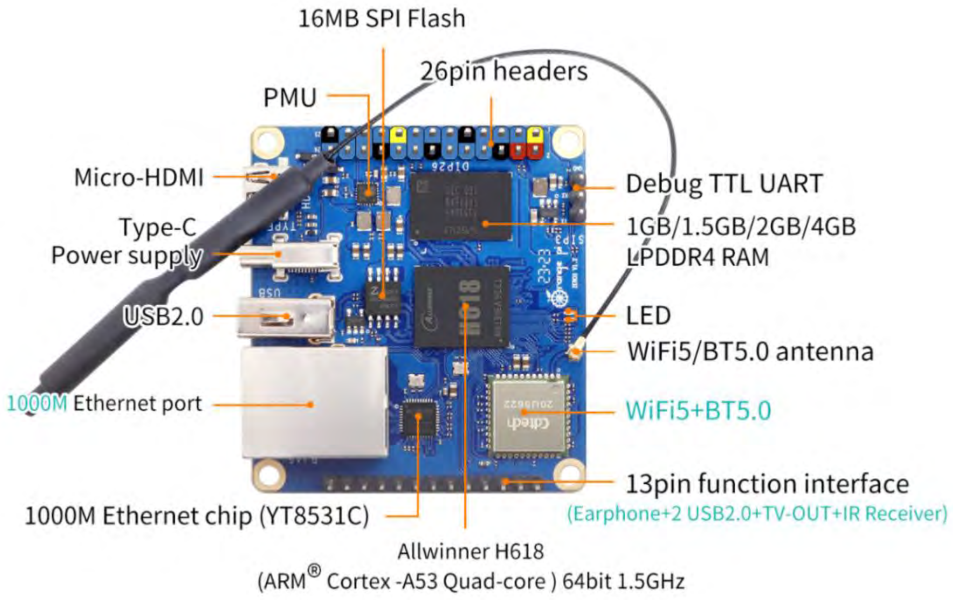
Top view:



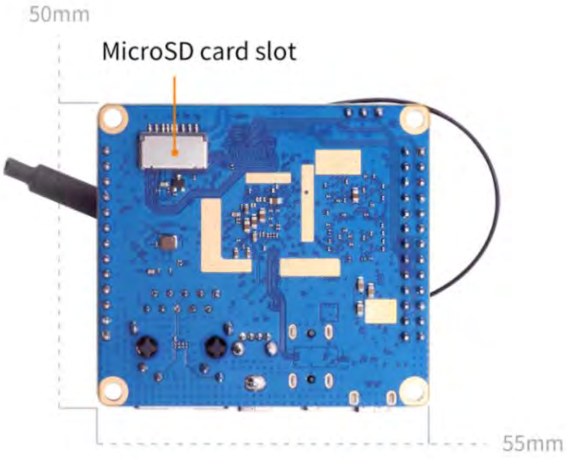
Bottom view:



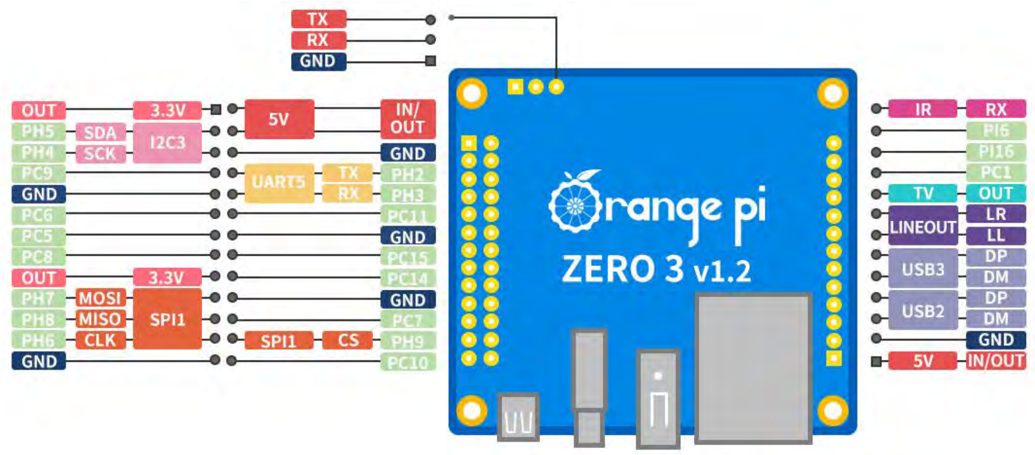
1.6. Interface details of OrangePi Zero 3



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 required accessories

1) Micro SD card, a high-speed SanDisk card of class 10 or above with a minimum capacity of 8GB

Using other brands of micro SD cards (non-SanDisk micro SD cards), as shown in the picture below (including but not limited to these cards), some friends have reported that there will be problems during the system startup process, such as the system is stuck halfway through startup, Or the reboot command cannot be used normally, and it was finally resolved after changing the SanDisk micro SD card. So if you use a non-SanDisk micro SD card and find that there is a problem with the system startup or use, please replace the SanDisk micro SD card and then test.

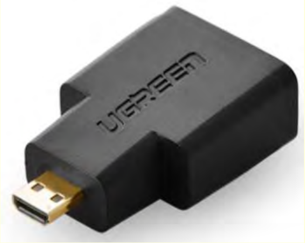
Current feedback is that there are some micro SD cards that have problems starting on OrangePi Zero 3

In addition, the micro SD card that can be used normally on other types of development boards does not guarantee that it can be started normally on OrangePi Zero 3, please pay special attention to this point.

2) Micro SD card reader, used to read and write micro SD card

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

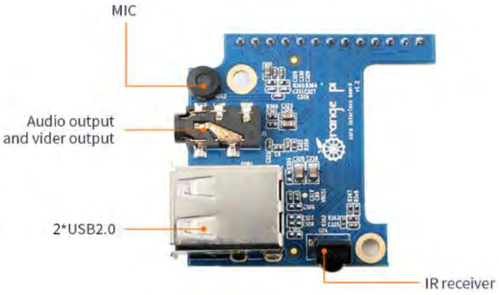
Note, please do not use the relatively wide Micro HDMI adapter as shown in the figure below, because the distance between the Micro HDMI interface of the development board and the Type-C power interface is relatively small, it may cause that the two cannot be inserted into the development board at the same time. plate.



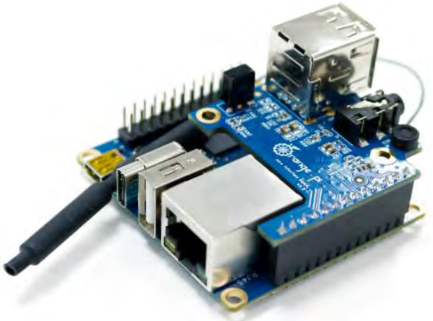
4) Power supply, if you have a 5V/2A or 5V/3A power head, you only need to prepare a USB-to-Type C interface data cable as shown in the picture on the left below, and you can also use a cable similar to the picture on the right below A 5V/2A or 5V/3A high-quality USB Typc C interface power adapter integrated with the power head.

5) 13pin expansion board

a. The physical object of the expansion board is as follows



b. The way to insert the expansion board into the development board is as follows, remember not to insert it backwards

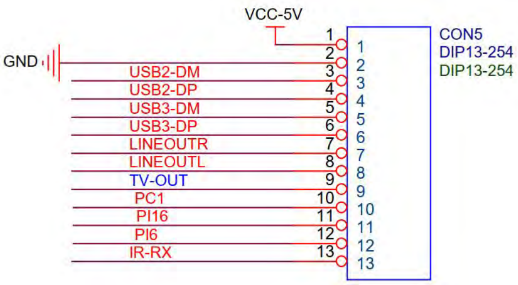


c. The 13pin pin header on the OrangePi Zero 3 development board can be connected to the expansion board to expand the functions that are not on the development board. The functions that the expansion board can use are as follows

1	Microphone (Mic)	**No support, no support, no support! ! !**

		The 13pin expansion board is a general-purpose expansion board, which is suitable for various development boards of Orange Pi, but the 13pin interface of OrangePi Zero3 has no Mic function, so although there is a Mic on the 13pin expansion board, it is on the OrangePi Zero 3. Unusable, the 13pin expansion board is mainly used to expand functions other than Mic on OrangePi Zero 3.
2	Analog audio and video output interface	Supported, it can be used to connect headphones to play music, or connect to TV through AV cable to output analog audio and video signals **(Android system only).**
3	USB 2.0 Host x 2	Support, used to connect USB keyboard, mouse and USB storage device.
4	Infrared receiving function	Support, Android system can be controlled by infrared remote control.

d. The schematic diagram of the 13pin header of the OrangePi Zero 3 development board is shown below



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

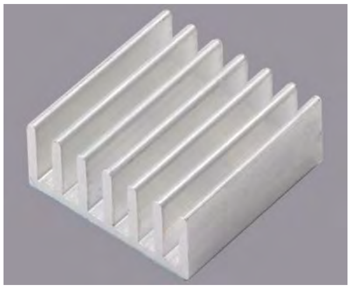
7) Infrared remote control, mainly used to control Android TV system

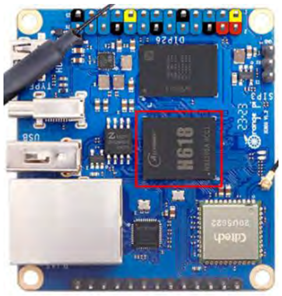
Note that the remote control of the air conditioner or the remote control of the TV cannot control the OrangePi development board. By default, only the remote control provided by OrangePi can.

8) 100M or 1000M network cable, used to connect the development board to the Internet

9) AV video cable, if you want to display video through the AV interface instead of the HDMI interface, then you need to connect the development board to the TV through the AV video cable

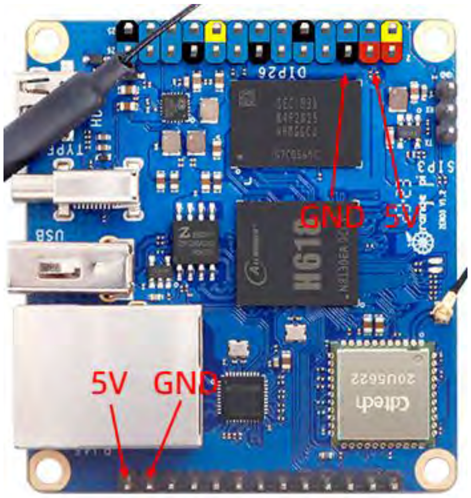
10) Heat sink, if you are worried that the temperature of the development board is too high, you can add a heat sink, and the heat sink can be pasted on the H618 chip





11) 5V cooling fan, as shown in the figure below, both 5V and GND pins on the 26pin and 13pin interfaces of the development board can be connected to the cooling fan. The spacing between the 26pin and 13pin headers is 2.54mm. Refer to this Specifications can be purchased.

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



12) Matching shell **(pictures to be added)**

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

Note that the TTL level used by the development board is 3.3v. In addition to the USB to TTL module shown in the above figure, other similar 3.3v USB to TTL modules are generally available.

14) X64 computer with Ubuntu and Windows operating systems installed

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

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

1) The website for downloading the English version is

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-Zero-3.html

2) The information mainly includes

a. **Android source code:** Save on Google Drive
b. **Linux source code:** Save on Github
c. **Android source code:** Save on Google Drive
d. **Ubuntu source code:** Save on Google Drive
e. **Debian source code:** Save on Google Drive
f. **User Manual and Schematic:** Chip-related data sheets will also be placed here
g. **Official tool:** It mainly includes the software that needs to be used during the use of the development board

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

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

Linux distributions such as Debian or Ubuntu downloaded from the OrangePi data download page.

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

1) First prepare a micro SD card with a capacity of 8GB or more. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

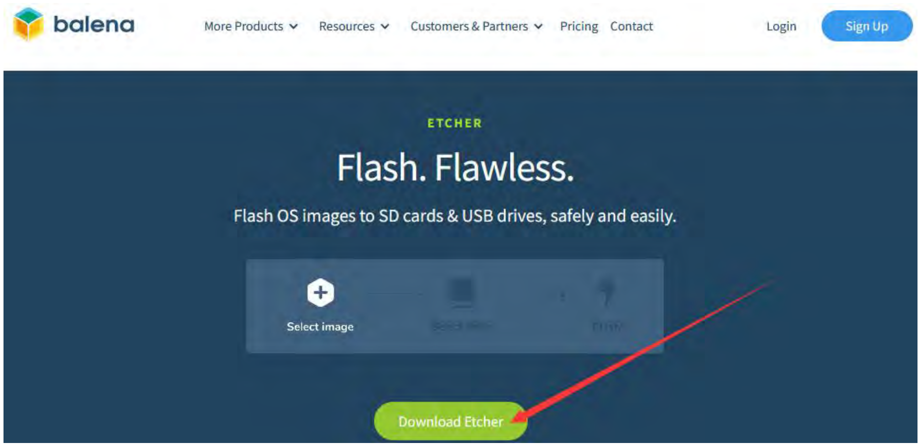
2) Then use the card reader to insert the micro SD card into the computer

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

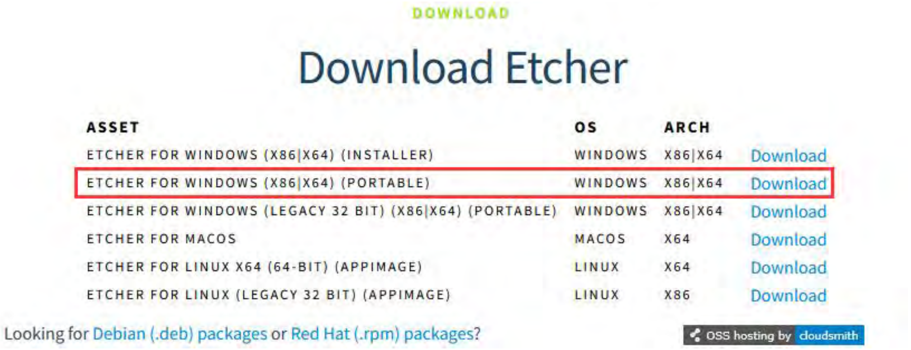
4) Then download the burning software of Linux image——**balenaEtcher,** the download address is

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

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



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



DOWNLOAD

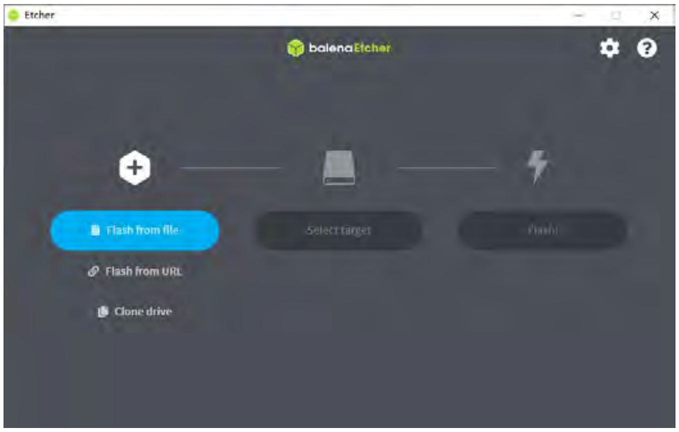
Download Etcher

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

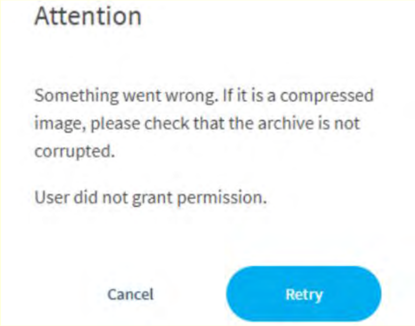
Looking for Debian (.deb) packages or Red Hat (.rpm) packages?

OSS hosting by cloudsmith

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



When opening balenaEtcher, if the following error is prompted:

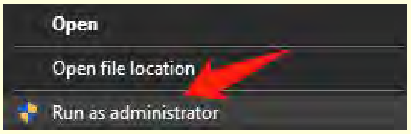


Attention

Something went wrong. If it is a compressed image, please check that the archive is not corrupted.

User did not grant permission.

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

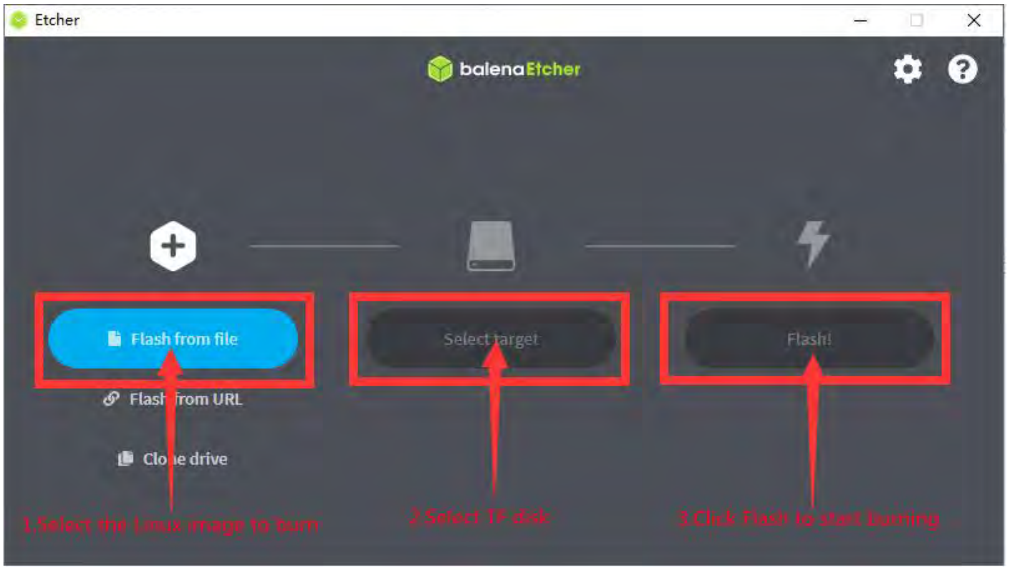


8) The specific steps to use balenaEtcher to burn the Linux image are as follows

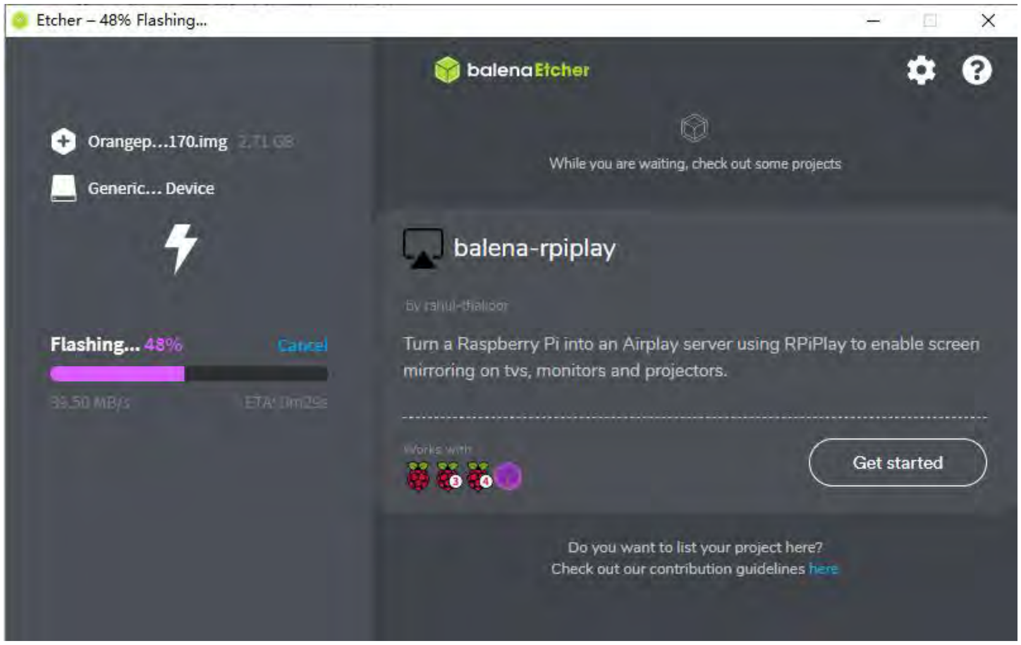
a. First select the path of the Linux image file to be burned

b. Then select the drive letter of the micro SD card

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



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

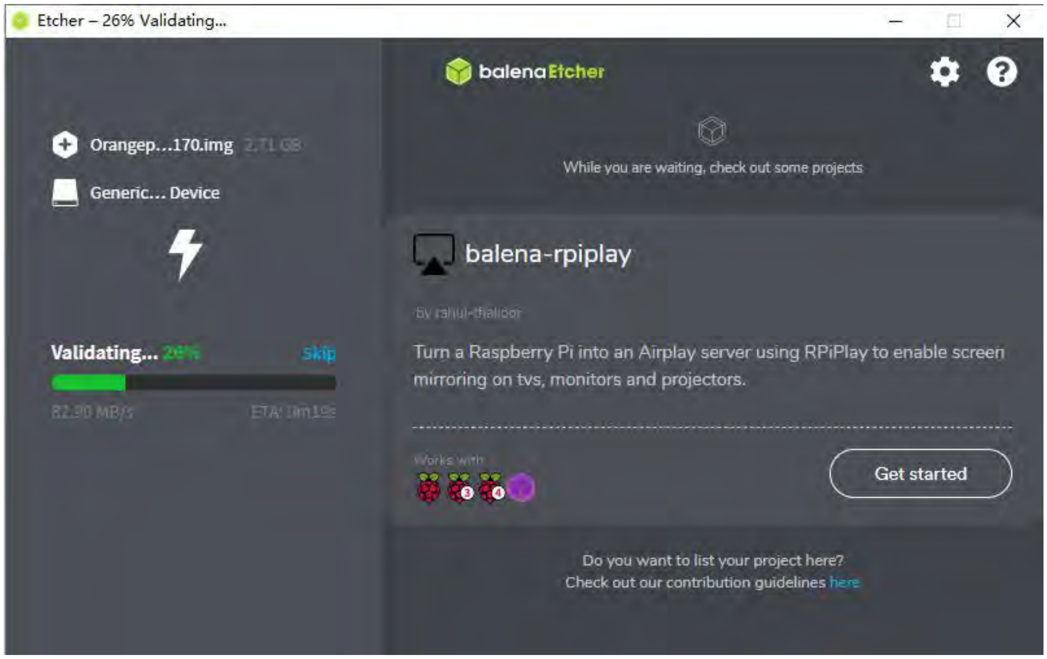


While you are waiting, check out some projects

Turn a Raspberry Pi into an Airplay server using RPiPlay to enable screen mirroring on tvs, monitors and projectors.

Do you want to list your project here?
Check out our contribution guidelines here

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


Etcher – 26% Validating...

Orangep...170.img 2.71 GB

Generic... Device

Validating... 26%

82.30 MB/s ETA: 0m19s

While you are waiting, check out some projects

balena-rpiplay

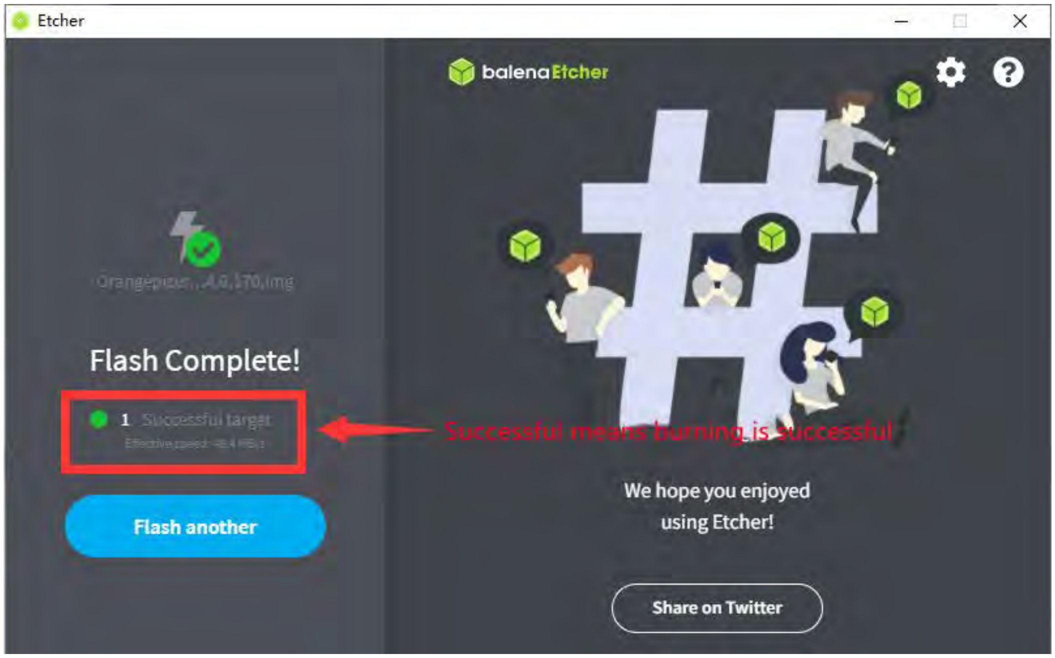
by rahul-thakoor

Turn a Raspberry Pi into an Airplay server using RPiPlay to enable screen mirroring on tvs, monitors and projectors.

Works with

Do you want to list your project here?
Check out our contribution guidelines here

11) After successful burning, the display interface of balenaEtcher is as 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 micro SD card and insert it into the micro SD card slot of the development board. .



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

1) First prepare a micro SD card with a capacity of 8GB or more. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

2) Then use the card reader to insert the micro SD card into the computer

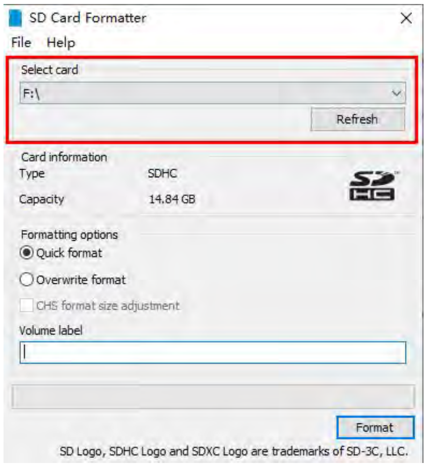
3) Then format the micro SD card

a. **SD Card Formatter** can be used to format the micro SD card. The download link is

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

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

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



Select card
F:\

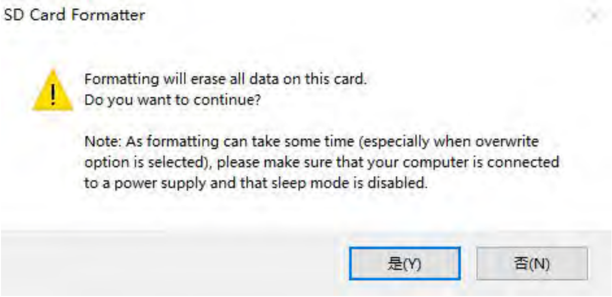
Card information
Type SDHC
Capacity 14.84 GB

Formatting options
● Quick format
○ Overwrite format
□ CHS format size adjustment

Volume label

SD Logo, SDHC Logo and SDXC Logo are trademarks of SD-3C, LLC.

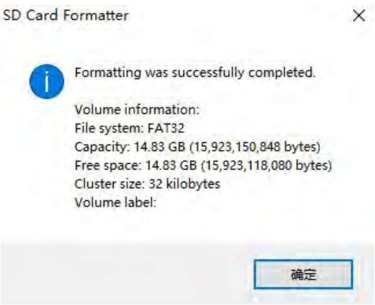
d. hen click " **Format** ", a warning box will pop up before formatting, and formatting will start after selecting "Yes (Y)"



Formatting will erase all data on this card.
Do you want to continue?

Note: As formatting can take some time (especially when overwrite option is selected), please make sure that your computer is connected to a power supply and that sleep mode is disabled.

e. After formatting the micro SD card, the message shown in the figure below will pop up, click OK



Formatting was successfully completed.

Volume information:
File system: FAT32
Capacity: 14.83 GB (15,923,150,848 bytes)
Free space: 14.83 GB (15,923,118,080 bytes)
Cluster size: 32 kilobytes
Volume label:

4) Download the Linux operating system image file compression package that you want to burn from the **OrangePi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system. The size is generally more than 1GB

5) Use **Win32Diskimager** to burn the Linux image to the micro SD card

a. The download page of Win32Diskimager is

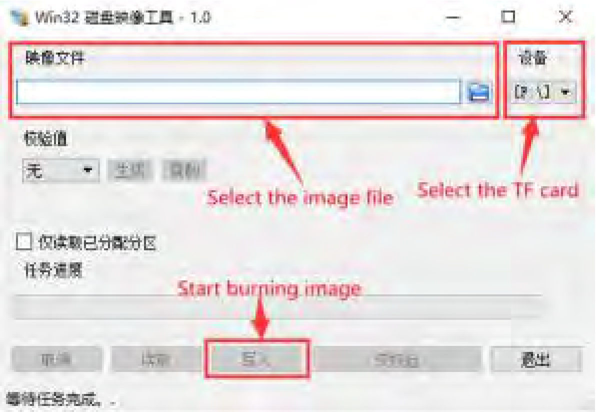
http://sourceforge.net/projects/win32diskimager/files/Archive/

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

a) First select the path of the image file

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

c) Finally click "**Write**" to start burning



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

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

Note that the Linux image mentioned here specifically refers to the image of Linux distributions such as Debian or Ubuntu downloaded from the OrangePi data download page, and the Ubuntu PC refers to the personal computer with the Ubuntu system installed.

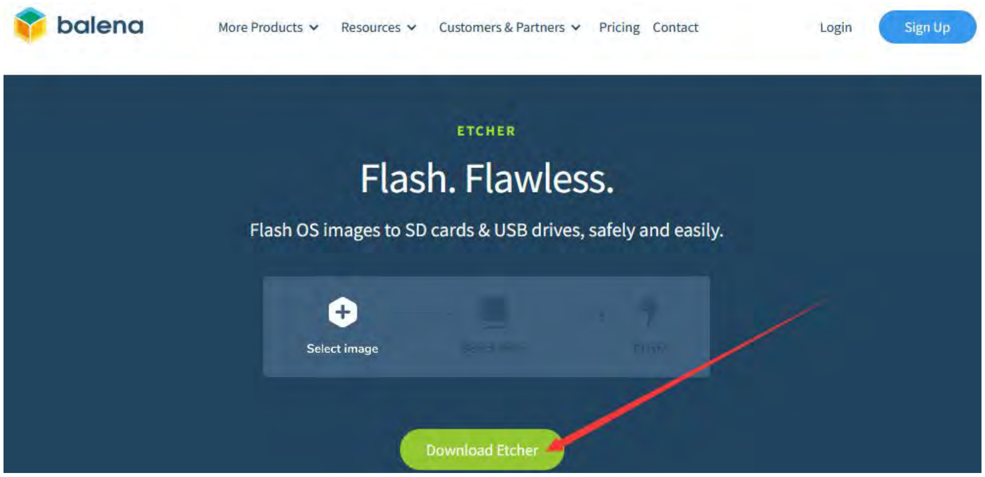
1) First prepare a micro SD card with a capacity of 8GB or more. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

2) Then use the card reader to insert the micro SD card into the computer

3) Download the balenaEtcher software, the download address is

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

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

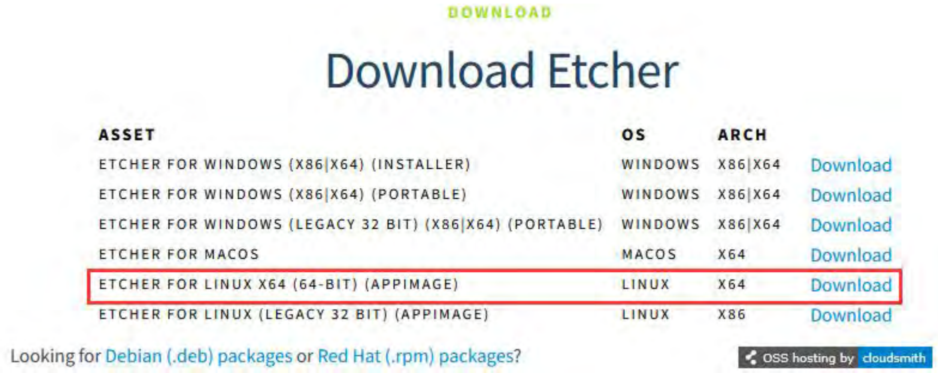


ETCHER

Flash. Flawless.

Flash OS images to SD cards & USB drives, safely and easily.

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



DOWNLOAD

Download Etcher

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

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

6) Download the image file compression package of the Linux operating system that you want to burn from the **OrangePi data download page**, and then use the decompression software to decompress it. Among the decompressed files, the file ending with "**.img**" is the image file of the operating system. The size is generally above 1GB. The decompression command for the compressed package ending in 7z is as follows:

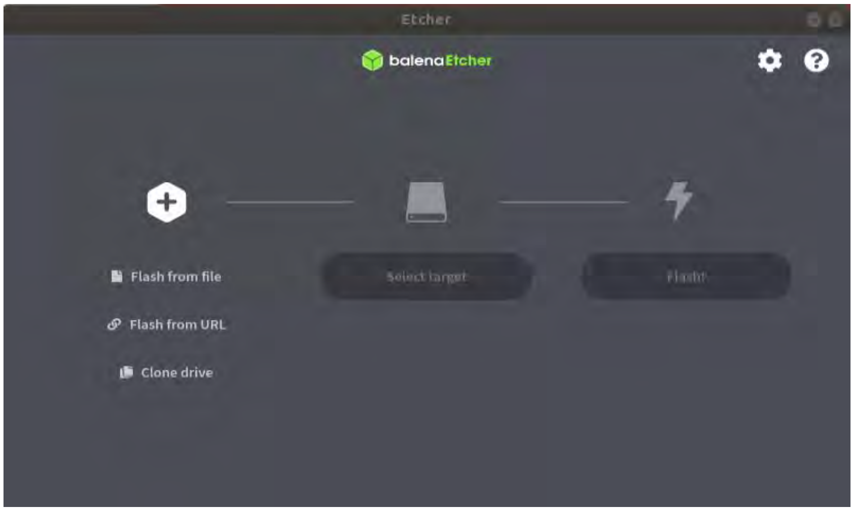
```
test@test:~$ 7z x orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.7z
```

```
test@test:~$ ls orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.*
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.7z
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.sha     # checksum file
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.img      # mirror file
```

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

```
test@test:~$ sha256sum -c *.sha
orangepizero3_1.0.0_ubuntu_focal_desktop_linux6.1.31.img: success
```

8) Then double-click **balenaEtcher-1.14.3-x64.AppImage** on the graphical interface of Ubuntu PC to open balenaEtcher (no installation required), and the interface after balenaEtcher is opened is displayed as shown in the figure below

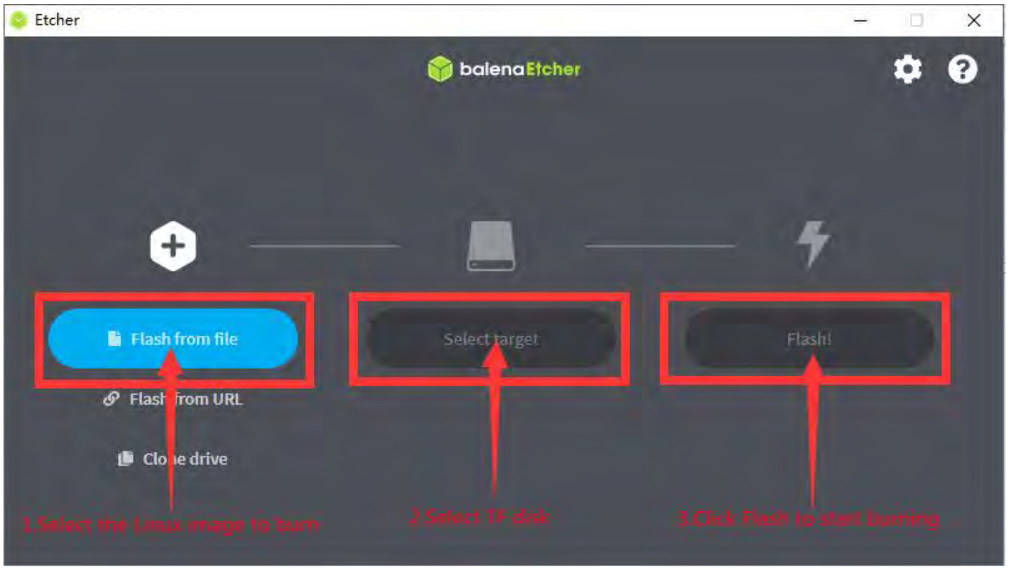


9) The specific steps to use balenaEtcher to burn the Linux image are as follows

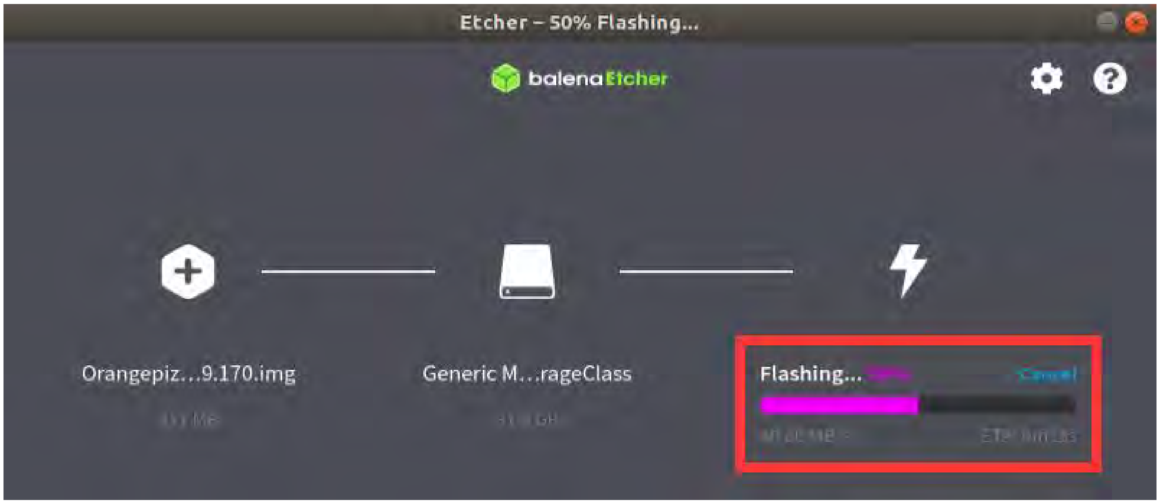
a. First select the path of the Linux image file to be burned

b. Then select the drive letter of the micro SD card

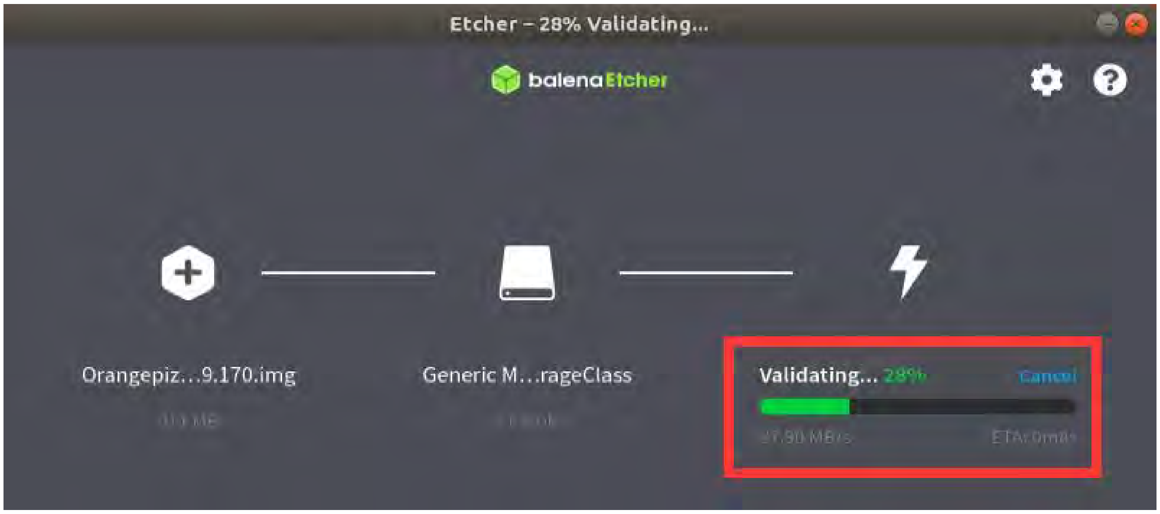
c. Finally click Flash to start burning the Linux image to the micro SD card



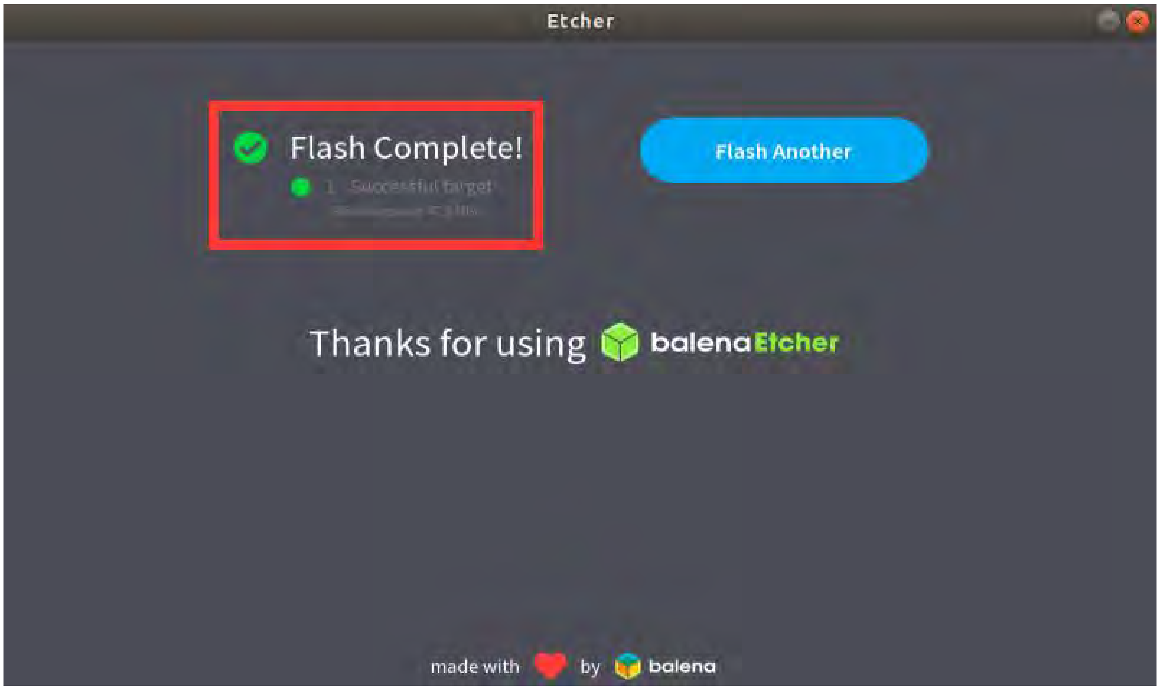
10) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the micro SD card



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



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



2. 5. How to burn Android image to micro SD card

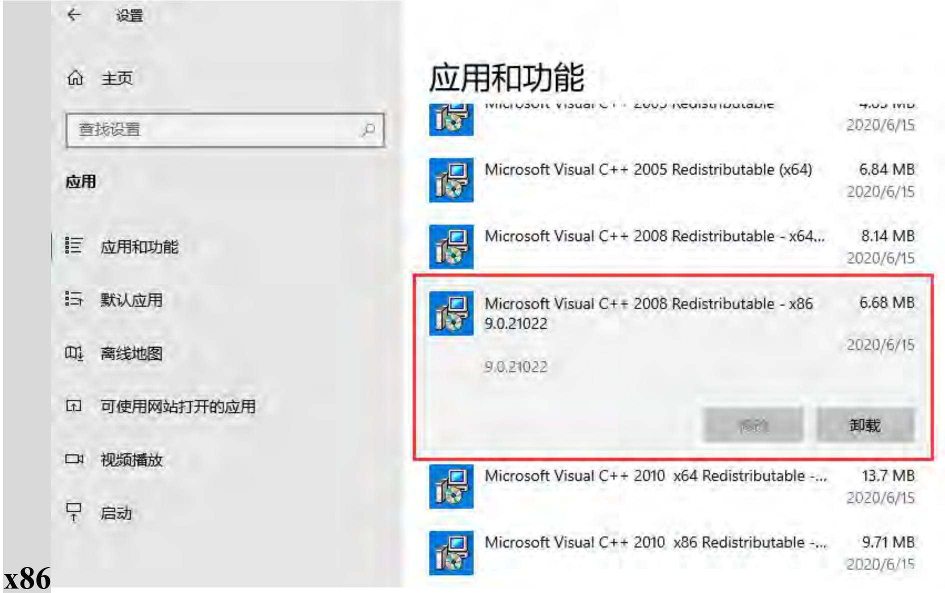
The Android image of the development board can only be burned into the micro

SD card under the Windows platform using the PhoenixCard software, and the version of the PhoenixCard software must be PhonixCard-4.2.8.

Please do not use software for burning Linux images, such as Win32Diskimager or balenaEtcher, to burn Android images.

In addition, the PhoenixCard software does not have versions for Linux and Mac platforms, so it is impossible to burn Android images to micro SD cards under Linux and Mac platforms.

1) First, please make sure that the Windows system has installed **Microsoft Visual C++ 2008 Redistrbutable - x86**



应用和功能

Microsoft Visual C++ 2005 Redistributable 2020/6/15

Microsoft Visual C++ 2005 Redistributable (x64) 6.84 MB 2020/6/15

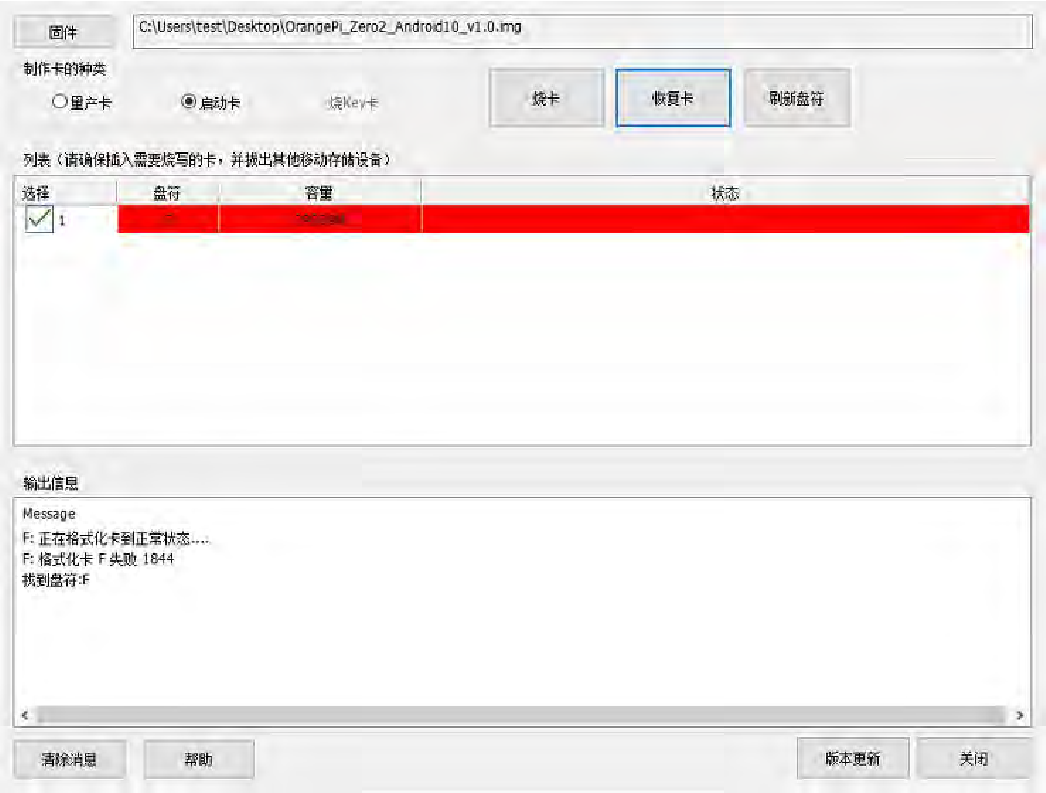
Microsoft Visual C++ 2008 Redistributable - x64... 8.14 MB 2020/6/15

Microsoft Visual C++ 2008 Redistributable - x86 9.0.21022 6.68 MB 2020/6/15 9.0.21022

Microsoft Visual C++ 2010 x64 Redistributable -... 13.7 MB 2020/6/15

Microsoft Visual C++ 2010 x86 Redistributable -... 9.71 MB 2020/6/15

2) If not installed **Microsoft Visual C++ 2008 Redistrbutable - x86**, Using PhoenixCard to format micro SD card or burn Android image will prompt the following error



固件 C:\Users\test\Desktop\OrangePi_Zero2_Android10_v1.0.img

制作卡的种类

○量产卡 ●启动卡 烧Key卡

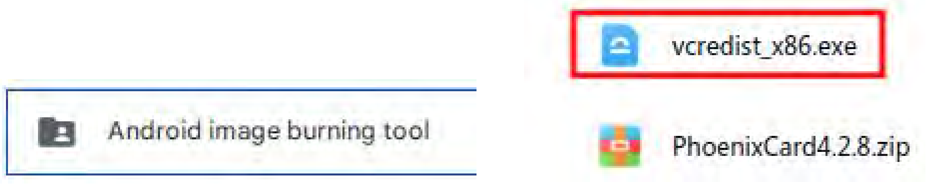
列表（请确保插入需要烧写的卡，并拔出其他移动存储设备）

选择	盘符	容量	状态
☑ 1			

输出信息

```
Message
F: 正在格式化卡到正常状态.....
F: 格式化卡 F 失败 1844
找到盘符:F
```

3) **Microsoft Visual C++ 2008 Redistrbutable - x86** the installation package can be downloaded from the **official tool** of OrangePi Zero 3, or you can go to the **official website of Microsoft** to download

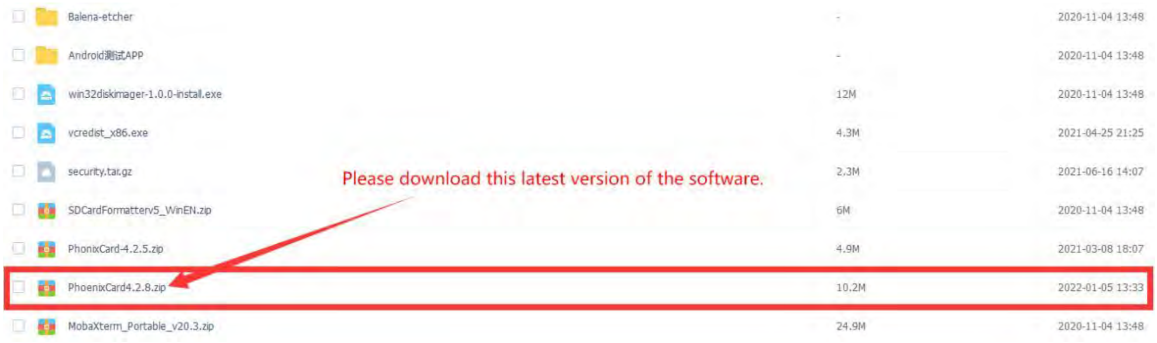


4) Then prepare a micro SD card with 8GB or larger capacity. The transmission speed of the micro SD card must be **class 10 or above**. It is recommended to use a micro SD card of SanDisk and other brands

5) Then use the card reader to insert the micro SD card into the computer

6) Download the Android image and PhoenixCard programming tool from the **OrangePi data download page**. **Please ensure that the version of the PhonenixCrad tool is PhonixCard-4.2.8.** Please do not use the PhonixCard software version lower than 4.2.8

to burn the Android image. There may be problems with the Android image flashed by this version of the PhonixCard tool

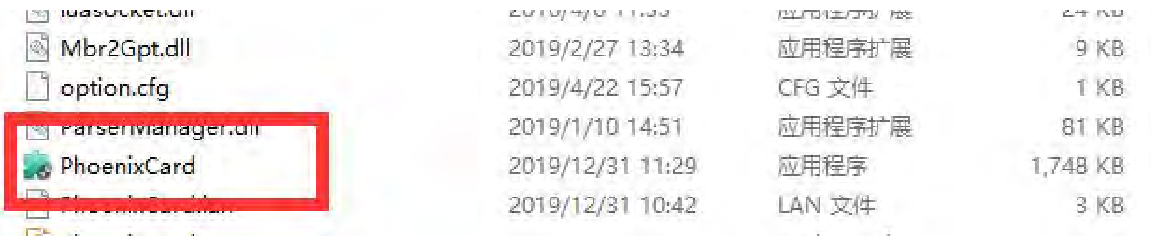


□ Balena-etcher	-	2020-11-04 13:48
□ Android测试APP	-	2020-11-04 13:48
□ win32diskimager-1.0.0-install.exe	12M	2020-11-04 13:48
□ vcredist_x86.exe	4.3M	2021-04-25 21:25
□ security.tar.gz	2.3M	2021-06-16 14:07
□ SDCardFormatterv5_WinEN.zip	6M	2020-11-04 13:48
□ PhonixCard-4.2.5.zip	4.9M	2021-03-08 18:07
□ PhoenixCard4.2.8.zip	10.2M	2022-01-05 13:33
□ MobaXterm_Portable_v20.3.zip	24.9M	2020-11-04 13:48

Please download this latest version of the software.

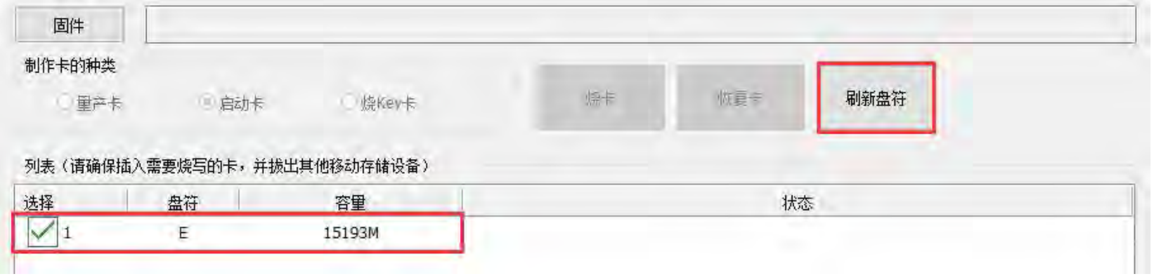
7) Then use the decompression software to decompress the compressed package of the downloaded Android image. Among the decompressed files, the file ending with ".img" is the Android image file, and the size is more than 1GB.

8) Then use decompression software to decompress **PhonixCard4.2.8.zip**, this software does not need to be installed, just find PhoenixCard in the decompressed folder and open it



Mbr2Gpt.dll	2019/2/27 13:34	应用程序扩展	9 KB
option.cfg	2019/4/22 15:57	CFG 文件	1 KB
ParserManager.dll	2019/1/10 14:51	应用程序扩展	81 KB
PhoenixCard	2019/12/31 11:29	应用程序	1,748 KB
PhoenixCard.lan	2019/12/31 10:42	LAN 文件	3 KB

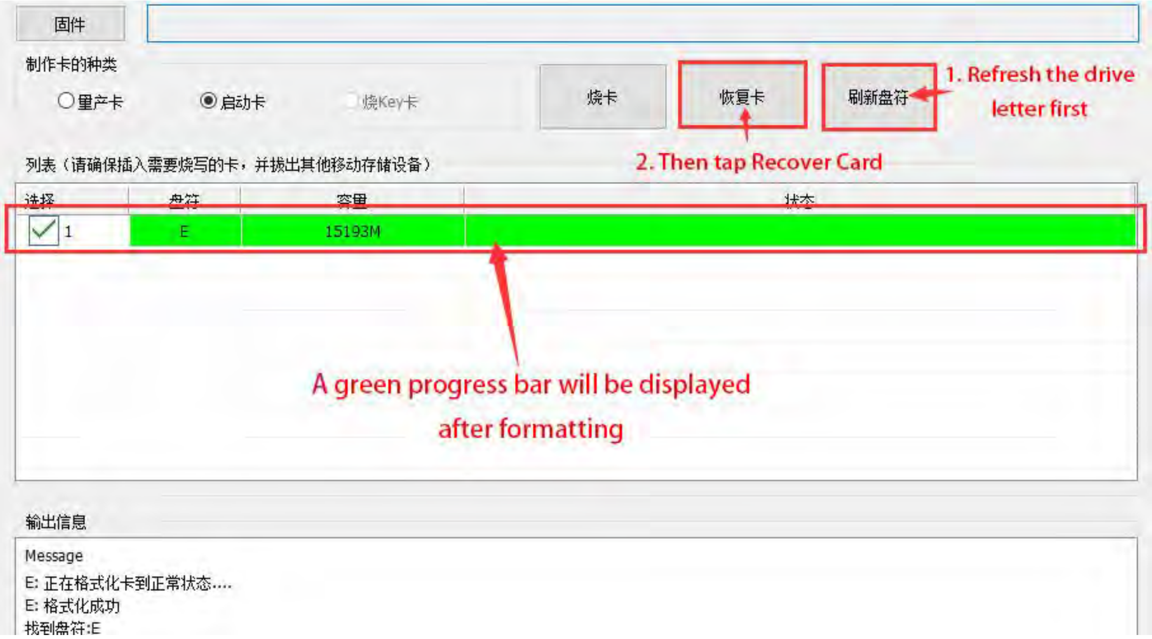
9) After opening PhoenixCard, if the micro SD card is recognized normally, the drive letter and capacity of the micro SD card will be displayed in the middle list. **Please make sure that the displayed drive letter is consistent with the drive letter of the micro SD card you want to burn**. If there is no display, you can try to unplug the micro SD card, or click the "**Refresh Drive Letter**" button in PhoenixCard



列表（请确保插入需要烧写的卡，并拔出其他移动存储设备）

选择	盘符	容量	状态
☑ 1	E	15193M	

10) After confirming the drive letter, format the micro SD card first, and click the "**Recover Card**" button in PhoenixCard (if the "**Recover Card**" button is gray and cannot be pressed, you can first click the "**Refresh Drive Letter**" button)


固件
制作卡的种类
○量产卡
●启动卡
烧Key卡
烧卡
恢复卡
刷新盘符
1. Refresh the drive letter first
2. Then tap Recover Card
列表（请确保插入需要烧写的卡，并拔出其他移动存储设备）
A green progress bar will be displayed after formatting
输出信息

选择	盘符	容量	状态
☑ 1	E	15193M	🟩

```
Message
E: 正在格式化卡到正常状态....
E: 格式化成功
找到盘符:E
```

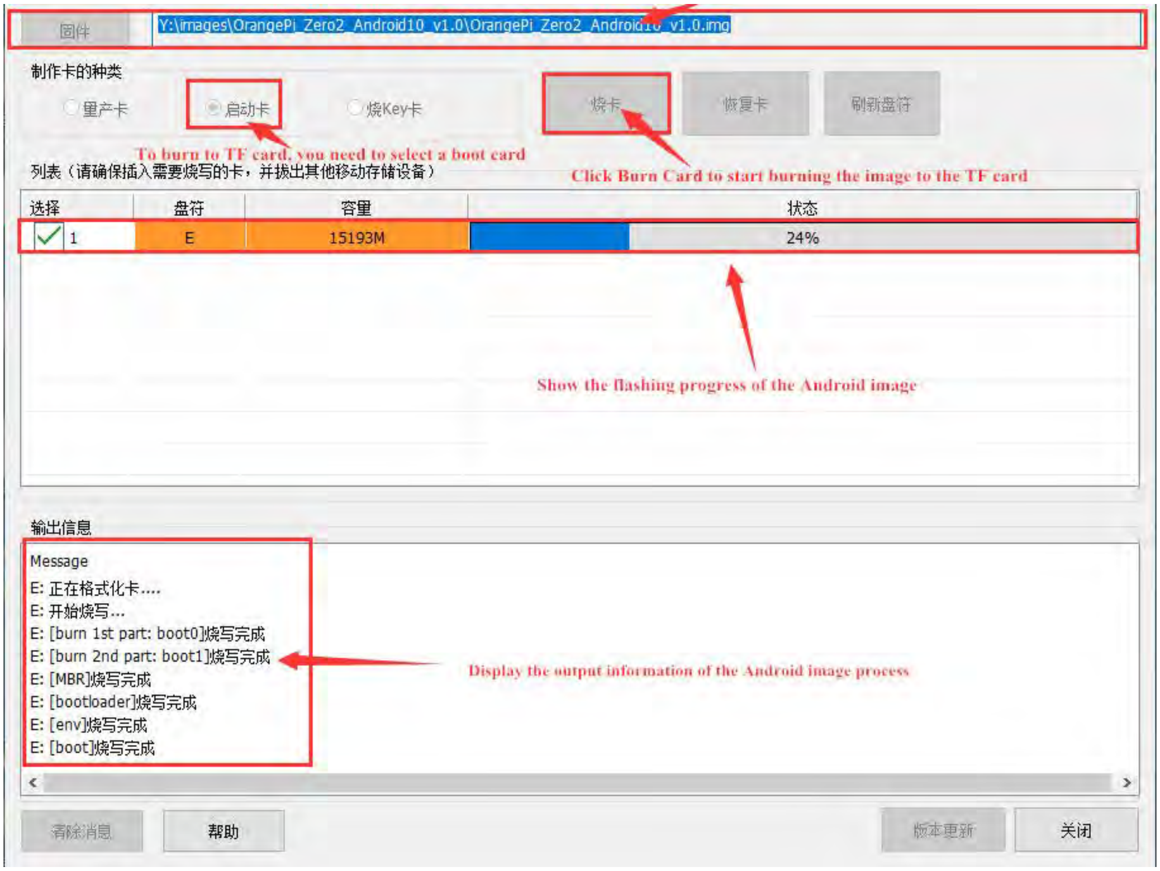
If there is a problem with formatting, please try to unplug and insert the micro SD card and then test again. If the problem still exists after re-plugging and inserting the micro SD card, you can restart the Windows computer or try another computer.

11) Then start to write the Android image into the micro SD card

a. First select the path of the Android image in the "**Firmware**" column

b. Select "**Activation Card**" in "Card Type"

c. Then click the "**burn card**" button to start burning



固件 Y:\images\OrangePi_Zero2_Android10_v1.0\OrangePi_Zero2_Android10_v1.0.img

制作卡的种类

○ 量产卡 ● 启动卡 ○ 烧Key卡

To burn to TF card, you need to select a boot card

Click Burn Card to start burning the image to the TF card

列表（请确保插入需要烧写的卡，并拔出其他移动存储设备）

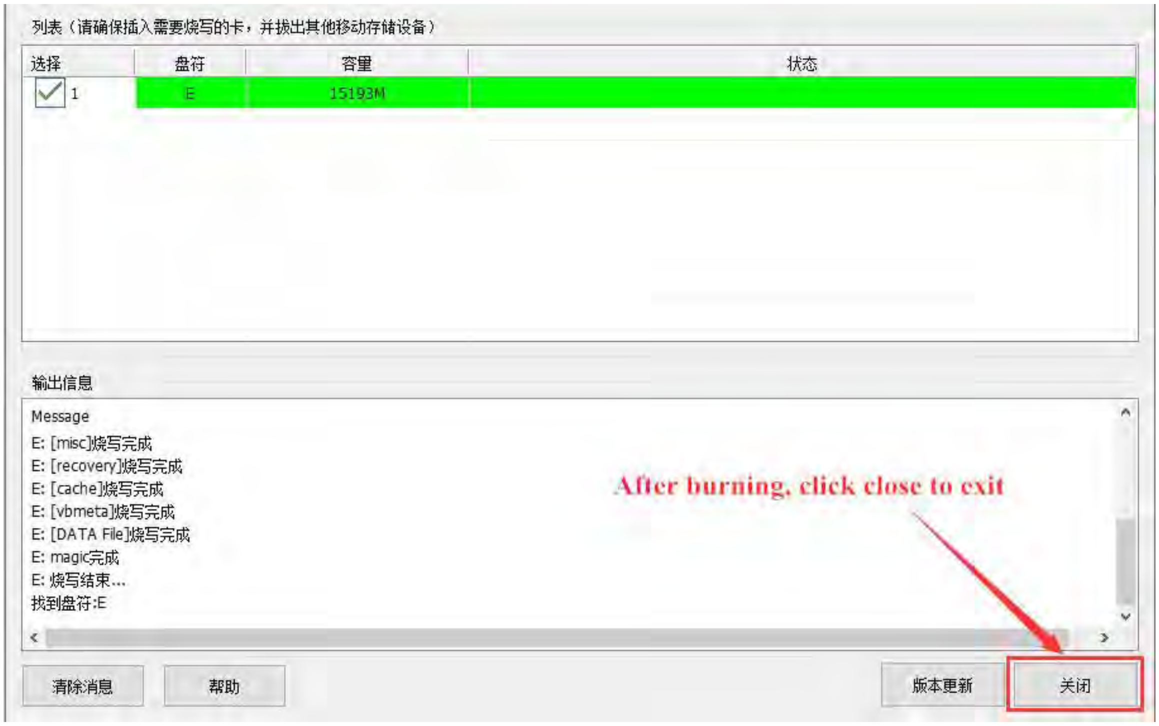
选择	盘符	容量	状态
✓ 1	E	15193M	24%

Show the flashing progress of the Android image

```
Message
E: 正在格式化卡....
E: 开始烧写...
E: [burn 1st part: boot0]烧写完成
E: [burn 2nd part: boot1]烧写完成
E: [MBR]烧写完成
E: [bootloader]烧写完成
E: [env]烧写完成
E: [boot]烧写完成
```

Display the output information of the Android image process

12) After burning, the display of PhoenixCard is as shown in the figure below. At this time, click the "**Close**" button to exit PhoenixCard, and then you can pull out the micro SD card from the computer and insert it into the development board to start



列表（请确保插入需要烧写的卡，并拔出其他移动存储设备）

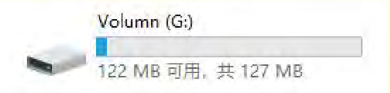
选择	盘符	容量	状态
☑ 1	E	15193M	

输出信息

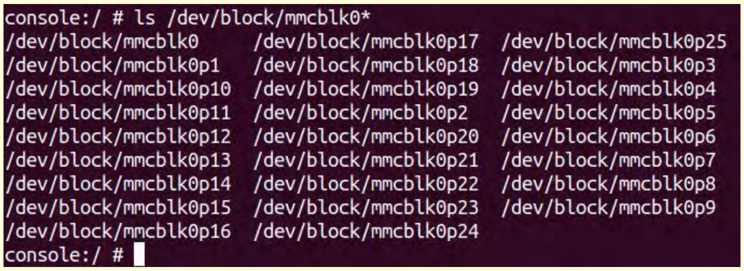
```
Message
E: [misc]烧写完成
E: [recovery]烧写完成
E: [cache]烧写完成
E: [vbmeta]烧写完成
E: [DATA File]烧写完成
E: magic完成
E: 烧写结束...
找到盘符:E
```

After burning, click close to exit

After burning the Android system, the micro SD card can only see a 128 MB partition in Windows, and the displayed partition is as shown in the figure below (some computers may pop up more than 20 disk partitions, but only the 128 MB partition can be opened. partition), please note that this is normal, please do not think that the micro SD card is burned out. The reason for this is that the Android system has a total of more than 20 partitions, but most of them cannot be recognized normally in the Windows system. At this point, please safely unplug the micro SD card and insert it into the development board to start it.

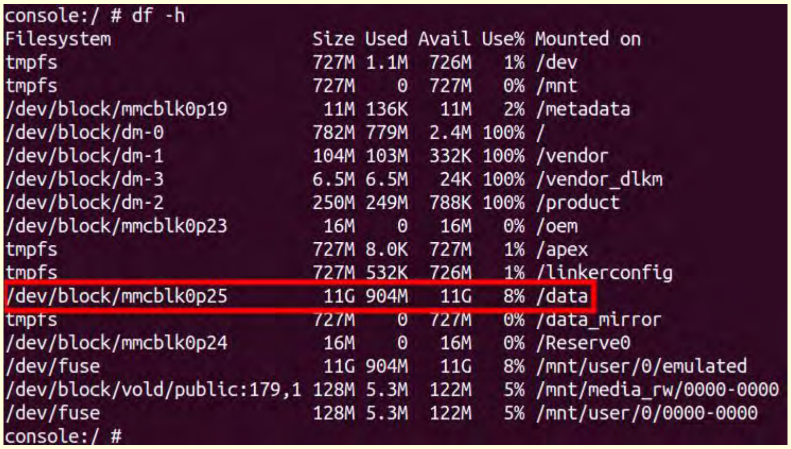


After the Android system starts, use the following command to see the twenty or so partitions in the micro SD card:



```
console:/ # ls /dev/block/mmcblk0*
/dev/block/mmcblk0      /dev/block/mmcblk0p17  /dev/block/mmcblk0p25
/dev/block/mmcblk0p1    /dev/block/mmcblk0p18  /dev/block/mmcblk0p3
/dev/block/mmcblk0p10   /dev/block/mmcblk0p19  /dev/block/mmcblk0p4
/dev/block/mmcblk0p11   /dev/block/mmcblk0p2   /dev/block/mmcblk0p5
/dev/block/mmcblk0p12   /dev/block/mmcblk0p20  /dev/block/mmcblk0p6
/dev/block/mmcblk0p13   /dev/block/mmcblk0p21  /dev/block/mmcblk0p7
/dev/block/mmcblk0p14   /dev/block/mmcblk0p22  /dev/block/mmcblk0p8
/dev/block/mmcblk0p15   /dev/block/mmcblk0p23  /dev/block/mmcblk0p9
/dev/block/mmcblk0p16   /dev/block/mmcblk0p24
console:/ #
```

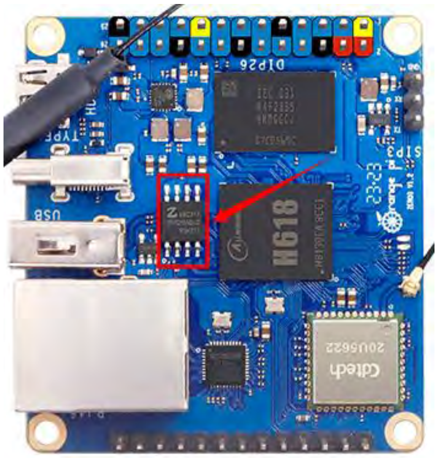
Use the df -h command to see that the 16GB micro SD card has about 11 GB of space available after burning the Android system (more than 20 partitions will not be mounted to the Android system, focus on these to the partition).



```
console:/ # df -h
Filesystem                    Size Used Avail Use% Mounted on
tmpfs                         727M 1.1M  726M   1% /dev
tmpfs                         727M    0  727M   0% /mnt
/dev/block/mmcblk0p19          11M 136K   11M   2% /metadata
/dev/block/dm-0               782M 779M  2.4M 100% /
/dev/block/dm-1               104M 103M  332K 100% /vendor
/dev/block/dm-3               6.5M 6.5M   24K 100% /vendor_dlkm
/dev/block/dm-2               250M 249M  788K 100% /product
/dev/block/mmcblk0p23          16M    0   16M   0% /oem
tmpfs                         727M 8.0K  727M   1% /apex
tmpfs                         727M 532K  726M   1% /linkerconfig
/dev/block/mmcblk0p25          11G 904M   11G   8% /data
tmpfs                         727M    0  727M   0% /data_mirror
/dev/block/mmcblk0p24          16M    0   16M   0% /Reserve0
/dev/fuse                      11G 904M   11G   8% /mnt/user/0/emulated
/dev/block/vold/public:179,1 128M 5.3M  122M   5% /mnt/media_rw/0000-0000
/dev/fuse                     128M 5.3M  122M   5% /mnt/user/0/0000-0000
console:/ #
```

2. 6. Instructions for using the micro linux system in the onboard SPI Flash

There is a 16MB SPI Flash on the development board, and its location is shown in the figure below:



There is a tiny linux system programmed into SPI Flash by default, which is mainly used to prove that the development board can start normally. When you get the development board, you don't need to burn the system into the micro SD card, you only need to connect the Type-C power supply to the development board to start the micro linux system in the SPI Flash. The main functions of this system are:

a) During the u-boot start-up phase, the red LED light will be lit, and after entering the kernel, the red LED light will be turned off and the green LED light will be set to flash;

b) If the development board is connected to an HDMI screen, after the system starts up, the command line interface of the micro-linux system can be seen on the HDMI screen;

c) If the development board is connected with a USB keyboard, some simple linux commands can be run on the command line, such as ls, cd, etc.

Due to the limited functions of the tiny linux system in SPI Flash, if you want to use all the functions of the development board normally, please burn the linux image or Android image to the micro SD card, and then use it.

2. 7. Start the OrangePi development board

1) Insert the micro SD card with the burned image into the micro SD card slot of the OrangePi development board

2) The development board has a Micro HDMI interface, and the development board can be connected to a TV or HDMI display through a Micro HDMI to HDMI cable

3) If you have purchased a 13pin expansion board, you can plug the 13pin expansion board into the 13pin interface of the development board

4) Connect the USB mouse and keyboard to control the OrangePi development board

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

6) Connect a high-quality power adapter with a 5V/2A (5V/3A is also available) USB Type C interface

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

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

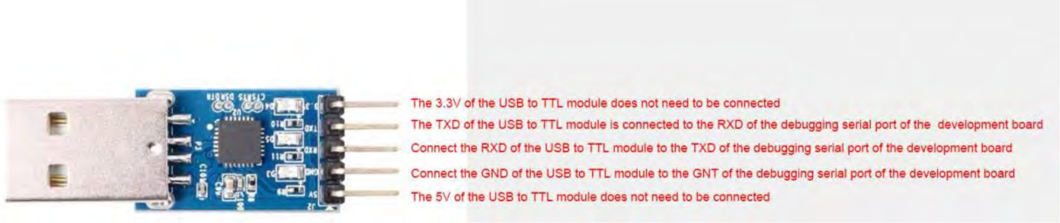
7) Then turn on the switch of the power adapter, if everything is normal, the HDMI monitor can see the system startup screen at this time

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

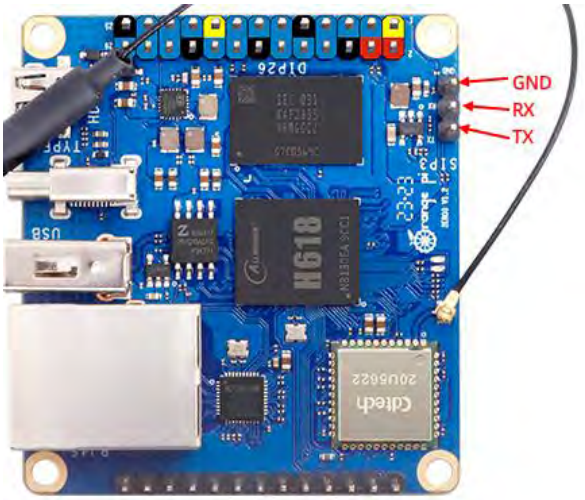
2. 8. How to use the debugging serial port

2. 8. 1. Connection instruction of debugging serial port

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


The 3.3V of the USB to TTL module does not need to be connected
The TXD of the USB to TTL module is connected to the RXD of the debugging serial port of the development board
Connect the RXD of the USB to TTL module to the TXD of the debugging serial port of the development board
Connect the GND of the USB to TTL module to the GNT of the debugging serial port of the development board
The 5V of the USB to TTL module does not need to be connected

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



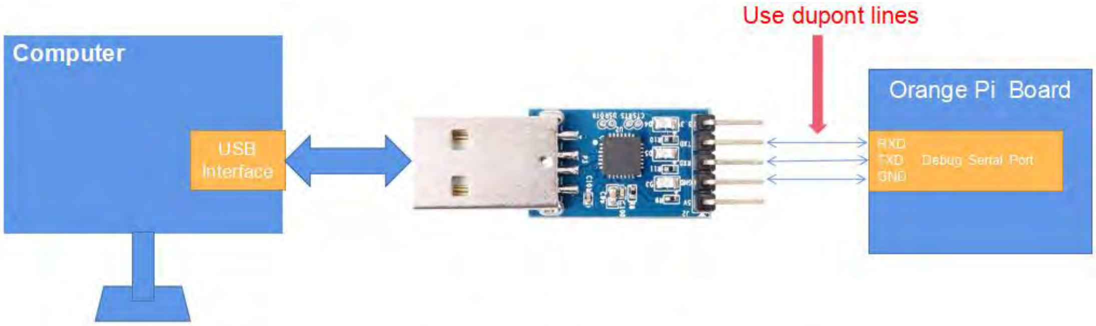
3) The GND, TX and RX pins of the USB to TTL module need to be connected to the debugging serial port of the development board through a DuPont line

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

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

c. Connect the TX of the USB **to TTL module to the RX of the development board**

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



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

The TX and RX of the serial port need to be cross-connected. If you don't want to carefully distinguish the order of TX and RX, you can connect the TX and RX of

the serial port casually first. If there is no output from the test serial port, then exchange the order of TX and RX. One order is right.

2. 8. 2. How to use the debugging serial port on the Ubuntu platform

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

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

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

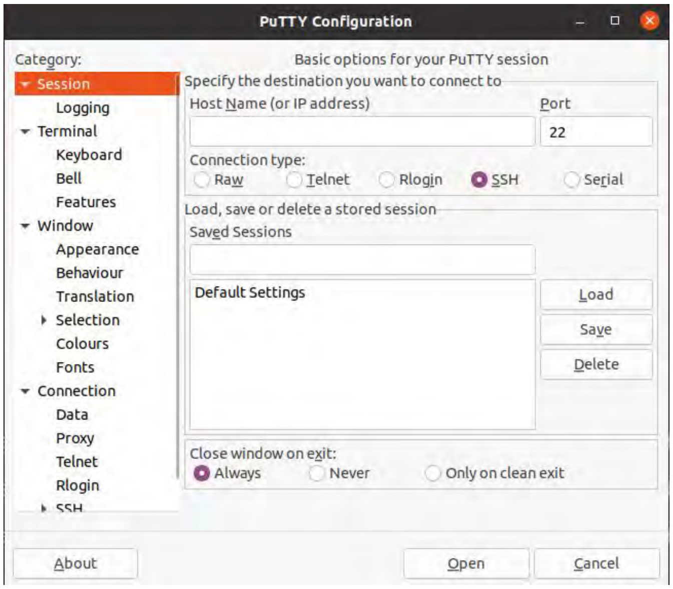
2) Then use the following command to install putty on Ubuntu PC

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

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

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

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



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address)

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

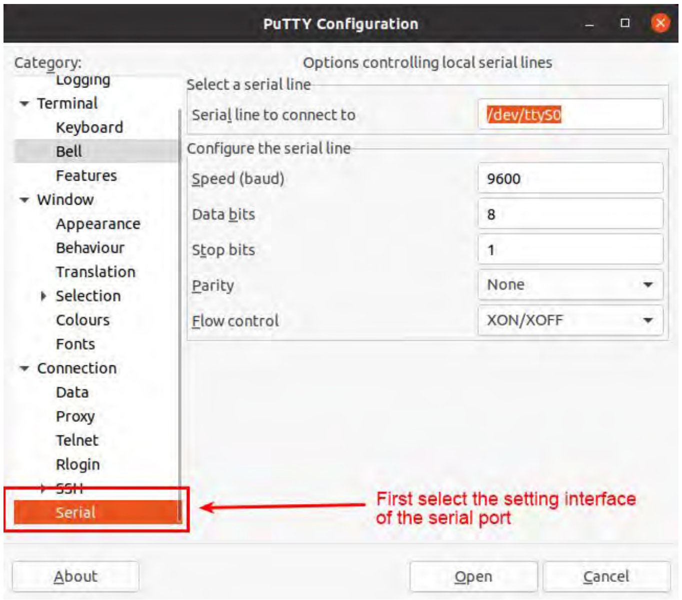
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit:
● Always ○ Never ○ Only on clean exit

5) First select the setting interface of the serial port



Options controlling local serial lines

Select a serial line

Serial line to connect to: /dev/ttyS0

Configure the serial line

Speed (baud): 9600

Data bits: 8

Stop bits: 1

Parity: None

Flow control: XON/XOFF

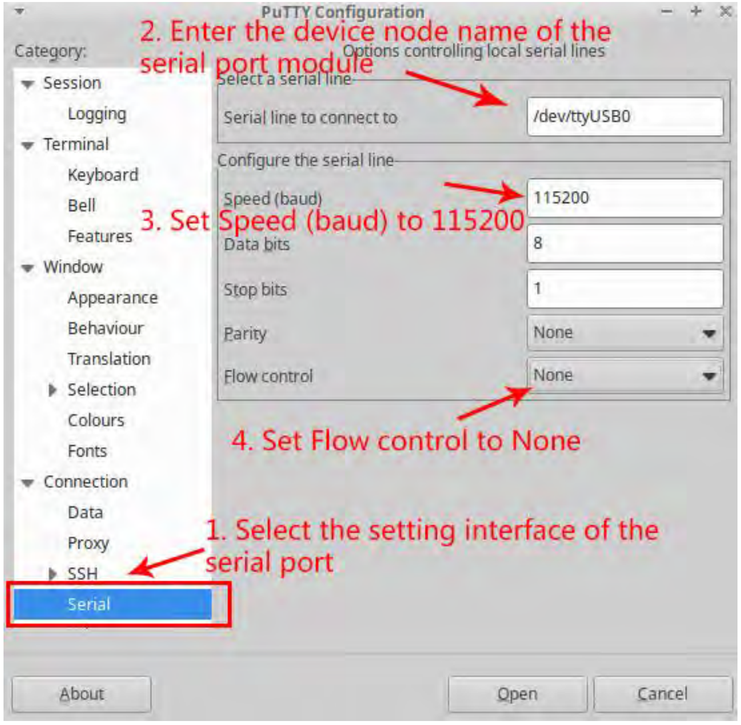
First select the setting interface of the serial port

6) Then set the parameters of the serial port

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

b. Set **Speed(baud)** to 115200 (the baud rate of the serial port)

c. Set **Flow control** as **None**



2. Enter the device node name of the serial port module

Options controlling local serial lines

Select a serial line

Serial line to connect to: /dev/ttyUSB0

Configure the serial line

Speed (baud): 115200

3. Set Speed (baud) to 115200

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

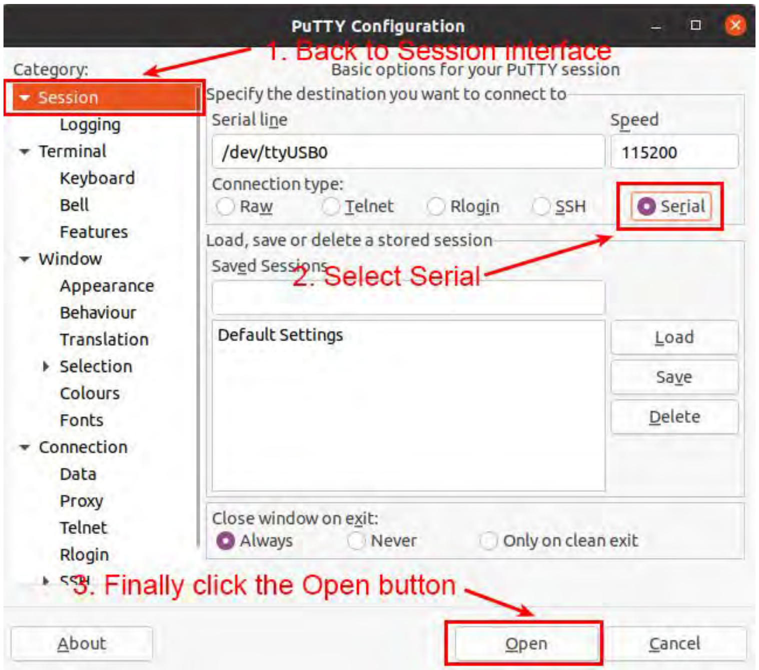
4. Set Flow control to None

1. Select the setting interface of the serial port

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

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

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



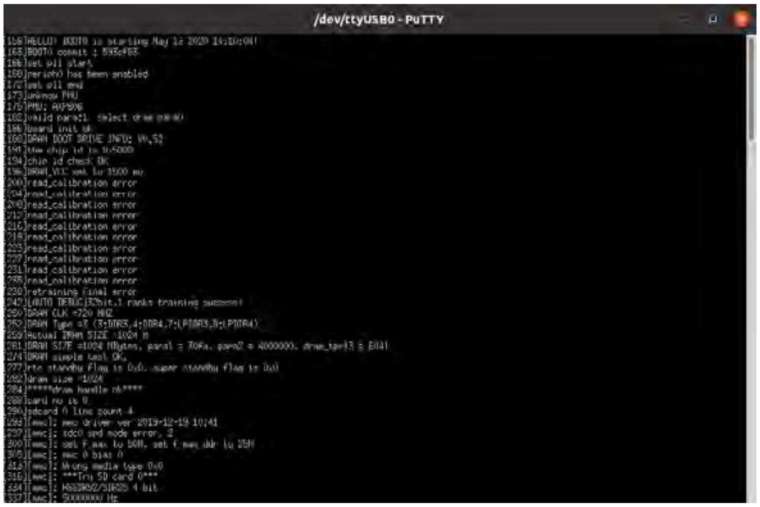
Serial line: /dev/ttyUSB0

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Close window on exit: ● Always ○ Never ○ Only on clean exit

8) Then start the development board, and you can see the Log information output by the system from the opened serial port terminal



```
[156]HELLO! BOOT0 is starting May 12 2020 14:10:04!
[163]BOOT0 commit : 593cf85
[166]set pll start
[169]periph0 has been enabled
[172]set pll end
[173]unknow PMU
[175]PMU: AXP806
[182]valid para:1  select dram para0
[186]board init ok
[188]DRAM BOOT DRIVE INFO: V0.52
[191]the chip id is 0x5000
[194]chip id check OK
[196]DRAM_VCC set to 1500 mv
[200]read_calibration error
[204]read_calibration error
[208]read_calibration error
[212]read_calibration error
[216]read_calibration error
[219]read_calibration error
[223]read_calibration error
[227]read_calibration error
[231]read_calibration error
[235]read_calibration error
[239]retraining final error
[247][AUTO DEBUG]32bit,1 ranks training success!
[250]DRAM CLK =720 MHZ
[252]DRAM Type =7 (3:DDR3,4:DDR4,7:LPDDR3,8:LPDDR4)
[259]Actual DRAM SIZE =1024 M
[261]DRAM SIZE =1024 MBytes, para1 = 30fa, para2 = 4000000, dram_tpr13 = 6041
[274]DRAM simple test OK.
[277]rtc standby flag is 0x0, super standby flag is 0x0
[282]dram size =1024
[284]*****dram handle ok****
[288]card no is 0
[290]sdcard 0 line count 4
[293][mmc]: mmc driver ver 2019-12-19 10:41
[297][mmc]: sdc0 spd mode error, 2
[300][mmc]: set f_max to 50M, set f_max_ddr to 25M
[305][mmc]: mmc 0 bias 0
[313][mmc]: Wrong media type 0x0
[316][mmc]: ***Try SD card 0***
[334][mmc]: HSSDR52/SDR25 4 bit
[337][mmc]: 50000000 Hz
```

2. 8. 3. How to use the debugging serial port on Windows platform

There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use

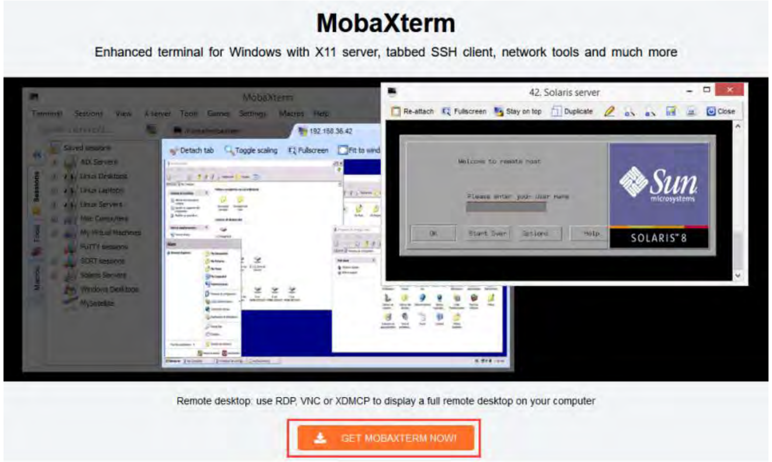
MobaXterm. This software has a free version and can be used without buying a serial number.

1) Download MobaXterm

   a. Download MobaXterm website as follows

https://mobaxterm.mobatek.net/

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



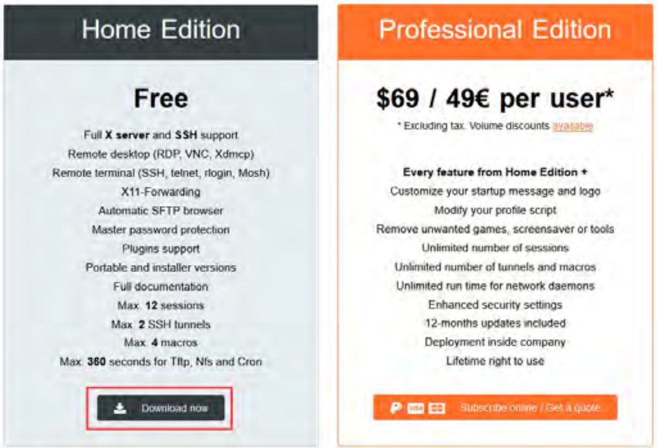
MobaXterm

Enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more

Remote desktop: use RDP, VNC or XDMCP to display a full remote desktop on your computer

GET MOBAXTERM NOW!

   c. Then choose to download the Home version



Home Edition

Free

Full **X server** and **SSH** support
Remote desktop (RDP, VNC, Xdmcp)
Remote terminal (SSH, telnet, rlogin, Mosh)
X11-Forwarding
Automatic SFTP browser
Master password protection
Plugins support
Portable and installer versions
Full documentation
Max. **12** sessions
Max. **2** SSH tunnels
Max. **4** macros
Max. **360** seconds for Tftp, Nfs and Cron

Download now

Professional Edition

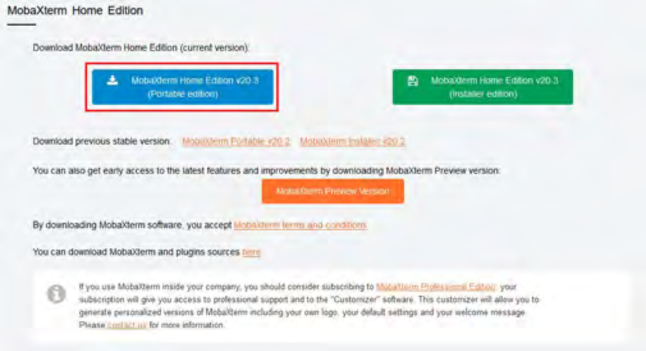
$69 / 49€ per user*

* Excluding tax. Volume discounts available

Every feature from Home Edition +
Customize your startup message and logo
Modify your profile script
Remove unwanted games, screensaver or tools
Unlimited number of sessions
Unlimited number of tunnels and macros
Unlimited run time for network daemons
Enhanced security settings
12-months updates included
Deployment inside company
Lifetime right to use

Subscribe online / Get a quote

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



MobaXterm Home Edition

Download MobaXterm Home Edition (current version):

MobaXterm Home Edition v20.3
(Portable edition)
MobaXterm Home Edition v20.3
(Installer edition)

Download previous stable version: MobaXterm Portable v20.2 MobaXterm Installer v20.2

You can also get early access to the latest features and improvements by downloading MobaXterm Preview version:

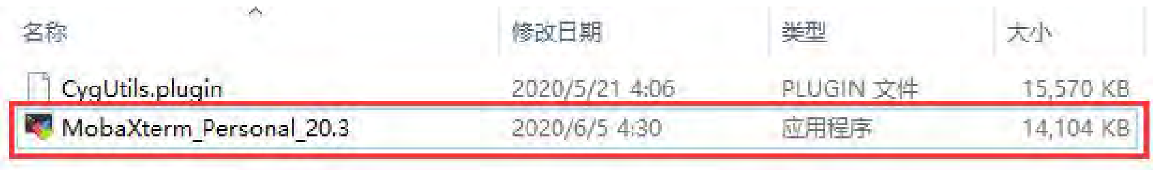
MobaXterm Preview Version

By downloading MobaXterm software, you accept MobaXterm terms and conditions

You can download MobaXterm and plugins sources here

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

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



名称	修改日期	类型	大小
CygUtils.plugin	2020/5/21 4:06	PLUGIN 文件	15,570 KB
MobaXterm_Personal_20.3	2020/6/5 4:30	应用程序	14,104 KB

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

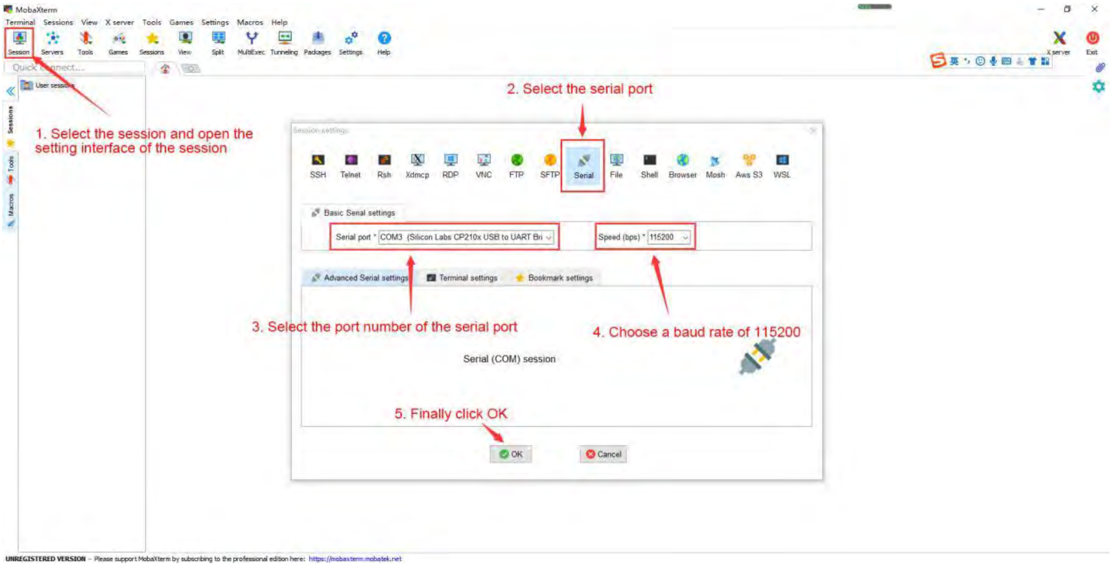
a. Open the session settings interface

b. Select the serial port type

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

d. Select the baud rate of the serial port as **115200**

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



1. Select the session and open the setting interface of the session

2. Select the serial port

Serial port * COM3 (Silicon Labs CP210x USB to UART Bri

Speed (bps) * 115200

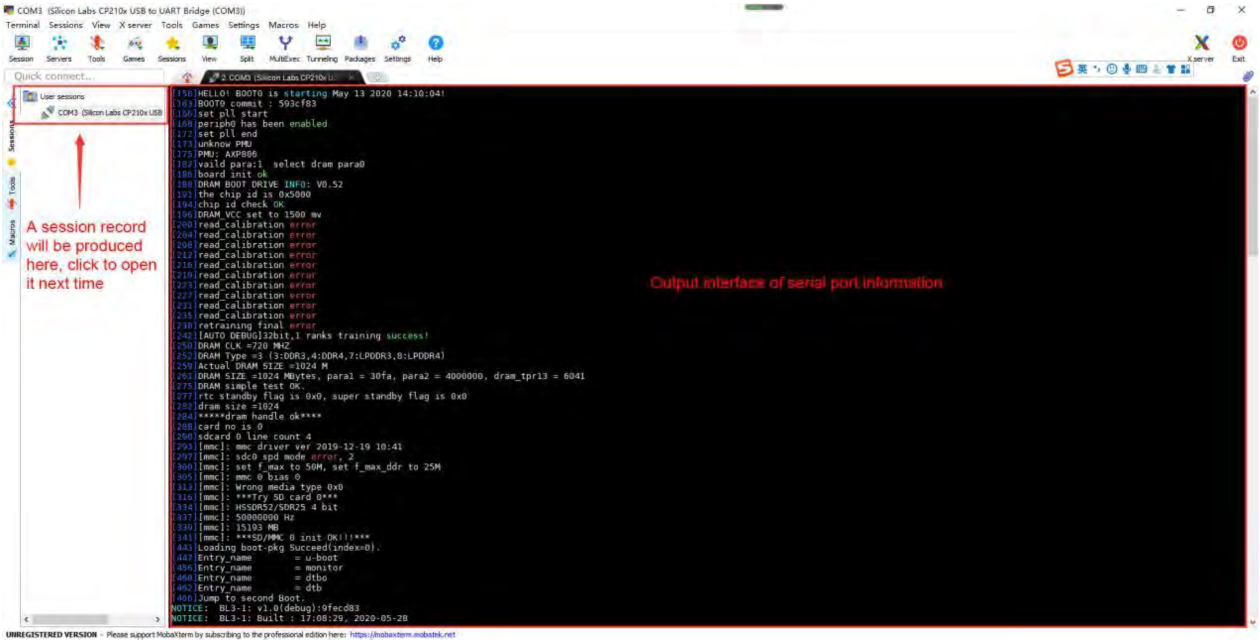
3. Select the port number of the serial port

4. Choose a baud rate of 115200

5. Finally click OK

MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
User sessions
Sessions Tools Macros
Session settings
SSH Telnet Rsh Xdmcp RDP VNC FTP SFTP Serial File Shell Browser Mosh Aws S3 WSL
Basic Serial settings
Advanced Serial settings Terminal settings Bookmark settings
Serial (COM) session
OK Cancel
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

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



A session record will be produced here, click to open it next time

Output interface of serial port information

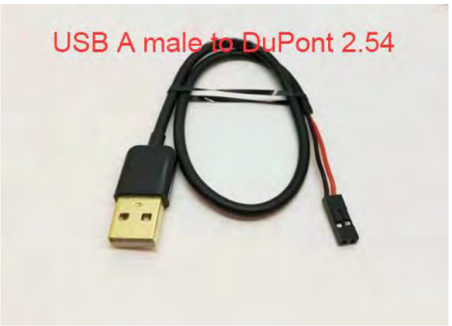
```
[158]HELLO! BOOT0 is starting May 13 2020 14:10:04!
[163]BOOT0 commit : 593cf83
[166]set pll start
[168]periph0 has been enabled
[172]set pll end
[173]unknow PMU
[175]PMU: AXP806
[182]vaild para:1  select dram para0
[186]board init ok
[188]DRAM BOOT DRIVE INFO: V0.52
[191]the chip id is 0x5000
[194]chip id check OK
[196]DRAM_VCC set to 1500 mv
[200]read_calibration error
[204]read_calibration error
[208]read_calibration error
[212]read_calibration error
[216]read_calibration error
[219]read_calibration error
[223]read_calibration error
[227]read_calibration error
[231]read_calibration error
[235]read_calibration error
[238]retraining final error
[242][AUTO DEBUG]32bit,1 ranks training success!
[250]DRAM CLK =720 MHZ
[252]DRAM Type =3 (3:DDR3,4:DDR4,7:LPDDR3,8:LPDDR4)
[259]Actual DRAM SIZE =1024 M
[261]DRAM SIZE =1024 MBytes, para1 = 30fa, para2 = 4000000, dram_tpr13 = 6041
[275]DRAM simple test OK.
[277]rtc standby flag is 0x0, super standby flag is 0x0
[282]dram size =1024
[284]*****dram handle ok****
[288]card no is 0
[290]sdcard 0 line count 4
[293][mmc]: mmc driver ver 2019-12-19 10:41
[297][mmc]: sdc0 spd mode error, 2
[300][mmc]: set f_max to 50M, set f_max_ddr to 25M
[305][mmc]: mmc 0 bias 0
[313][mmc]: Wrong media type 0x0
[316][mmc]: ***Try SD card 0***
[334][mmc]: HSSDR52/SDR25 4 bit
[337][mmc]: 50000000 Hz
[339][mmc]: 15193 MB
[341][mmc]: ***SD/MMC 0 init OK!!!***
[443]Loading boot-pkg Succeed(index=0).
[447]Entry_name        = u-boot
[456]Entry_name        = monitor
[460]Entry_name        = dtbo
[462]Entry_name        = dtb
[466]Jump to second Boot.
NOTICE:  BL3-1: v1.0(debug):9fecd83
NOTICE:  BL3-1: Built : 17:08:29, 2020-05-28
```

COM3 (Silicon Labs CP210x USB to UART Bridge (COM3))
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
2. COM3 (Silicon Labs CP210x U
User sessions
COM3 (Silicon Labs CP210x USB
Sessions Tools Macros
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

2. 9. Instructions for using the 5v pin in the 26pin or 13pin interface of the development board to supply power

The power supply method we recommend for the development board is to use the 5V/2A or 5V/3A Type C interface power cord to plug into the Type C power interface of the development board for power supply. If you need to use the 5V pin in the 26pin or 13pin interface to power the development board, please make sure that the power cable used can meet the power supply requirements of the development board. If the use is unstable, please switch back to the Type C power supply.

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



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

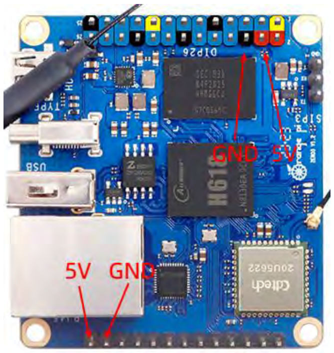
2) Use the 5V pin in the 26pin or 13pin interface to supply power to the development board. The connection method of the power line is as follows

a. The USB A port of the power cord shown in the above figure needs to be plugged into the 5V/2A or 5V/3A power adapter connector (**it is not recommended to plug into the USB port of the computer for power supply. If there are too many peripherals connected to the development board, use will be unstable**)

b. The red DuPont wire needs to be plugged into the 5V pin of the 26pin or 13pin interface of the development board

c. The black Dupont wire needs to be inserted into the GND pin of the 26pin or 13pin interface

d. The positions of the 5V pin and GND pin of the 26pin and 13pin interfaces in the development board are shown in the figure below**, remember not to reverse the connection.**

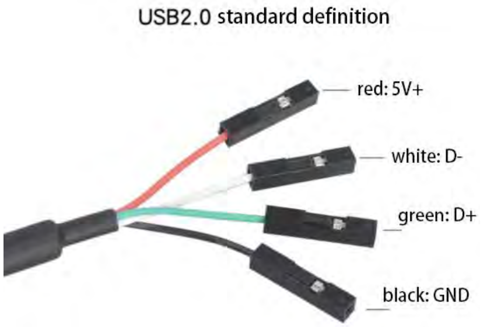


2. 10. The method of using the 13pin interface of the development board to expand the USB interface

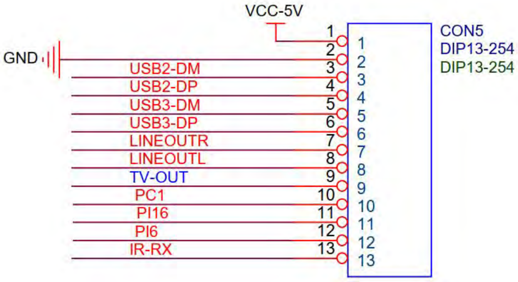
1) If you have purchased a 13pin expansion board for Orange Pi, insert the expansion board into the 13pin interface of the development board to expand 2 USB interfaces

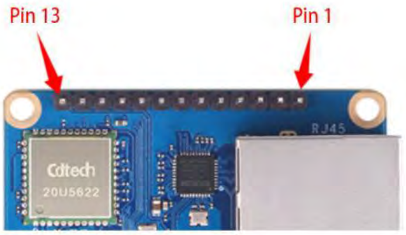
2) If there is no 13pin expansion board, you can use a 4pin 2.54mm DuPont to USB2.0 female cable to expand the USB interface. The specific method is as follows:

a. First, you need to prepare a 4pin 2.54mm Dupont to USB2.0 female cable (this cable can be bought on Amazon, please search and buy by yourself), as shown in the figure below:

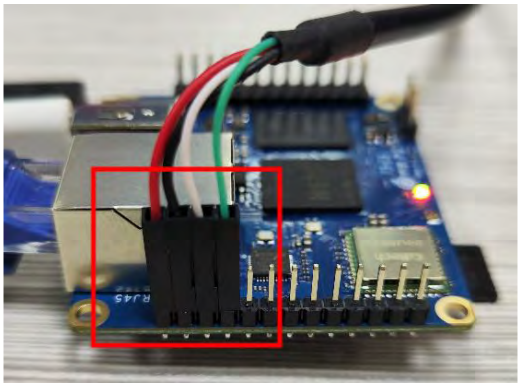


b. The schematic diagram of the 13pin interface is shown below

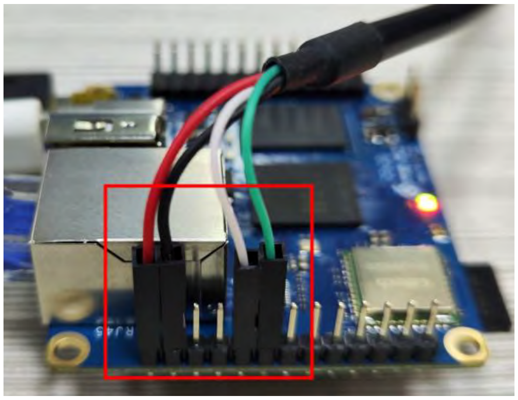




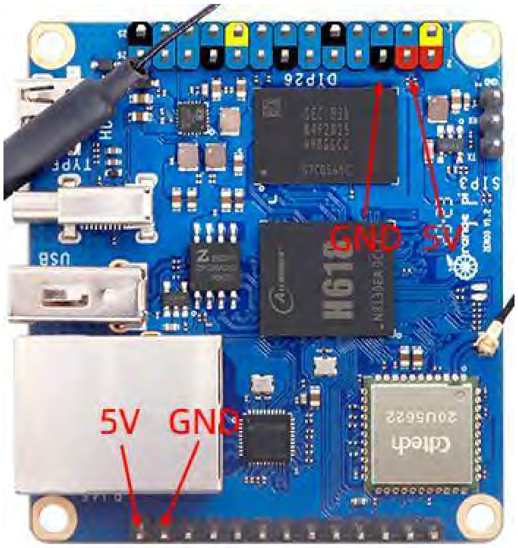
c. The wiring of USB2 is as follows



d. The wiring of USB3 is as follows



e. If you need to connect two USB devices to the 13pin interface at the same time, you will find that the 5V and GND pins on the 13pin interface are not enough. At this time, one of the USB devices can use the 5V and GND pins in the 26pin interface. The location is as shown in the figure below Shown:

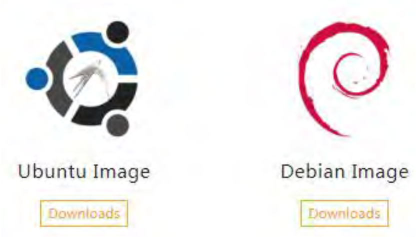


3. Instructions for use of Debian/Ubuntu Server and Xfce desktop system

3. 1. Supported linux image types and kernel versions

Linux image type	kernel version	server version	desktop version
Ubuntu 20.04 - Focal	Linux5.4	Support	Support
Ubuntu 22.04 - Focal	Linux5.4	Support	Support
Debian 11 - Bullseye	Linux5.4	Support	Support
Ubuntu 22.04 - Jammy	Linux6.1	Support	Support
Debian 11 - Bullseye	Linux6.1	Support	Support
Debian 12 - Bookworm	Linux6.1	Support	Support

After entering the download page of the corresponding development board **on the OrangePi data download page**, you can see the following download options. In the description below, **the Ubuntu image and the Debian image are generally referred to as the Linux image**.



The naming rules of Linux images are as follows:

Development board model_version number_Linux distribution type_release code name_server or desktop_kernel version

a. **The model of the development board**: Both are **orangepizero3**. 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, **1.x.x**, this version number will increase with the update of the image function, and the last number of the version number of the Linux image on 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 between the two systems in terms of usage. However, there are still some differences in the default configuration of some software and the use of commands. In addition, both Ubuntu and Debian have their own software warehouses supported by maintenance, and there are also some differences in the supported and installable software packages. These need to be experienced in person to have a deeper understanding. For more details, you can refer to the official documentation provided by Ubuntu and Debian.

d. **Release code name**: Used to distinguish between different versions of a specific Linux distribution such as Ubuntu or Debian. Among them, both **focal** and **jammy** are Ubuntu distributions, focal means Ubuntu20.04, and jammy means Ubuntu22.04. The biggest difference between different versions is that the software in the software warehouse maintained by the new version of Ubuntu system is much better than that of the old version of Ubuntu system. The ones in it should be new, such as Python and GCC compilation tool chains. **bullseye** is the specific version code of Debian, bullseye means Debian11, and bookworm means Debian12.

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. The storage space and resources occupied by the image are relatively small, and the command line is mainly used to operate and control the system. If it is **desktop_xfce**, it means that the system is installed with the XFCE desktop environment by default. The storage space and resources occupied by the image are relatively large. You can connect the monitor, mouse and keyboard to operate the 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 **linux5.4** and **linux6.1.**

3. 2. Linux kernel driver adaptation

Function	Linux5.4	Linux6.1
HDMI video	OK	OK
HDMI audio	OK	OK
USB2.0 x 3	OK	OK
micro SD card start	OK	OK
Gigabit Ethernet	OK	OK
Infrared receiver	OK	OK
WIFI	OK	OK
Bluetooth	OK	OK
headphone audio	OK	OK
usb camera	OK	OK
LED light	OK	OK
26pin GPIO	OK	OK
26pin I2C	OK	OK
26pin SPI1	OK	OK
26pin UART	OK	OK
PWM	OK	OK
Temperature Sensor	OK	OK
hardware watchdog	OK	OK
Mali GPU	NO	NO

Video codec	**NO**	**NO**
TV-OUT	**NO**	**NO**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3. 4. Linux system login instructions

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

Account	password
root	orangepi
orangepi	orangepi

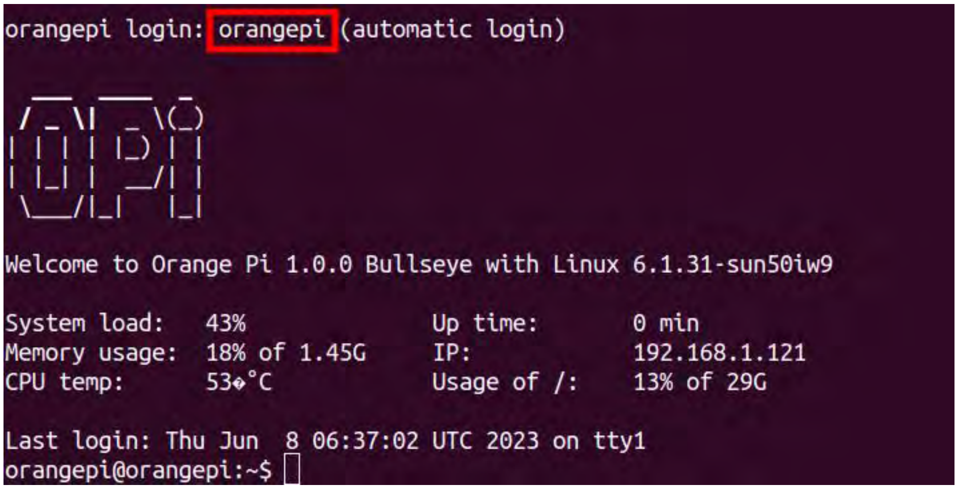
Note that when entering the password, the specific content of the entered password will not be displayed on the screen, please do not think that there is any fault, just press Enter after inputting.

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

other reasons.

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

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



```
orangepi login: orangepi (automatic login)

  ___  ____  _
 / _ \|  _ \(_)
| | | | |_) | |
| |_| |  __/| |
 \___/|_|   |_|

Welcome to Orange Pi 1.0.0 Bullseye with Linux 6.1.31-sun50iw9

System load:   43%               Up time:       0 min
Memory usage:  18% of 1.45G      IP:            192.168.1.121
CPU temp:      53�°C             Usage of /:    13% of 29G

Last login: Thu Jun  8 06:37:02 UTC 2023 on tty1
orangepi@orangepi:~$ 
```

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

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

3) Use the following command to disable automatic login terminal

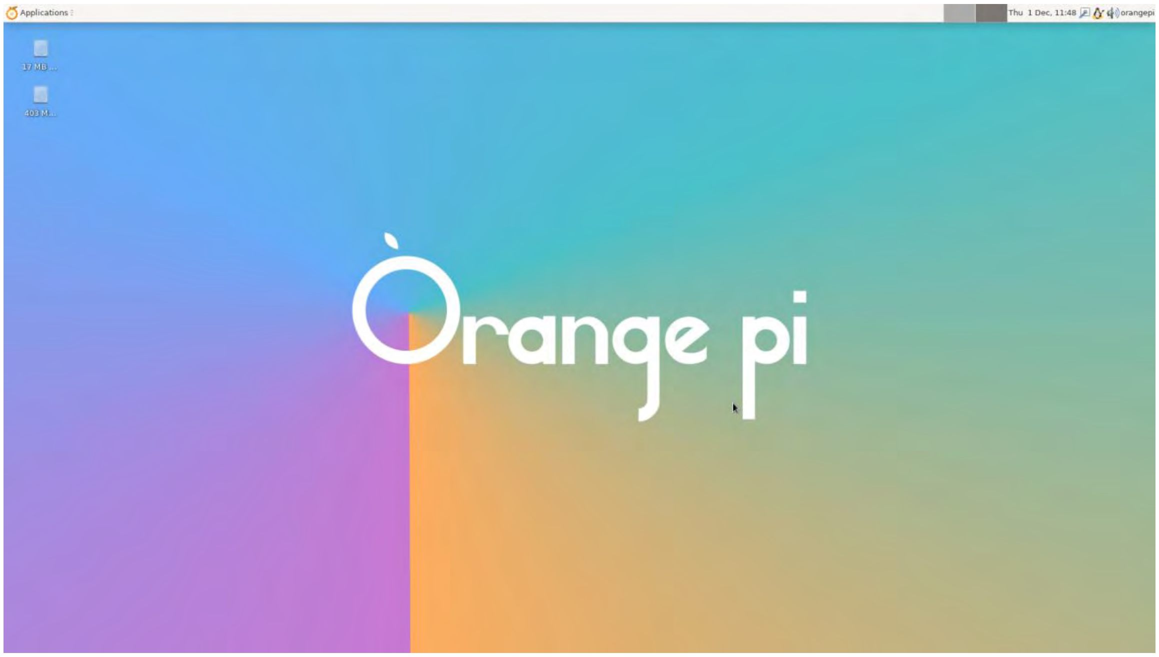
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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

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

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

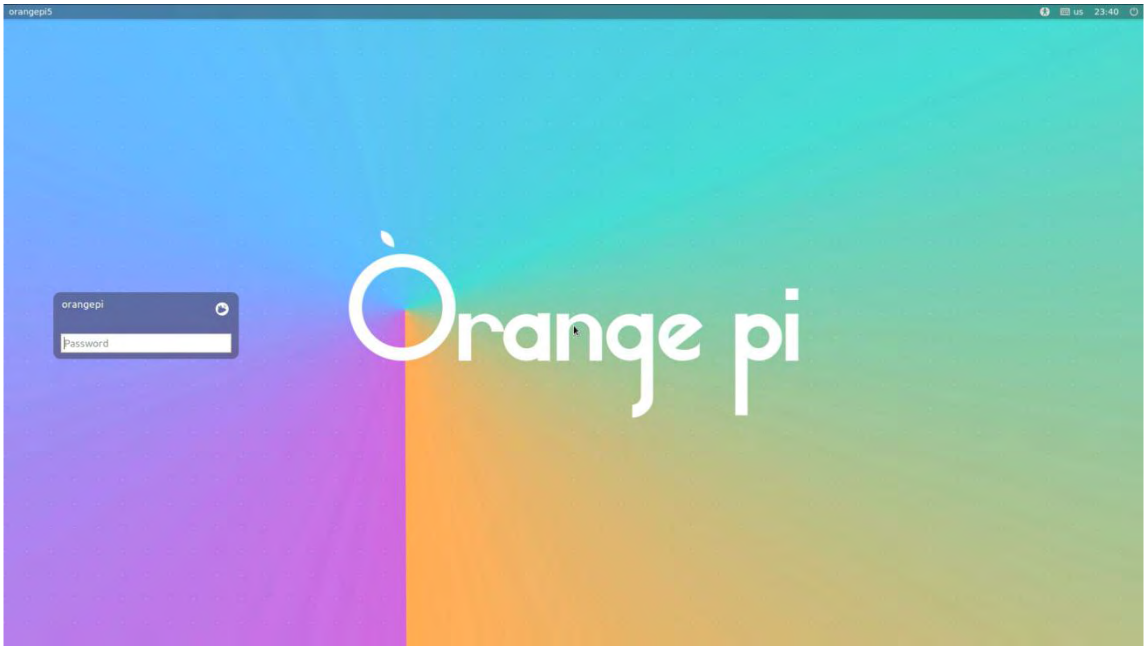
1) After the desktop system starts, it will automatically log in to the desktop without entering a password



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

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

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

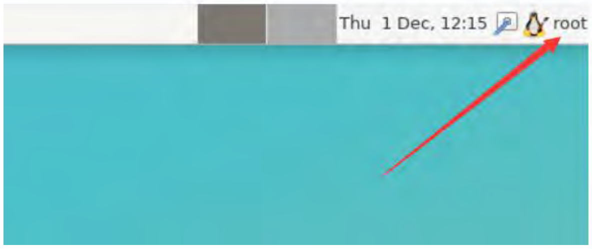


3. 4. 4. The setting method of root user automatic login in Linux desktop version system

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

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

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



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

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

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

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

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

1) First enter the following command on the command line, **Please remember to add sudo permission**

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

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

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

3) The command to reopen the desktop is as follows, **Please remember to add sudo permission**

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

3. 5. Onboard LED Light Test Instructions

1) There are two LED lights on the development board, one is green and the other is red. When the system starts, the default display of the LED lights is as follows:

	Green light	Red light
u-boot startup phase	off	on
The kernel boots into the system	flashing	off
GPIO interface	PC13	PC12

The two LED lights on the development board are controlled by software.

When you get the development board, you may find that even if the micro SD card with the system programmed is not inserted into the development board, the two LED lights will be on after the development board is connected to the power supply. This is because the 16MB on the development board SPI Flash will burn a miniature linux system by default when leaving the factory. This system will light up the red light during the u-boot startup phase. After entering the kernel, it will turn off the red light and set the green light to flash.

If the linux system in the SPI Flash is cleared, then the two LED lights on the development board will not light up after the power is turned on without inserting the micro SD card with the system programmed

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

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

a. First enter the setting directory of the green light

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

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

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

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

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

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

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

3) The method of setting the red light on/off and flashing is as follows:

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

a. First enter the setting directory of the red light

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

b. The command to set the red light to be on is as follows

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

c. The command to set the flashing red light is as follows

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

d. The command to set the red light to stop flashing is as follows

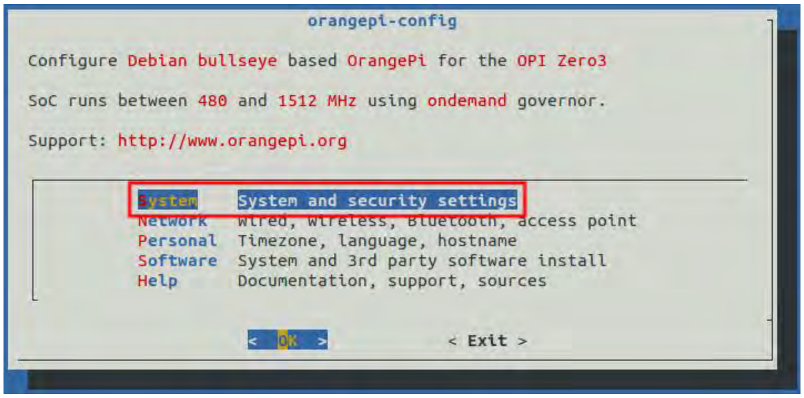
```
root@orangepi:/sys/class/leds/red_led# echo none > trigger
```

4) If you do not need the LED light to flash after booting, you can use the following method to turn off the green light flashing

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

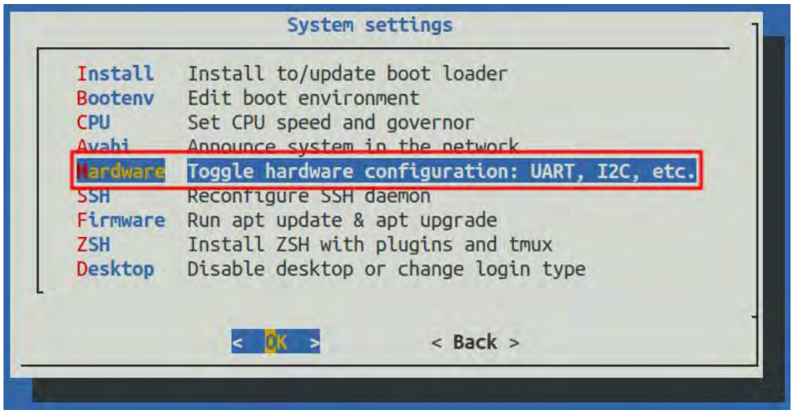
```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



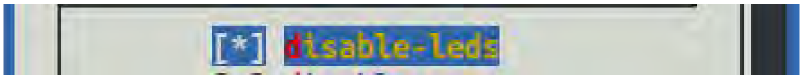
```
orangepi-config

Configure Debian bullseye based OrangePi for the OPI Zero3

SoC runs between 480 and 1512 MHz using ondemand governor.

Support: http://www.orangepi.org

System      System and security settings
Network     Wired, wireless, Bluetooth, access point
Personal    Timezone, language, hostname
Software    System and 3rd party software install
Help        Documentation, support, sources

<  OK  >                <  Exit  >
```

c. Then select **Hardware**



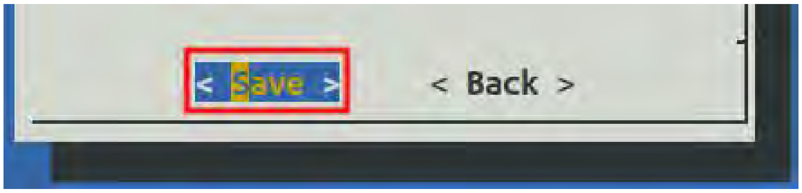
```
System settings

Install    Install to/update boot loader
Bootenv    Edit boot environment
CPU        Set CPU speed and governor
Avahi      Announce system in the network
Hardware   Toggle hardware configuration: UART, I2C, etc.
SSH        Reconfigure SSH daemon
Firmware   Run apt update & apt upgrade
ZSH        Install ZSH with plugins and tmux
Desktop    Disable desktop or change login type

<  OK  >                <  Back  >
```

d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select **disable-leds**

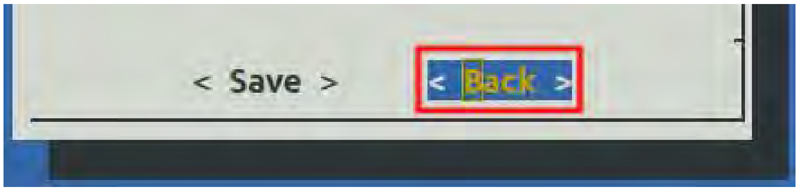


[*] disable-leds

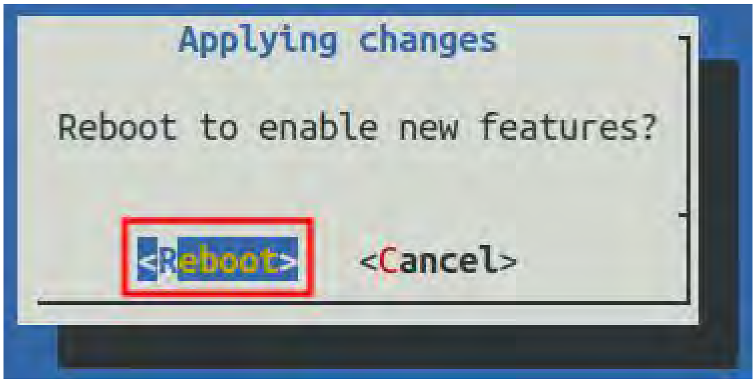
e. Then selec **<Save>**



f. Then select **<Back>**



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



Applying changes

Reboot to enable new features?

h. After restarting and entering the system, you can see that the two LED lights on the development board will not light up

3. 6. Instructions for operating the rootfs partition capacity of the linux system in the micro SD card

3. 6. 1. The first startup will automatically expand the capacity of the rootfs partition in the micro SD card

1) After burning the Linux image of the development board into the micro SD card, you can check the capacity usage of the micro SD card on the **Ubuntu computer**, the steps are as follows:

Note that this step does 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 micro SD card after burning the Linux image on the micro SD card.

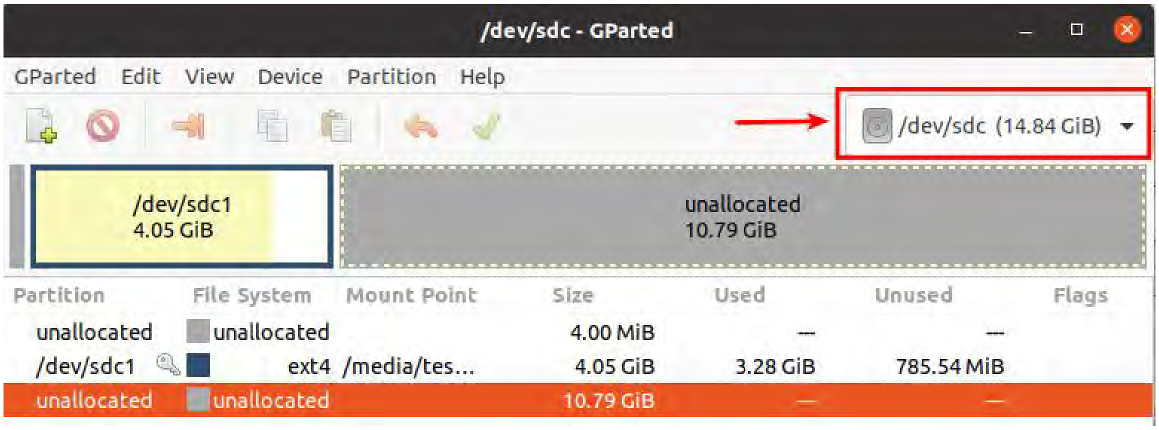
a. First install the gparted software on the Ubuntu computer

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

b. Then open gparted

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

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



Partition	File System	Mount Point	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	—	—	
/dev/sdc1	ext4	/media/tes...	4.05 GiB	3.28 GiB	785.54 MiB	
unallocated	unallocated		10.79 GiB	—	—	

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

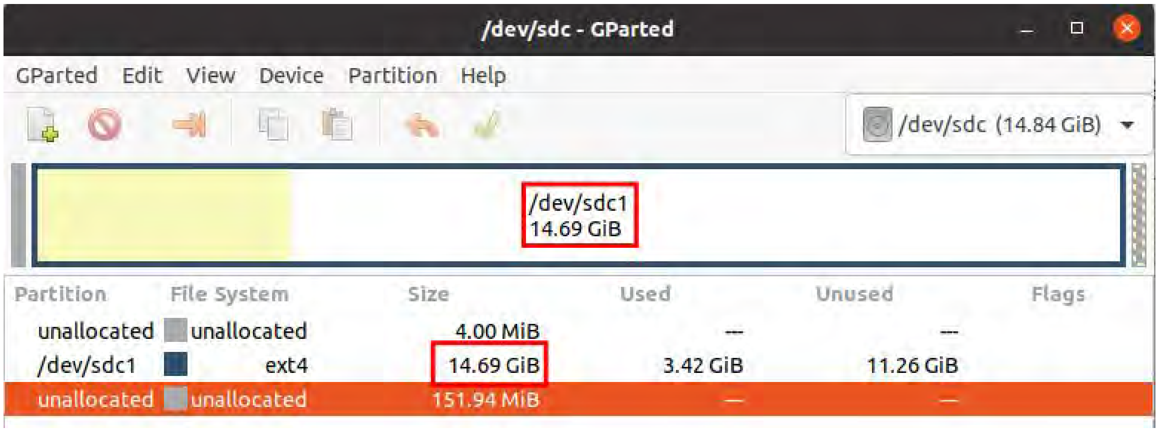
2) Then you can insert the micro SD card that has burned the Linux system into the development board to start. When the micro SD card starts the Linux system for the first time, it will call the **orangepi-resize-filesystem** script automatically through the systemd service **orangepi-resize-filesystem.service**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

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



Partition	File System	Size	Used	Unused	Flags
unallocated	unallocated	4.00 MiB	---	---	
/dev/sdc1	ext4	14.69 GiB	3.42 GiB	11.26 GiB	
unallocated	unallocated	151.94 MiB	---	---	

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 the kernel image, so there is no problem of expanding the BOOT partition.

3. 6. 2. The method of prohibiting automatic expansion of the capacity of the rootfs partition in the micro SD card

1) First burn the linux image of the development board to the micro SD card on the **Ubuntu computer** (Windows not available), **and then re-plug and insert the micro SD card**

2) Then the Ubuntu computer will automatically mount the partition of the micro SD card. If the automatic mounting is normal, use the ls command to see the following output

```
test@test:~$ ls /media/test/opi_root/
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 on the Ubuntu computer

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

```
root@test:~#
```

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

```
root@test:~# cd /media/test/opi_root/
root@test:/media/test/opi_root/# cd root
root@test:/media/test/opi_root/root# touch .no_rootfs_resize
root@test:/media/test/opi_root/root# ls .no_rootfs*
.no_rootfs_resize
```

5) Then you can uninstall the micro SD card, and then pull out the micro SD 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 not be automatically expanded.

6) After entering the Linux system after prohibiting automatic expansion of rootfs, 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 micro SD card, indicating that the automatic expansion of rootfs is prohibited.

```
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 micro SD 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:~# sudo 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 micro SD 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. The method of manually expanding the capacity of the rootfs partition in the micro SD card

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

1) First burn the linux image of the development board to the micro SD card on the **Ubuntu computer** (Windows not available), **and then re-plug and insert the micro SD card**

2) Then the Ubuntu computer will automatically mount the partition of the micro SD card. If the automatic mounting is normal, use the ls command to see the following output

```
test@test:~$ ls /media/test/opi_root/
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 on the Ubuntu computer

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

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

```
root@test:~# cd /media/test/opi_root/
root@test:/media/test/opi_root/# cd root
root@test:/media/test/opi_root/root# touch .no_rootfs_resize
root@test:/media/test/opi_root/root# ls .no_rootfs*
.no_rootfs_resize
```

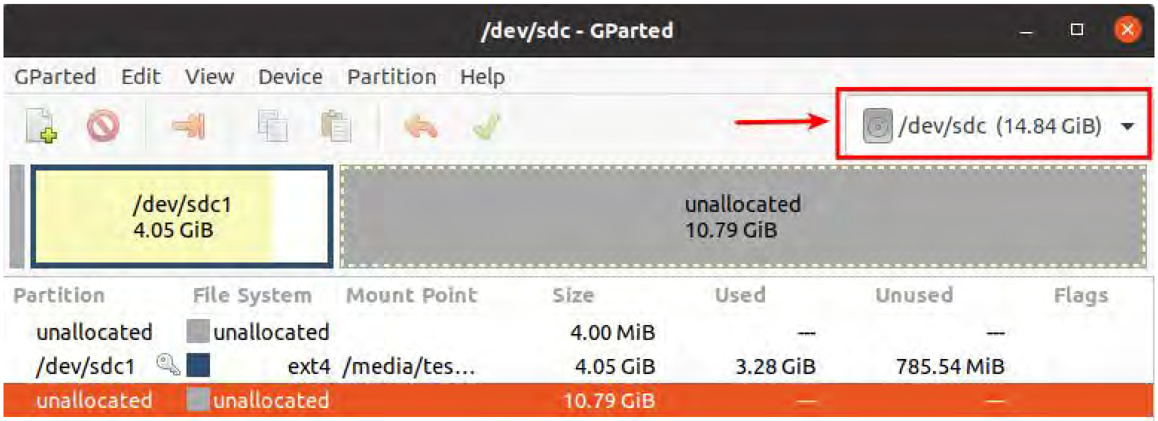
5) Then install the gparted software on the Ubuntu computer

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

6) Then open gparted

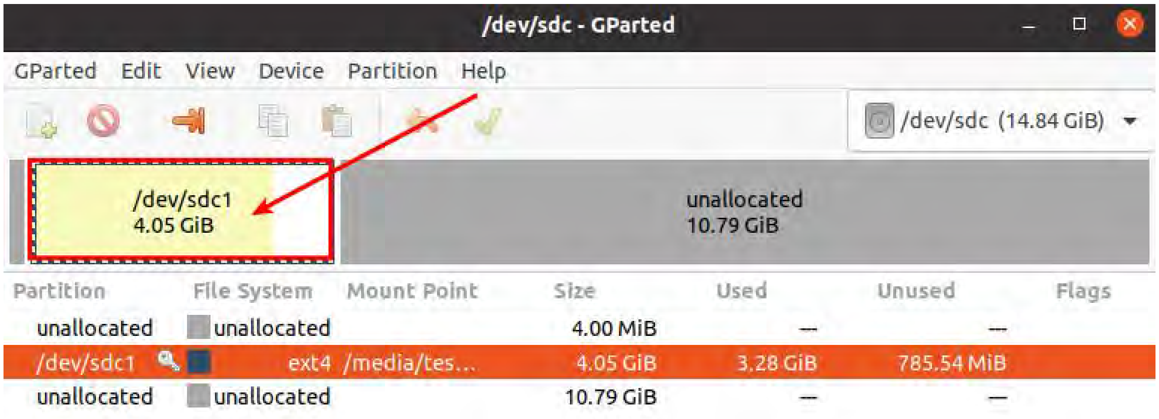
```
test@test:~$ sudo gparted
```

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



Partition	File System	Mount Point	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	—	—	
/dev/sdc1	ext4	/media/tes...	4.05 GiB	3.28 GiB	785.54 MiB	
unallocated	unallocated		10.79 GiB	—	—	

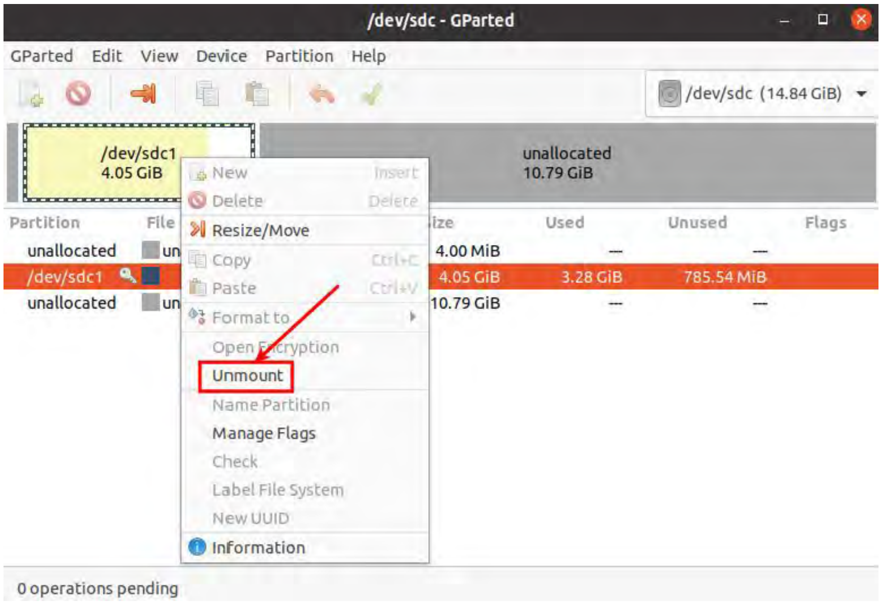
8) Then select the rootfs partition (/dev/sdc1)



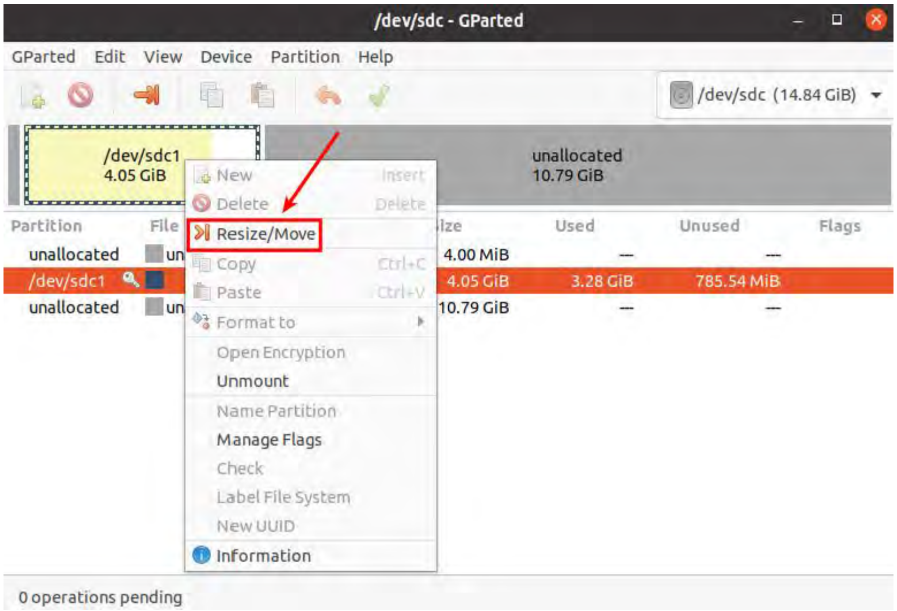
Partition	File System	Mount Point	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	—	—	
/dev/sdc1	ext4	/media/tes...	4.05 GiB	3.28 GiB	785.54 MiB	
unallocated	unallocated		10.79 GiB	—	—	

9) Click the right mouse button again to see the operation options shown in the figure below. If the micro SD card has been mounted, first you need to Umount the rootfs

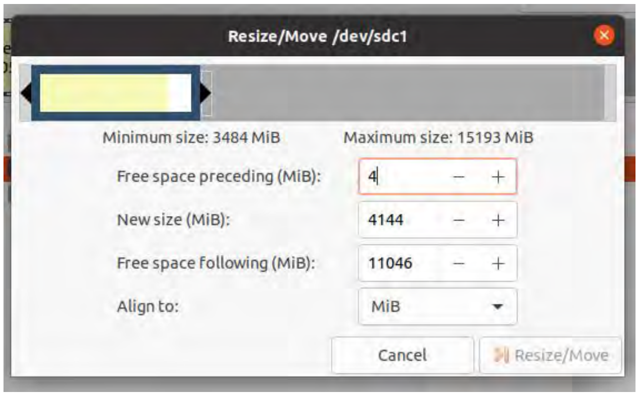
partition of the micro SD card



10) Then select the rootfs partition again, click the right mouse button, 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



Resize/Move /dev/sdc1

Minimum size: 3484 MiB Maximum size: 15193 MiB

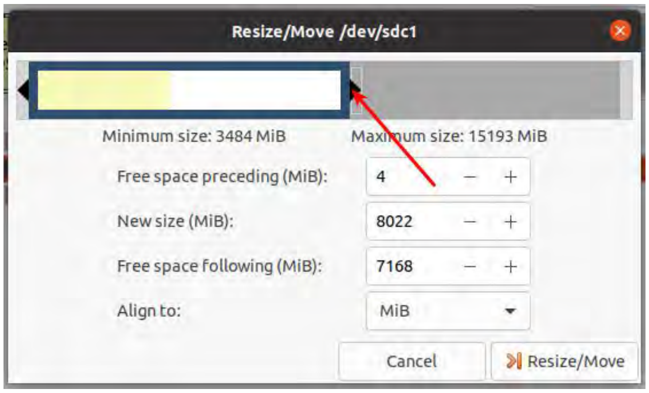
Free space preceding (MiB): 4

New size (MiB): 4144

Free space following (MiB): 11046

Align to: MiB

12) Then you can directly drag the location 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 size(MiB)**



Resize/Move /dev/sdc1

Minimum size: 3484 MiB Maximum size: 15193 MiB

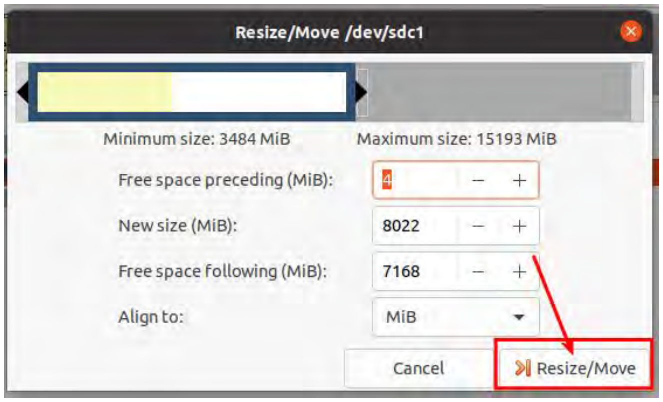
Free space preceding (MiB): 4

New size (MiB): 8022

Free space following (MiB): 7168

Align to: MiB

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



Resize/Move /dev/sdc1

Minimum size: 3484 MiB Maximum size: 15193 MiB

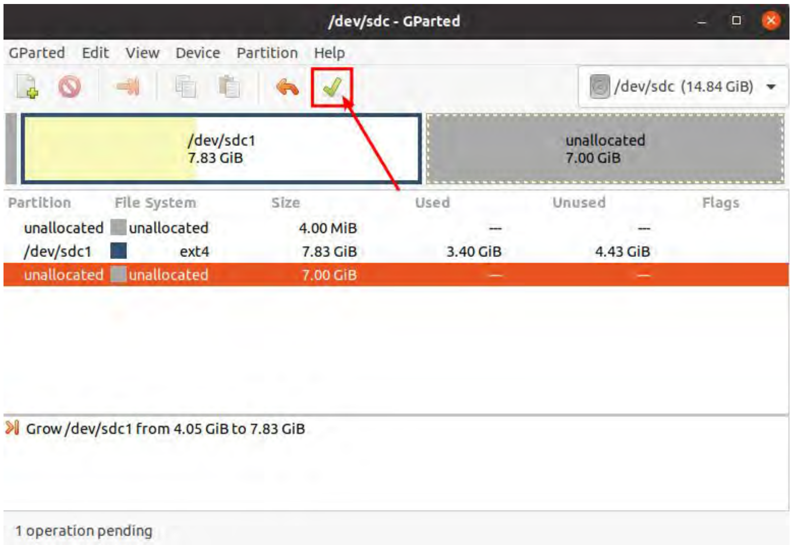
Free space preceding (MiB): 4

New size (MiB): 8022

Free space following (MiB): 7168

Align to: MiB

14) After confirming that it is correct, click the green button shown in the figure below √

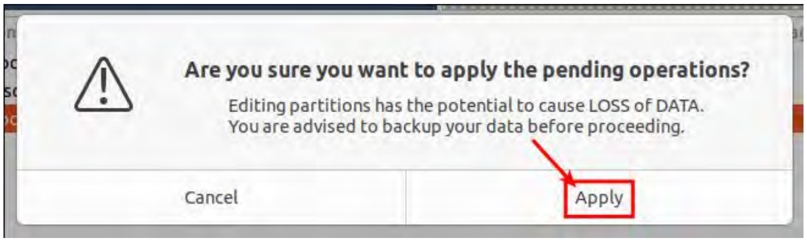


Partition	File System	Size	Used	Unused	Flags
unallocated	unallocated	4.00 MiB	—	—	
/dev/sdc1	ext4	7.83 GiB	3.40 GiB	4.43 GiB	
unallocated	unallocated	7.00 GiB	—	—	

Grow /dev/sdc1 from 4.05 GiB to 7.83 GiB

1 operation pending

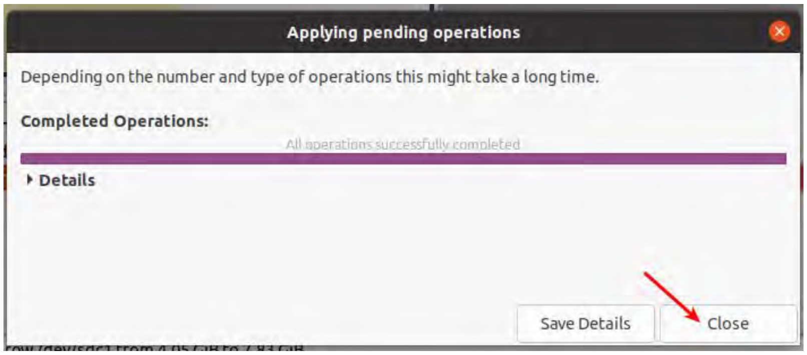
15) Then select **Apply**, it will officially start to expand the capacity of the rootfs partition



Are you sure you want to apply the pending operations?

Editing partitions has the potential to cause LOSS of DATA.
You are advised to backup your data before proceeding.

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



Applying pending operations

Depending on the number and type of operations this might take a long time.

Completed Operations:

All operations successfully completed

17) Then you can pull out the micro SD card, insert it into the development board and start it up. After entering the Linux system of the development board, if you can see that the size of the rootfs partition is the same as the size set before, it means manual 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. How to reduce the capacity of the rootfs partition in the micro SD card

After configuring the application program or other development environment in the Linux system of the micro SD card, if you want to back up the Linux system in the micro SD 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 micro SD card you want to operate in the **Ubuntu computer** (Windows not available)

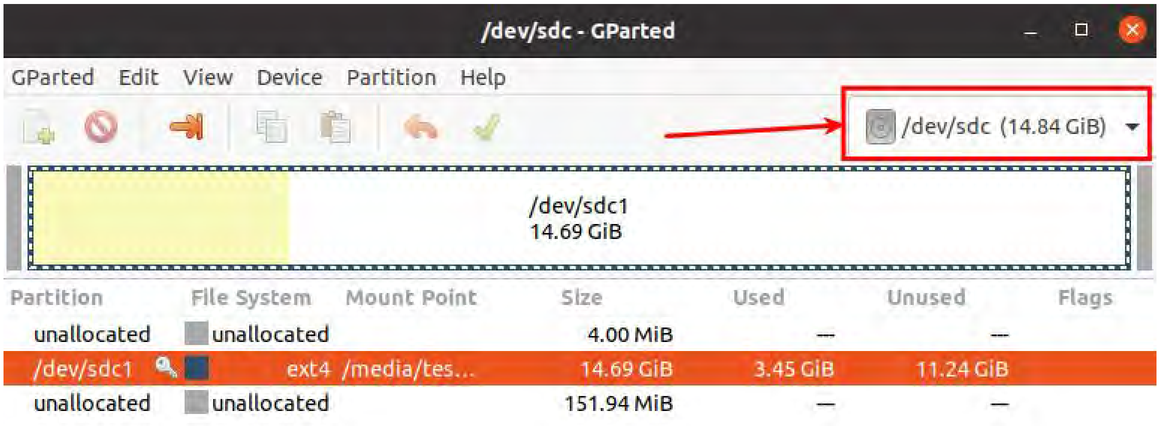
2) Then install the gparted software on the Ubuntu computer

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

3) Then open gparted

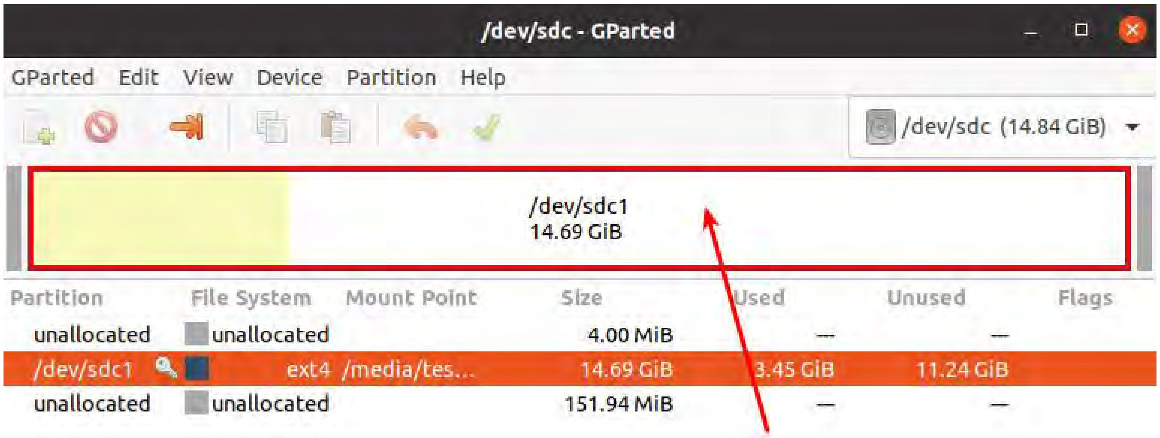
```
test@test:~$ sudo gparted
```

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



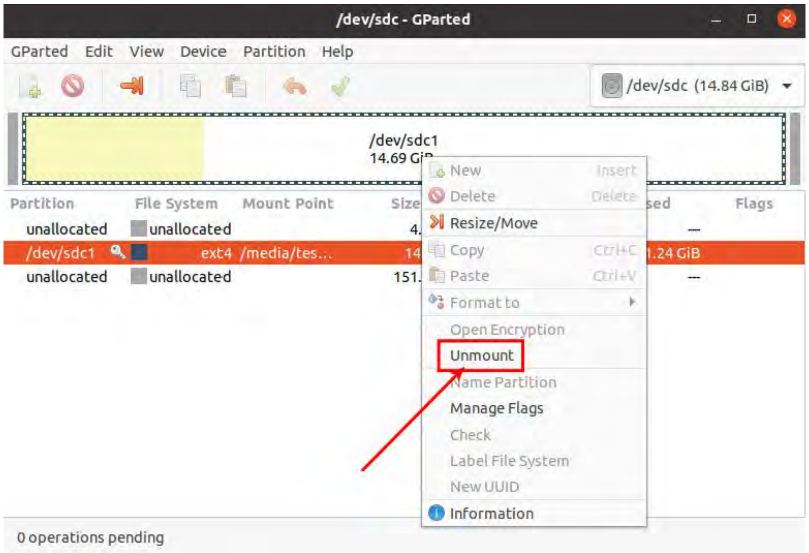
Partition	File System	Mount Point	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	—	—	
/dev/sdc1	ext4	/media/tes...	14.69 GiB	3.45 GiB	11.24 GiB	
unallocated	unallocated		151.94 MiB	—	—	

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

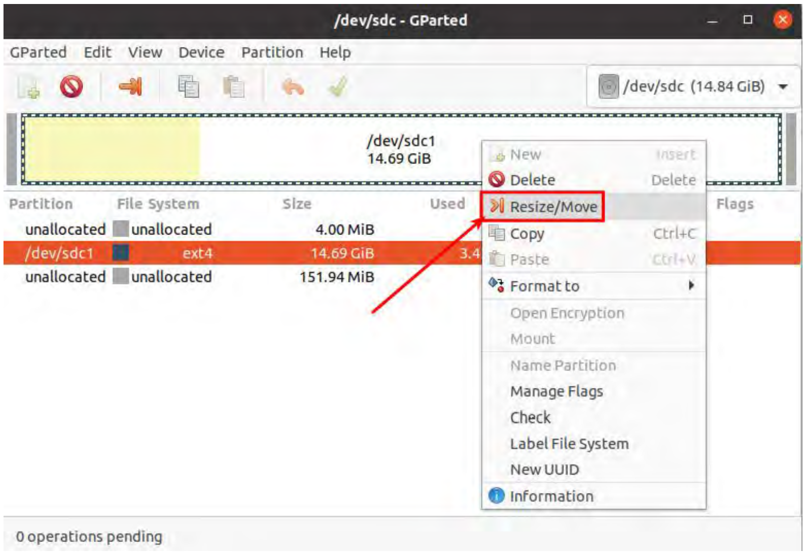


Partition	File System	Mount Point	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	—	—	
/dev/sdc1	ext4	/media/tes...	14.69 GiB	3.45 GiB	11.24 GiB	
unallocated	unallocated		151.94 MiB	—	—	

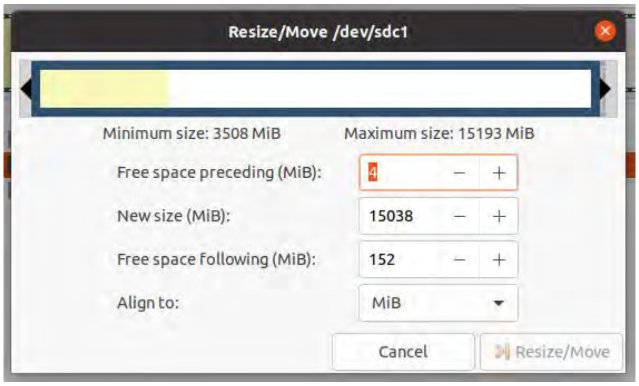
6) Click the right mouse button again to see the operation options shown in the figure below. If the micro SD card has been mounted, first you need to Umount the rootfs partition of the micro SD card



7) Then select the rootfs partition again, click the right mouse button, 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


Resize/Move /dev/sdc1

Minimum size: 3508 MiB Maximum size: 15193 MiB

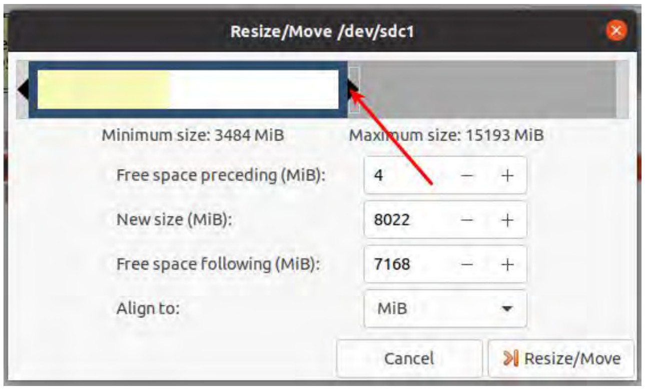
Free space preceding (MiB): 4

New size (MiB): 15038

Free space following (MiB): 152

Align to: MiB

9) Then you can directly drag the location 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)**


Resize/Move /dev/sdc1

Minimum size: 3484 MiB Maximum size: 15193 MiB

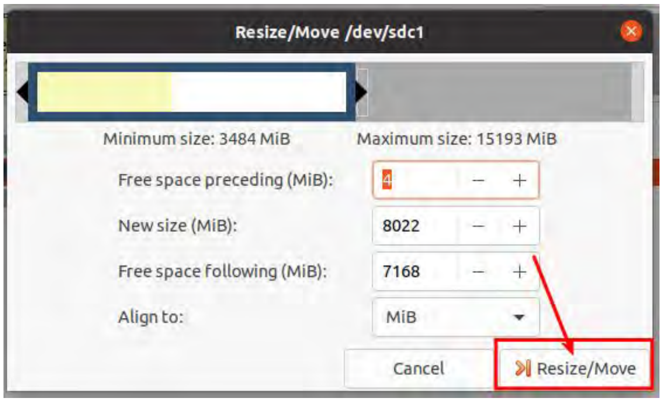
Free space preceding (MiB): 4

New size (MiB): 8022

Free space following (MiB): 7168

Align to: MiB

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



Resize/Move /dev/sdc1

Minimum size: 3484 MiB Maximum size: 15193 MiB

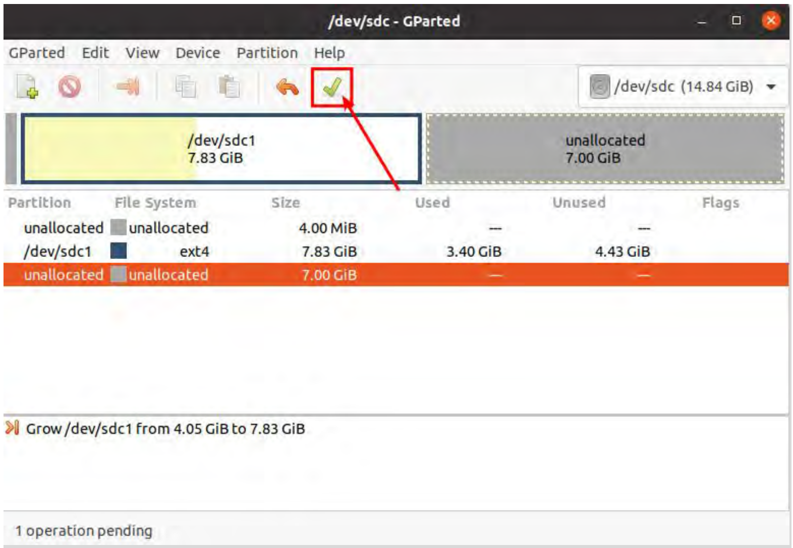
Free space preceding (MiB): 4

New size (MiB): 8022

Free space following (MiB): 7168

Align to: MiB

11) After confirming that it is correct, click the green button shown in the figure below √



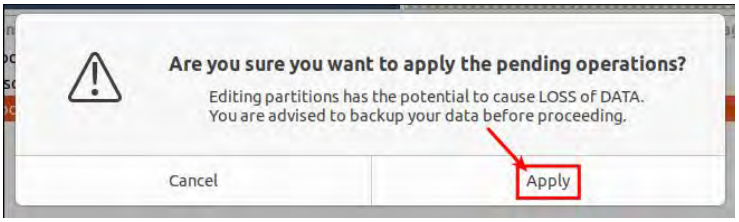
/dev/sdc - GParted

Partition	File System	Size	Used	Unused	Flags
unallocated	unallocated	4.00 MiB	—	—	
/dev/sdc1	ext4	7.83 GiB	3.40 GiB	4.43 GiB	
unallocated	unallocated	7.00 GiB	—	—	

Grow /dev/sdc1 from 4.05 GiB to 7.83 GiB

1 operation pending

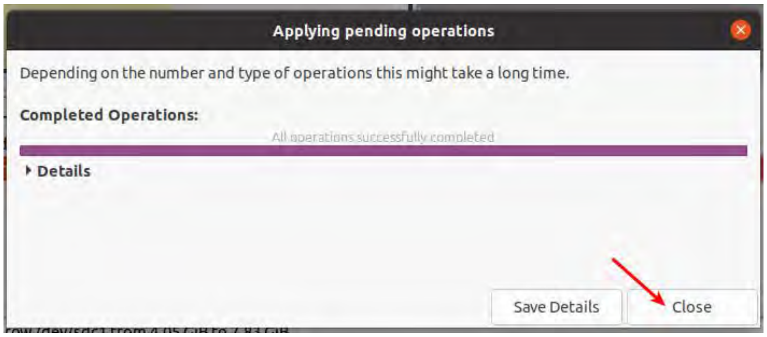
12) Then select **Apply**, and the expansion of the rootfs partition will officially start



Are you sure you want to apply the pending operations?

Editing partitions has the potential to cause LOSS of DATA.
You are advised to backup your data before proceeding.

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



Applying pending operations

Depending on the number and type of operations this might take a long time.

Completed Operations:

All operations successfully completed

14) Then you can pull out the micro SD card, insert it into the development board and start it up. After entering the Linux system of the development board, if you can use the **df -h** command to see that the size of the rootfs partition is the same as the size set before, it means that the size has been 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. Network connection test

3. 7. 1. Ethernet port test

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

2) After the system starts, it will automatically assign an IP address to the Ethernet card through **DHCP, without any other configuration**

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

Please do not copy the following commands. For example, the network node name in debian12 is end0, and the following commands need to be changed to ip a s end0.

```
orangepi@orangepi:~$ ip a s 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 starts:

1. Connect the HDMI monitor, then log in to the system and use the ip a s eth0 command to view the IP address

2. Enter the ip a s eth0 command in the debugging serial terminal to view the IP address

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

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

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

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

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

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

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

Please do not copy the following commands. For example, the network node name in debian12 is end0, and the following command needs to be changed to ping www.baidu.com -I end0.

```
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. 7. 2. WIFI connection test

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

3. 7. 2. 1. The server image connects to WIFI through commands

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

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

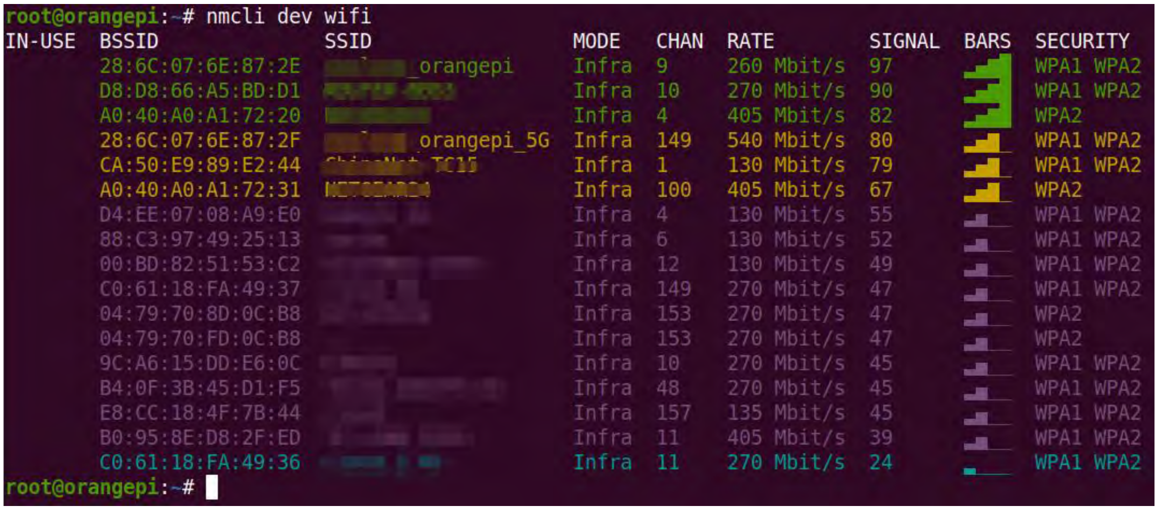
a. If the development board is connected with a network cable, **SSH remote login development board under Ubuntu**

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

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

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

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



```
root@orangepi:~# nmcli dev wifi
IN-USE  BSSID              SSID                 MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
        28:6C:07:6E:87:2E  [redacted]_orangepi     Infra  9     260 Mbit/s  97      ▂▄▆█  WPA1 WPA2
        D8:D8:66:A5:BD:D1  [redacted]              Infra  10    270 Mbit/s  90      ▂▄▆█  WPA1 WPA2
        A0:40:A0:A1:72:20  [redacted]              Infra  4     405 Mbit/s  82      ▂▄▆█  WPA2
        28:6C:07:6E:87:2F  [redacted]_orangepi_5G  Infra  149   540 Mbit/s  80      ▂▄▆_  WPA1 WPA2
        CA:50:E9:89:E2:44  [redacted]              Infra  1     130 Mbit/s  79      ▂▄▆_  WPA1 WPA2
        A0:40:A0:A1:72:31  [redacted]              Infra  100   405 Mbit/s  67      ▂▄▆_  WPA2
        D4:EE:07:08:A9:E0  [redacted]              Infra  4     130 Mbit/s  55      ▂▄__  WPA1 WPA2
        88:C3:97:49:25:13  [redacted]              Infra  6     130 Mbit/s  52      ▂▄__  WPA1 WPA2
        00:BD:82:51:53:C2  [redacted]              Infra  12    130 Mbit/s  49      ▂▄__  WPA1 WPA2
        C0:61:18:FA:49:37  [redacted]              Infra  149   270 Mbit/s  47      ▂▄__  WPA1 WPA2
        04:79:70:8D:0C:B8  [redacted]              Infra  153   270 Mbit/s  47      ▂▄__  WPA2
        04:79:70:FD:0C:B8                          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 replace it with the name of the WIFI hotspot you want to connect to

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

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

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

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

```
        valid_lft 259192sec preferred_lft 172792sec
    inet6 fe80::42f1:6019:a80e:4c31/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

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

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

3. 7. 2. 2. The server image connects to WIFI in a graphical way

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

a. **If the development board is connected with a network cable**, **SSH remote login development board under Ubuntu**

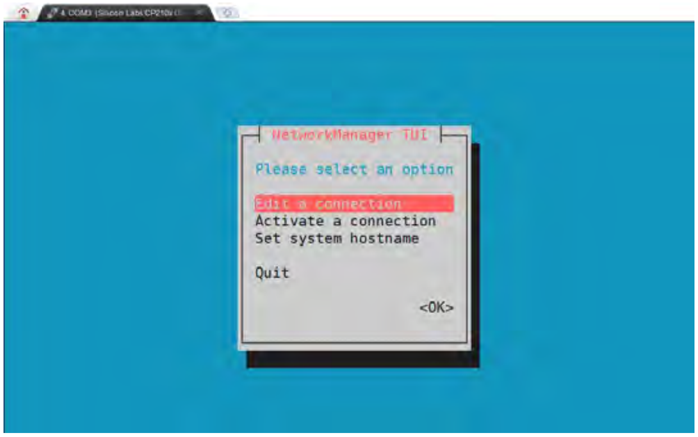
b. If the development board is connected to the debugging serial port, you can use the serial port terminal to log in to the linux system (please use MobaXterm for the serial port software, and the minicom cannot display the graphical interface)

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

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

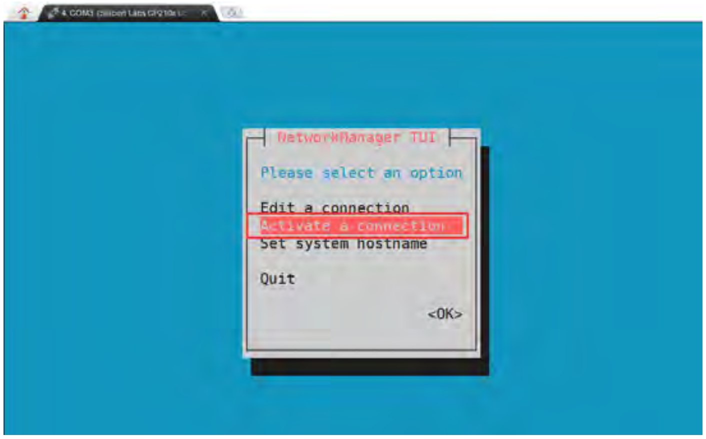
```
orangepi@orangepi:~$ sudo nmtui
```

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



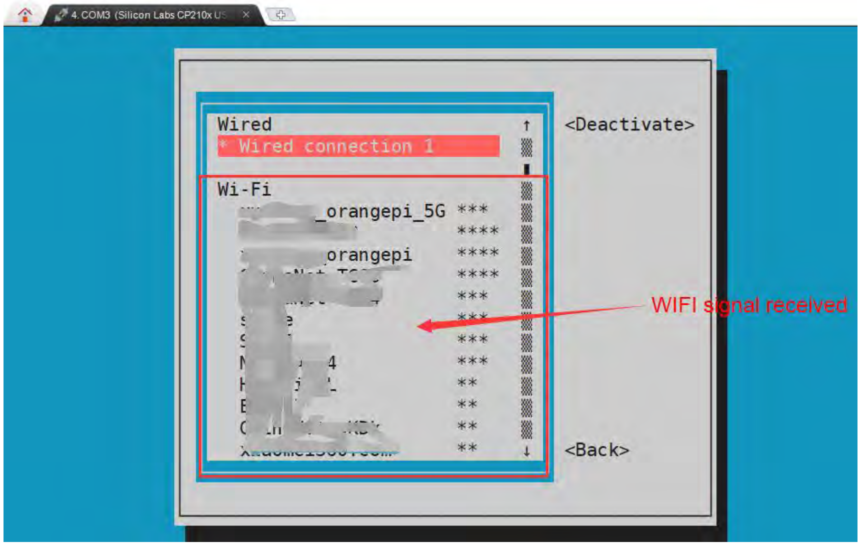
```
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

Quit

                 <OK>
```

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



```
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

Quit

                 <OK>
```

5) Then you can see all the searched WIFI hotspots

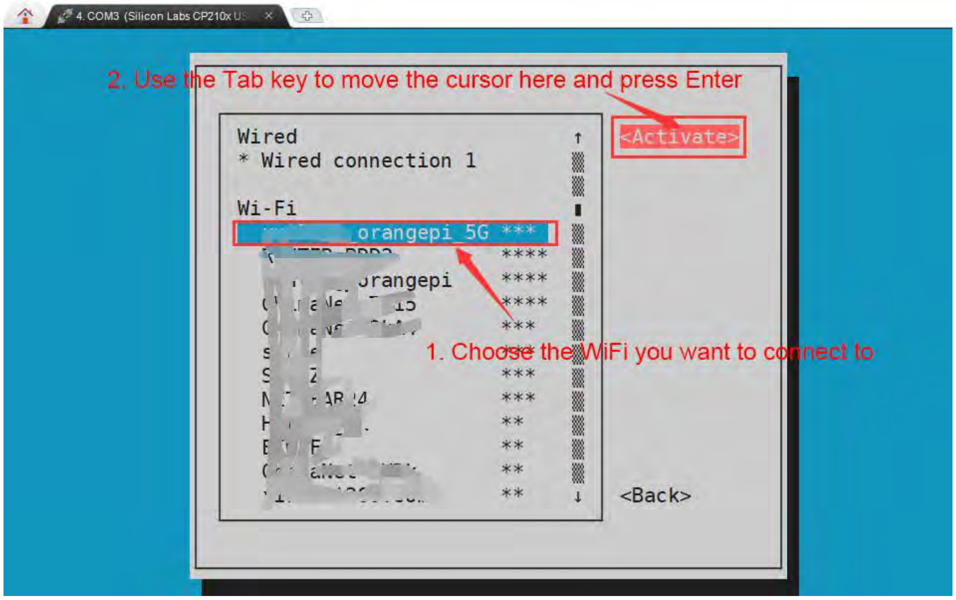


```
Wired                                   <Deactivate>
* Wired connection 1

Wi-Fi
  [redacted]_orangepi_5G  ***
                          ****
  [redacted]orangepi      ****
  [redacted]              ****
                          ***
                          ***
                          ***
                          ***
                          **
                          **
                          **
                          **        <Back>
```

WIFI signal received

6) Select the WIFI hotspot you want to connect to, then use the Tab key to position the

cursor on **Activate** and press Enter

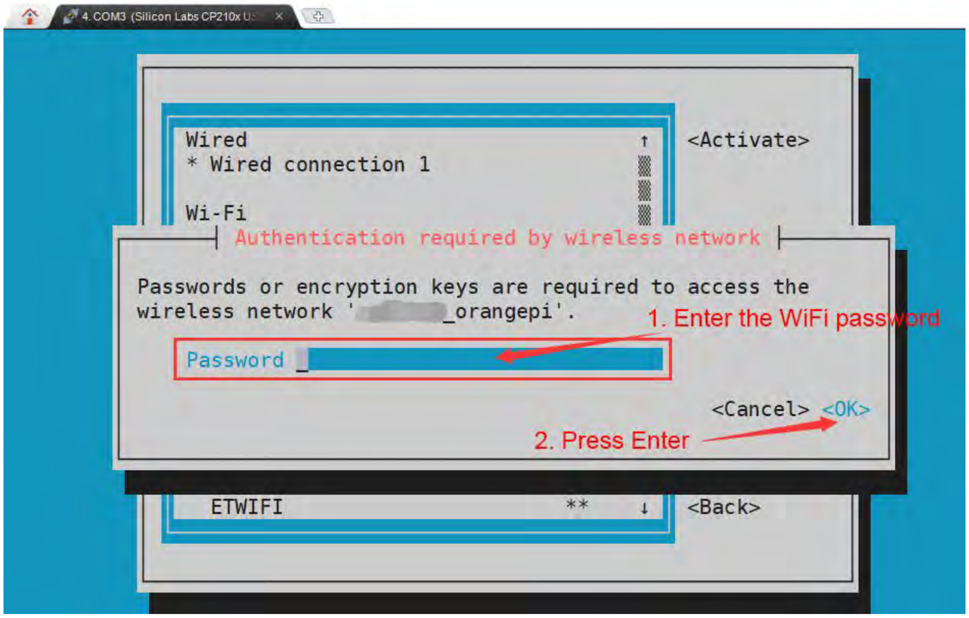


```
Wired                              ↑    <Activate>
* Wired connection 1
                                   
Wi-Fi
  [redacted]orangepi_5G ***
[redacted]
                                   ↓    <Back>
```

2. Use the Tab key to move the cursor here and press Enter

1. Choose the WiFi you want to connect to

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

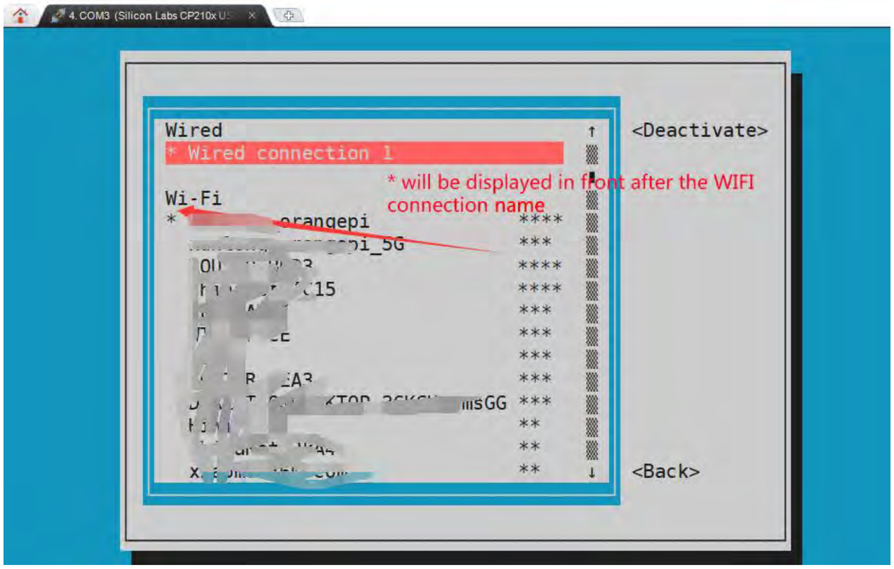


```
Wired                              ↑    <Activate>
* Wired connection 1

Wi-Fi
┤ Authentication required by wireless network ├

Passwords or encryption keys are required to access the
wireless network '[redacted]_orangepi'.

Password _

                                    <Cancel> <OK>

  ETWIFI                        **  ↓    <Back>
```

1. Enter the WiFi password

2. Press Enter

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



```
Wired                                    ↑   <Deactivate>
* Wired connection 1
Wi-Fi
*           orangepi              ****
[redacted]
                                         ↓   <Back>
```

9) You can view the IP address of wifi through the **ip a s wlan0** command

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

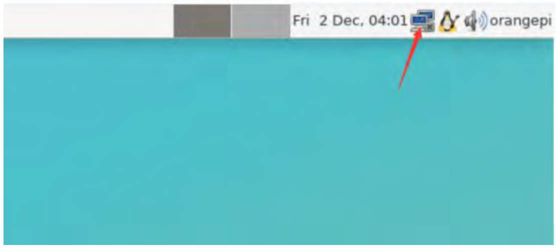
10) Use the **ping** command to test the connectivity of the wifi network, and the **ping** command can be interrupted through the shortcut key **Ctrl+C**

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

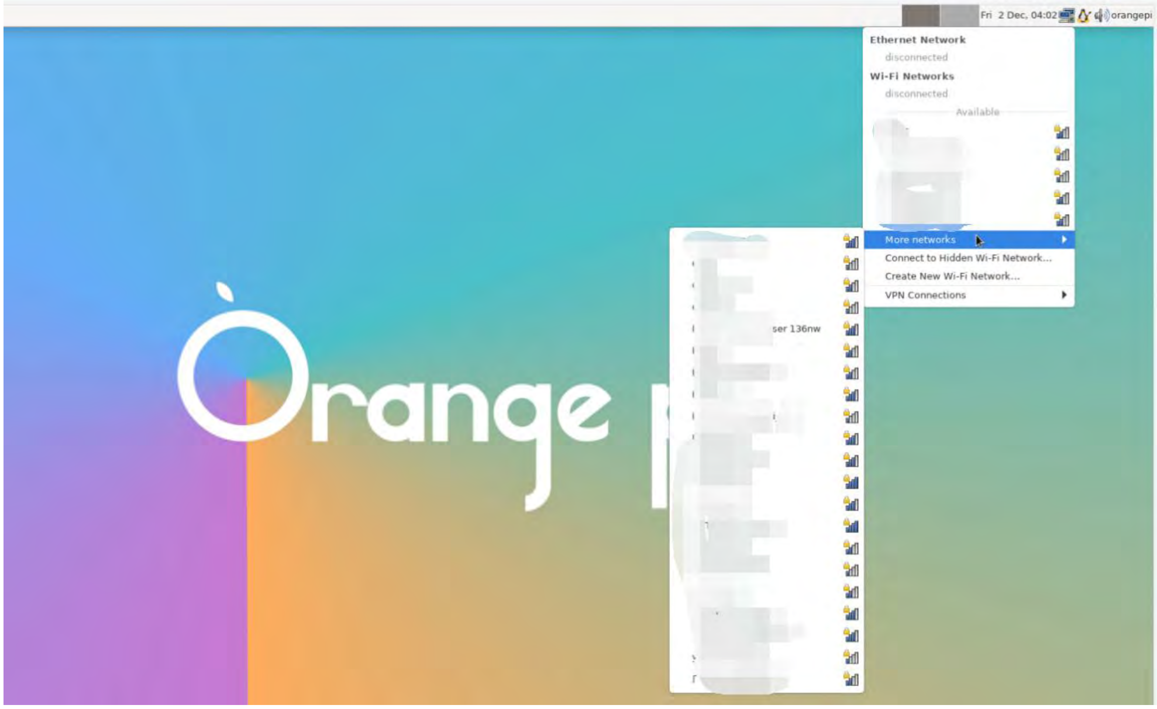
```
^C
--- www.orangepi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

3. 7. 2. 3. Test method of desktop image

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

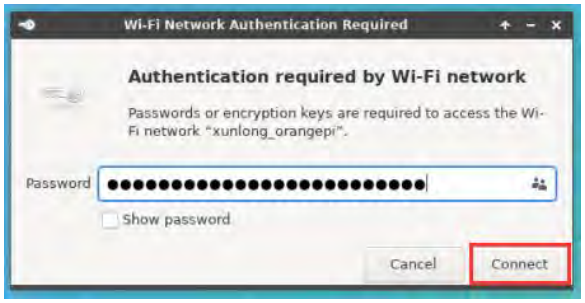


2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to



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

to WIFI



Wi-Fi Network Authentication Required

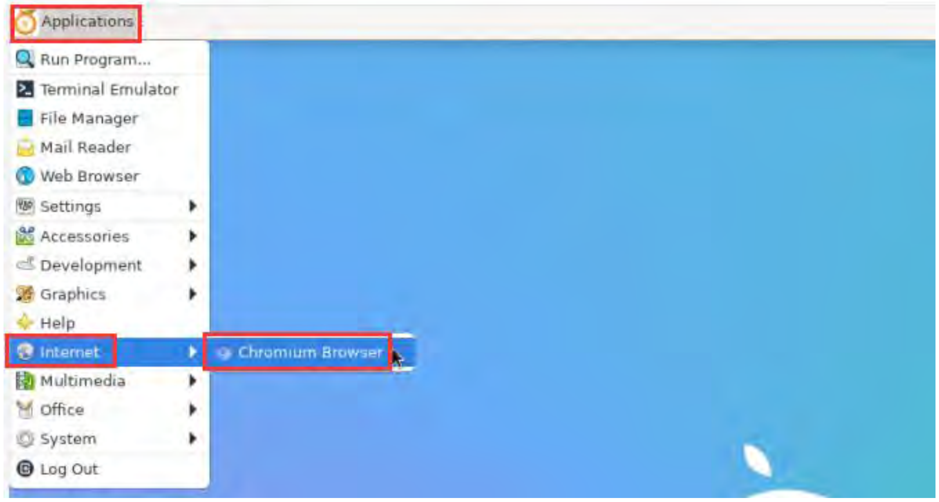
Authentication required by Wi-Fi network

Passwords or encryption keys are required to access the Wi-Fi network "xunlong_orangepi".

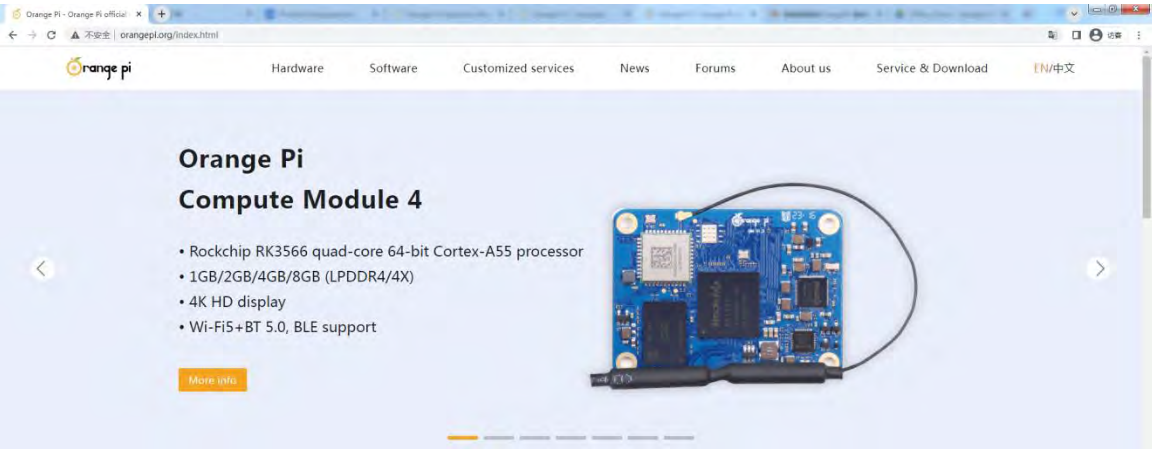
Password ●●●●●●●●●●●●●●●●●●●●●●●●●●

□ Show password

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



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



Orange Pi
Compute Module 4

- Rockchip RK3566 quad-core 64-bit Cortex-A55 processor
- 1GB/2GB/4GB/8GB (LPDDR4/4X)
- 4K HD display
- Wi-Fi5+BT 5.0, BLE support

3. 7. 3. The method of creating WIFI hotspot through create_ap

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

https://github.com/oblique/create_ap

The Linux image released by OPi has pre-installed the create_ap script. You can create a WIFI hotspot through the create_ap command. The basic command format of create_ap is as follows:

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

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

3. 7. 3. 1. create_ap method to create WIFI hotspot in NAT mode

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

Note that in the following commands, Debian12 needs to change eth0 to end0.

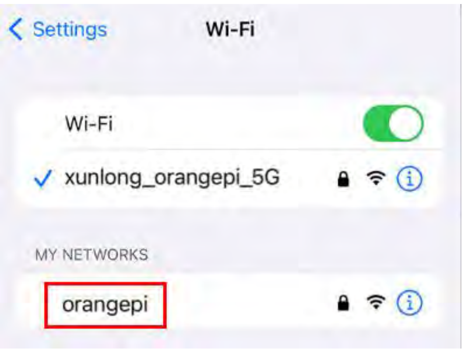
```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --no-virt
```

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

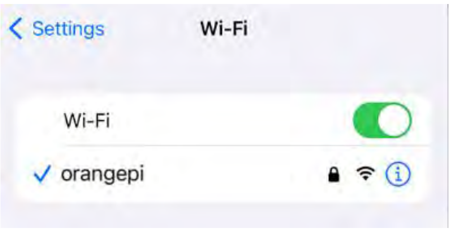
```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --no-virt
Config dir: /tmp/create_ap.wlan0.conf.TQkJtsz1
PID: 26139
```

```
Network Manager found, set wlan0 as unmanaged device... DONE
Sharing Internet using method: nat
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.TQkJtsz1/hostapd_ctrl
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA ce:bd:9a:dd:a5:86 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED ce:bd:9a:dd:a5:86
wlan0: STA ce:bd:9a:dd:a5:86 RADIUS: starting accounting session
D4FBF7E5C604F169
wlan0: STA ce:bd:9a:dd:a5:86 WPA: pairwise key handshake completed (RSN)
wlan0: EAPOL-4WAY-HS-COMPLETED ce:bd:9a:dd:a5:86
```

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



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

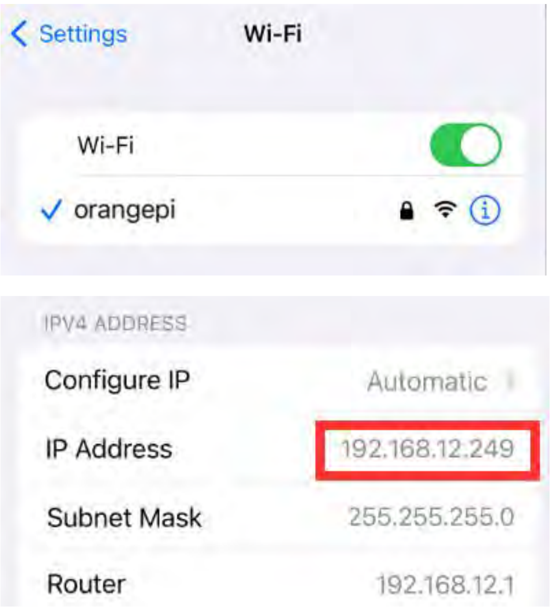


5）In NAT mode, the wireless device connected to the hotspot of the development board requests an IP address from the DHCP service of the development board, so there will be two different network segments, for example, the IP of the development board is 192.168.1.X

Note that in the following commands, Debian12 needs to change eth0 to end0.

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

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



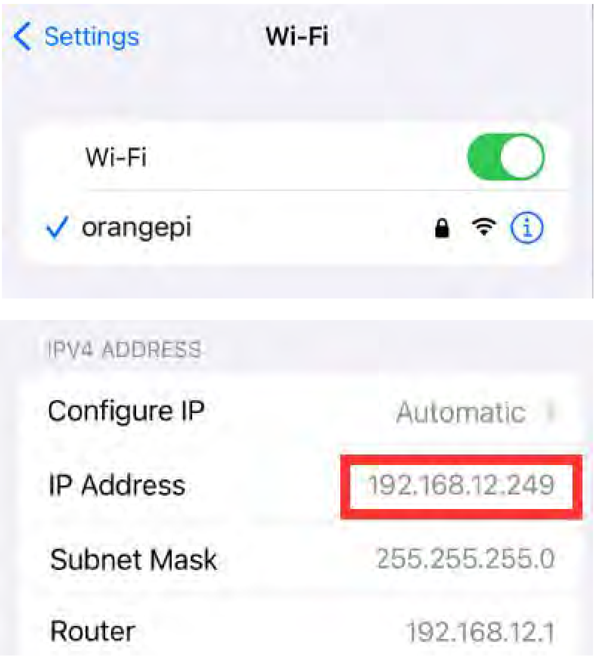
Configure IP: Automatic
IP Address: 192.168.12.249
Subnet Mask: 255.255.255.0
Router: 192.168.12.1

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

Note that in the following commands, Debian12 needs to change eth0 to end0.

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

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



IPV4 ADDRESS

Configure IP Automatic

IP Address 192.168.12.249

Subnet Mask 255.255.255.0

Router 192.168.12.1

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

Note that in the following commands, Debian12 needs to change eth0 to end0.

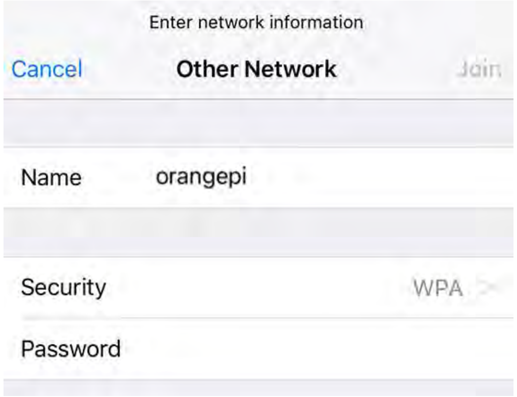
```
orangepi@orangepi:~$ sudo create_ap -m nat wlan0 eth0 orangepi orangepi --freq-band 5    --no-virt
```

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

Note that in the following commands, Debian12 needs to change eth0 to end0.

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

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



Enter network information

Cancel Other Network Join

Name orangepi

Security WPA

Password

3. 7. 3. 2. create_ap method to create WIFI hotspot in bridge mode

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

Note that in the following commands, Debian12 needs to change eth0 to end0.

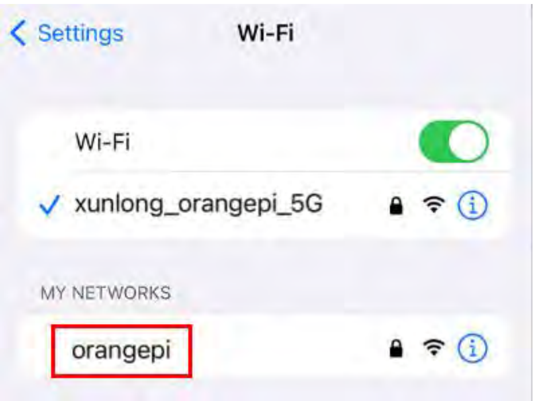
```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi    --no-virt
```

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

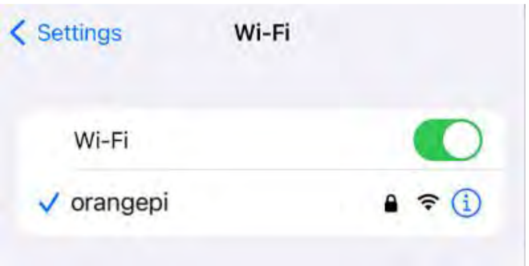
```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi    --no-virt
Config dir: /tmp/create_ap.wlan0.conf.zAcFlYTx
PID: 27707
Network Manager found, set wlan0 as unmanaged device... DONE
Sharing Internet using method: bridge
Create a bridge interface... br0 created.
hostapd command-line interface: hostapd_cli -p
/tmp/create_ap.wlan0.conf.zAcFlYTx/hostapd_ctrl
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA ce:bd:9a:dd:a5:86 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED ce:bd:9a:dd:a5:86
wlan0: STA ce:bd:9a:dd:a5:86 RADIUS: starting accounting session
937BF40E51897A7B
wlan0: STA ce:bd:9a:dd:a5:86 WPA: pairwise key handshake completed (RSN)
```

```
wlan0: EAPOL-4WAY-HS-COMPLETED ce:bd:9a:dd:a5:86
```

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



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

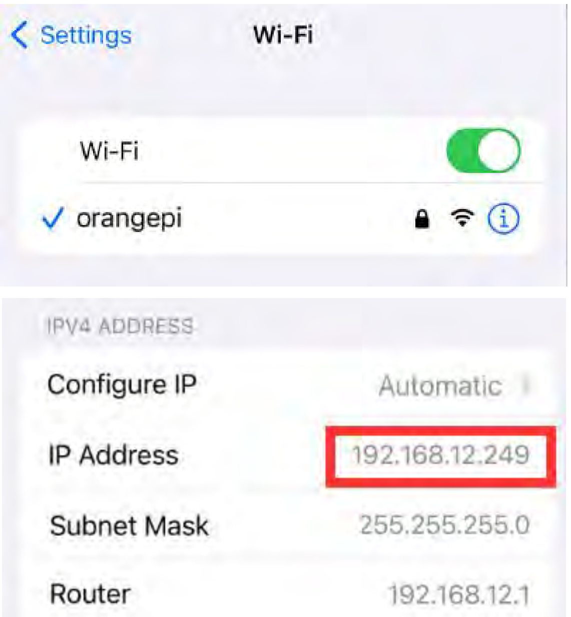


5）In bridge mode, the wireless device connected to the hotspot of the development board also requests an IP address from the DHCP service of the main router (the router connected to the development board), for example, the IP of the development board is **192.168.1.X**

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

```
            device interrupt 83
```

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



Wi-Fi

✓ orangepi

IPV4 ADDRESS

Configure IP Automatic

IP Address 192.168.12.249

Subnet Mask 255.255.255.0

Router 192.168.12.1

6）If the **--freq-band** y band. If you want to create a hotspot in the 5G frequency band, you can specify the **--freq-band 5** parameter. The specific command is as follows

Note that in the following commands, Debian12 needs to change eth0 to end0.

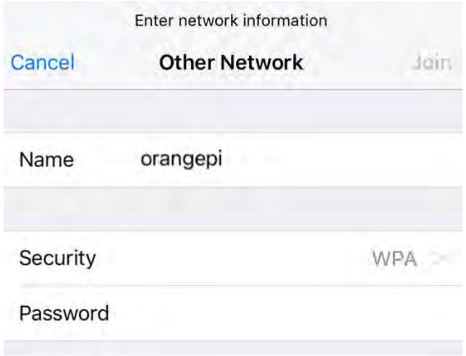
```
orangepi@orangepi:~$ sudo create_ap -m bridge wlan0 eth0 orangepi orangepi --freq-band 5    --no-virt
```

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

Note that in the following commands, Debian12 needs to change eth0 to end0.

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

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



Enter network information

Cancel Other Network Join

Name orangepi

Security WPA

Password

3. 7. 4. How to set static IP address

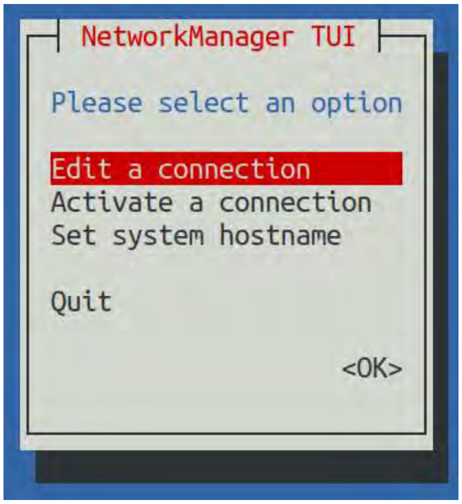
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file.

3. 7. 4. 1. Use the nmtui command to set a static IP address

1) First run the **nmtui** command

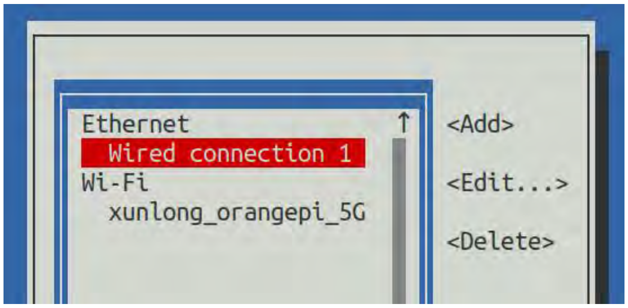
```
orangepi@orangepi:~$ sudo nmtui
```

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



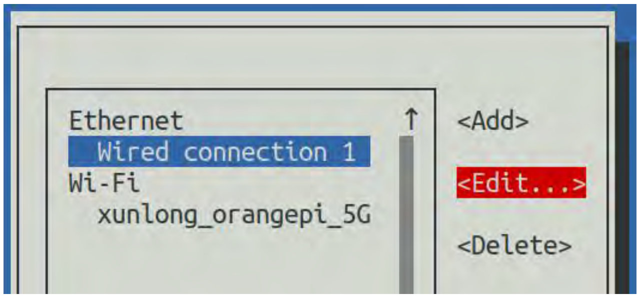
```
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

Quit

                  <OK>
```

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



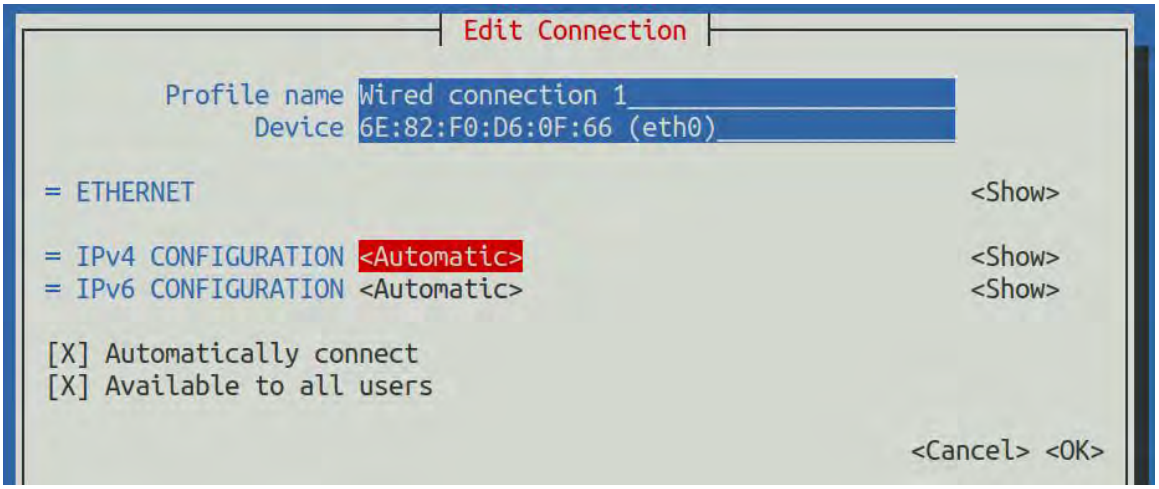
```
Ethernet                        ↑   <Add>
  Wired connection 1
Wi-Fi                               <Edit...>
  xunlong_orangepi_5G
                                    <Delete>
```

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



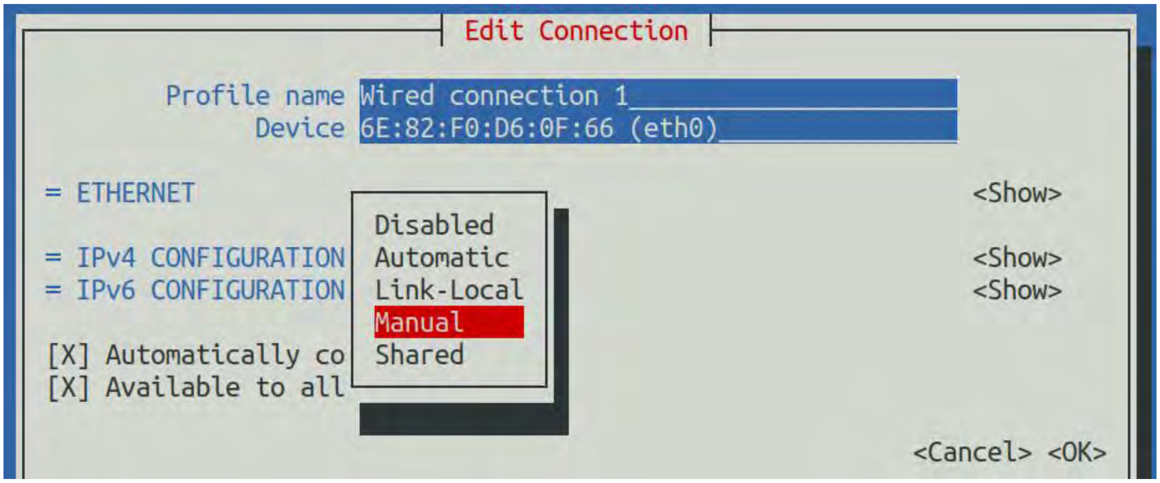
```
Ethernet                        ↑   <Add>
  Wired connection 1
Wi-Fi                               <Edit...>
  xunlong_orangepi_5G
                                    <Delete>
```

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



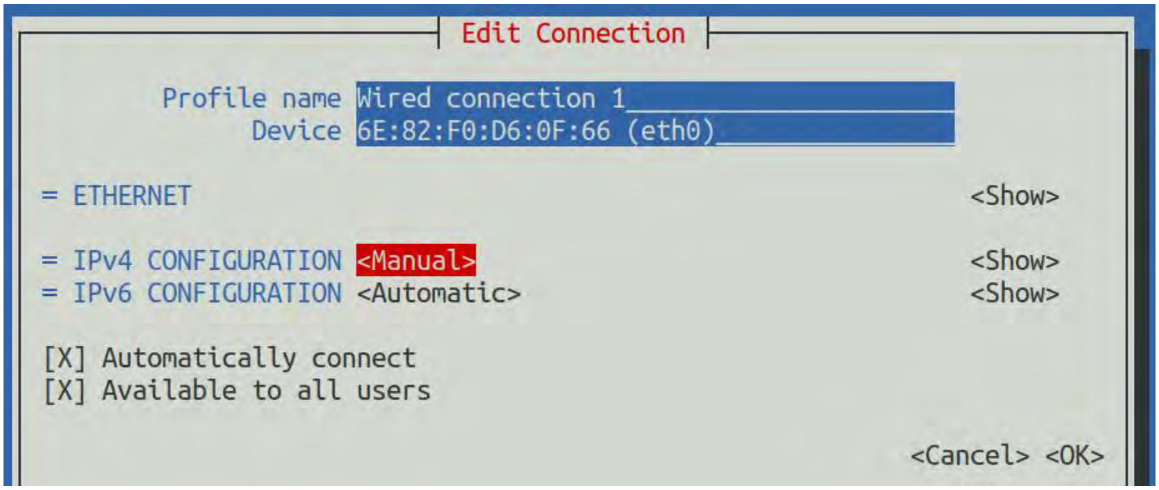
```
                         Edit Connection

           Profile name Wired connection 1_______________
                 Device 6E:82:F0:D6:0F:66 (eth0)_________

= ETHERNET                                          <Show>

= IPv4 CONFIGURATION <Automatic>                    <Show>
= IPv6 CONFIGURATION <Automatic>                    <Show>

[X] Automatically connect
[X] Available to all users

                                              <Cancel> <OK>
```

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



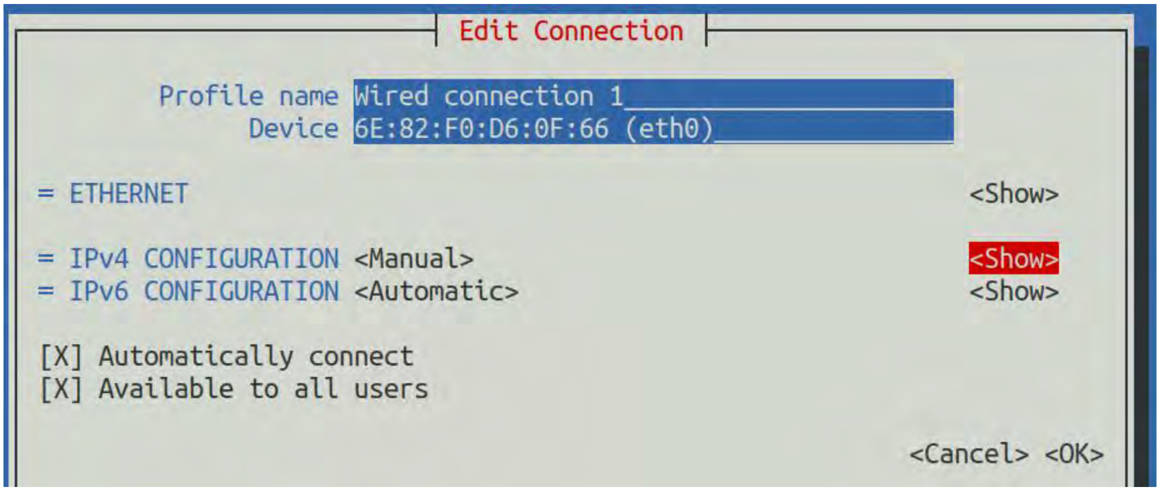
```
                        ┤ Edit Connection ├

       Profile name Wired connection 1_________________
             Device 6E:82:F0:D6:0F:66 (eth0)___________

 = ETHERNET                                                  <Show>
                     Disabled
 = IPv4 CONFIGURATION Automatic                              <Show>
 = IPv6 CONFIGURATION Link-Local                             <Show>
                     Manual
 [X] Automatically co Shared
 [X] Available to all

                                                       <Cancel> <OK>
```

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



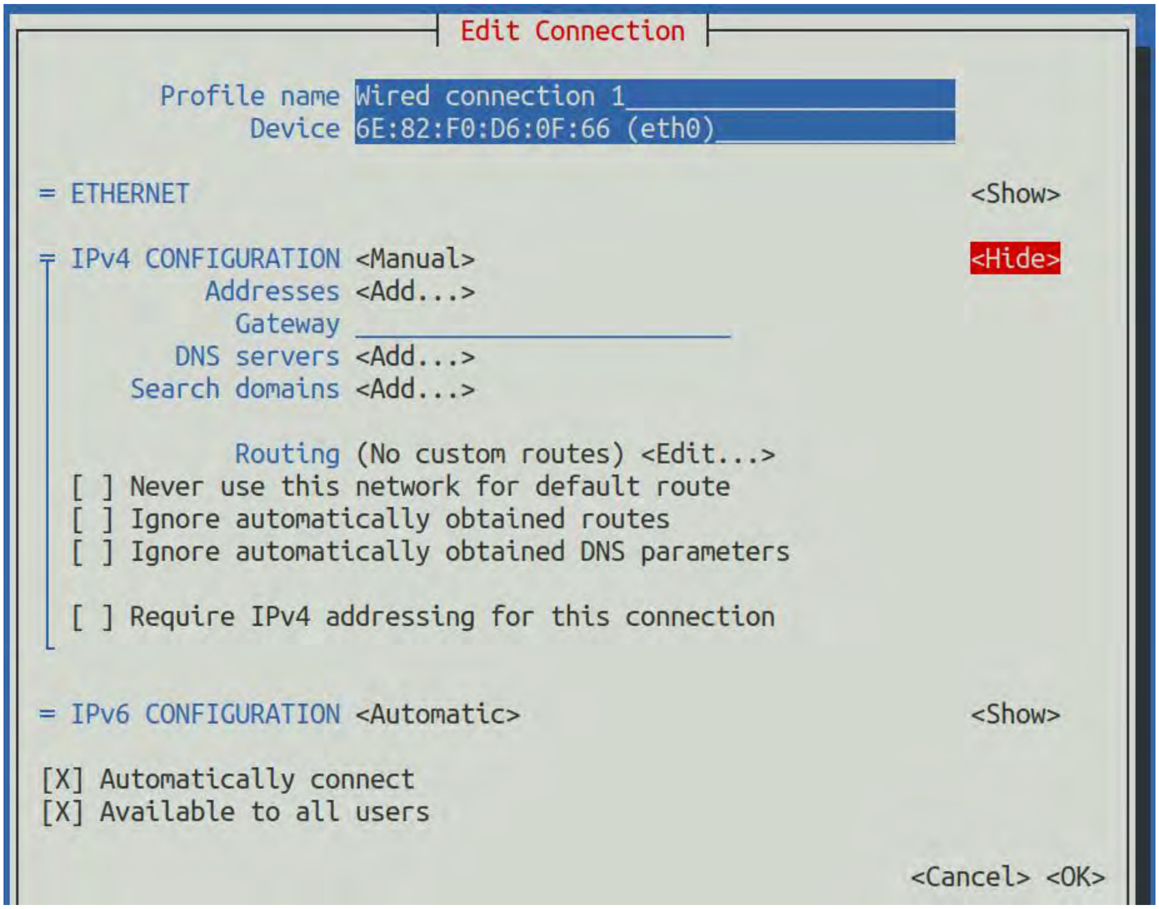
```
                        ┤ Edit Connection ├

       Profile name Wired connection 1_________________
             Device 6E:82:F0:D6:0F:66 (eth0)___________

 = ETHERNET                                                  <Show>

 = IPv4 CONFIGURATION <Manual>                               <Show>
 = IPv6 CONFIGURATION <Automatic>                            <Show>

 [X] Automatically connect
 [X] Available to all users

                                                       <Cancel> <OK>
```

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



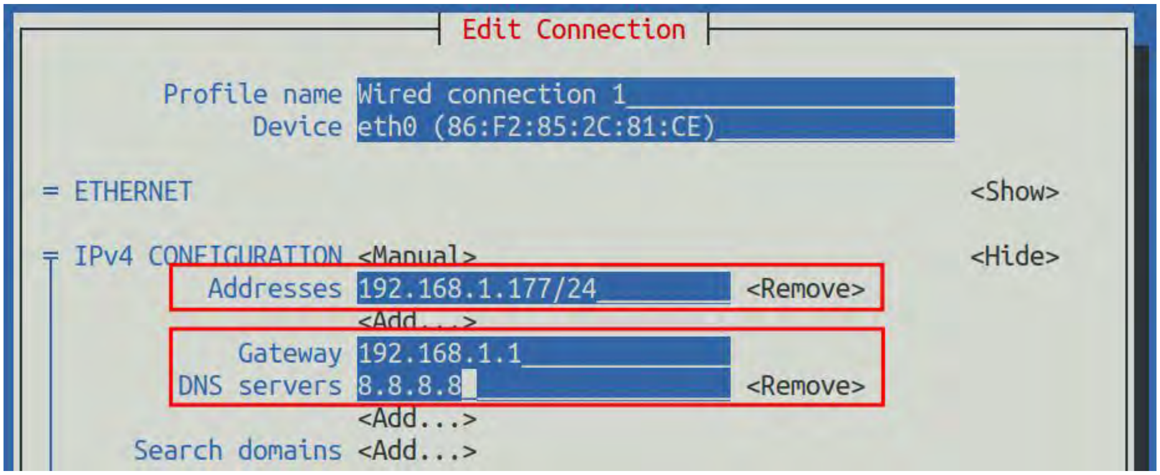
```
                        ┤ Edit Connection ├

       Profile name Wired connection 1_________________
             Device 6E:82:F0:D6:0F:66 (eth0)___________

 = ETHERNET                                                  <Show>

 = IPv4 CONFIGURATION <Manual>                               <Show>
 = IPv6 CONFIGURATION <Automatic>                            <Show>

 [X] Automatically connect
 [X] Available to all users

                                                       <Cancel> <OK>
```

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



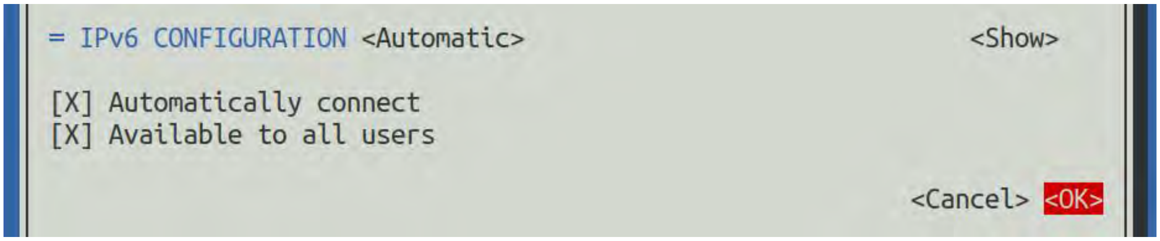
```
                          Edit Connection

     Profile name Wired connection 1
           Device 6E:82:F0:D6:0F:66 (eth0)

= ETHERNET                                                      <Show>

= IPv4 CONFIGURATION <Manual>                                   <Hide>
           Addresses <Add...>
             Gateway
         DNS servers <Add...>
      Search domains <Add...>

             Routing (No custom routes) <Edit...>
  [ ] Never use this network for default route
  [ ] Ignore automatically obtained routes
  [ ] Ignore automatically obtained DNS parameters

  [ ] Require IPv4 addressing for this connection

= IPv6 CONFIGURATION <Automatic>                                <Show>

[X] Automatically connect
[X] Available to all users

                                                          <Cancel> <OK>
```

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



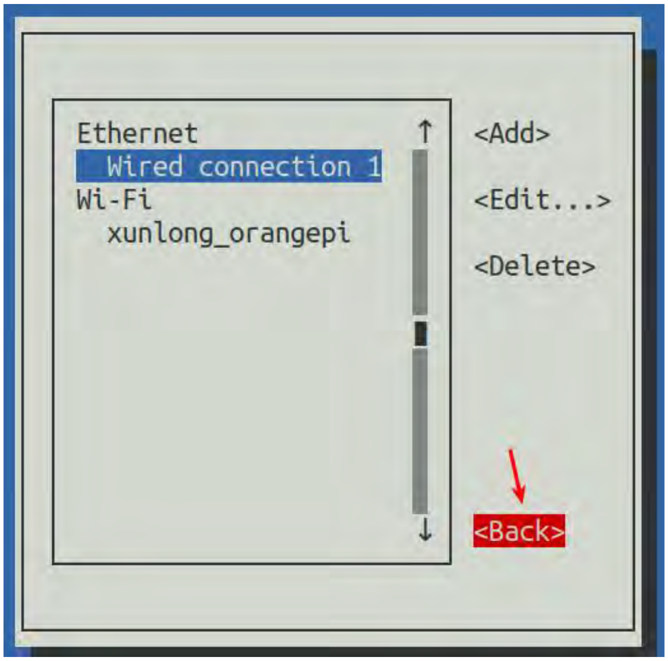
```
                          Edit Connection

     Profile name Wired connection 1
           Device eth0 (86:F2:85:2C:81:CE)

= ETHERNET                                                      <Show>

= IPv4 CONFIGURATION <Manual>                                   <Hide>
           Addresses 192.168.1.177/24          <Remove>
                     <Add...>
             Gateway 192.168.1.1
         DNS servers 8.8.8.8                   <Remove>
                     <Add...>
      Search domains <Add...>
```

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



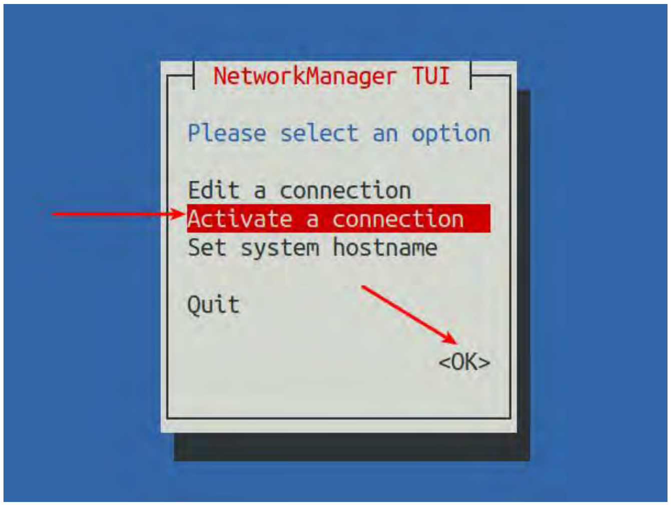
```
= IPv6 CONFIGURATION <Automatic>                    <Show>

[X] Automatically connect
[X] Available to all users

                                            <Cancel> <OK>
```

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

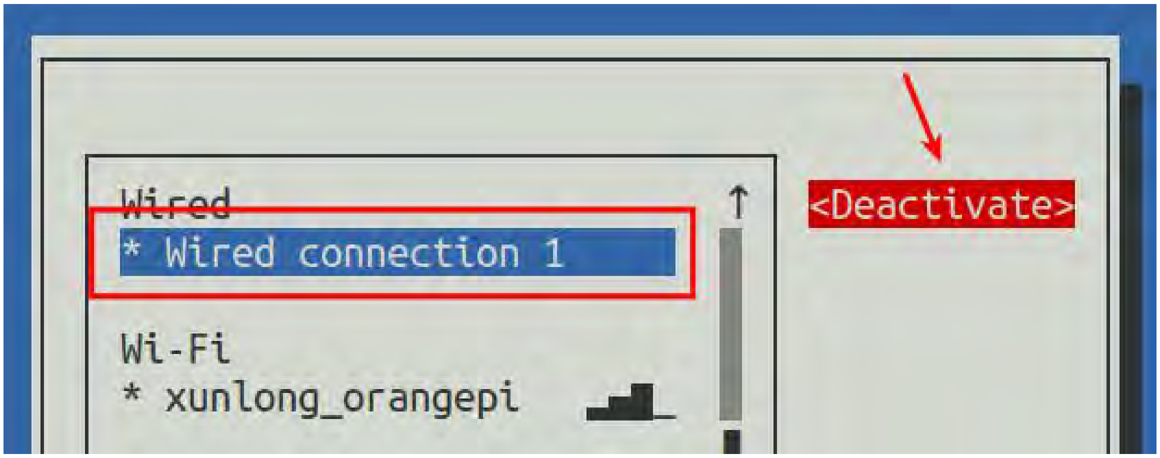


```
Ethernet                  ↑  <Add>
  Wired connection 1
Wi-Fi                        <Edit...>
  xunlong_orangepi
                             <Delete>



                          ↓  <Back>
```

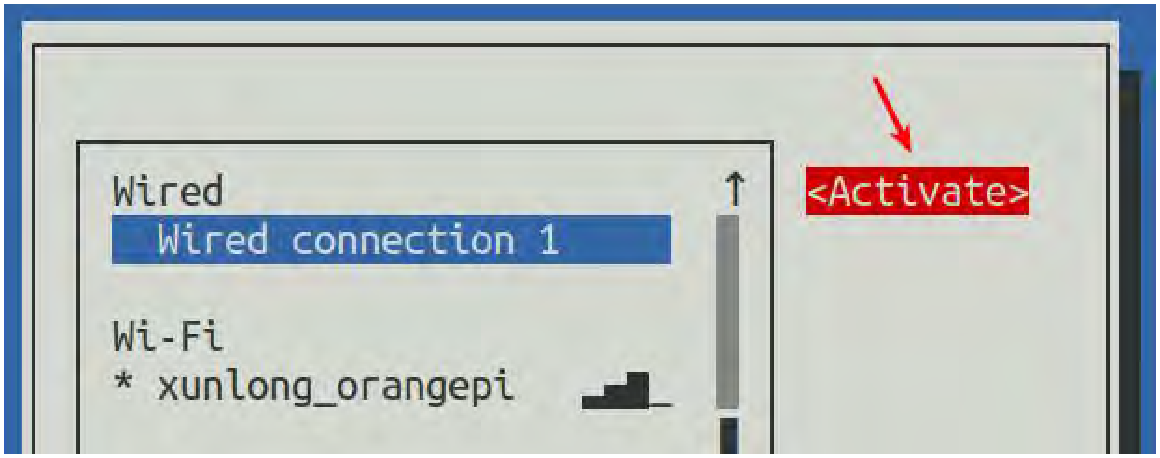
13) Then select **Activate a connection**, then move the cursor to **<OK>**, and finally click Enter



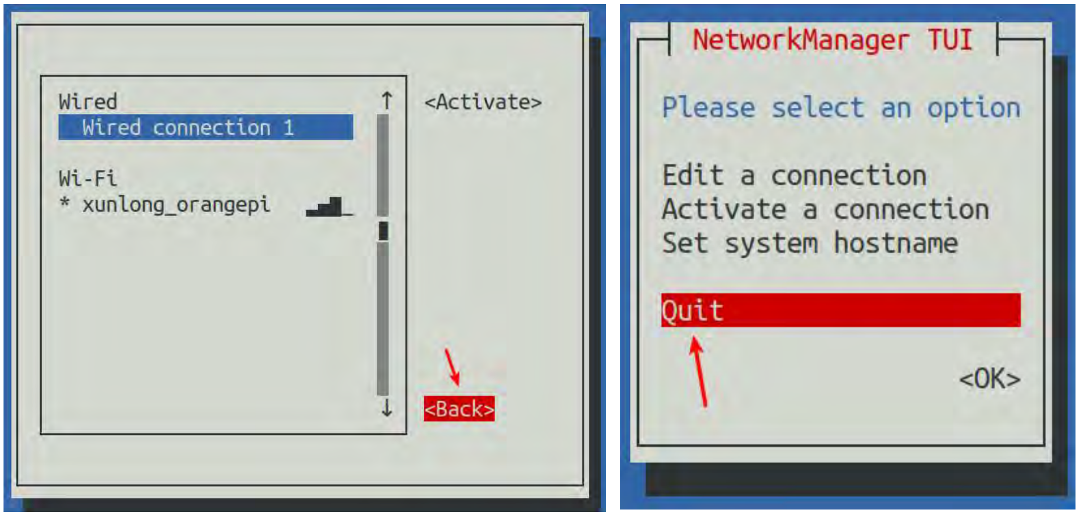
14) Then select the network interface that needs to be set, such as **Wired connection 1**, then move the cursor to <Deactivate>, and press the Enter key to disable **Wired connection 1**



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



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



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

Note that in the following commands, Debian12 needs to change eth0 to end0.

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

```
        valid_lft 259149sec preferred_lft 172749sec
    inet6 fe80::957d:bbbe:4928:3604/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

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

Note that in the following commands, I is capital i, Debian12 needs to change eth0 to end0.

```
orangepi@orangepi:~$ ping 192.168.1.177 -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. 7. 4. 2. Use the nmcli command to set a static IP address

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

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

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

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

```
orangepi@orangepi:~$ nmcli con show
NAME                UUID                                    TYPE        DEVICE
orangepi            cfc4f922-ae48-46f1-84e1-2f19e9ec5e2a    wifi        wlan0
```

```
Wired connection 1    9db058b7-7701-37b8-9411-efc2ae8bfa30    ethernet        eth0
```

3) Then enter the following command, where

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

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

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

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

3.7.5. The method of setting the Linux system to automatically connect to the network 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 in a network cable that can access the Internet normally 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 view 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 ssh and then connect to WIFI through commands, or through commands on the HDMI screen or serial port. Connect to WIFI.

But 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 checked through the background of the router. Or there is no HDMI screen, serial port module and network cable, and only WIFI can be connected, then you can use the method introduced in this section to automatically connect to WIFI and also set the static IP address of 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 a virtual machine with Ubuntu system installed.

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

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

In addition, if you have any problems with Paragon ExtFS for Windows software, please solve it yourself, we will not answer questions.

1) First burn the Linux image of the development board you want to use into the micro SD card, and then use a card reader to insert the micro SD card that has burned the Linux image of the development board into a machine with a Linux system (such as a machine with an Ubuntu system installed) computer, the following uses Ubuntu computer as an

example to demonstrate)

2) When the micro SD card is inserted into the Ubuntu computer, the Ubuntu computer will generally automatically mount the partition of the Linux root file system in the micro SD card. You can know from the following command that **/media/test/opi_root** is the Linux root file in the micro SD card The path where the system is mounted

```
test@test:~$ df -h | grep "media"
/dev/sdd1    1.4G    1.2G    167M    88% /media/test/opi_root
test@test:~$ ls /media/test/opi_root
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 micro SD card

```
test@test:~$ cd /media/test/opi_root/boot/
```

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

```
test@test:/media/test/opi_root/boot$ sudo cp orangepi_first_run.txt.template orangepi_first_run.txt
```

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

```
test@test:/media/test/opi_root/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 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 WIFI configuration. If you need to set the development board to automatically connect to WIFI hotspots, you must set it to 1. In addition, please note that if this variable is set to 1, the setting of the Ethernet port will be will fail. That is to say, the WIFI and Ethernet ports cannot be set at the same time (why, because it is not necessary...)

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 WIFI or Ethernet port

h. **FR_net_static_ip** variable is used to set static IP address, please set 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 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** to the password of the WIFI hotspot you want to connect to

b. For example, if you want the Linux system of the development board to automatically connect to the WIFI hotspot after the first startup, and set the IP address of the WIFI to a specific static IP address (so that when the Linux system is started, you can directly use the set static IP address to ssh remotely Log in to the development board without checking the IP address of the development board through the background of the router), 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** to 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, if you want to automatically set the IP address of the Ethernet port to the desired static IP address after the Linux system of the development board starts for the first time, you can set it like this:

a) Set **FR_net_change_defaults** to **1**

b) Set **FR_net_ethernet_enabled** 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 development board Linux system in the micro SD card, uninstall the micro SD card, and then insert the micro SD card into the development board to start

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

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, the orangepi_first_run.txt configuration file will be reset, and then the Linux system of the development board will be restarted, orangepi_first_run. The configuration in txt will not take effect again, because this configuration will only be effective when the Linux system is started for the first time after burning, please pay special attention to this point.

3. 8. SSH remote login development board

Linux systems enable ssh remote login by default and allow the root user to log in to the system. Before logging in with ssh, you first need to ensure that the Ethernet or wifi network is connected, and then use the ip addr command or check the router to obtain the IP address of the development board.

3. 8. 1. SSH remote login development board under Ubuntu

1) Obtain the IP address of the development board

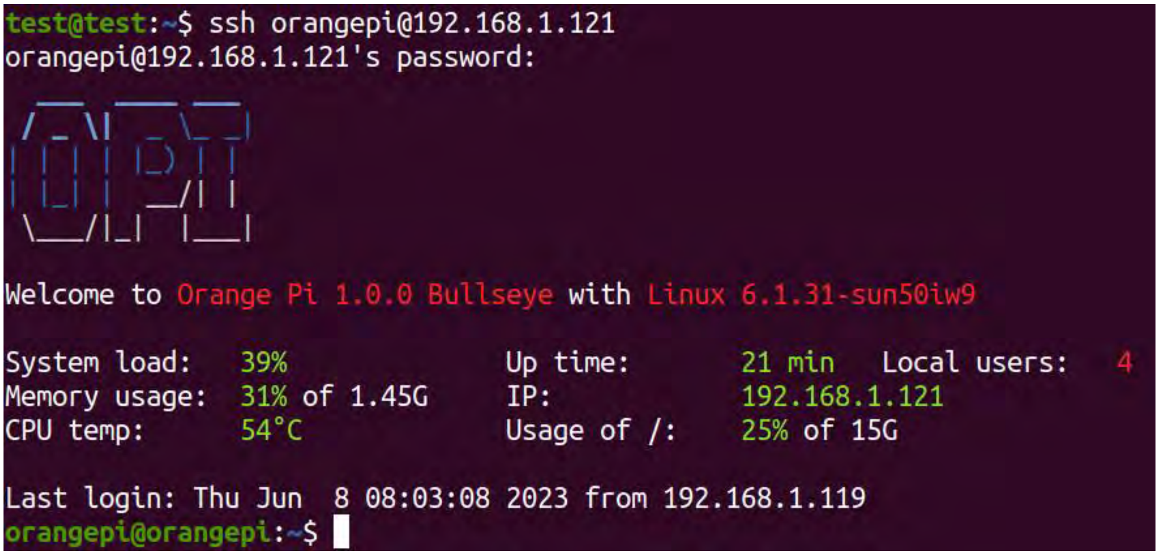
2) Then you can remotely log in to the linux system through the ssh command

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

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

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

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



```
test@test:~$ ssh orangepi@192.168.1.121
orangepi@192.168.1.121's password:
  ___  ____ ___
 / _ \|  _ \_ _|
| | | | |_) | |
| |_| |  __/| |
 \___/|_|  |___|

Welcome to Orange Pi 1.0.0 Bullseye with Linux 6.1.31-sun50iw9

System load:   39%             Up time:       21 min    Local users:   4
Memory usage:  31% of 1.45G    IP:            192.168.1.121
CPU temp:      54°C            Usage of /:    25% of 15G

Last login: Thu Jun  8 08:03:08 2023 from 192.168.1.119
orangepi@orangepi:~$
```

If ssh fails to log in to the linux system normally, first check whether the IP address of the development board can be pinged. If the ping is ok, you can log in to the linux system through the serial port or HDMI display and then enter the following command on the development board and try again. Is it possible to connect:

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

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

3. 8. 2. SSH remote login development board under Windows

1) First obtain the IP address of the development board

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

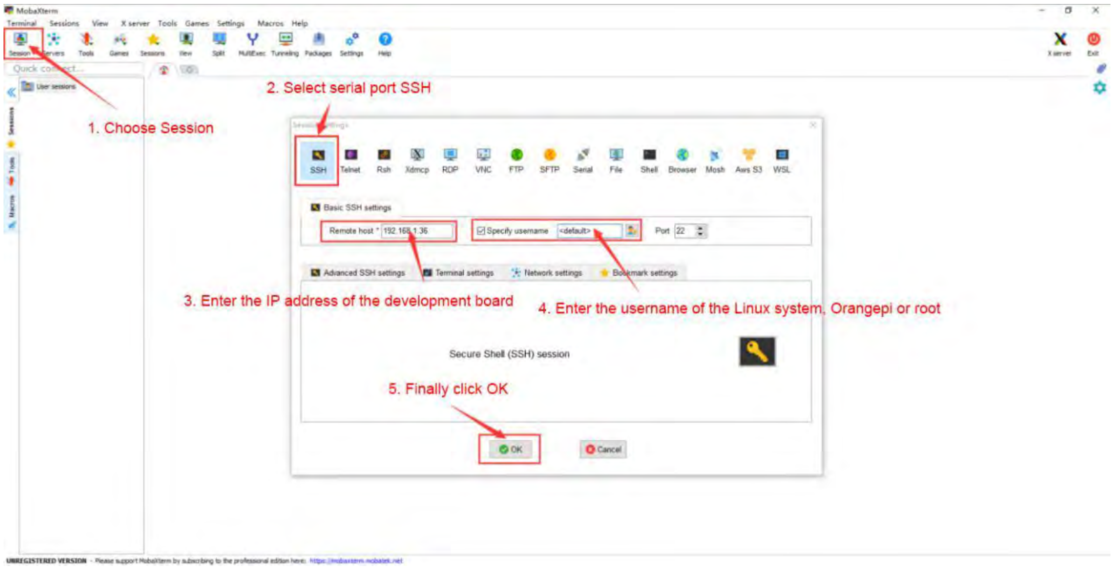
a. Open **Session**

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

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

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

e. Finally click **OK**



1. Choose Session

2. Select serial port SSH

3. Enter the IP address of the development board

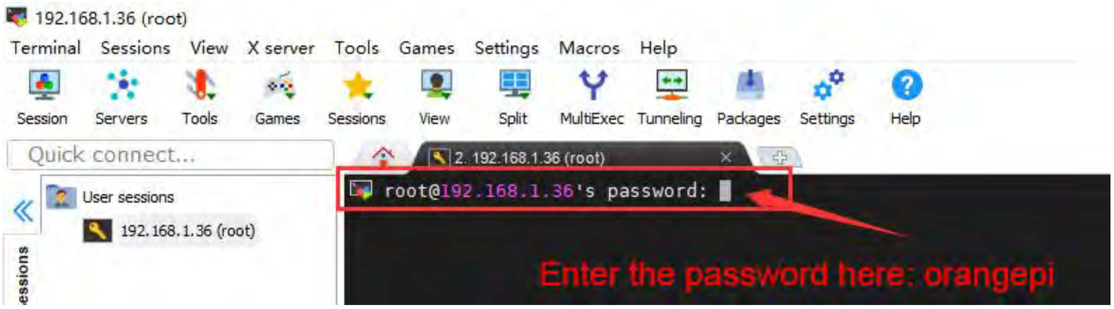
4. Enter the username of the Linux system, Orangepi or root

5. Finally click OK

Remote host * 192.168.1.36
☑ Specify username <default>
Port 22
Secure Shell (SSH) session
OK
Cancel

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

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

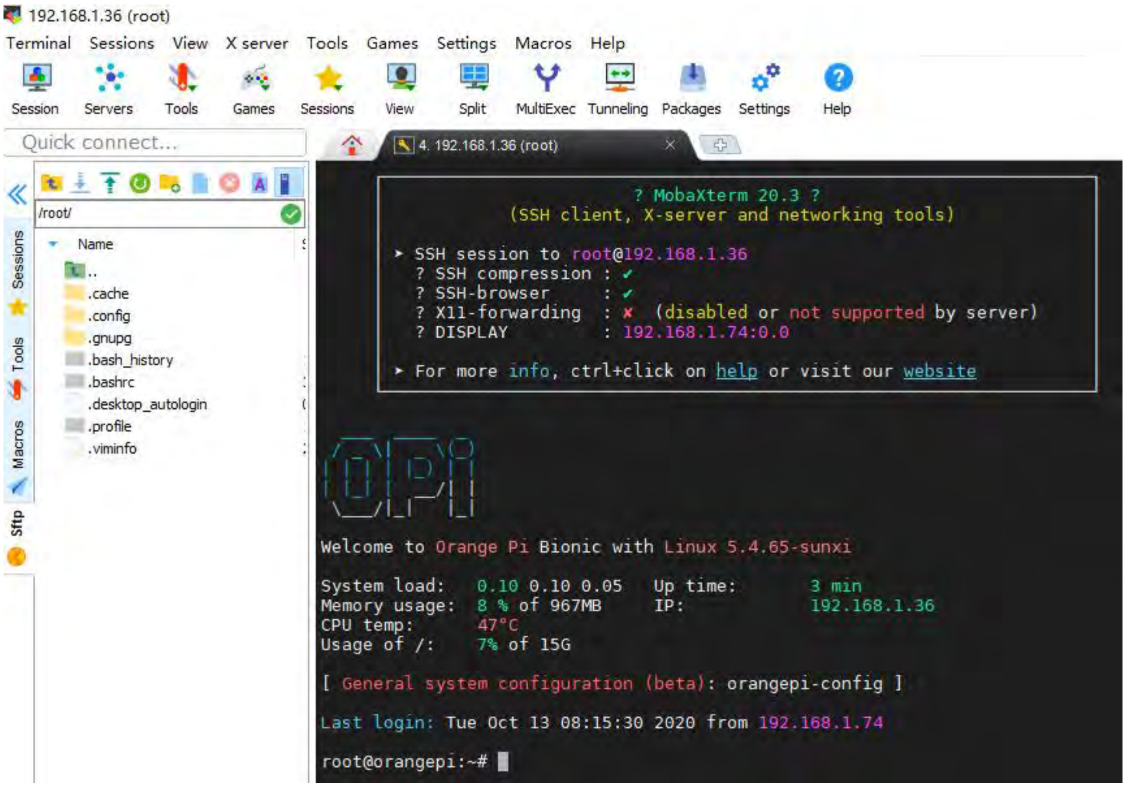


```
root@192.168.1.36's password:
```

Enter the password here: orangepi

192.168.1.36 (root)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 192.168.1.36 (root)
User sessions
192.168.1.36 (root)

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



```
              ? MobaXterm 20.3 ?
    (SSH client, X-server and networking tools)

 ➤ SSH session to root@192.168.1.36
   ? SSH compression : ✔
   ? SSH-browser     : ✔
   ? X11-forwarding  : ✘  (disabled or not supported by server)
   ? DISPLAY         : 192.168.1.74:0.0

 ➤ For more info, ctrl+click on help or visit our website

Welcome to Orange Pi Bionic with Linux 5.4.65-sunxi

System load:   0.10 0.10 0.05   Up time:       3 min
Memory usage:  8 % of 967MB     IP:            192.168.1.36
CPU temp:      47°C
Usage of /:    7% of 15G

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

Last login: Tue Oct 13 08:15:30 2020 from 192.168.1.74

root@orangepi:~#
```

3. 9. HDMI test

3. 9. 1. HDMI display test

1) Use a Micro HDMI to HDMI cable to connect the OrangePi development board and HDMI display

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

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

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

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

3. 9. 2. HDMI to VGA display test

1) First you need to prepare the following accessories

a. HDMI to VGA converter

b. A VGA cable and a Micro HDMI male to HDMI female conversion cable

c. A monitor or TV that supports VGA interface

2) HDMI to VGA display test as shown below

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

3. 9. 3. How to set HDMI resolution in Linux5.4 system

Note: This method is only applicable to systems with linux5.4 kernel.

1) There is a disp_mode variable in **/boot/orangepiEnv.txt** of the linux system, which can be used to set the resolution of the HDMI output. The default resolution of the linux system is 1080p60

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
console=both
disp_mode=1080p60
fb0_width=1920
fb0_height=1080
```

2) The values supported by the disp_mode variable are shown in the table below

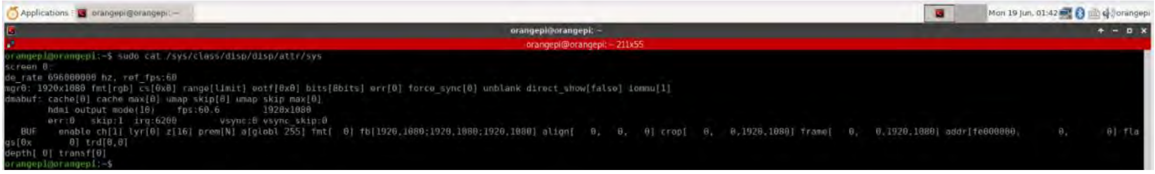
Supported values for disp_mode	HDMI resolution	HDMI refresh rate
480i	720x480	60
576i	720x480	50
480p	720x480	60
576p	720x576	60
720p50	1280x720	50
720p60	1280x720	60
1080i50	1920x1080	50
1080i60	1920x1080	60
1080p24	1920x1080	24
1080p50	1920x1080	50
1080p60	1920x1080	60

Note: Linux systems currently do not support 4K resolution.

3) Change the value of the disp_mode variable to the desired output resolution, and then restart the system, HDMI will output the set resolution

4) The method of viewing the HDMI output resolution is as follows. If the displayed resolution is the same as the set resolution, it means that the setting of the development board is correct.

```
orangepi@orangepi:~$ sudo cat /sys/class/disp/disp/attr/sys
```



```
orangepi@orangepi:~$ sudo cat /sys/class/disp/disp/attr/sys
screen 0:
de_rate 696000000 hz, ref_fps:60
mgr0: 1920x1080 fmt[rgb] cs[0x0] range[limit] eotf[0x0] bits[8bits] err[0] force_sync[0] unblank direct_show[false] iommu[1]
dmabuf: cache[0] cache max[0] umap skip[0] umap skip max[0]
        hdmi output mode(10)    fps:60.6        1920x1080
        err:0   skip:1  irq:6200        vsync:0 vsync_skip:0
   BUF    enable ch[1] lyr[0] z[16] prem[N] a[globl 255] fmt[  0] fb[1920,1080;1920,1080;1920,1080] align[   0,   0,   0] crop[   0,   0,1920,1080] frame[   0,   0,1920,1080] addr[fe000000,       0,       0] fla
gs[0x       0] trd[0,0]
depth[ 0] transf[0]
orangepi@orangepi:~$
```

3.9.4. How to Modify the Framebuffer Width and Height of Linux5.4 System

Note: This method is only applicable to systems with linux5.4 kernel.

There are two variables fb0_width and fb0_height in the **/boot/orangepiEnv.txt** of the linux system, which can be used to set the width and height of the Framebuffer. The default setting of the linux system is fb0_width=1920 and fb0_height=1080.

```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
console=both
disp_mode=1080p60
fb0_width=1920
fb0_height=1080
```

The **reference** values corresponding to different resolutions of fb0_width and fb0_height are as follows:

HDMI resolution	fb0_width	fb0_height
480p	720	480
576p	720	576
720p	1280	720
1080p	1920	1080

Under the same HDMI resolution, the larger the value of fb0_width and fb0_height, the smaller the text displayed on the screen, and the smaller the value of fb0_width and fb0_height, the larger the text displayed on the screen.

3. 9. 5. Framebuffer Cursor Setting

1) The softcursor used by Framebuffer, the method of setting the cursor to blink or not is as follows

```
root@orangepi:~# echo 1 > /sys/class/graphics/fbcon/cursor_blink                #cursor
blinking
root@orangepi:~# echo 0 > /sys/class/graphics/fbcon/cursor_blink            #cursor not
blinking
```

2) If you need to hide the cursor, you can add **vt.global_cursor_default=0** in the **extraargs** variable of **/boot/orangepiEnv.txt** (the value of extraargs will be assigned to the **bootargs** environment variable and finally passed to the kernel) (if **vt.global_cursor_default=1**, the cursor will be displayed) , and then restart the system to see that the cursor has disappeared

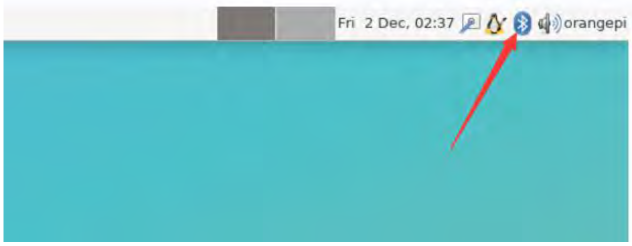
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
verbosity=1
console=both
```

```
disp_mode=1080p60
fb0_width=1920
fb0_height=1080
extraargs=vt.global_cursor_default=0
```

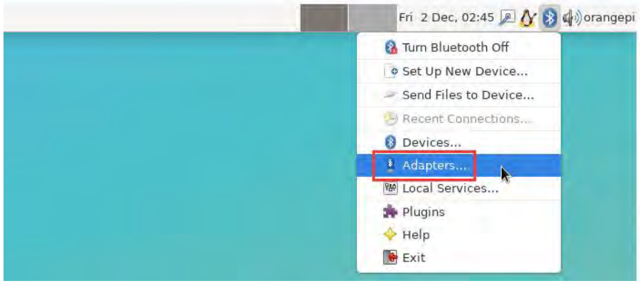
3. 10. How to use Bluetooth

3. 10. 1. Test method of desktop image

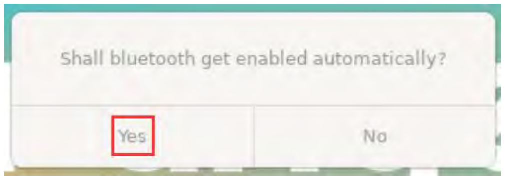
1) Click the Bluetooth icon in the upper right corner of the desktop



2) Then select the adapter

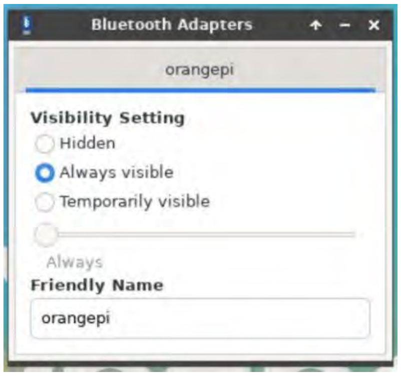


3) If there is a prompt on the following interface, please select **Yes**



Shall bluetooth get enabled automatically?

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



Bluetooth Adapters

orangepi

Visibility Setting

○ Hidden

● Always visible

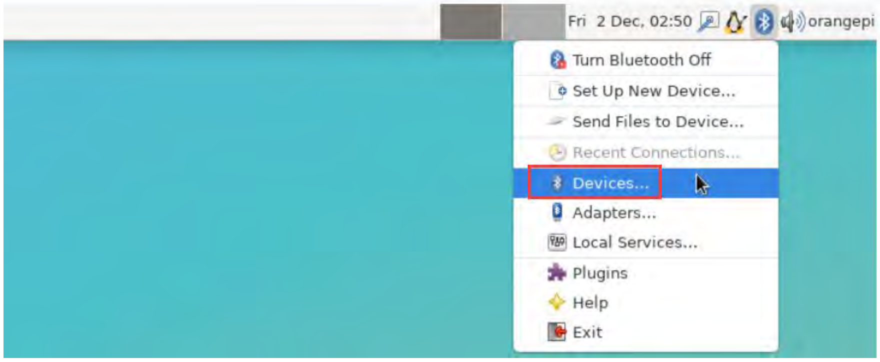
○ Temporarily visible

Always

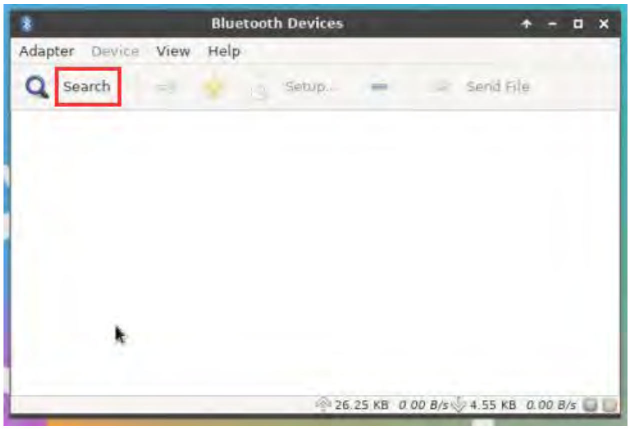
Friendly Name

orangepi

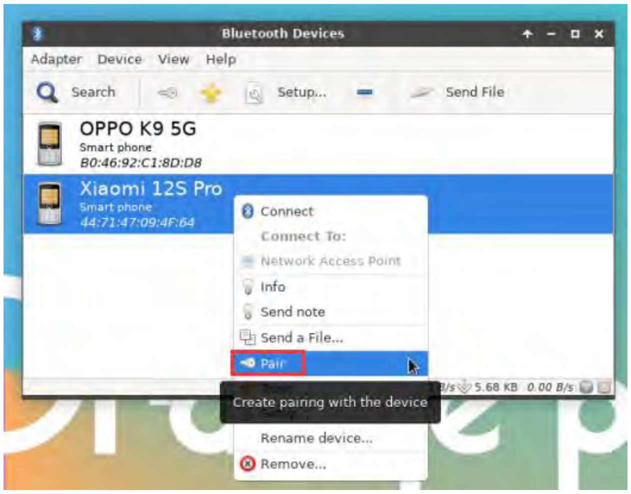
5) Then open the configuration interface of the Bluetooth device



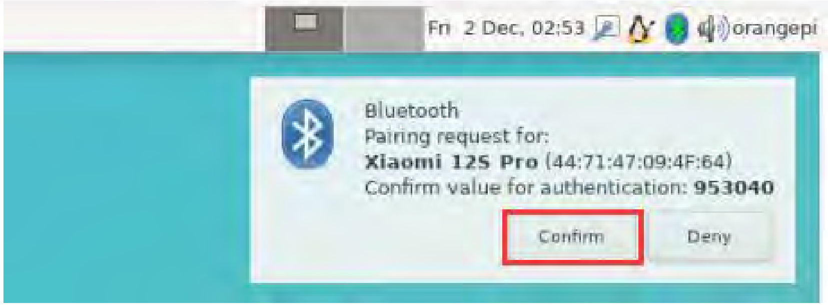
6) Click **Search** to start scanning the surrounding Bluetooth devices



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

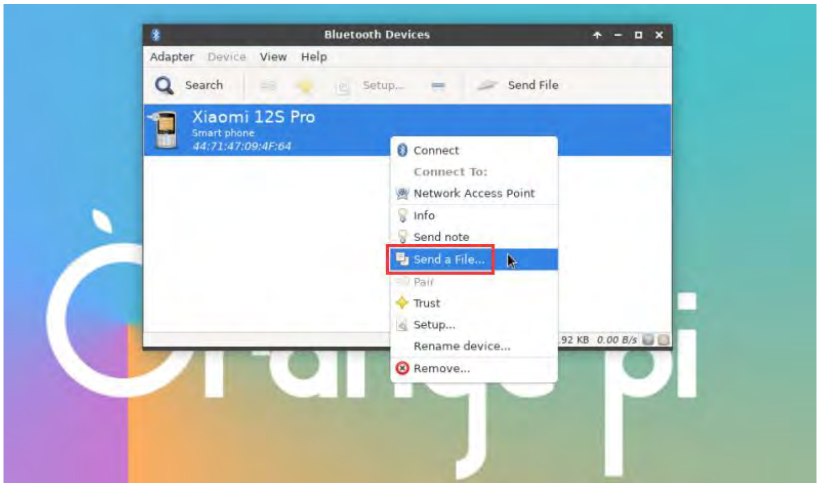


8) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop, just select **Confirm** to confirm, and the phone also needs to confirm at this time

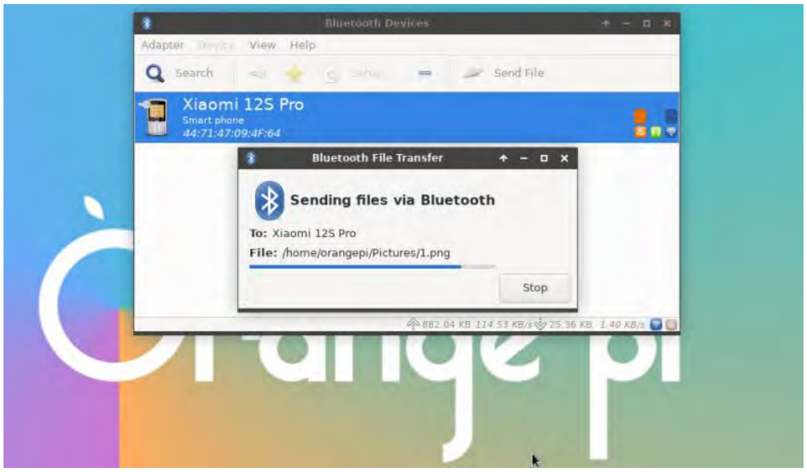


Bluetooth
Pairing request for:
Xiaomi 12S Pro (44:71:47:09:4F:64)
Confirm value for authentication: **953040**

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



10) The interface for sending pictures is as follows



Xiaomi 12S Pro
Smart phone
44:71:47:09:4F:64

Sending files via Bluetooth

To: Xiaomi 12S Pro

File: /home/orangepi/Pictures/1.png

882.04 KB 114.53 KB/s 25.36 KB 1.40 KB/s

3. 10. 2. How to use the server image

1) After entering the system, you can first check whether there is a Bluetooth device node through the **hciconfig** command. If it exists, it means that the Bluetooth initialization is normal.

```
orangepi@orangepi:~$ sudo apt update && sudo apt install -y bluez
orangepi@orangepi:~$ hciconfig -a
hci0:    Type: Primary   Bus: UART
    BD Address: 3E:61:3D:19:0E:52   ACL MTU: 1021:8   SCO MTU: 240:3
    UP RUNNING
    RX bytes:925 acl:0 sco:0 events:72 errors:0
    TX bytes:5498 acl:0 sco:0 commands:72 errors:0
    Features: 0xbf 0xff 0x8d 0xfe 0xdb 0x3d 0x7b 0xc7
    Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
    Link policy: RSWITCH SNIFF
    Link mode: SLAVE ACCEPT
    Name: 'orangepi'
    Class: 0x3c0000
    Service Classes: Rendering, Capturing, Object Transfer, Audio
    Device Class: Miscellaneous,
    HCI Version: 5.0 (0x9)   Revision: 0x400
    LMP Version: 5.0 (0x9)   Subversion: 0x400
    Manufacturer: Spreadtrum Communications Shanghai Ltd (492)
```

2) Use **bluetoothctl** to scan bluetooth devices

```
orangepi@orangepi:~$ sudo bluetoothctl
```

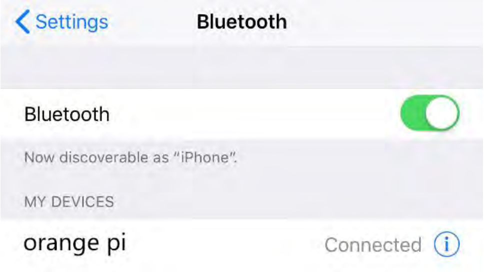
```
[NEW] Controller 10:11:12:13:14:15 orangepizero3 [default]
Agent registered
[bluetooth]# power on           #enable controller
Changing power on succeeded
[bluetooth]# discoverable on      #Make the controller discoverable
Changing discoverable on succeeded
[CHG] Controller 10:11:12:13:14:15 Discoverable: yes
[bluetooth]# pairable on      #Set the controller as 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 MiPhone
[NEW] Device DC:72:9B:4C:F4:CF orangepi
[bluetooth]# scan off            #After scanning to 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 name of the Bluetooth 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
```

3) After scanning to the device you want to pair, you can pair it. You need to use the MAC address of the device for pairing

```
[bluetooth]# pair DC:72:9B:4C:F4:CF        #Use the MAC address of the scanned 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, and you need to confirm on the mobile 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 that the pairing is successful
[CHG] Device DC:72:9B:4C:F4:CF ServicesResolved: no
[CHG] Device DC:72:9B:4C:F4:CF Connected: no
```

4) After the pairing is successful, the Bluetooth interface of the mobile phone will be displayed as follows



5) To connect a Bluetooth device, you need to install the **pulseaudio-module-bluetooth** package, and then start the **pulseaudio** service

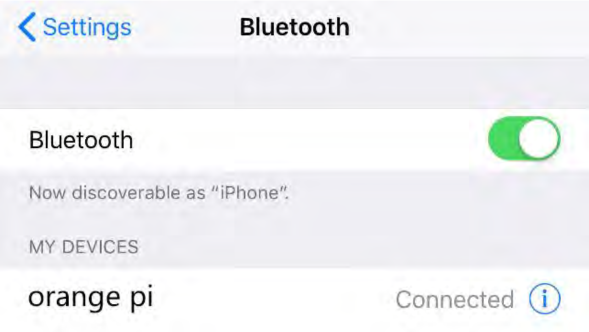
```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt -y install pulseaudio-module-bluetooth
orangepi@orangepi:~$ pulseaudio --start
```

6) How to connect to a Bluetooth device

```
orangepi@orangepi:~$ sudo bluetoothctl
Agent registered
[bluetooth]# paired-devices   #View the MAC address of the paired Bluetooth device
Device DC:72:9B:4C:F4:CF orangepi
[bluetooth]# connect DC:72:9B:4C:F4:CF     #Connect to bluetooth device using MAC address
Attempting to connect to DC:72:9B:4C:F4:CF
[CHG] Device DC:72:9B:4C:F4:CF Connected: yes
Connection successful
[CHG] Device DC:72:9B:4C:F4:CF ServicesResolved: yes
[CHG] Controller 10:11:12:13:14:15 Discoverable: no
[orangepi]#      #This prompt appears to indicate that the connection is successful
```

7) After connecting the bluetooth device, the bluetooth configuration interface of the

Android mobile phone can see **the prompt that the audio has been connected for calls and media**



< Settings Bluetooth

Bluetooth

Now discoverable as "iPhone".

MY DEVICES

orange pi Connected

3. 11. USB interface test

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

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

1) Insert the USB interface keyboard into the USB interface of the OrangePi development board

2) Connect the OrangePi development board to the HDMI display

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

3. 11. 2. Connect USB storage device test

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

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

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

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

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

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

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

3. 11. 3. USB Ethernet card test

1) The usable USB Ethernet cards that **have been tested** so far are as follows. Among them, the RTL8153 USB Gigabit network card can be used normally when it is inserted into the USB 2.0 Host interface of the development board, but the speed cannot reach Gigabit. Please note that

serial number	model
1	RTL8152B USB 100M network card
2	RTL8153 USB Gigabit LAN

2) First insert the USB network card into the USB interface of the development board, and then insert the network cable into the USB network card to ensure that the network cable can access the Internet normally. If you can see the following log information through the **dmesg** command, it means that the USB network card is recognized normally.

```
orangepi@orangepi:~$ dmesg | tail
[   121.985016] usb 3-1: USB disconnect, device number 2
[   126.873772] sunxi-ehci 5311000.ehci3-controller: ehci_irq: highspeed device connect
[   127.094054] usb 3-1: new high-speed USB device number 3 using sunxi-ehci
[   127.357472] usb 3-1: reset high-speed USB device number 3 using sunxi-ehci
[   127.557960] r8152 3-1:1.0 eth1: v1.08.9
[   127.602642] r8152 3-1:1.0 enx00e04c362017: renamed from eth1
[   127.731874] IPv6: ADDRCONF(NETDEV_UP): enx00e04c362017: link is not ready
[   127.763031] IPv6: ADDRCONF(NETDEV_UP): enx00e04c362017: link is not ready
[   129.892465] r8152 3-1:1.0 enx00e04c362017: carrier on
[    129.892583]  IPv6:  ADDRCONF(NETDEV_CHANGE):  enx00e04c362017:  link
becomes ready
```

3) Then you can see the device node of the USB network card and the automatically assigned IP address through the ifconfig command

```
orangepi@orangepi:~$ sudo ifconfig
enx00e04c362017:   flags=4163<UP,BROADCAST,RUNNING,MULTICAST>       mtu
1500
            inet 192.168.1.177    netmask 255.255.255.0    broadcast 192.168.1.255
            inet6 fe80::681f:d293:4bc5:e9fd    prefixlen 64    scopeid 0x20<link>
            ether 00:e0:4c:36:20:17    txqueuelen 1000    (Ethernet)
            RX packets 1849    bytes 134590 (134.5 KB)
            RX errors 0    dropped 125    overruns 0    frame 0
            TX packets 33    bytes 2834 (2.8 KB)
            TX errors 0    dropped 0 overruns 0    carrier 0    collisions 0
```

4) The command to test network connectivity is as follows

```
orangepi@orangepi:~$ ping www.baidu.com -I enx00e04c362017
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. 11. 4. USB camera test

1) First insert the USB camera into the USB port of the OrangePi development board

2) Then through the lsmod command, you can see that the kernel has automatically loaded the following modules

```
orangepi@orangepi:~$ lsmod
```

```
Module                  Size  Used by
uvcvideo              106496  0
```

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

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y v4l-utils
orangepi@orangepi:~$ v4l2-ctl --list-devices
USB 2.0 Camera (usb-sunxi-ehci-1):
    /dev/video0
```

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

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

4) Use fswebcam to test the USB camera

a. Install fswebcam

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

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

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

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

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

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

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

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

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

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

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

5) Use mjpg-streamer to test 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

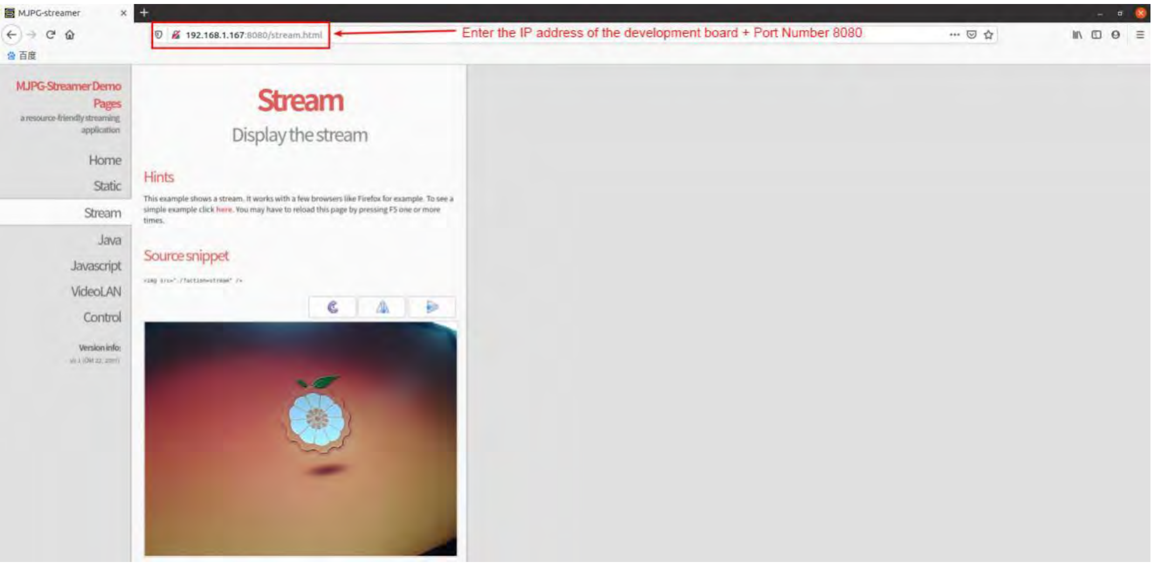
```
orangepi@orangepi:~$ cd mjpg-streamer/mjpg-streamer-experimental
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ make -j4
orangepi@orangepi:~/mjpg-streamer/mjpg-streamer-experimental$ sudo make install
```

d. Then enter the following command to start mjpg_streamer

Note that the serial number of the video is not necessarily video0, 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/video0 -u -f 30" -o "./output_http.so -w ./www"
```

e. Then enter **[the 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



192.168.1.167:8080/stream.html

Enter the IP address of the development board + Port Number 8080

Stream

Display the stream

Hints

This example shows a stream. It works with a few browsers like Firefox for example. To see a simple example click here. You may have to reload this page by pressing F5 one or more times.

Source snippet

```
<img src="./?action=stream" />
```

3. 12. Audio Test

3. 12. 1. How to play audio using the command line

3. 12. 1. 1. Headphone jack playback audio test

1) First, you need to insert the 13pin expansion board into the 13pin interface of the OrangePi development board, and then insert the earphone into the audio interface

2) Through the **aplay -l** command, you can view the sound card devices supported by the Linux system

a. The output of the linux5.4 system is as follows, where **card 0: audiocodec** is the sound card device required for earphone playback

```
root@orangepi:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: audiocodec [audiocodec], device 0: SUNXI-CODEC sun50iw9-codec-0 []
```

```
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

b. The output of the linux6.1 system is as follows, where **audiocodec** is the sound card device required for earphone playback

```
root@orangepi:~# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: audiocodec [audiocodec], device 0: CDC PCM Codec-0 [CDC PCM Codec-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

3) Then use the **aplay** command to play the audio, and the headset can hear the sound

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

If there is noise in the earphone test, please pull out some earphones, do not insert all the earphones.

3. 12. 1. 2. HDMI audio playback test

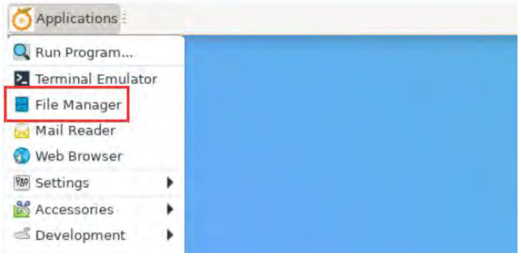
1) First use the Micro HDMI to HDMI cable to connect the OrangePi development board to the TV (other HDMI monitors need to ensure that they can play audio)

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

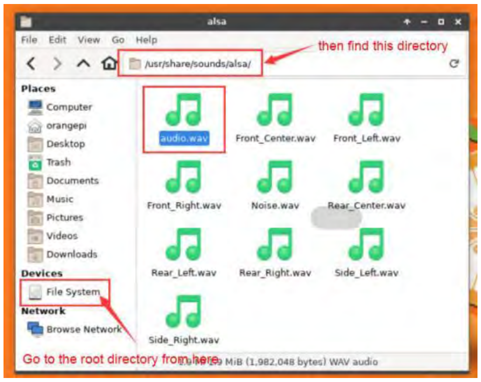
```
root@orangepi:~# aplay -D hw:2,0 /usr/share/sounds/alsa/audio.wav
```

3. 12. 2. Testing audio methods on desktop systems

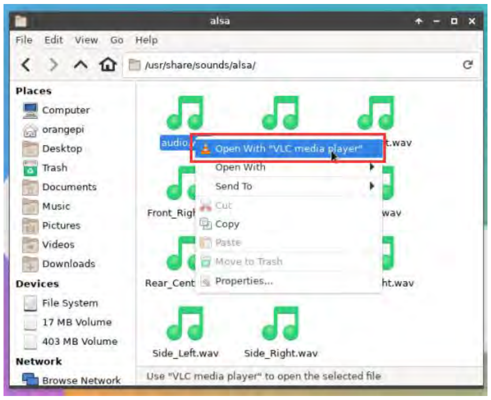
1) First open the file manager



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

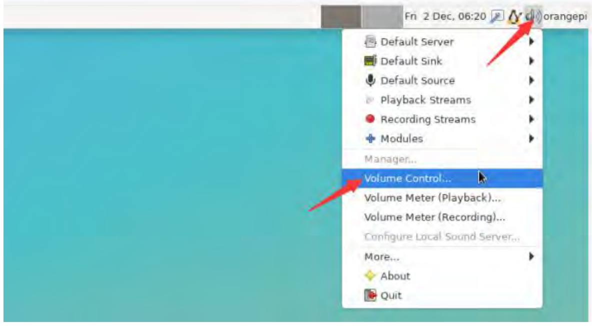


3) Then select the audio.wav file, right click and select open with vlc to start playing

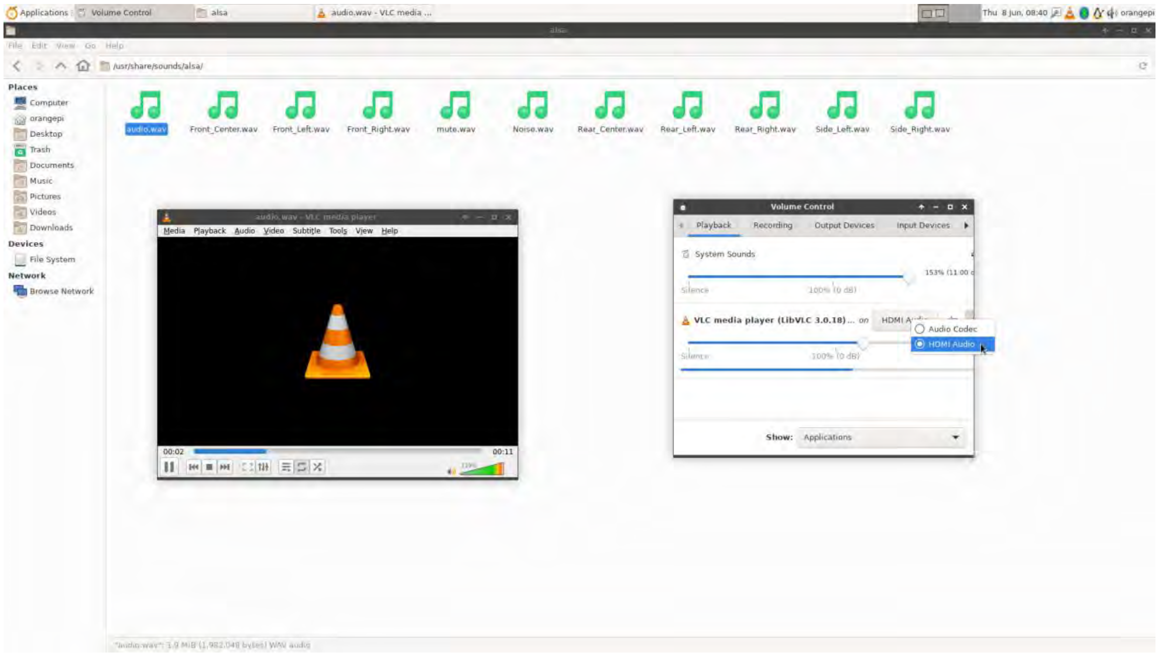


4) How to switch between different audio devices such as HDMI playback and headphone playback

a. First open the volume control interface

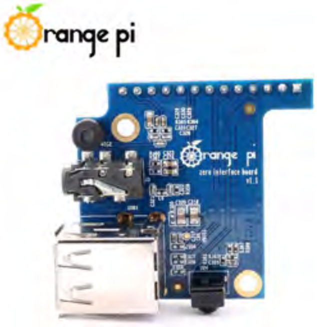


b. When playing audio, the audio device options that the playback software can use will be displayed in **Playback**, as shown in the figure below, where you can set which audio device to play to



3. 13. Infrared receiving test

1) First, you need to insert the 13pin expansion board into the 13pin interface of the OrangePi development board. After inserting the expansion board, the development board can use the infrared receiving function



2) Install ir-keytable infrared testing software

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

3) Then execute ir-keytable to view the information of the infrared device

a. linux5.4 system output is as follows

```
orangepi@orangepi:~$ ir-keytable
Found /sys/class/rc/rc0/ (/dev/input/event1) with:
```

```
            Driver: sunxi-rc-recv, table: rc_map_sunxi
            lirc device: /dev/lirc0
            Supported protocols: lirc nec
            Enabled protocols: lirc nec
            Name: sunxi_ir_recv
            bus: 25, vendor/product: 0001:0001, version: 0x0100
            Repeat delay = 500 ms, repeat period = 125 ms
```

b. The output of the linux6.1 system is as follows

```
orangepi@orangepi:~$ ir-keytable
Found /sys/class/rc/rc0/ with:
     Name: sunxi-ir
     Driver: sunxi-ir
     Default keymap: rc-empty
     Input device: /dev/input/event5
     LIRC device: /dev/lirc0
     Attached BPF protocols: Operation not permitted
     Supported kernel protocols: lirc rc-5 rc-5-sz jvc sony nec sanyo mce_kbd rc-6 sharp
xmp imon rc-mm
     Enabled kernel protocols: lirc
     bus: 25, vendor/product: 0001:0001, version: 0x0100
     Repeat delay = 500 ms, repeat period = 125 ms
```

4) Before testing the infrared receiving function, you need to prepare an infrared remote controller dedicated to Orange Pi, **other remote controllers do not support it**

5) Then enter the **ir-keytable -t** command in the terminal, and then use the infrared remote control to press the button against the infrared receiver of the OrangePi development board to see the received key code in the terminal

a. linux5.4 system output is as follows

```
orangepi@orangepi:~$ sudo ir-keytable -t
Testing events. Please, press CTRL-C to abort.
1598339152.260376: event type EV_MSC(0x04): scancode = 0xfb0413
1598339152.260376: event type EV_SYN(0x00).
1598339152.914715: event type EV_MSC(0x04): scancode = 0xfb0410
```

b. linux6.1 system output is as follows

```
orangepi@orangepi:~$ sudo ir-keytable -c -p NEC -t
Old keytable cleared
Protocols changed to nec
Testing events. Please, press CTRL-C to abort.
202.063219: lirc protocol(nec): scancode = 0x45c
202.063249: event type EV_MSC(0x04): scancode = 0x45c
202.063249: event type EV_SYN(0x00).
```

3. 14. Temperature sensor

1) H618 has a total of 4 temperature sensors, the command to view the temperature is as follows:

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

a. sensor0: CPU temperature sensor, the first command is used to view the type of temperature sensor, and the second command is used to view the value of the temperature sensor

```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone0/type
cpu_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone0/temp
57734
```

b. sensor1: DDR temperature sensor, the first command is used to view the type of temperature sensor, and the second command is used to view the value of the temperature sensor

```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone1/type
ddr_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone1/temp
57410
```

c. sensor2: GPU temperature sensor, the first command is used to view the type of temperature sensor, the second command is used to view the value of the temperature sensor

```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone2/type
gpu_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone2/temp
59273
```

d. sensor3: VE temperature sensor, the first command is used to view the type of temperature sensor, and the second command is used to view the value of the temperature sensor

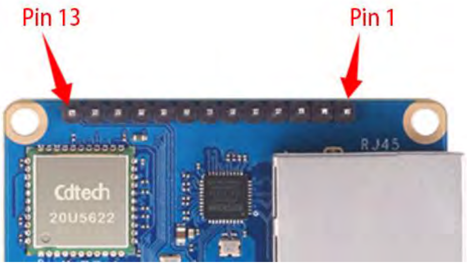
```
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone3/type
ve_thermal_zone
orangepi@orangepi:~$ cat /sys/class/thermal/thermal_zone3/temp
58949
```

2) The system with Linux6.1 kernel can also use the **sensors** command to view the current temperature of the system

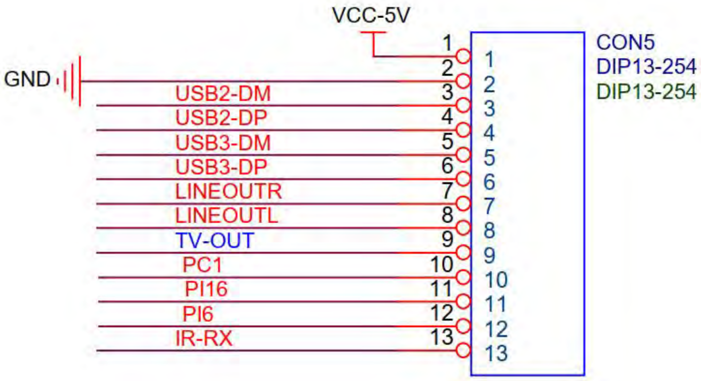
```
orangepi@orangepi:~$ sensors
cpu_thermal-virtual-0
Adapter: Virtual device
temp1:         +47.4°C  (crit = +110.0°C)

gpu_thermal-virtual-0
Adapter: Virtual device
temp1:         +48.7°C  (crit = +110.0°C)

ddr_thermal-virtual-0
Adapter: Virtual device
temp1:         +47.8°C  (crit = +110.0°C)

ve_thermal-virtual-0
Adapter: Virtual device
temp1:         +47.2°C  (crit = +110.0°C)
```

3. 15. 13 Pin Description of the Pin Expansion Board Interface

1) Please refer to the figure below for the order of the interface pins of the 13 pin expansion board on the development board



2) The schematic diagram of the 13pin interface of the development board is as follows



3) The function description of the 13 pin expansion board interface pins of the development board is as follows

a. When the 13pin pin is connected to the expansion board, it can be provided additionally

   a) 2 个 USB 2.0 Host

   b) Audio output of the left and right audio channels of the earphones

   c) TV-OUT video output

   d) Infrared receiving function

   e) After the expansion board is connected, pins 10, 11 and 12 of the 13pin interface cannot be used

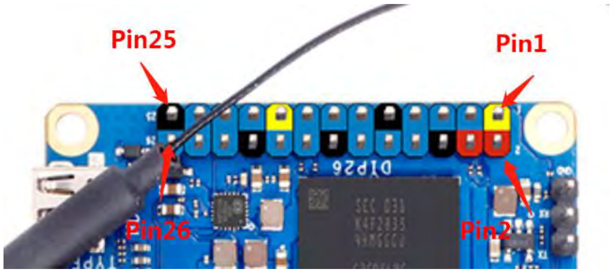
   f) **In addition, it should be noted that the MIC on the 13pin expansion board cannot be used on the OrangePi Zero 3**

b. When pin 13 is not connected to the expansion board, pins 10, 11, 12 and 13 can be used as ordinary GPIO ports

GPIO serial number	Function	Pin
	5V	1
	GND	2
	USB2-DM	3
	USB2-DP	4
	USB3-DM	5
	USB3-DP	6
	LINEOUTR	7
	LINEOUTL	8
	TV-OUT	9
65	PC1	10
272	PI16	11
262	PI6	12
234	IR-RX/PH10	13

3.16. 26 Pin Interface Pin Description

1) For the order of the 26-pin interface pins on the OrangePi Zero 3 development board, please refer to the silkscreen diagram on the development board



2) The functions of the 26 pin interface pins on the development board are shown in the table below

GPIO S/N	GPIO	Function	Pin	Pin	Function	GPIO	GPIO S/N
		3.3V	1	2	5V		
229	PH5	TWI3-SDA	3	4	5V		
228	PH4	TWI3-SCK	5	6	GND		

73	PC9	PC9	7	8	UART5_TX	PH2	226
		GND	9	10	UART5_RX	PH3	227
70	PC6	PC6	11	12	PC11	PC11	75
69	PC5	PC5	13	14	GND		
72	PC8	PC8	15	16	PC15	PC15	79
		3.3V	17	18	PC14	PC14	78
231	PH7	SPI1_MOSI	19	20	GND		
232	PH8	SPI1_MISO	21	22	PC7	PC7	71
230	PH6	SPI1_CLK	23	24	SPI1_CS	PH9	233
		GND	25	26	PC10	PC10	74

3) 3) There are a total of 17 GPIO ports in the 26pin interface, and the voltage of all GPIO ports is **3.3v**

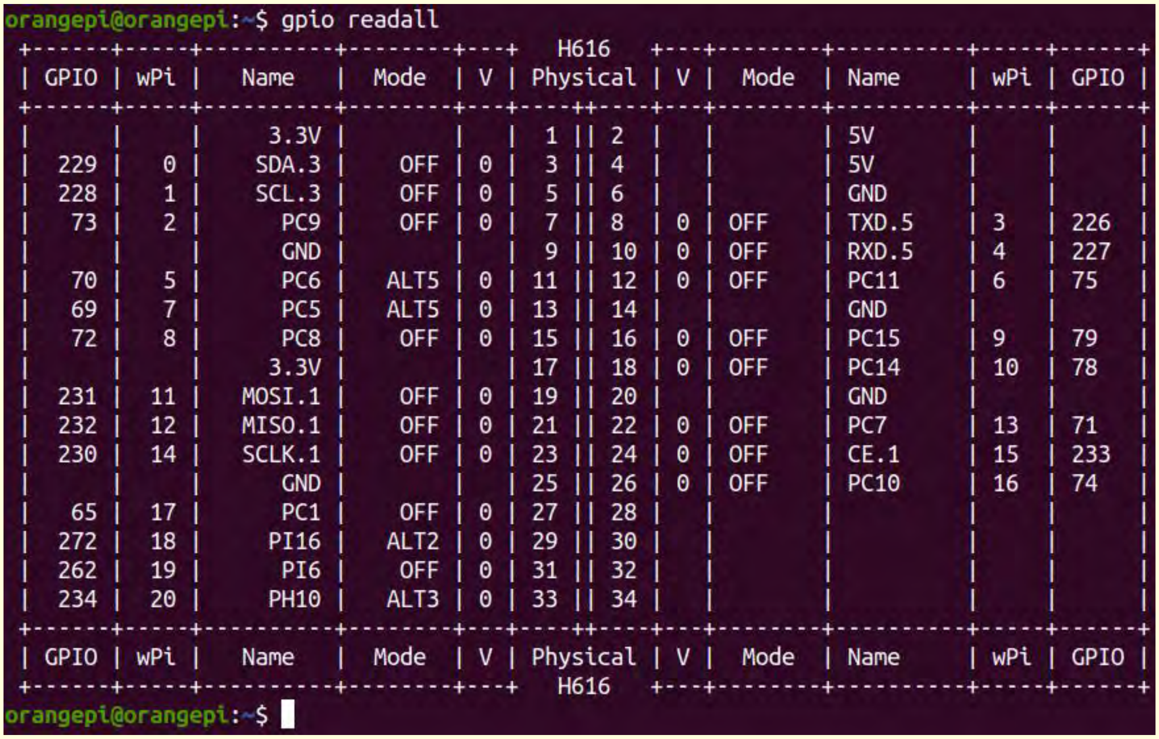
3. 17. How to install wiringOP

Note that wiringOP has been pre-installed in the linux image released by Orange Pi. Unless the code of wiringOP is updated, there is no need to re-download, compile and install, just use it directly.

The storage path of the compiled wiringOP deb package in orangepi-build is:

orangepi-build/external/cache/debs/arm64/wiringpi_x.xx.deb

After entering the system, you can run the gpio readall command. If you can see the following output, it means that wiringOP has been pre-installed and can be used normally.



```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
 |   70 |   5 |      PC6 |   ALT5 | 0 | 11 || 12 | 0 | OFF    | PC11     | 6   | 75   |
 |   69 |   7 |      PC5 |   ALT5 | 0 | 13 || 14 |   |        | GND      |     |      |
 |   72 |   8 |      PC8 |    OFF | 0 | 15 || 16 | 0 | OFF    | PC15     | 9   | 79   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PC14     | 10  | 78   |
 |  231 |  11 |   MOSI.1 |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  232 |  12 |   MISO.1 |    OFF | 0 | 21 || 22 | 0 | OFF    | PC7      | 13  | 71   |
 |  230 |  14 |   SCLK.1 |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 233  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | PC10     | 16  | 74   |
 |   65 |  17 |      PC1 |    OFF | 0 | 27 || 28 |   |        |          |     |      |
 |  272 |  18 |     PI16 |   ALT2 | 0 | 29 || 30 |   |        |          |     |      |
 |  262 |  19 |      PI6 |    OFF | 0 | 31 || 32 |   |        |          |     |      |
 |  234 |  20 |     PH10 |   ALT3 | 0 | 33 || 34 |   |        |          |     |      |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
orangepi@orangepi:~$
```

1) Download the code of wiringOP

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y git
orangepi@orangepi:~$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

Note that the source code needs to download the code of the wiringOP next branch, please don't miss the -b next parameter.

If you have problems downloading the code from GitHub, you can directly use the wiringOP source code that comes with the Linux image, and the storage location is: /usr/src/wiringOP.

2) Compile and install wiringOP

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

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

a. There is a one-to-one correspondence between pins 1 to 26 and 26 Pins on the development board

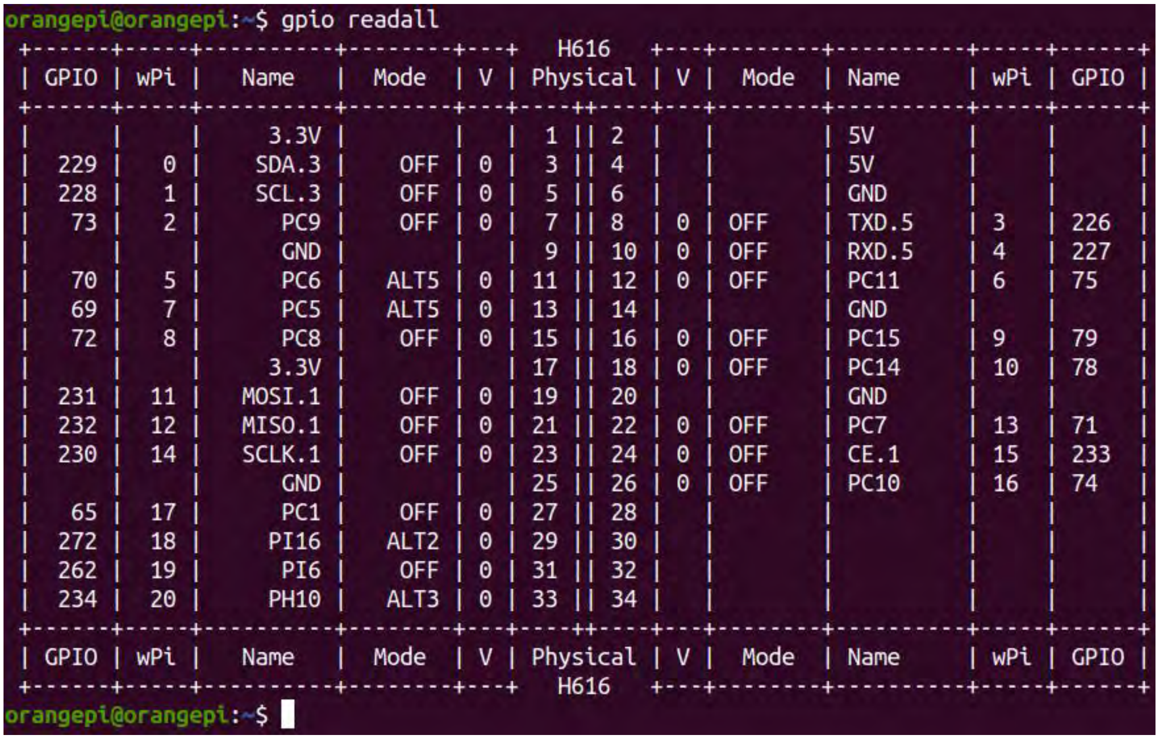
b. Pin 27 corresponds to pin 10 of 13pin on the development board

c. Pin 29 corresponds to pin 11 of 13pin on the development board

d. Pin 31 corresponds to pin 12 of 13pin on the development board

e. Pin 33 corresponds to pin 13 of 13pin on the development board

f. Pins 28, 30, 32, and 34 are empty, please ignore them

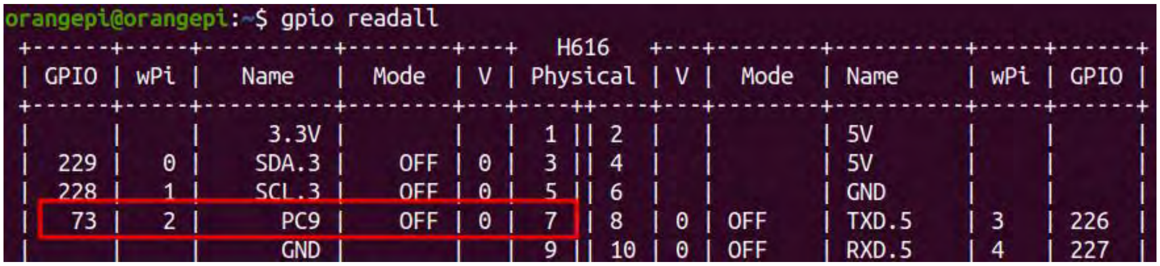


```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
 |   70 |   5 |      PC6 |   ALT5 | 0 | 11 || 12 | 0 | OFF    | PC11     | 6   | 75   |
 |   69 |   7 |      PC5 |   ALT5 | 0 | 13 || 14 |   |        | GND      |     |      |
 |   72 |   8 |      PC8 |    OFF | 0 | 15 || 16 | 0 | OFF    | PC15     | 9   | 79   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PC14     | 10  | 78   |
 |  231 |  11 |   MOSI.1 |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  232 |  12 |   MISO.1 |    OFF | 0 | 21 || 22 | 0 | OFF    | PC7      | 13  | 71   |
 |  230 |  14 |   SCLK.1 |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 233  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | PC10     | 16  | 74   |
 |   65 |  17 |      PC1 |    OFF | 0 | 27 || 28 |   |        |          |     |      |
 |  272 |  18 |     PI16 |   ALT2 | 0 | 29 || 30 |   |        |          |     |      |
 |  262 |  19 |      PI6 |    OFF | 0 | 31 || 32 |   |        |          |     |      |
 |  234 |  20 |     PH10 |   ALT3 | 0 | 33 || 34 |   |        |          |     |      |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
orangepi@orangepi:~$
```

3. 18. 26pin interface GPIO, I2C, UART, SPI and PWM test

3. 18. 1. 26pin GPIO port test

1) The following takes pin 7—corresponding to GPIO as PC9—corresponding to wPi number 2—as an example to demonstrate how to set the high and low levels of the GPIO port



```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

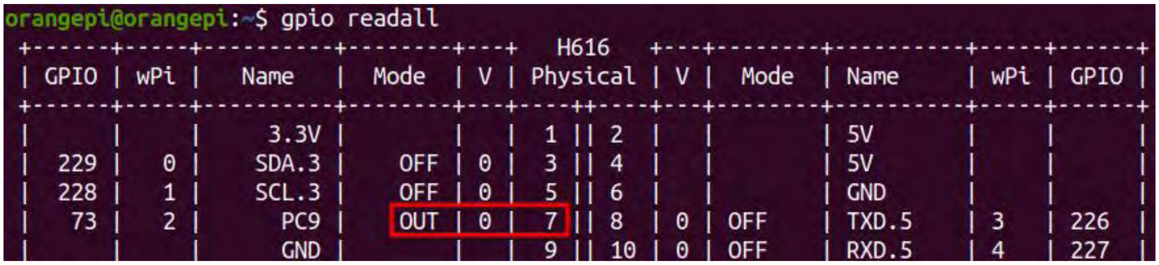
2) First set the GPIO port to output mode, where the third parameter requires the serial number of the wPi corresponding to the input pin

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

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

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

Use gpio readall to see that the value (V) of pin 7 has changed to 0

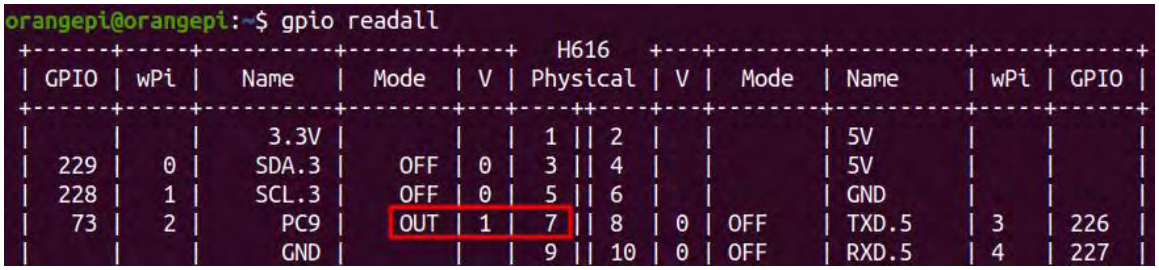


```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OUT | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

4) Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is set successfully.

```
root@orangepi:~/wiringOP# gpio write 2 1
```

Use gpio readall to see that the value (V) of pin 7 has changed to 1

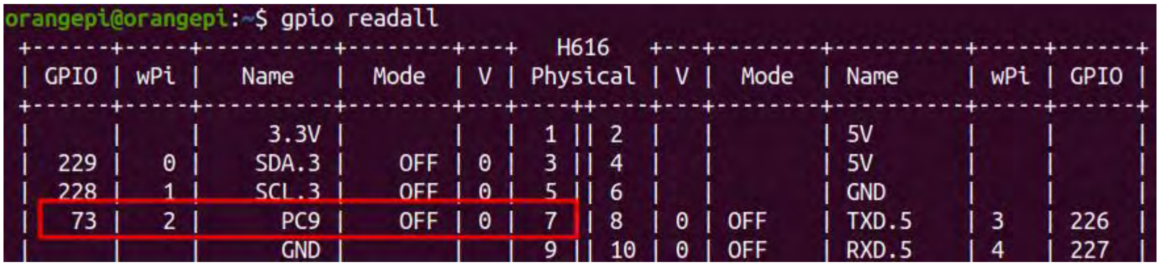


```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OUT | 1 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

5) The setting method of other pins is similar, just modify the serial number of wPi to the corresponding serial number of the pin

3. 18. 2. How to set pull-up and pull-down resistors on 26 pin GPIO ports

1) The following takes pin 7—the corresponding GPIO is PC9—the corresponding wPi number is 2 — as an example to demonstrate how to set the pull-up and pull-down resistors of the GPIO port



```
orangepi@orangepi:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
```

2) First, you need to set the GPIO port to the input mode, and the third parameter needs to be the serial number of the wPi corresponding to the input pin

```
root@orangepi:~/wiringOP# gpio mode 2 in
```

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

```
root@orangepi:~/wiringOP# gpio mode 2 up
```

4) Then enter the following command to read the level of the GPIO port, if the level is 1, it means that the pull-up mode is set successfully

```
root@orangepi:~/wiringOP# gpio read 2
1
```

5) Then execute the following command to set the GPIO port to pull-down mode

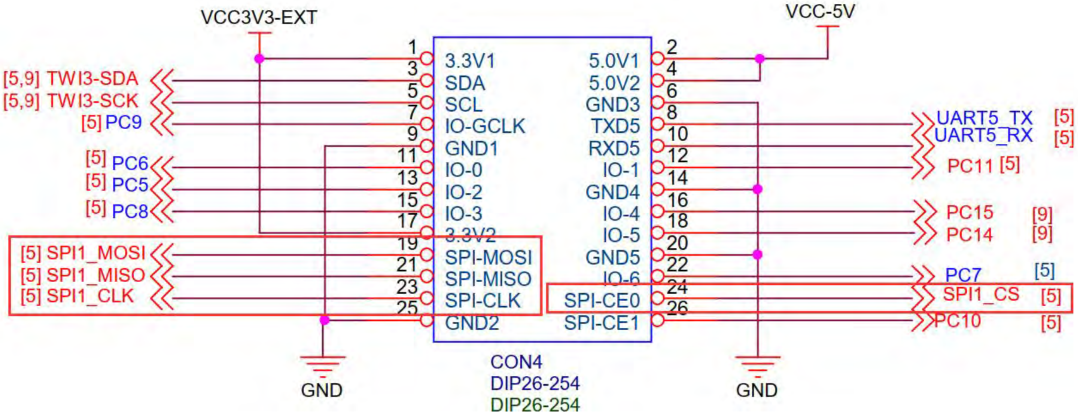
```
root@orangepi:~/wiringOP# gpio mode 2 down
```

6) Then enter the following command to read the level of the GPIO port, if the level is 0, the pull-down mode is set successfully

```
root@orangepi:~/wiringOP# gpio read 2
0
```

3. 18. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the available spi is spi1

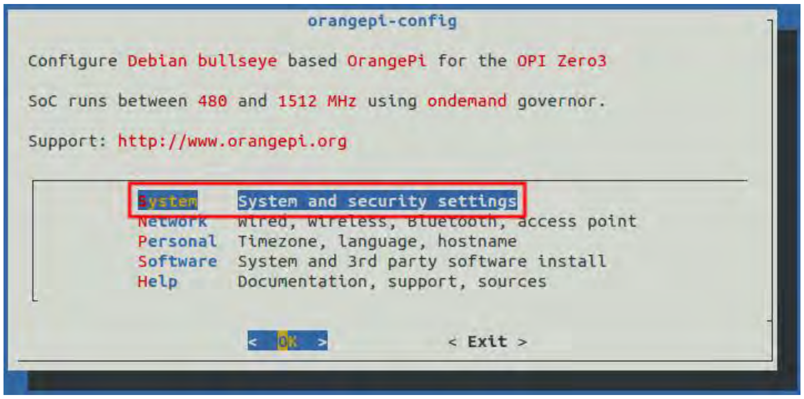


2) The spi1 is disabled by default in the Linux system and needs to be manually enabled before it can be used. The steps to open are as follows:

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

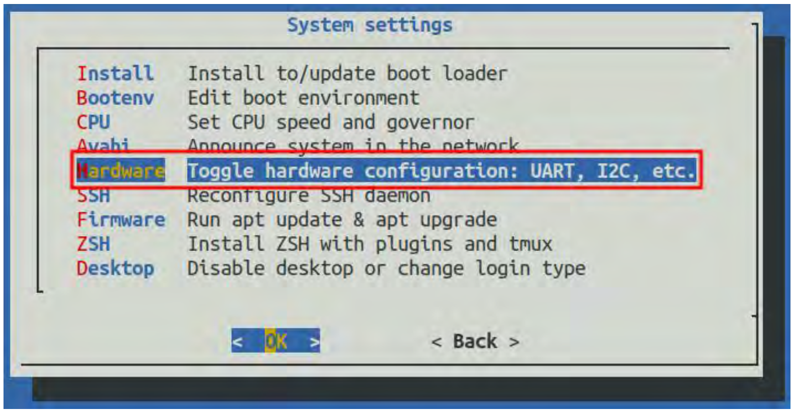
```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



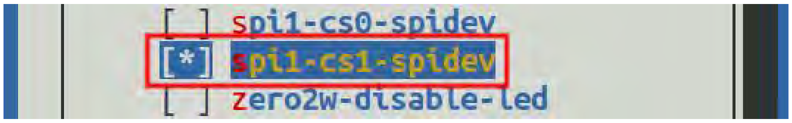
```
orangepi-config

Configure Debian bullseye based OrangePi for the OPI Zero3

SoC runs between 480 and 1512 MHz using ondemand governor.

Support: http://www.orangepi.org

System      System and security settings
Network     Wired, wireless, Bluetooth, access point
Personal    Timezone, language, hostname
Software    System and 3rd party software install
Help        Documentation, support, sources

< OK >                  < Exit >
```

c. Then select **Hardware**



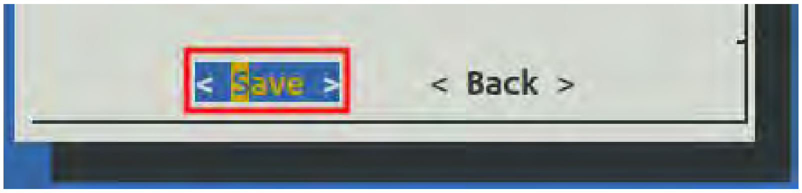
```
                System settings

 Install   Install to/update boot loader
 Bootenv   Edit boot environment
 CPU       Set CPU speed and governor
 Avahi     Announce system in the network
 Hardware  Toggle hardware configuration: UART, I2C, etc.
 SSH       Reconfigure SSH daemon
 Firmware  Run apt update & apt upgrade
 ZSH       Install ZSH with plugins and tmux
 Desktop   Disable desktop or change login type

           < OK >              < Back >
```

d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **spi1-cs1-spidev**

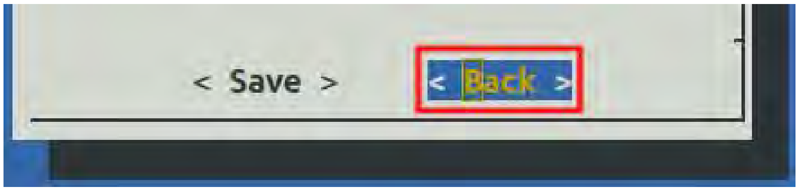


```
[ ] spi1-cs0-spidev
[*] spi1-cs1-spidev
[ ] zero2w-disable-led
```

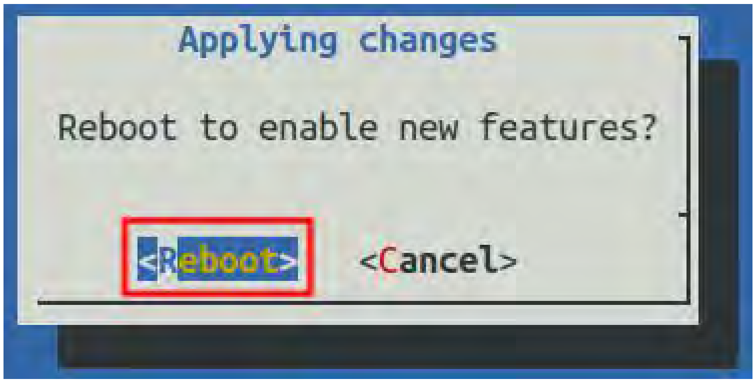
e. Then select **<Save>** to save



f. Then select **<Back>**



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



```
Applying changes

Reboot to enable new features?

<Reboot>      <Cancel>
```

3) Then check whether there is a **spidev1.1** device node in the Linux system. If it exists, it means that the configuration of SPI1 has taken effect

```
orangepi@orangepi:~$ ls /dev/spidev1*
/dev/spidev1.1
```

4) Do not short-circuit the mosi and miso 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

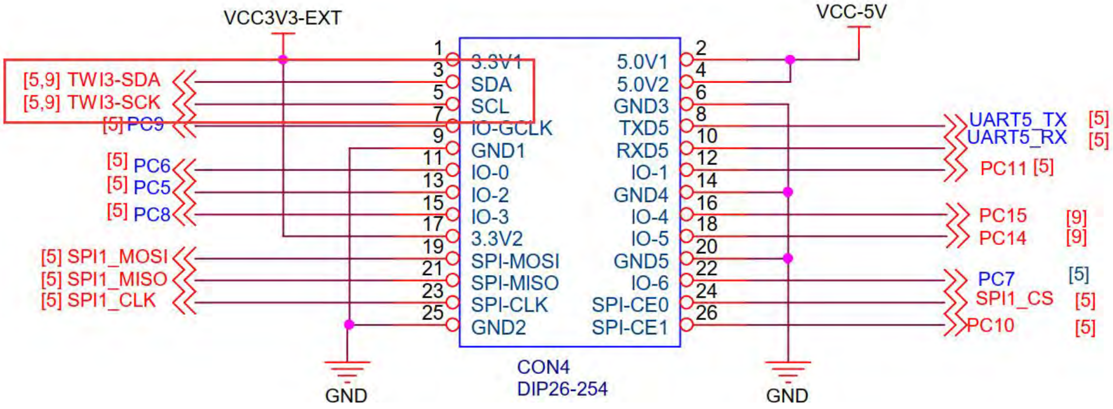
```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF   | ................................
```

5) Then short the two pins of mosi (the 19th pin in the 26pin interface) and miso (the 21st pin in the 26pin interface) of SPI1, and then run the output of spidev_test as follows, you can see the sending and receiving same data

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.1
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
```

3. 18. 4. 26pin I2C test

1) According to the schematic diagram of 26pin, the available i2c is i2c3

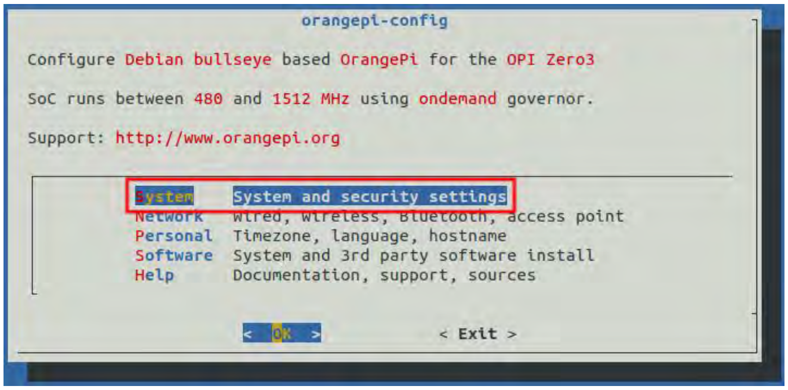


2) i2c3 is disabled by default in the Linux system, and it needs to be manually enabled before it can be used. The steps to open are as follows:

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

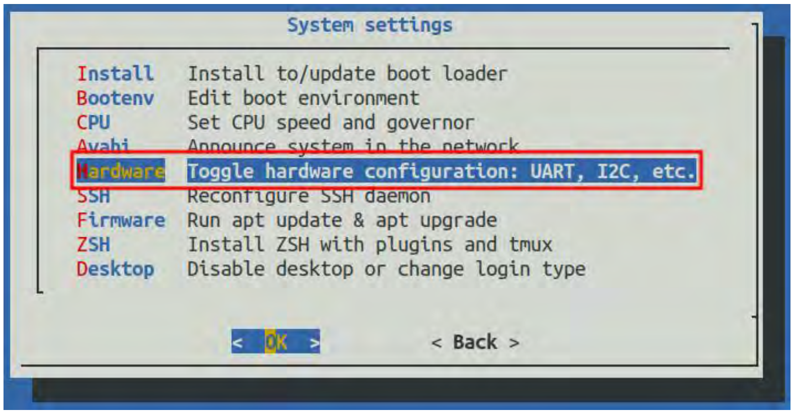
```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



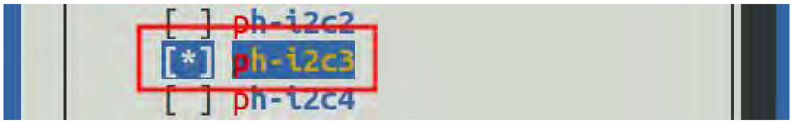
```
orangepi-config

Configure Debian bullseye based OrangePi for the OPI Zero3

SoC runs between 480 and 1512 MHz using ondemand governor.

Support: http://www.orangepi.org

     System     System and security settings
     Network    Wired, Wireless, Bluetooth, access point
     Personal   Timezone, language, hostname
     Software   System and 3rd party software install
     Help       Documentation, support, sources

          < OK >                    < Exit >
```

c. Then select **Hardware**



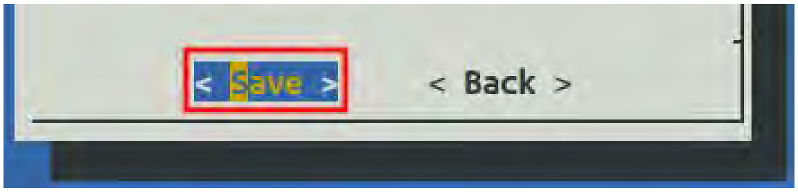
```
System settings

  Install    Install to/update boot loader
  Bootenv    Edit boot environment
  CPU        Set CPU speed and governor
  Avahi      Announce system in the network
  Hardware   Toggle hardware configuration: UART, I2C, etc.
  SSH        Reconfigure SSH daemon
  Firmware   Run apt update & apt upgrade
  ZSH        Install ZSH with plugins and tmux
  Desktop    Disable desktop or change login type

          < OK >                    < Back >
```

d. d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-i2c3**

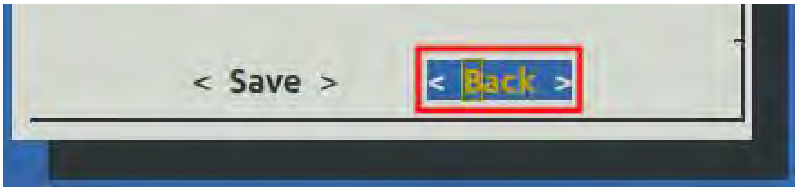


```
[ ] ph-i2c2
[*] ph-i2c3
[ ] ph-i2c4
```

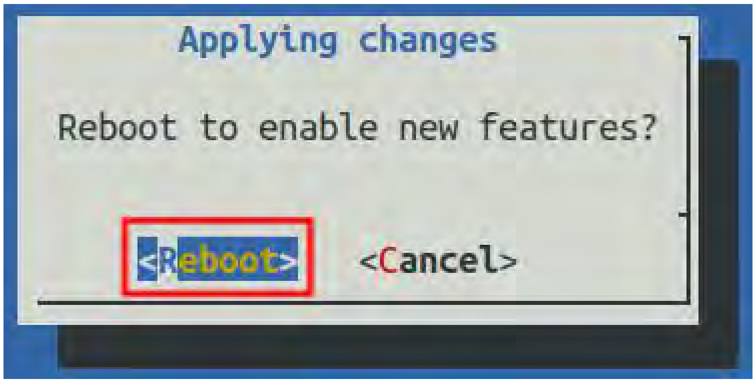
e. Then select **<Save>** to save



f. Then select **<Back>**



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



Applying changes

Reboot to enable new features?

3) After starting the linux system, first confirm that there is an i2c3 device node under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-3
/dev/i2c-3
```

4) Then start testing i2c, first install i2c-tools

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y i2c-tools
```

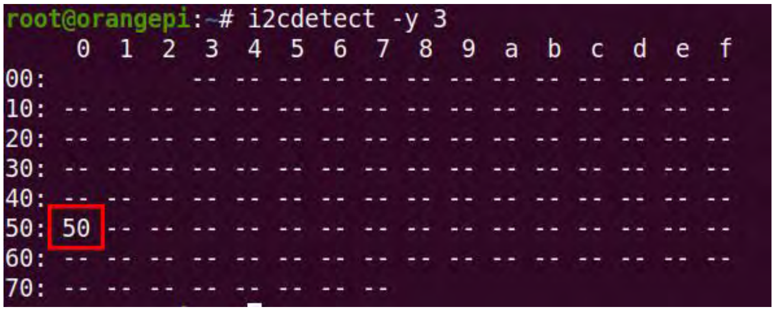
5) Then connect an i2c device to the i2c3 pin of the 26pin connector

Please select the 5V and 3.3V pins according to the specific i2c device, and the voltage value required by different i2c devices may be different.

sda pin	**Corresponding to pin 3 in 26pin**
sck pin	**Corresponding to pin 5 in 26pin**
5v pin	**Corresponding to pin 2 in 26pin**
3.3v pin	**Corresponding to pin 1 in 26pin**
gnd pin	**Corresponding to pin 6 in 26pin**

6) Then use the **i2cdetect -y 3** command, if the address of the connected i2c device can be detected, it means that i2c can be used normally

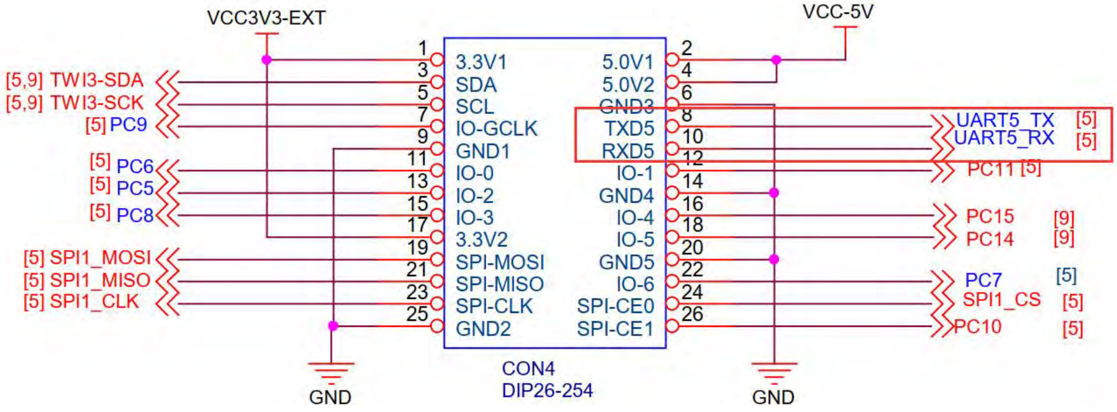
Different i2c device addresses are different, the address 0x50 in the figure below is just an example. Please refer to what you actually see.



```
root@orangepi:~# i2cdetect -y 3
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

3. 18. 5. 26pin UART test

1) According to the schematic diagram of the 26pin interface, the available uart is uart5

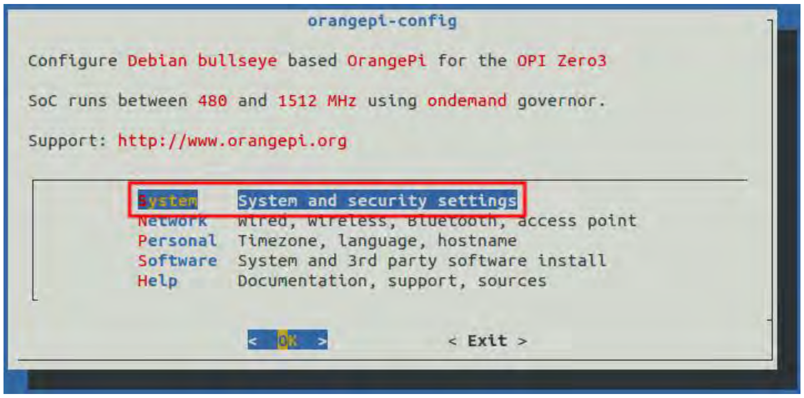


2) Uart5 is disabled by default in the Linux system, and it needs to be opened manually to use it. The steps to open are as follows:

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

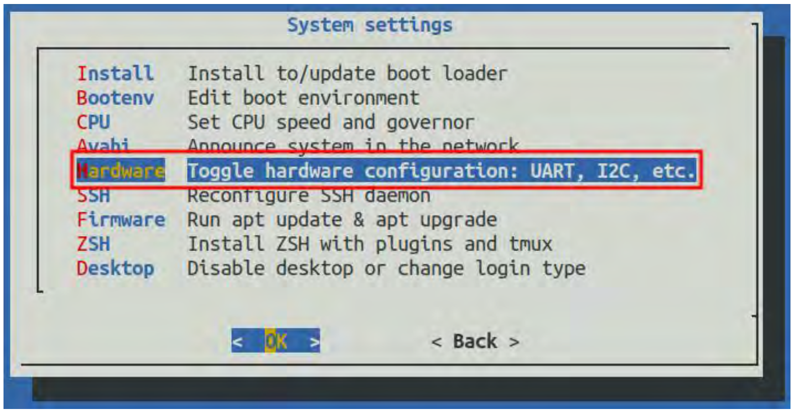
```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



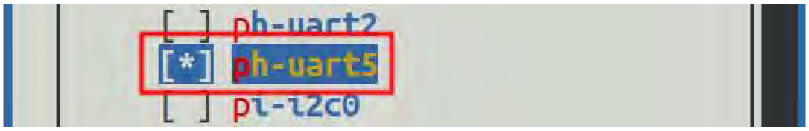
```
orangepi-config

Configure Debian bullseye based OrangePi for the OPI Zero3

SoC runs between 480 and 1512 MHz using ondemand governor.

Support: http://www.orangepi.org

System     System and security settings
Network    Wired, wireless, Bluetooth, access point
Personal   Timezone, language, hostname
Software   System and 3rd party software install
Help       Documentation, support, sources

        < OK >              < Exit >
```

c. Then select **Hardware**



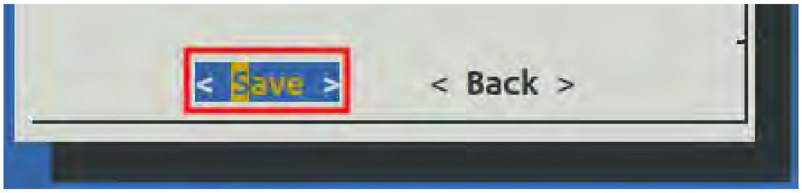
```
System settings

Install    Install to/update boot loader
Bootenv    Edit boot environment
CPU        Set CPU speed and governor
Avahi      Announce system in the network
Hardware   Toggle hardware configuration: UART, I2C, etc.
SSH        Reconfigure SSH daemon
Firmware   Run apt update & apt upgrade
ZSH        Install ZSH with plugins and tmux
Desktop    Disable desktop or change login type

        < OK >              < Back >
```

d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-uart5**



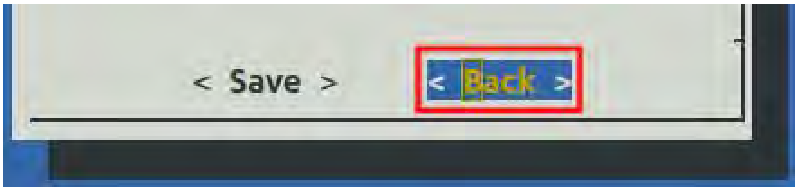
```
[ ] ph-uart2
[*] ph-uart5
[ ] pi-i2c0
```

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



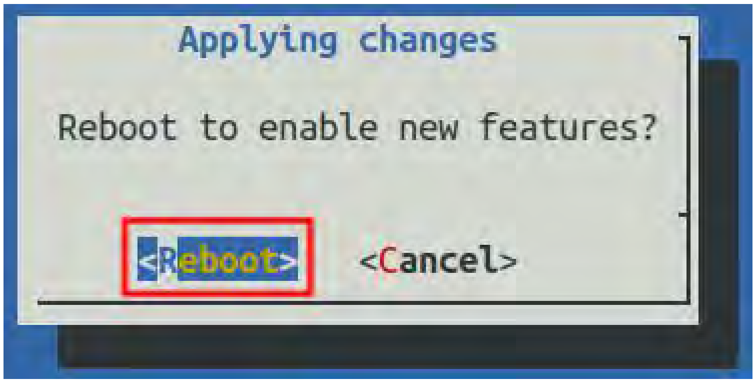
```
< Save >      < Back >
```

f. Then select **<Back>**



```
< Save >      < Back >
```

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



Applying changes

Reboot to enable new features?

3) After entering the linux system, first confirm whether there is a uart5 device node under **/dev**

Note that the linux5.4 system is /dev/ttyAS5.

```
orangepi@orangepi:~$ ls /dev/ttyS5
/dev/ttyS5
```

4) Then start to test the uart5 interface, first use the DuPont line to short the rx and tx of the uart5 interface to be tested

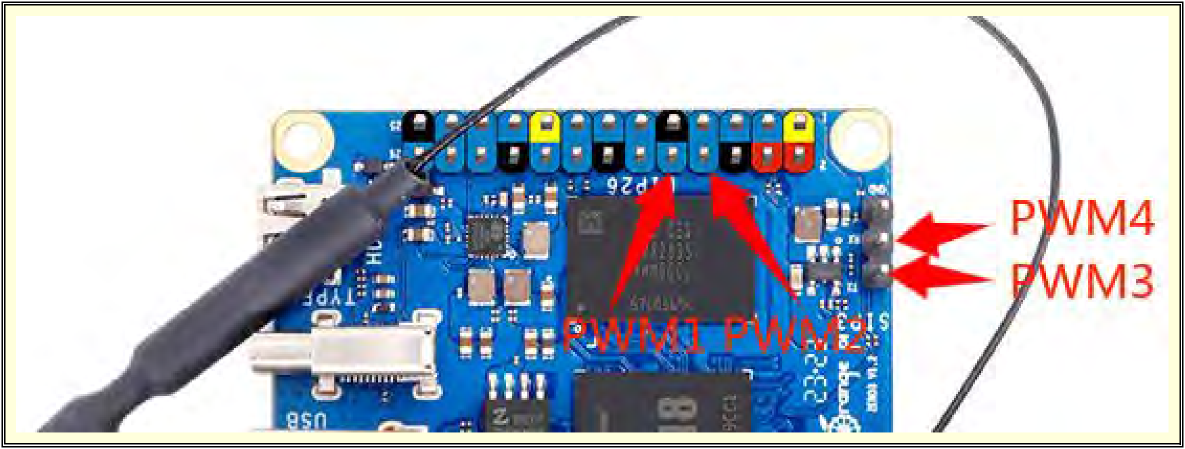
	uart5
tx pin	Corresponding to pin 8 of 26pin
rx pin	Corresponding to pin 10 of 26pin

5) 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

```
orangepi@orangepi:~$ gpio serial /dev/ttyS5          # linux-6.1 test command
orangepi@orangepi:~$ gpio serial /dev/ttyAS5         # linux-5.4 test command

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

3. 18. 6. How to test PWM using /sys/class/pwm/

The development board can use up to 4 channels of PWM, and the positions of their pins are shown in the figure below:

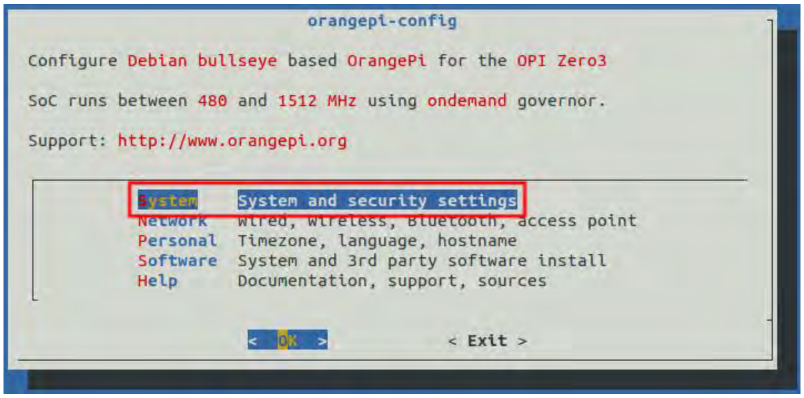


1) In the Linux system, the pwm is closed by default, and it needs to be opened manually to use it. The steps to open are as follows:

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

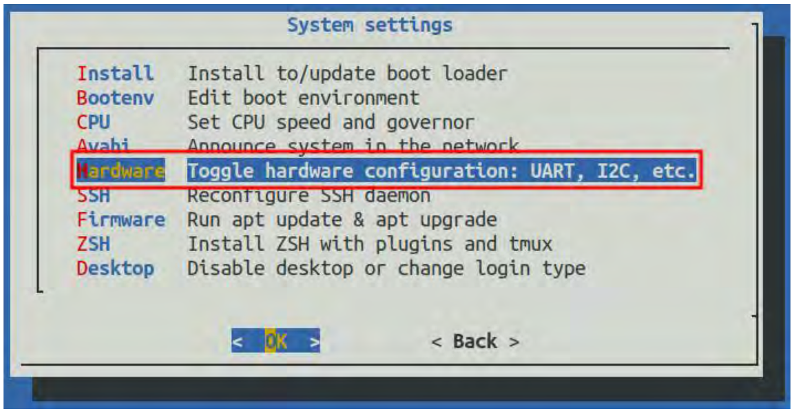
```
orangepi@orangepi:~$ sudo orangepi-config
```

b. Then select **System**



```
orangepi-config

Configure Debian bullseye based OrangePi for the OPI Zero3

SoC runs between 480 and 1512 MHz using ondemand governor.

Support: http://www.orangepi.org

        System     System and security settings
        Network    Wired, wireless, Bluetooth, access point
        Personal   Timezone, language, hostname
        Software   System and 3rd party software install
        Help       Documentation, support, sources

              < OK >              < Exit >
```

c. Then select **Hardware**

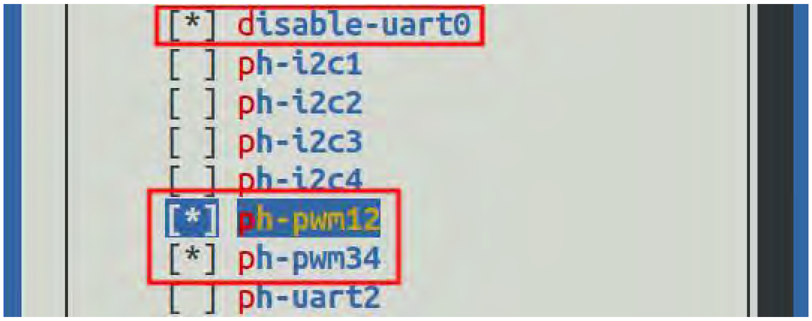


```
System settings

   Install    Install to/update boot loader
   Bootenv    Edit boot environment
   CPU        Set CPU speed and governor
   Avahi      Announce system in the network
   Hardware   Toggle hardware configuration: UART, I2C, etc.
   SSH        Reconfigure SSH daemon
   Firmware   Run apt update & apt upgrade
   ZSH        Install ZSH with plugins and tmux
   Desktop    Disable desktop or change login type

              < OK >              < Back >
```

d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the space to select the configuration corresponding to pwm

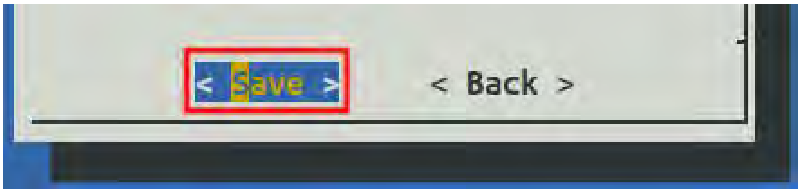
Since the RX and TX pins of UART5 in the PWM1, PWM2 and 26pin interfaces are multiplexed, please make sure that the UART5 configuration is not selected (do not check ph-uart5) when turning on PWM1 and PWM2 (you need to select ph-pwm12).

PWM3, PWM4 and the TX and RX pins in the debug serial port are multiplexed, so when using PWM3 and PWM4 (need to select ph-pwm34), please turn off the configuration of UART0 (need to select disable-uart0), after turning off UART0 The debug serial port cannot be used.

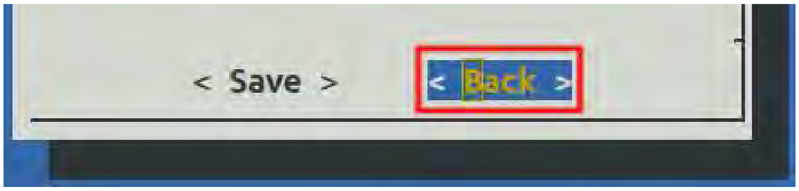


```
[*] disable-uart0
[ ] ph-i2c1
[ ] ph-i2c2
[ ] ph-i2c3
[ ] ph-i2c4
[*] ph-pwm12
[*] ph-pwm34
[ ] ph-uart2
```

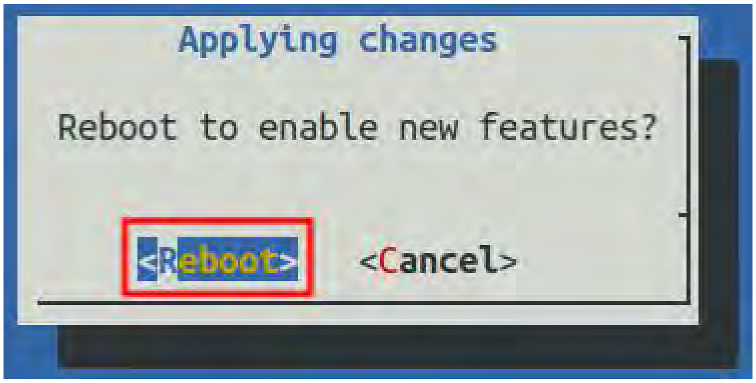
e. Then select **<Save>** to save



f. Then select **<Back>**



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


Applying changes

Reboot to enable new features?

2) After restarting, the PWM test can be started

Please execute the following commands under the root user.

a. Enter the following command on the command line to make pwm1 output a 50Hz square wave

```
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm1/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm1/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm1/enable
```

b. Enter the following command on the command line to make pwm2 output a 50Hz square wave

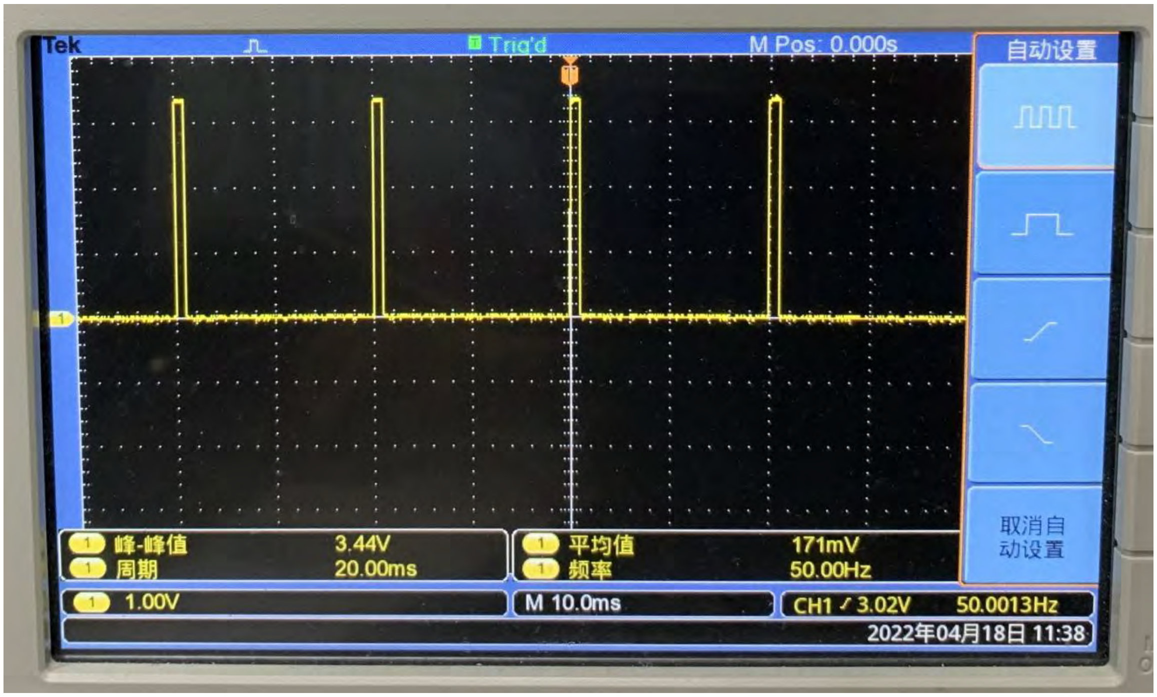
```
root@orangepi:~# echo 2 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm2/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm2/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm2/enable
```

c. Enter the following command on the command line to make pwm3 output a 50Hz square wave

```
root@orangepi:~# echo 3 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm3/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm3/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm3/enable
```

d. Enter the following command on the command line to make pwm4 output a 50Hz square wave

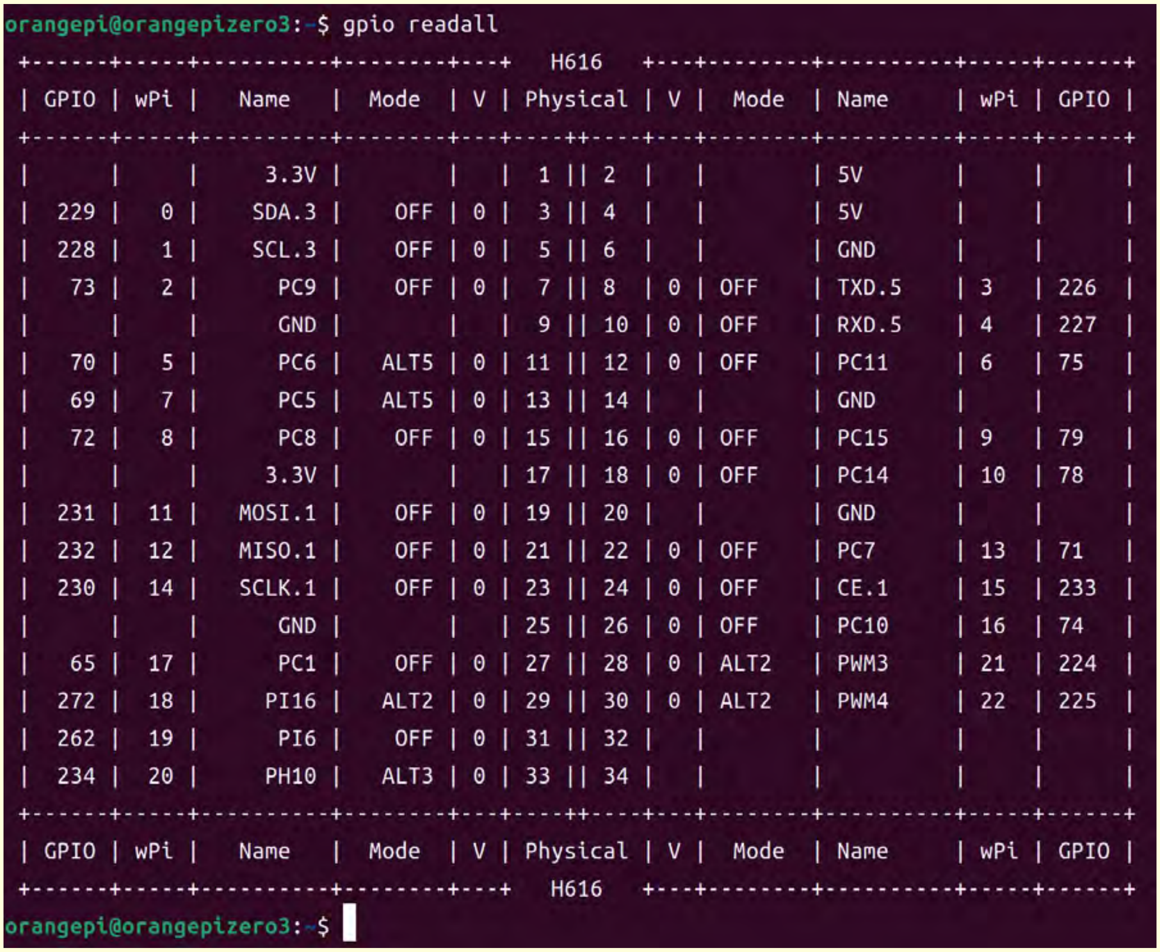
```
root@orangepi:~# echo 4 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm4/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm4/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm4/enable
```



3. 19. How to use wiringOP hardware PWM

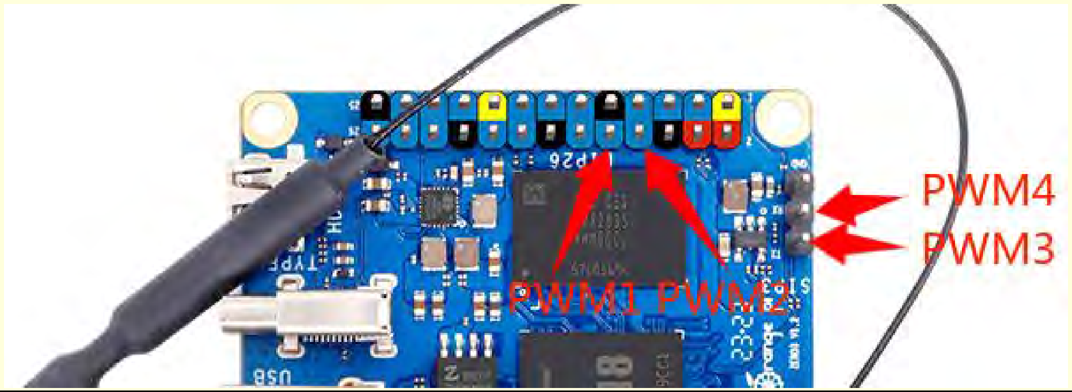
Before using the wiringOP hardware PWM function, please download the latest wiringOP source code and then compile and install it. The wiringOP preinstalled in the v1.0.2 version of the Linux image cannot use the hardware PWM function.

For how to download and install wiringOP, please refer to the instructions in the section How to Install wiringOP. The output of the gpio readall command of the latest version of wiringOP is as follows. Compared with the old version, pins 28 and 30 are added to represent PWM3 and PWM4.



```
orangepi@orangepizero3:~$ gpio readall
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |  229 |   0 |    SDA.3 |    OFF | 0 |  3 || 4  |   |        | 5V       |     |      |
 |  228 |   1 |    SCL.3 |    OFF | 0 |  5 || 6  |   |        | GND      |     |      |
 |   73 |   2 |      PC9 |    OFF | 0 |  7 || 8  | 0 | OFF    | TXD.5    | 3   | 226  |
 |      |     |      GND |        |   |  9 || 10 | 0 | OFF    | RXD.5    | 4   | 227  |
 |   70 |   5 |      PC6 |   ALT5 | 0 | 11 || 12 | 0 | OFF    | PC11     | 6   | 75   |
 |   69 |   7 |      PC5 |   ALT5 | 0 | 13 || 14 |   |        | GND      |     |      |
 |   72 |   8 |      PC8 |    OFF | 0 | 15 || 16 | 0 | OFF    | PC15     | 9   | 79   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | OFF    | PC14     | 10  | 78   |
 |  231 |  11 |   MOSI.1 |    OFF | 0 | 19 || 20 |   |        | GND      |     |      |
 |  232 |  12 |   MISO.1 |    OFF | 0 | 21 || 22 | 0 | OFF    | PC7      | 13  | 71   |
 |  230 |  14 |   SCLK.1 |    OFF | 0 | 23 || 24 | 0 | OFF    | CE.1     | 15  | 233  |
 |      |     |      GND |        |   | 25 || 26 | 0 | OFF    | PC10     | 16  | 74   |
 |   65 |  17 |      PC1 |    OFF | 0 | 27 || 28 | 0 | ALT2   | PWM3     | 21  | 224  |
 |  272 |  18 |     PI16 |   ALT2 | 0 | 29 || 30 | 0 | ALT2   | PWM4     | 22  | 225  |
 |  262 |  19 |      PI6 |    OFF | 0 | 31 || 32 |   |        |          |     |      |
 |  234 |  20 |     PH10 |   ALT3 | 0 | 33 || 34 |   |        |          |     |      |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+   H616   +---+--------+----------+-----+------+
orangepi@orangepizero3:~$
```

The development board can use up to 4 channels of PWM. The locations of their pins are as shown in the figure below:



3. 19. 1. How to set PWM using wiringOP’ s gpio command

3. 19. 1. 1. Set the corresponding pin to PWM mode

1) The corresponding relationship between the 4 PWM pins and the wPi serial number is as shown in the following table:

PWM pin	wPi serial number
PWM1	4
PWM2	3
PWM3	21
PWM4	22

2) The command to set the pin to PWM mode is as follows, taking PWM1 as an example. The third parameter needs to enter the serial number of the wPi corresponding to the PWM1 pin.

```
orangepi@orangepi:~$ gpio mode 4 pwm
```

3) After the pin is set to PWM mode, it will output a square wave with a frequency of 23475Hz and a duty cycle of 50% by default. At this time, we use an oscilloscope to measure the corresponding PWM pin and you can see the following waveform.