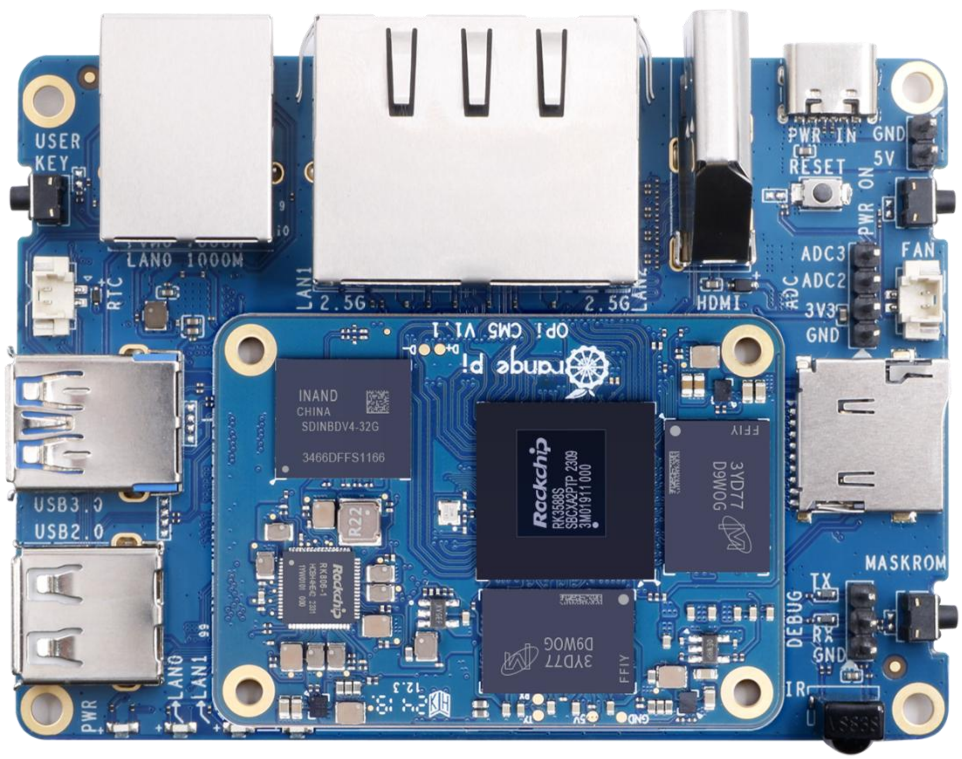
Orange Pi CM5 Base
Base board + Core board
User Manual



Catalogue

1. Basic characteristics of Orange Pi CM5 Base

1. 1. What is Orange Pi CM5 Base

The Orange Pi CM5 core board adopts the new generation eight core 64 bit ARM processor of Ruixin Micro RK3588S, specifically the quad core A76 and quad core A55. It adopts the Samsung 8nm LP process technology, with a large core main frequency of up to 2.4GHz, integrated with ARM Mali-G610 MP4 GPU, embedded with high-performance 3D and 2D image acceleration modules, and an AI accelerator NPU with up to 6 Tops of computing power. It has 4GB/8GB/16GB of memory and up to 8K display processing capability. In addition, the Orange Pi CM5 core board is also equipped with onboard eMMC, with optional capacities of 32GB/64GB/128GB/256GB.

The Orange Pi CM5 Base offers a wide range of interfaces, including dual 2.5G Ethernet ports, gigabit Ethernet ports, HDMI output, MIPI CSI, USB2.0, USB3.0, and 12 pin expansion interfaces. It can be widely used in high-end tablet, edge computing, artificial intelligence, cloud computing, AR/VR, intelligent security, smart home and other fields, covering all AIoT industries.

Orange Pi CM5 Base supports the official operating system Orange Pi OS, as well as Android 12.1, Android 13, OpenWRT, Debian11, Debian12, Ubuntu 20.04, and Ubuntu 22.04.

1. 2. Purpose of Orange Pi CM5 Base

We can use it to achieve:

- A Linux desktop computer
- A Linux network server
- Android game consoles, etc
- Router

Of course, there are also many other features. With a powerful ecosystem and a variety of expansion accessories, Orange Pi can help users easily achieve delivery

from creativity to prototype and then to mass production. It is an ideal creative platform for makers, dreamers, and hobbyists.

1. 3. Hardware characteristics of Orange Pi CM5 core board

OPi CM5 Core Board Hardware Specifications	
Main control chip	Rockchip RK3388S(8nm LP Process）
CPU	• 8-core 64 bit processor • Typical size core architectures of 4-core Cortex-A76 and 4-core Cortex-A55 • Large core frequency 2.4GHz, small core frequency 1.8GHz
GPU	• Integrated ARM Mali-G610 • Compatible with OpenGL ES1.1/2.0/3.2, OpenCL 2.2, and Vulkan 1.2
NPU	6 Tops of computing power, supporting mixed operations of INT4/INT8/INT16
PMU	RK806-1
RAM	LPDDR4/4x：2GB、4GB、8GB、16GB
EMMC	eMMC：32GB、64GB、128GB、256GB
Interface	3 * 100PIN (model: DF40C-100DP-0.4V (51)), including the following interfaces: 1*TYPE C or DP1.4 3*USB2.0 1*HDMI 2.1 or eDP1.3 1*uSD 1*4-lane MIPI DPHY TX 1*2-lane MIPI DPHY TX 2*2-lane MIPI DPHY RX 1*4-lane MIPI CSI RX or 2*2-lane MIPI CSI RX 1*SATA III or PCIe2.0 1*SATA III or PCIe2.0 or USB3.0 USB3.0*1+USB2.0*3 POWER_ON、RESET、MASKROM、RECOVERY etc

	SDIO 3.0 or RGMII I2C、I2S、UART、SPI、CAN、PWM、PDM、GPIO etc
Power Supply	input：DC 5V MAX1800mA output：DC3.3V MAX600mA 和 DC1.8V MAX600mA
PCB	Length: 55mm, width: 40mm, thickness: 1.6mm
rangePi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd	

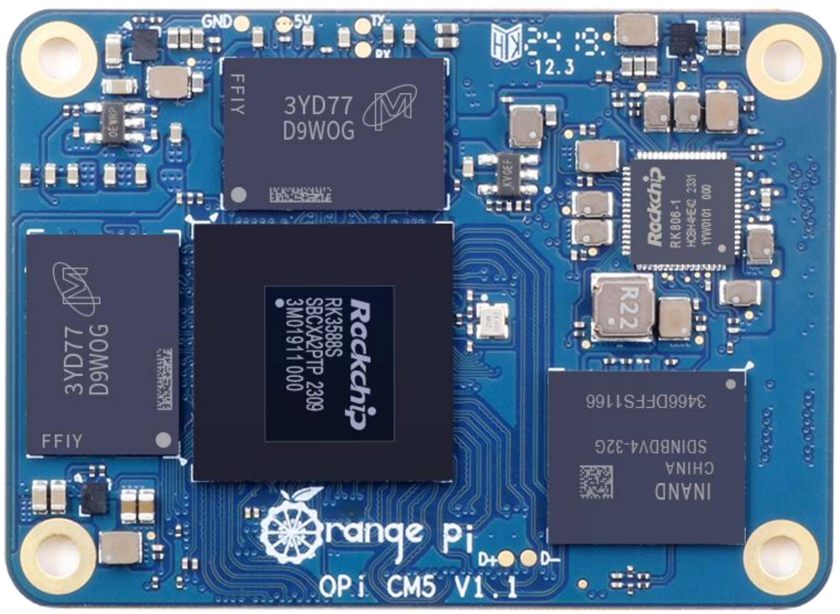
1. 4. Hardware characteristics of Orange Pi CM5 Base board

Orange Pi CM5 Base Hardware Features	
Board to board connectors	3 * 100PIN, 0.4mm PIN pitch, connector model: df40c-100ds-0.4v
storage	• MicroSD (TF) slot • eMMC module
USB Interface	1 * USB3.1 and 1 * USB2.0
Video output	HDMI2.1，Maximum support 8K@60Hz
camera	4 * MIPI CSI 2Lane
audio frequency	HDMI Audio output
Ethernet port	1 * 1000M ethernet port（YT8531C-CA） 2 * 2.5G ethernet port（RTL8125B）
12PIN Expansion interface	• 12Pin FPC Socket，Pitch:0.5mm • Supports UART, PWM, I2C, SPI, CAN, GPIO and other functions
Key	Button and On/Off button
power supply	Supports Type-C power supply, 5V@4A or 5V@5A
RTC battery interface	2 pins, 1.25mm specification, used to power the RTC module
LED	Power indicator light, 3 * Ethernet port indicator light
FAN	5V 2PIN 1.25mm socket
Infra-red	Infrared receiver
ADC	4PIN single row pin, Pitch: 2.54mm

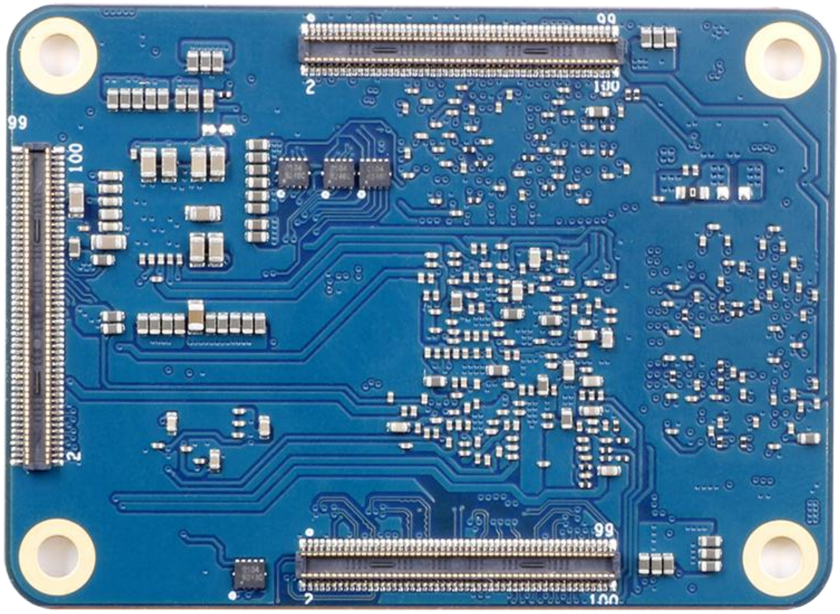
	Two ADC inputs, maximum 1.8V input, 12bit
Debug UART	3Pin debugging serial port
PCB	Length: 90mm, width: 66mm, thickness:
rangePi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd	

1. 5. Top and Bottom Views of Orange Pi CM5 Core Board

Top view:

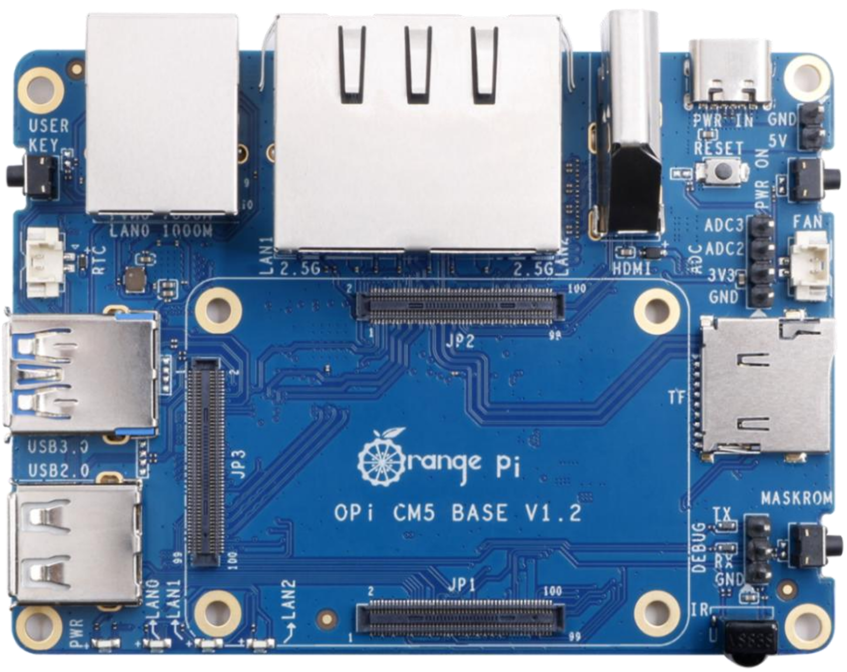


Bottom view:

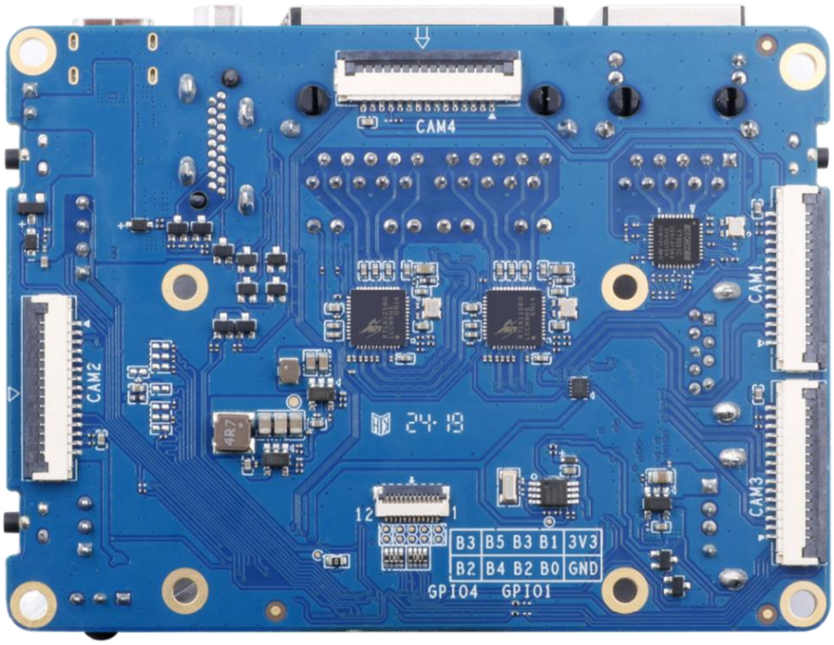


1. 6. Top and bottom views of Orange Pi CM5 Base board

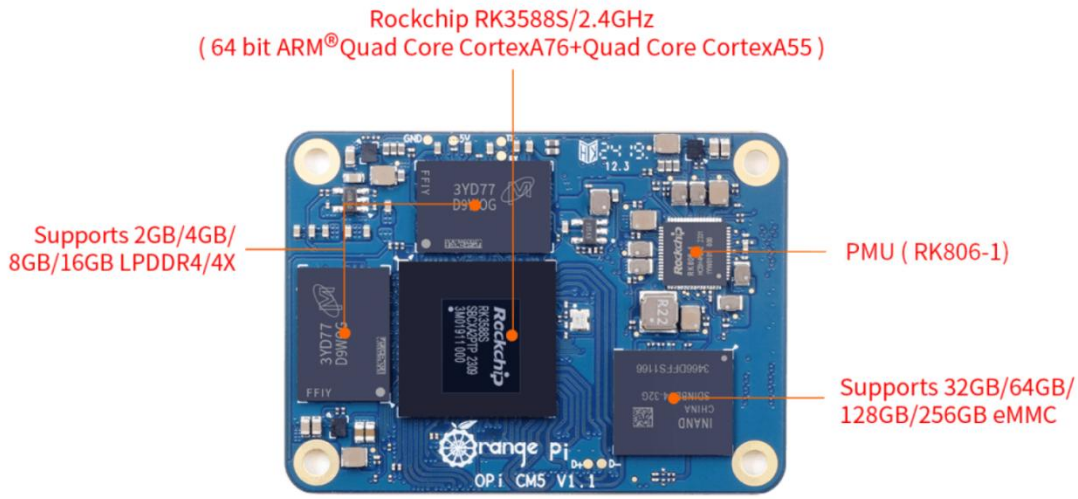
Top view:



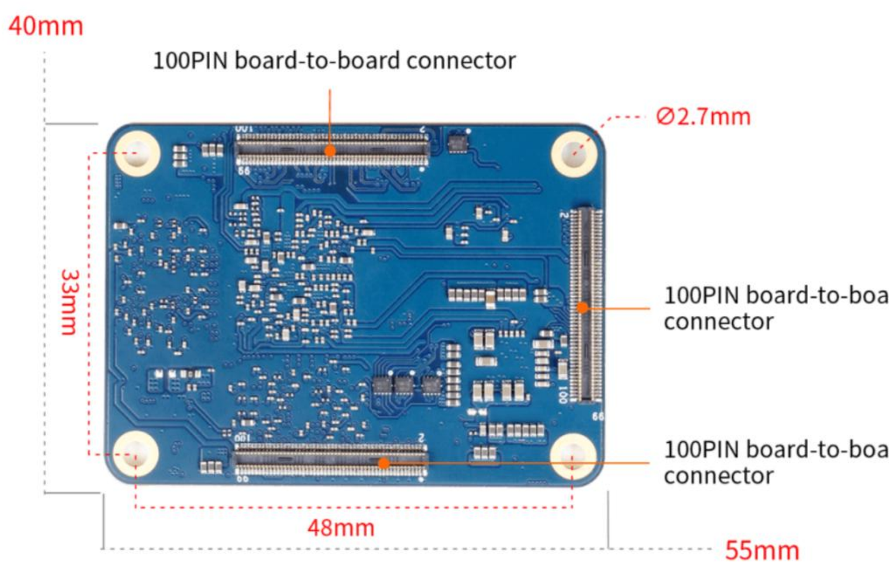
Bottom view:


CAM4
CAM1
CAM2
CAM3
B3 B5 B3 B1 3V3
B2 B4 B2 B0 GND
GPIO4 GPIO1

1.7. Interface Details of Orange Pi CM5 Core Board

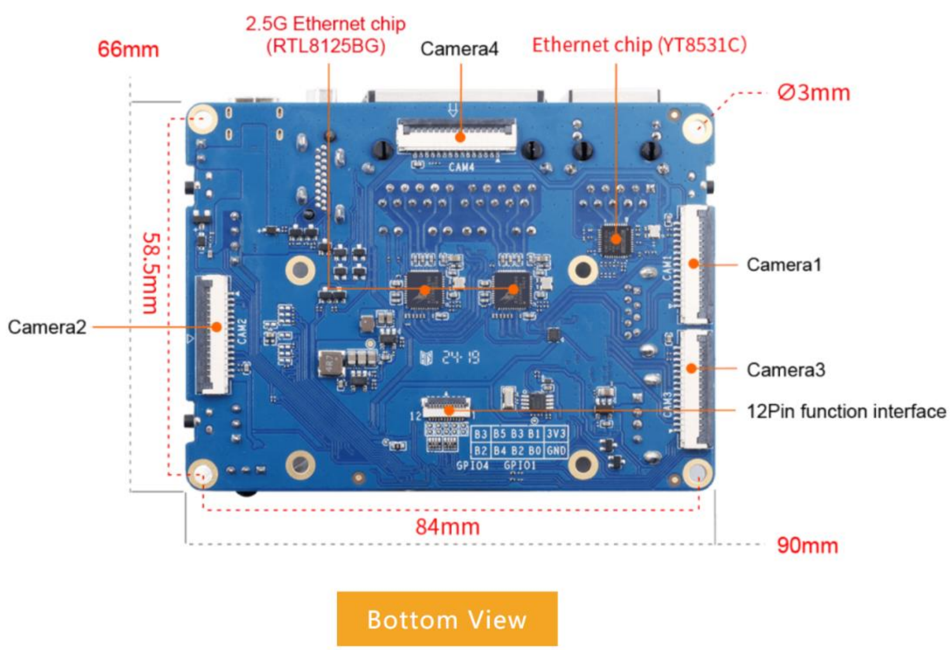


Top View



Bottom View

1.8. Interface detail diagram of Orange Pi CM5 Base board



The diameter of the four positioning holes on the core board is 2.7mm, and the diameter of the four positioning holes on the bottom board is 3.0mm.

2. Introduction to using the development board

2. 1. Prepare the necessary accessories

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

2) TF card reader, used to burn images into TF cards

3) Display with HDMI interface.

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

Note that if you want to connect to a 4K or 8K monitor, please ensure that the HDMI cable supports 4K or 8K video output.

5) USB public to public data cable, used for burning images and using ADB and other functions.

6) Power adapter, it is recommended to use a 5V/4A or 5V/5A Type-C power supply for power supply.

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

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

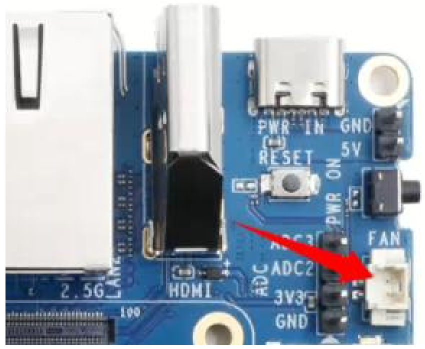
8) USB camera.

9) Infrared remote control.

> **Note that the remote control of the air conditioning or TV cannot control the Orange Pi development board. The operating system provided by Orange Pi can only guarantee that the remote control provided by Orange Pi can be used by default.**

10) A 5V cooling fan. As shown in the figure below, there is an interface on the development board for connecting the cooling fan, with a interface specification of **2pin 1.25mm** spacing.

The fan on the development board can be adjusted for speed and switch through PWM.



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

12) A personal computer with Ubuntu and Windows operating systems installed.

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

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

1) The download website for the English version of the materials is:

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-CM5.html

2) The information mainly includes：

a. **Android source code**: saved on Google Drive

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

2. 3. Method of burning Linux images to TF cards based on Windows PC

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

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

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

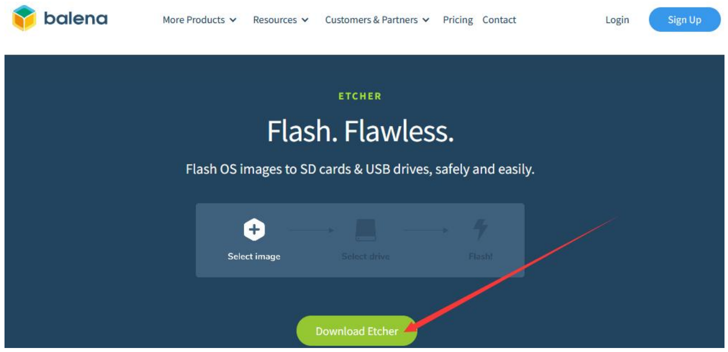
2) Then use the card reader to insert the TF card into the computer

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

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

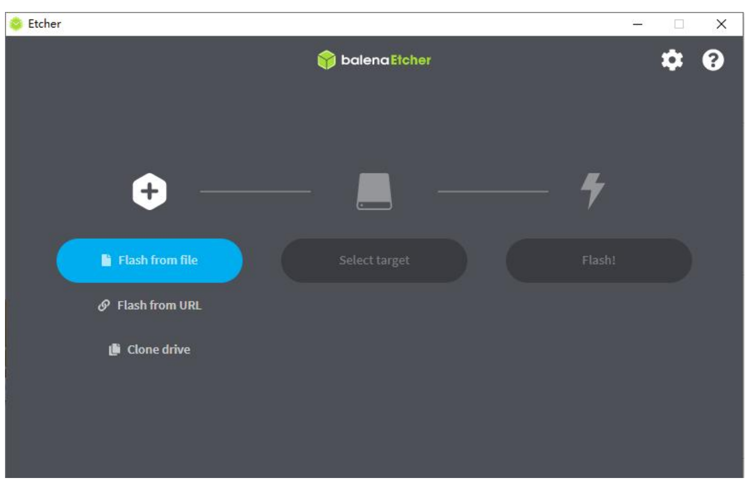
https://www.balena.io/etcher/

5) After entering the BalenaEtcher download page, clicking the green download button will redirect you to the software download location

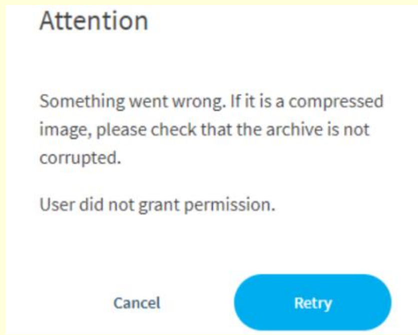


6) Then you can choose to download the Portable version of BalenaEtcher software. The Portable version does not need to be installed, and can be opened by double clicking to use it

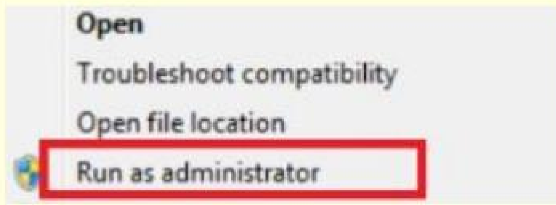
7) If you are downloading a version of BalenaEtcher that requires installation, please install it before using it. If you download the Portable version of balenaEtcher, simply double-click to open it. The interface of balenaEtcher after opening is shown in the following figure:



If the following error is prompted when opening balenaEtcher:



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



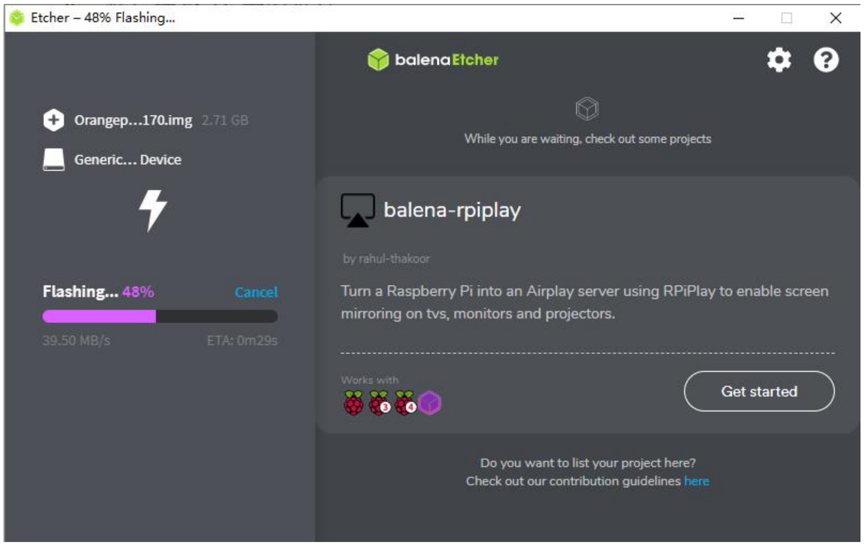
8) The specific steps to use balenaEtcher to burn the Linux image are aa follow

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

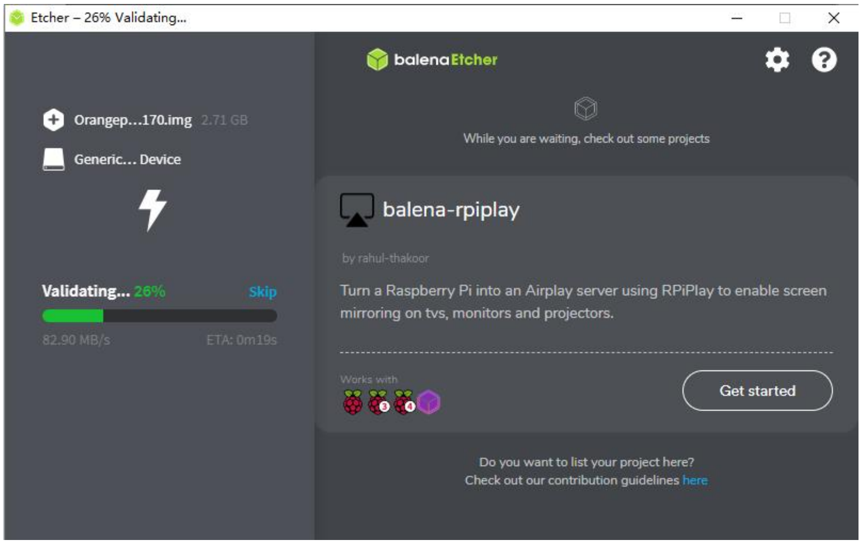
b. Then select the drive letter of the TF card

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

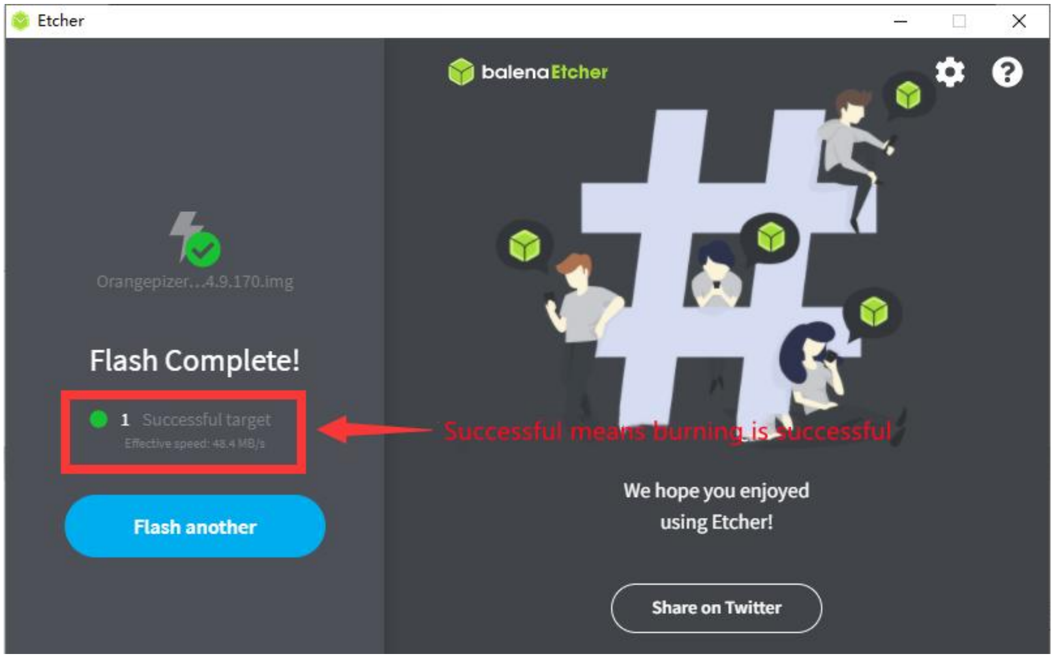
9) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card



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



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



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

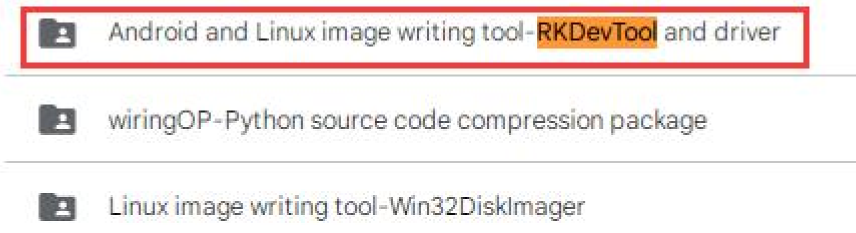
1) Please select balenaEtcher, right-click, and then select Run as administrator.

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

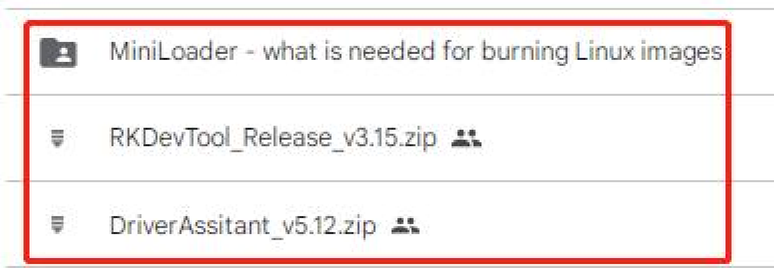
3) Then insert the TF card into the card slot of the development board

4) Then download the Ruixin micro driver **DriverAssitant_v5.12.zip**. zip and MiniLoader, as well as the burning tool **RKDevTool_Release_v3.15.zip**, from **Orange Pi's data download page**

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



b. Then download all the files below



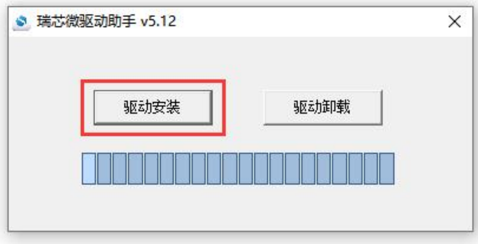
Note that the "MiniLoader - something needed to burn Linux images" folder is hereinafter referred to as the MiniLoader folder.

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

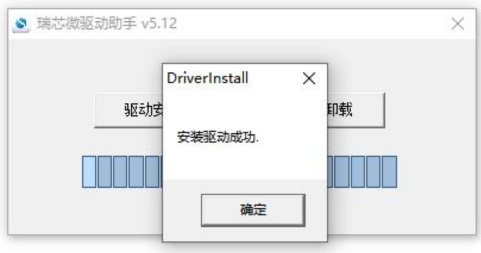
6) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**. zip, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

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

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



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

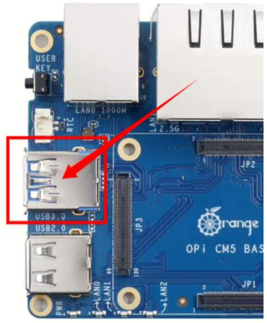


8) Then decompress **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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

10) Then start burning the Linux image to the TF card.

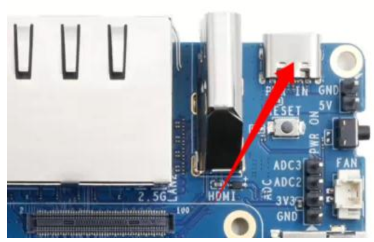
a. Firstly, connect the development board to the Windows computer through a USB male to female data cable. The location of the USB flash port on the development board is shown in the following figure:



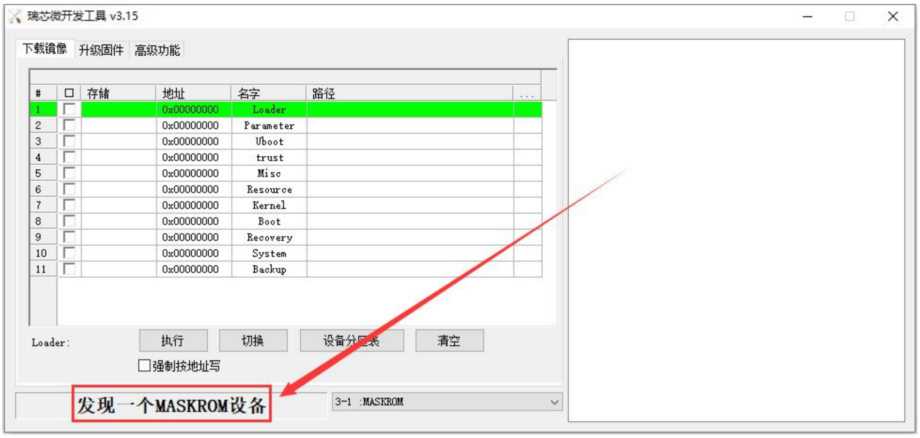
b. Ensure that the development board is not connected to the Type-C power supply.

c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:

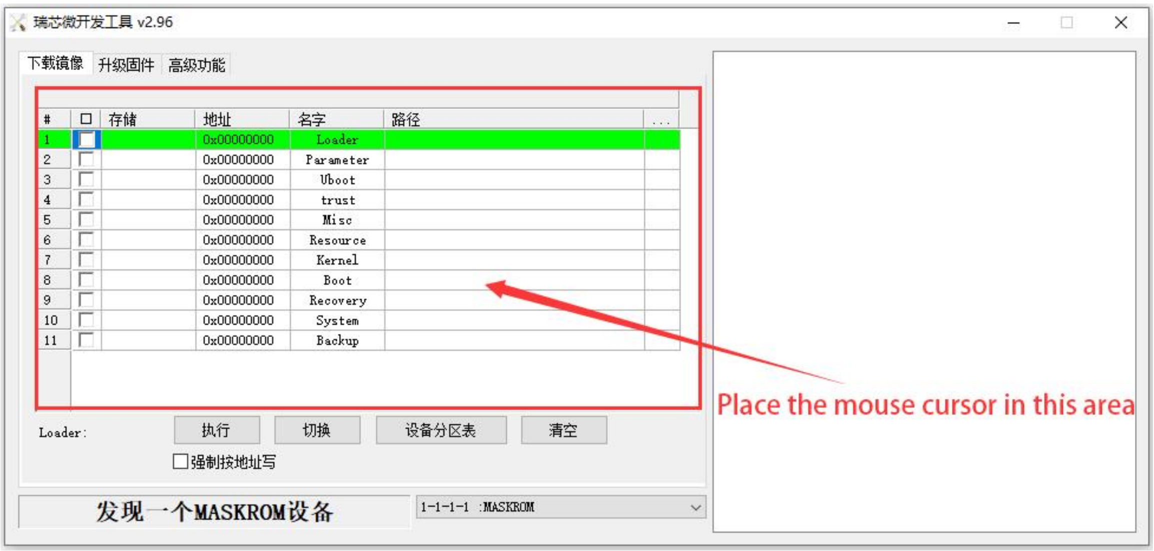
d. Then connect the Type-C interface power to the development board, power it on, and then release the MaskROM button. The location of the Type-C power interface is as follows:



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

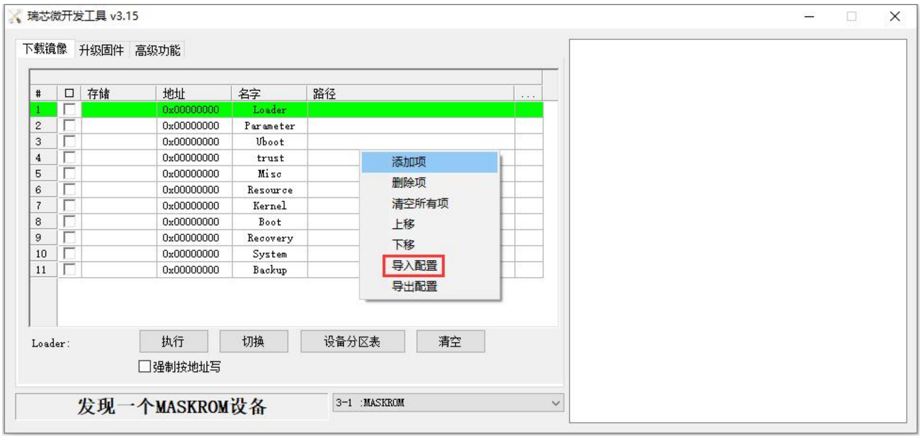


f. Then place the mouse cursor in the area below.

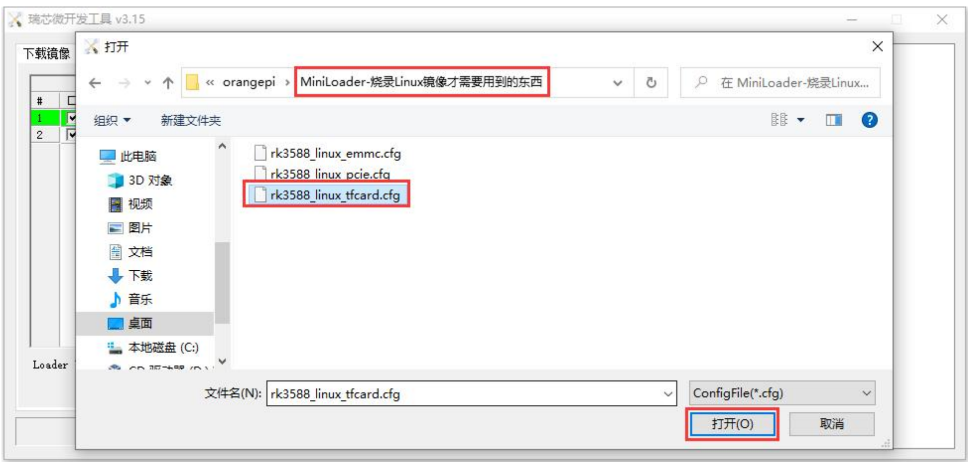


g. Then, clicking the right mouse button will bring up the selection interface shown in the following figure.

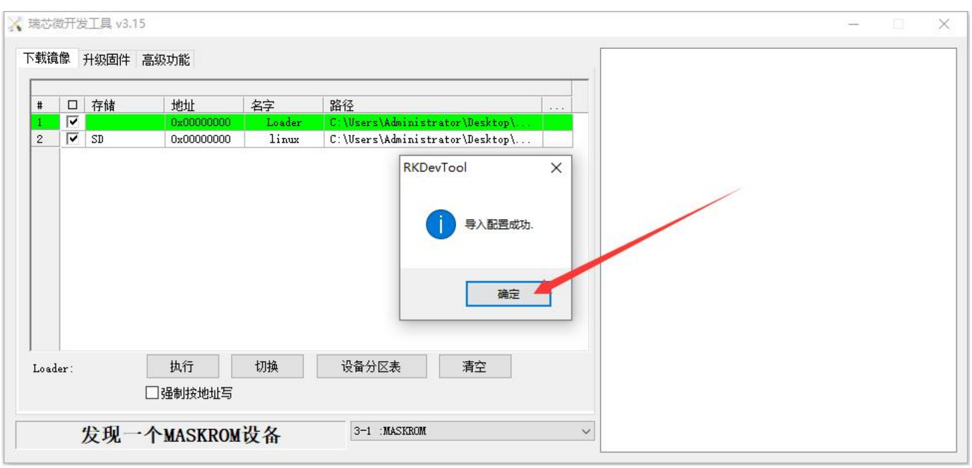
h. Then select the **import configuration** option.



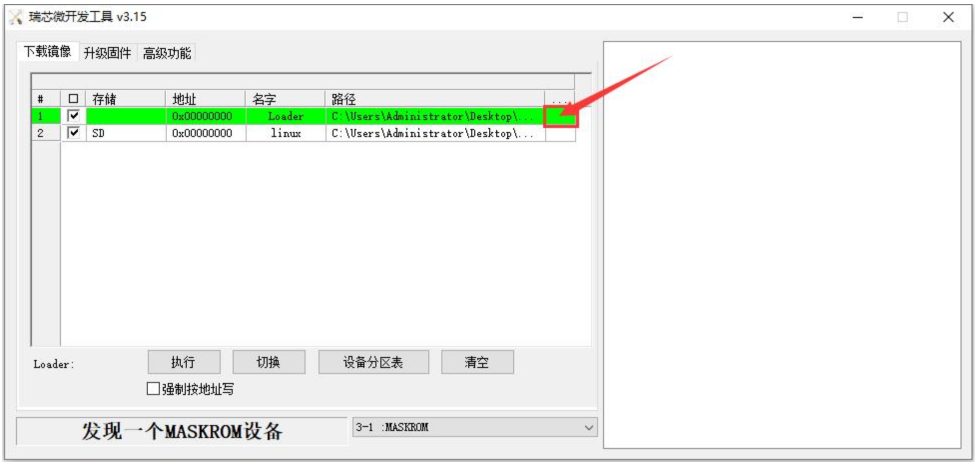
i. Then select the **rk3588_linux_tfcard.cfg** configuration file from the **MiniLoader** folder downloaded earlier, and click **open**.



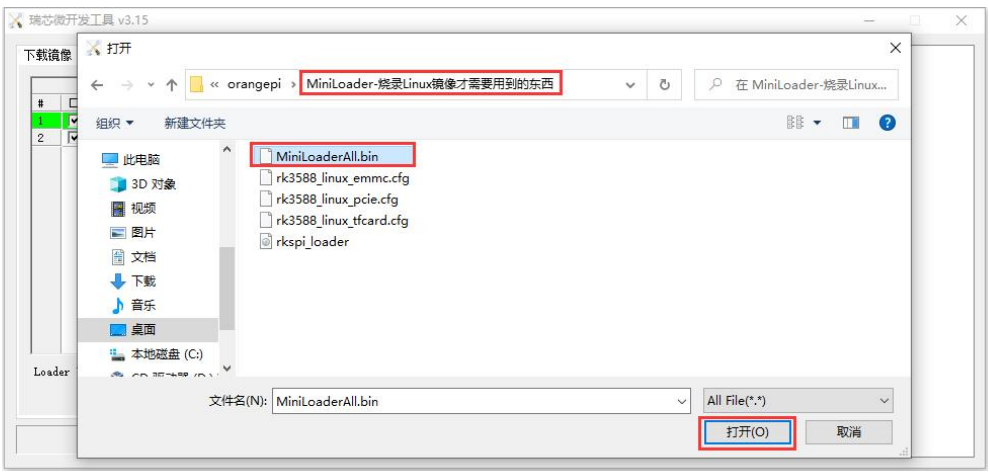
j. Then click **OK**.



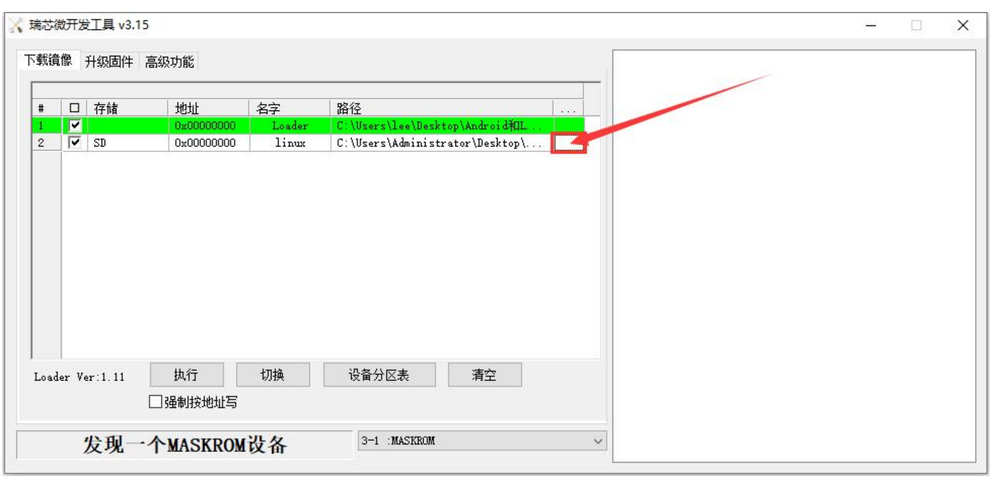
k. Then click on the location shown in the figure below.



l. Select **MiniLoaderAll.bin** from the **MiniLoader** folder downloaded earlier, and then click to **open** it.



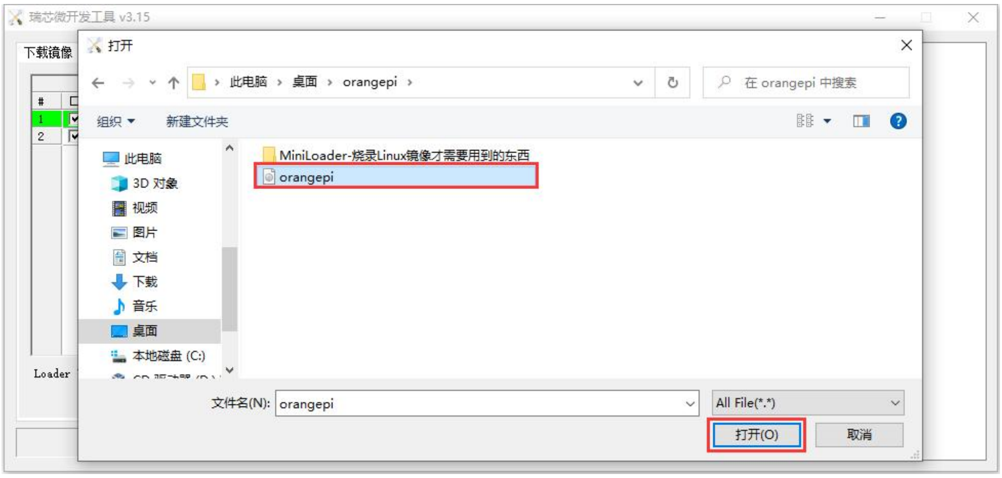
m. Then click on the location shown in the figure below.



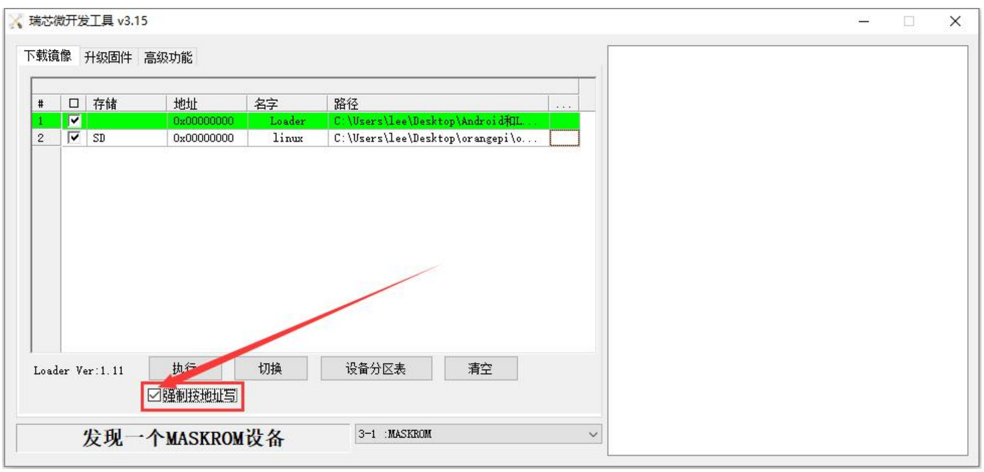
n. Then select the path to the Linux image you want to burn, and click **Open**.

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

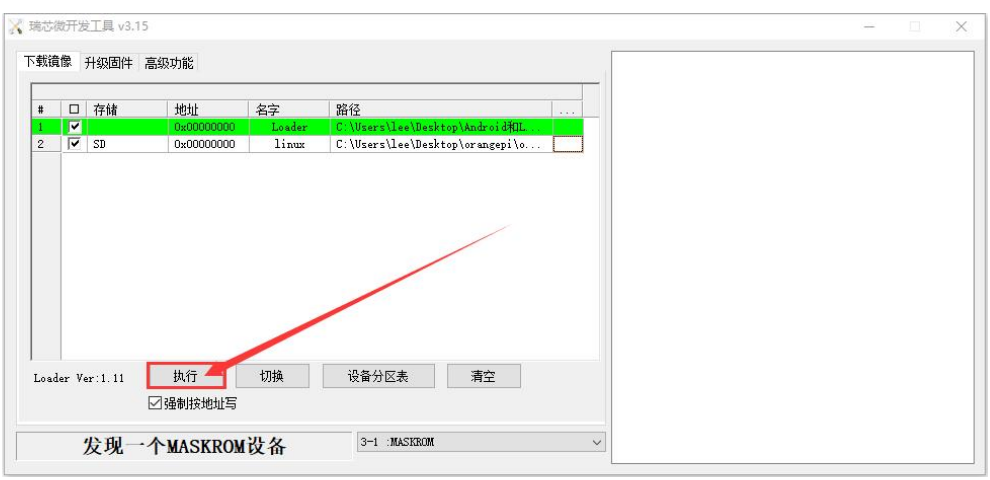
orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



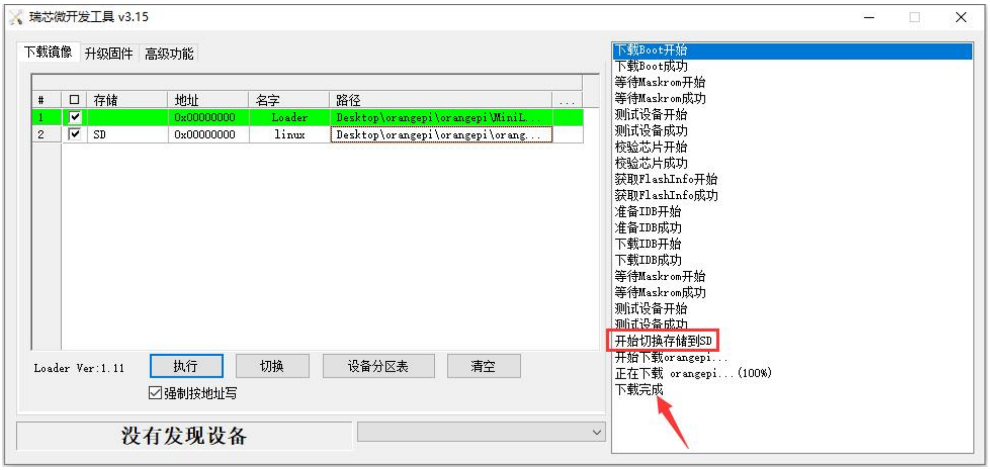
o. Then please check the option to **force writing by address**.



p. Clicking the execute button again will start burning the Linux image to the development board's tf card.



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



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

2. 4. How to burn Linux image to TF card based on Ubuntu

Note that the Linux image referred to here specifically refers to Linux distribution images such as Debian, Ubuntu, OpenWRT, or OPi OS Arch downloaded from the Orange Pi download page. Ubuntu PC refers to a personal computer with the Ubuntu system installed.

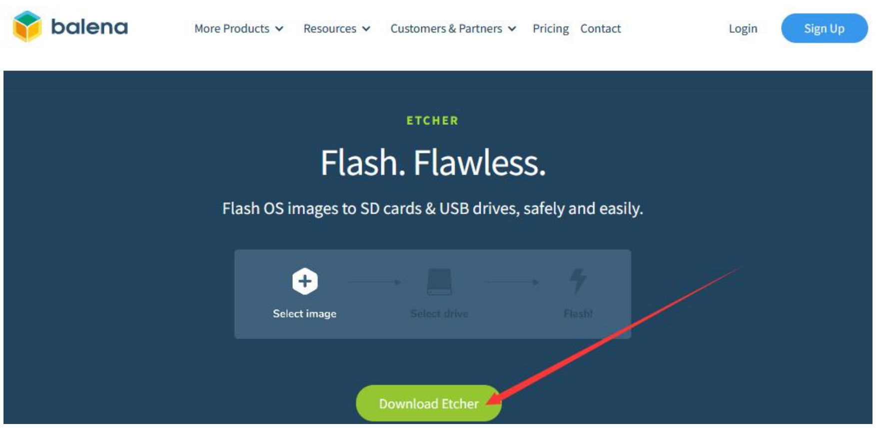
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brand

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

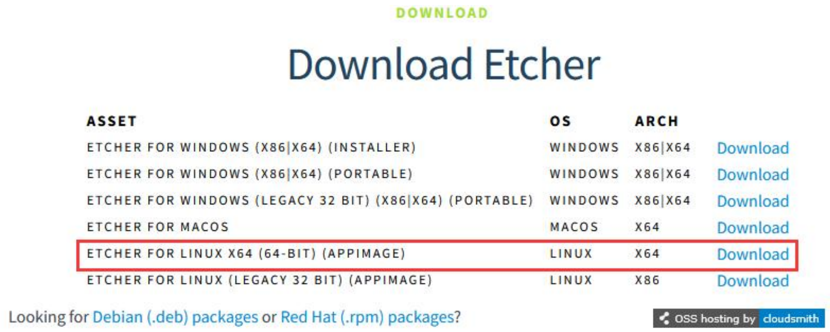
3) Download the balenaEtcher software, the download address is:

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

4) After entering the BalenaEtcher download page, clicking the green download button will redirect you to the software download location



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



6) Download the compressed file of the Linux operating system image that you want to burn from **Orange Pi's information download page**, and then use decompression software to extract it. In the extracted file, the file ending in "**.img**" is the operating system image file, which is generally over 2GB in size

a) The decompression command for the compressed file ending in 7z is as follows:

```
test@test:~$ 7z x orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
test@test:~$ ls orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.*
orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.sha     #Checksum file
orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img     #Mirror file
```

b) If you are downloading an OpenWRT image and the compressed file ends in gz, the decompression command is as follows:

```
test@test:~$ gunzip openwrt-aarch64-opicm5-24.03-linux-6.1.43-ext4.img.gz
test@test:~$ ls openwrt-aarch64-opicm5-24.03-linux-6.1.43-ext4.img
openwrt-aarch64-opicm5-24.03-linux-6.1.43-ext4.img     #Mirror file
```

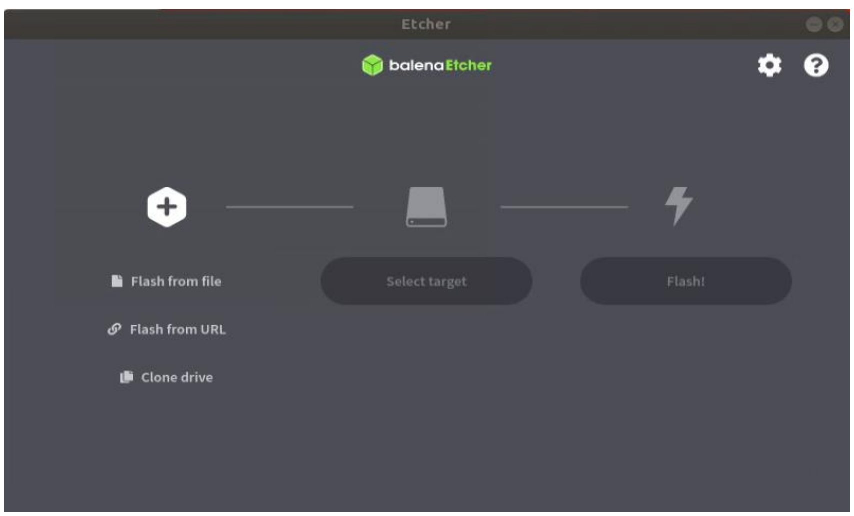
7) After decompressing the image, you can first use the **sha256sum -c *.sha** command to calculate whether the checksum is correct. If the prompt is **successful**, it indicates that the downloaded image is not incorrect. You can rest assured to burn it to the TF card. If the prompt is that the **checksum does not match**, it indicates that there is a problem with the downloaded image. Please try downloading again

```
test@test:~$ sha256sum -c *.sha
orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img: OK
```

If you are downloading an OpenWRT image, you need to verify the compressed file instead of decompressing it before verifying it

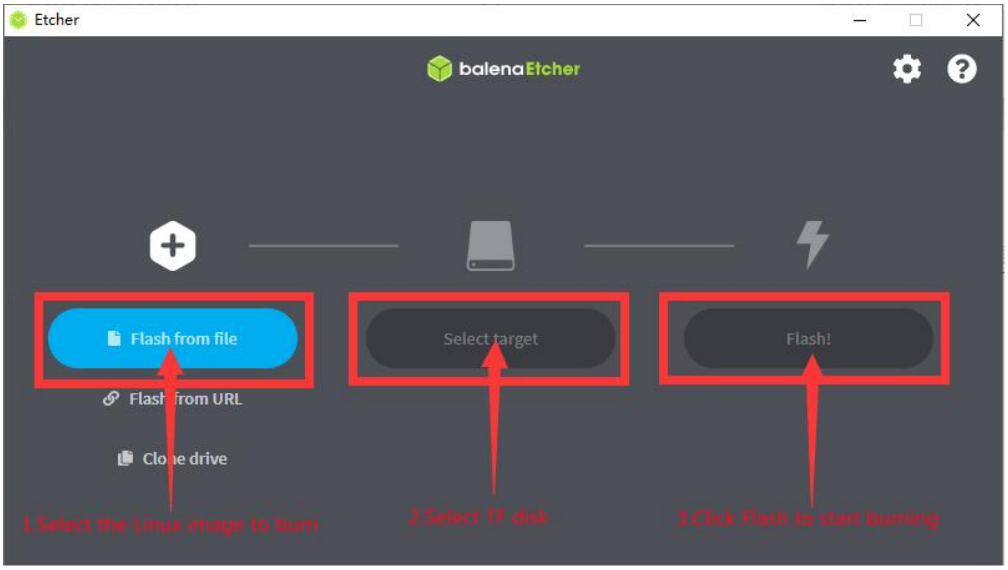
```
test@test:~$ sha256sum -c openwrt-aarch64-opicm5-24.03-linux-6.1.43-ext4.img.gz.sha
openwrt-aarch64-opicm5-24.03-linux-6.1.43-ext4.img.gz: OK
```

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

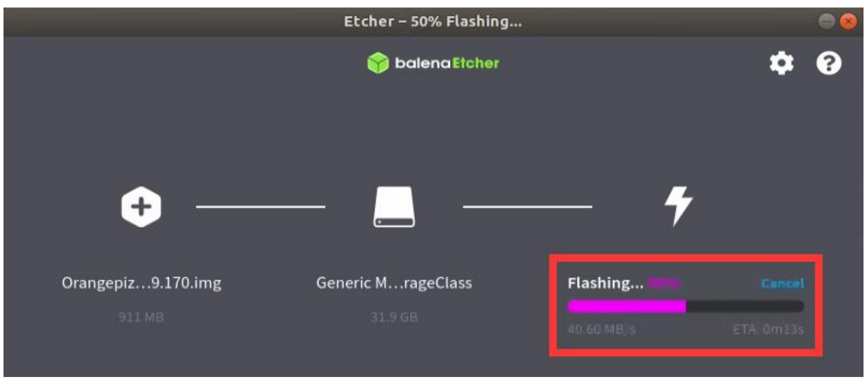


9) The specific steps to use balenaEtcher to burn the Linux image are as follows

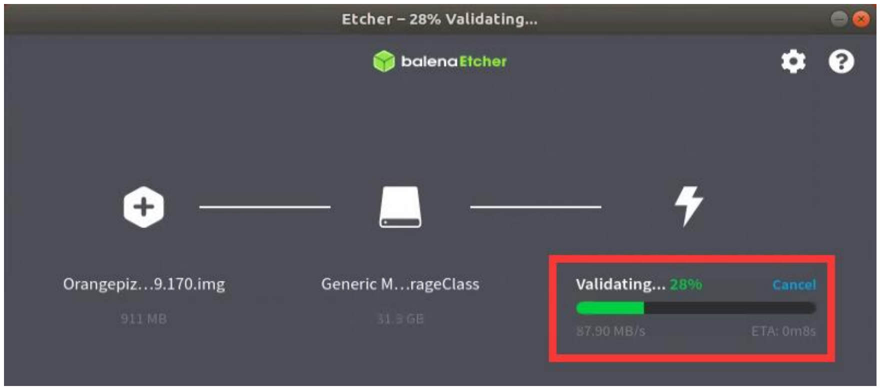
a. First select the path of the Linux image file to be burned
b. Then select the drive letter of the TF Card
c. Finally, click Flash to start burning the Linux image to the TF Card



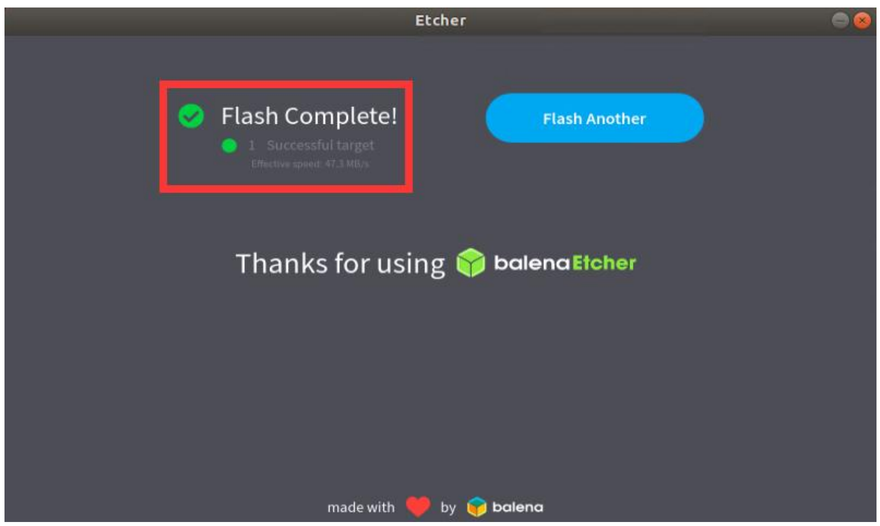
10) The interface displayed during the process of burning a Linux image with balenaEtcher is shown in the following figure. In addition, the progress bar displays purple to indicate that the Linux image is being burned to the TF card.



11) After the Linux image is burned, BalenaEtcher defaults to verifying the images burned to the TF card to ensure that there are no issues during the burning process. As shown in the figure below, a green progress bar indicates that the image has been burned and BalenaEtcher is verifying the burned image.



12) After the successful burning is completed, the display interface of balenaEtcher is shown in the following figure. If a green indicator icon is displayed, it indicates that the image burning is successful. At this point, you can exit balenaEtcher, then unplug the TF card and insert it into the TF card slot of the development board for use.



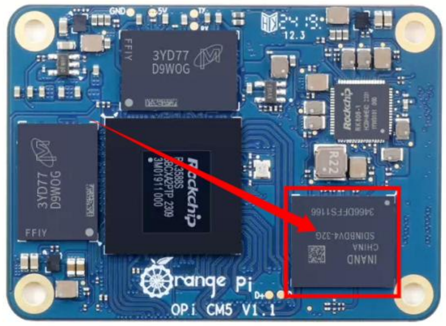
2.5. The method of burning Linux images into eMMC

2.5.1. Method of burning Linux images into eMMC using RKDevTool

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

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

1) The Orange Pi CM5 core board has an eMMC module located as follows:

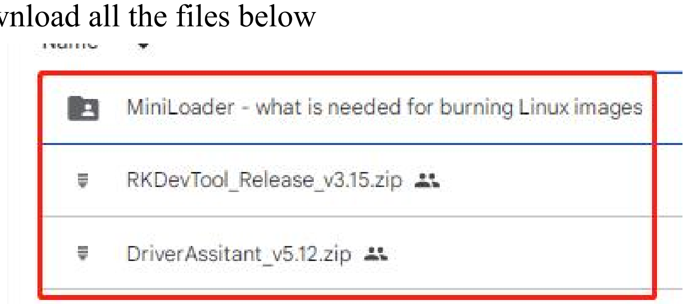


2) Firstly, it is necessary to prepare a high-quality USB male to female data cable.

3) Then download the Ruixin micro driver **DriverAssitant_v5.12.zip** and **MiniLoader**, as well as the burning tool **RKDevTool_Release_v3.15.zip**, from **Orange Pi's data download page**

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

b. Then download all the files below



Note that the "MiniLoader - something needed to burn Linux images" folder is

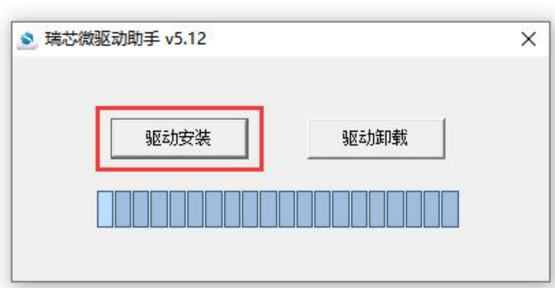
hereinafter referred to as the MiniLoader folder.

4) Then download the compressed file of the Linux operating system image that you want to burn from **Orange Pi's information download page**, and use decompression software to extract it. In the extracted file, the file ending in "**.img**" is the operating system image file, which is generally over 2GB in size

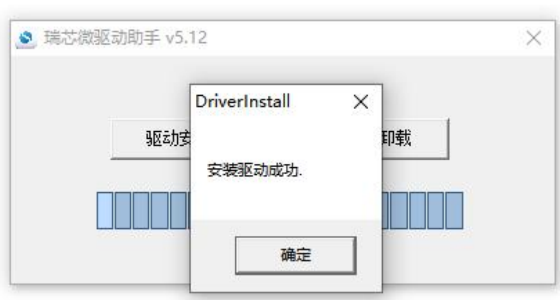
5) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

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

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



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

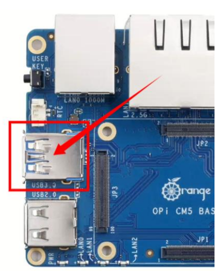


7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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

9) Then start burning Linux images into eMMC

a. Firstly, connect the development board to the Windows computer through a USB male to female data cable. The location of the USB flash port on the development board is shown in the following figure:

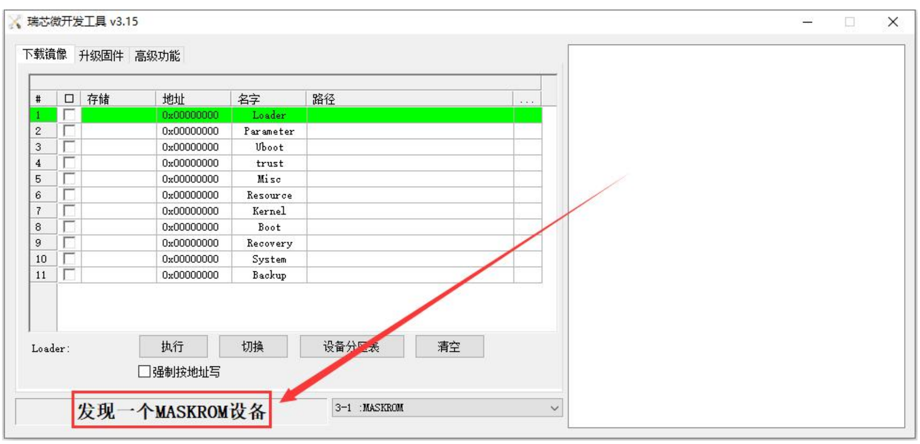


b. Ensure that the development board is not plugged into a TF card or connected to a power source.

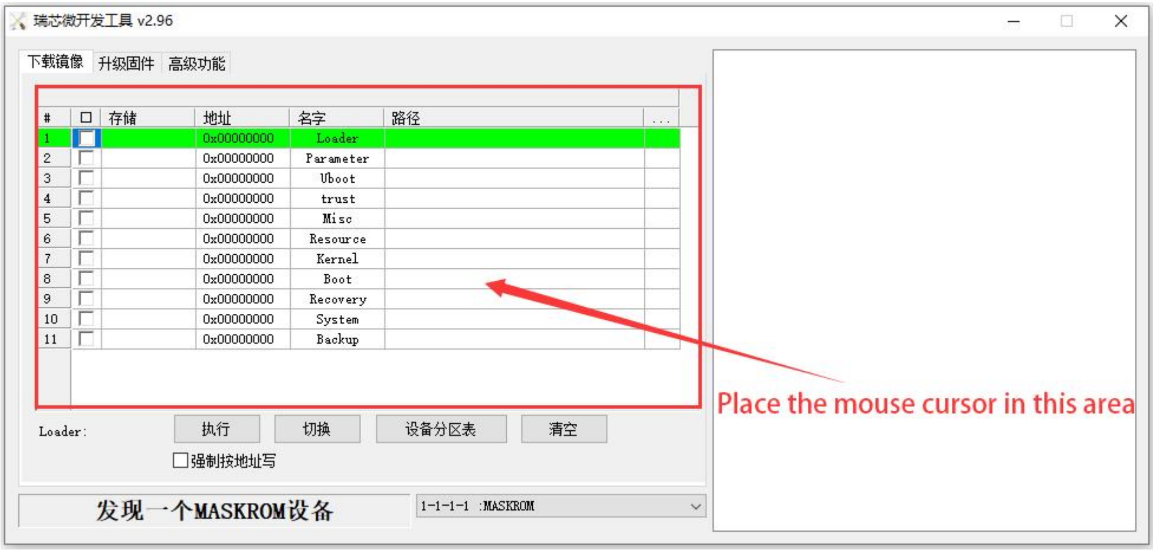
c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:

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

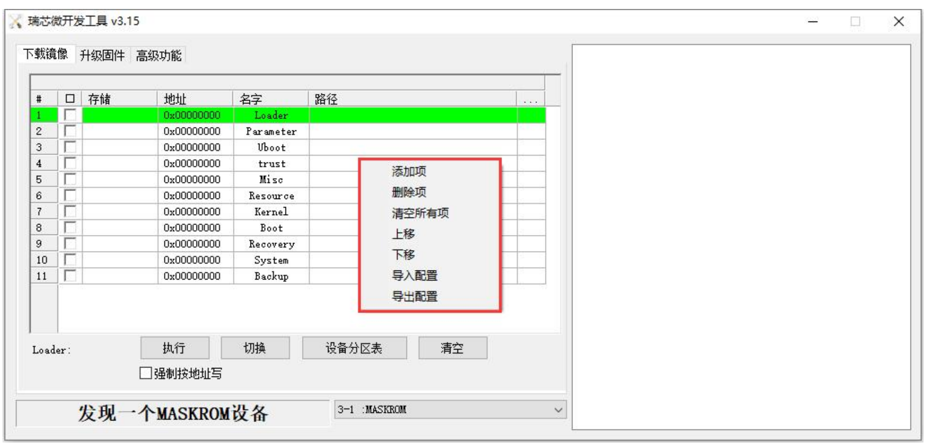
e. If the previous steps are successful, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**Found a MASKROM device**".



f. Then place the mouse cursor in the area below

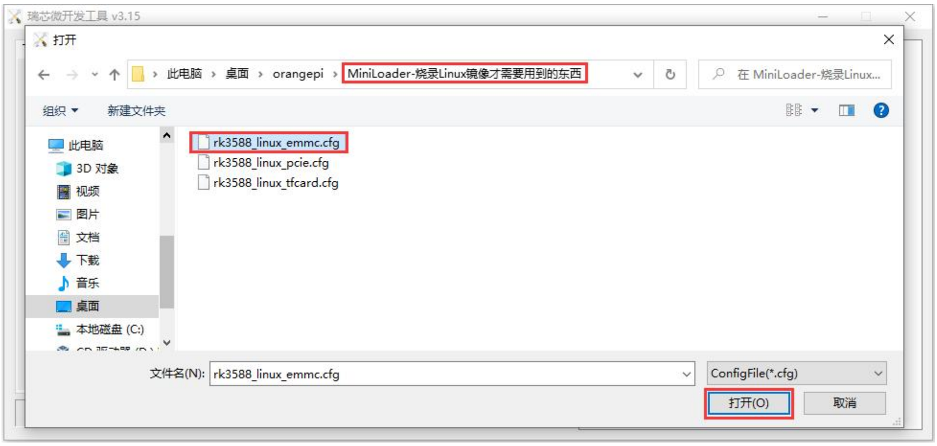


g. Then, clicking the right mouse button will pop up the selection interface shown in the following figure

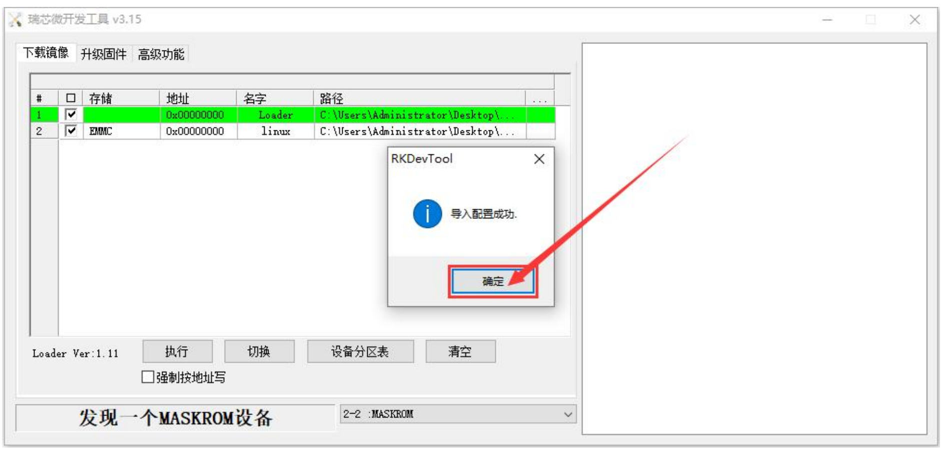


h. Then select the **import configuration** option

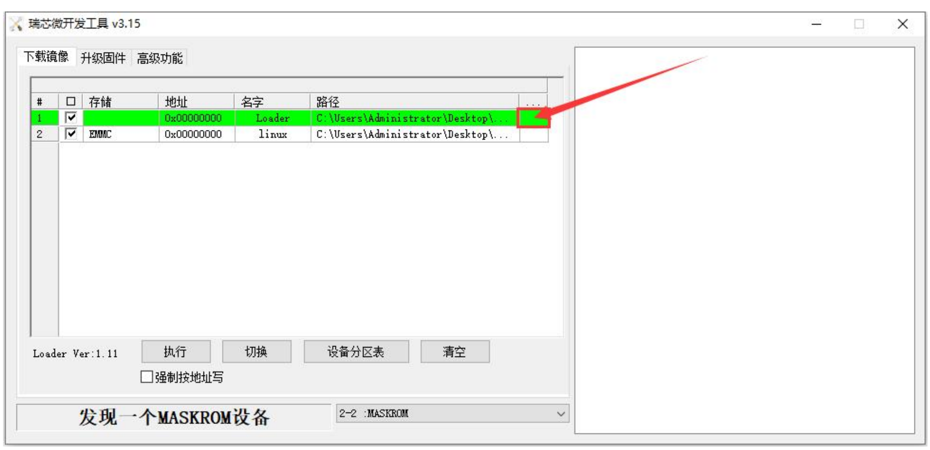
i. Then select the **rk3588_linux_emmc.cfg** configuration file from the **MiniLoader** folder downloaded earlier, and click to **open**



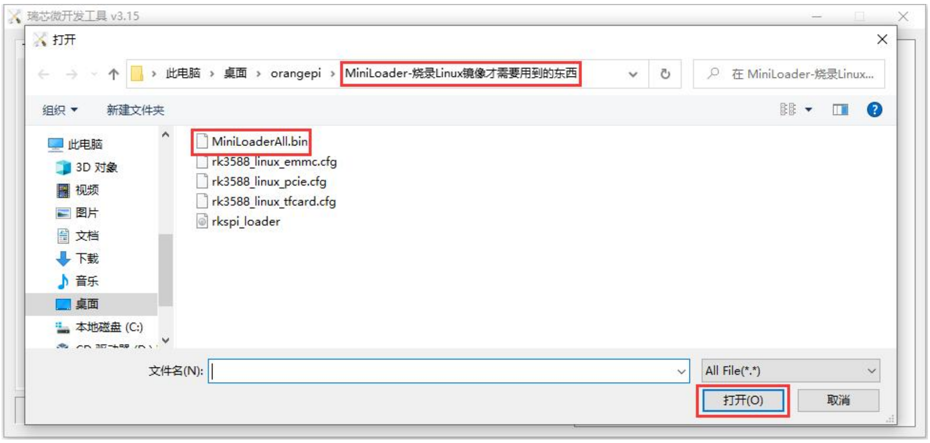
j. Then click **OK**



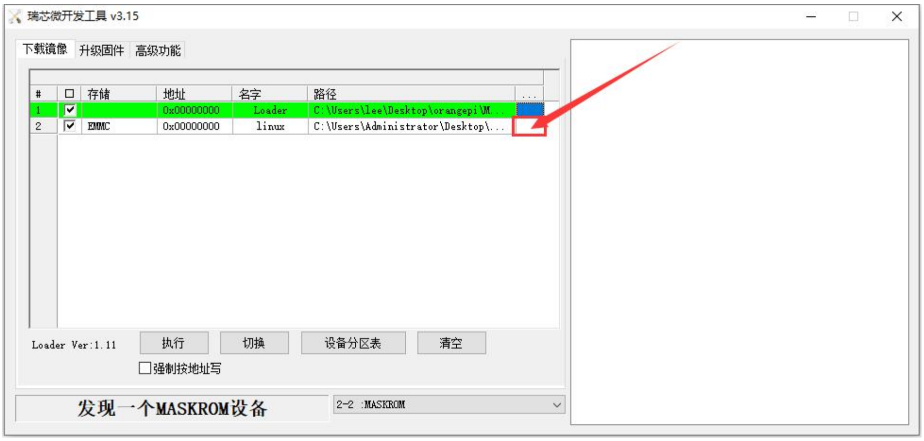
k. Then click on the location shown in the following image



l. Select **MiniLoaderAll.bin** from the **MiniLoader** folder downloaded earlier, and then click **open**

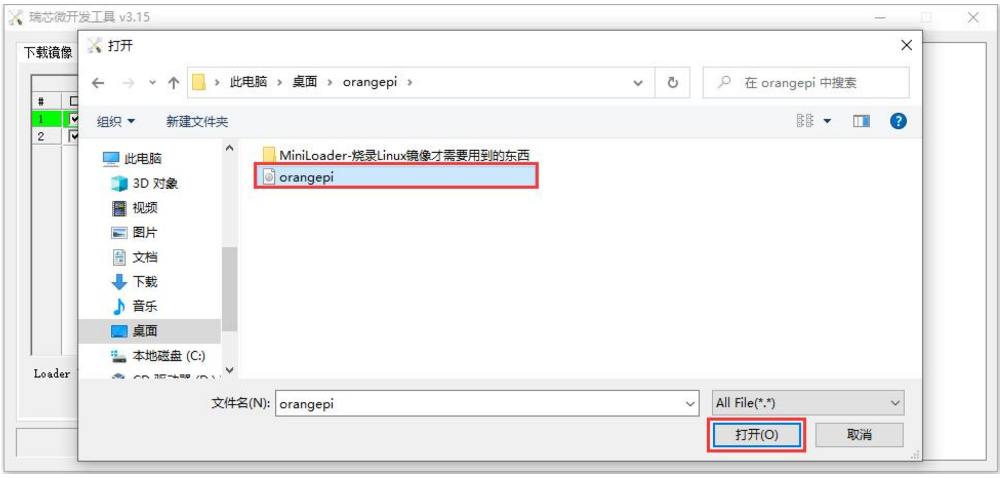


m. Then click on the location shown in the following image

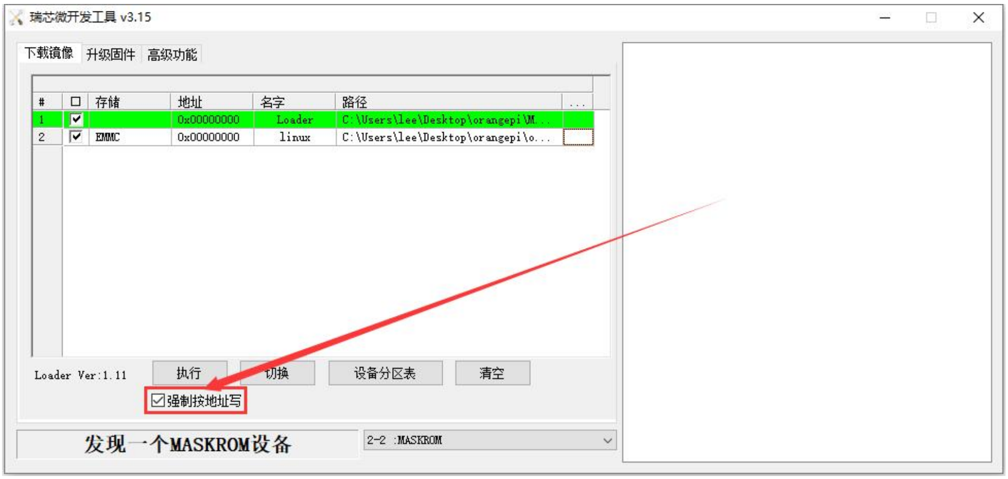


n. Then select the path to the Linux image you want to burn, and click **open**

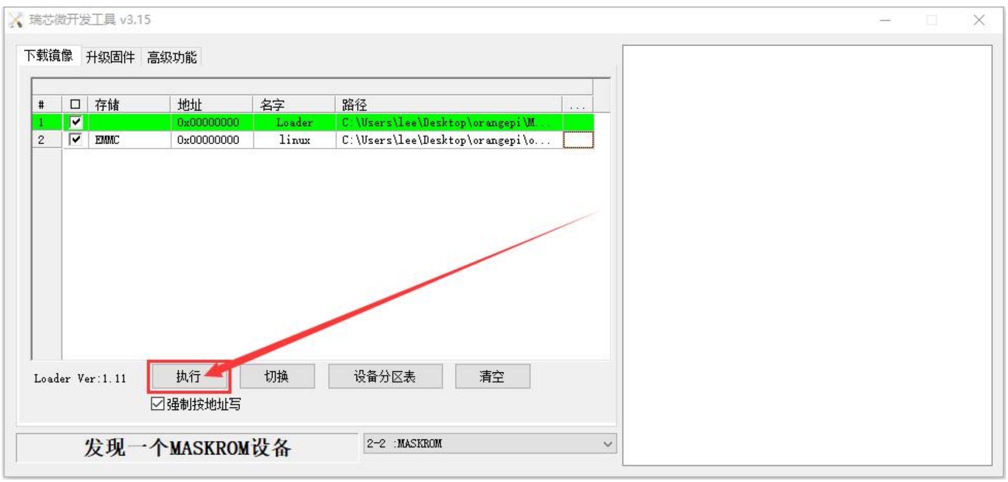
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



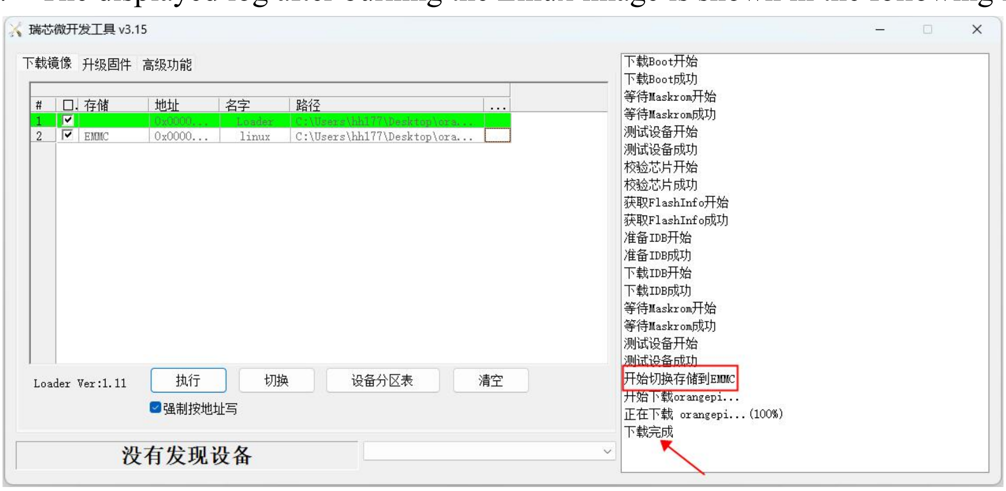
o. Then please check the option to **force writing by address**



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



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

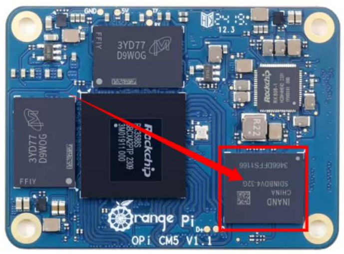


r. After burning the Linux image into eMMC, the Linux system will automatically start.

2.5.2. The method of burning Linux images into eMMC using the dd command

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

1) The Orange Pi CM5 core board has an eMMC module located as follows:



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

3) After using the TF card to start the linux system, we first upload the decompressed linux image file (Debian, Ubuntu image or OPi Arch image downloaded from the official website) to the TF card. For the method of **uploading the linux image file to the development board,** please refer to the description in the section of the method of uploading files to the development board Linux system.

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

```
orangepi@orangepi:~$ cd /home/orangepi/Desktop
orangepi@orangepi:~/Desktop$ ls
orangepicm5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img
```

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

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

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

3. If a display screen such as HDMI or LCD is connected, you can open a command line terminal on the desktop.

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

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

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

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

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

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

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

```
sudo dd bs=1M if=orangepicm5_x.x.x_debian_bullseye_desktop_xfce_linux5.10.160.img of=/dev/mmcblk1 status=progress

sudo sync
```

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

The detailed description of all parameters of the dd command and more usage

can be viewed by executing the `man dd` command in the linux system.

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

2. 6. Method of burning Android images into TF cards

2. 6. 1. Method of burning using RKDevTool

1) Firstly, it is necessary to prepare a high-quality USB male to female data cable.

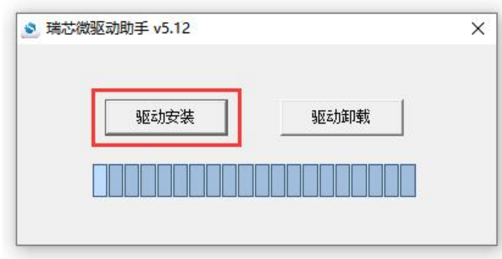
2) Then download the Ruixin micro driver **DriverAssitant_v5.12.zip** and the burning tool **RKDevTool_Release_v3.15.zip** from **Orange Pi's data download page**.

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

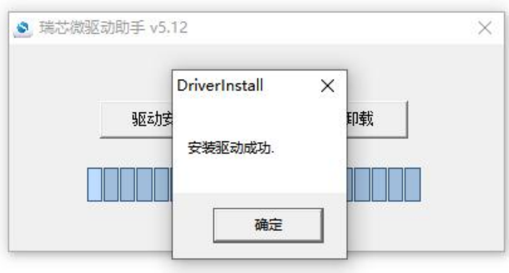
4) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it.

5) The steps to install the Ruixin micro driver after opening **DriverInstall.exe** are as follows:

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



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

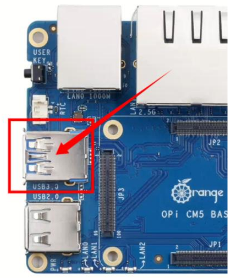


6) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it.

7) After opening the **RKDevTool**burning tool, because the computer has not yet connected to the development board through the USB2.0 male-to-male data cable, a message "**No device found**" will appear in the lower left corner.

8) Then start burning the Android image into the TF card.

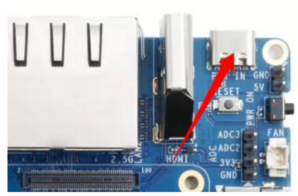
a. Firstly, connect the development board to the Windows computer through a USB male to female data cable. The location of the USB burning interface on the development board is shown in the following figure:



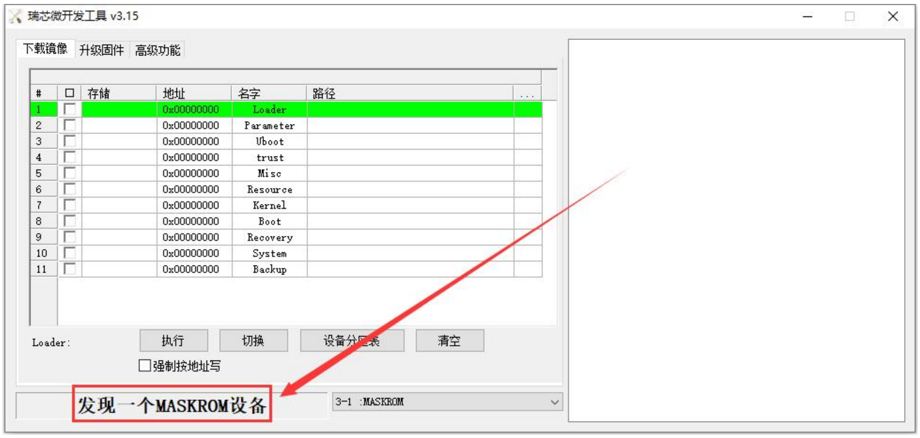
b. Then insert the TF card into the development board and ensure that the board is not connected to a power source.

c. Then hold down the MaskROM button on the development board and hold it down. The position of the MaskROM button on the development board is shown in the following figure:

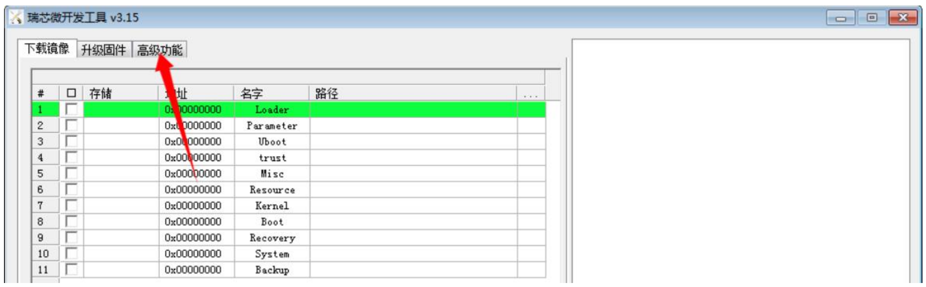
d. Then connect the Type-C interface power to the development board, power it on, and then release the MaskROM button.



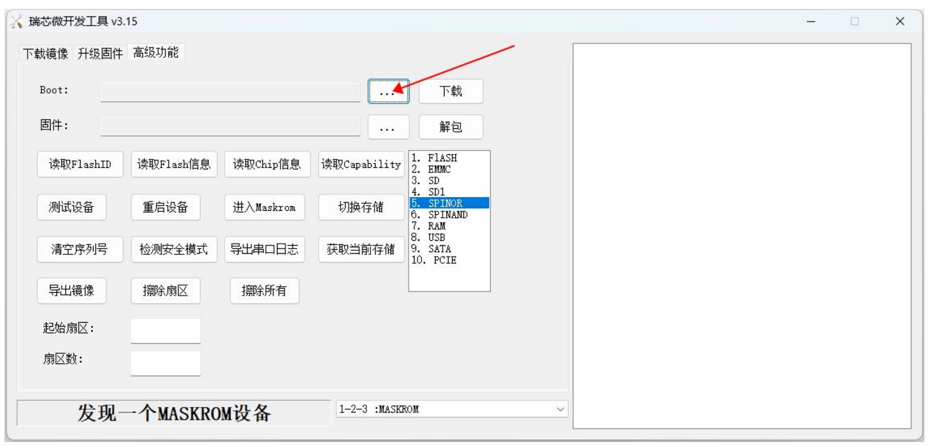
e. If the previous steps go well, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device was found**"



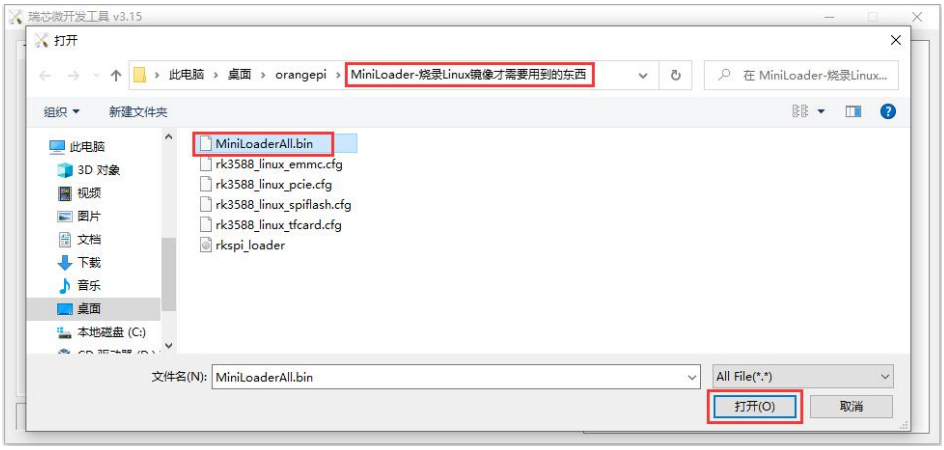
f. Then please select **Advanced Features**



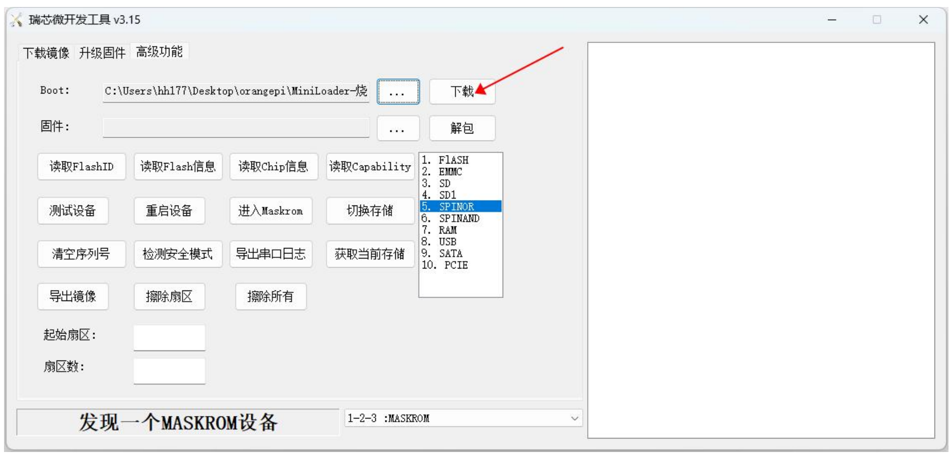
g. Then click the location shown in the picture below



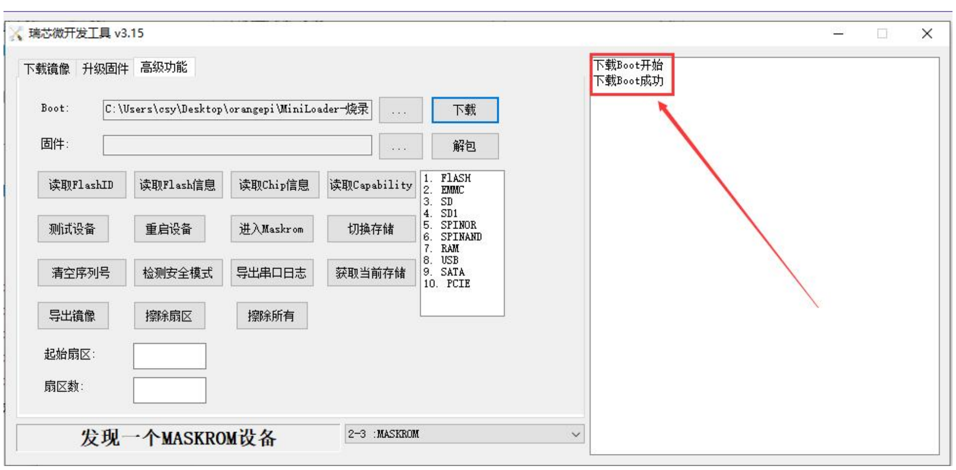
h. Then select **MiniLoaderAll.bin** in the **MiniLoader**r folder downloaded earlier, and then click Open



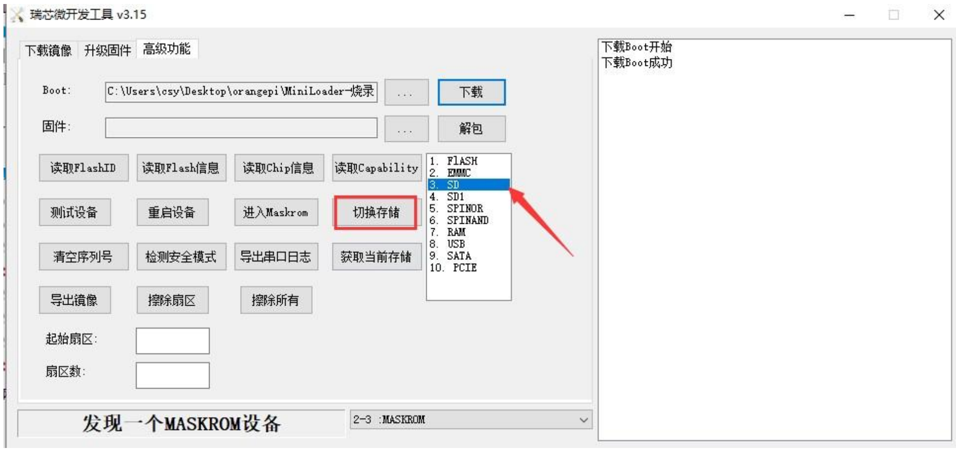
i. Then click **Download**



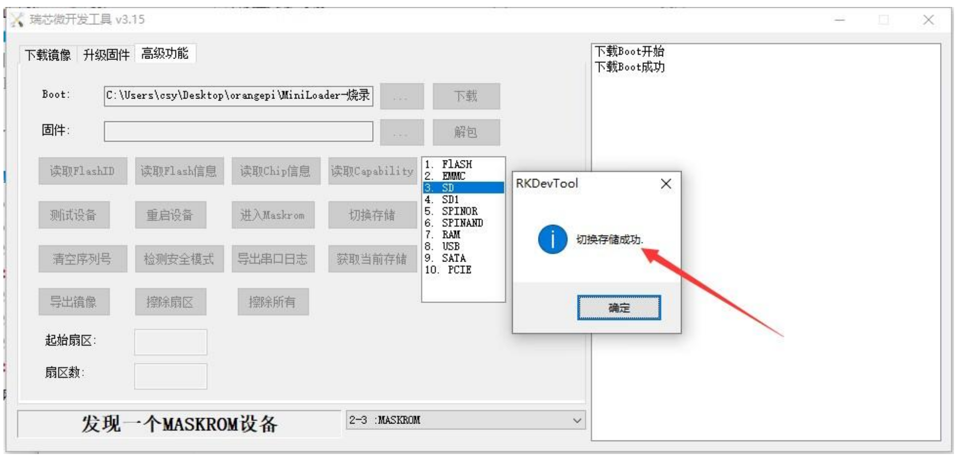
j. After downloading **MiniLoaderAll.bin**, the display is as shown below



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



l. The successful switching is displayed as shown below



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

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

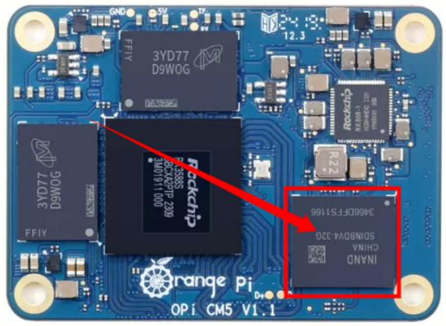
o. Finally, click the "**Upgrade**" button to start burning. The log during the burning process is as shown below. After the burning is completed, the Android system will automatically start.

2. 7. How to burn Android image to eMMC

2. 7. 1. How to use RKDevTool to burn

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

1) There is an eMMC module on the Orange Pi CM5 core board, and its location is as follows:

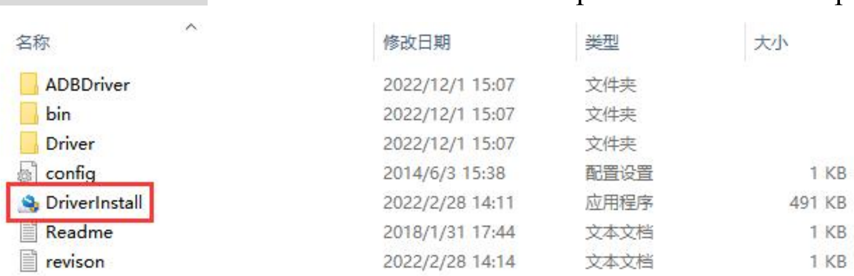


2) You will also need to prepare a good quality USB male-to-male data cable.

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

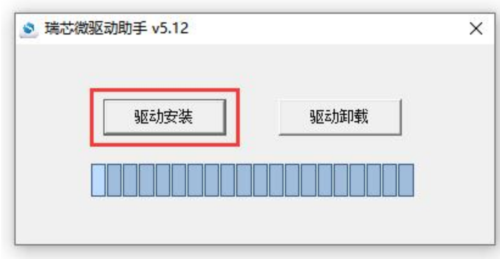
4) Then download the Android image from **Orange Pi's download page**.

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

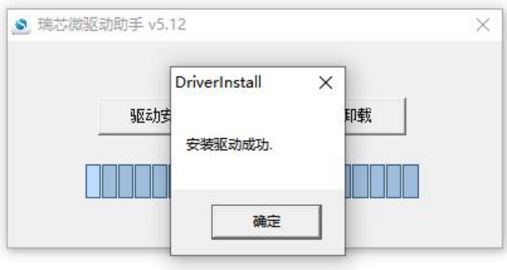


6) After opening **DriverInstall.exe** the steps to install the Rockchip driver are as follows:

a. Click the "Driver Installation" button.



b. After waiting for a while, a window will pop up saying "**Driver installation successful**", then click the "**OK**" button.

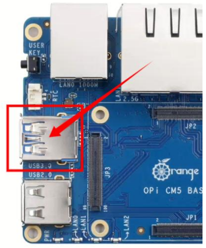


7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

8) After opening the **RKDevTool** burning tool, because the computer is not yet connected to the development board via the Type-C data cable, the lower left corner will prompt "**No device found**".

9) Then start burning the Android image to eMMC.

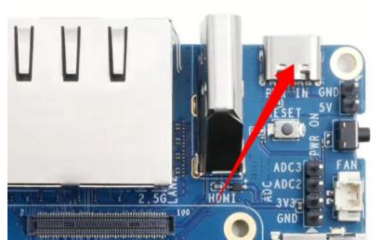
a. First, connect the development board to the Windows computer via a USB male-to-male data cable. The location of the development board's USB burning interface is shown in the figure below:



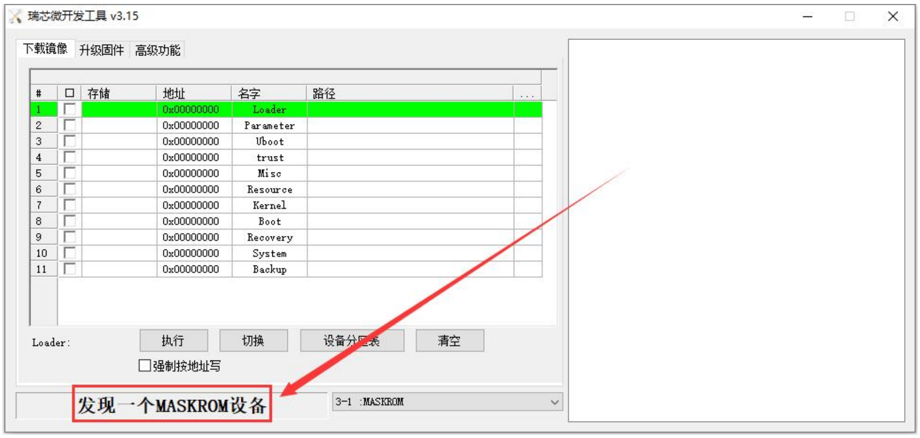
b. Make sure the development board is not connected to the power supply and the TF card is not inserted.

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

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



e. If the previous steps are successful, the development board will enter the **MASKROM** mode and the burning tool interface will prompt "**A MASKROM device is found**".



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

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

h. Finally, click the "Upgrade" button to start burning. The log of the burning process is shown in the figure below. After the burning is completed, the Android system will automatically start.

2. 8. Start the Orange Pi Development Board

1) The core board's eMMC is pre-installed with an image, which we can directly start and use. Or insert the TF card with the image burned into the TF card slot of the Orange Pi development board.

2) The development board has an HDMI interface, which can be connected to a TV or HDMI monitor via an HDMI to HDMI cable.

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

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

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

Many unstable phenomena during the system power-on startup 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.

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

Please do not connect to the USB interface of the computer to power the development board.

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

6) If you want to view the system output information through the debug serial port, please use a serial cable to connect the development board to the computer. For the serial port connection method, please **refer to the section "How to use the debug serial port".**

2.9. How to use the debug serial port

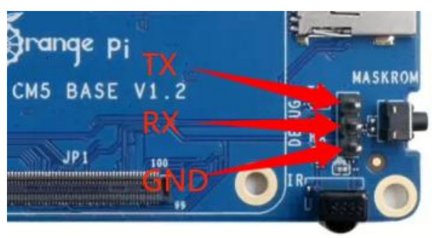
2.9.1. Debug serial port connection instructions

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

For better compatibility, it is recommended to use the CH340 USB to TTL module. Please do not use the CP2102 or PL2303 type USB to TTL modules.

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

2) The corresponding relationship between the GND, RX and TX pins of the debug serial port of the baseboard is shown in the figure below:



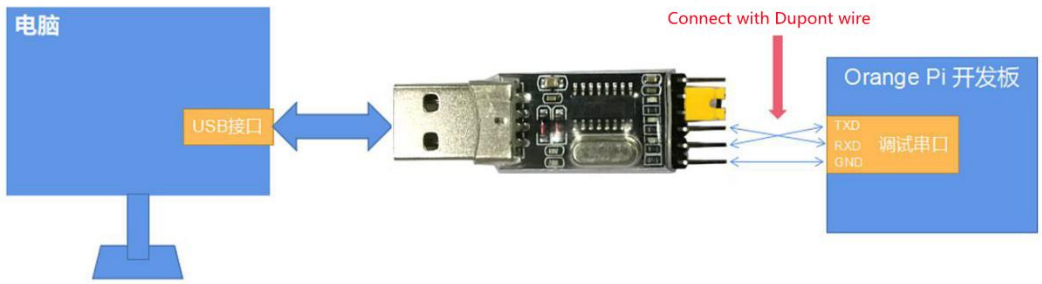
3) The GND, TX and RX pins of the USB to TTL module need to be connected to the debug serial port of the development board via a Dupont cable.

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

b. Connect the **RX of the USB to TTL module 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 Orange Pi development board is as follows:



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

> The TX and RX of the serial port need to be cross-connected. If you don't want to carefully distinguish the order of TX and RX, you can connect the TX and RX of the serial port randomly. If there is no output in the test, swap the order of TX and RX. In this way, there is always one order that is correct.

2. 9. 2. How to use the debug serial port on Ubuntu

> **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 USB to TTL module is connected and recognized normally, you can see the corresponding device node name under **/dev** of the Ubuntu PC. Remember this node name, which will be used when setting up the serial port software later.

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

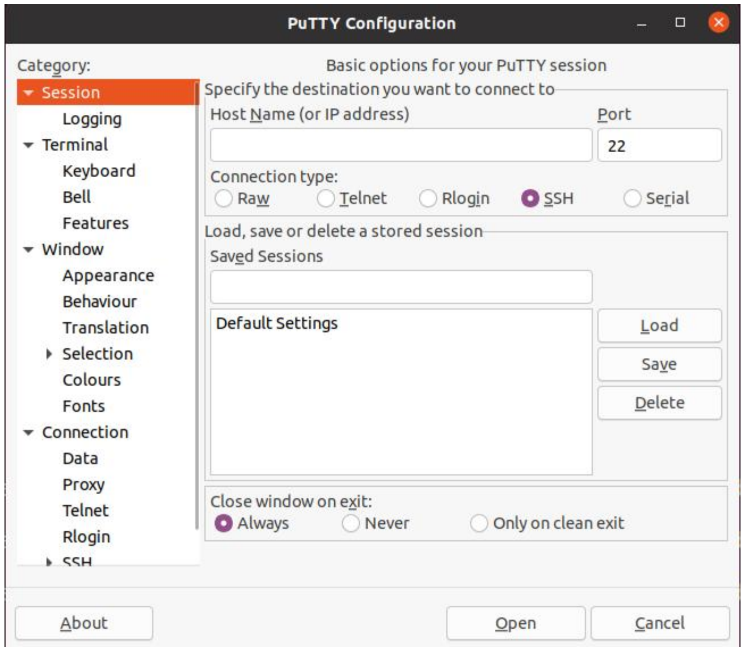
2) Then install putty on your Ubuntu PC using the command below.

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

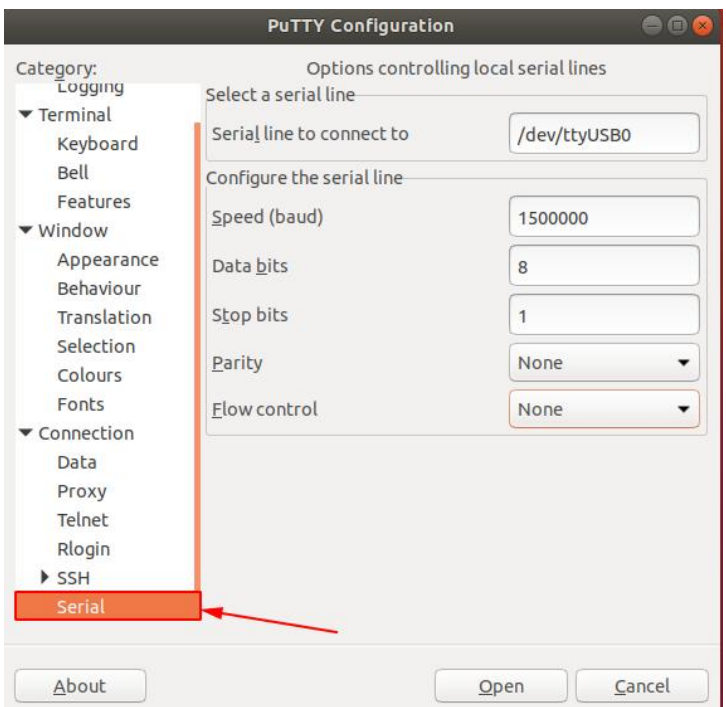
3) Then run putty and remember to add sudo permissions.

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

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



5) First select the serial port settings interface.

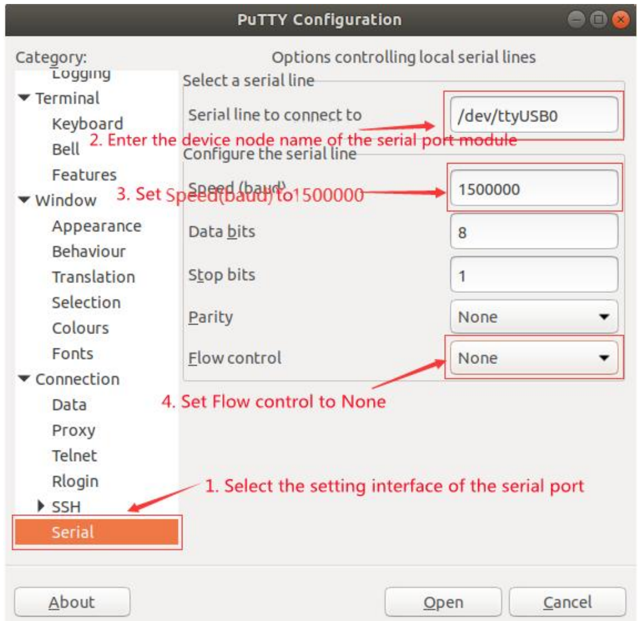


6) Then set the serial port parameters.

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

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

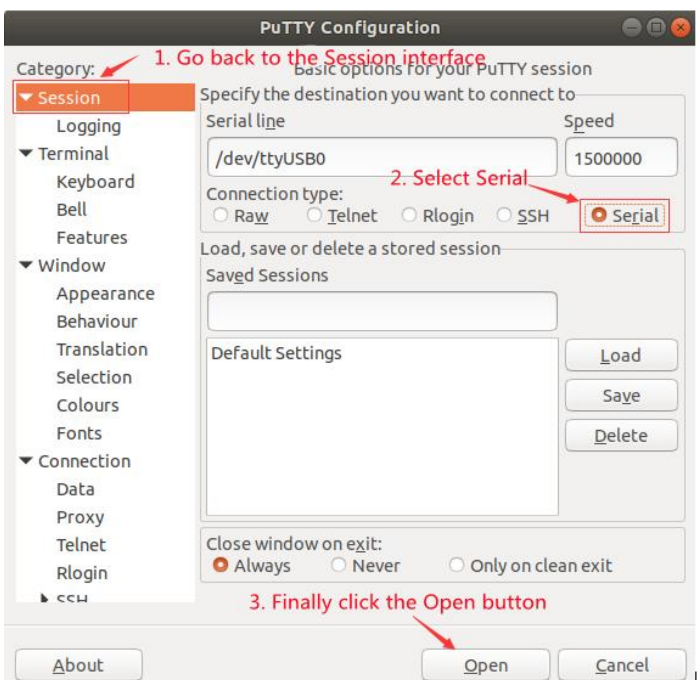
c. Set Flow control to None.



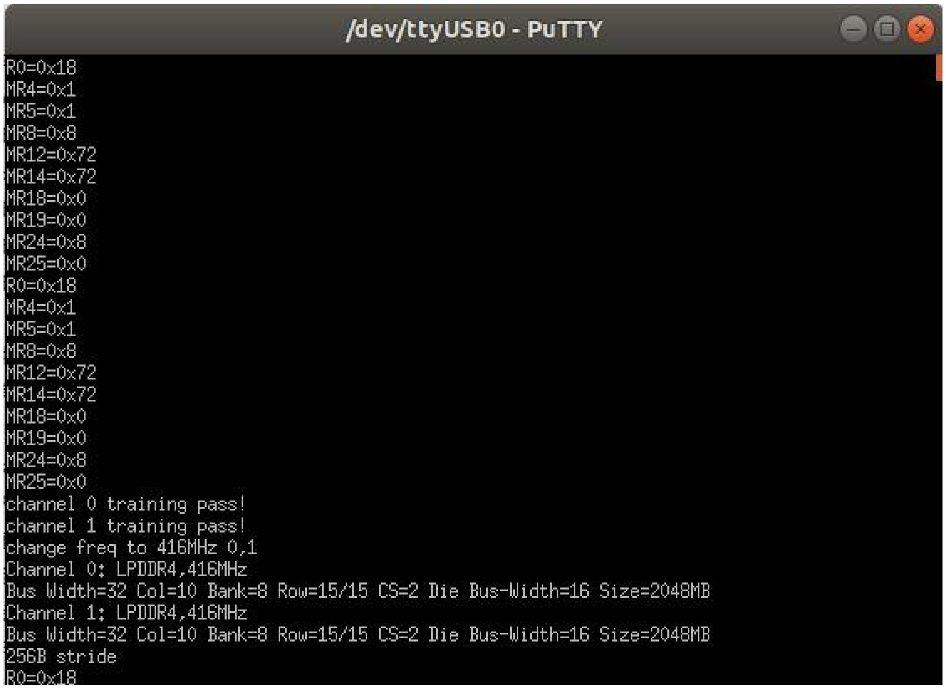
7) After completing the settings on the serial port settings interface, return to the Session interface.

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

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



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



2. 9. 3. How to use the debug serial port on Windows platform

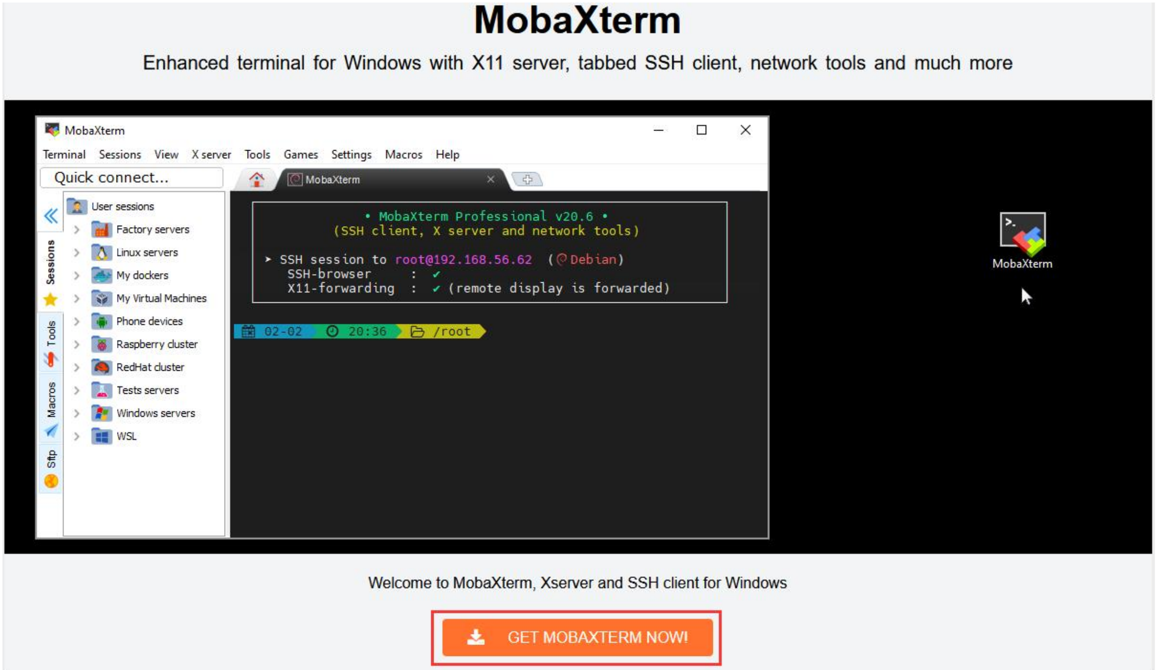
There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without purchasing a serial number.

1) Download MobaXterm.

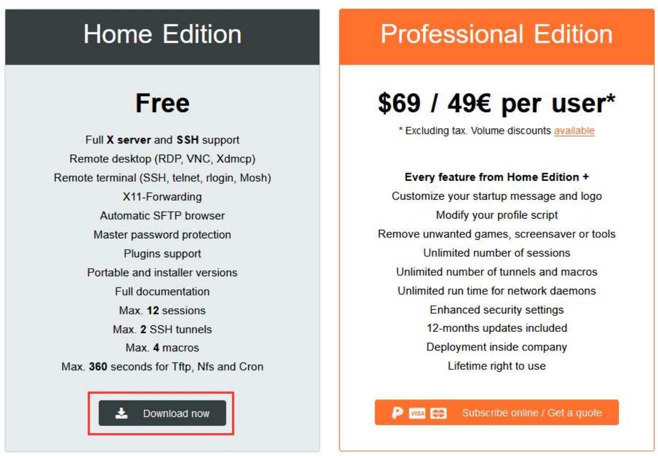
a. Download MobaXterm from the following URL:

https://mobaxterm.mobatek.net

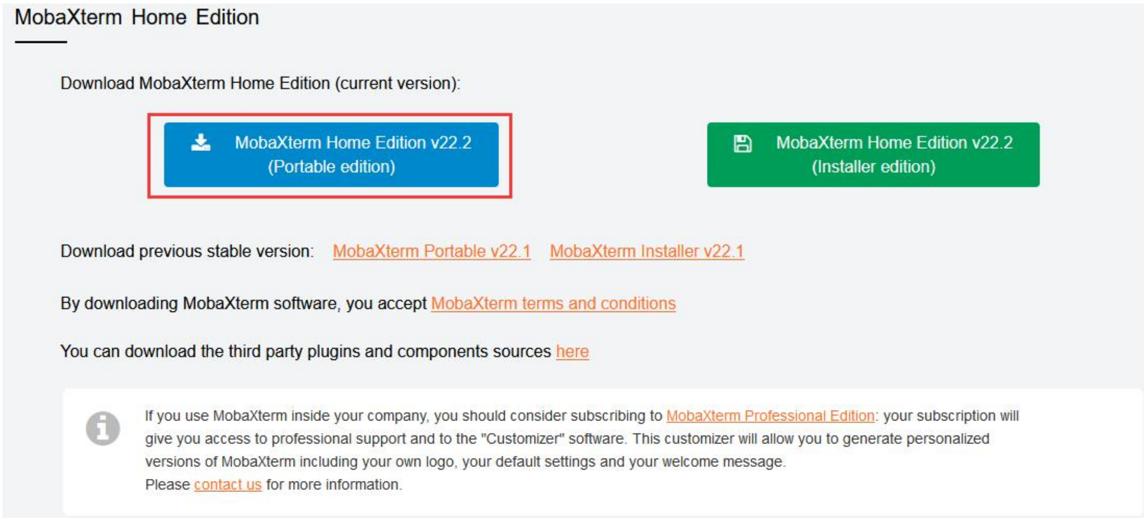
b. Go to the MobaXterm download page and click **GET XOBATERM NOW!**.



c. Then choose to download the Home version.



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



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

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

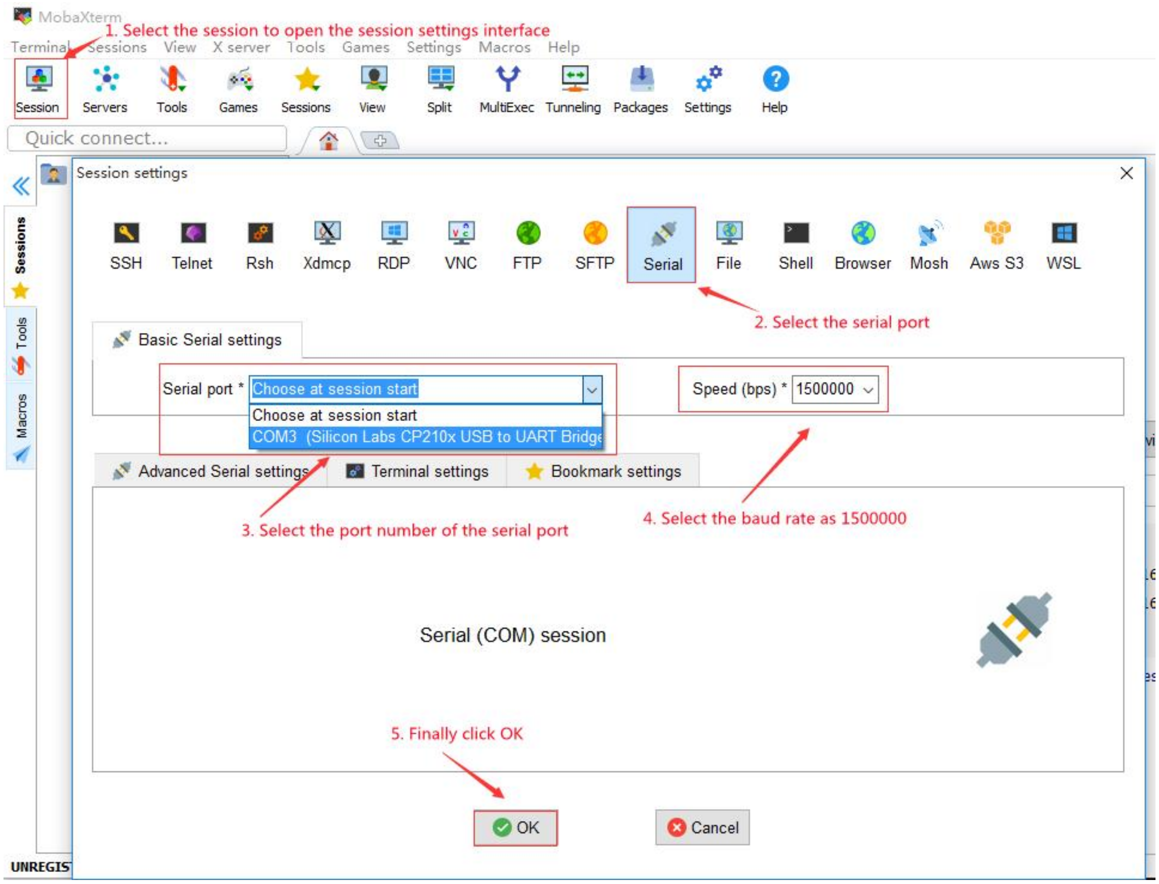
a. Open the session settings interface.

b. Select the serial port type.

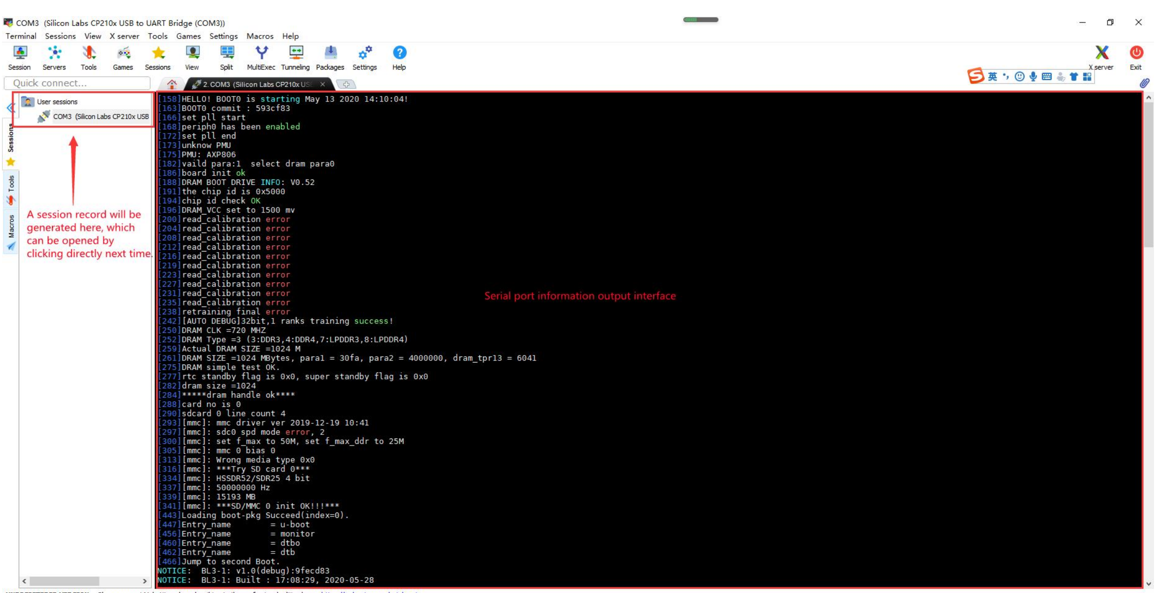
c. Select the serial port number (select the corresponding port number according to the actual situation). If you cannot see the port number, use 360 Driver Master to scan and install the USB to TTL serial port chip driver.

d. Select the serial port baud rate as **1500000**.

e. Finally, click the "**OK**" button to complete the settings.



4) Click the " **OK** " button to enter the following interface. Now start the development board and you can see the output information of the serial port.



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

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

3. 1. Supported Linux image types and kernel versions

Linux Image Type	Kernel version	Server Edition	desktop version
Debian 11 - Bullseye	Linux5.10	Support	Support
Debian 12 - Bookworm	Linux5.10	Support	Support
Ubuntu 20.04 - Focal	Linux5.10	Support	Support
Ubuntu 22.04 - Jammy	Linux5.10	Support	Support
Debian 12 - Bookworm	Linux6.1	Support	Support
Ubuntu 22.04 - Jammy	Linux6.1	Support	Support

3. 2. Linux 5.10 system compatibility

Function	Debian11	Debian12	Ubuntu20.04	Ubuntu22.04
HDMI Display	OK	OK	OK	OK
HDMI Audio	OK	OK	OK	OK
USB 2.0	OK	OK	OK	OK
USB 3.0	OK	OK	OK	OK
Gigabit Ethernet	OK	OK	OK	OK
2.5G LAN x 2	OK	OK	OK	OK
Debug serial port	OK	OK	OK	OK
FAN	OK	OK	OK	OK
RTC	OK	OK	OK	OK
eMMC	OK	OK	OK	OK
GPIO（12pin）	OK	OK	OK	OK
UART（12pin）	OK	OK	OK	OK
SPI（12pin）	OK	OK	OK	OK

I2C（12pin）	**OK**	**OK**	**OK**	**OK**
CAN（12pin）	**OK**	**OK**	**OK**	**OK**
Camera x 4	**OK**	**OK**	**OK**	**OK**
LED Light	**OK**	**OK**	**OK**	**OK**
TF Card Start	**OK**	**OK**	**OK**	**OK**
Infrared	**OK**	**OK**	**OK**	**OK**
GPU	**OK**	**OK**	**OK**	**OK**
NPU	**OK**	**OK**	**OK**	**OK**
VPU	**OK**	**OK**	**OK**	**OK**
Power button	**OK**	**OK**	**OK**	**OK**
Watchdog test	**OK**	**OK**	**OK**	**OK**
Chromium hard decoding video	**OK**	**OK**	**OK**	**OK**

3. 3. Linux 6.1 system compatibility

Function	Debian12	Ubuntu22.04
HDMI Display	**OK**	**OK**
HDMI Audio	**OK**	**OK**
USB 2.0	**OK**	**OK**
USB 3.0	**OK**	**OK**
Gigabit Ethernet	**OK**	**OK**
2.5G LAN x 2	**OK**	**OK**
Debug serial port	**OK**	**OK**
FAN	**OK**	**OK**
RTC	**OK**	**OK**
eMMC	**OK**	**OK**
GPIO（12pin）	**OK**	**OK**
UART（12pin）	**OK**	**OK**
SPI（12pin）	**OK**	**OK**
I2C（12pin）	**OK**	**OK**
CAN（12pin）	**OK**	**OK**

Camera x 4	**OK**	**OK**
LED Light	**OK**	**OK**
TF Card Start	**OK**	**OK**
Infrared	**OK**	**OK**
GPU	**OK**	**OK**
NPU	**OK**	**OK**
VPU	**OK**	**OK**
Power button	**OK**	**OK**
Watchdog test	**OK**	**OK**
Chromium hard decoding video	**OK**	**OK**

3. 4. Linux command format description in this manual

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

```

```

As shown below, the contents in the yellow box indicate the contents that require special attention, except for the commands inside.

```

```

2) Description of the prompt type before the command.

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

```
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 common user. When executing privileged commands, **sudo** is required.

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

d. **test@test:~$** The prompt indicates that this command is entered in an 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 a normal user. When executing privileged commands, you need to add **sudo**.

e. **root@test:~#** The prompt indicates that this command is entered in an 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 bold black part is the command that needs to be entered, and the content below the command is the output (some commands have output, some may not). This part does not need to be entered.

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

b. As shown below, some commands cannot fit in one line and will be placed on the next line. The bold black parts are the commands that need to be entered. When these commands are entered on one line, the "\" at the end of each line needs to be removed, as it is not part of the command. In addition, there are spaces between different parts of the command, so please do not miss them.

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

3. 5. Linux system login instructions

3. 5. 1. Linux system default login account and password

Account	Password
root	orangepi
orangepi	orangepi

Please note that when you enter the password, the specific content of the password will not be displayed on the screen. Please do not think that there is any malfunction. Just press Enter after entering it.

3. 5. 2. How to set up automatic login for Linux system terminal

1) The Linux system automatically logs into the terminal by default, and the default login username is **orangepi**.

```
orangepicm5 login: orangepi (automatic login)
```

```
Welcome to Orange Pi 1.0.0 Jammy with Linux 5.10.160-rockchip-rk3588

System load:   19%              Up time:       0 min
Memory usage:  11% of 3.83G     IP:            192.168.2.231
CPU temp:      45°C             Usage of /:    9% of 57G

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

Last login: Wed May 29 20:58:50 CST 2024 on tty1
orangepi@orangepicm5:~$
```

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 to the terminal.

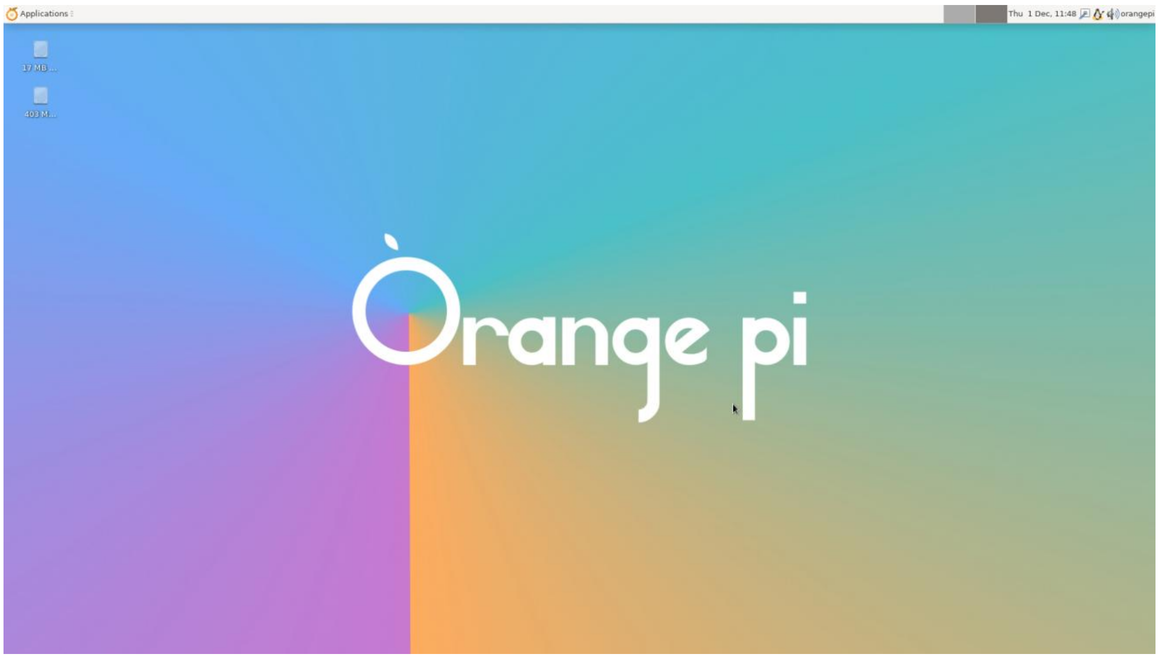
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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. 5. 3. Linux desktop system automatic login instructions

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



2) Run the following command to disable the desktop version of the 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. You need to enter the password to enter the system.

3.5.4. How to set up automatic login for root user in Linux desktop 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 you will automatically log in to the desktop as the root user.

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

Also, please note that this is not a bug, because pulseaudio is not allowed to run under the root user.

3) Run the following command to set the desktop system to automatically log in using the orangepi user again.

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

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

1) First enter the following command in the command line. Please remember to **add sudo permissions.**

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

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

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

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

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

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

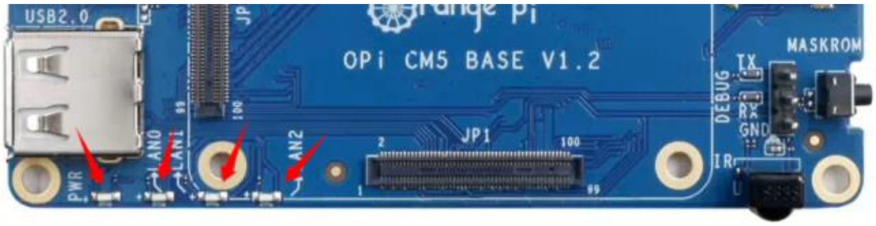
b. After making your selection, the monitor will display the desktop.

3. 6. Onboard LED Light Test Instructions

1) There is a red light and three green lights on the bottom panel, and their locations are shown in the figure below:

a. The leftmost light with PWR printed on it is red;

b. The other three lights are all green.



2) As long as the development board is powered on, the red LED light will be always on. This is controlled by hardware and cannot be turned off by software. The red LED light can be used to determine whether the power of the development board has been turned on normally.

3) The three green LED lights are the indicator lights for the Gigabit Ethernet port and the two 2.5G Ethernet ports. When the Ethernet port is linked, the corresponding green light will be on, and when there is data transmission, the corresponding green light will flash.

3. 7. Network connection test

3. 7. 1. Ethernet port test

1) First, plug one end of the network cable into the Ethernet port of the development board, and the other end of the network cable into the router, and make sure the network is unobstructed.

2) After the system starts, the IP address will be automatically assigned to the Ethernet card through**DHCP**, and no other configuration is required.

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

Note that in the following commands, Debian 12 needs to change eth0 to end1.

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

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

```
orangepi@orangepi:~$ ifconfig
Command 'ifconfig' is available in the following places
 * /sbin/ifconfig
 * /usr/sbin/ifconfig
```

The command could not be located because '/sbin:/usr/sbin' is not included in the PATH environment variable.
This is most likely caused by the lack of administrative privileges associated with your user account.
ifconfig: command not found

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

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

3.7.2. WIFI connection test

First of all, please note that there is no WIFI module on the development board. You need an external USB network card to use the WIFI function.

For instructions on using the external USB network card, please refer to the USB wireless network card test section.

Please do not connect to WIFI by modifying the /etc/network/interfaces configuration file. This method may cause problems when connecting to the WIFI network.

3.7.2.1. Server version image connects to WIFI through command

When the development board is not connected to Ethernet, not connected to HDMI display, and only connected to the serial port, it is recommended to use the command demonstrated in this section to connect to the WIFI network. Because

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

1) Log in to the Linux system first. There are three ways:
   a. If the development board is connected to the network cable, you can log in to the Linux system remotely through SSH.
   b. If the development board is connected to the debug serial port, you can log in to the Linux system using the serial terminal.
   c. If the development board is connected to an HDMI display, you can log in to the Linux system through the HDMI display terminal.

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

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

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

2) Then use the **nmcli**command to connect to the scanned WIFI hotspot, where:
   a. **wifi_name** You need to change it to the name of the WIFI hotspot you want to connect to.
   b. **wifi_passwd**You need to change it to the password of the WIFI hotspot you want to connect to.

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

3) Through **ip addr show wlan0** The command can view the IP address of the wifi.

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

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

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

3. 7. 2. 2. The server version image connects to WIFI through a graphical method

1) Log in to the Linux system first. There are three ways:

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

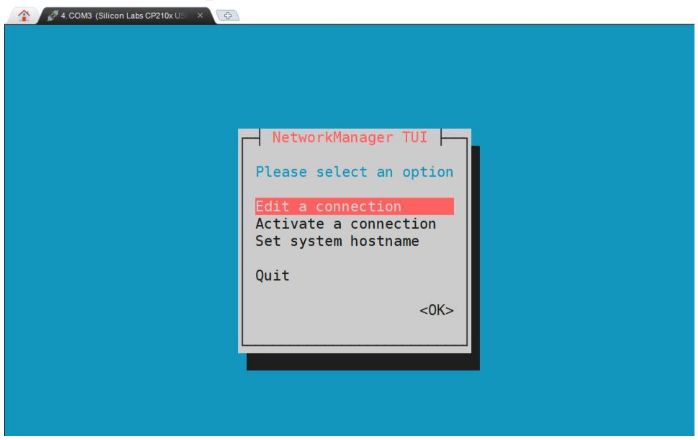
b. If the development board is connected to the debug serial port, you can use the serial terminal to log in to the Linux system (use MobaXterm as the serial software, minicom cannot display the graphical interface).

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

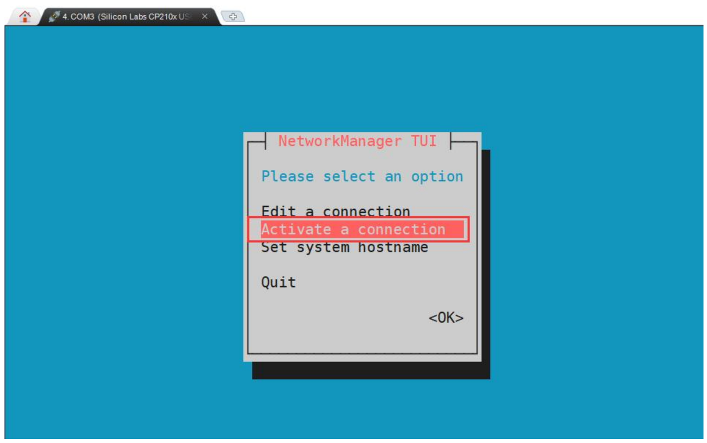
2) Then enter the nmtui command in the command line to open the wifi connection interface.

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

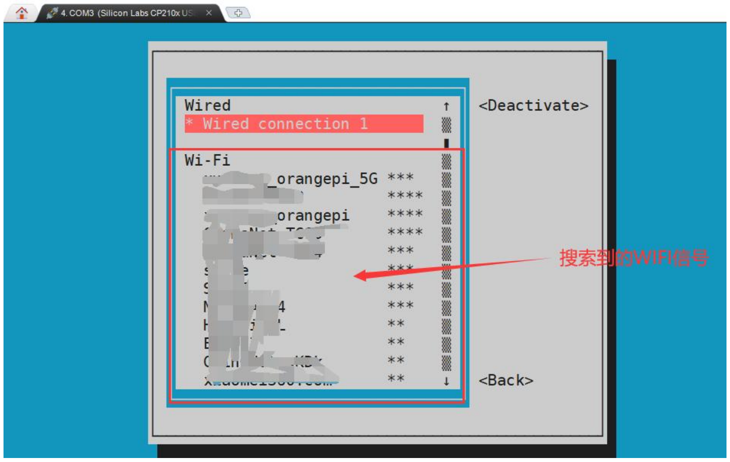
3) Enter the nmtui command to open the interface as shown below:



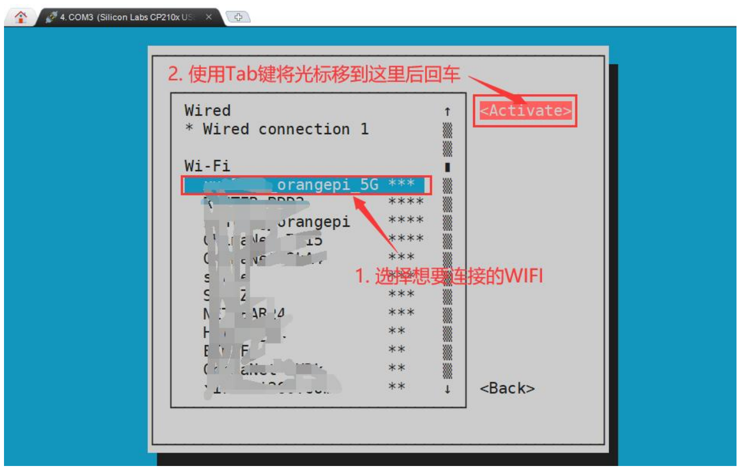
4) Select **Activate a connect** and press Enter.



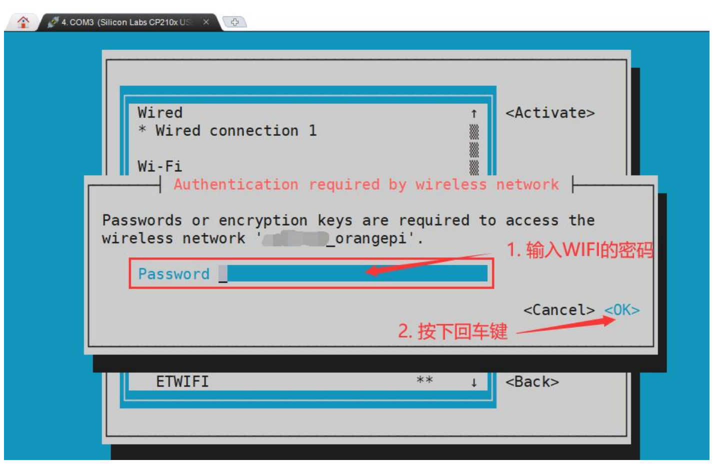
5) Then you can see all the searched WIFI hotspots.



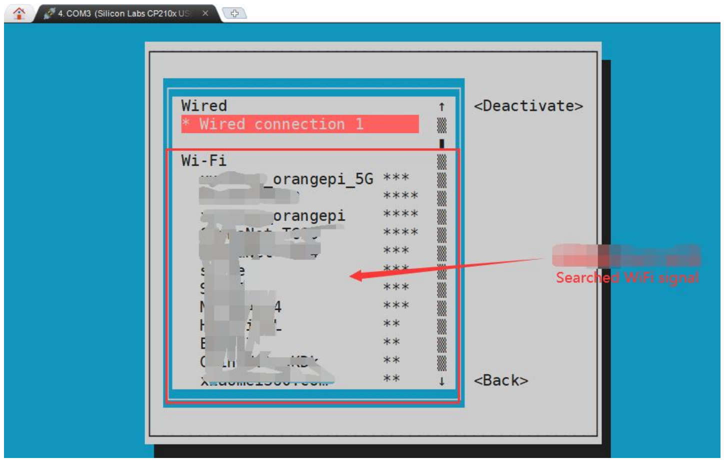
6) Select the WIFI hotspot you want to connect to, then use the Tab key to move the cursor to **Activate** and press Enter.



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



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



9) You can view the IP address of the wifi network by using the **ip addr show wlan0** command.

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

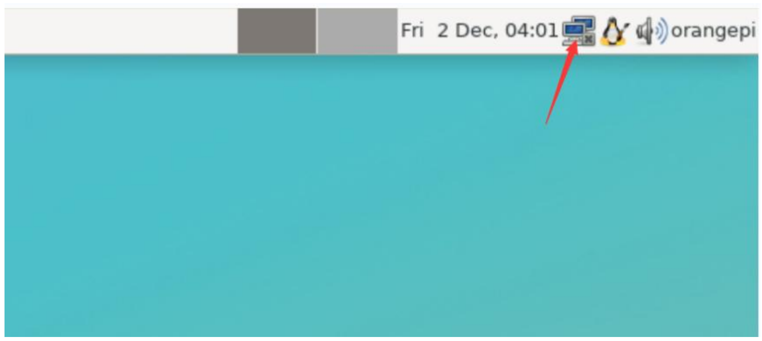
10) Use the **ping**command to test the connectivity of the WiFi network. The **ping** command can be interrupted by pressing the **Ctrl+C** shortcut key.

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

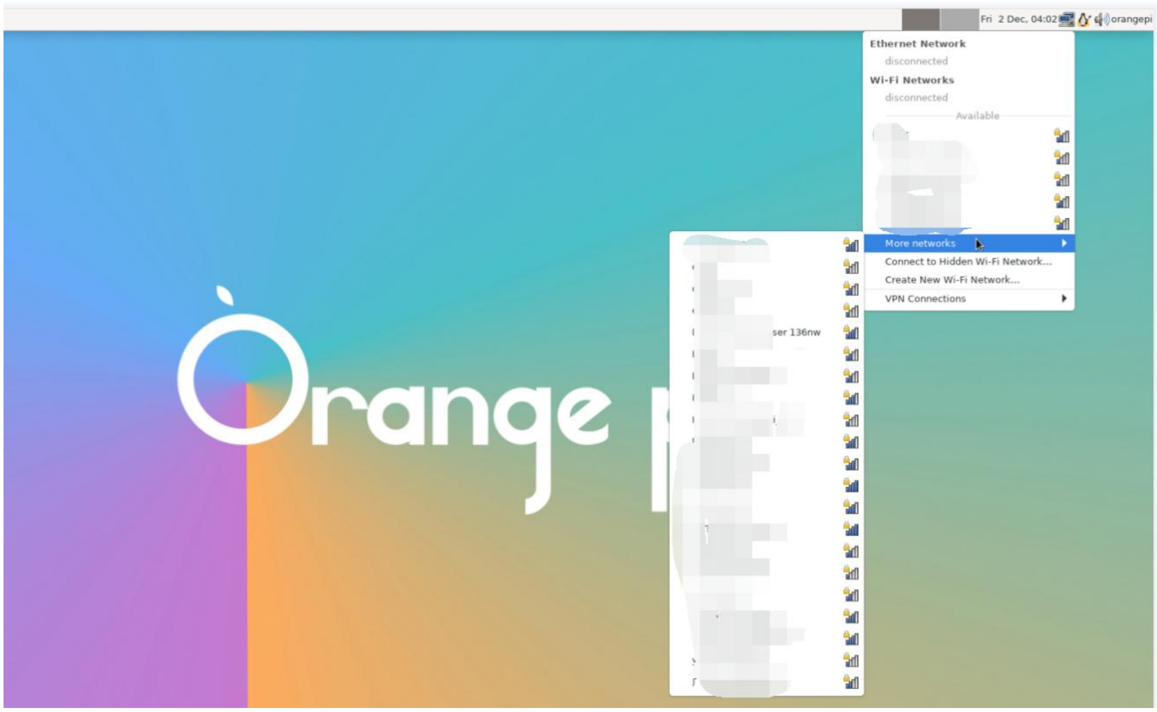
```
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=4 ttl=52 time=45.6 ms
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=5 ttl=52 time=48.8 ms
^C
--- www.orangepi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

3. 7. 2. 3. Testing methods for desktop images

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

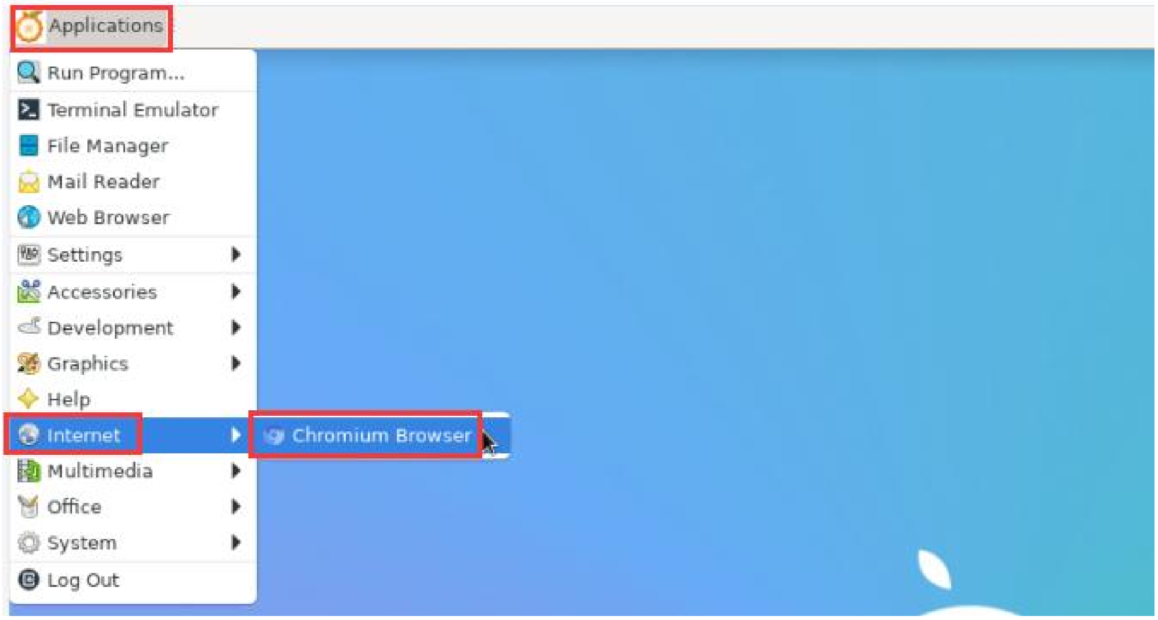


2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to.



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

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



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

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

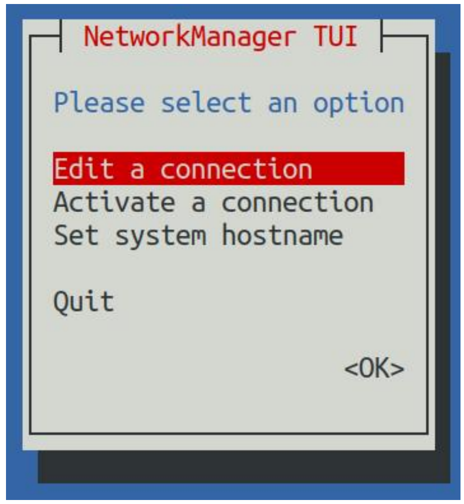
Please do not set a static IP address by modifying the /etc/network/interfaces configuration file.

3. 7. 3. 1. Using nmtui command to set static IP address

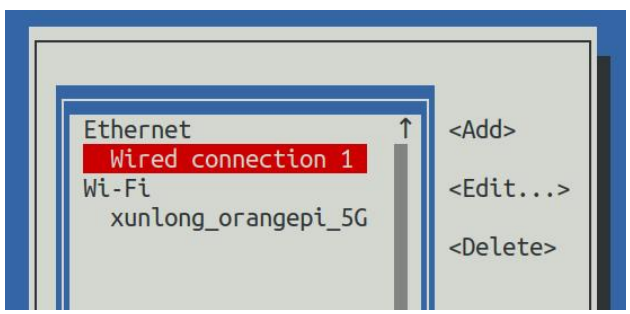
1) First run the **nmtui**command.

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

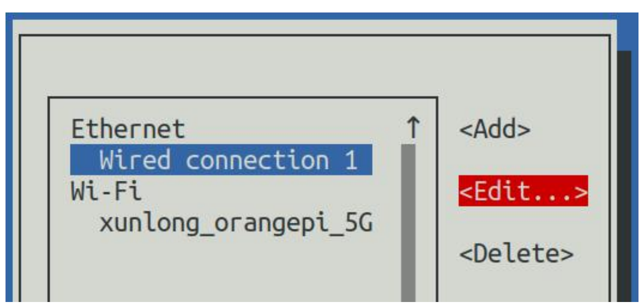
2) Then select **Edit a connection** and press Enter.



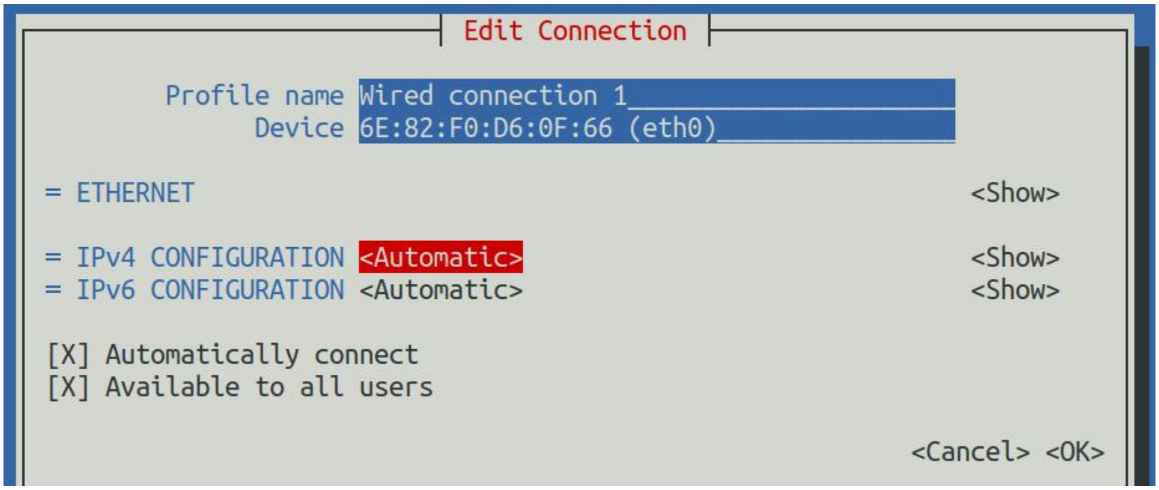
3) Then select the network interface for which you want to set a static IP address. For example, to set a static IP address for an **Ethernet** interface, just select **Wired connection 1**.



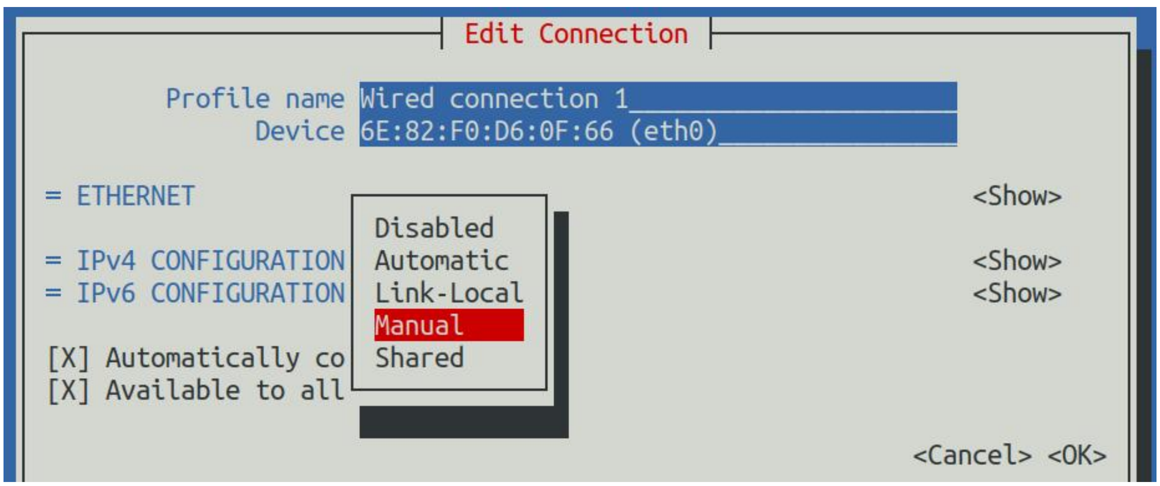
4) Then select **Edit**using the **Tab** key and press Enter.



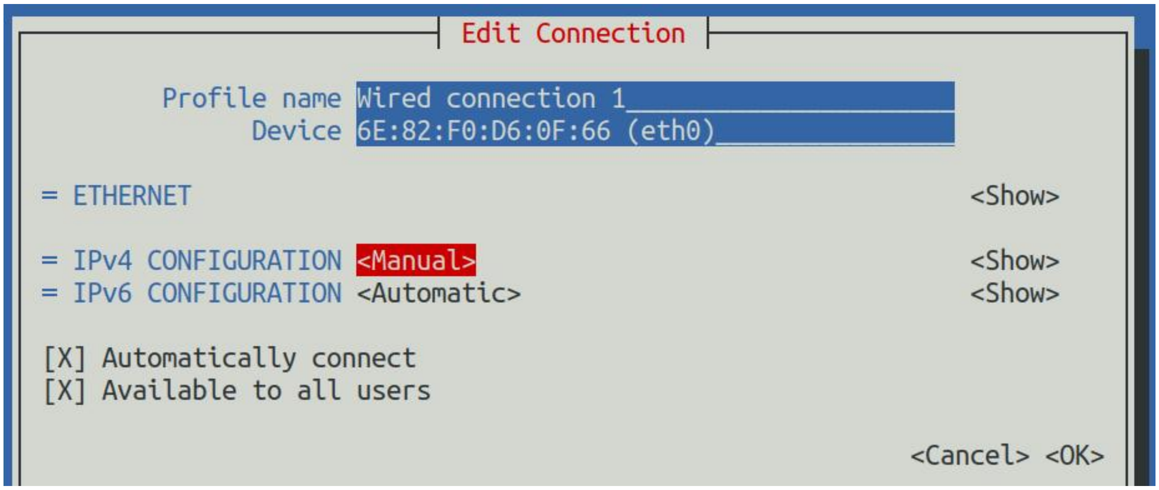
5) Then use the Tab key to move the cursor to the **<Automatic>** position shown in the figure below to configure IPv4.



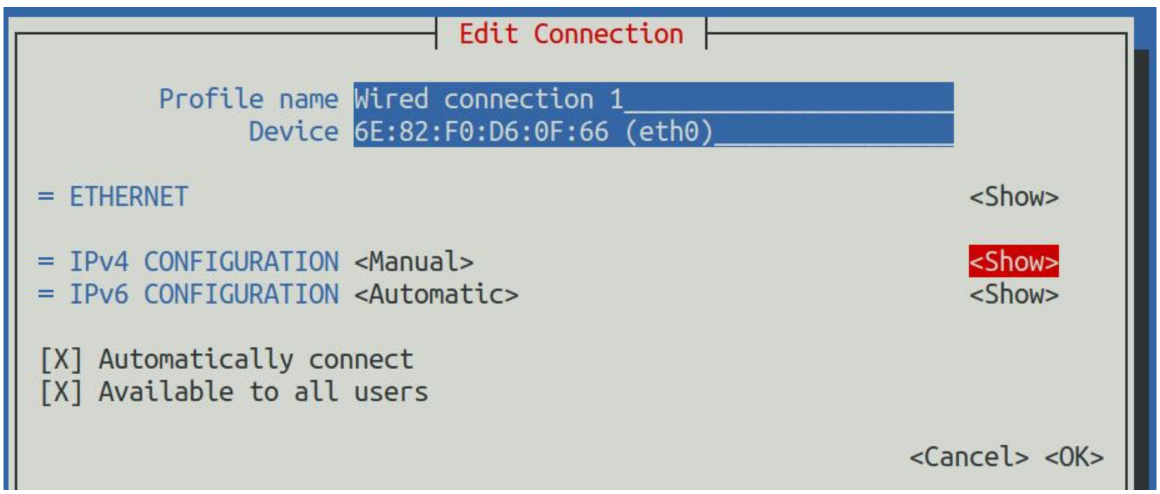
6) Press Enter, use the up and down arrow keys to select **Manual**, and then press Enter to confirm.



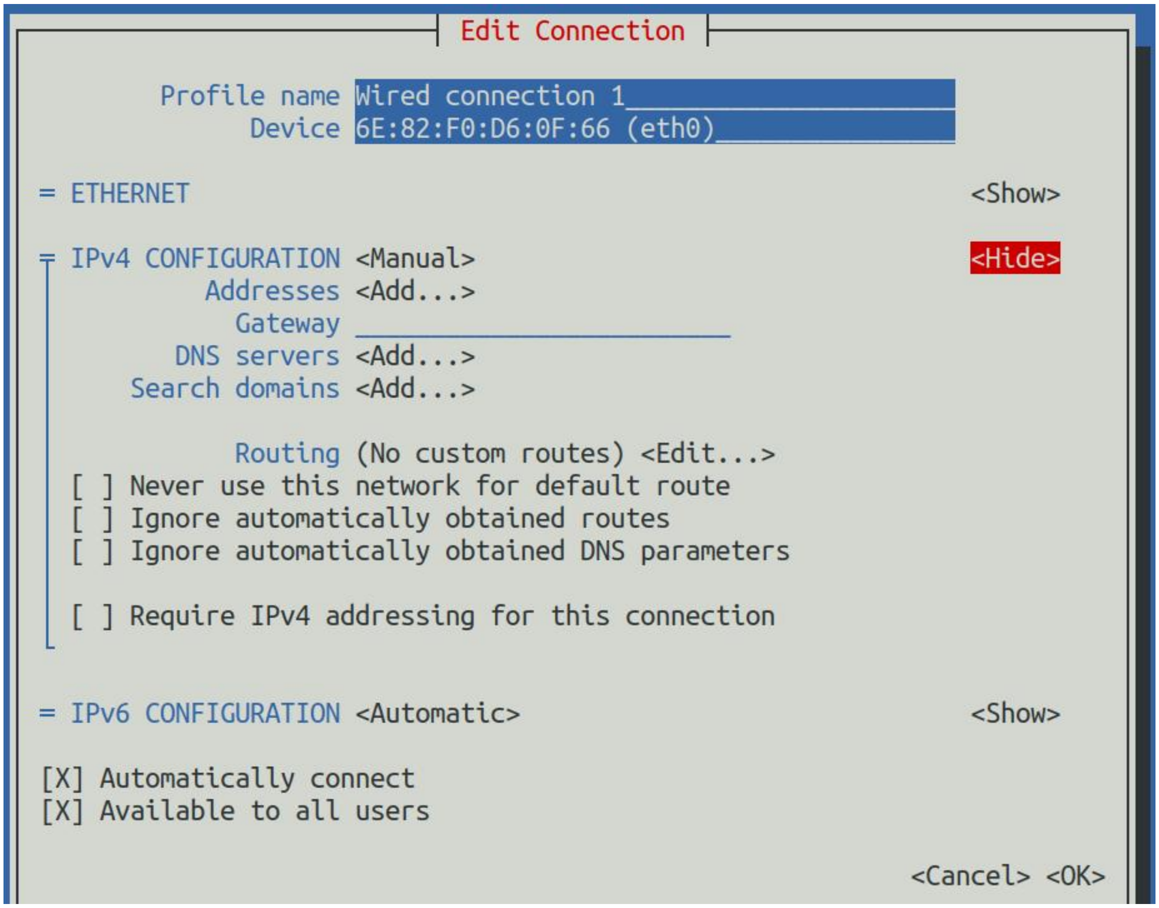
7) After the selection is completed, the display is as shown below:



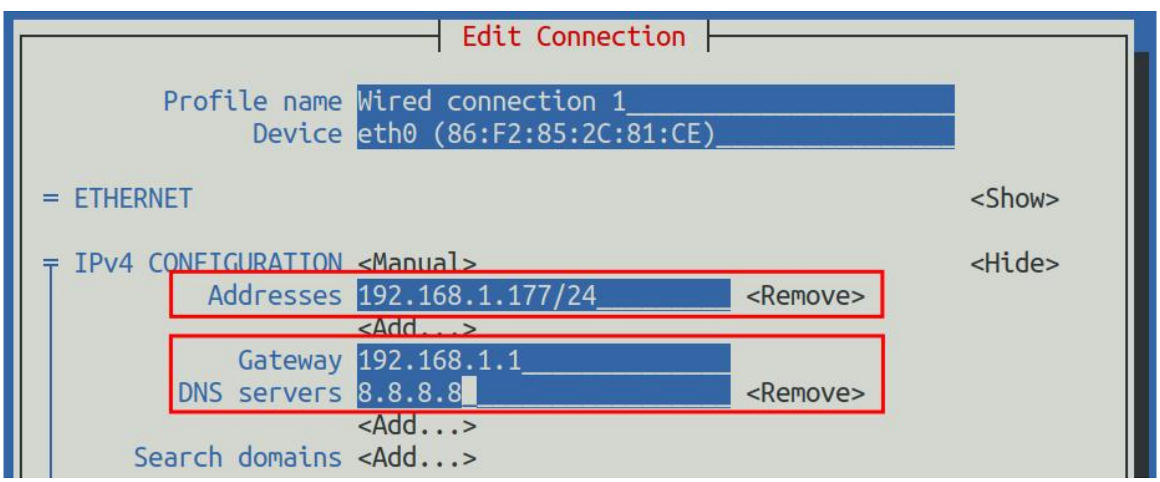
8) Then use the Tab key to move the cursor to **<Show>**.



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

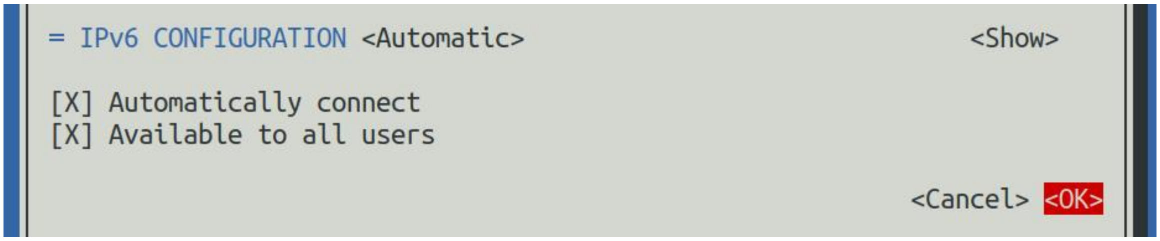


10) Then you can set the IP address, gateway and DNS server address as shown in the figure below (there are many other setting options, please explore them yourself). Please set them according to your specific needs. The value set in the figure below is just an example.

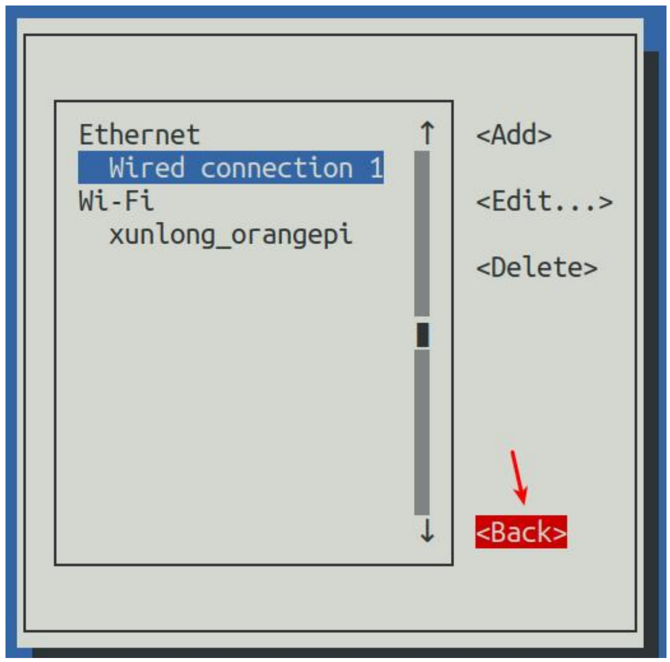


11) After setting, move the cursor to **<OK>** in the lower right corner and press Enter to

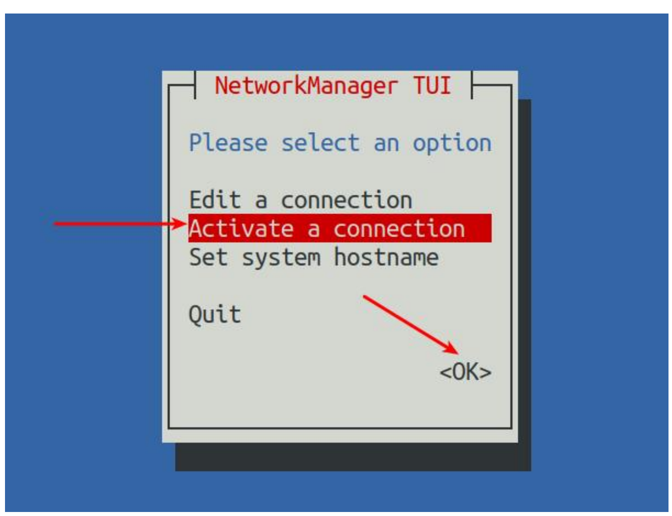
confirm.



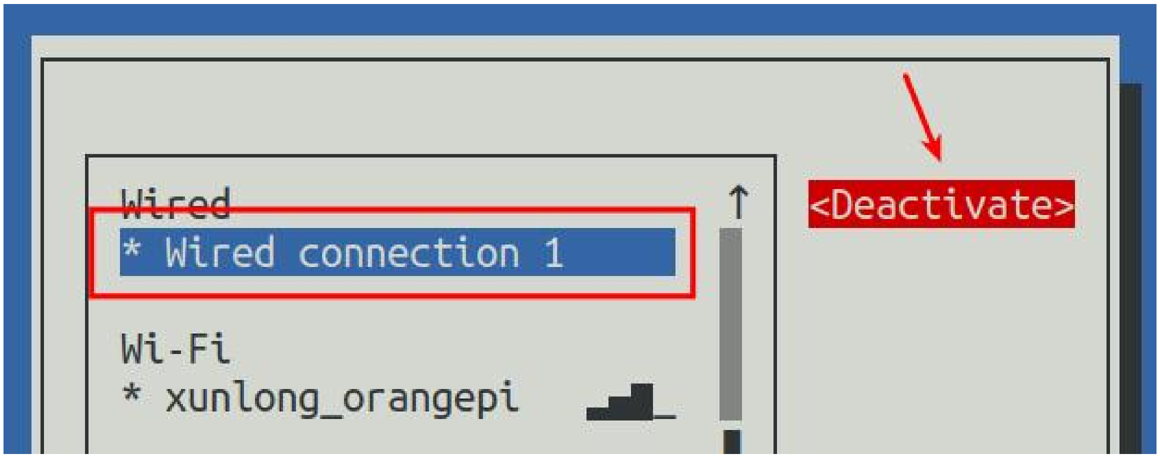
12) Then click **<Back>** to return to the previous selection interface.



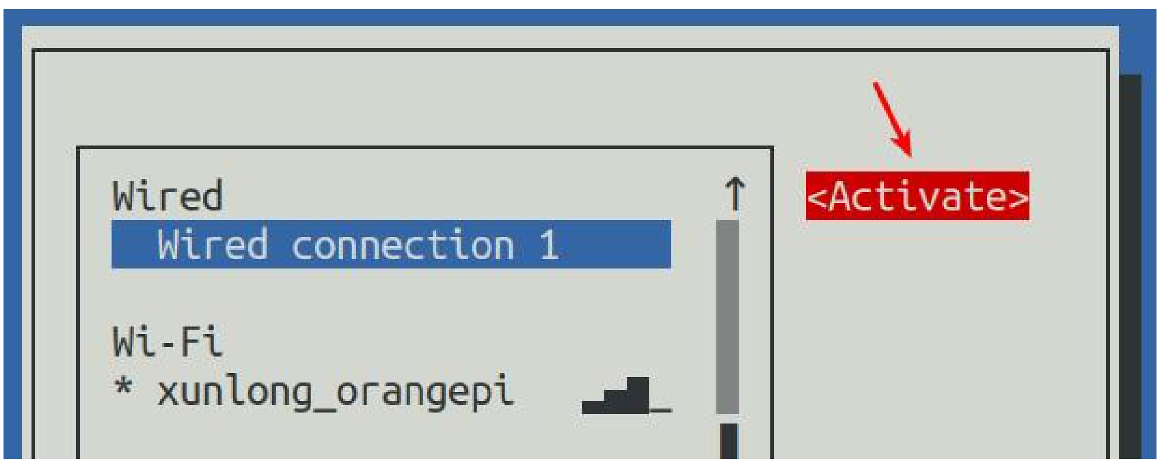
13) Then select **Activate a connection**, move the cursor to **<OK>**, and finally press Enter.



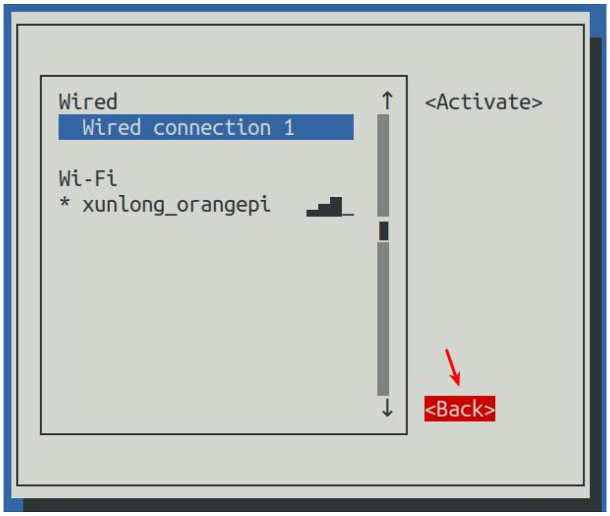
14) Then select the network interface you want to configure, such as **Wired connection 1**, and then move the cursor to **<Deactivate>** and press Enter to disable **Wired connection 1**.

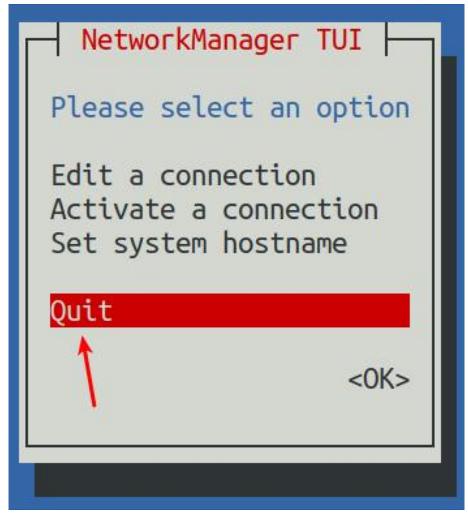


15) Then please do not move the cursor and press the Enter key to re-enable **Wired connection 1**, so that the static IP address set previously will take effect.



16) Then you can exit nmtui by pressing the **<Back>** and **Quit** buttons.





17) Then use i **ip addr show eth0** to see that the IP address of the network port has become the static IP address set earlier.

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

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

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

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

3. 7. 3. 2. Using nmcli to set a static IP address

1) If you want to set a static IP address for the network port, please plug the network cable into the development board first, and then start setting the static IP address.

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

```
root@orangepi:~# nmcli con show
NAME                UUID                                  TYPE      DEVICE
Wired connection 3  9b60145e-35a0-336d-92e3-e3f14a12633b  ethernet  eth0
Wired connection 1  5af6324c-922c-3186-a76c-35aa4afcf560  ethernet  --
Wired connection 2  505a47f7-60e4-37d0-aaf7-d666b34cff8a  ethernet  --
```

3) Then enter the following command, where

a. "**Wired connection 3**" means setting the static IP address of the Gigabit Ethernet port. If you need to set the static IP address of other network ports, please change it to the corresponding name of other network interfaces.

b. The static IP address to be set after **ipv4.addresses** can be changed to the value you want to set.

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

```
orangepi@orangepi:~$ nmcli con mod "Wired connection 3"  \
ipv4.addresses "192.168.1.110"  \
ipv4.gateway "192.168.1.1"  \
ipv4.dns "8.8.8.8"  \
ipv4.method "manual"
```

4) Then restart the Linux system.

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

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

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

3. 8. SSH remote login to the development board

By default, Linux systems enable ssh remote login and allow root users to log in to the system. Before logging in through ssh, you must first ensure that the 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 to the development board under Ubuntu

1) Get the IP address of the development board.

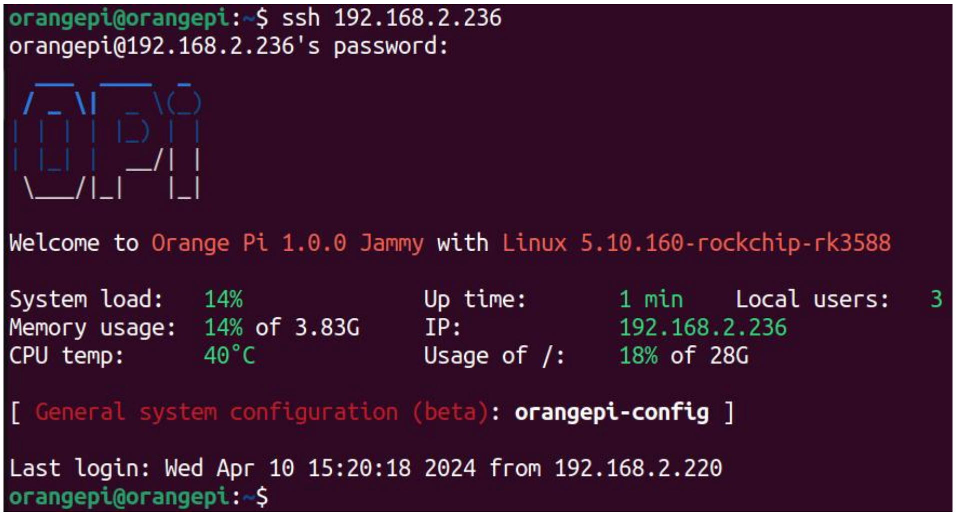
2) Then you can log in to the Linux system remotely through the ssh command.

```
test@test:~$ ssh root@192.168.x.xxx          (Need to be replaced with the IP address
of the development board)
root@192.168.x.xx's password:          （Enter the password here. The default password
is orangepi）
```

Note that when you enter the password, the specific content of the password you entered will not be displayed on the screen. Please do not think that there is any problem. Just press Enter after entering it.

If the prompt refuses to connect, as long as you are using the image provided by Orange Pi, please do not doubt whether the password orangepi is wrong, but look for other reasons.

3) After successfully logging into the system, the display is as shown below:



If ssh cannot log in to the Linux system normally, first check whether the IP address of the development board can be pinged. If the ping is successful, 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 to try to connect:

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

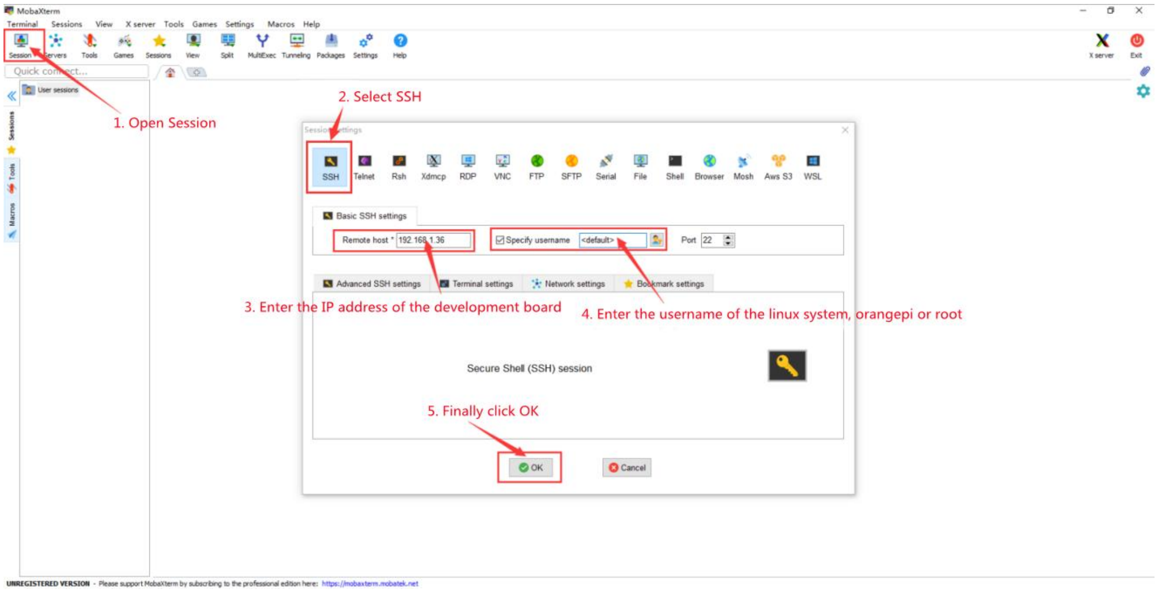
If it still doesn't work, please re-burn the system and try again.

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

1) First, obtain the IP address of the development board.

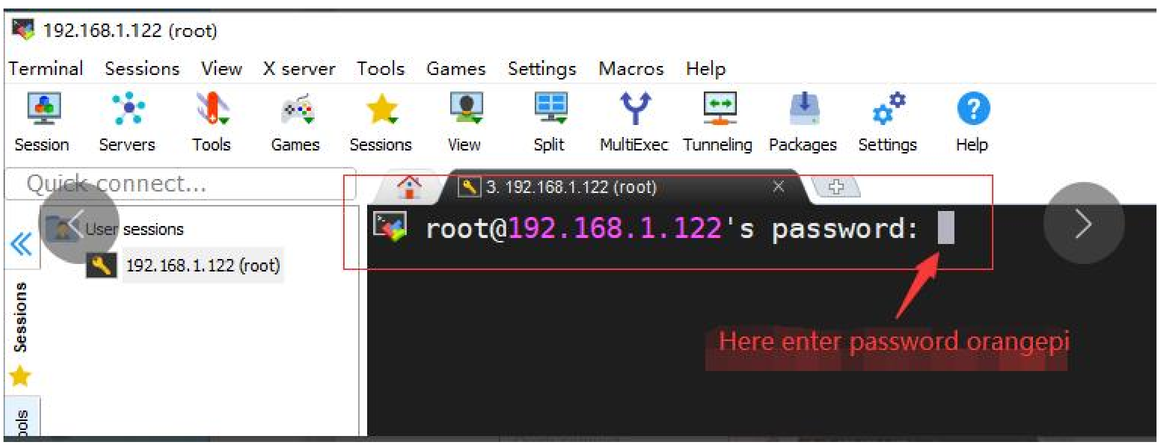
2) You can use MobaXterm to remotely log in to the development board under Windows. First, create a new ssh session.

a. Open **Session**.
b. Select **SSH** in **Session Setting**.
c. Enter the **IP** address of the development board in **Remote host**.
d. Enter the Linux user name **root** or **orangepi** in **Specify username**.
e. Finally, click **OK**.

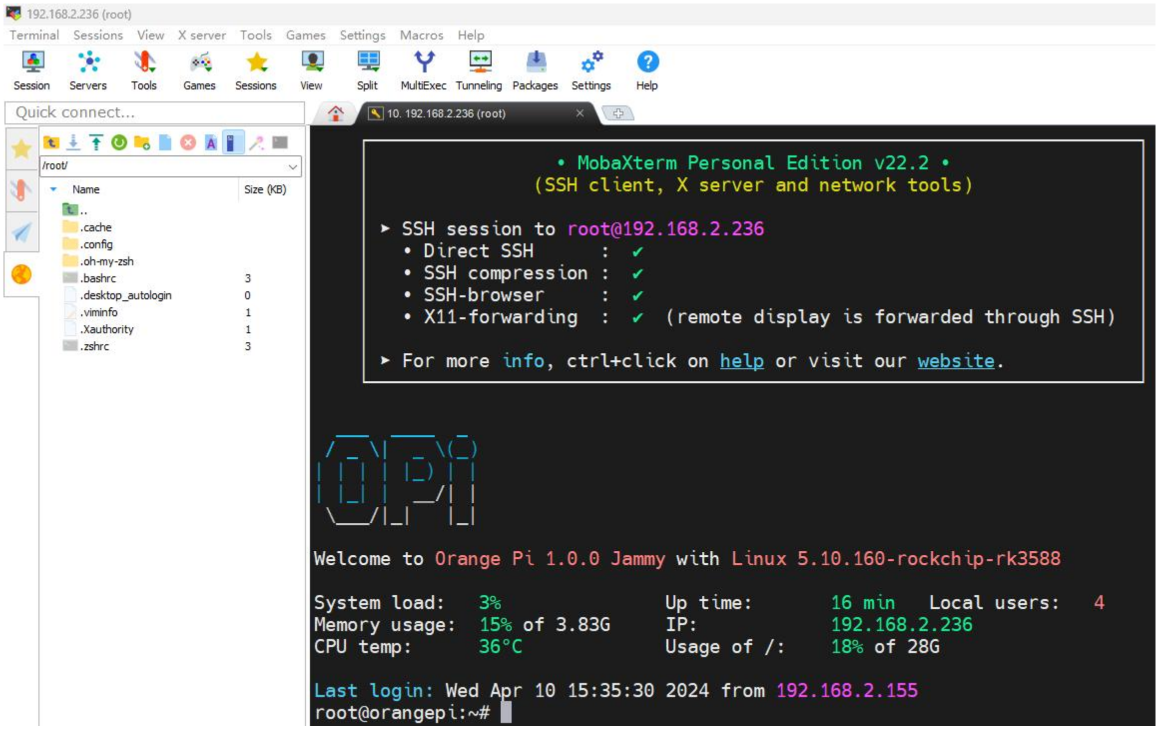


3) You will be prompted to enter a password. The default password for both root and orangepi users is orangepi.

Please note that when you enter the password, the specific content of the password will not be displayed on the screen. Please do not think that there is any malfunction. Just press Enter after entering it.



4) After successfully logging into the system, the display is as shown below:



3. 9. How to use ADB

3. 9. 1. How to use network adb

1) After the system starts, please make sure that **adbd** has been started.

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

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

3) Then install adb tool on your Ubuntu PC.

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

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

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

```
* daemon started successfully
connected to 192.168.1.xx:5555
test@test:~$ adb devices
List of devices attached
192.168.1.xx:5555           device
```

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

```
test@test:~$ adb shell
root@orangepicm5:/#     <--- Seeing this prompt means you have successfully logged into the development board
```

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

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

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

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

If you do not have the adb tool in your Windows system, you can use the adb program in the RKDevTool software.

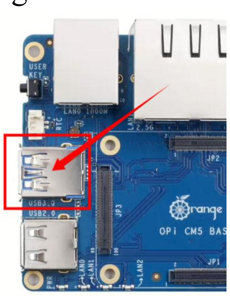
An example using adb in Windows is shown below:

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

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

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

 C:\Users\Administrator\Desktop\RKDevTool_Release_v2.92\bin 的目录

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

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

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


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

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

3.9.2. Use a USB2.0 male-to-male data cable to connect to adb

1) First, prepare a good quality USB male-to-male data cable.

2) Then connect the development board to the Ubuntu PC via a USB male-to-male data cable. The location of the USB interface that supports the device function on the development board is shown in the figure below:



3) Then run the following command to set the USB interface to **device** mode.

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

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

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

4) Then please make sure adbd is started.

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

5) Then install the adb tool on your Ubuntu PC.

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

6) Then use the following command to check whether the adb device is recognized.

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

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

```
test@test:~$ adb shell
root@orangepicm5:/#     <--- Seeing this prompt means you have successfully logged into the development board
```

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

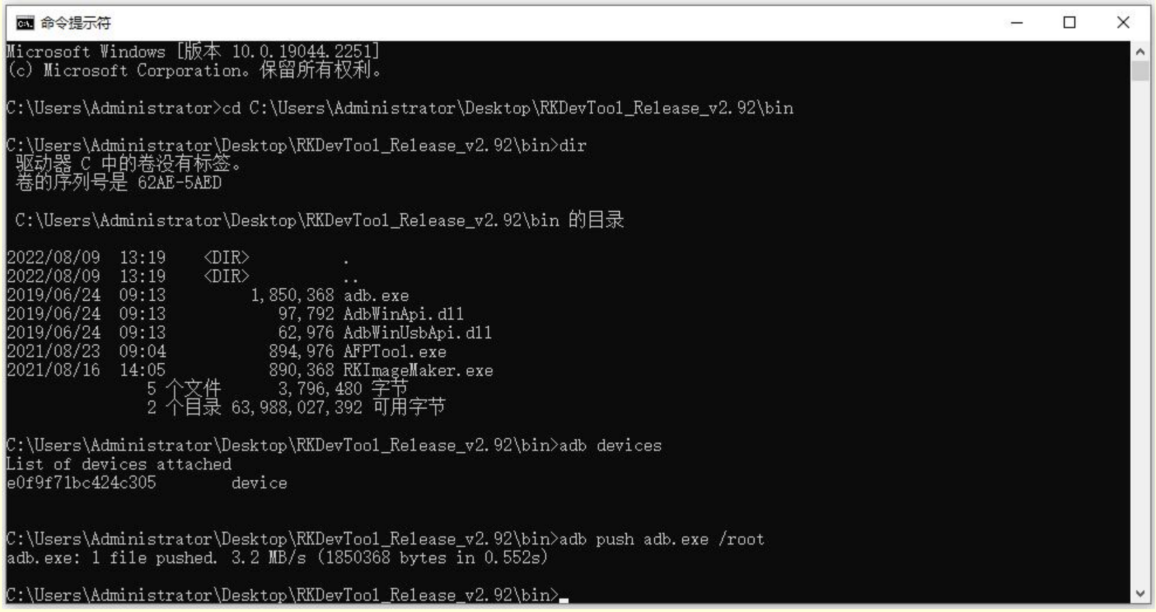
```
test@test:~$ adb push filename /root
filename: 1 file pushed. 3.7 MB/s (1075091 bytes in 0.277s)
```

10) The command to switch USB from device mode back to host mode is as follows:

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

If you do not have the adb tool in your Windows system, you can use the adb program in the RKDevTool software.

An example using adb in Windows is shown below:



3.10. How to upload files to the Linux system of the development board

3.10.1. How to upload files from Ubuntu PC to the Linux system of the development board

3.10.1.1. How to upload files using the scp command

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

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

b. **orangepi：** The user name of the development board's Linux system can also be replaced with other names, such as root

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

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

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

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

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

3) There are more uses for scp. Please use the following command to view the man manual.

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

3. 10. 1. 2. How to upload files using FileZilla

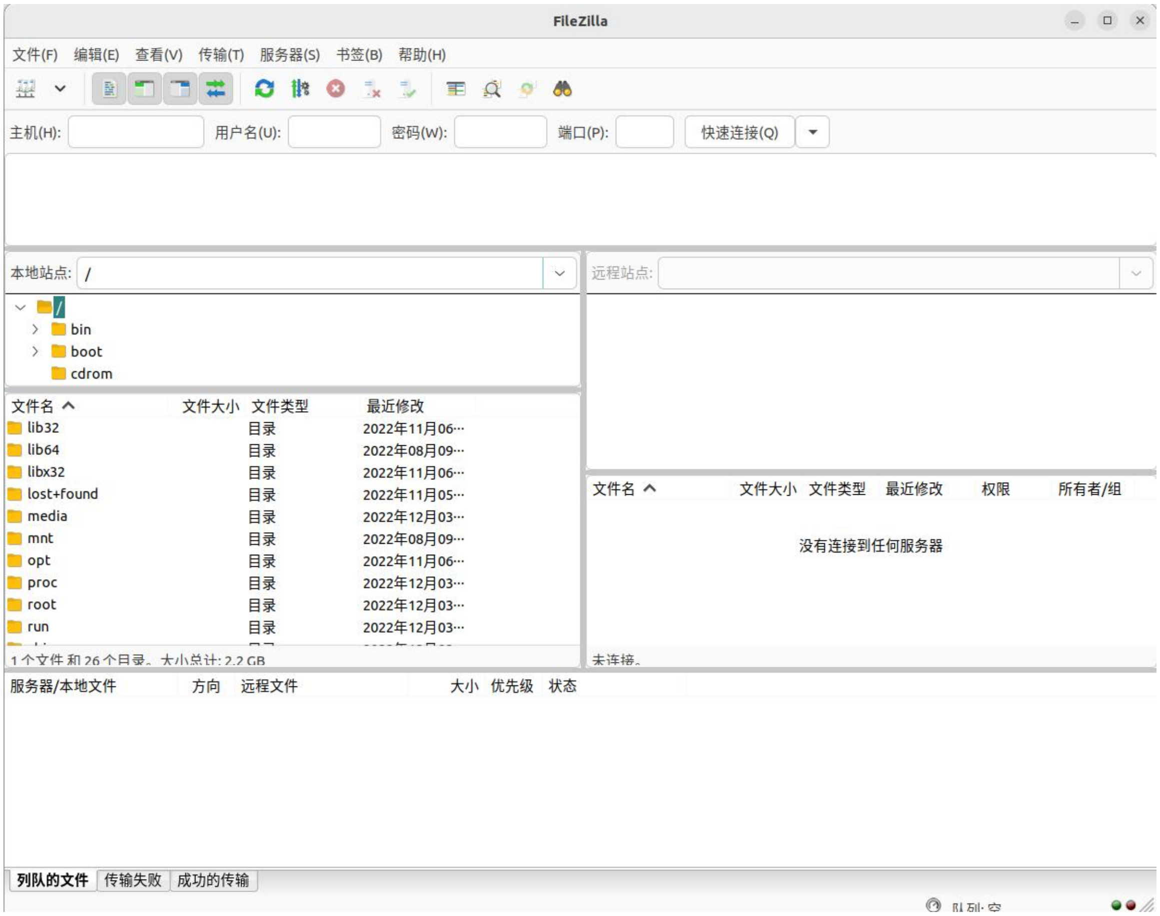
1) First install filezilla in your Ubuntu PC.

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

2) Then open filezilla using the command below.

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

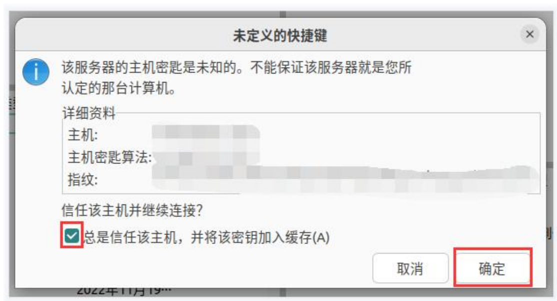
3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.



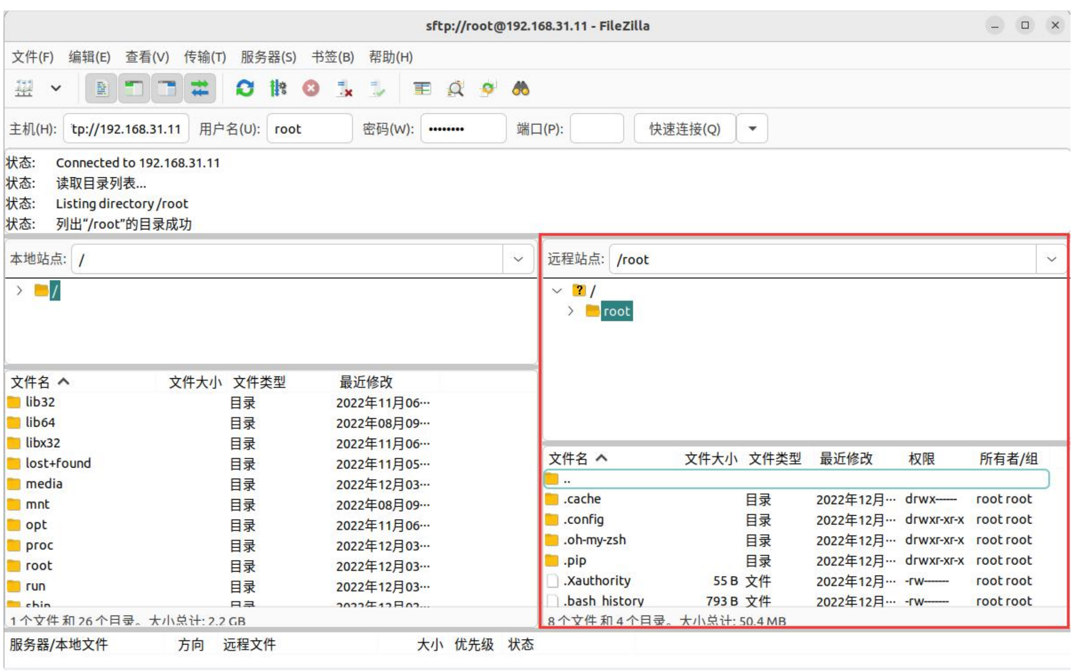
4) The method of connecting the development board is shown in the figure below:

5) Then select **Save Password** and click **OK**.

6) Then select **Always trust this host** and click **OK**.

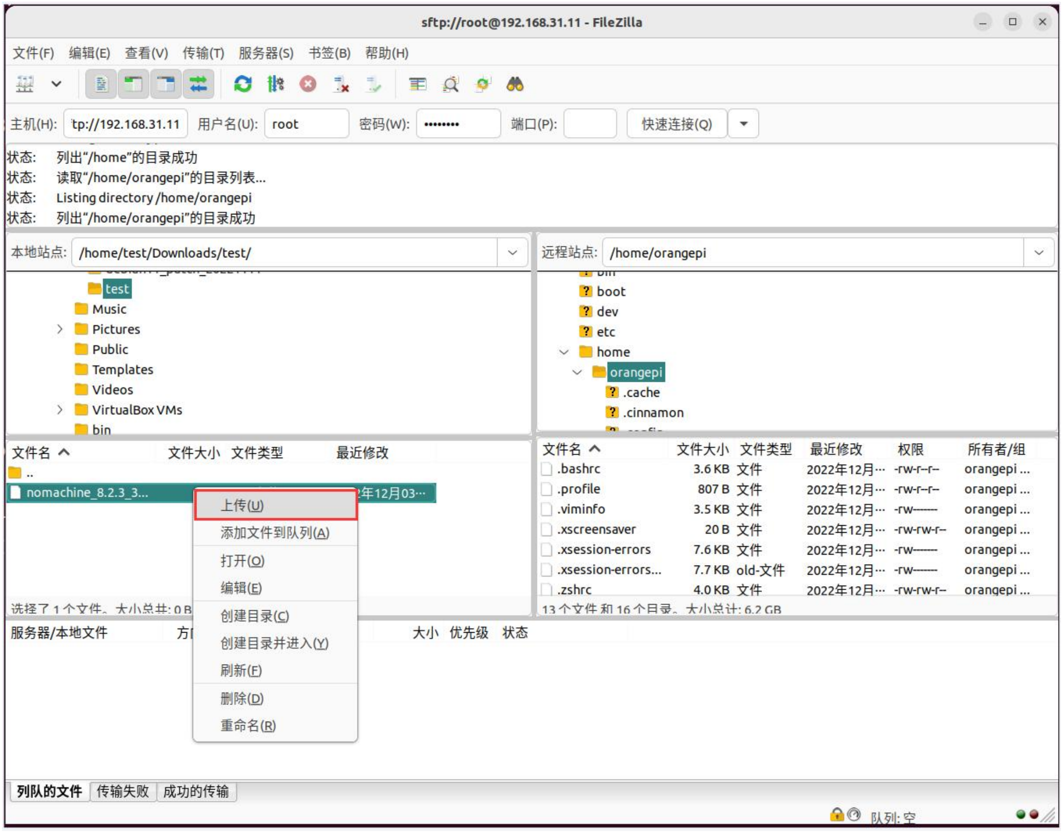


7) After the connection is successful, you can see the directory structure of the development board's Linux file system on the right side of the filezilla software.



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

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



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

10) The method for uploading a folder is the same as that for uploading a file, so I will not go into details here.

3. 10. 2. How to upload files from Windows PC to the Linux system of the development board

3. 10. 2. 1. How to upload files using FileZilla

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

https://filezilla-project.org/download.php?type=client

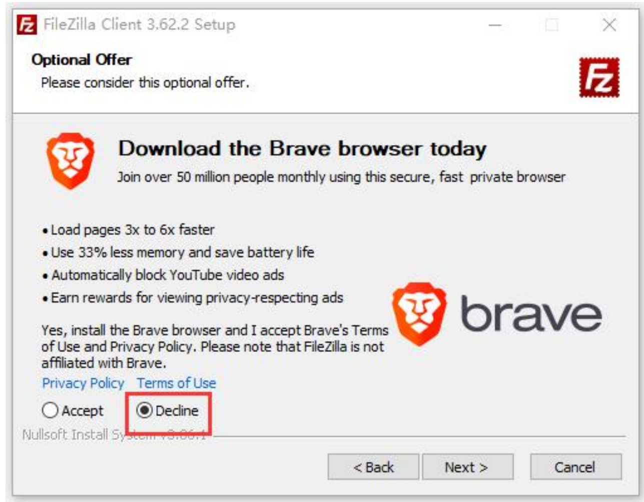
Please select your edition of FileZilla Client

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

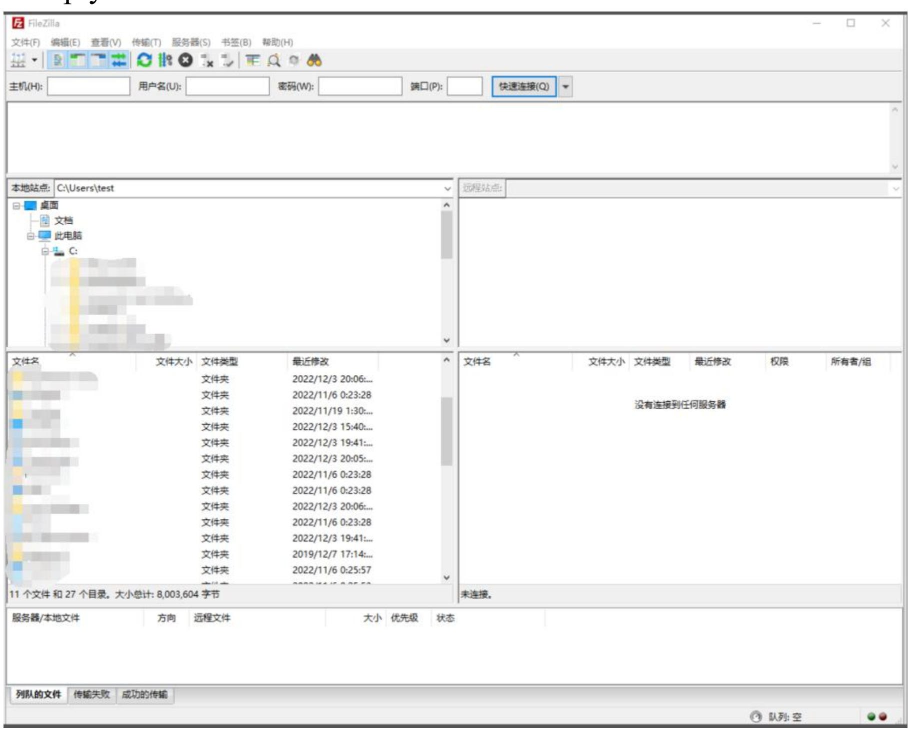
2) The downloaded installation package is as shown below, and you can double-click it to install it directly.

FileZilla_Server_1.5.1_win64-setup.exe

During the installation process, select `Decline` on the following installation interface, and then select `Next>`.

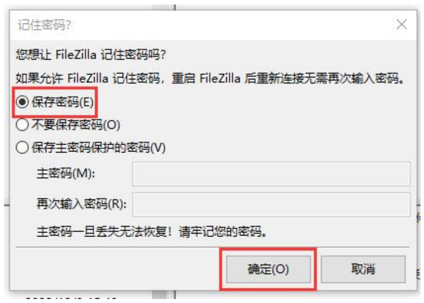


3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.

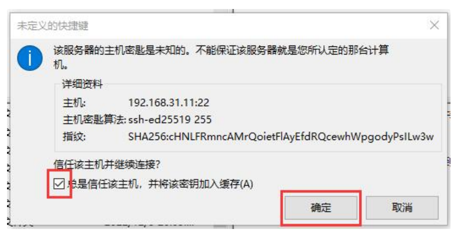


4) The method of connecting the development board is shown in the figure below:

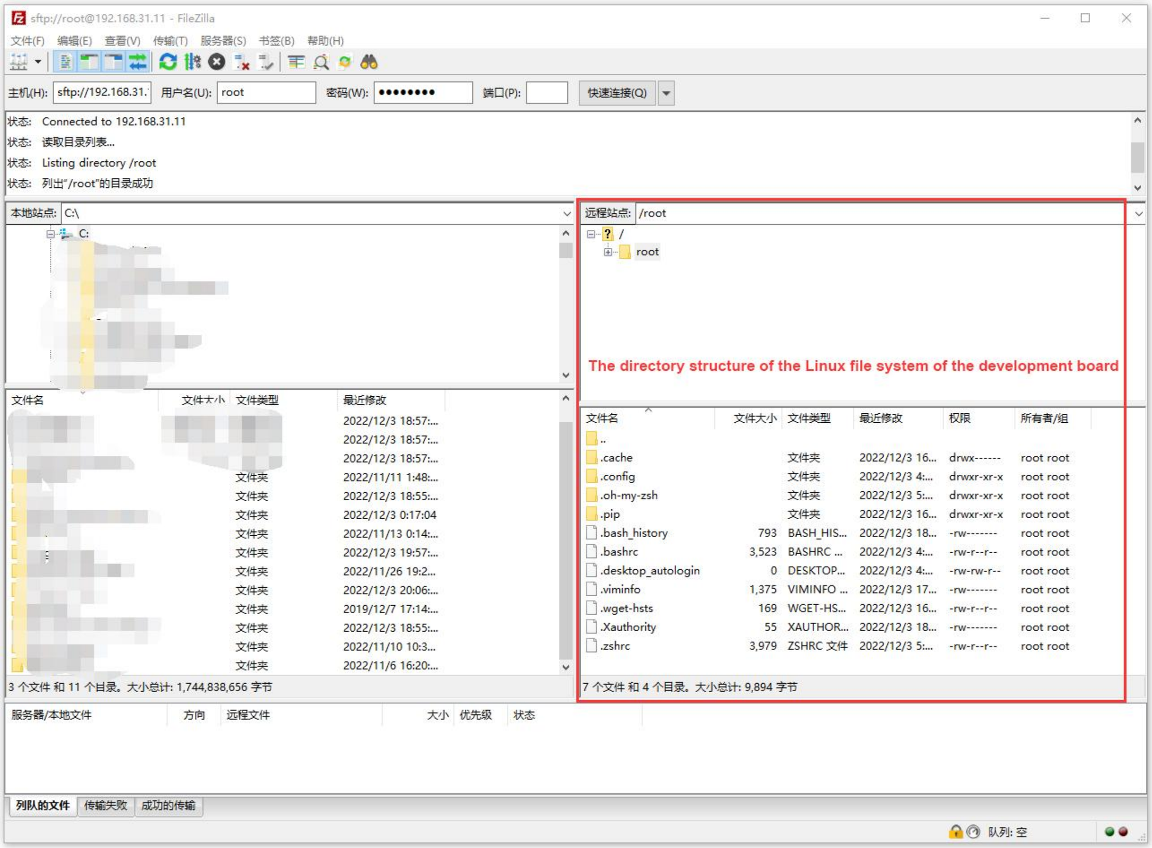
5) Then select **Save Password** and click **OK**.



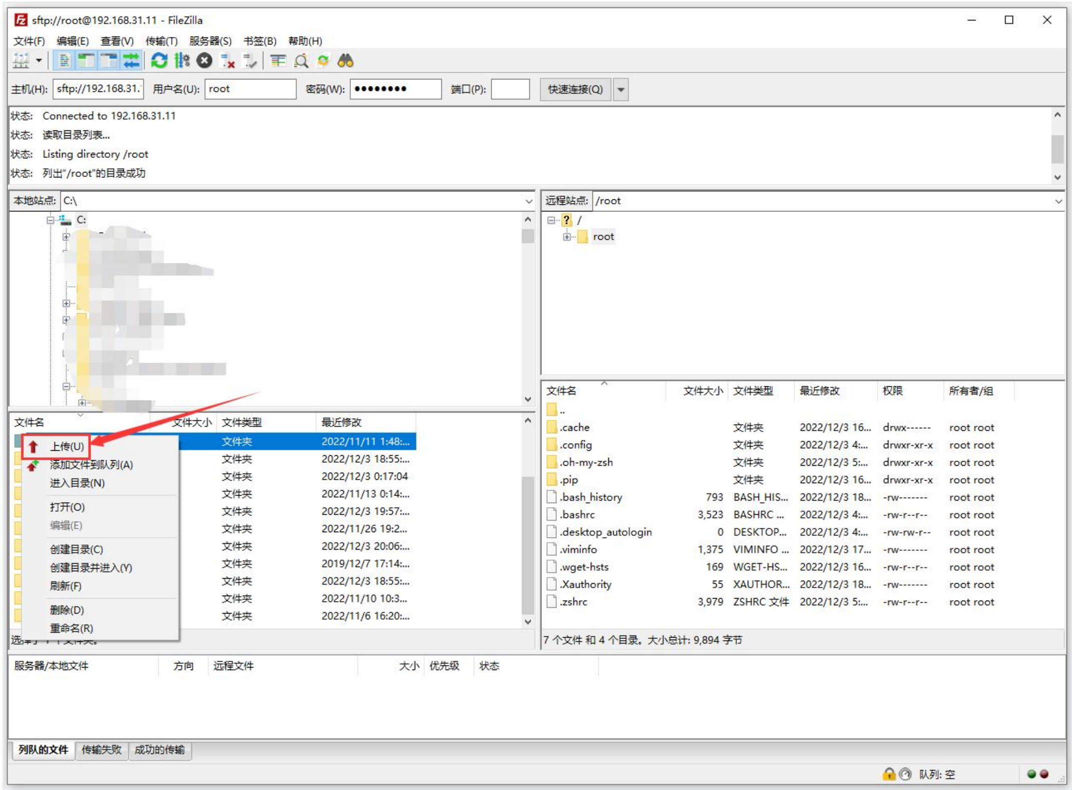
6) Then select **Always trust this host** and click **OK**.



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



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



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

10) The method for uploading a folder is the same as that for uploading a file, so I will not go into details here.

3.11. HDMI test

3.11.1. HDMI display test

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

2) After starting the Linux system, if the HDMI monitor has image output, it means that the HDMI interface is working properly.

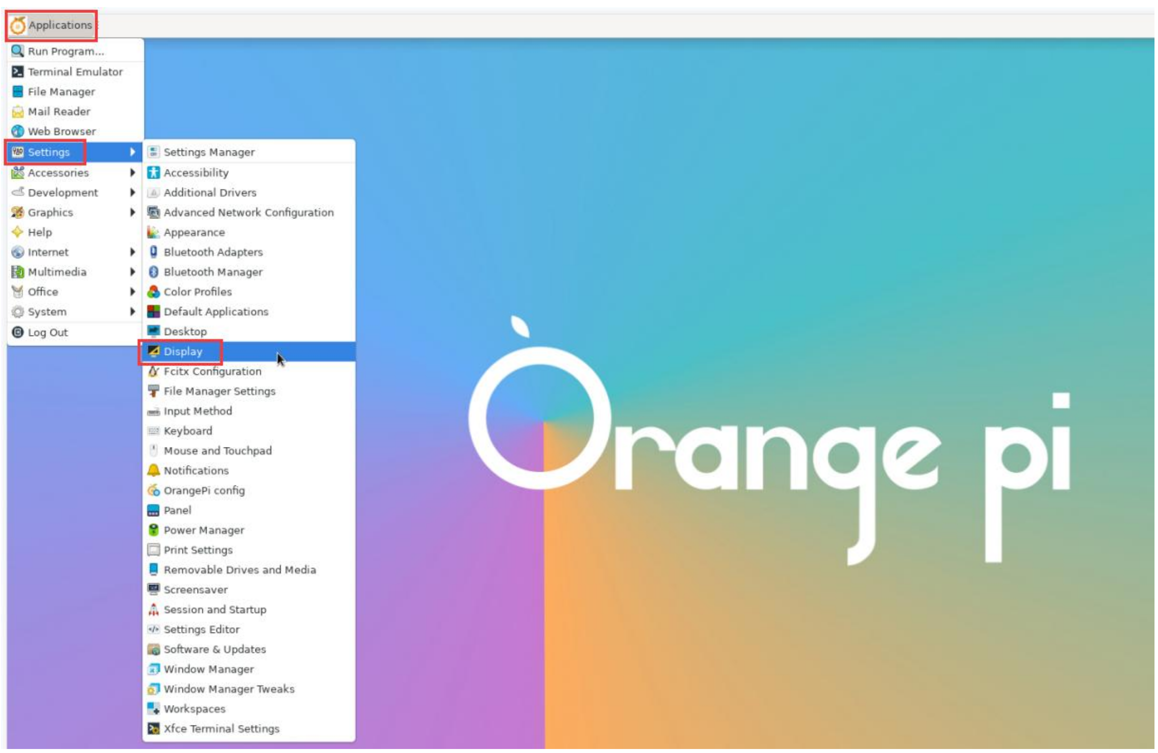
Please note that although many laptops are equipped with HDMI interfaces, the HDMI interfaces of laptops generally only have output functions and do not have HDMI in functions, which means that the HDMI output of other devices cannot be displayed on the laptop screen.

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

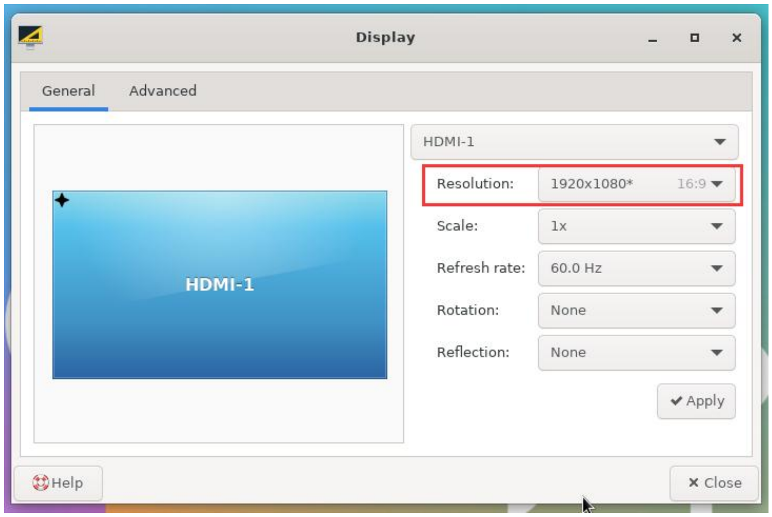
When there is no display on HDMI, please first check whether the HDMI cable is plugged in tightly. After confirming that the connection is OK, you can try a different screen to see if there is any display.

3. 11. 2. HDMI resolution setting method

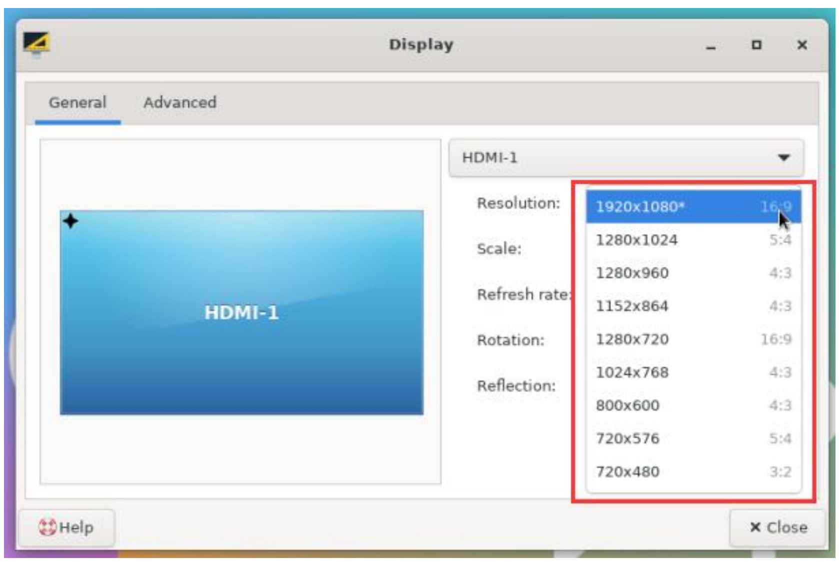
1) First, open **Display** in **Settings**.



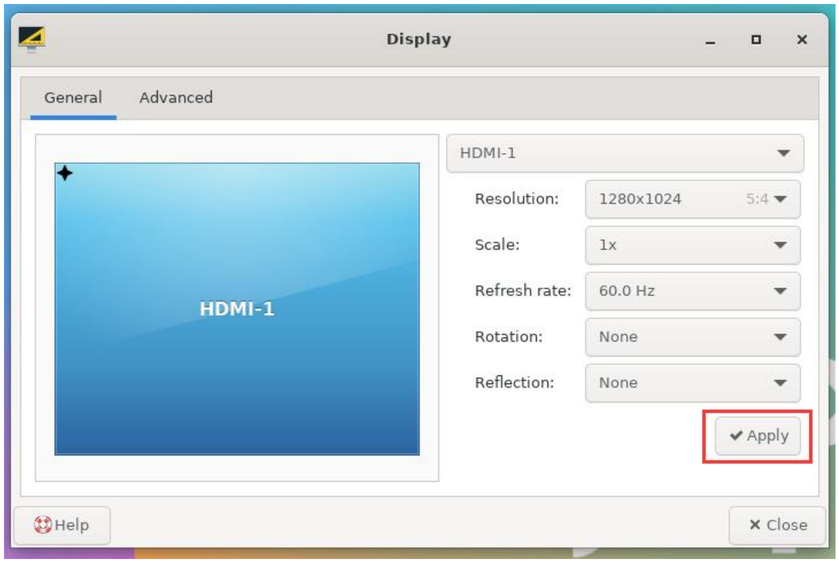
2) Then you can see the current resolution of the system.



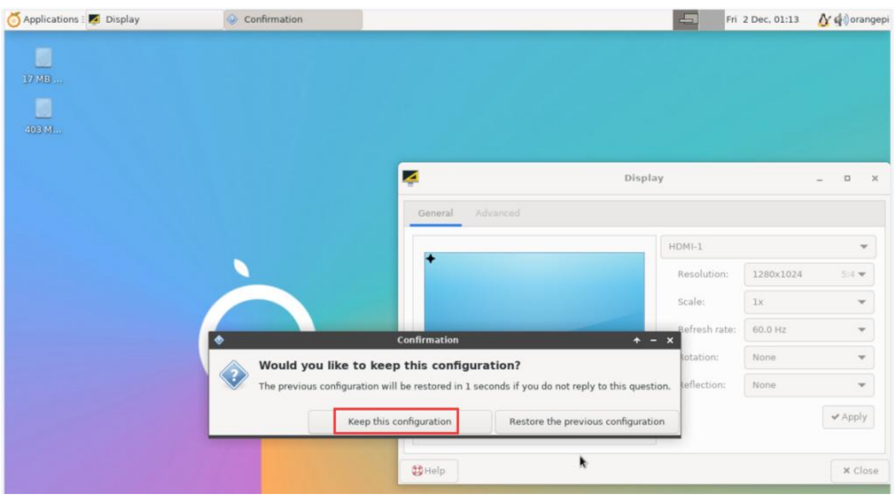
3) Click the drop-down box of Resolution to see all the resolutions currently supported by the monitor.



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



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



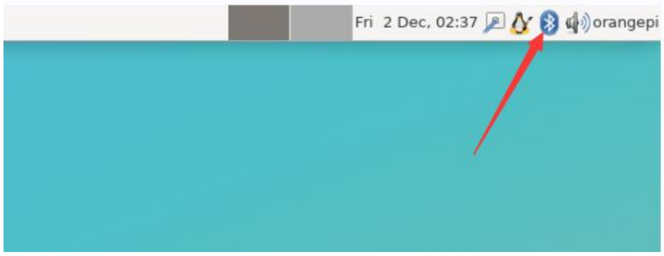
3. 12. How to use Bluetooth

Please note that there is no Bluetooth module on the Orange Pi CM5 Base development board. An external USB network card with Bluetooth is required to use the Bluetooth function.

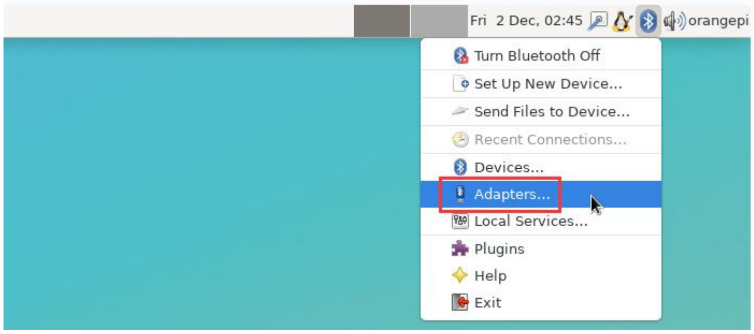
For instructions on using the external USB network card, please refer to the USB wireless network card testing section.

3. 12. 1. Testing methods for desktop images

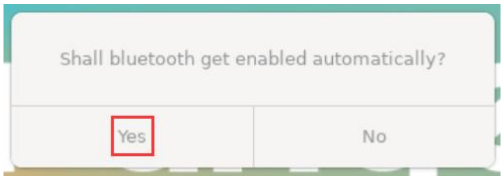
1) Click the Bluetooth icon in the upper right corner of the desktop.



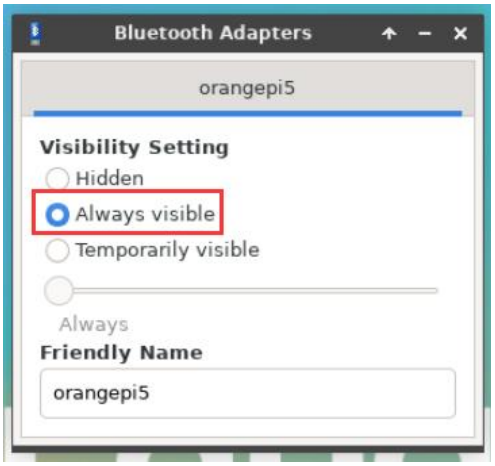
2) Then select the adapter.



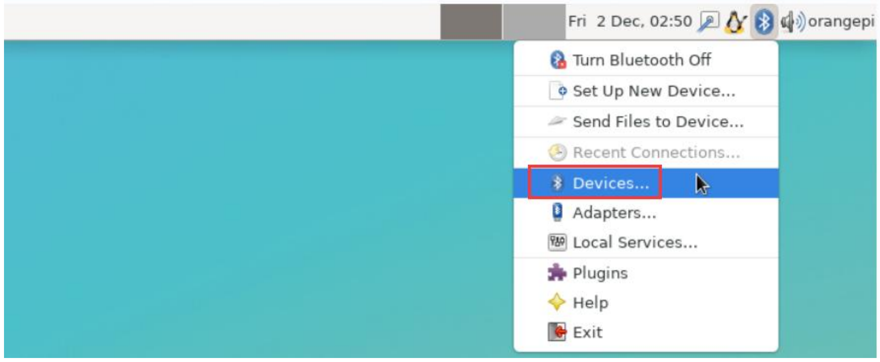
3) If the following interface is displayed, select **Yes.**



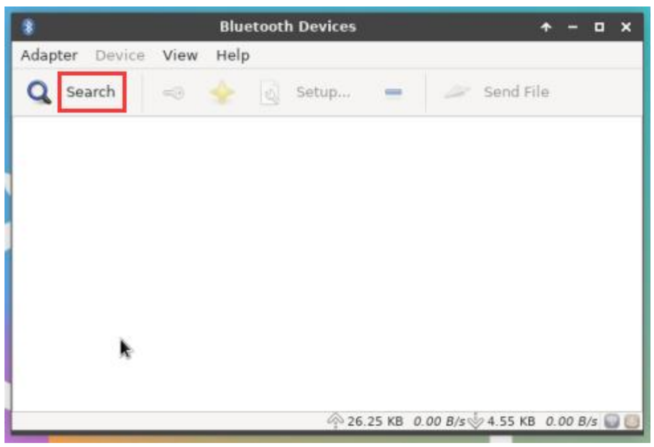
4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter settings interface, and then turn it off.



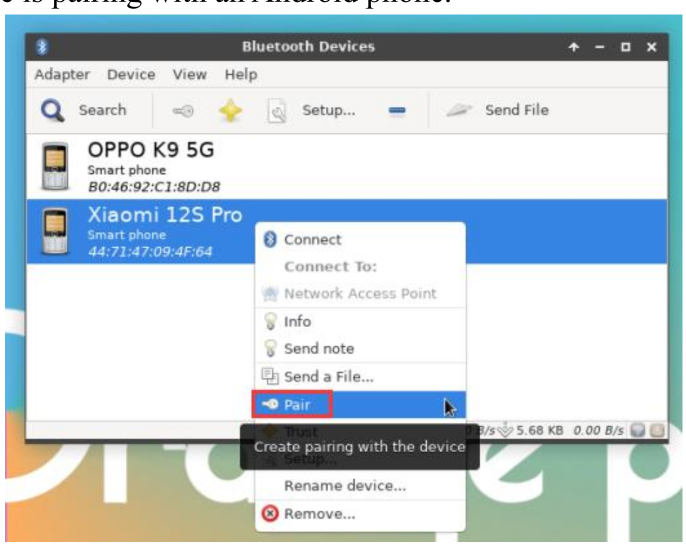
5) Then open the configuration interface of the Bluetooth device.



6) Click **Search** to start scanning for nearby Bluetooth devices.

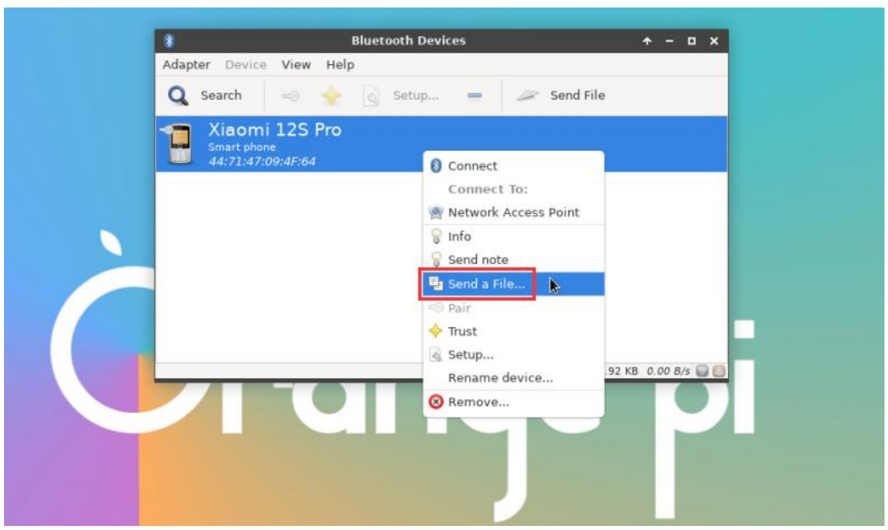


6) Then select the Bluetooth device you want to connect to, and right-click the mouse to pop up the operation interface for this Bluetooth device. Select **Pair** to start pairing. The demonstration here is pairing with an Android phone.

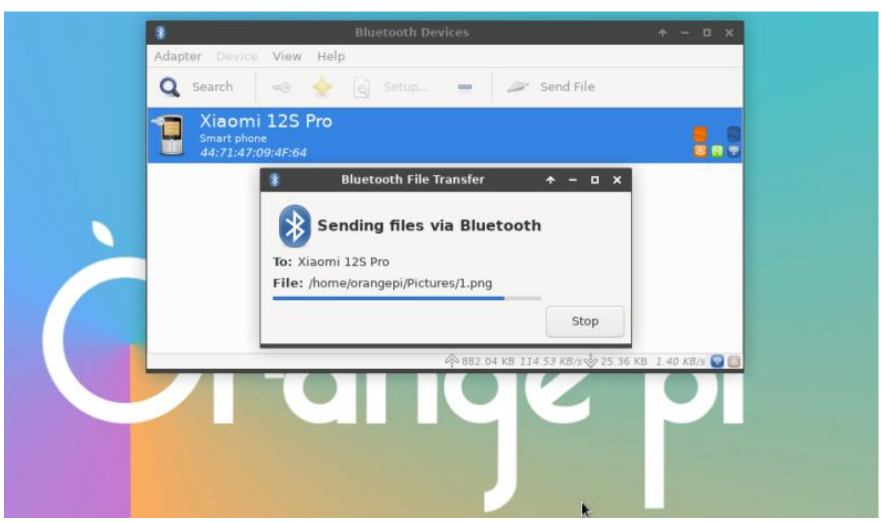


7) When pairing, a pairing confirmation box will pop up in the upper right corner of the desktop. Just select **Confirm**. Confirmation is also required on the mobile phone.

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



9) The interface for sending pictures is as follows:



3. 13. USB interface test

The USB port can be connected to a USB hub to expand the number of USB ports.

3. 13. 1. Test by connecting USB mouse or keyboard

1) Plug the USB keyboard into the USB port of the Orange Pi development board.

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

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

3. 13. 2. Connect USB storage device to test

1) First, insert the USB flash drive or USB mobile hard drive into the USB port of the Orange Pi development board.

2) Execute the following command. If you can see the output of sdX, it means the USB disk is 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 USB drive to **/mnt**, and then you can view the files in the USB drive.

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

4) After mounting, you can use the **df -h** command to view the capacity usage and mount point of the USB drive.

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

3. 13. 3. USB wireless network card test

The currently **tested** USB wireless network cards are shown below. Please test other types of USB wireless network cards by yourself. If they cannot be used, you need to transplant the corresponding USB wireless network card driver.

Serial number	model	
1	RTL8723BU Support 2.4G WIFI+BT4.0	
2	RTL8811 Support 2.4G +5G WIFI	
3	RTL8821CU Support 2.4G +5G WIFI Support BT 4.2	

3. 13. 3. 1. RTL8723BU test

1) First, insert the RTL8723BU wireless network card module into the USB port of the development board.

2) Then the Linux system will automatically load the RTL8723BU Bluetooth and WIFI related kernel modules. Through the lsmod command, you can see that the following kernel modules have been automatically loaded.

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

3) The loading information of the RTL8723BU module can be seen through the dmesg command.

```
orangepi@orangepi:~$ dmesg
```

```
......
[   83.438901] usb 2-1: new high-speed USB device number 2 using ehci-platform
[   83.588375] usb 2-1: New USB device found, idVendor=0bda, idProduct=b720,
bcdDevice= 2.00
[   83.588403] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   83.588422] usb 2-1: Product: 802.11n WLAN Adapter
[   83.588443] usb 2-1: Manufacturer: Realtek
[   83.588460] usb 2-1: SerialNumber: 00e04c000001
[   83.601974] Bluetooth: hci0: RTL: examining hci_ver=06 hci_rev=000b lmp_ver=06
lmp_subver=8723
[   83.603894] Bluetooth: hci0: RTL: rom_version status=0 version=1
[   83.603920] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_fw.bin
[   83.610108] Bluetooth: hci0: RTL: loading rtl_bt/rtl8723b_config.bin
[   83.611274] Bluetooth: hci0: RTL: cfg_sz 68, total sz 22564
[   83.658494] rtk_btusb: Realtek Bluetooth USB driver ver
3.1.6d45ddf.20220519-142432
[   83.658651] usbcore: registered new interface driver rtk_btusb
[   83.667124] usb 2-1: This Realtek USB WiFi dongle (0x0bda:0xb720) is untested!
[   83.667137] usb 2-1: Please report results to Jes.Sorensen@gmail.com
[   83.890140] usb 2-1: Vendor: Realtek
[   83.890153] usb 2-1: Product: 802.11n WLAN Adapter
[   83.890159] usb 2-1: rtl8723bu_parse_efuse: dumping efuse (0x200 bytes):
......
[   83.890412] usb 2-1: RTL8723BU rev E (SMIC) 1T1R, TX queues 3, WiFi=1, BT=1,
GPS=0, HI PA=0
[   83.890417] usb 2-1: RTL8723BU MAC: 00:13:ef:f4:58:ae
[   83.890421] usb 2-1: rtl8xxxu: Loading firmware rtlwifi/rtl8723bu_nic.bin
[   83.895289] usb 2-1: Firmware revision 35.0 (signature 0x5301)
[   84.050893] Bluetooth: hci0: RTL: fw version 0x0e2f9f73
[   84.266905] Bluetooth: RFCOMM TTY layer initialized
[   84.266949] Bluetooth: RFCOMM socket layer initialized
[   84.266999] Bluetooth: RFCOMM ver 1.11
[   84.884270] usbcore: registered new interface driver rtl8xxxu
[   84.912046] rtl8xxxu 2-1:1.2 wlx0013eff458ae: renamed from wlan0
```

4) Then use the **sudo ifconfig** command to see the device node of RTL8723BU WIFI.

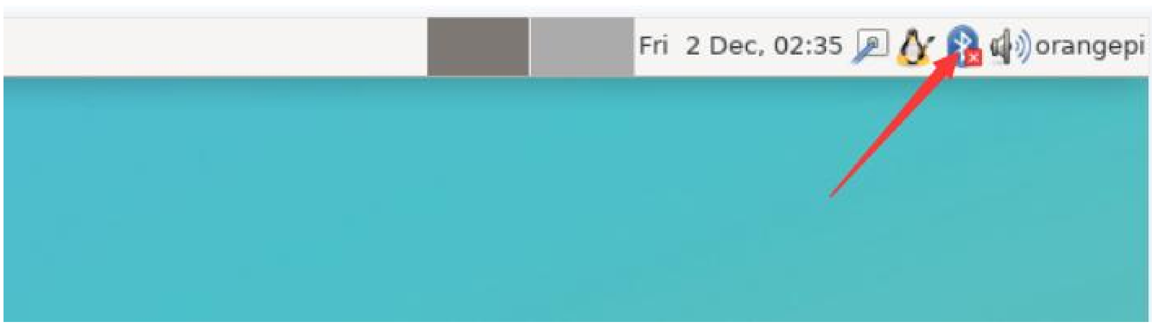
For the connection and test methods of WIFI, please refer to the **WIFI connection test** section, which will not be repeated here.

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

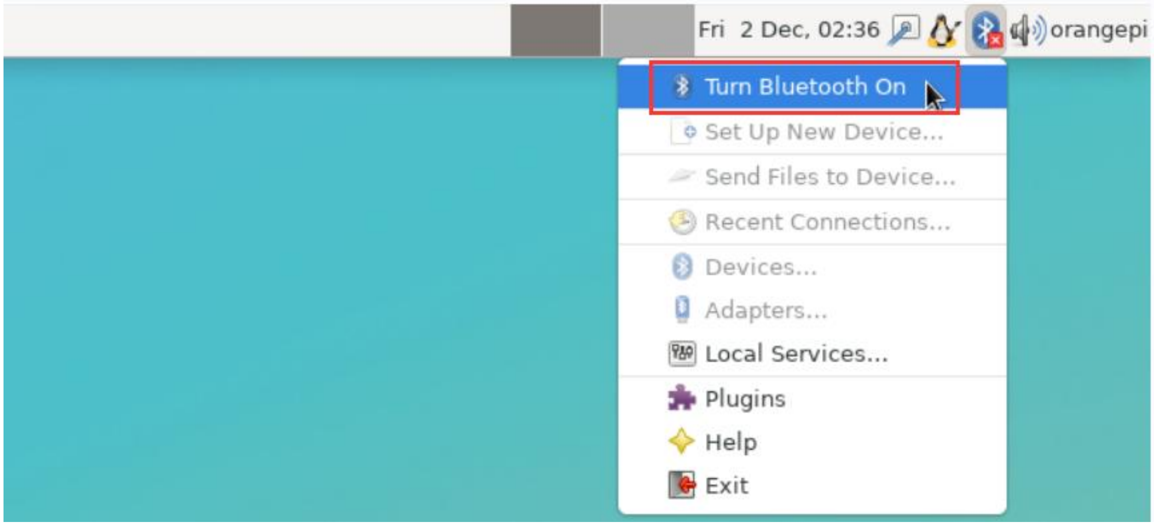
5) Then you can see the USB Bluetooth device through the **hciconfig** command.

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

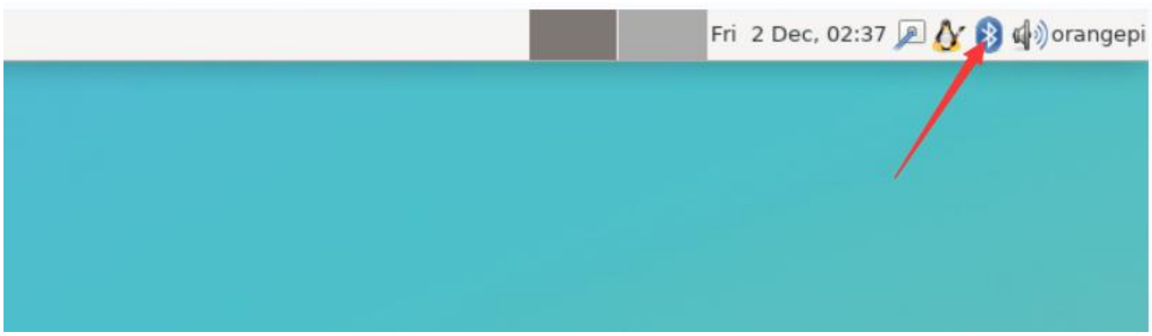
6) You can also see the Bluetooth icon on the desktop. At this time, Bluetooth is not turned on, so a red **x** will be displayed.



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



8) The display after turning on Bluetooth is as shown below.



9) For the Bluetooth testing method, please refer to the **Bluetooth Usage** section, which will not be repeated here.

3. 13. 3. 2. RTL8811 Test

1) First, insert the RTL8811 wireless network card module into the USB port of the development board.

2) Then the Linux system will automatically load the kernel modules related to RTL8811 WIFI. Through the lsmod command, you can see that the following kernel modules have been automatically loaded.

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

3) The loading information of the RTL8811 module can be seen through the dmesg command.

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

4) Then use the **sudo ifconfig** command to view the device node of WIFI. For the connection and test methods of WIFI, please refer to the **WIFI connection test** section, which will not be repeated here.

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

3.13.3.3. RTL8821CU test

1) First, insert the rtl8821cu wireless network card module into the USB interface of the development board.

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

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

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

If the USB WIFI module is in Driver CDROM Mode as seen by the lsusb command, please unplug and re-plug the USB WIFI module. If it still doesn't work, please manually execute the following command to switch the mode:

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

3) The Linux system will automatically load the rtl8821cu Bluetooth and WiFi related kernel modules. You can see through the lsmod command that the following kernel modules have been automatically loaded.

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

4) The loading information of the rtl8821cu module can be seen through the dmesg command.

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

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

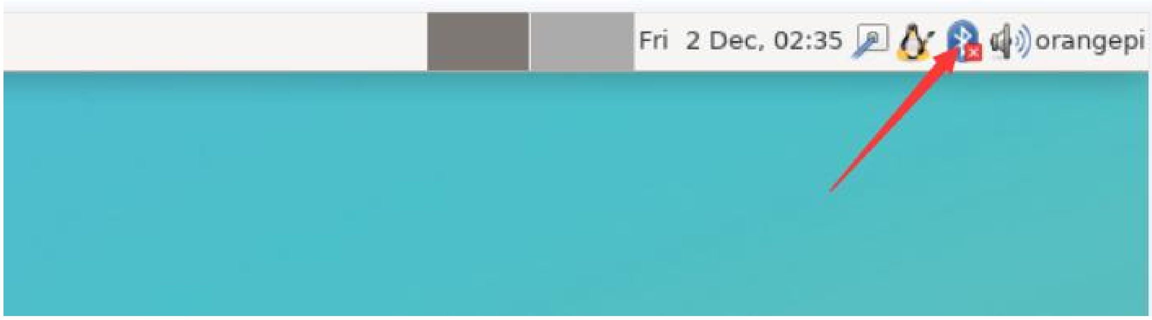
5) Then use the **sudo ifconfig** command to see the device node of rtl8821cu wifi. For the connection and test methods of wifi, please refer to the **WIFI connection test** section, which will not be repeated here.

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

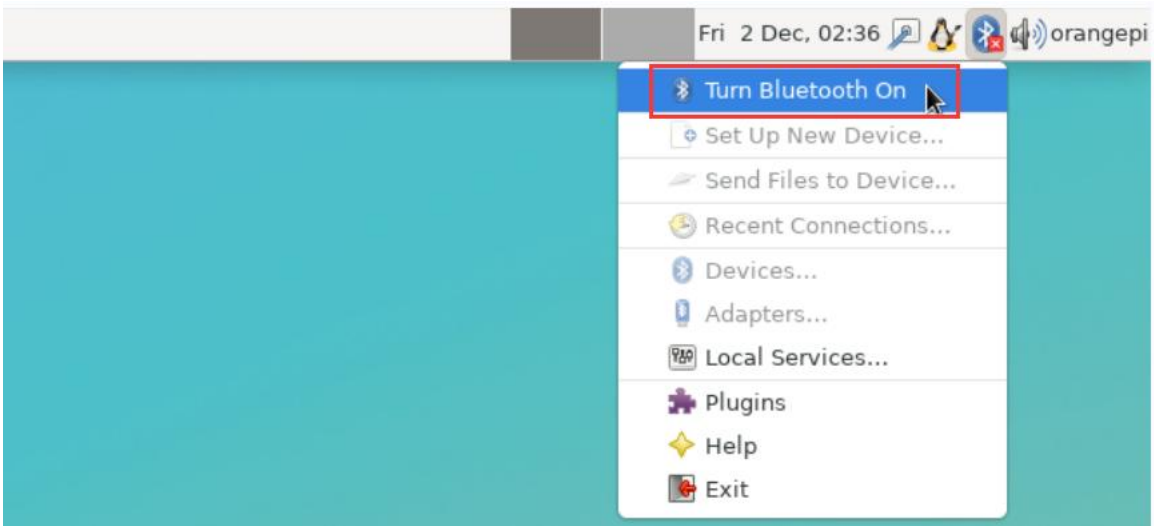
6) Then you can see the USB Bluetooth device through the **hciconfig** command.

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

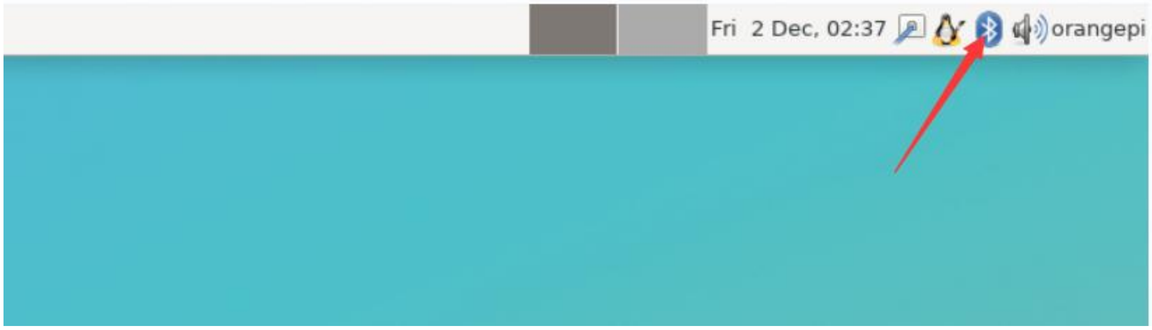
7) You can also see the Bluetooth icon on the desktop. At this time, Bluetooth is not turned on, so a red **x** will be displayed.



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



9) The display after turning on Bluetooth is as follows:



10) For the Bluetooth testing method, please refer to the **Bluetooth Usage** section, which will not be repeated here.

3. 13. 4. USB camera test

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

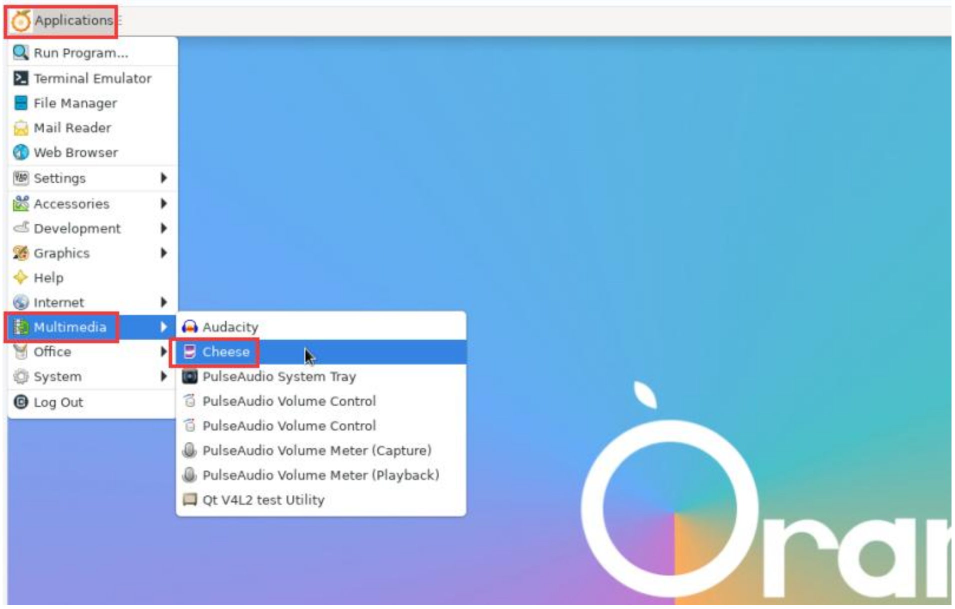
2) Through the v4l2-ctl command, you can see that the device node information of the USB camera is /dev/video0.

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

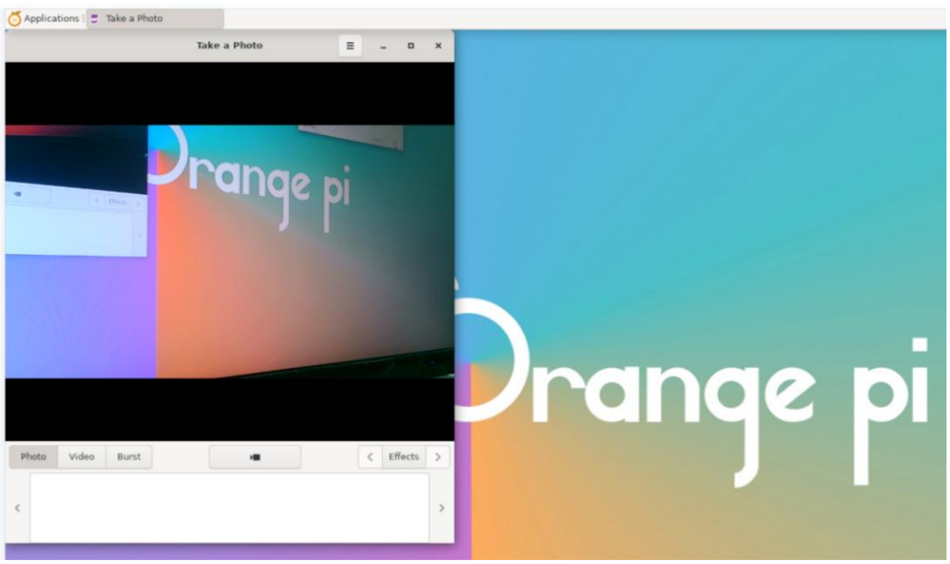
Note that the l in v4l2 is a lowercase letter l, not the number 1.

In addition, the video serial number is not always video0, please refer to the actual one you see.

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



The interface after Cheese opens the USB camera is as shown below:



4) How to test USB camera using fswebcam.

a. Install fswebcam.

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

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

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

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

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

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

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 Linux, after taking the photo, you can use the scp command to transfer the photo to the Ubuntu PC for mirror viewing.

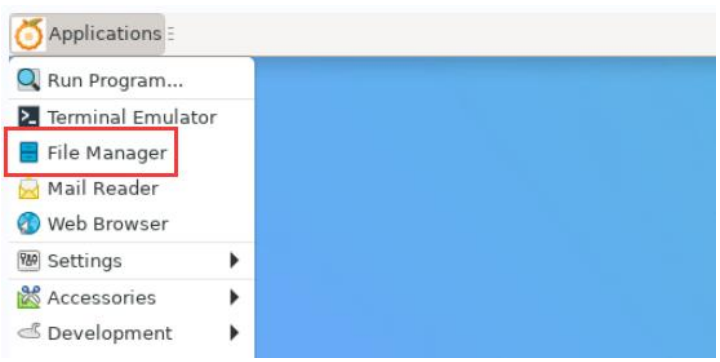
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test  （Modify the IP address and path according to the actual situation）
```

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

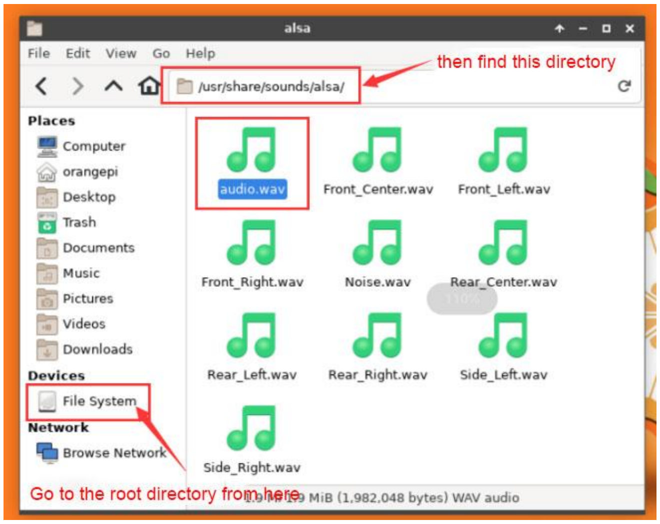
3. 14. Audio Test

3. 14. 1. Testing Audio Methods on Desktop Systems

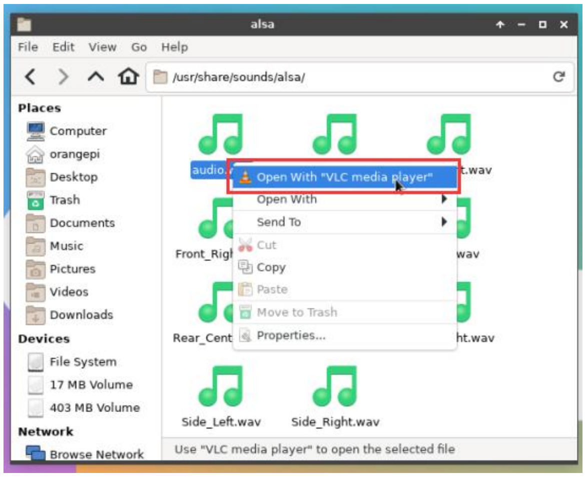
1) First open the file manager.



2) Then find the file below (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 choose to open it with vlc or mpv to start playing.



3. 14. 2. How to play audio using commands

3. 14. 2. 1. HDMI audio playback test

1) First, use an HDMI to HDMI cable to connect the Orange Pi development board to the TV (other HDMI displays need to be able to play audio).

2) Then check the HDMI sound card serial number. From the output below, you can know that the HDMI sound card is **card 0.**

```
orangepi@orangepi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
Authorization required, but no authorization protocol specified
```

```
xcb_connection_has_error() returned true
card 0: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

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

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

3. 15. Temperature sensor

The command to view the system temperature sensor is:

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

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

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

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

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

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

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

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

3. 16. Hardware watchdog test

The Linux system released by Orange Pi has the watchdog_test program pre-installed, which can be used for direct testing.

The method to run the watchdog_test program is as follows:

a. The second parameter 10 represents the watchdog counting time. If the watchdog is not fed within this time, the system will restart.
b. We can feed the dog by pressing any key on the keyboard (except ESC). After feeding the dog, the program will print a line of keep alive to indicate that the dog was fed successfully.

```
orangepi@orangepi:~$ sudo watchdog_test 10
open success
options is 33152,identity is sunxi-wdt
put_usr return,if 0,success:0
The old reset time is: 16
return ENOTTY,if -1,success:0
return ENOTTY,if -1,success:0
put_user return,if 0,success:0
put_usr return,if 0,success:0
keep alive
keep alive
keep alive
```

3. 17. Check the serial number of the RK3588S chip

The command to check the serial number of the RK3588S chip is as follows. The serial number of each chip is different, so the serial number can be used to distinguish multiple development boards.

```
orangepi@orangepi:~$ cat_serial.sh
Serial          : 1404a7682e86830c
```

3. 18. How to install Docker

1) The Linux image provided by Orange Pi has Docker pre-installed, but the Docker service is not enabled by default.

2) Use the **enable_docker.sh** script to enable the docker service. Then you can start using the docker command, and the docker service will be automatically started the next time you start the system.

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

3) Then you can use the following command to test docker. If you can run hello-world, it means that docker can be used normally.

```
orangepi@orangepi:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
256ab8fe8778: Pull complete
Digest:
sha256:7f0a9f93b4aa3022c3a4c147a449ef11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
.....
```

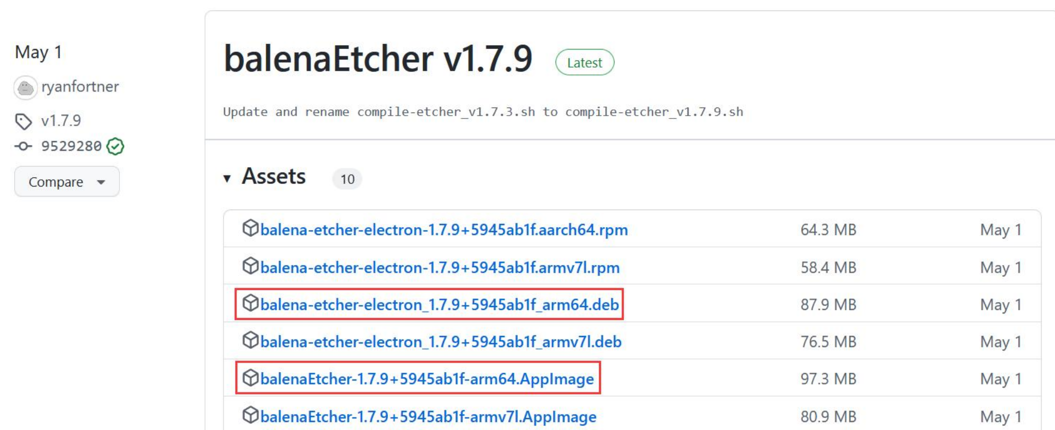
3.19. How to download and install the arm64 version of balenaEtcher

1) The download address of balenaEtcher arm64 version is:
   a. The download address of the deb installation package is as follows, which needs to be installed before use.

https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balena-etcher-electron_1.7.9+5945ab1f_arm64.deb

   b. The download address of the AppImage version that does not require installation is as follows:

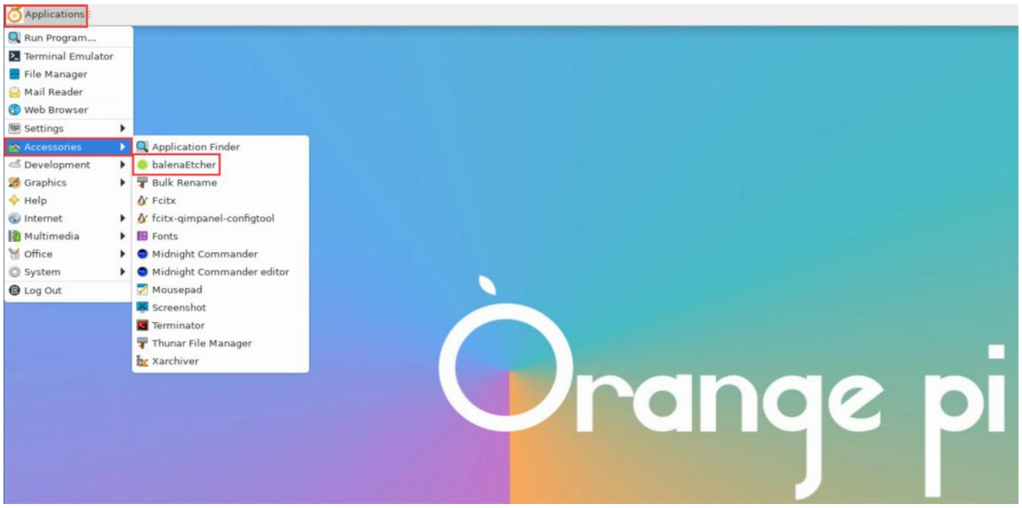
https://github.com/Itai-Nelken/BalenaEtcher-arm/releases/download/v1.7.9/balenaEtcher-1.7.9+5945ab1f-arm64.AppImage



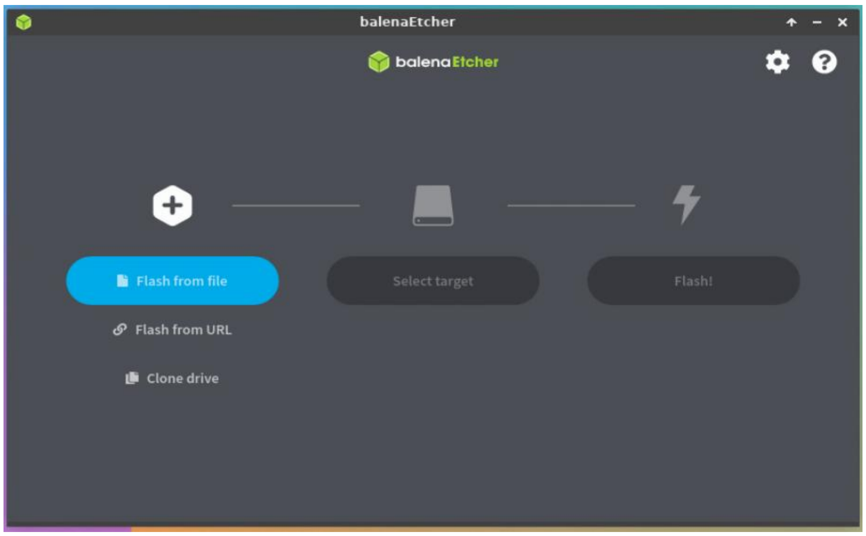
2) How to install and use the deb version of balenaEtcher:
   a. The installation command of the deb version of balenaEtcher is as follows:

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

   b. After the deb version of balenaEtcher is installed, you can open it in Application.



c. balenaEtcher opens with the following interface:

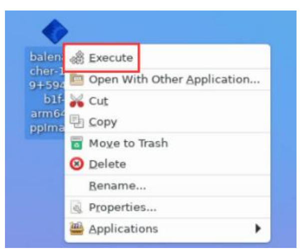


3) How to use the AppImage version of balenaEtcher:

a. First, add permissions to balenaEtcher.

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

b. Then select the AppImage version of balenaEtcher, right-click your mouse, and click Execute to open balenaEtcher.



3. 20. How to install Baota Linux Panel

Baota Linux Panel is a server management software that improves operation and maintenance efficiency. It supports more than 100 server management functions such as one-click LAMP/LNMP/cluster/monitoring/website/FTP/database/JAVA (excerpted from Baota official website)

1) The recommended order of Baota Linux system compatibility is:

Debian11 > Ubuntu 22.04 > Debian12

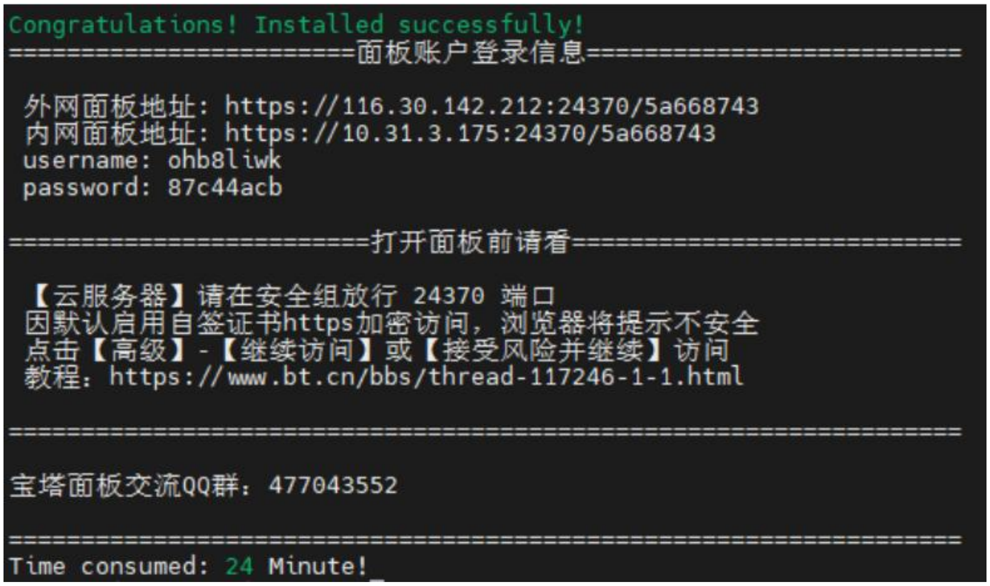
2) Then enter the following command in the Linux system to start the installation of the pagoda.

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

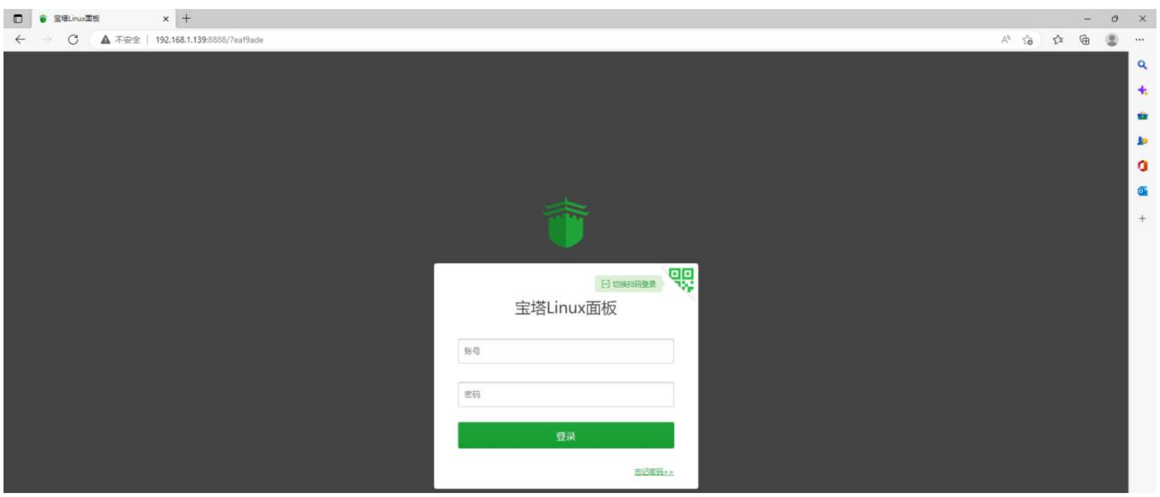
3) Then the Baota installation program will prompt whether to install **Bt-Panel** to the **/www** folder, just enter **y**.

```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------
| The WebPanel URL will be http://SERVER_IP:8888 when installed.
+----------------------------------------------------------------------

Do you want to install Bt-Panel to the /www directory now?(y/n): y
```

4) Then all you have to do is wait patiently. When you see the following print information output by the terminal, it means that the pagoda has been installed. The entire installation process takes about 12 minutes, which may vary depending on the network speed.

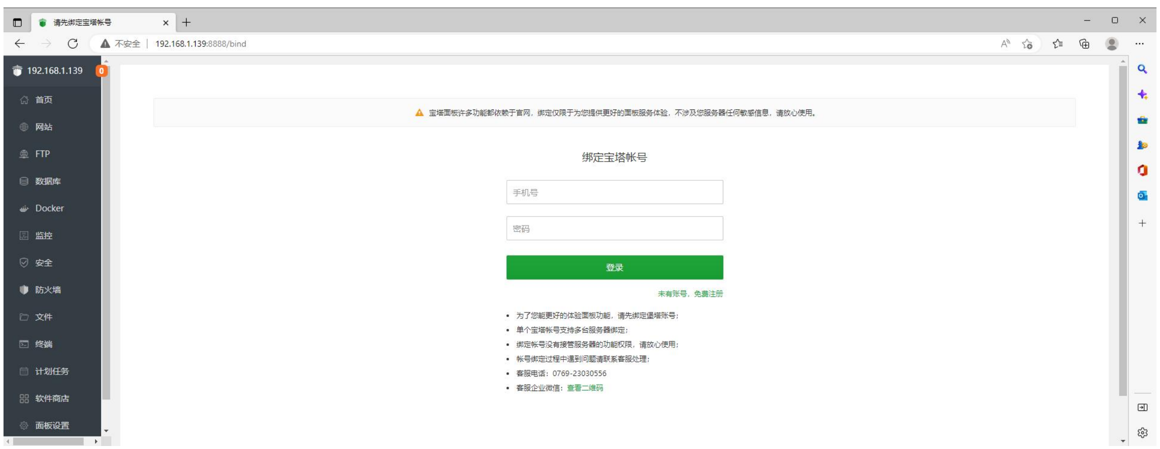


5) At this time, enter the **panel address** shown above in the browser to open the login interface of the Baota Linux panel, and then enter the **username** and **password** shown in the above picture in the corresponding positions to log in to Baota.

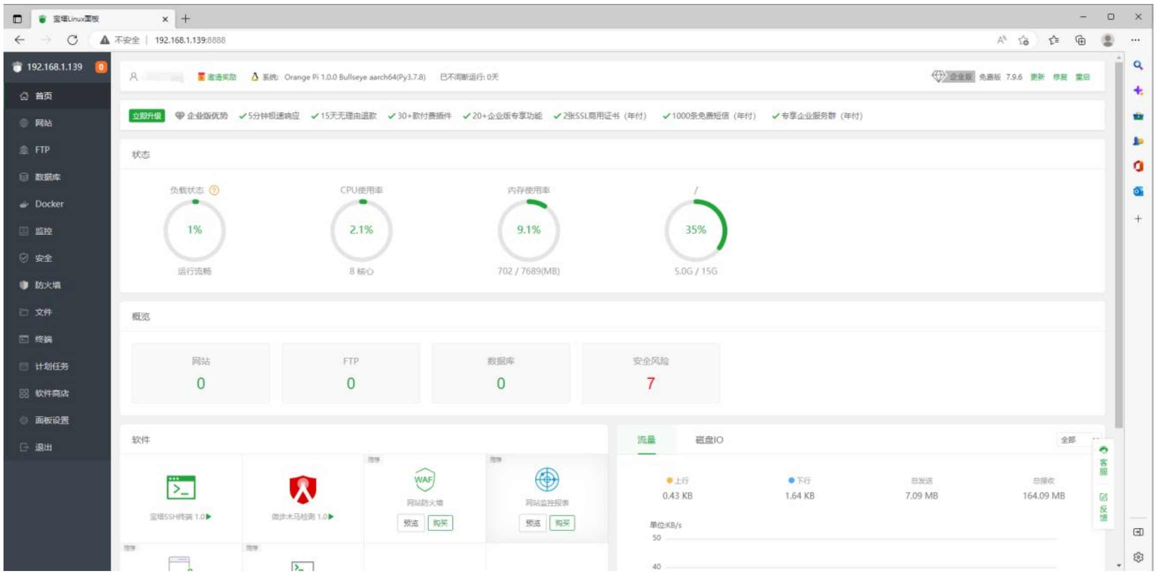


6) After successfully logging into the pagoda, the welcome interface below will pop up. First, please read the user instructions in the middle and drag them to the bottom. Then you can select "I have agreed and read the User Agreement", and then click "Enter Panel" to enter the pagoda.

7) After entering the bt.com, you will be prompted to bind your account on the bt.com official website. If you do not have an account, you can go to the bt.com official website (**https://www.bt.cn**) to register one.

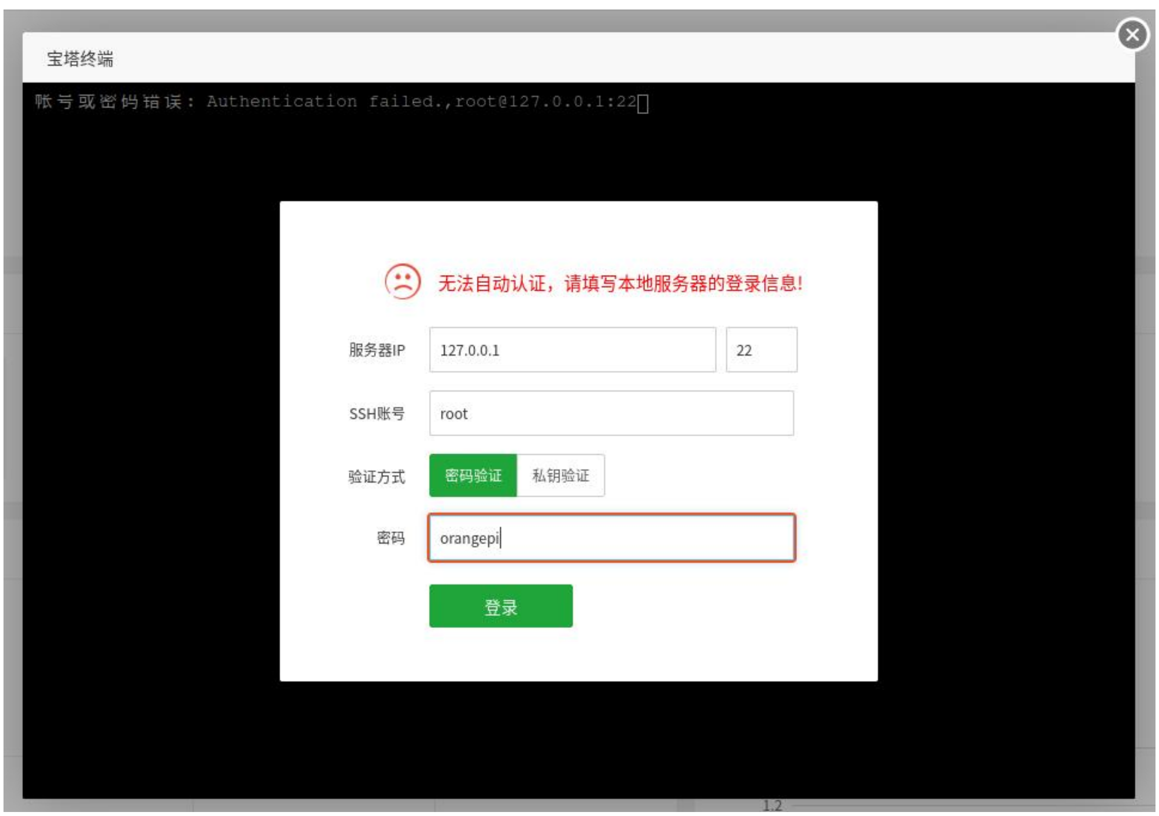


8) The final displayed interface is shown in the figure below, where you can intuitively see some status information of the development board Linux system, such as load status, CPU usage, memory usage, and storage space usage.

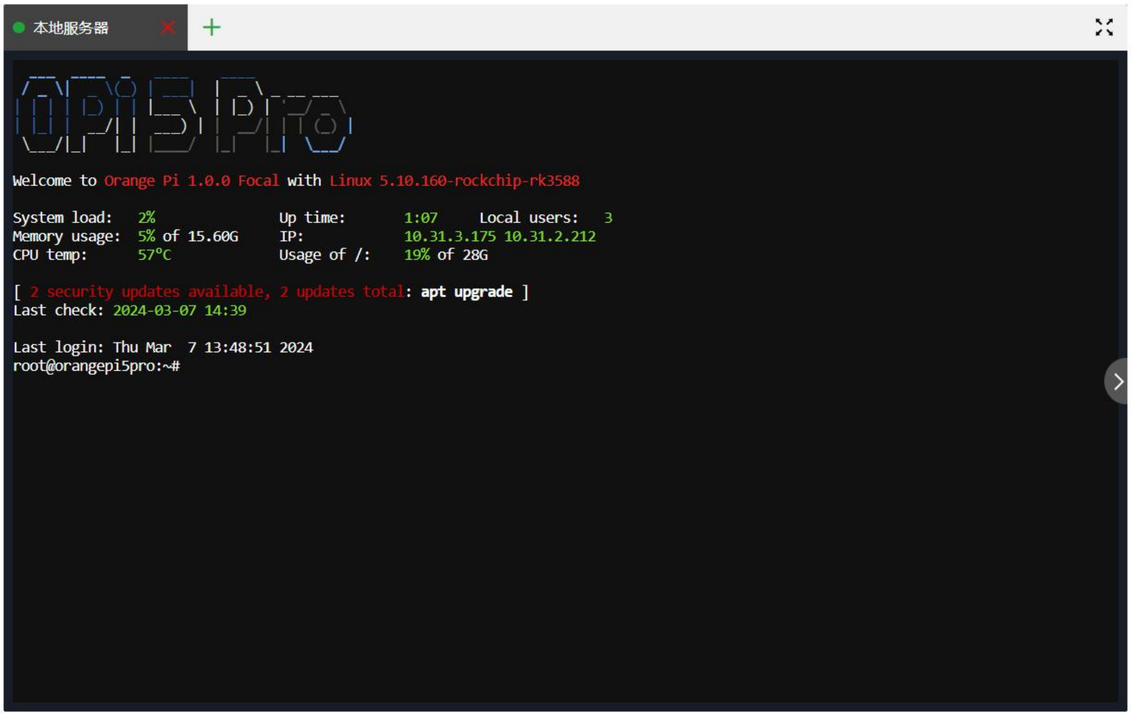


9) Test the SSH terminal login of Baota.

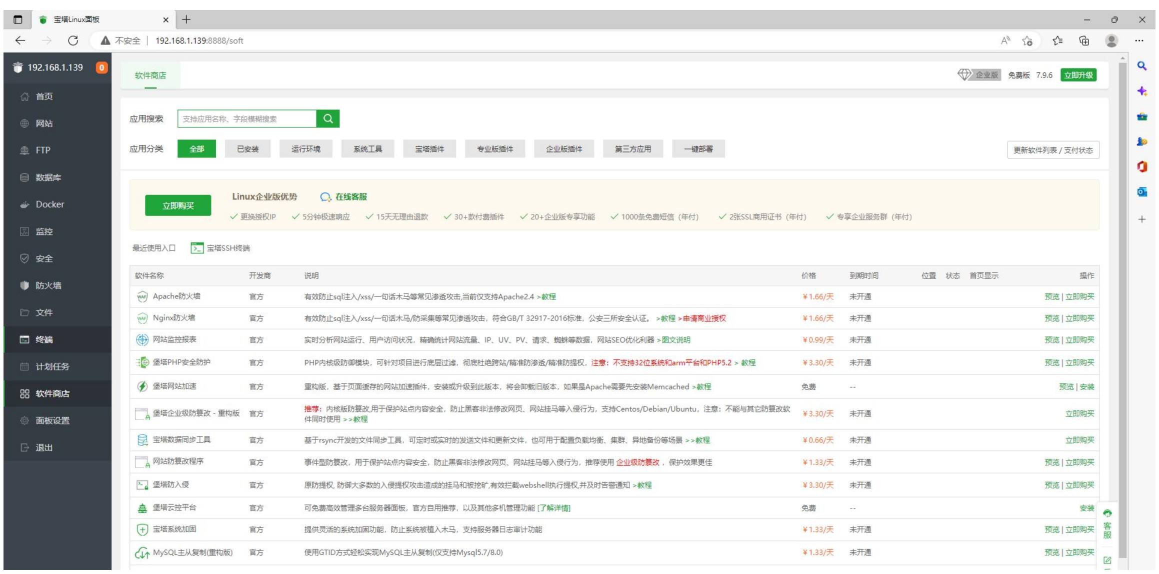
a. After opening the SSH terminal of Baota, you will be prompted to enter the password of the development board system. At this time, enter **orangepi** in the password box (the default password, if you have changed it, please fill in the modified password).



b. The display after successful login is as shown below:



10) In Baota's software store, you can install software such as Apache, MySQL, and PHP, and you can also deploy various applications with one click. Please explore this part of the function by yourself, and I will not demonstrate it here one by one.



11) Baota command line tool test.

```
orangepi@orangepi5pro:~$ sudo bt
==================================宝塔面板命令行==================================
(1) 重启面板服务                  (8) 改面板端口                                 |
(2) 停止面板服务                  (9) 清除面板缓存                               |
(3) 启动面板服务                  (10) 清除登录限制                              |
(4) 重载面板服务                  (11) 设置是否开启IP + User-Agent验证           |
(5) 修改面板密码                  (12) 取消域名绑定限制                          |
(6) 修改面板用户名                (13) 取消IP访问限制                            |
(7) 强制修改MySQL密码             (14) 查看面板默认信息                          |
(22) 显示面板错误日志             (15) 清理系统垃圾                              |
(23) 关闭BasicAuth认证            (16) 修复面板(检查错误并更新面板文件到最新版)  |
(24) 关闭动态口令认证             (17) 设置日志切割是否压缩                      |
(25) 设置是否保存文件历史副本     (18) 设置是否自动备份面板                      |
(26) 关闭面板ssl                  (19) 关闭面板登录地区限制                      |
(28) 修改面板安全入口             (29) 取消访问设备验证                          |
(0) 取消                                                                         |
==================================================================================
请输入命令编号: 14
===============================================
正在执行(14)...
===============================================
==================================================================
BT-Panel default info!
==================================================================
外网面板地址: https://116.30.142.212:24370/5a668743
内网面板地址: https://10.31.3.175:24370/5a668743
username: ohb8liwk
password: ********
Warning:
If you cannot access the panel,
release the following port (8888|888|80|443|20|21) in the security group
注意: 初始密码仅在首次登录面板前能正确获取, 其它时间请通过 bt 5 命令修改密码
==================================================================
orangepi@orangepi5pro:~$
```

12) For more functions of the pagoda, please refer to the following information to explore it yourself.

User manual：**http://docs.bt.cn**
Forum Address：**https://www.bt.cn/bbs**
GitHub Link：**https://github.com/aaPanel/BaoTa**

3. 21. Set up Chinese environment and install Chinese input method

Note: Before installing the Chinese input method, please make sure that the Linux system used by the development board is the desktop version.

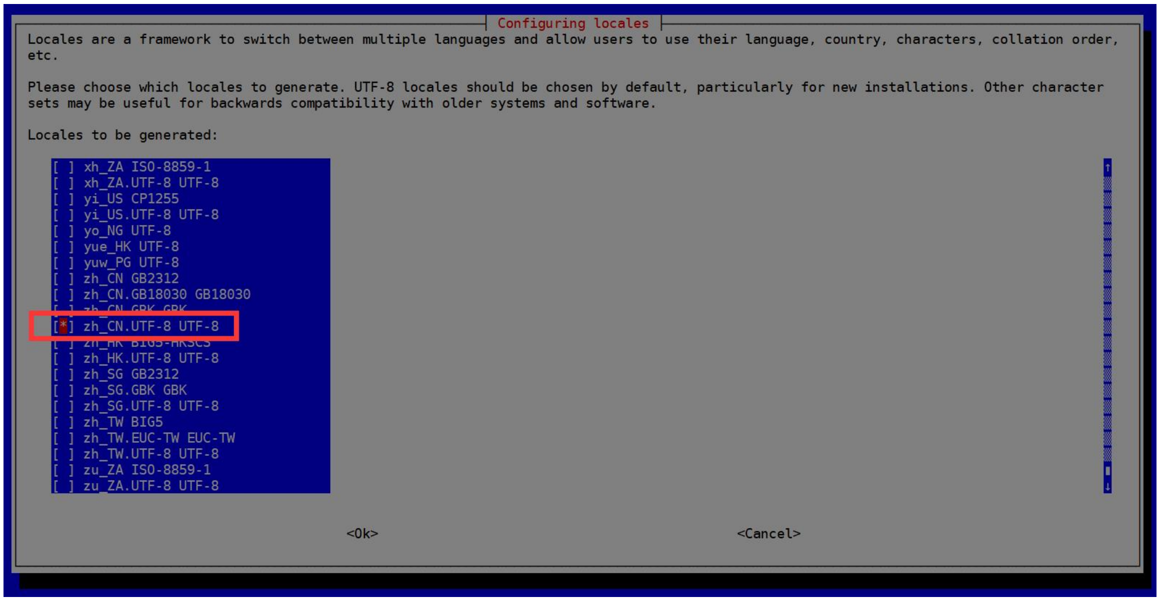
3. 21. 1. Debian system installation method

1) First set the default **locale** to Chinese.

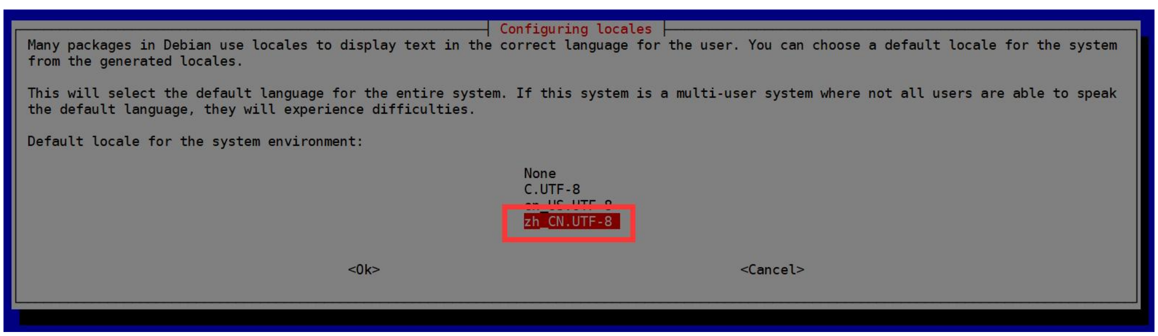
a. Enter the following command to start configuring **locale**.

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

b. Then select **zh_CN.UTF-8 UTF-8** in the pop-up interface (use the up and down arrow keys on the keyboard to move up and down, use the space bar to select, and finally use the Tab key to move the cursor to **<OK>**, and then press Enter).



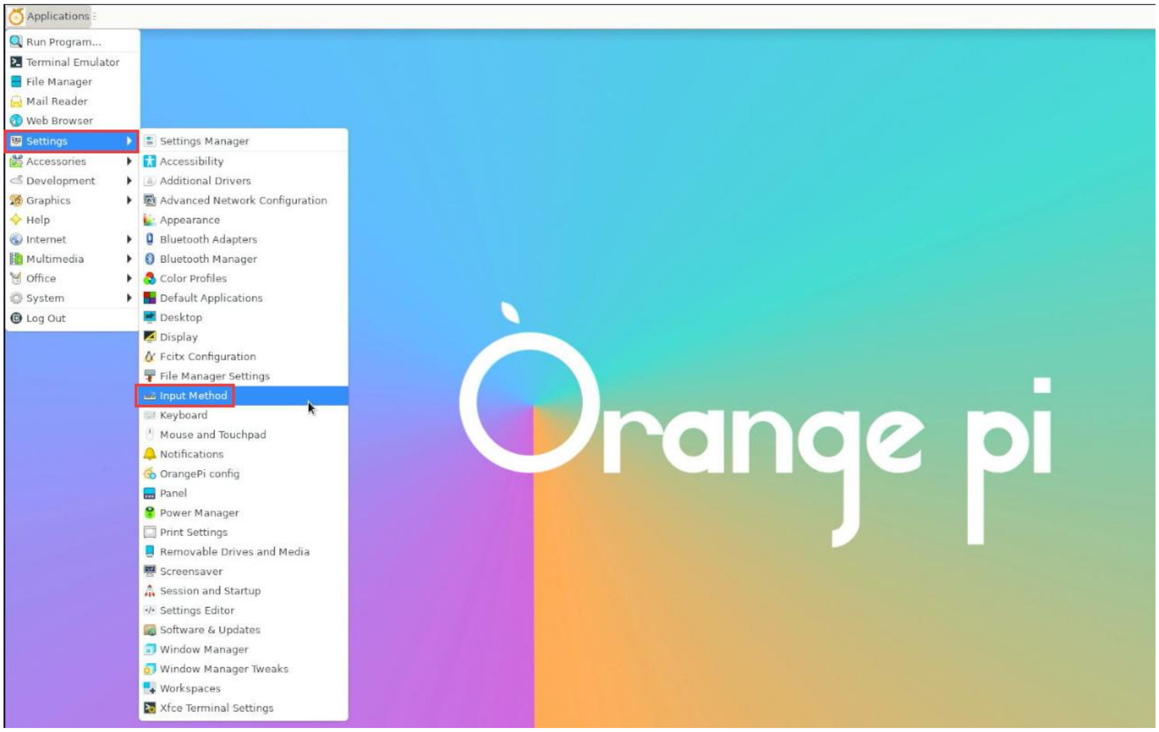
c. Then set the default **locale** to **zh_CN.UTF-8.**



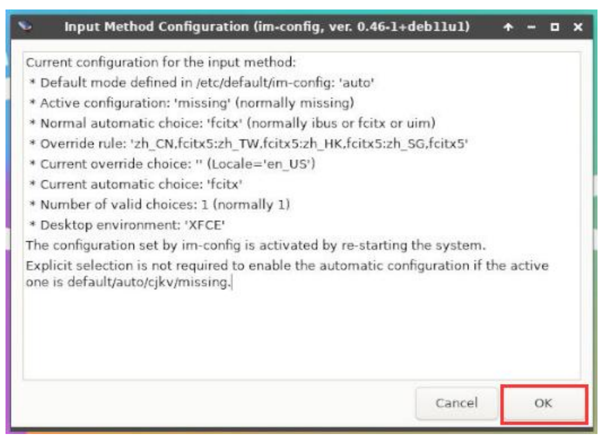
d. After exiting the interface, the **locale** setting will begin. The output displayed on the command line is as follows:

```
orangepi@orangepi:~$ sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
   en_US.UTF-8... done
   zh_CN.UTF-8... done
Generation complete.
```

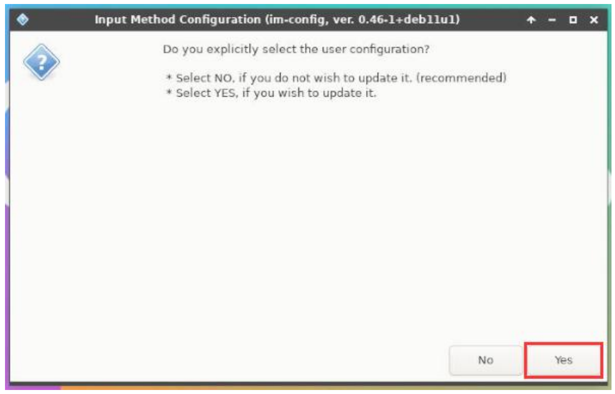
2) Then open **Input Method.**



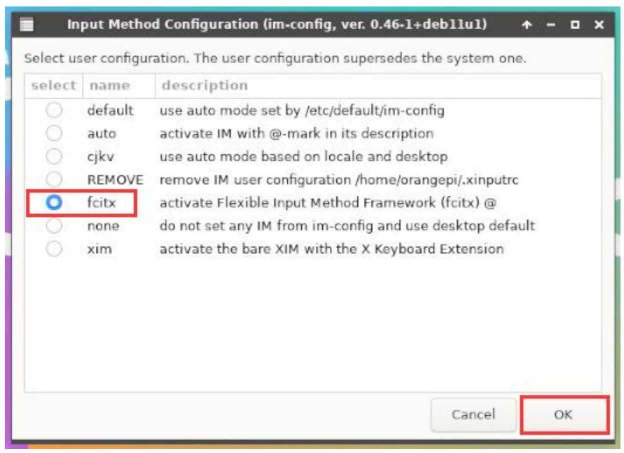
3) Then select **OK**.



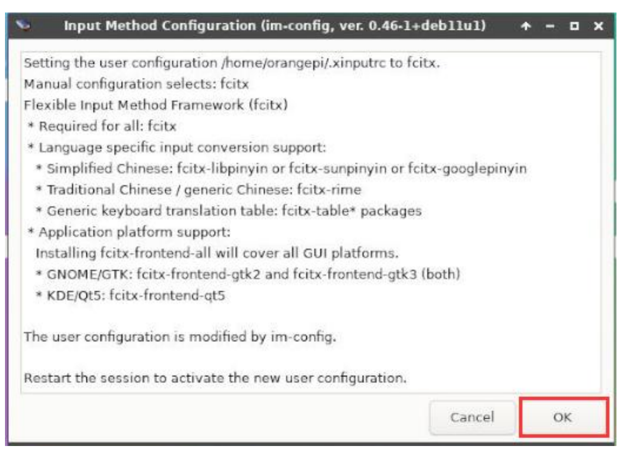
4) Then select **Yes.**



5) Then select **fcitx**.

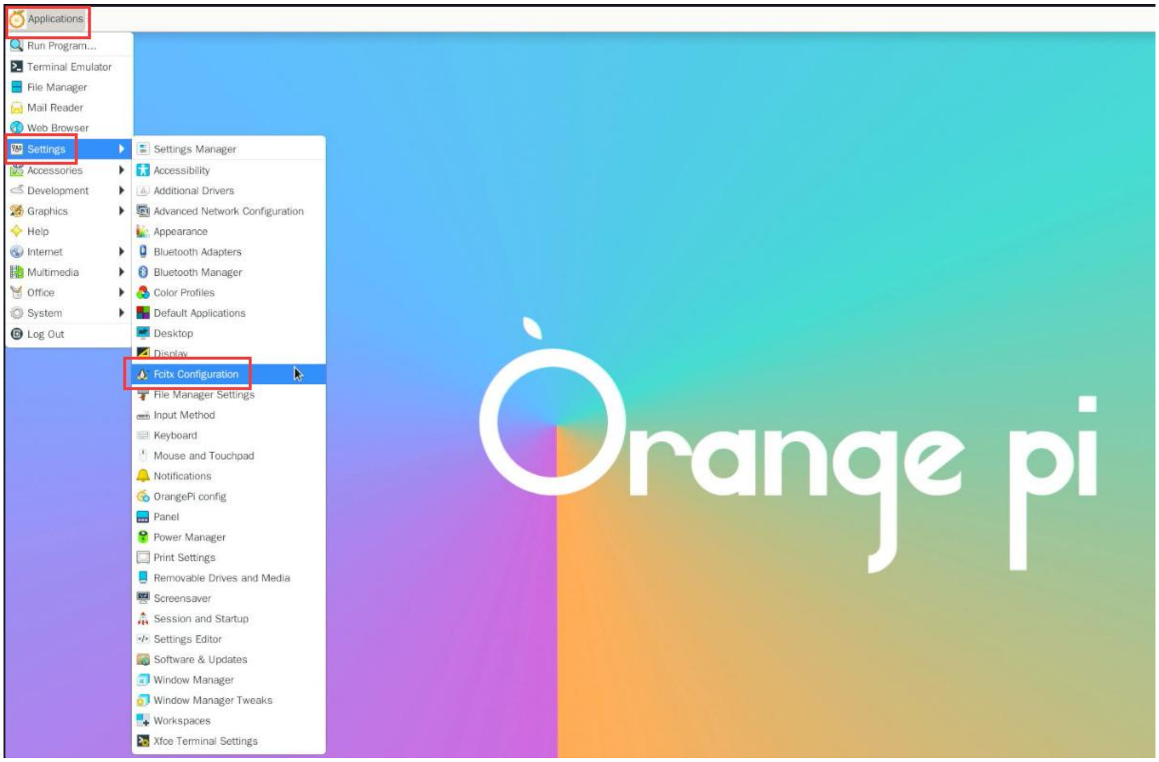


6) Then select **OK.**

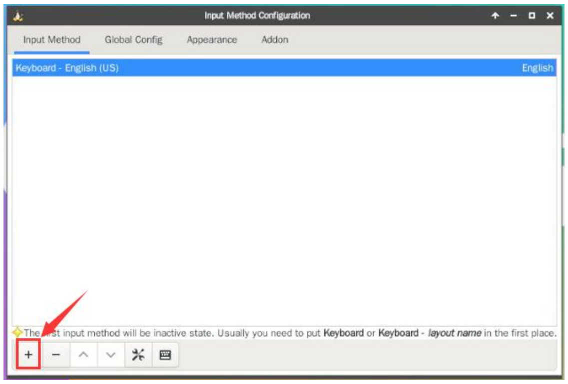


7) Then restart the Linux system to make the configuration take effect.

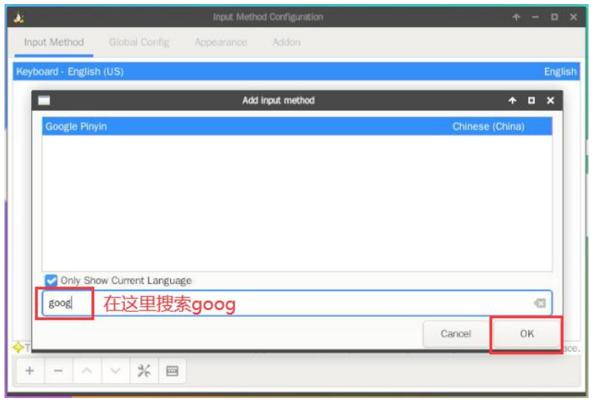
8) Then open **Fcitx configuration.**



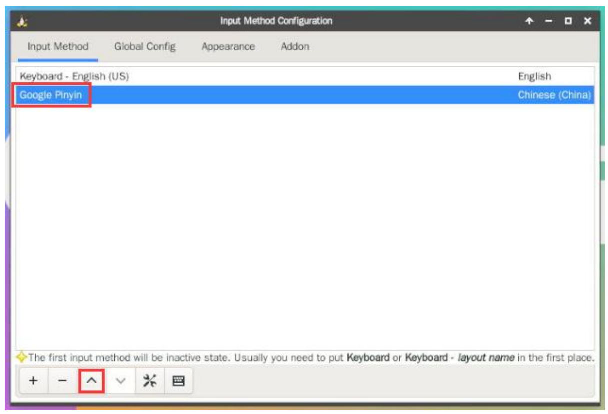
9) Then click the + sign in the location shown in the picture below.

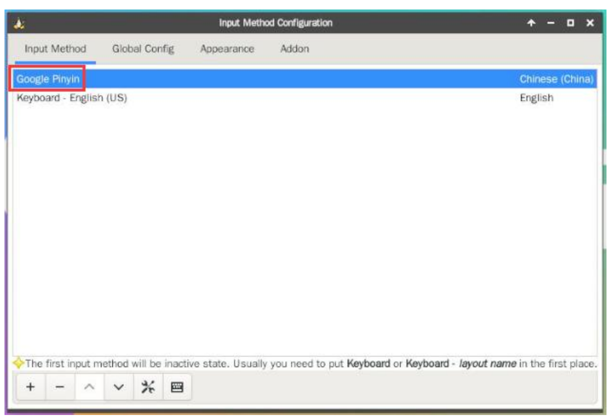


10) Then search **Google Pinyin** and click **OK**.



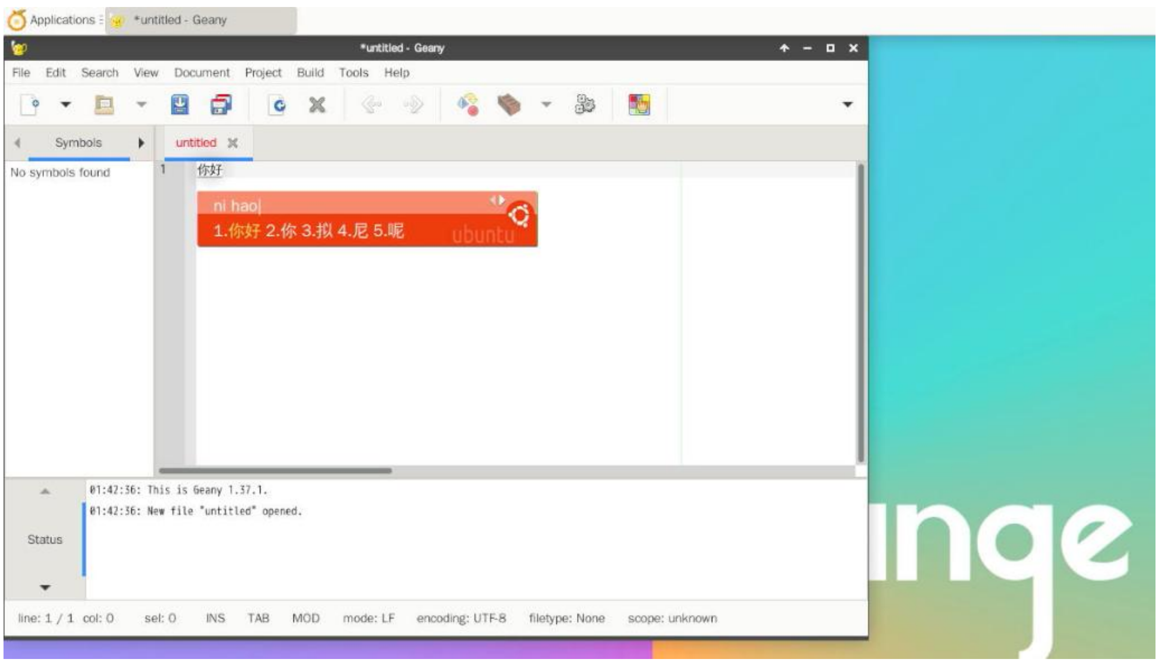
11) Then put **Google Pinyin** at the front.





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

13) The Chinese input method test is as follows:



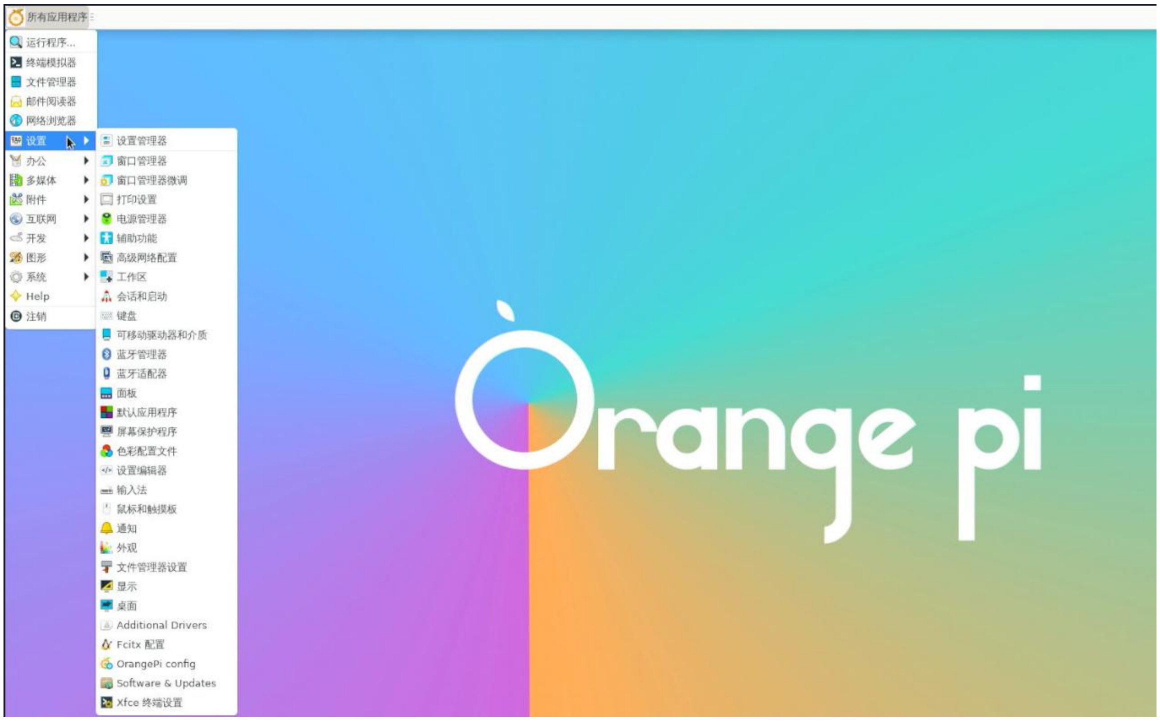
14) You can switch between Chinese and English input methods using the **Ctrl+Space** shortcut key.

15) If you need the entire system to display in Chinese, you can set the variables in **/etc/default/locale** to **zh_CN.UTF-8**.

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

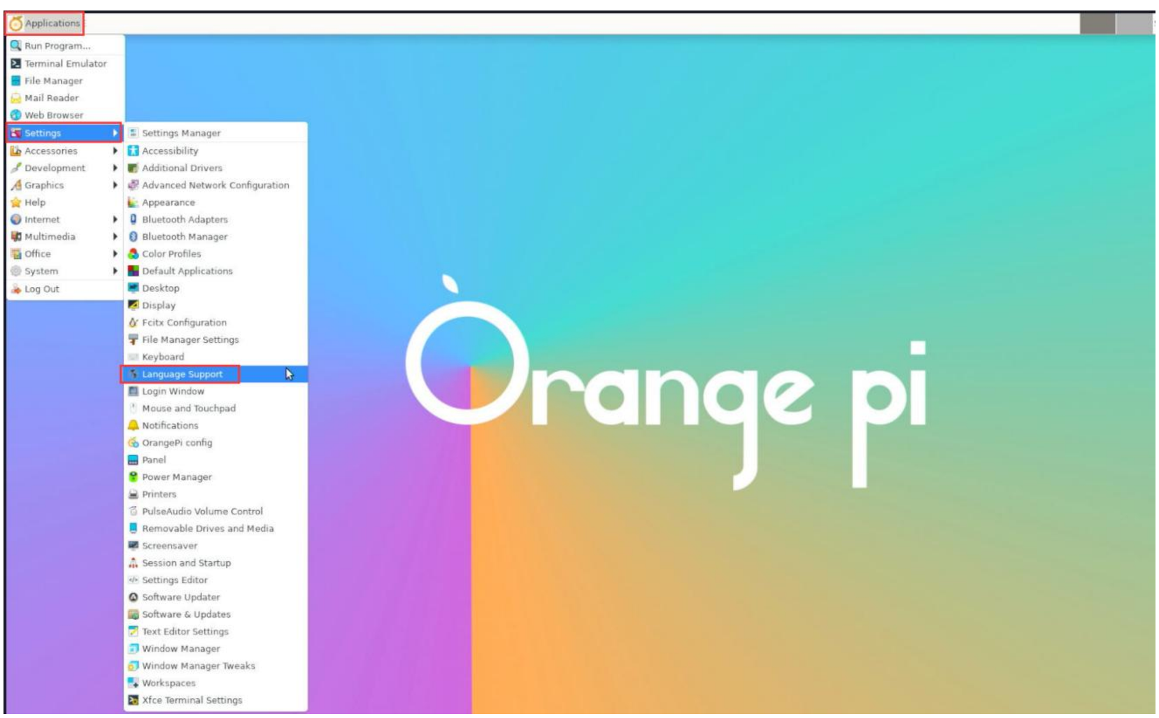
```
LANGUAGE=zh_CN.UTF-8
```

16) Then **restart the system** and you can see that the system is displayed in Chinese.

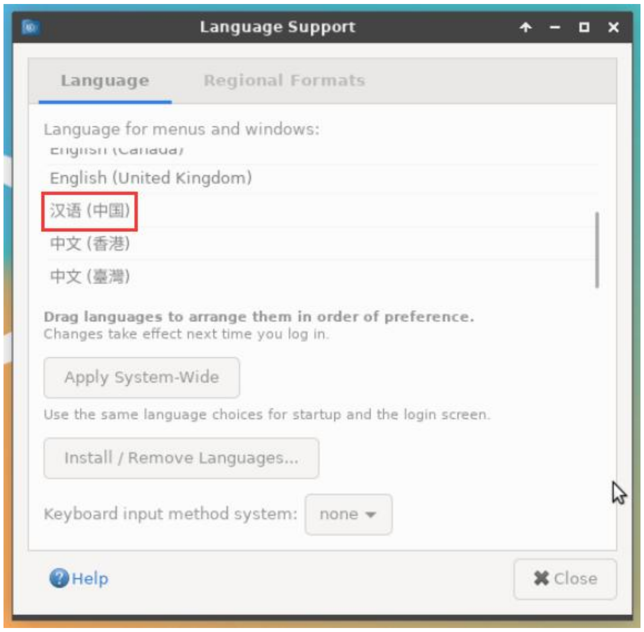


3. 21. 2. How to install Ubuntu 20.04 system

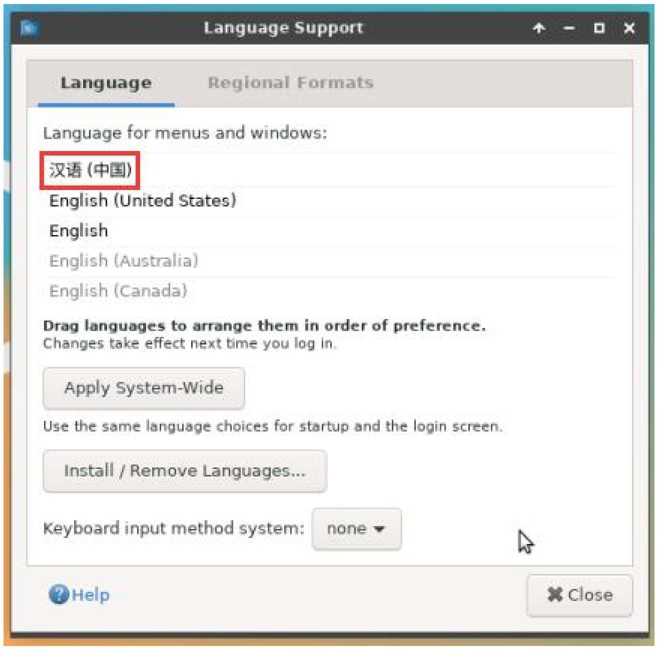
1) First open **Language Support.**



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

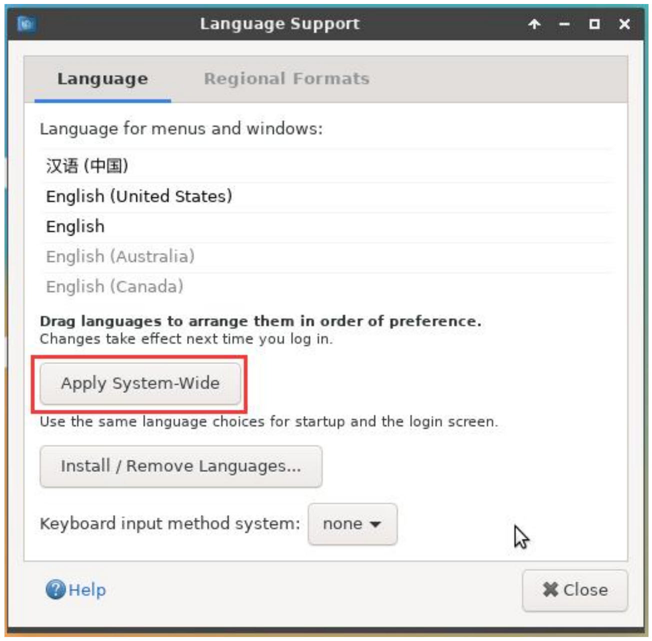


3) Then use the left mouse button to select **Chinese (China)** and hold it down, then drag it upwards to the starting position. The display after dragging is as shown below:

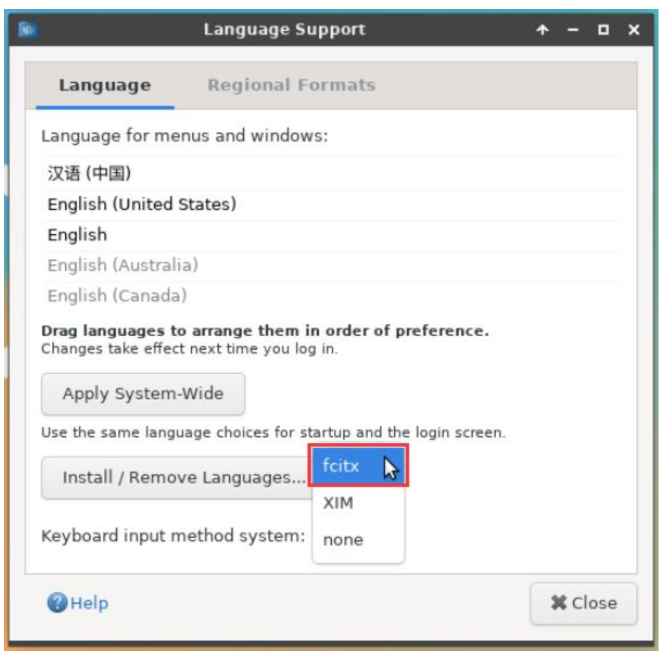


Note that this step is not easy to drag, please be patient and try a few more times.

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



5) Then set the **Keyboard input method system** to **fcitx**.

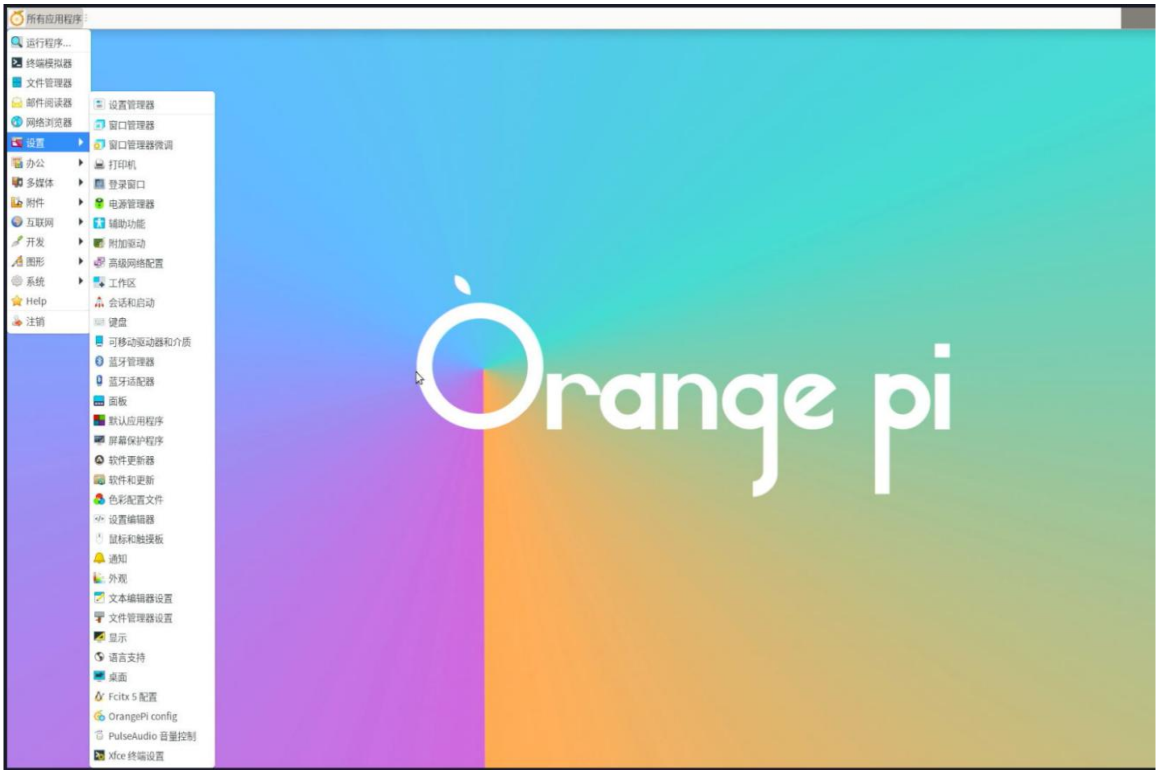


6) Then restart the Linux system to make the configuration take effect.

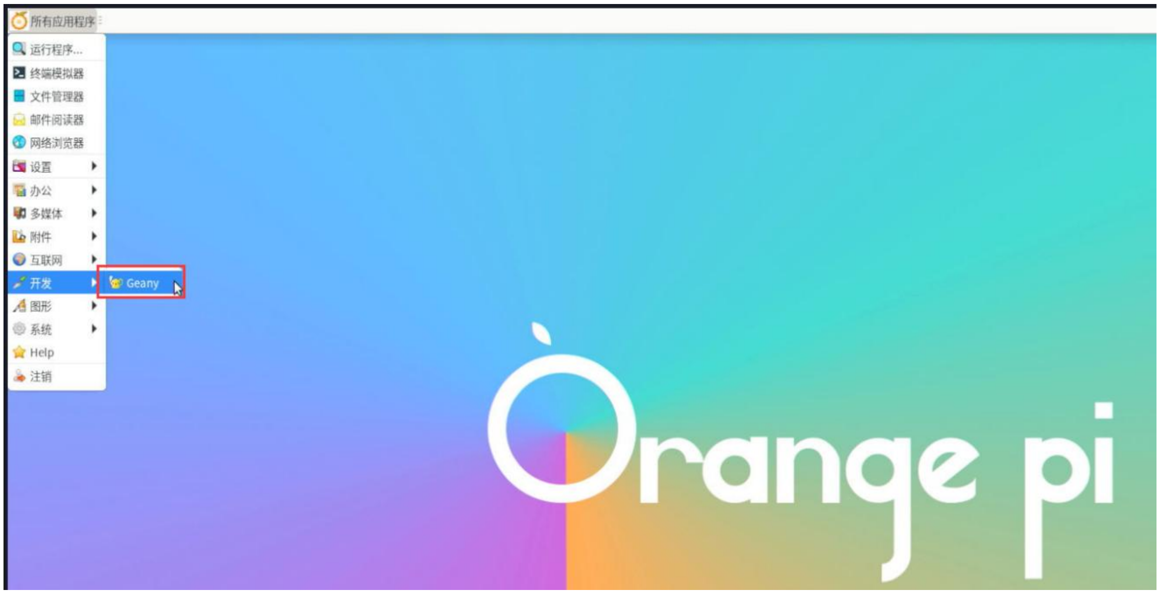
7) After re-entering the system, please select "**Do not ask me again**" in the interface below, and then decide whether to update the standard folders to Chinese according to

your preferences.

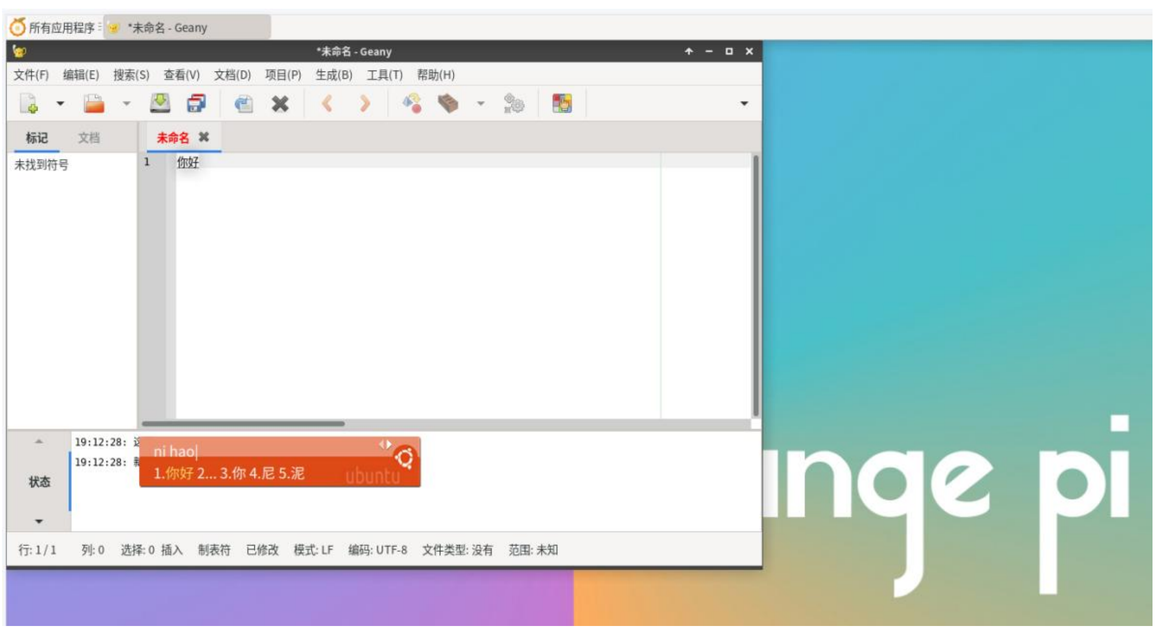
8) Then you can see that the desktop is displayed in Chinese.



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

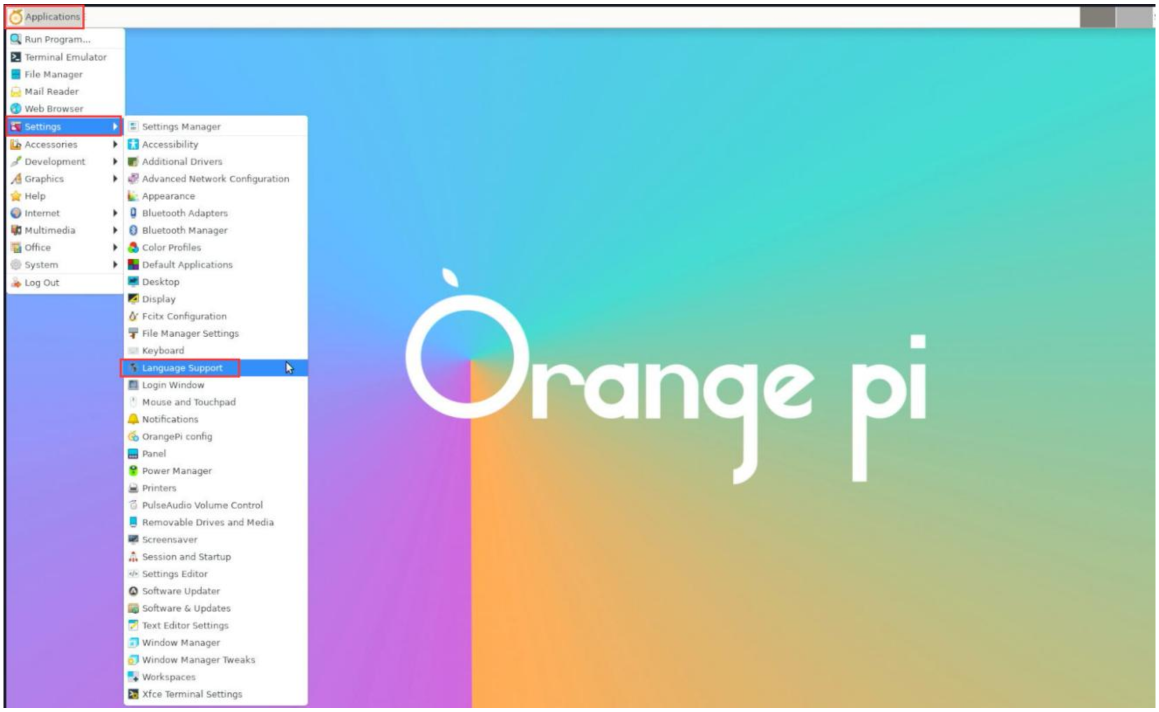


10) After opening **Geany**, the default input method is still English. We can switch to Chinese input method through the **Ctrl+Space** shortcut key, and then we can input Chinese.

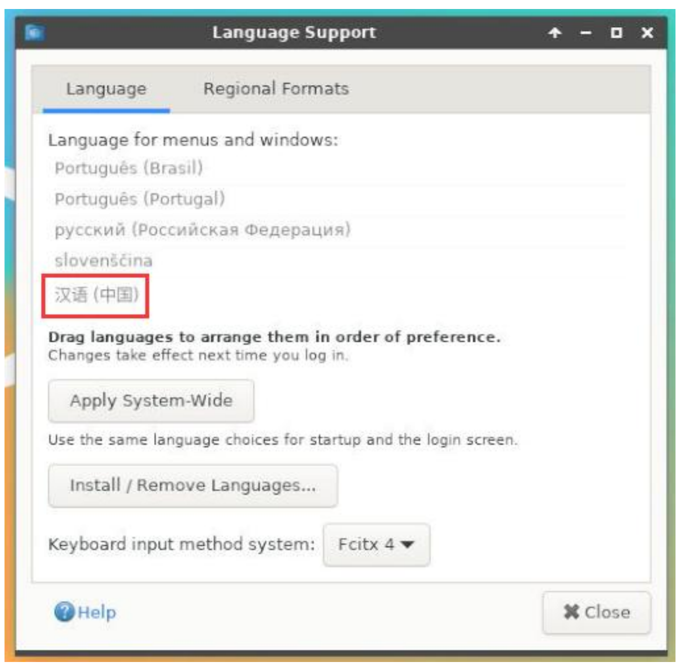


3. 21. 3. Installation method for Ubuntu 22.04 system

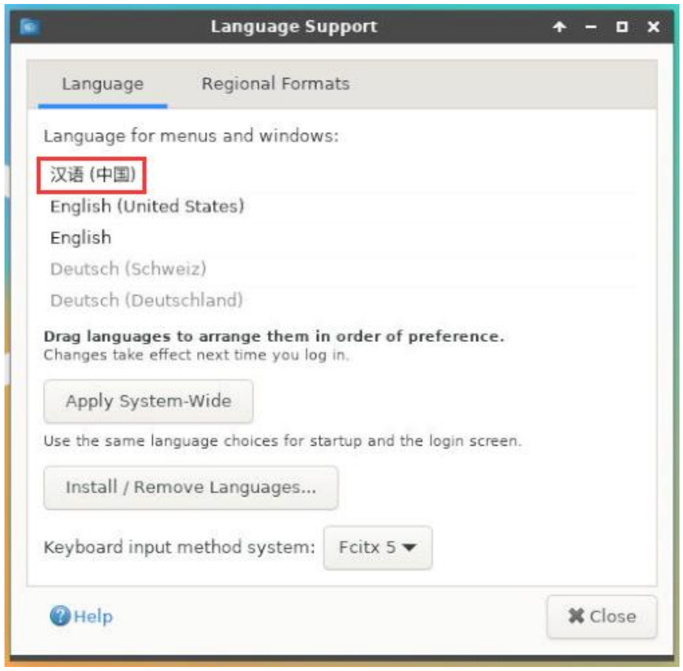
1) First open **Language Support**。



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

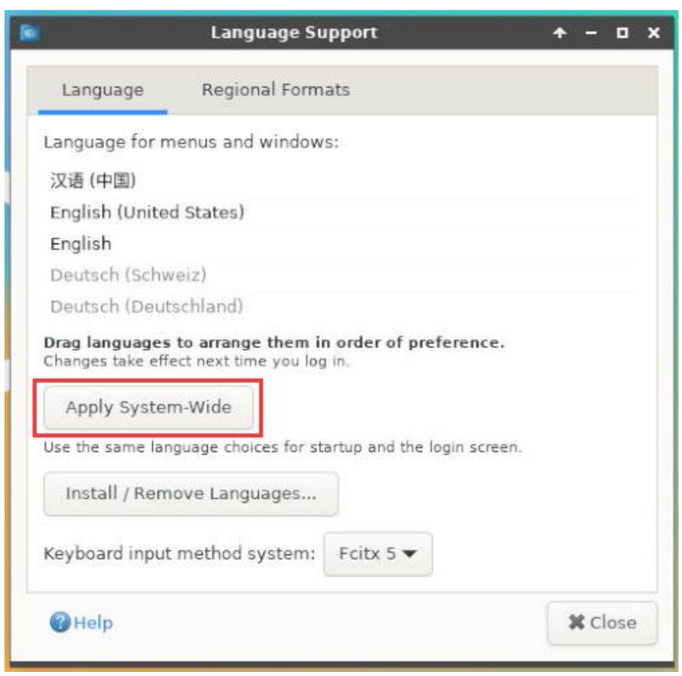


3) Then use the left mouse button to select **Chinese (China)** and hold it down, then drag it upwards to the starting position. The display after dragging is as shown below:



Note that this step is not easy to drag, please be patient and try a few more times.

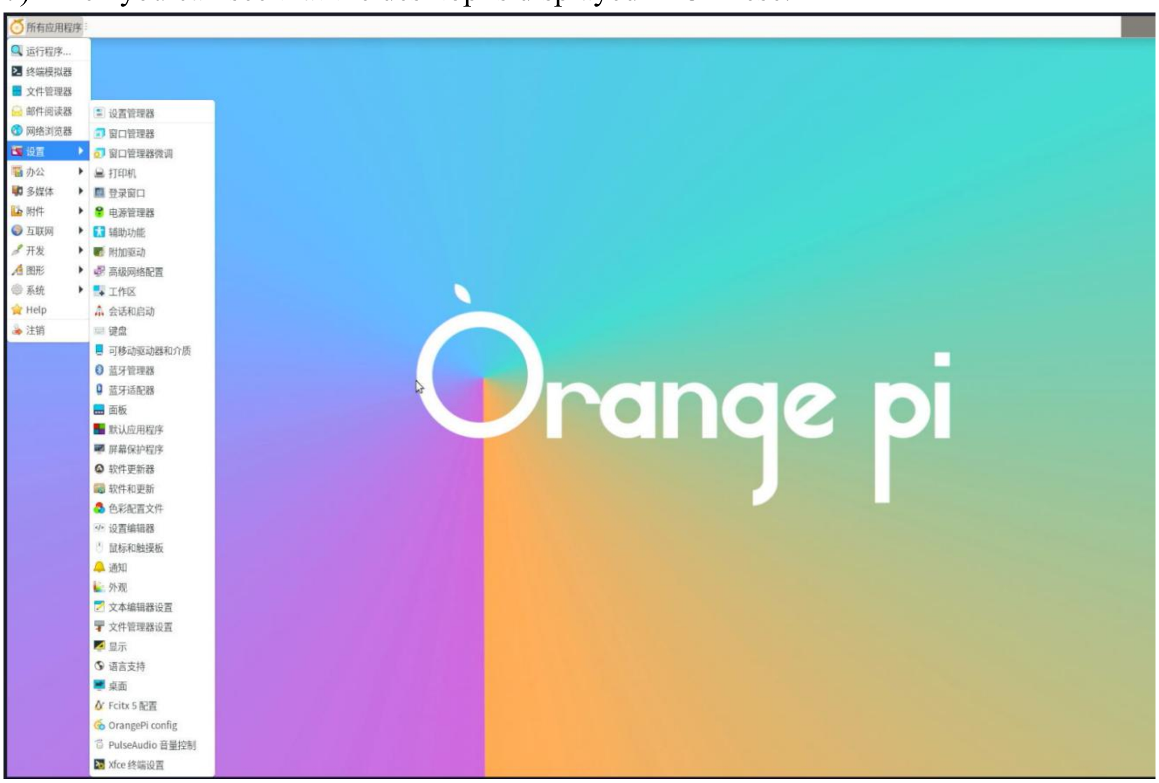
4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system.



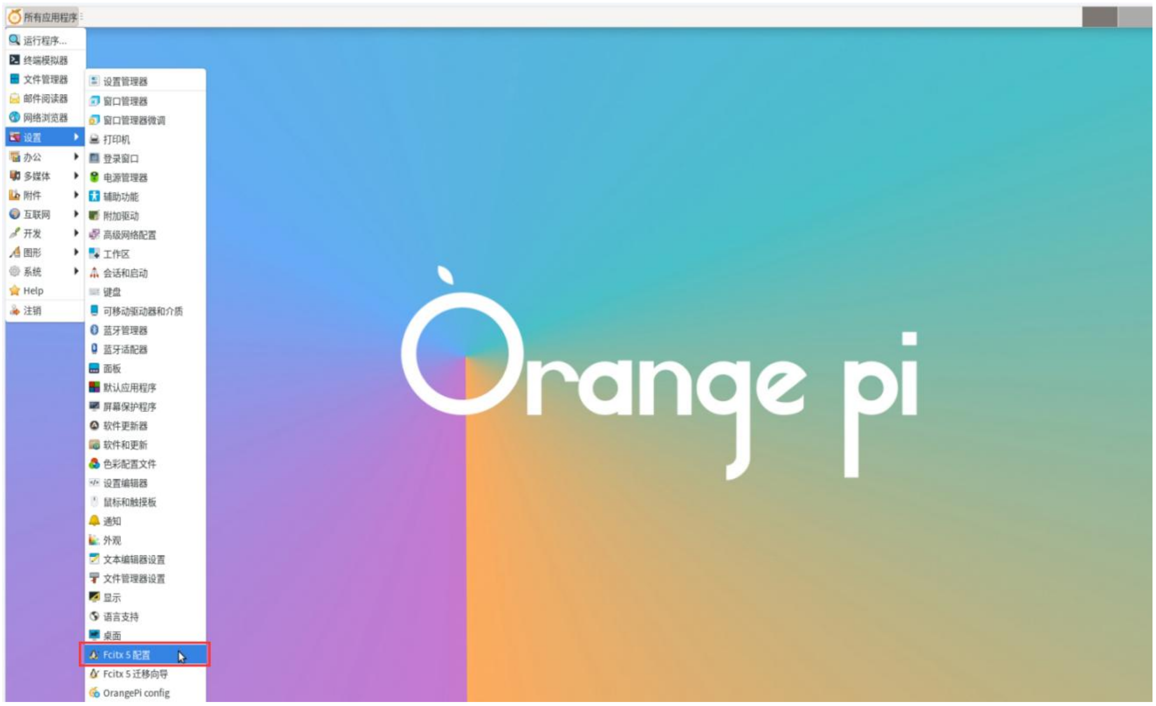
5) **Then restart the Linux system to make the configuration take effect**

6) After re-entering the system, please select "**Do not ask me again**" in the interface below, and then decide whether to update the standard folders to Chinese according to your preferences.

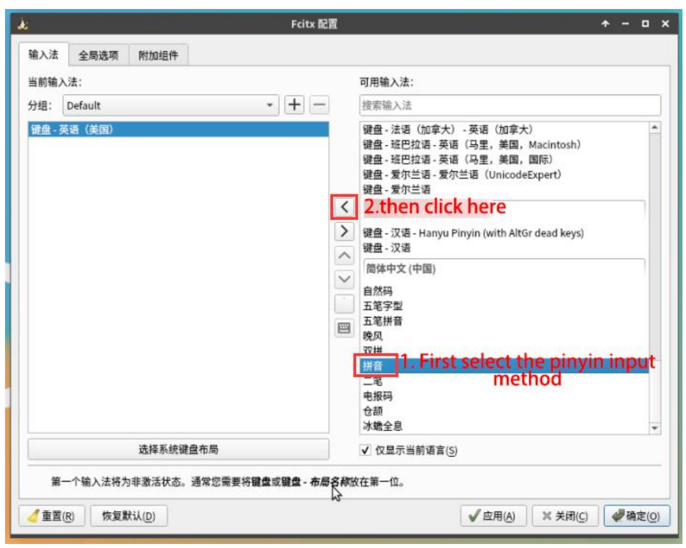
7) Then you can see that the desktop is displayed in Chinese.



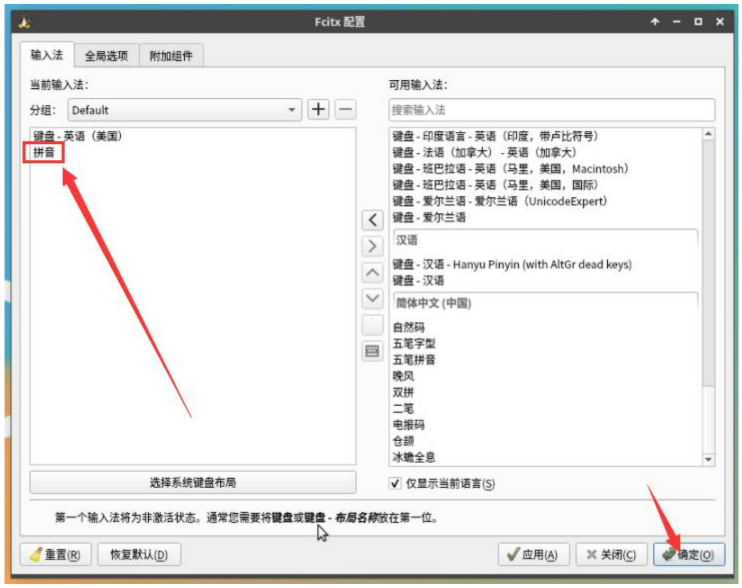
8) Then open the Fcitx5 configuration program.



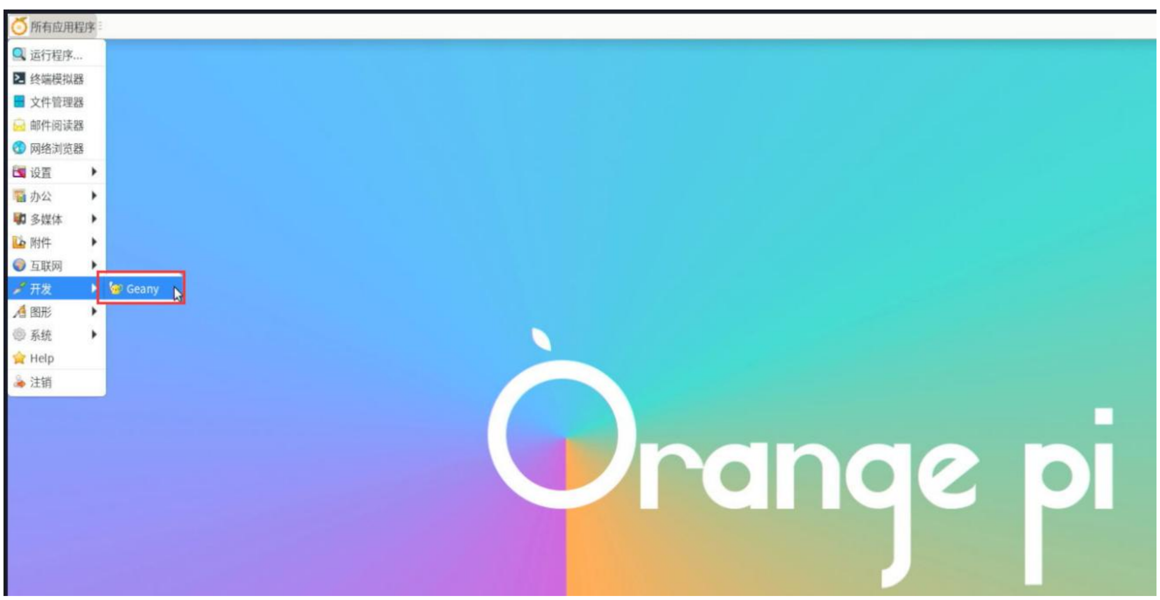
9) Then select Pinyin input method.



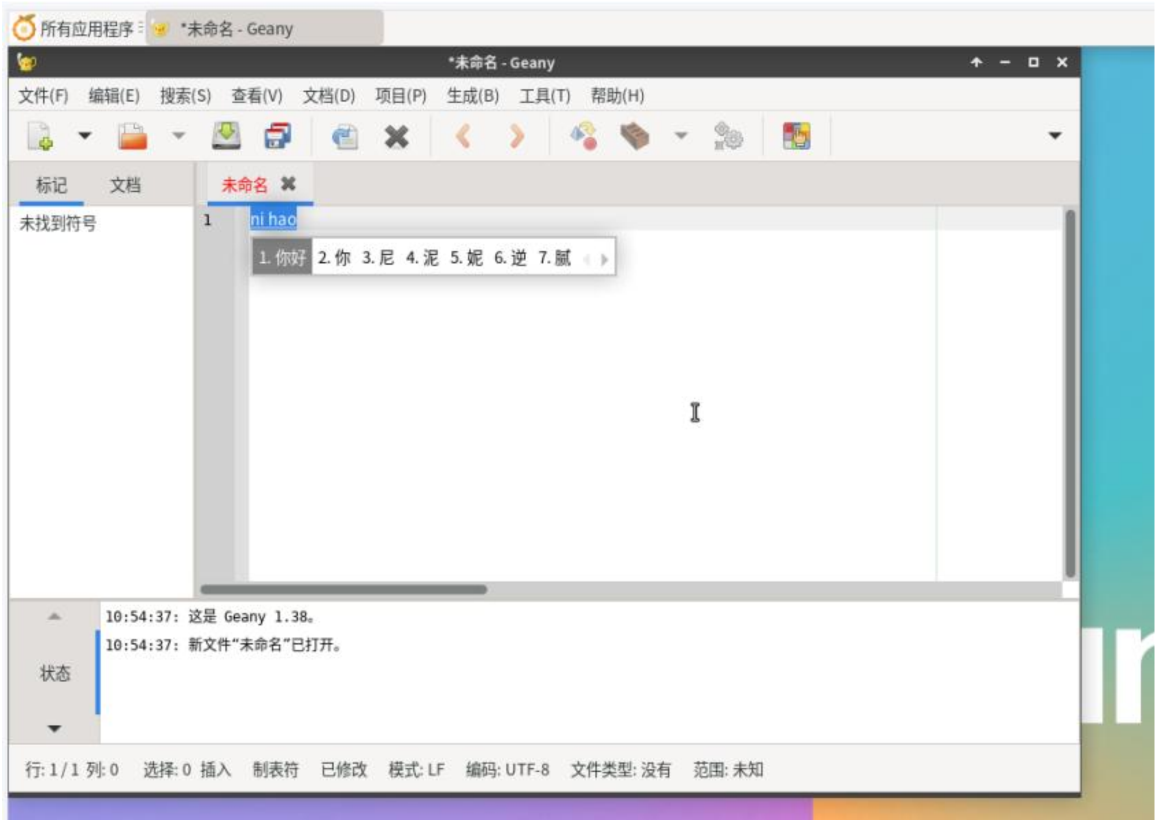
10) The interface after selection is as shown below, then click OK.



11) Then we can open **Geany** to test the Chinese input method. The opening method is as shown in the figure below:



12) After opening **Geany**, the default input method is still English. We can switch to Chinese input method through the **Ctrl+Space** shortcut key, and then we can input Chinese.



3. 22. How to remotely log in to the Linux system desktop

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

3. 22. 1. Remote login using NoMachine

Please make sure that the Ubuntu or Debian system installed on the development board is a desktop version. In addition, NoMachine also provides detailed usage documentation. It is strongly recommended to read this document to familiarize yourself with the use of NoMachine. The document link is as follows:

https://knowledgebase.nomachine.com/DT10R00166

NoMachine supports Windows, Mac, Linux, iOS and Android platforms, so we can use NoMachine to remotely log in and control the Orange Pi development board on multiple devices. The following demonstrates how to remotely log in to the Linux system desktop of the Orange Pi development board through NoMachine in

Windows. For installation methods on other platforms, please refer to the official documentation of NoMachine.

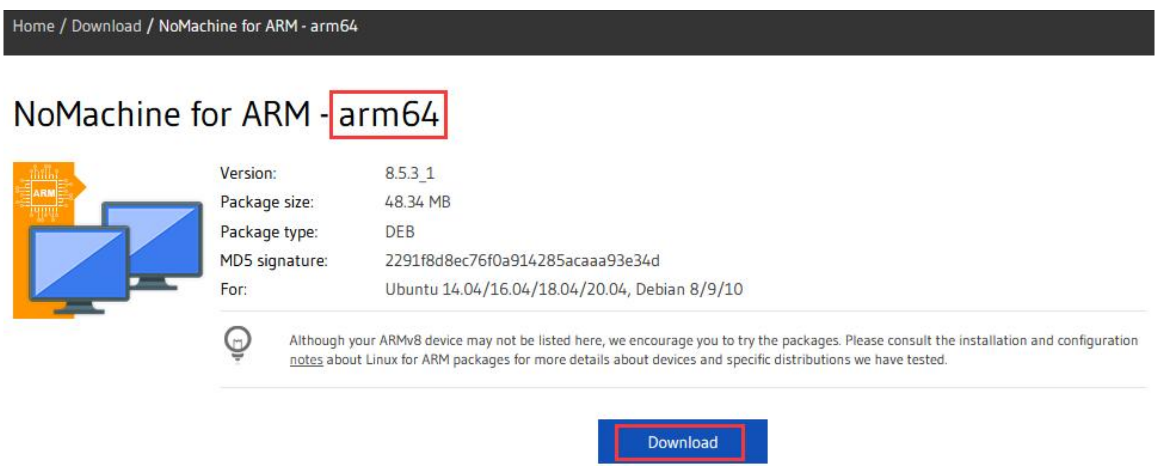
Before operation, please make sure that the Windows computer and the development board are in the same LAN and can log in to the Ubuntu or Debian system of the development board normally through SSH.

1) First download the installation package of the NoMachine software Linux **arm64** deb version, and then install it into the Linux system of the development board.

a. Since RK3588S is an ARMv8 SOC, we use Ubuntu or Debian as the system, so we need to download the **NoMachine for ARM ARMv8 DEB** installation package. The download link is as follows:

Note that this download link may change, please look for the Armv8/Arm64 version of the deb package.

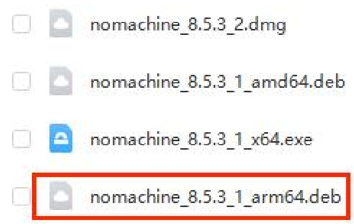
https://downloads.nomachine.com/download/?id=114&distro=ARM



b. In addition, you can also download the **NoMachine** installation package in the **official tool**.

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

Then download the arm64 version of the deb installation package.



c. Then upload the downloaded **nomachine_x.x.x_x_arm64.deb** to the Linux system of the development board

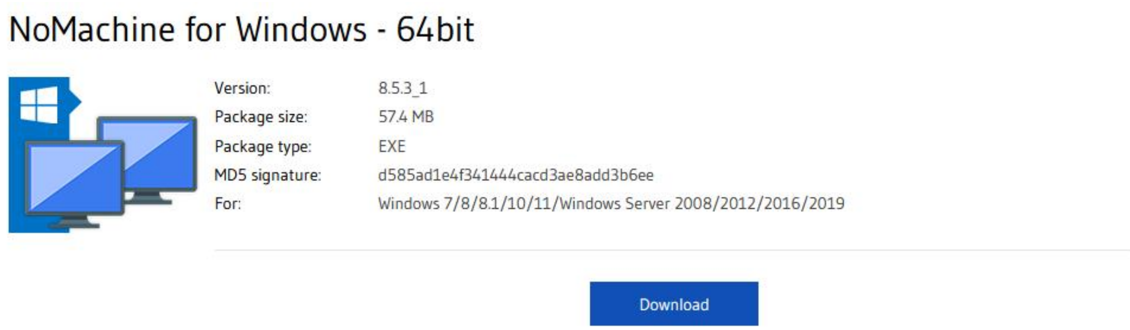
d. Then use the following command to install **NoMachine** in the Linux system of the development board.

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

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

Note that this download link may change.

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

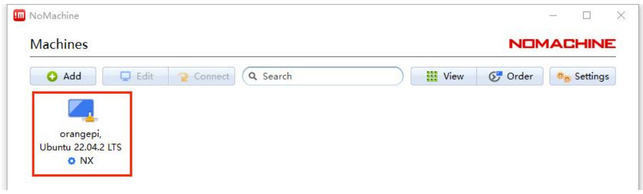


3) Then install NoMachine in Windows. **After the installation, please restart the computer**.

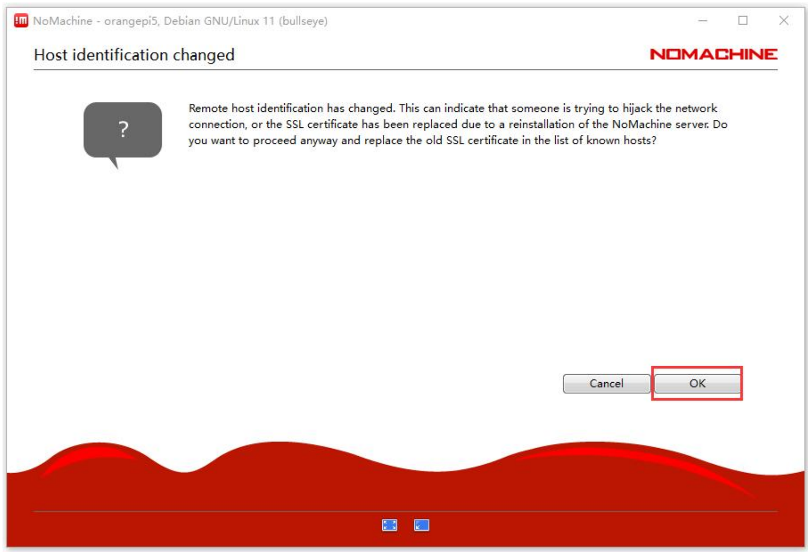
4) Then open **NoMachine** in Windows.

5) After NoMachine is started, it will automatically scan other devices with NoMachine installed in the LAN. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices. Then click the

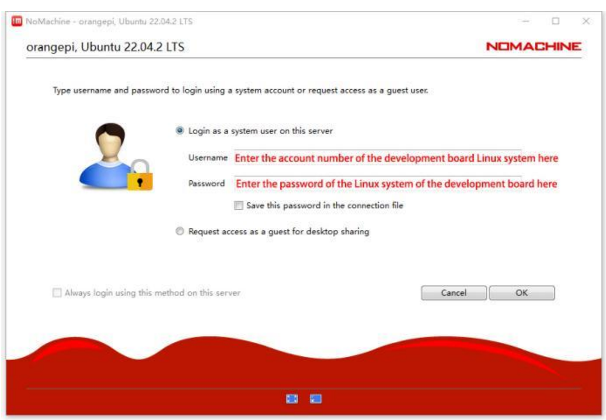
location shown in the red box in the figure below to start logging into the Linux system desktop of the development board.



6) Then click **OK**。

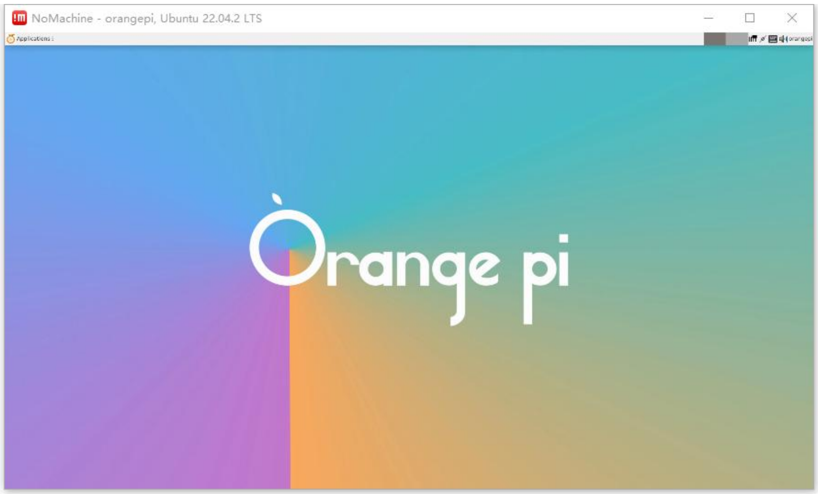


7) Then enter the user name and password of the development board Linux system in the corresponding position in the figure below, and click **OK** to start logging in.



8) Then click OK in the following interface.

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



3. 22. 2. Remote login using VNC

> **Before operation, please make sure that the Windows computer and the development board are in the same LAN and can log in to the Ubuntu or Debian system of the development board normally through SSH.**
>
> **There are many problems with testing VNC on Ubuntu 20.04, so please do not use this method.**

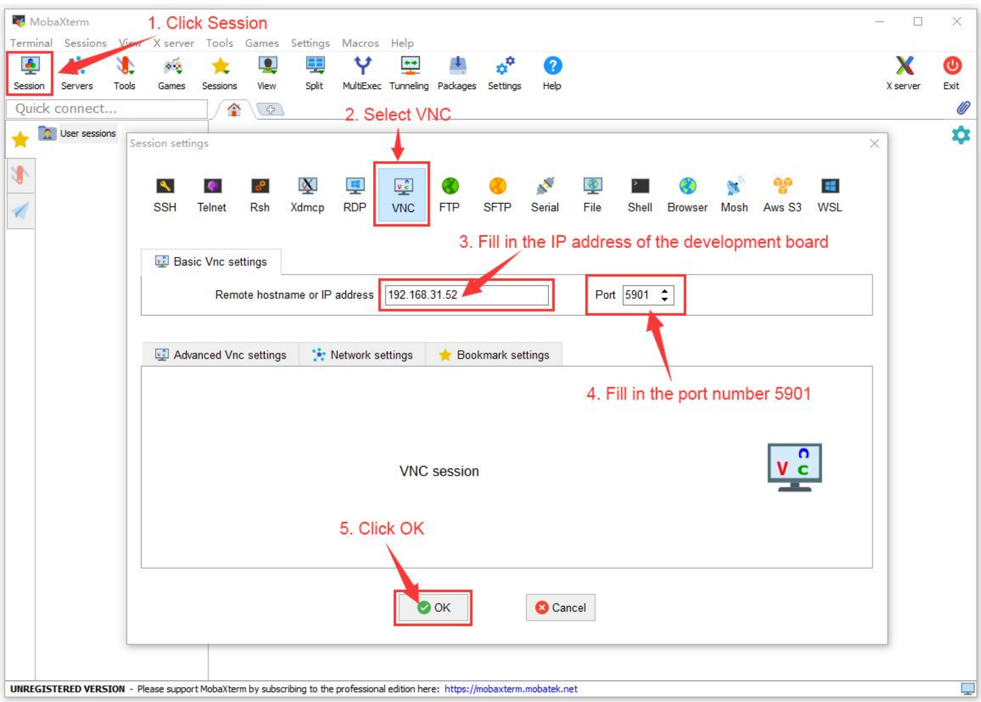
1) First run the **set_vnc.sh** script to set up vnc, **remember to add sudo permissions**.

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

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

New 'X' desktop is orangepicm5:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepicm5:1.log

Killing Xtightvnc process ID 3047
```

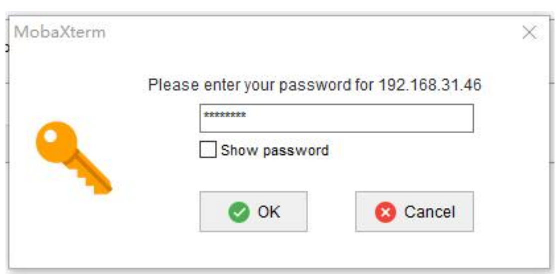
```
New 'X' desktop is orangepicm5:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/orangepicm5:1.log
```

2) The steps to use MobaXterm software to connect to the Linux system desktop of the development board are as follows:

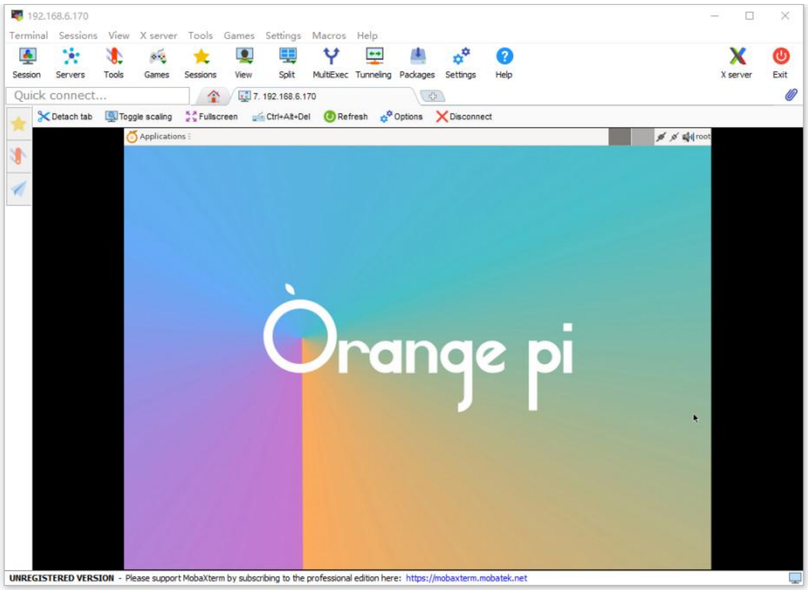
a. First click Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm.



b. Then enter the VNC password set previously.



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



3. 23. Test of some programming languages supported by Linux system

3. 23. 1. Debian Bullseye System

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

a. The versions of gcc are as follows:

```
orangepi@orangepi:~$ gcc --version
gcc (Debian 10.2.1-6) 10.2.1 20210110
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language.

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");
```

```
        return 0;
}
```

c. Then compile and run **hello_world.c**

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

2) Debian Bullseye has Python 3 installed by default.

a. The specific Python versions are as follows

```
orangepi@orangepi:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

b. Write the **hello_world.py** program in Python language.

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

c. The result of running **hello_world.py** is as follows:

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

3) Debian Bullseye does not install Java compilation tools and runtime environment by default.

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

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

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

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

c. Write a **hello_world.java** of Java version

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

```
        }
}
```

d. Then compile and run **hello_world.java**。

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

3. 23. 2. Debian Bookworm System

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

a. The versions of gcc are as follows:

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

b. Write the **hello_world.c** program in C language.

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**.

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

2) Debian Bookworm has Python 3 installed by default.

a. The specific Python versions are as follows

```
orangepi@orangepi:~$ python3
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

b. Write the **hello_world.py** program in Python language.

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

c. The result of running **hello_world.py** is as follows:

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

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

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

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

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

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

c. Write a Java version of **hello_world.java**。

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

d. Then compile and run **hello_world.java**。

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

3. 23. 3. Ubuntu Focal system

1) Ubuntu Focal is installed with the gcc compilation tool chain by default, which allows you to compile C language programs directly in the Linux system of the development board.

a. The versions of gcc are as follows:

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
```

b. Write the **hello_world.c** program in C language.

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**。

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

2) Ubuntu Focal has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.8.10 (default, Nov 14 2022, 12:59:47)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

b. Write the **hello_world.py** program in Python language.

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

c. The result of running **hello_world.py** is as follows:

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

3) Ubuntu Focal does not have Java compilation tools and runtime environment installed by default.

a. You can use the following command to install openjdk-17.

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

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

```
orangepi@orangepi:~$ java --version
openjdk 17.0.2 2022-01-18
OpenJDK Runtime Environment (build 17.0.2+8-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 17.0.2+8-Ubuntu-120.04, mixed mode, sharing)
```

c. Write a Java version of **hello_world.java**。

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

d. Then compile and run **hello_world.java**。

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

3. 23. 4. Ubuntu Jammy system

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

a. The version of gcc is as follows：

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.2.0-19ubuntu1) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.   There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

b. Write the **hello_world.c** program in C language.

```
orangepi@orangepi:~$ vim hello_world.c
#include <stdio.h>
```

```
int main(void)
{
        printf("Hello World!\n");

        return 0;
}
```

c. Then compile and run **hello_world.c**。

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

5) Ubuntu Jammy has Python 3 installed by default.

a. The specific version of Python3 is as follows:

```
orangepi@orangepi:~$ python3
Python 3.10.4 (main, Apr  2 2022, 09:04:19) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

b. Write the **hello_world.py** program in Python language.

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

c. The result of running **hello_world.py** is as follows：

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

6) Ubuntu Jammy does not have Java compilation tools and runtime environment installed by default.

a. You can use the following command to install openjdk-18.

```
orangepi@orangepi:~$ sudo apt install -y openjdk-18-jdk
```

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

```
orangepi@orangepi:~$ java --version
openjdk 18-ea 2022-03-22
OpenJDK Runtime Environment (build 18-ea+36-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 18-ea+36-Ubuntu-1, mixed mode, sharing)
```

c. Write a Java version of **hello_world.java**。

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

d. Then compile and run **hello_world.java**。

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

3. 24. QT installation method

1) Use the following script to install QT5 and QT Creator.

```
orangepi@orangepi:~$ install_qt.sh
```

2) After installation, the QT version number will be automatically printed.

a. The Qt version that comes with Ubuntu 20.04 is **5.12.8**.

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.12.8 in /usr/lib/aarch64-linux-gnu
```

b. The QT version that comes with Ubuntu 22.04 is **5.15.3**.

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.3 in /usr/lib/aarch64-linux-gnu
```

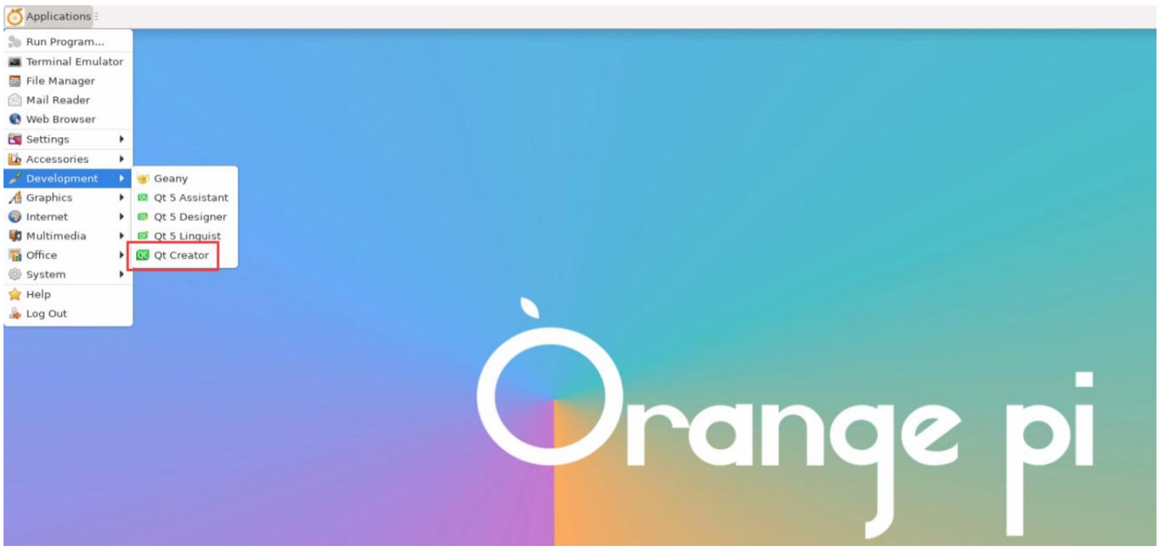
c. The QT version that comes with Debian11 is **5.15.2**.

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.2 in /usr/lib/aarch64-linux-gnu
```

d. The QT version that comes with Debian12 is **5.15.8**.

```
orangepi@orangepi:~$ install_qt.sh
......
QMake version 3.1
Using Qt version 5.15.8 in /usr/lib/aarch64-linux-gnu
```

3) Then you can see the QT Creator startup icon in **Applications**.



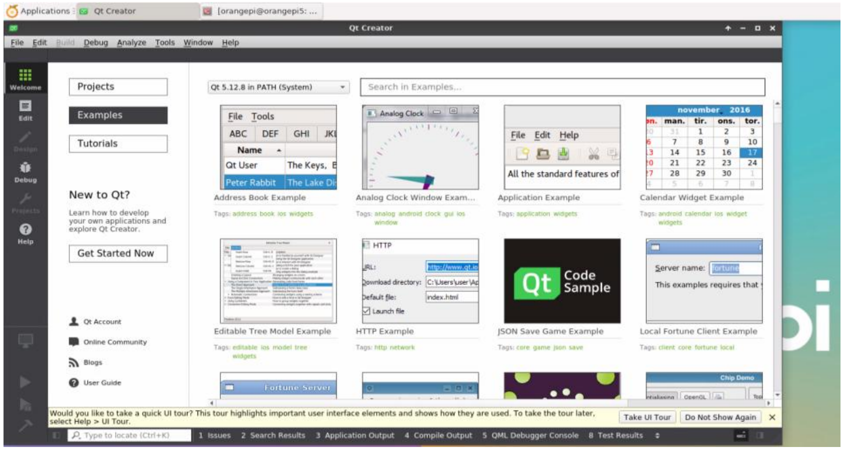
You can also use the following command to open QT Creator.

```
orangepi@orangepi:~$ qtcreator
```

During the startup of QT and QT applications, if the following error is prompted, please ignore it directly. This error will not affect the operation of the application.

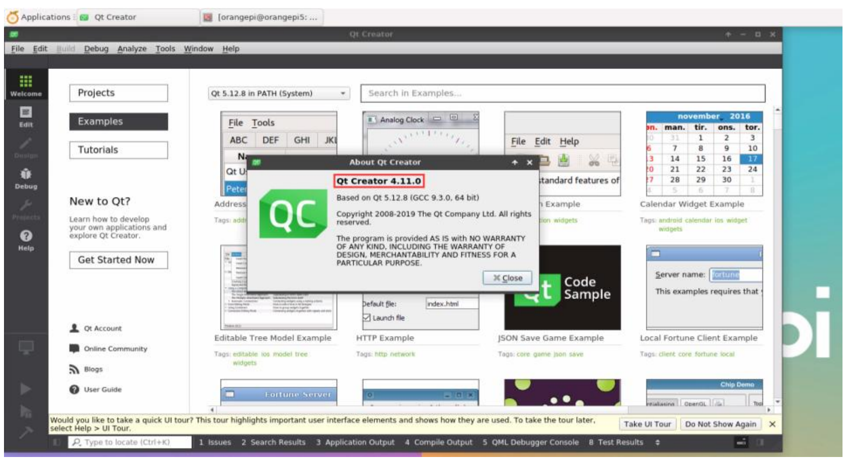
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip

4) The interface after QT Creator is opened is as follows:

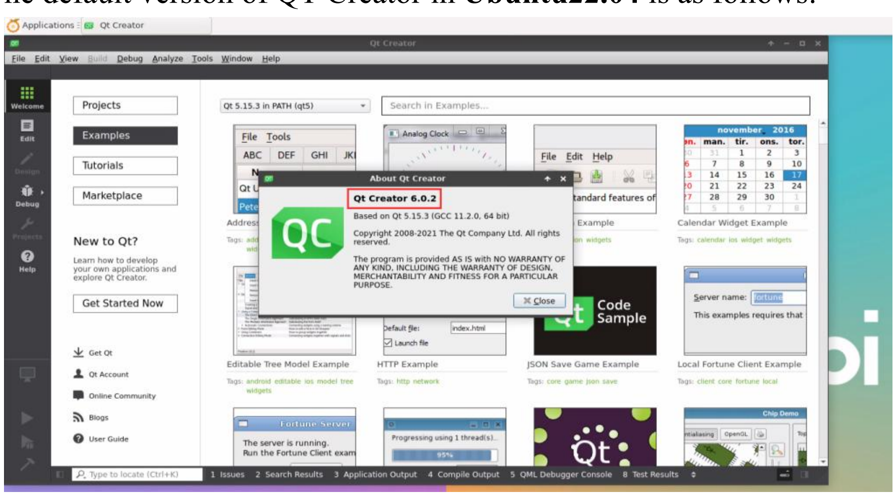


5) The version of QT Creator is as follows:

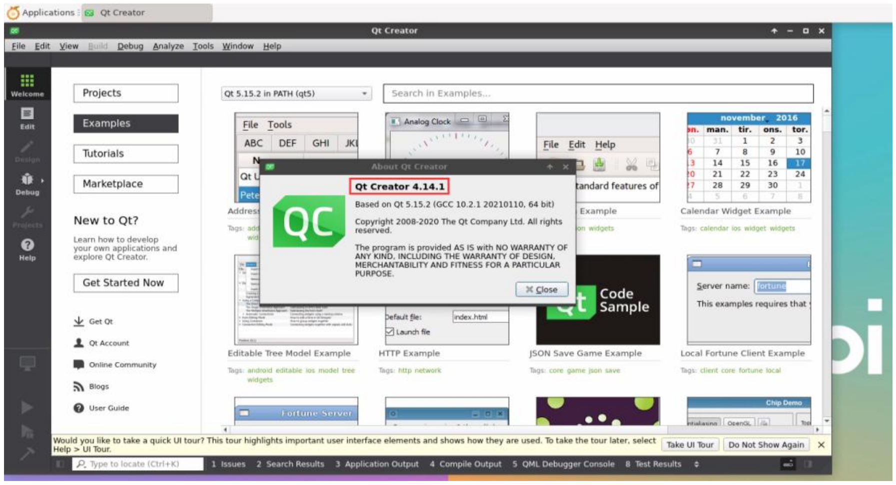
a. The default version of QT Creator in **Ubuntu20.04** is as follows:



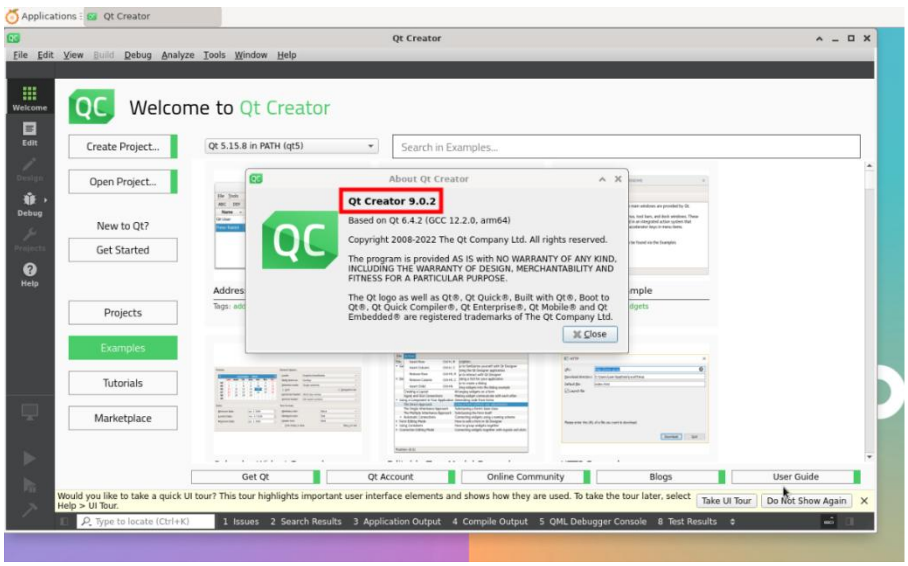
b. The default version of QT Creator in **Ubuntu22.04** is as follows:



c. The default version of QT Creator in **Debian11** is as follows:

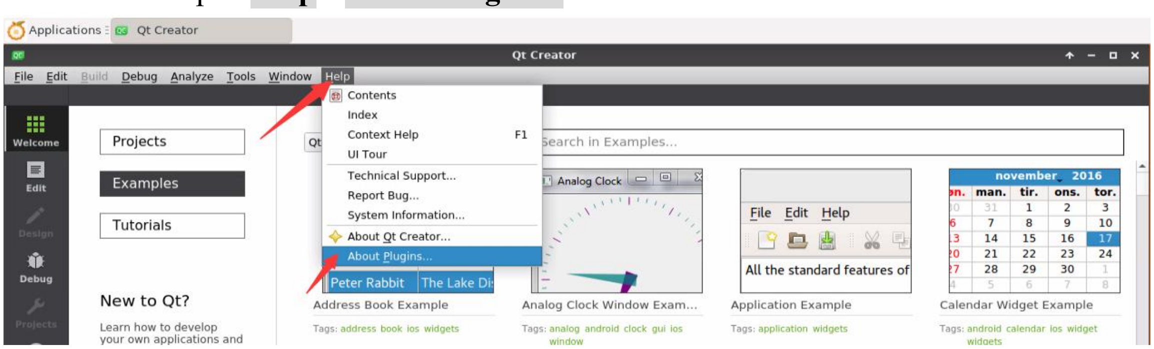


d. The default version of QT Creator in **Debian12** is as follows:

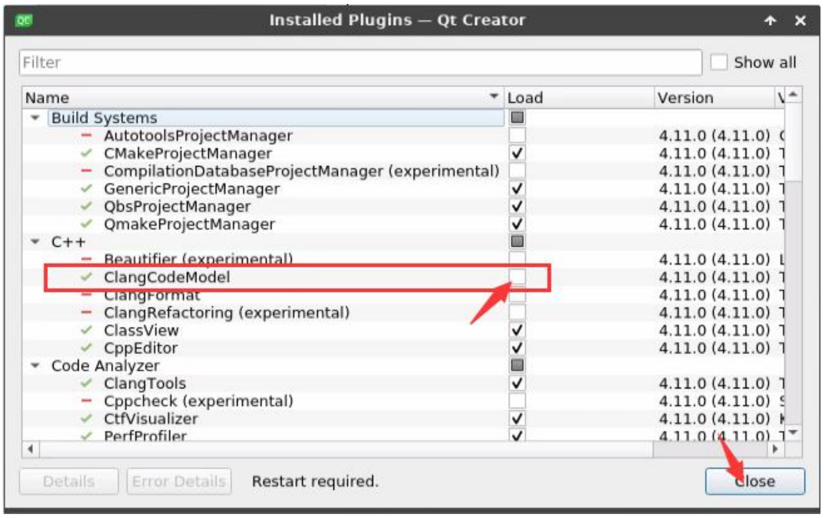


6) Then set QT。

a. First open **Help**->**About Plugins...**。



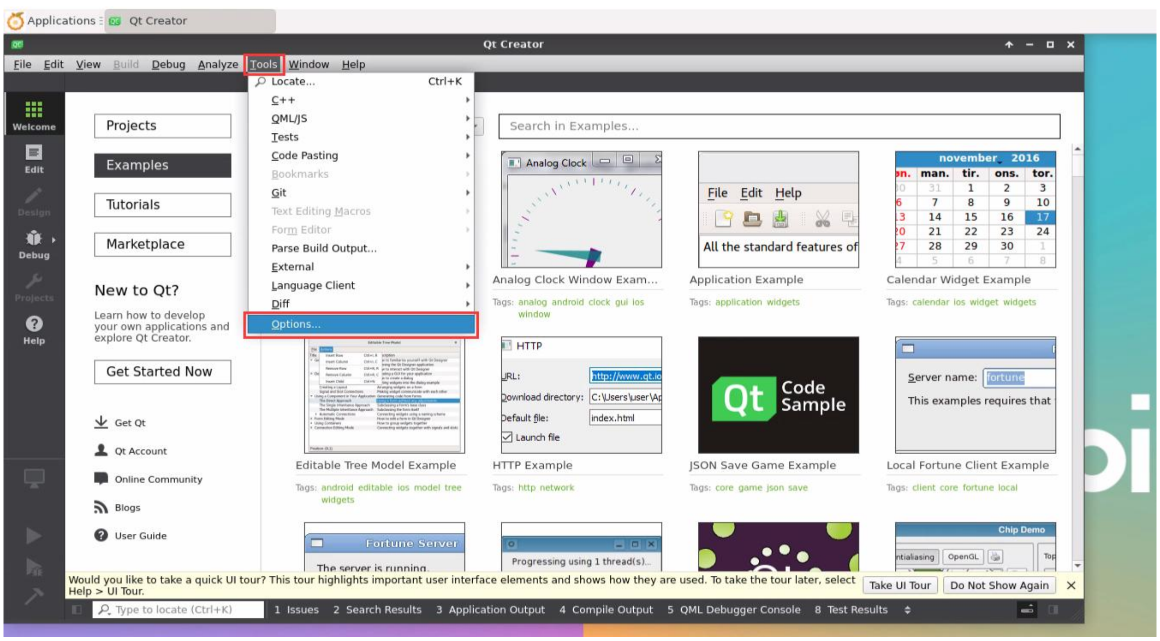
b. Then remove the check mark of **ClangCodeModel**.

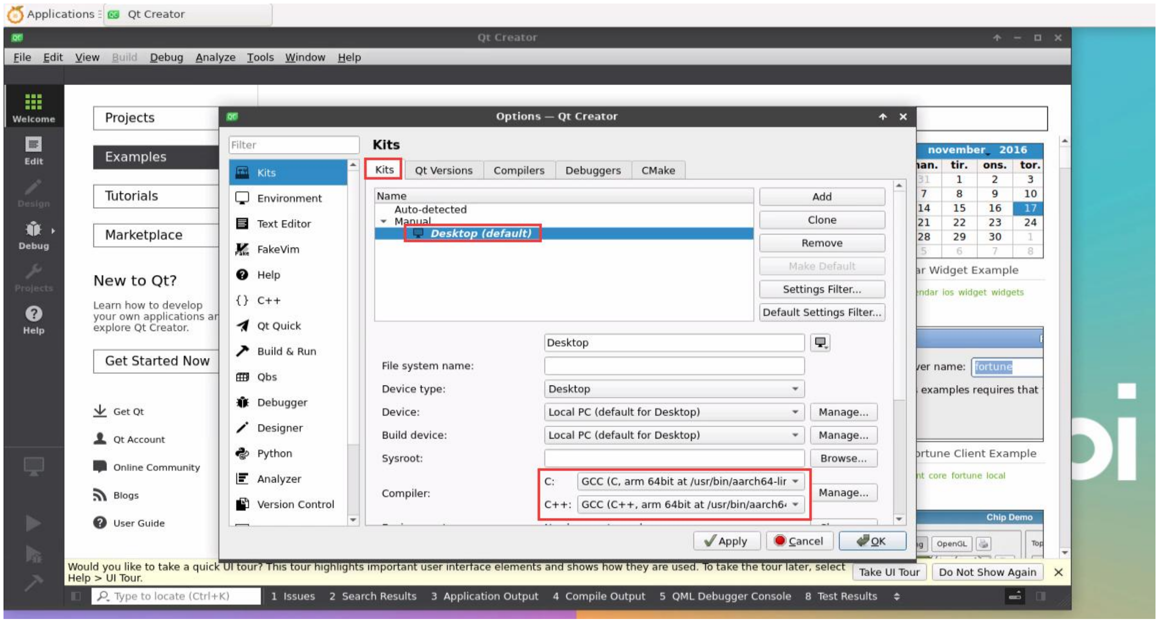


c. **After setting, you need to restart QT Creator.**

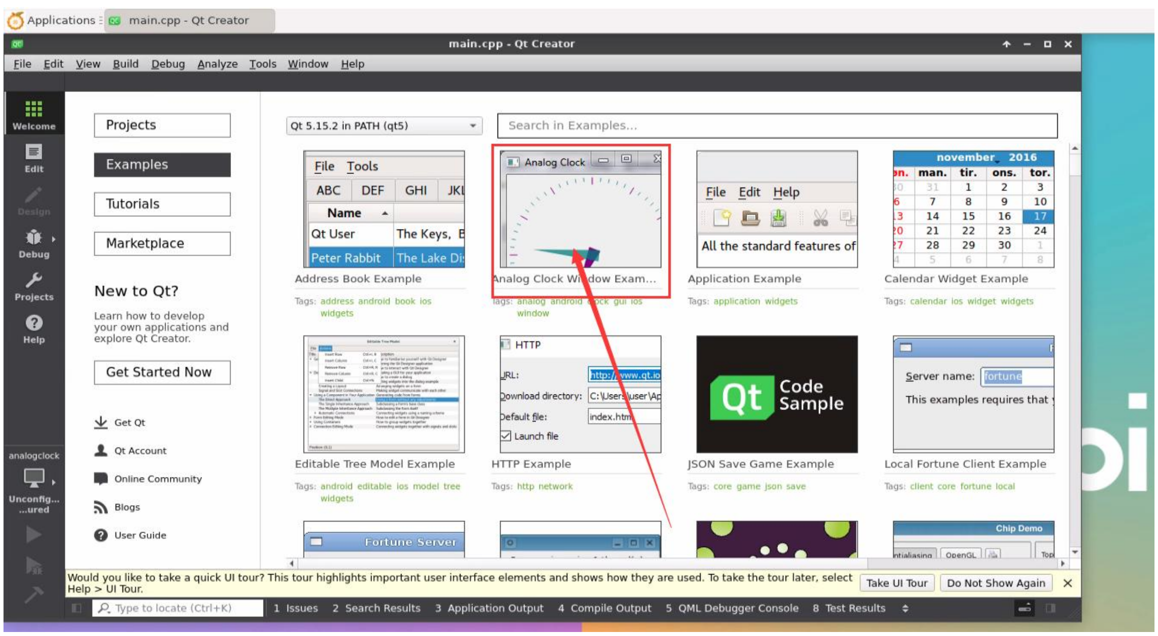
d. Then make sure that QT Creator uses the GCC compiler. If it defaults to Clang, change it to GCC.

For Debian 12, please skip this step.

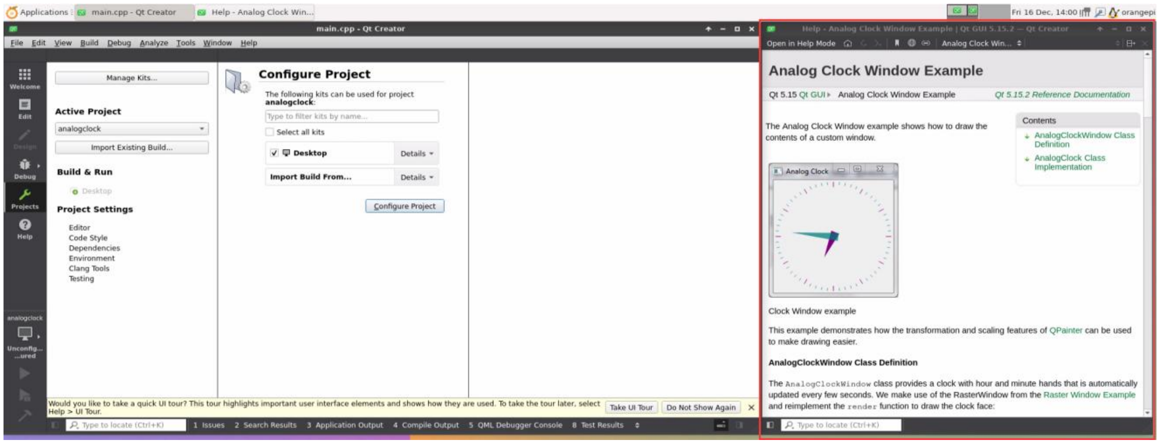




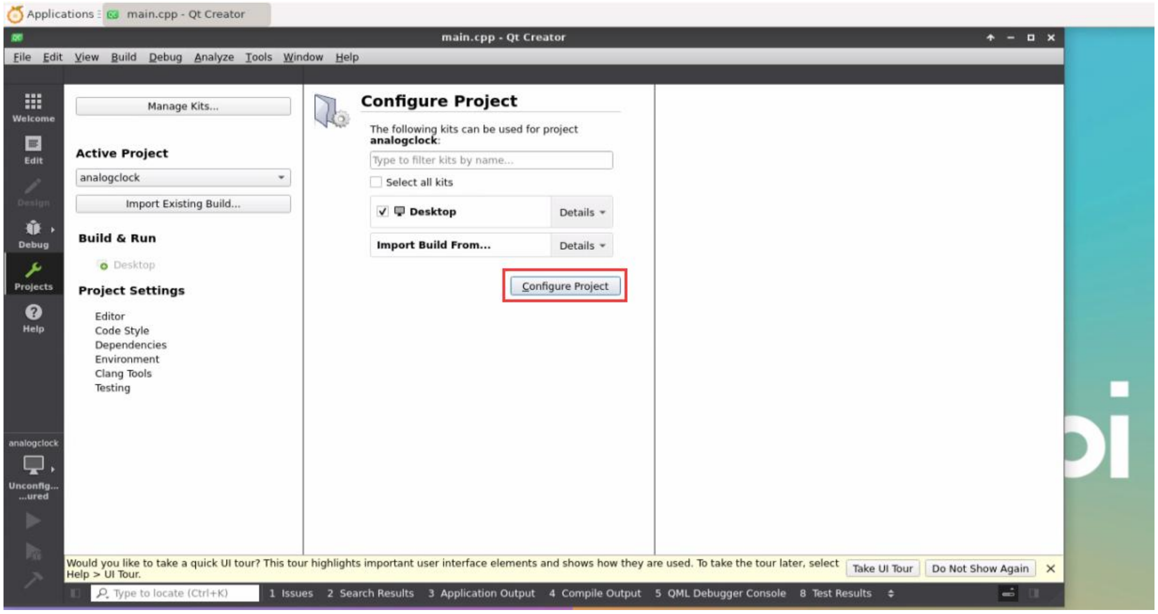
7) Then you can open a sample code.



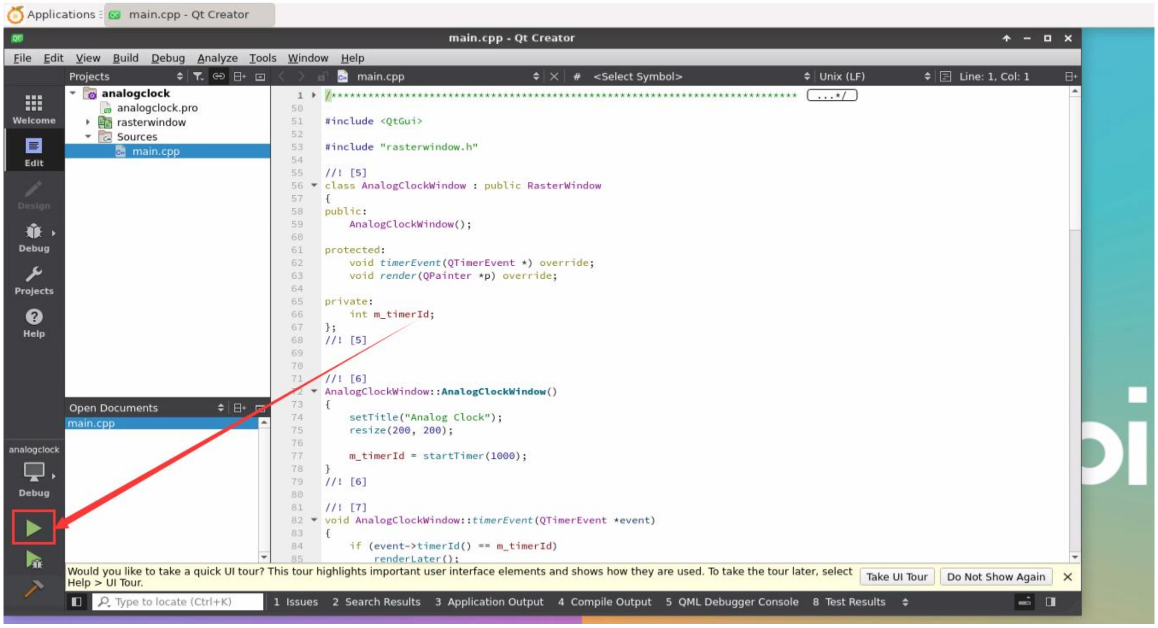
8) Clicking on the sample code will automatically open the corresponding documentation. You can read the instructions carefully.



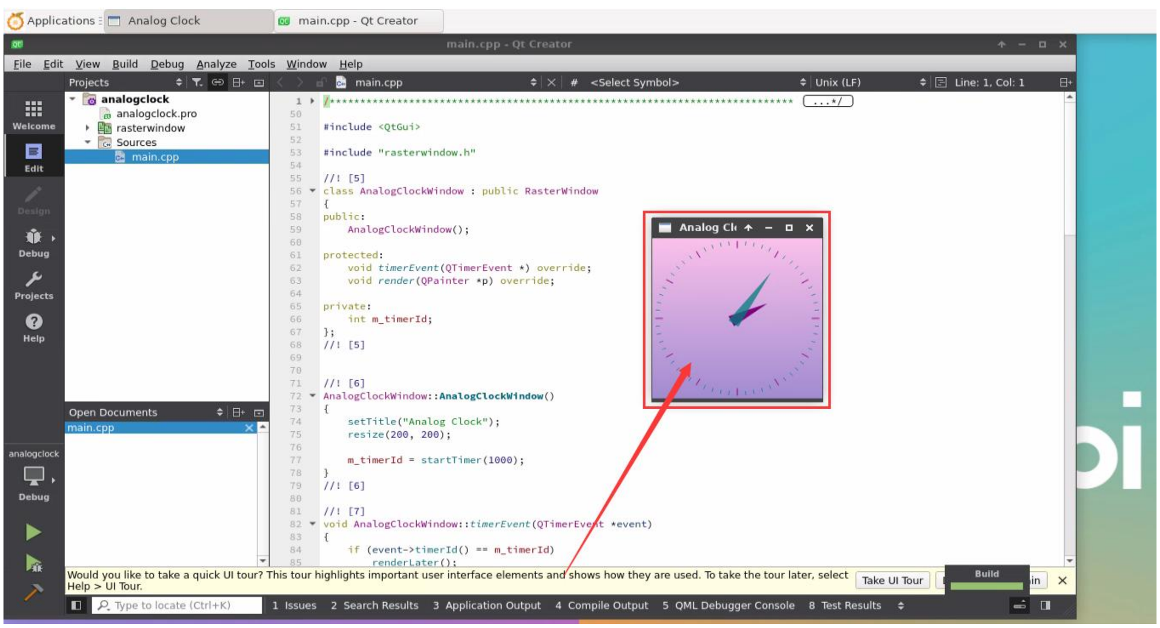
9) Then click **Configure Project**。



10) Then click the green triangle in the lower left corner to compile and run the sample code.



11) After waiting for a while, the interface shown in the figure below will pop up, which means that QT can compile and run normally.



12) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu
https://download.qt.io/archive/qtcreator
https://download.qt.io/archive/qt

3. 25. ROS installation method

3. 25. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The currently active versions of ROS 1 are as follows. The recommended version is **Noetic Ninjemys**.

Active ROS 1 distributions

Recommended

Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

http://docs.ros.org
https://wiki.ros.org/Distributions

2) The official installation document link for ROS 1 **Noetic Ninjemys** is as follows:

http://wiki.ros.org/noetic/Installation/Ubuntu

3) The official installation document of ROS **Noetic Ninjemys** recommends using Ubuntu 20.04, so make sure that the system used by the development board is the **Ubuntu 20.04 desktop system**.

http://wiki.ros.org/noetic/Installation

4) Then install ros1 using the script below.

```
orangepi@orangepicm5:~$ install_ros.sh ros1
```

5) Before using ROS tools, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code.

Note that when running the following command, you need to ensure that the development board can access GitHub normally, otherwise an error will be reported due to network problems.

The install_ros.sh script will try to modify /etc/hosts and automatically run the following command. However, this method cannot guarantee that GitHub can be accessed normally every time. If the following error is prompted after install_ros.sh installs ros1, please find other ways to enable the Linux system of the development board to access GitHub normally, and then manually run the following command.

https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
ERROR: error loading sources list:
The read operation timed out

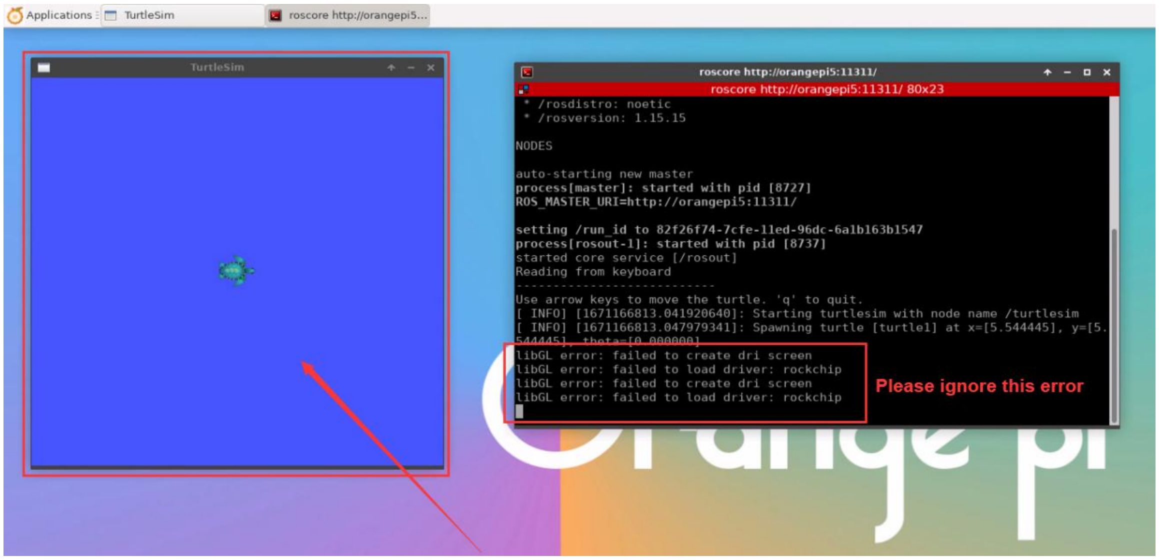
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run
```

```
        rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

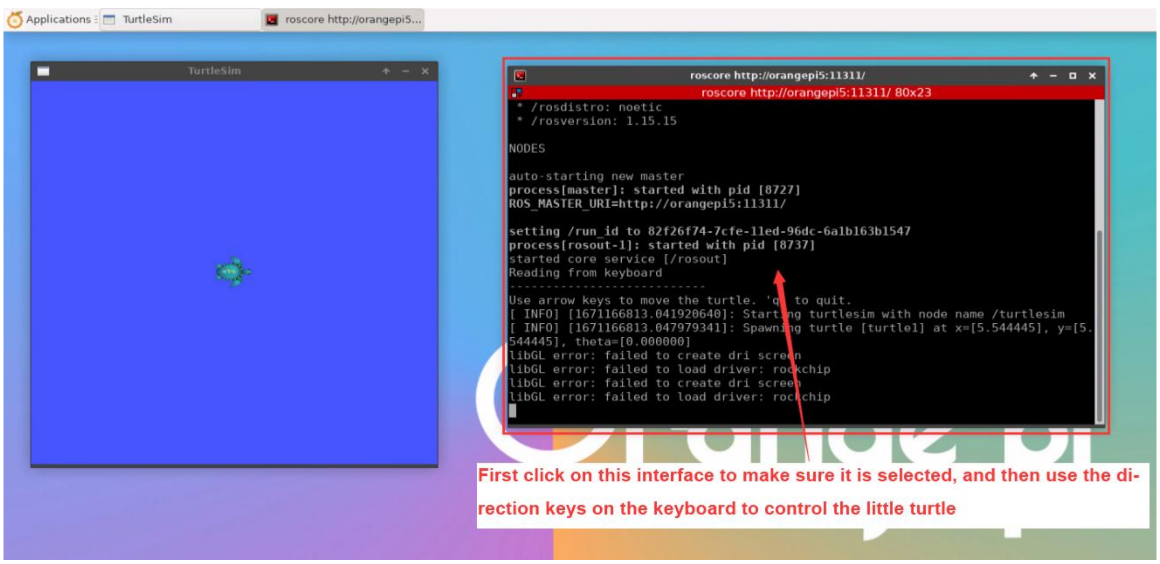
6) Then open a command line terminal window on the **desktop**, and use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally.

```
orangepi@orangepi:~$ test_ros.sh
```

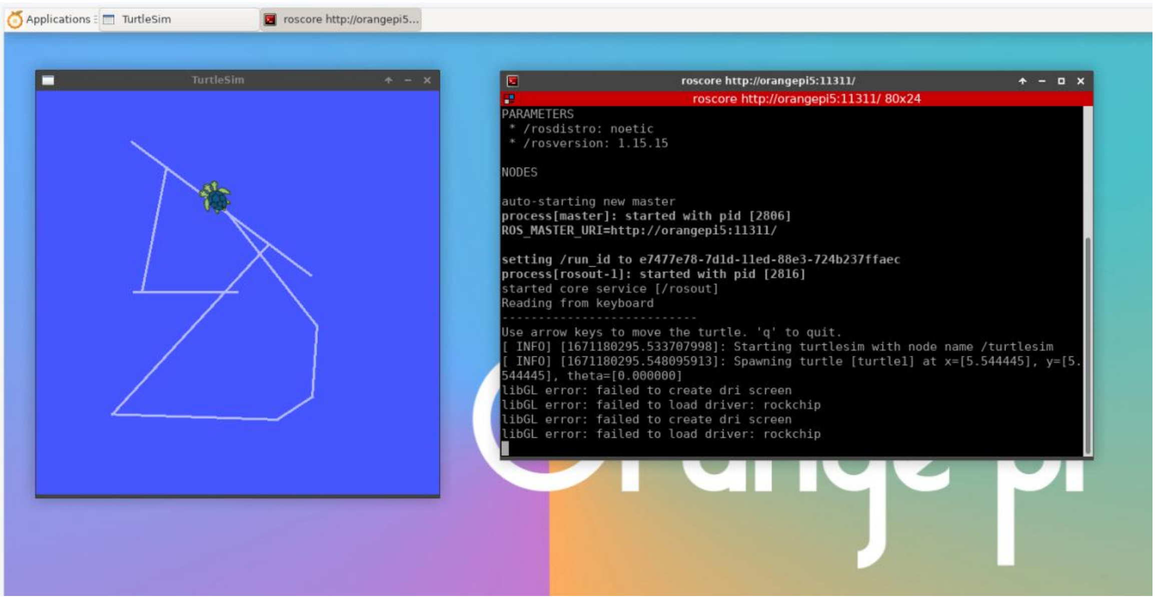
7) After running the **test_ros.sh** script, a small turtle will pop up as shown in the figure below.



8) Then please keep the terminal window you just opened on top.

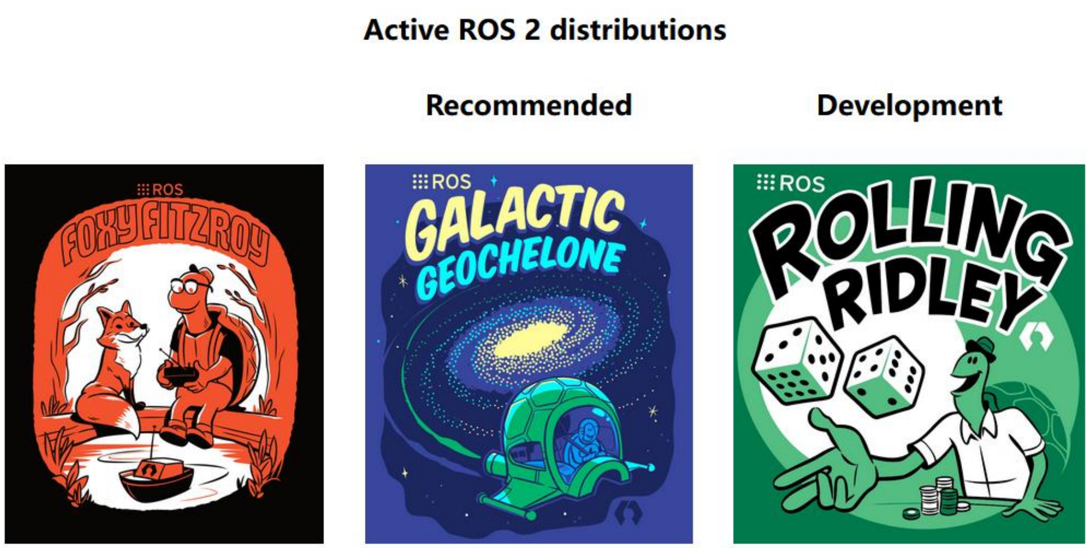


9) At this time, press the direction keys on the keyboard to control the little turtle to move up, down, left and right.



3.25.2. How to install ROS 2 Galactic on Ubuntu 20.04

1) The currently active versions of ROS 2 are as follows. The recommended version is **Galactic Geochelone**



Distro	Release date	Logo	EOL date
Humble Hawksbill	May 23rd, 2022		May 2027
Galactic Geochelone	May 23rd, 2021		November 2022
Foxy Fitzroy	June 5th, 2020		May 2023

http://docs.ros.org
http://docs.ros.org/en/galactic/Releases.html

2) The official installation document link for ROS 2 **Galactic Geochelone** is as follows:

docs.ros.org/en/galactic/Installation.html
http://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Debians.html

3) The official installation document of ROS 2 **Galactic Geochelone** recommends using Ubuntu 20.04 for Ubuntu Linux, so make sure that the system used by the development board is the **Ubuntu 20.04 desktop system**. There are several ways to install ROS 2. The following demonstrates how to install ROS 2 **Galactic Geochelone** through **Debian packages**

4) Use the **install_ros.sh** script to install ros2.

orangepi@orangepi:~$ **install_ros.sh ros2**

5) After the **install_ros.sh** script installs ros2, it will automatically run the **ros2-h** command. If you can see the following print, it means that ros2 is installed successfully.

```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
```

```
  -h, --help            show this help message and exit

Commands:
  action     Various action related sub-commands
  bag        Various rosbag related sub-commands
  component  Various component related sub-commands
  daemon     Various daemon related sub-commands
  doctor     Check ROS setup and other potential issues
  interface  Show information about ROS interfaces
  launch     Run a launch file
  lifecycle  Various lifecycle related sub-commands
  multicast  Various multicast related sub-commands
  node       Various node related sub-commands
  param      Various param related sub-commands
  pkg        Various package related sub-commands
  run        Run a package specific executable
  security   Various security related sub-commands
  service    Various service related sub-commands
  topic      Various topic related sub-commands
  wtf        Use `wtf` as alias to `doctor`

  Call `ros2 <command> -h` for more detailed usage.
```

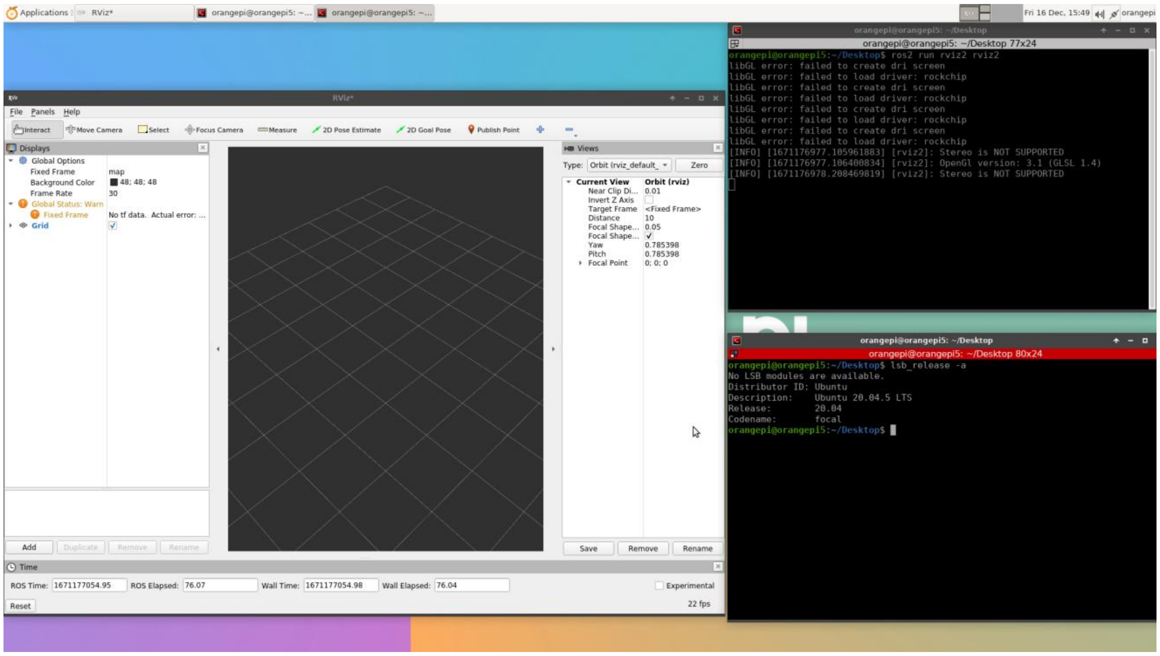
6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally.

```
orangepi@orangepicm5:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2.

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
```

```
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



8) For the usage of ROS, please refer to the ROS 2 documentation.

http://docs.ros.org/en/galactic/Tutorials.html

3. 25. 3. How to install ROS 2 Humble on Ubuntu 22.04

1) Use the **install_ros.sh** script to install ros2.

```
orangepi@orangepi:~$ install_ros.sh ros2
```

2) After the **install_ros.sh** script installs ros2, it will automatically run the **ros2-h** command. If you can see the following print, it means that ros2 is installed successfully.

```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action     Various action related sub-commands
  bag        Various rosbag related sub-commands
  component  Various component related sub-commands
```

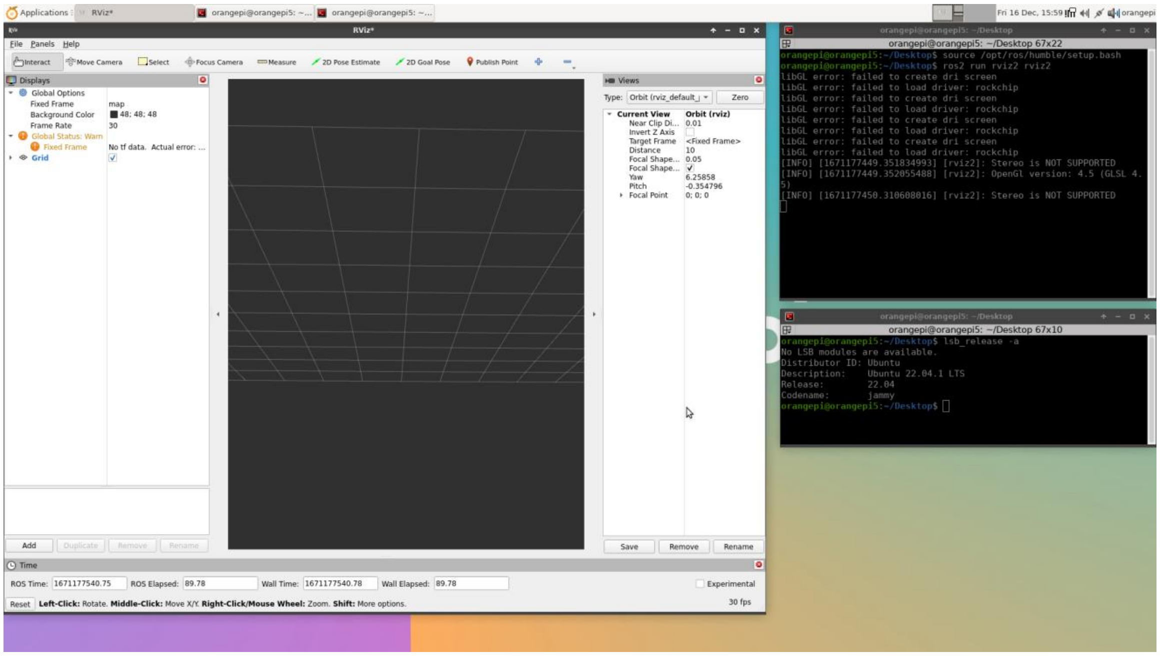
```
daemon        Various daemon related sub-commands
doctor        Check ROS setup and other potential issues
interface     Show information about ROS interfaces
launch        Run a launch file
lifecycle     Various lifecycle related sub-commands
multicast     Various multicast related sub-commands
node          Various node related sub-commands
param         Various param related sub-commands
pkg           Various package related sub-commands
run           Run a package specific executable
security      Various security related sub-commands
service       Various service related sub-commands
topic         Various topic related sub-commands
wtf           Use `wtf` as alias to `doctor`

Call `ros2 <command> -h` for more detailed usage.
```

3) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally.

```
orangepi@orangepicm5:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2.

```
orangepi@orangepi:~$ source /opt/ros/humble/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



5) Reference documents.

http://docs.ros.org/en/humble/index.html
http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

3. 26. How to install kernel header files

1) The Linux image released by OPi comes with the deb package of the kernel header file by default, which is stored in **/opt/**.

```
orangepi@orangepi:~$ ls /opt/linux-headers*
/opt/linux-headers-legacy-rockchip-rk3588_x.x.x_arm64.deb
```

2) Use the following command to install the kernel header file deb package.

The name of the kernel header file deb package needs to be replaced with the actual name, please do not copy it.

```
orangepi@orangepi:~$ sudo dpkg -i /opt/linux-headers-legacy-rockchip-rk3588_1.x.x_arm64.deb
```

3) After installation, you can see the folder where the kernel header files are located under **/usr/src**.

```
orangepi@orangepi:~$ ls /usr/src
linux-headers-5.10.160-rockchip-rk3588
```

4) Then you can write a hello kernel module to test the kernel header file.

a. First, write the code for the hello kernel module as follows:

```
orangepi@orangepi:~$ vim hello.c
#include <linux/init.h>
#include <linux/module.h>

static int hello_init(void)
{
        printk("Hello Orange Pi -- init\n");

        return 0;
}
static void hello_exit(void)
{
        printk("Hello Orange Pi -- exit\n");

        return;
}

module_init(hello_init);
module_exit(hello_exit);

MODULE_LICENSE("GPL");
```

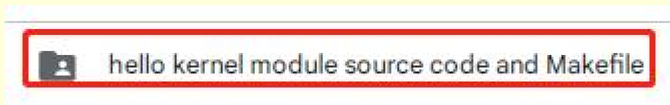
b. Then write the Makefile file to compile the hello kernel module as follows:

```
orangepi@orangepi:~$ vim Makefile
ifneq ($(KERNELRELEASE),)
obj-m:=hello.o
else
KDIR :=/lib/modules/$(shell uname -r)/build
PWD  :=$(shell pwd)
all:
    make -C $(KDIR) M=$(PWD) modules
clean:
    rm -f *.ko *.o *.mod.o *.mod *.symvers *.cmd  *.mod.c *.order
```

```
endif
```

c. Then use the make command to compile the hello kernel module. The output of the compilation process is as follows:

If there is a problem compiling the code you copied here, please download the source code from the official tool and upload it to the Linux system of the development board for testing.




```
orangepi@orangepi:~$ make
make -C /lib/modules/5.10.160-rockchip-rk3588/build M=/home/orangepi modules
make[1]: Entering directory '/usr/src/linux-headers-5.10.160-rockchip-rk3588'
  CC [M]  /home/orangepi/hello.o
  MODPOST /home/orangepi/Module.symvers
  CC [M]  /home/orangepi/hello.mod.o
  LD [M]  /home/orangepi/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.10.160-rockchip-rk3588'
```

d. After compilation, the **hello.ko** kernel module will be generated.

```
orangepi@orangepi:~$ ls *.ko
hello.ko
```

e. Use the **insmod** command to insert the **hello.ko** kernel module into the kernel.

```
orangepi@orangepi:~$ sudo insmod hello.ko
```

f. Then use the **demsg** command to view the output of the **hello.ko** kernel module. If you can see the following output, it means that the **hello.ko** kernel module is loaded correctly.

```
orangepi@orangepi:~$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
```

g. Use the **rmmod** command to uninstall the **hello.ko** kernel module.

```
orangepi@orangepi:~$ sudo rmmod hello
orangepi@orangepi:~$ dmesg | grep "Hello"
[ 2871.893988] Hello Orange Pi -- init
[ 3173.800892] Hello Orange Pi -- exit
```

3. 27. Instructions for using the power on/off logo

1) The power on/off logo is displayed only in the desktop version of the system by default.

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the power on/off logo.

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=false
```

3) Set the **bootlogo** variable to **true** in **/boot/orangepiEnv.txt** to enable the power on/off logo.

```
orangepi@orangepi:~$ vim /boot/orangepiEnv.txt
verbosity=1
bootlogo=true
```

4) The location of the boot logo image in the Linux system is.

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

5) After replacing the boot logo image, you need to run the following command to make it take effect.

```
orangepi@orangepi:~$ sudo update-initramfs -u
```

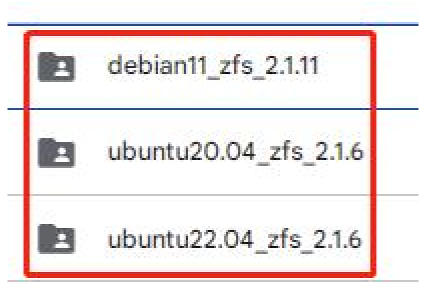
3. 28. How to use ZFS file system

3. 28. 1. How to install ZFS

Before installing zfs, you need to install the kernel header file first. For the method of installing the kernel header file, please refer to the instructions in the section "**How to install the kernel header file**".

In Ubuntu 20.04, Ubuntu 22.04 and Debian 11 systems, zfs cannot be installed directly through apt. This is because the zfs version in the default apt source is lower than 2.1.6, which is incompatible with the rk linux 5.10 kernel. This problem has been fixed in zfs 2.1.6 and later versions.

To solve this problem, we provide a deb package of zfs that can be installed normally, which can be downloaded from the **official tool** of the development board. Open the **official tool**, and then enter **the deb package folder related to zfs used by Ubuntu and Debian systems**. You can see three types of deb packages: Ubuntu20.04, Ubuntu22.04, and Debian11. Please download the required version.



After downloading the corresponding version of the zfs deb package, please upload them to the Linux system of the development board. For the upload method, please refer to the instructions in the section "**How to upload files to the Linux system of the development board**".

After uploading, use the **cd** command in the command line of the Linux system of the development board to enter the directory of the deb package, and then use the following command to install the zfs deb package.

```
orangepi@orangepi:~$ sudo apt install ./*.deb
```

After the installation is complete, you can see the zfs-related kernel modules using the following command:

```
orangepi@orangepi:~$ ls /lib/modules/5.10.160-rockchip-rk3588/updates/dkms/
icp.ko   spl.ko   zavl.ko   zcommon.ko   zfs.ko   zlua.ko   znvpair.ko   zunicode.ko
zzstd.ko
```

Then restart the Linux system and you will see that the zfs kernel module will be automatically loaded:

```
orangepi@orangepi:~$ lsmod | grep "zfs"
zfs                    2801664    0
zunicode                327680    1 zfs
zzstd                   471040    1 zfs
zlua                    139264    1 zfs
```

```
zcommon                  69632  1 zfs
znvpair                  61440  2 zfs,zcommon
zavl                     16384  1 zfs
icp                     221184  1 zfs
spl                      77824  6 zfs,icp,zzstd,znvpair,zcommon,zavl
```

In Debian 12, the default version of zfs is 2.1.11, so we can install zfs directly through the following command. Once again, please make sure that the system has installed the deb package of the kernel header file before installation.

```
orangepi@orangepi:~$ sudo apt install -y zfsutils-linux zfs-dkms
```

3. 29. How to install and use CasaOS

CasaOS is an open source home cloud system based on the Docker ecosystem, which allows you to run a variety of home applications on your own development board, such as NAS, home automation, media server, etc.

There are many problems with installing CasaOS on Debian 12, so please do not use this method to install it.

3. 29. 1. How to install CasaOS

1) First, you need to install docker. Docker is already pre-installed in the system released by Orangepi Pi. This step can be skipped. You can use the following command to view the installed docker version.

```
orangepi@orangepi:~$ docker --version
Docker version 24.0.2, build cb74dfc          # Output from Ubuntu Jammy system
```

2) Then enter the following command in the Linux system to start the installation of CasaOS.

```
orangepi@orangepi:~$ curl -fsSL https://get.casaos.io | sudo bash
```

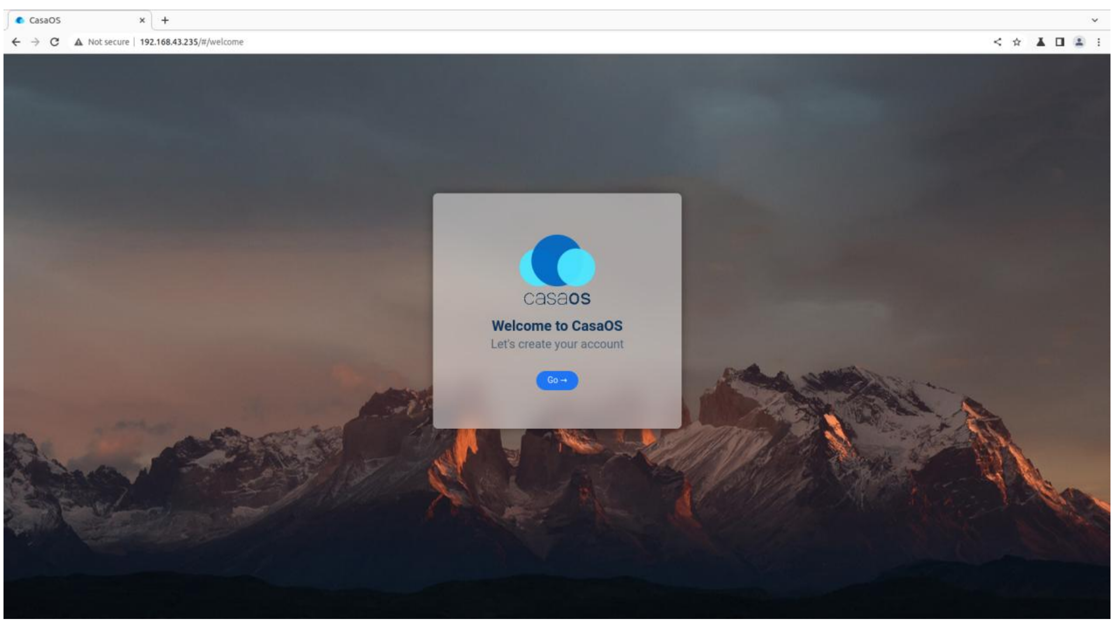
3) When you see the following print information output in the terminal, it means that CasaOS has been installed.

```
CasaOS v0.4.4.2 is running at:

Open your browser and visit the above address.


CasaOS Project  : https://github.com/IceWhaleTech/CasaOS
CasaOS Team     : https://github.com/IceWhaleTech/CasaOS#maintainers
CasaOS Discord  : https://discord.gg/knqAbbBbeX
Website         : https://www.casaos.io
Online Demo     : http://demo.casaos.io

Uninstall       : casaos-uninstall
```

3. 29. 2. How to use CasaOS

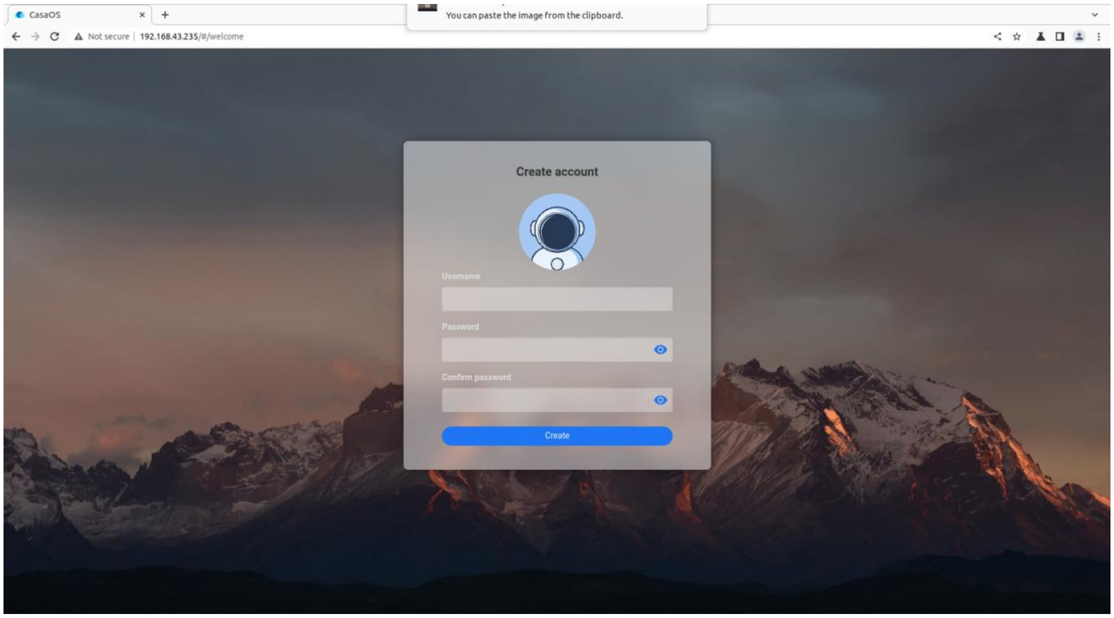
1) After installing CasaOS, enter **http://development board's IP address** in the browser to open CasaOS.

2) After opening CasaO, the welcome interface below will pop up. Click "Go" to proceed to the next step.

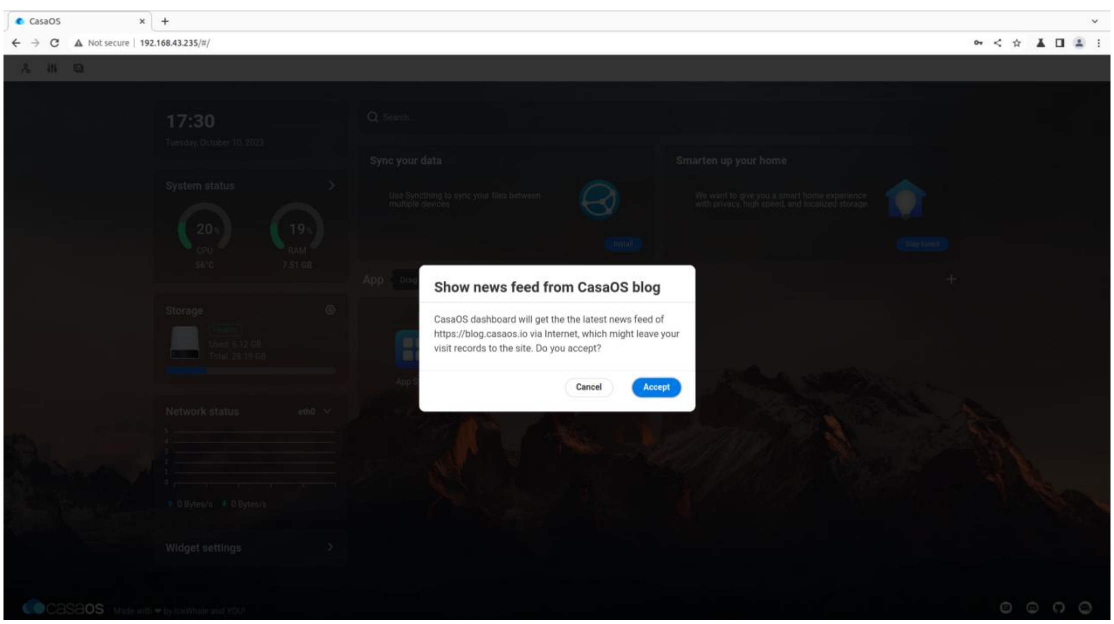


3) When you log in to CasaOS for the first time, the login interface is the interface for setting the account and password. When you log in again in the future, only the interface for entering the account and password will appear. After setting the account and password,

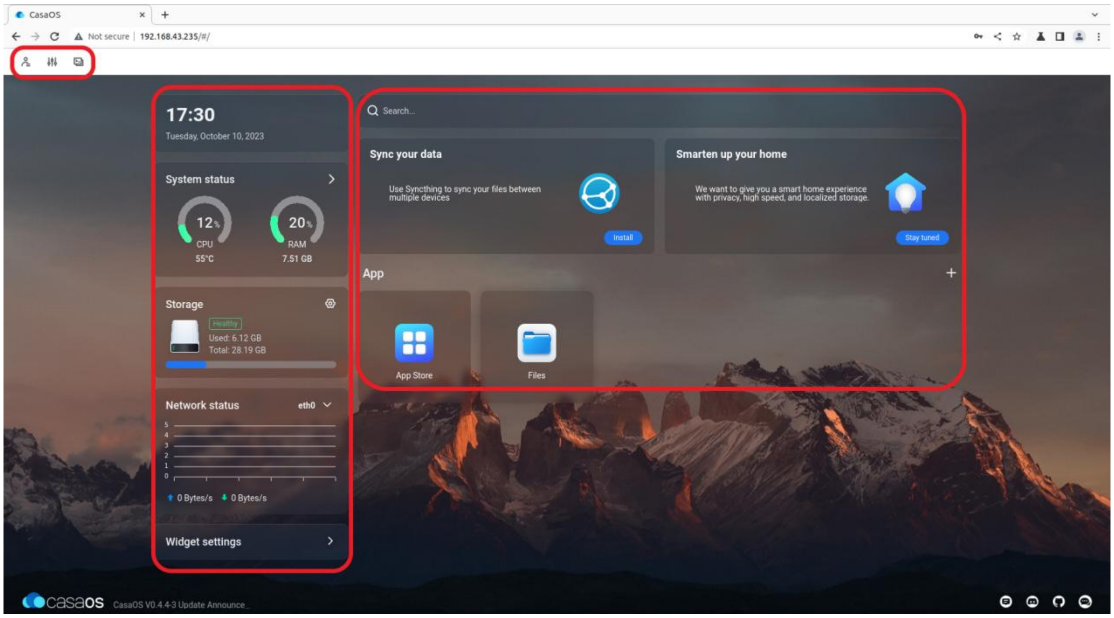
click "Create" to proceed to the next step.



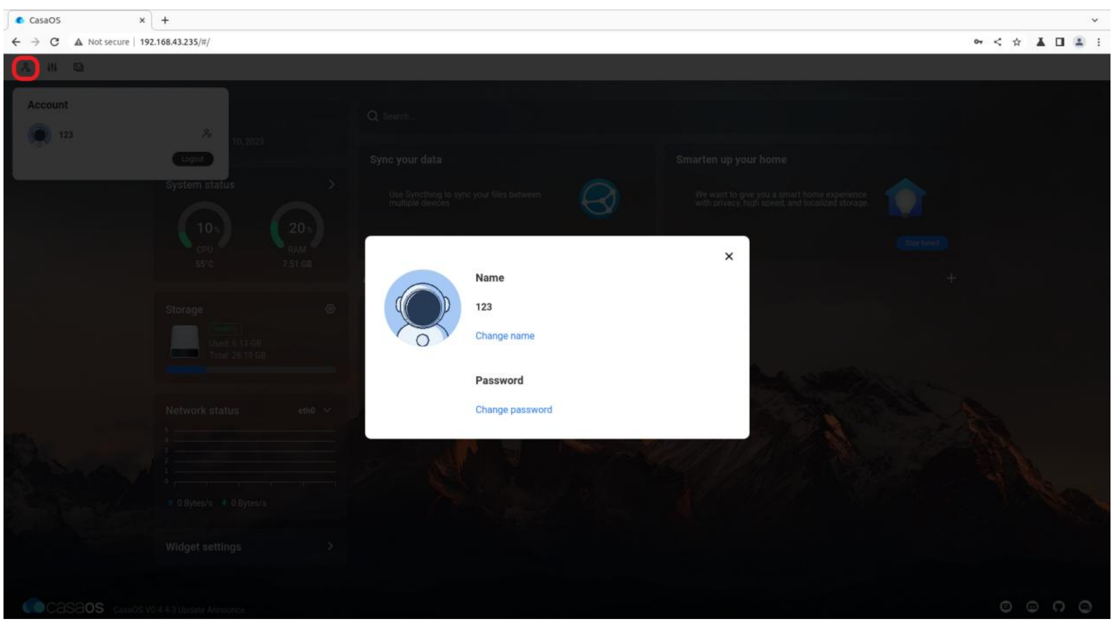
4) In the following interface, just click “Accept” to proceed to the next step.



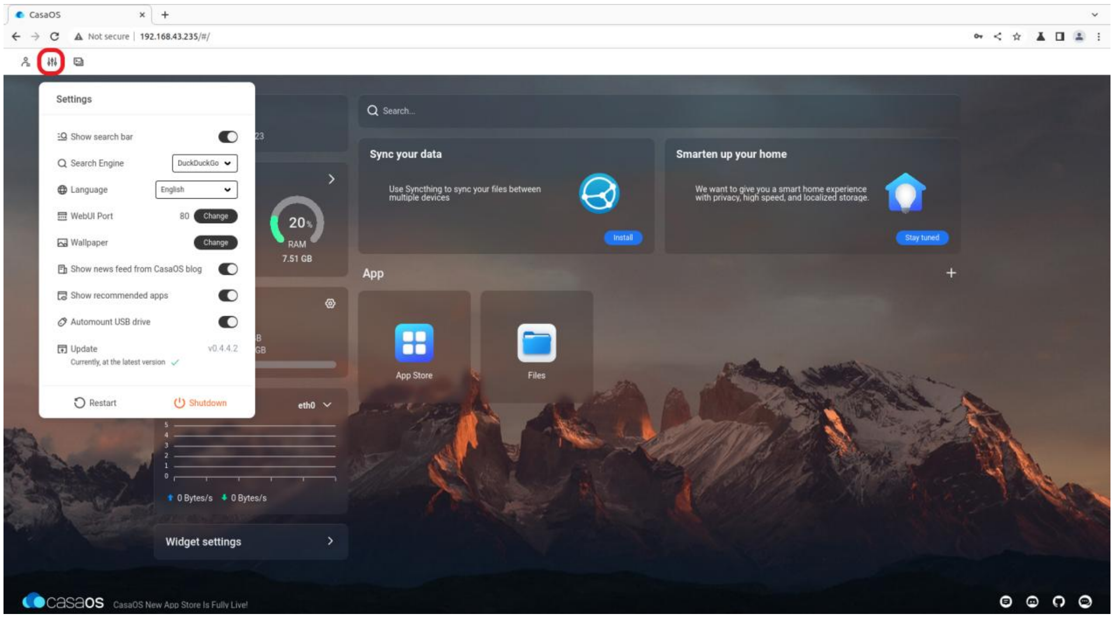
5) Now you have entered the main page of CasaOS. There are three icons in the upper left corner for function settings. On the left is the performance panel, which can display the current time and the status information of CPU, RAM, storage, and network. On the right is the function panel with functions such as search, application recommendation, application store, and file management.



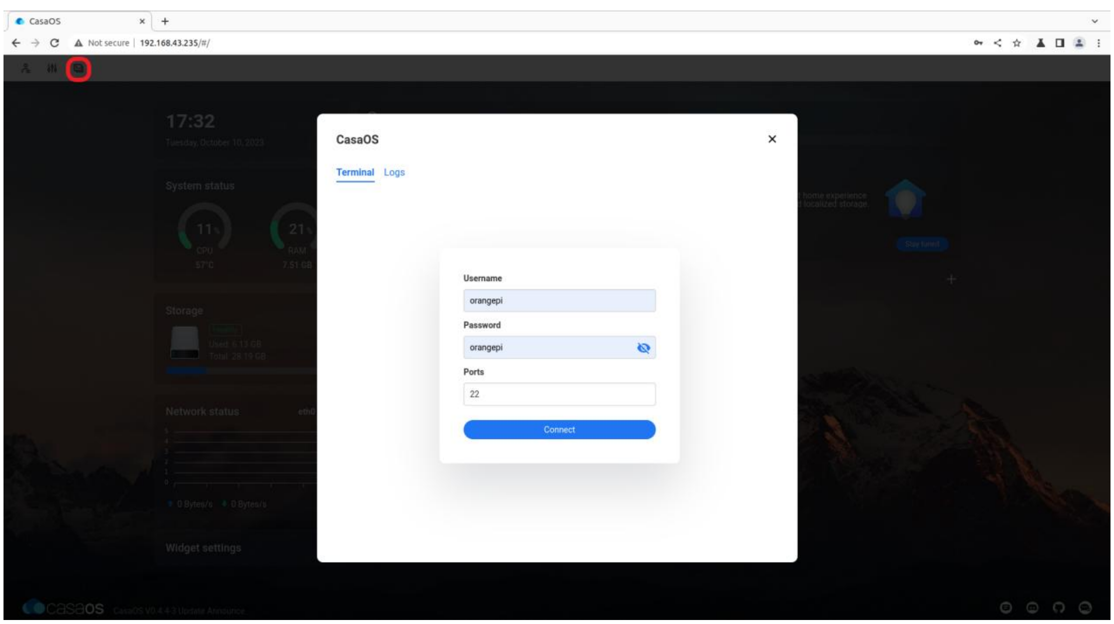
6) You can click the first icon in the upper left corner to modify your account and password.



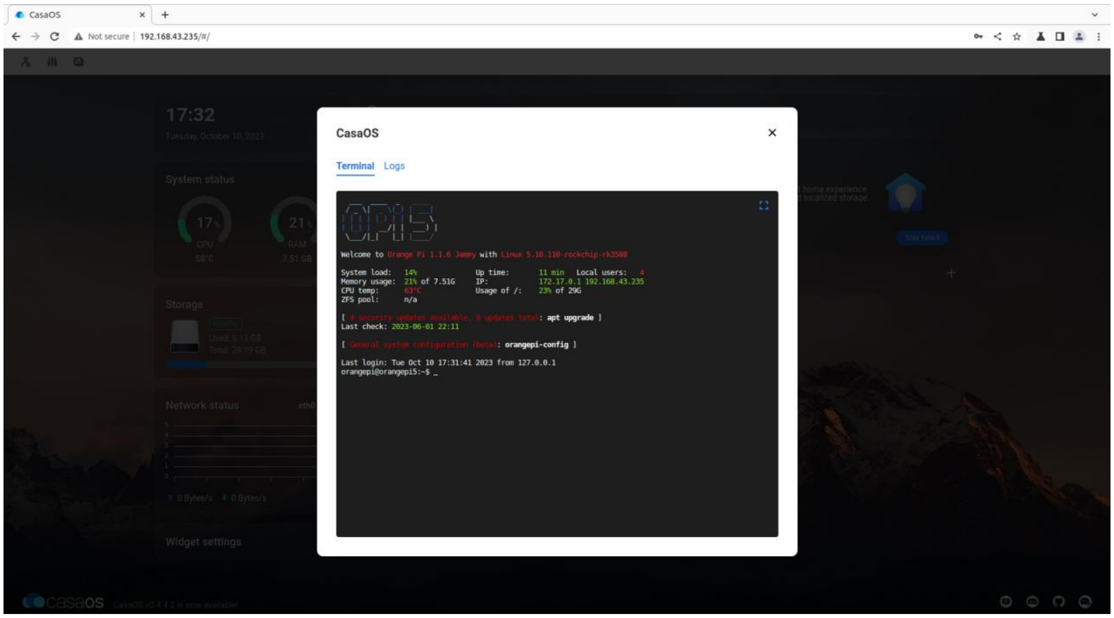
7) You can click the second icon to set basic functions.



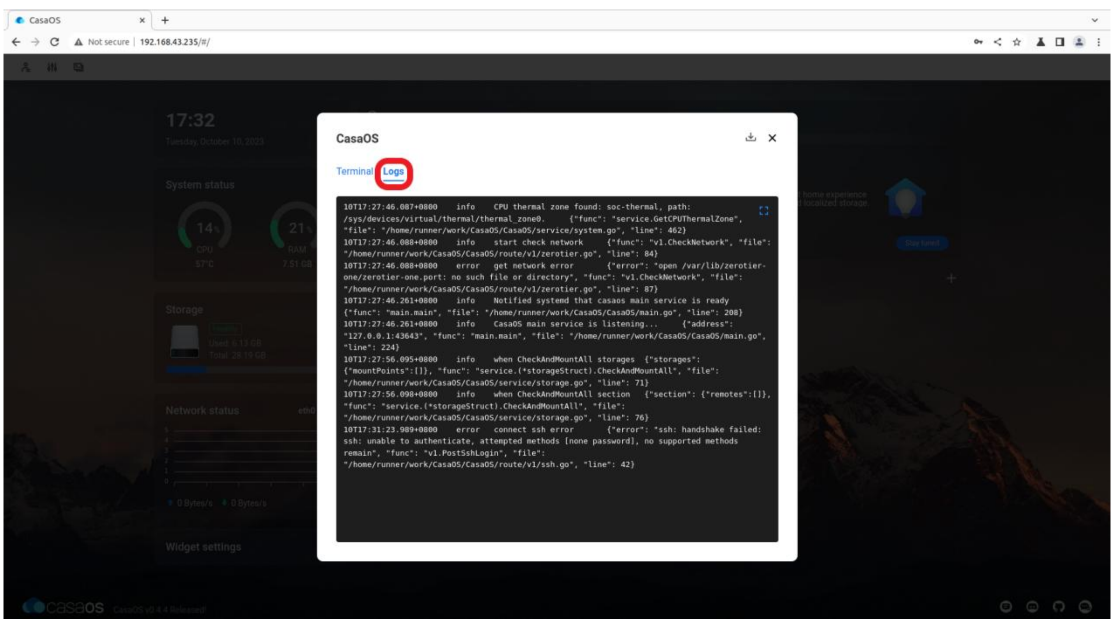
8) The third icon in the upper left corner has two main functions, namely switching to command line mode and printing log information. When switching to command line mode, you need to enter the account and password. The account and password here refer to the account and password of the development board Linux system. The port system defaults to 22.



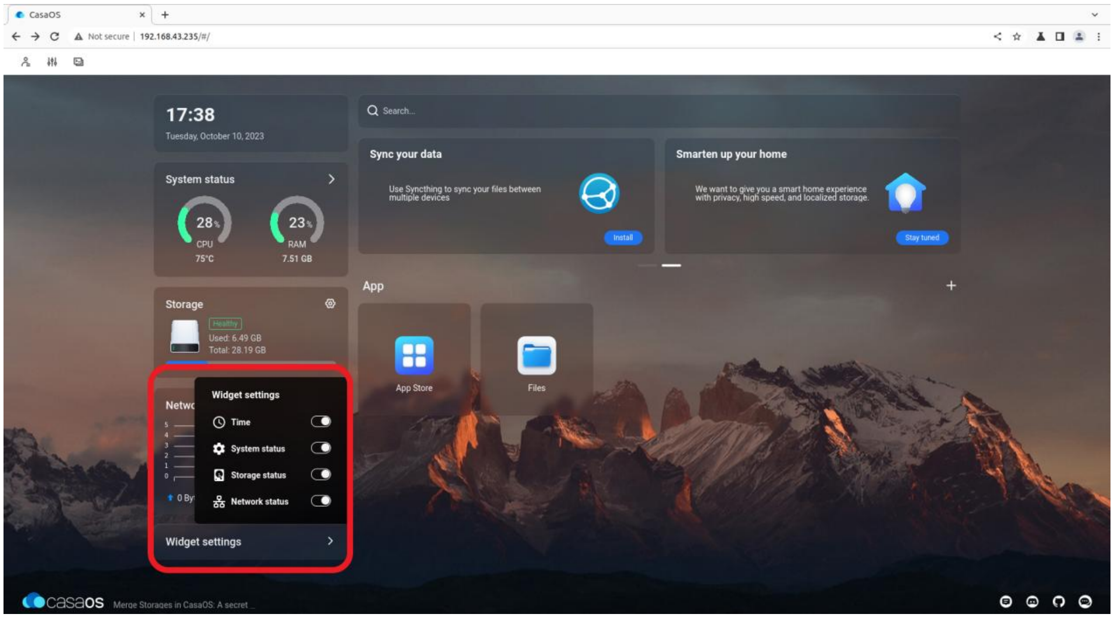
9) Then click "Connect" to enter the command line interface:



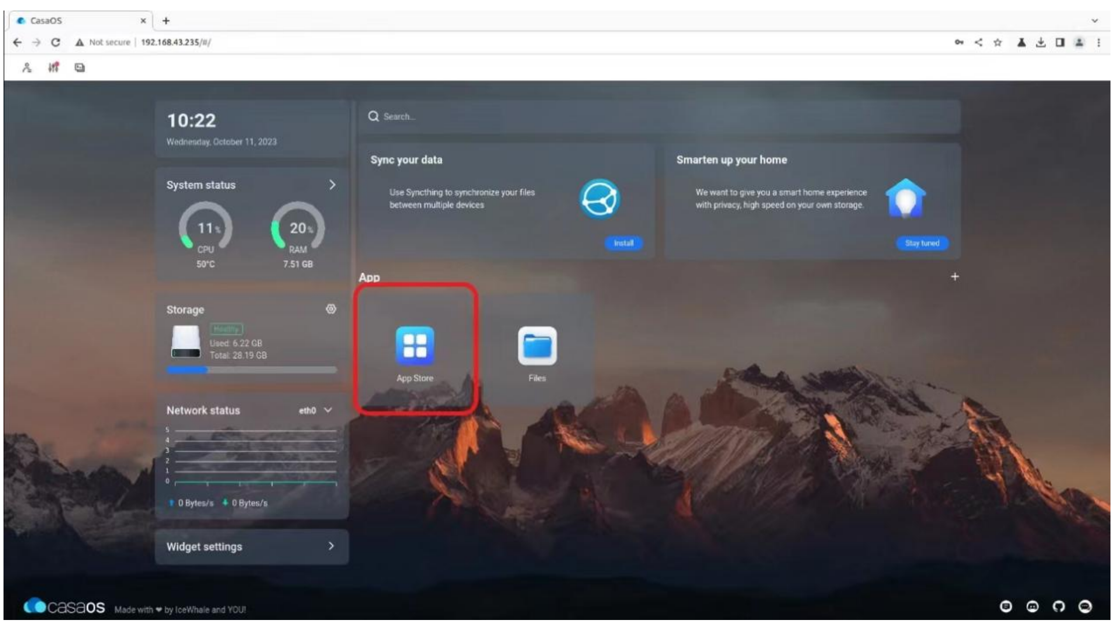
10) Another function under the third icon is to print the CasaOS log. Click "Logs" to enter. The interface is as follows:



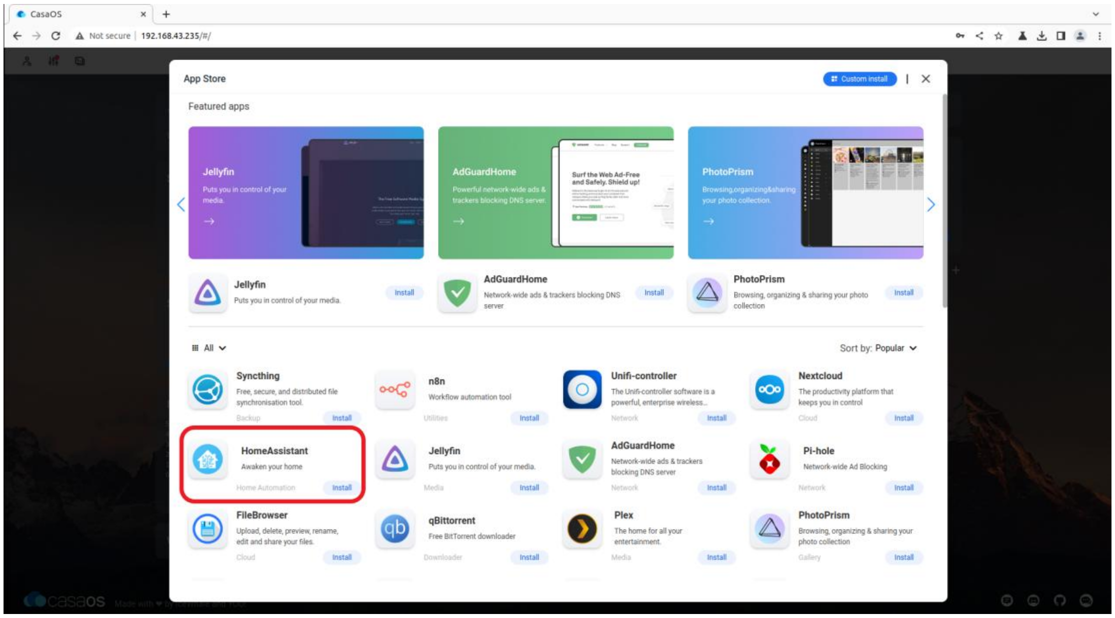
11) Click "Widget settings" in the lower left corner to set whether to display the performance panel widget on the main page.



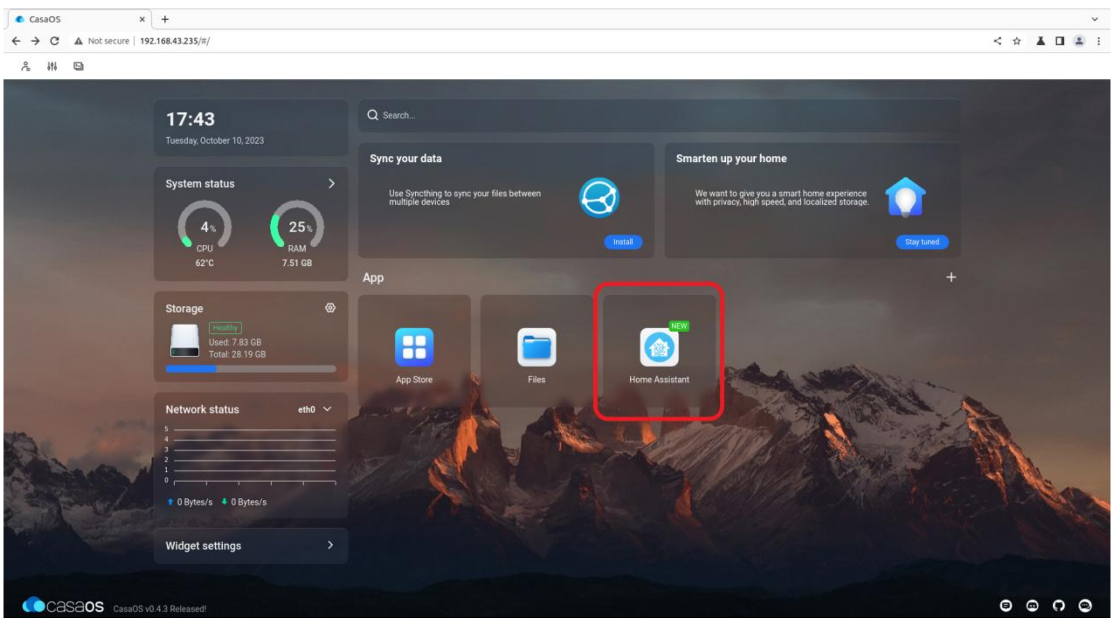
12) Click "APP Store" on the main interface to open the App Store. Currently, there are more than 70 APPs available in the App Store.



13) Here we take Home Assistant as an example for downloading. Find Home Assistant in the APP Store and click the corresponding "install".

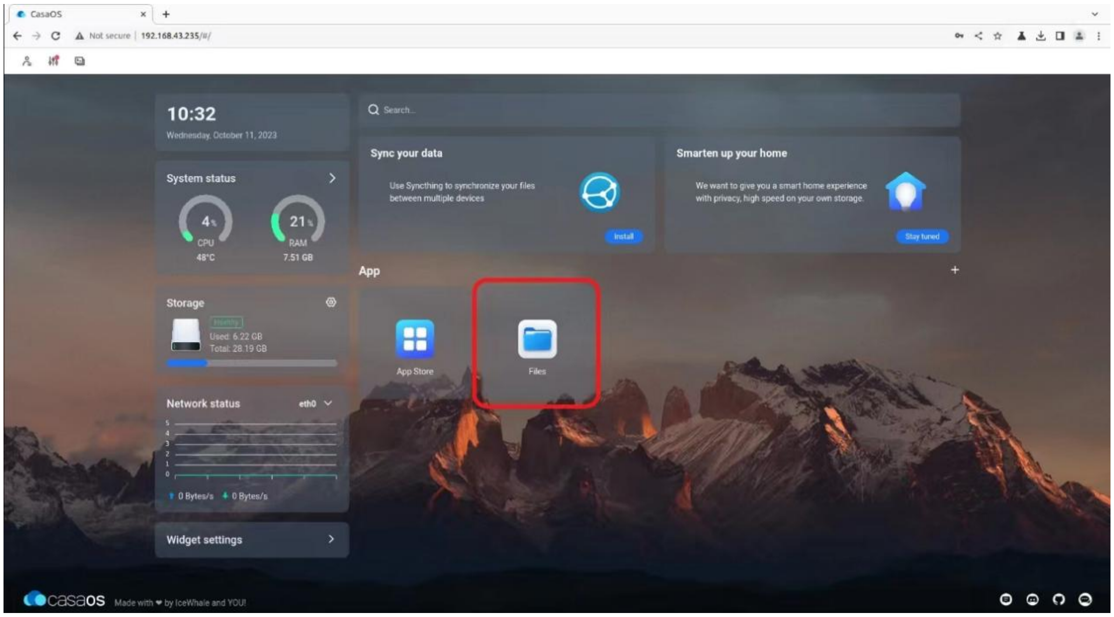


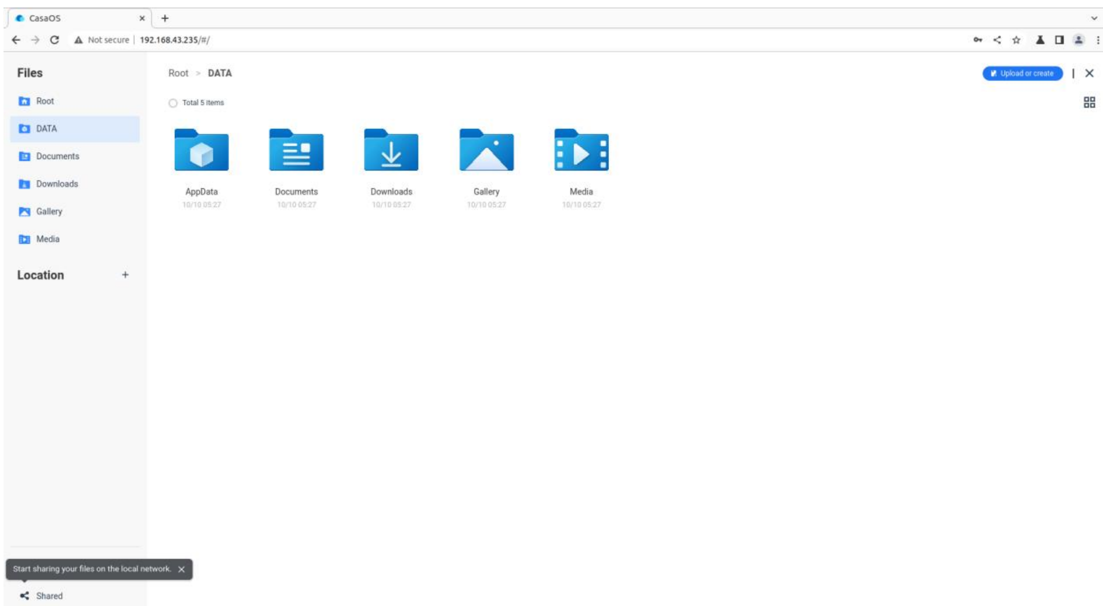
14) After the download is complete, HostAssitant will appear on the main page.



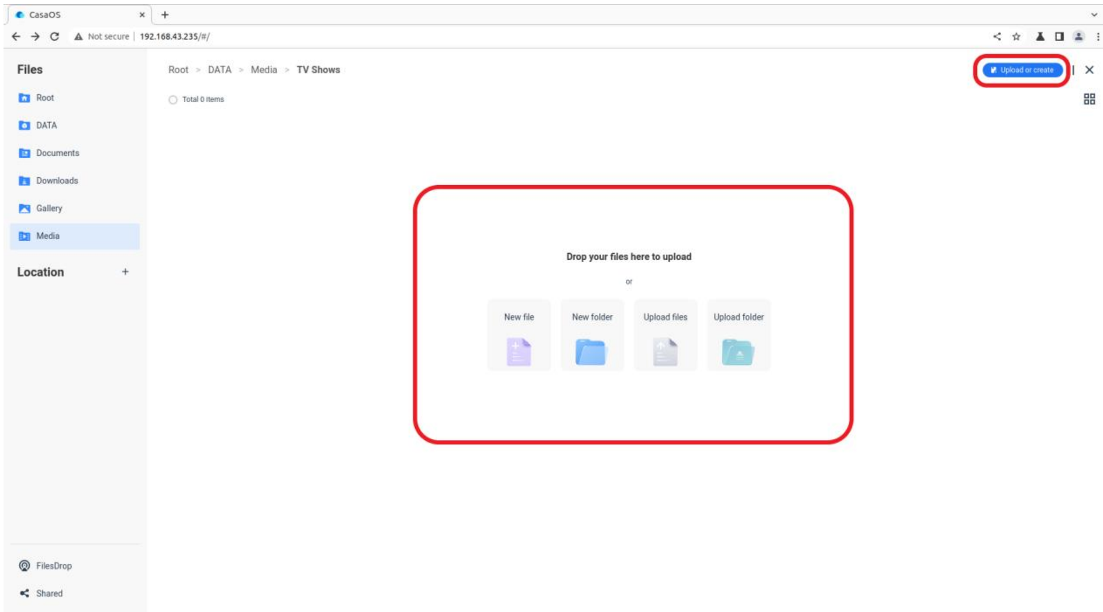
15) Click "Files" in the main interface to open the file system that comes with CasaOS, and then you can upload and save files.

Please make sure other devices and the development board are in the same LAN.





16) When uploading files, you need to switch to the target folder, then drag the local file to the indicated area in the figure, or click "Upload or Create" in the upper right corner to select the file to upload.



17) If you want to uninstall CasaOS, you can use the following command:

```
orangepi@orangepicm5:~$ casaos-uninstall
```

3. 30. Methods of using NPU

3. 30. 1. Prepare tools

1) A PC with Ubuntu 20.04 operating system installed

According to the official documentation of RKNN-Toolkit2, the current version of RKNN-Toolkit2 supports the following operating systems:

a. Ubuntu18.04 (x64)

b. Ubuntu20.04 (x64)

c. Ubuntu22.04 (x64)

In this document, we demonstrate using the Ubuntu20.04 (x64) operating system. Please test other versions of the operating system yourself.

2) An Orange Pi 5 Plus development board with Debian 11 system installed

3) A Type-C interface data cable for using adb function

3. 30. 2. Installing RKNN-Toolkit2 on Ubuntu PC

Toolkit2 is a development kit used on the Ubuntu PC platform. Users can easily complete functions such as model conversion, inference, and performance evaluation using the Python interface provided by the tool.

1) On the Ubuntu PC side, open a command-line window and enter the following command to install python3 and pip3

```
test@test:~$ sudo apt-get install python3 python3-dev python3-pip
```

2) You can use the following command to view the installed version of python3

```
test@test:~$ python3 --version
Python 3.8.10
```

3) Then enter the following command to install the dependency package of RKNN-Toolkit2

```
test@test:~$  sudo  apt-get  update
test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 \
 libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

4) Then enter the following command to download RKNN-Toolkit2 version 1.5.2

```
test@test:~$ git clone git clone https://github.com/airockchip/rknn-toolkit2 -b v1.5.2
```

5) Then enter the following command to install the dependency package for the corresponding version of python3. This command will use the dependencies listed in the pip3 installation file requirements_cp38-1.5.2.txt. If the dependency installation is incomplete, do not specify the installation source and install each package separately.

```
test@test:~$  pip3  install  -r  rknn-toolkit2/doc/requirements_cp38-1.5.2.txt  -i  \
```

```
https://mirror.baidu.com/pypi/simple
```

6) Then enter the following command to use pip3 to install the RKNN-Toolkit2 package. After installation, you can use RKNN-Toolkit2

```
test@test:~$ pip3 install rknn-toolkit2/packages/rknn_toolkit2-1.5.2+b642f30c-cp38-cp38-linux_x86_64.whl
```

3. 30. 3. Use RKNN-Toolkit2 for model transformation and model inference

RKNN-Toolkit2 supports converting models such as Caffe, TensorFlow, TensorFlow Lite, ONNX, DarkNet, PyTorch, etc. into RKNN models, and then running the RKNN models through simulation or using the NPU of the development board on Ubuntu PC for inference.

The example folder in RKNN-Toolkit2 provides relevant examples to help users better understand how to operate. We take the ONNX model with yolov5 functionality as an example for illustration.

3. 30. 3. 1. Simulating and Running Models on Ubuntu PC

RKNN-Toolkit2 is equipped with a built-in simulator that allows users to simulate the inference process of models on Rockchip NPU on Ubuntu PC.

This way, both model conversion and inference can be completed on Ubuntu PC, helping users test and validate their models faster.

1) First, switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

2) Then run the test.py script, which first converts the yolov5s_relu.onnx model into an RKNN model that can be run on the simulator, and then uses the simulator to simulate and run the model to infer the bus.jpg image in the current directory

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

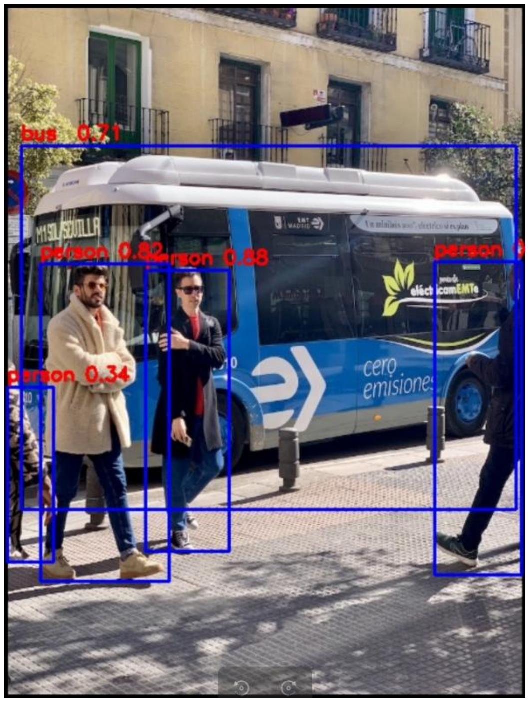
3) After the test. py script runs successfully, the following print message will be seen, indicating that the model has successfully detected four people and a bus in the bus.jpg image

```
done
--> Running model
```

```
W inference: The 'data_format' has not been set and defaults is nhwc!
done
class: person, score: 0.884139358997345
box coordinate left,top,right,down: [209.1040009856224, 244.4304337501526, 286.5742521882057,
506.7466902732849]
class: person, score: 0.8676778078079224
box coordinate left,top,right,down: [478.5757632255554, 238.58572268486023, 559.5273861885071,
526.479279756546]
class: person, score: 0.8246847987174988
box coordinate left,top,right,down: [110.57257843017578, 238.58099019527435,
230.54625701904297, 534.0008579492569]
class: person, score: 0.3392542004585266
box coordinate left,top,right,down: [79.96397459506989, 354.9062474966049, 122.13020265102386,
516.2529321908951]
class: bus , score: 0.7012234926223755
box coordinate left,top,right,down: [94.43931484222412, 129.53470361232758, 553.1492471694946,
468.0852304697037]
D NPUTransfer: Transfer client closed, fd = 3
```

4) The converted model file yolov5s_relu.rknn and the inferred image result result.jpg are saved in the current directory

5) The result.jpg image shows the object categories and confidence rates detected in the bus.jpg image using the yolov5s_relu.rknn model



3. 30. 3. 2. NPU running model using development board on Ubuntu PC

RKNN-Toolkit2 provides users with a Python interface for inference using the NPU of the development board through adb, allowing users to run mode ls for inference on Ubuntu PC using the NPU of the development board.

In this way, Ubuntu PC can optimize and adjust the model based on its actual performance when running on the NPU of the development board, usin g the machine learning library provided by Python.

3. 30. 3. 2. 1. Connect adb using Type-C data cable

Operate the development board through adb on Ubuntu PC, please refer to the section on **ADB usage for instructions** on how to use adb

3.30.3.2.2. Update the rknn_server and librknnrt.so of the development board

librknnrt.so is a runtime library on the board.

rknn_server is a backend proxy service running on the development board, used to receive protocols transmitted from the PC via USB, execute the corresponding interface in the runtime library on the board, and return the results to the PC.

1) First, enter the following command on Ubuntu PC to download RKNPU2 version 1.5.2

```
test@test:~$ git clone https://github.com/rockchip-linux/rknpu2 -b v1.5.2
```

2) Then, on the Ubuntu PC side, enter the following command to update the rknn_server of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/rknn_server/aarch64/usr/bin/* /usr/bin
```

3) Then enter the following command on Ubuntu PC to update the librknnrt.so library of the development board through adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/librknn_api/aarch64/librknnrt.so /usr/lib
```

4) Open the terminal of the development board through the adb tool

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

5) Open the rknn_server service on the development board

```
root@orangepi:/# sudo restart_rknn.sh
root@orangepi:/# start rknn server,version:1.5.2(8babfeabuild@2023-08-25T10:30:31)
I NPUTransfer: Starting NPU TransferServer,Transfer version 2.1.0(b5861e7@2020-11-23T11:50:51)
```

6) You can use the following command to check. If the process ID of rknn_server appears, it means that rknn_server is already open, and the running environment of the development board is set up

```
root@orangepi:/# pgrep rknn_server
```

```
3131
```

3. 30. 3. 2. 3. Modify parameters in the example

1) On the Ubuntu PC side, you can view the device ID of the development board connected to the Ubuntu PC using the following command. This ID will be used below

```
test@test:~$ adb devices
List of devices attached
4f9f859e5a120324    device
```

2) Switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

3) Use Vim editor to modify the test.py file

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ vim test.py
```

4) In the test.py file, we need to make modifications to the following content:

a. In the preprocessing configuration, modify the target platform to rk3588, so that the resulting model is an RKNN model suitable for NPU on the RK3588 development board

```
# pre-process config
print('--> Config model')
rknn.config(mean_values=[[0, 0, 0]], std_values=[[255, 255, 255]], target_platform='rk3588')
print('done')
```

b. In the initialization running environment, add explanations for the target platform and device ID. The target platform is rk3588, and the device ID is the device ID obtained from the development board through adb. The inference operation of the running model will be performed on the NPU of the RK3588 development board

```
# Init runtime environment
print('--> Init runtime environment')
ret = rknn.init_runtime(target='rk3588',device_id='4f9f859e5a120324')
if ret != 0:
    print('Init runtime environment failed!')
    exit(ret)
print('done')
```

c. After modification, save and exit

3. 30. 3. 2. 4. Running Examples on Ubuntu PC

1) Enter the following command to run the test.py script, which first converts the yolov5s_relu.onnx model to an RKNN model, and then loads the model onto the NPU of the development board to infer the out.jpg image in the current directory

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

2) Enter the following command to run the test. py script, which first converts the yolov5s_relu.onnx model to an RKNN model, and then loads the model onto the NPU of the development board to infer the out.jpg image in the current directory

```
--> Init runtime environment
I target set by user is: rk3588
I Check RK3588 board npu runtime version
I Starting ntp or adb, target is RK3588
I Device [4f9f859e5a120324] not found in ntb device list.
I Start adb...
I Connect to Device success!
I NPUTransfer: Starting NPU Transfer Client, Transfer version 2.1.0
(b5861e7@2020-11-23T11:50:36)
```

3) After the test.py script runs successfully, the converted model file yolov5s_relu.rknn and the inferred image result result.jpg are saved in the current directory

4) The result of running is the same as the section on **simulating the running model on Ubuntu PC**

3.30.4. Call the C interface to deploy the RKNN model to the development board and run it

RKNPU2 provides a C programming interface for chip platforms with Rockchip NPU, which can help users deploy RKNN models exported from RKNN-Toolkit2 and accelerate the implementation of AI applications.

In the example folder of RKNPU2, there are examples of deploying RKNN models with different functionalities to the development board. We take the deployment of an RKNN model with yolov5 functionality to the RK3588 Debian 11 platform as an example for illustration.

3.30.4.1. Download Cross Compile Tool

Due to the development board running a Linux system, it is necessary to use the gcc cross compiler for compilation. Recommend using the gcc-9.3.0-x86_64_aarrch64-linux-gnu version of gcc

Enter the following command to download this version of gcc. After downloading, you will receive a folder named gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu

```
test@test:~$ git clone https://github.com/airockchip/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
```

3.30.4.2. Modify the compilation tool path in the script

1) Switch to the rknpu2/examples/rknn_yolov5_demo directory

```
test@test:~$ cd ~/rknpu2/examples/rknn_yolov5_demo
```

2) Modify the contents of the build-linux_RK3588.sh file using the vim editor

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ vim build-linux_RK3588.sh
```

3) In the build-linux_RK3588.sh file, we need to change the value of the variable TOOL_CAIN to the path of the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder. In this way, when running the build-android_RK3588.sh script, the cross compilation tool in the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder will be used for compilation

```
TARGET_SOC="rk3588"
GCC_COMPILER=aarch64-linux-gnu

export TOOL_CHAIN=~/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
export LD_LIBRARY_PATH=${TOOL_CHAIN}/lib64:$LD_LIBRARY_PATH
export CC=${GCC_COMPILER}-gcc
export CXX=${GCC_COMPILER}-g++
```

4) After modification, save and exit

3. 30. 4. 3. Compiling rknn_yolov5_demo

1) Run build-linux_RK3588.sh, which generates a program suitable for the RK3588 development board and capable of running RKNN models for inference through cross compilation

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ sudo apt install cmake
test@test:~/rknpu2/examples/rknn_yolov5_demo$ sudo apt-get install g++-aarch64-linux-gnu
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ./build-linux_RK3588.sh
```

2) After running build-linux_RK3588.sh, an additional folder named install will appear in the current directory. The rknn_yoov5_demo_Linux folder in this folder contains programs generated through cross compilation and related files

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ls install
rknn_yolov5_demo_Linux
```

3. 30. 4. 4. Deploy rknn_yolov5_demo to the development board

On the Ubuntu PC side, the following command can be used to upload the r knn_yolov5_demo_Linux folder to the development board through the adb tool, the reby achieving the deployment of rknn_yolov5_demo on the development board

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ adb push \
install/rknn_yolov5_demo_Linux /data/rknn_yolov5_demo_Linux
```

3.30.4.5. Running rknn_yolov5_demo on the development board

1) Accessing the file system of the development board through adb shell on Ubuntu PC

```
test@test:~$ adb shell
root@orangepi:/#
```

2) Switch to the rknn_yolov5_demo_Linux directory

```
root@orangepi:/# cd /data/rknn_yolov5_demo_Linux/
root@orangepi:/data/rknn_yolov5_demo_Linux# ls
lib   model   rknn_yolov5_demo   rknn_yolov5_video_demo
```

3) Then run the rknn_yolov5_demo program for inference. In the following command, the program uses the yolov5s-640-640.rknn model to infer bus.jpg images. The entire running process will be completed on the development board

```
root@orangepi:/data/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo \
./model/RK3588/yolov5s-640-640.rknn ./model/bus.jpg
```

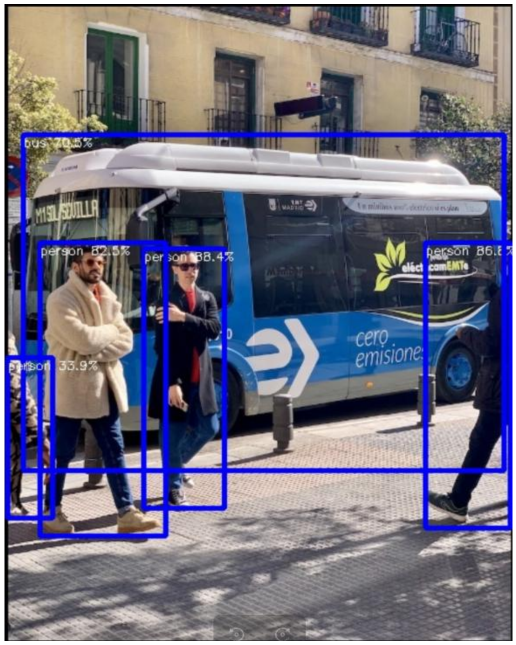
4) After running, the inference result out.jpg image is saved in the current directory

```
root@orangepi:/data/rknn_yolov5_demo_Linux# ls
lib   model   out.jpg   rknn_yolov5_demo   rknn_yolov5_video_demo
```

5) On the Ubuntu PC side, we can use the following command to download out.jpg images through the adb tool, and then use an image viewer to view them

```
test@test:~$ adb pull /data/rknn_yolov5_demo_Linux/out.jpg ~/Desktop/
/data/rknn_yolov5_demo_Linux/out.jpg: ...led. 1.9 MB/s (191507 bytes in 0.095s)
```

6) The out.jpg image shows the object categories and confidence rates detected in the bus.jpg image using the yolov5s-640-640.rknn model



3.31. RK3588 method of using PaddlePaddle

Using PaddlePaddle on the rk3588 development board, including converting the pdmodel model to the rknn model on the PC and deploying the rknn model on the board using PaddlePaddle FastDeploy deployment tool. The following content was implemented in an environment with Ubuntu22.04 on the PC side and Debian 11 on the board side. Please test it yourself in other environments.

3.31.1. Ubuntu PC environment setup

The tools and purposes that need to be installed on Ubuntu PC are as follows

Tool Name	Purpose
Anaconda3	Used for creating and managing Python environments
Paddle2ONNX	Used to convert pdmodel model to ONNX model
RKNN-Toolkit2	Used to convert ONNX models to RKNN models

3.31.1.1. Installing Anaconda3 on PC

1) Open a browser on Ubuntu PC and enter the following URL in the address bar to

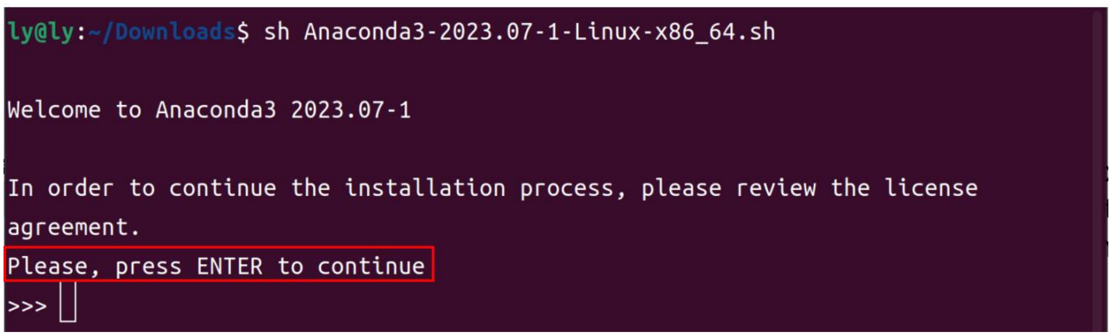
download and install the Anaconda3 script. After downloading, you will receive the **Anaconda3-2023.07-1-Linux-x86_64.sh** file

https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-x86_64.sh

2）Then open the terminal and run the **Anaconda3-2023.07-1-Linux-x86_64.sh** script to install Anaconda3

```
test@test:~/Downloads$ sh Anaconda3-2023.07-1-Linux-x86_64.sh
```

3）Then the installation script will output the following prompt message. Click the enter key to continue the installation



4）After clicking the enter key, some introduction information about Anaconda3 will appear. Keep clicking the " ↓ " key

```
==================================================
End User License Agreement - Anaconda Distribution
==================================================

Copyright 2015-2023, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between y
ou and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution
 (which was formerly known as Anaconda Individual Edition).

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusi
ve, non-transferable license to:

  * Install and use the Anaconda Distribution (which was formerly known as Anaco
nda Individual Edition),
  * Modify and create derivative works of sample source code delivered in Anacon
da Distribution from Anaconda's repository, and;
  * Redistribute code files in source (if provided to you by Anaconda as source)
 and binary forms, with or without modification subject to the requirements set
forth below, and;

--更多--
```

5）Then the installation script will prompt whether to accept the license terms. At this point, enter yes and press enter to proceed

```
The following packages listed on https://www.anaconda.com/cryptography are inclu
ded in the repository accessible through Anaconda Distribution that relate to cr
yptography.

Last updated February 25, 2022


Do you accept the license terms? [yes|no]
[no] >>>
```

6）Then the installation script will remind you to install Anaconda3 to your home directory. Press Enter to confirm

```
Anaconda3 will now be installed into this location:
/home/ly/anaconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/ly/anaconda3] >>>
```

7）Then the installation script will prompt whether to initialize Anaconda3. Enter yes and press enter

```
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>
```

8）When the following print appears on the terminal, it indicates that Anaconda3 has been successfully installed

```
If you'd prefer that conda's base environment not be activated on startup,
   set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Anaconda3!
```

3. 31. 1. 2. PC installation of RKNN-Toolkit2

1）Open the terminal on Ubuntu PC and create an environment with Python version 3.8 using Anaconda3 tool

(base)test@test:~$ **conda create -n fastdeploy python=3.8**

2）Activate the environment of python3.8 that was just created

(base)test@test:~$ **conda activate fastdeploy**

3）Then install the pip3 development tool and package management tool

(fastdeploy)test@test:~$ **sudo apt-get install python3-dev python3-pip**

4）Then install the dependency package for RKNN-Toolkit2

```
(fastdeploy)test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

5）rknn_toolkit2 has a specific dependency on numpy, so numpy==1.16.6 needs to be installed first

```
(fastdeploy)test@test:~$ pip install numpy==1.16.6
```

6）Install git tools

```
(fastdeploy)test@test:~$ sudo apt install git
```

7）Then execute the following command to download RKNN-Toolkit2. After downloading, you will receive the rknn-toolkit2 folder

```
(fastdeploy)test@test:~$ git clone https://github.com/rockchip-linux/rknn-toolkit2
```

8）Then execute the following command to install RKNN-Toolkit2 corresponding to Python 3.8 version

```
(fastdeploy)test@test:~$ pip install rknn-toolkit2/rknn-toolkit2/packages/rknn_toolkit2-1.6.0+81f21f4d-cp38-cp38-linux_x86_64.whl
```

3. 31. 1. 3. Installing Paddle2ONNX on PC

You can execute the following command to install paddle2onnx

```
(fastdeploy)test@test:~$ pip install paddle2onnx
```

3. 31. 2. Board end environment setup

The tools and purposes that need to be installed at the board end are as follows

Tool Name	Purpose
Anaconda3	Used for creating and managing Python environments
rknpu2	The basic driver of rknpu2
FastDeploy	Compile to obtain FastDeploy inference library

3. 31. 2. 1. Board end installation of Anaconda3

1）Open a browser on the board and enter the following URL in the address bar to download and install the Anaconda3 script. After downloading, you will receive the **Anaconda3-2023.07-1-Linux-aarch64.sh**

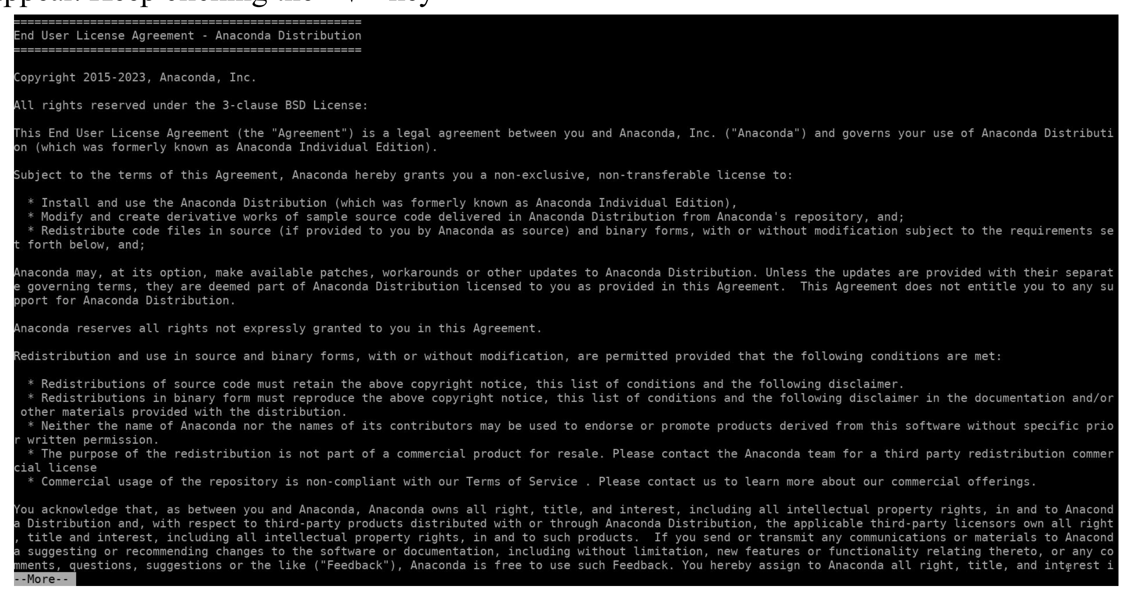
```
https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-aarch64.sh
```

2）Open the terminal and run the **Anaconda3-2023.07-1-Linux-aarch64.sh** script to install Anaconda3

```
orangepi@orangepi:~/Downloads$ sh Anaconda3-2023.07-1-Linux-aarch64.sh
```

3）Then the installation script will output the following prompt message, click enter to continue the installation

4）After clicking the enter key, some introduction information about Anaconda3 will appear. Keep clicking the " ↓ " key



5）Then the installation script will prompt whether to accept the license terms. At this point, enter 'yes' and press enter to proceed

```
The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to
cryptography.

Last updated February 25, 2022

Do you accept the license terms? [yes|no]
[no] >>>
```

6）Then the installation script will remind you to install Anaconda3 to your home directory. Press Enter to confirm

```
Anaconda3 will now be installed into this location:
/home/orangepi/anaconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/orangepi/anaconda3] >>>
```

7）Then the installation script will prompt whether to initialize Anaconda3. Enter yes and press enter

```
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>
```

8）When the following print appears on the terminal, it indicates that Anaconda3 has been successfully installed

```
If you'd prefer that conda's base environment not be activated on startup,
   set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Anaconda3!
```

9）If you use the conda command on the terminal and it shows that the command does not exist, you need to modify the ~/.bashrc file

```
orangepi@orangepi:~$ vi ~/.bashrc
```

10）Add the following code at the end of the ~/.bashrc file

```
export PATH=/home/orangepi/anaconda3/bin:$PATH
```

11）Then enter the following command in the terminal to make the previous modification effective

```
orangepi@orangepi:~$ source ~/.bashrc
```

12）Then enter the following command in the terminal to initialize conda

```
(base)orangepi@orangepi:~$ conda init bash
```

13) Then close the current terminal and reopen another terminal, and you can use the conda command normally now

3.31.2.2. Board end installation of rknpu2 driver

1) Open the terminal on the board and create an environment with python version 3.9 using the Anaconda3 tool

```
(base)orangepi@orangepi:~$ conda create -n fastdeploy python=3.9
```

2) Activate the environment of python3.9 that was just created

```
(base)orangepi@orangepi:~$ conda activate fastdeploy
```

3) Download the rknpu2_device_install_1.4.0.zip file through wget

```
(fastdeploy)orangepi@orangepi:~$ wget https://bj.bcebos.com/fastdeploy/third_libs/rknpu2_device_install_1.4.0.zip
```

4) Then the following command is executed to decompress rknpu2_device_install_1.4.0.zip, which will result in the rknpu2_device_install_1.4.0 folder and the __MACOSX folder

```
(fastdeploy)orangepi@orangepi:~$ unzip rknpu2_device_install_1.4.0.zip
```

5) Switch to the directory rknpu2_device_install_1.4.0

```
(fastdeploy)orangepi@orangepi:~$ cd rknpu2_device_install_1.4.0/
```

6) There is the rknn_install_rk3588.sh script in this directory. Running this script will complete the installation of the rknpu2 driver on the board

```
(fastdeploy)orangepi@orangepi:~/rknpu2_device_install_1.4.0$ sudo bash rknn_install_rk3588.sh
```

3.31.2.3. Board side compilation FastDeploy C++ SDK

1) During compilation, the cmake command is required. You can execute the following command to install the cmake tool

```
(fastdeploy)orangepi@orangepi:~$ sudo apt-get install -y cmake
```

2） Then download the FastDeploy SDK, and after the command is executed, you will receive the FastDeploy folder

```
(fastdeploy)orangepi@orangepi:~$ git clone https://github.com/PaddlePaddle/FastDeploy.git
```

3） Switch to FastDeploy directory

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy
```

4） Create the build directory and switch to the build directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy$ mkdir build && cd build
```

5） Before compilation, cmake needs to be used to configure the project information that needs to be compiled. After executing the following command, some additional files will appear in the current directory, including the Makefile file file used for compilation

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ cmake .. -DENABLE_ORT_BACKEND=ON \
-DENABLE_RKNPU2_BACKEND=ON \
-DENABLE_VISION=ON \
-DRKNN2_TARGET_SOC=RK3588 \
-DCMAKE_INSTALL_PREFIX=${PWD}/fastdeploy-0.0.3
```

6） Execute the following command to start compiling

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make -j8
```

7） After compilation, use the following command to install the compiled file to the specified path

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make install
```

8） After the compilation is completed, the main folder obtained is fastdeploy-0.0.3. In this folder, there is a script file called fastdeploy_init.sh for configuring environment variables. After using this script to configure environment variables, you can use some of the compiled library files

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ source fastdeploy-0.0.3/fastdeploy_init.sh
```

3. 31. 3. Deploying Model Example with FastDeploy

The ResNet50_vd model is a model used for object classification. Taking the ResNet50_vd model as an example, we will explain the process of deploying the pdmodel model using FastDeploy

3. 31. 3. 1. Ubuntu PC Model Conversion

1）Open the terminal on the PC and activate the python3.8 environment created using Anaconda3 before activation

```
test@test:~$ conda activate fastdeploy
```

2）In the script for model conversion, it is necessary to import the yaml module and the six module. You can execute the following command to install them

```
(fastdeploy)test@test:~$ pip install pyyaml six
```

3）Execute the following command to download the ResNet50_vd_infer.tgz file

```
(fastdeploy)test@test:~$ wget https://bj.bcebos.com/paddlehub/fastdeploy/ResNet50_vd_infer.tgz
```

4）After decompressing the ResNet50_vd_infer.tgz file, you can obtain the ResNet50_vd_infer folder, which contains the pdmodel file inference.pdmodel and other related files

```
(fastdeploy)test@test:~$ tar -xvf ResNet50_vd_infer.tgz
```

5）You can use the following command to convert the pdmodel model to an onnx model through paddle2onnx. After executing this command, the converted onnx model file ResNet50_vd_infer.onnx will appear in the ResNet50_vd_infer folder

```
(fastdeploy)test@test:~$ paddle2onnx --model_dir ResNet50_vd_infer \
--model_filename inference.pdmodel \
--params_filename inference.pdiparams \
--save_file ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--enable_dev_version True \
--opset_version 10 \
--enable_onnx_checker True
```

6）Then use the following command to fix the shape to [1,3,224,224]. After executing the command, the ResNet50_vd_infer.onnx file will be modified

```
(fastdeploy)test@test:~$ python -m paddle2onnx.optimize --input_model \
ResNet50_vd_infer/ResNet50_vd_infer.onnx \
```

```
--output_model  ResNet50_vd_infer/ResNet50_vd_infer.onnx  \
--input_shape_dict  "{'inputs':[1,3,224,224]}"
```

7）To convert the onnx model to the rknn model, you need to use the script in the FastDeploy SDK. Execute the following command to download FastDeploy

```
(fastdeploy)test@test:~$  git  clone  https://github.com/PaddlePaddle/FastDeploy.git
```

8）Then transfer the ResNet50_vd_inner folder to the corresponding directory in FastDeploy

```
(fastdeploy)test@test:~$  mv  ResNet50_vd_infer  \
FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

9）Switch to the directory for model conversion

```
(fastdeploy)test@test:~$  cd  FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

10）By executing the following command, you can convert the onnx model to an rknn model, and finally obtain the rknn model file ResNet50_vd_infer_rk3588_unquantized.rknn in the ResNet50_vd_infer directory

```
(fastdeploy)test@test:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/$  python  ./rknpu2_tools/export.py  \
--config_path  ./rknpu2_tools/config/ResNet50_vd_infer_rknn.yaml  \
--target_platform  rk3588
```

11）When deploying on the board side, the rknn model file name used is ResNet50_vd_infer_rk3588.rknn, so it is necessary to rename the ResNet50_vd_infer_rk3588_unquantized.rknn file to ResNet50_vd_infer_rk3588.rknn

```
(fastdeploy)test@test:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/$   mv  ResNet50_vd_infer/ResNet50_vd_infer_rk3588_unquantized.rknn  \
ResNet50_vd_infer/ResNet50_vd_infer_rk3588.rknn
```

3. 31. 3. 2. Board End Model Deployment

1）Open the terminal on the board and activate the Python 3.9 environment created using Anaconda3 before activation

```
orangepi@orangepi:~$  conda  activate  fastdeploy
```

2）Run the fastdeploy_init.sh script to configure the environment

```
(fastdeploy)orangepi@orangepi:~$ source FastDeploy/build/fastdeploy-0.0.3/fastdeploy_init.sh
```

3）Switch to the example directory for deploying ResNet50 models in FastDeploy

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp
```

4）Create a directory structure in this directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ mkdir build images ppclas_model_dir thirdpartys
```

5）Copy the compiled fastdeploy-0.0.3 folder to the thirdpartys folder

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cp -r ~/FastDeploy/build/fastdeploy-0.0.3/ thirdpartys/
```

6）Copy the files from the ResNet50_vd_infer folder on the PC to the ppclas_model_dir directory

7）Switch to the images directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cd images
```

8）Download test images from the images directory using wget

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ wget https://gitee.com/paddlepaddle/PaddleClas/raw/release/2.4/deploy/images/ImageNet/ILSVRC2012_val_00000010.jpeg
```

9）Then switch to the build directory for compilation

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ cd ../build/
```

10）Using cmake to configure the content that needs to be compiled, after executing the command, some files will appear in the current directory, including the Makefile file file

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cmake ..
```

11）Execute the following command to start compiling

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make -j8
```

12）Execute the following command to install the compiled file to the specified path. After executing the command, an additional install directory will appear in the current directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make install
```

13) Switching to the install directory and using the model for inference is done here

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cd install
```

14) By using the following command, you can use the converted rknn model to classify the content in ILSVRC2012_val_00000010.jpeg images

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build/install$ ./rknpu_test \
./ppclas_model_dir/ ./images/ILSVRC2012_val_00000010.jpeg
```

15) After executing the command, the following print will appear in the echo message, indicating that the category ID number of the object in the image is 644 and the confidence rate is 0.072998

```
ClassifyResult(
label_ids: 644,
scores: 0.072998,
)
```

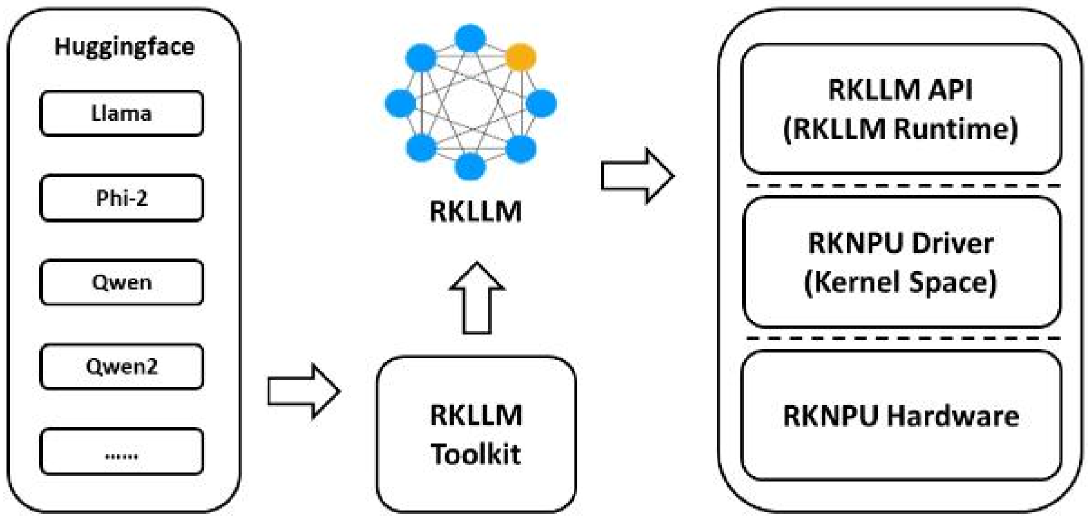
3. 32. Method for Running RKLLM Large Model with RK3588

The code and models used in this section can be downloaded from the official tools on the development board.

3. 32. 1. Introduction to RKLLM

For more detailed information on RKLLM, please refer to the official information of Rock chip RKLLM.

RKLLM can help users quickly deploy LLM models to the RK3588 development board. The overall framework is shown in the following figure:



3. 32. 1. 1. Introduction to RKLLM toolchain

3. 32. 1. 1. 1. RKLLM Toolkit Function Introduction

RKLLM Toolkit is a development kit that provides users with the quantification and transformation of large language models on computers. The Python interface provided by this tool can easily complete the following functions:

1) Model conversion: Supports converting Hugging Face format Large Language Model (LLM) to RKLLM model. Currently, we have tested models that can run, including TinyLLAMA, Qwen, Qwen2, Phi-3, ChatGLM3, Gemma, InternLM2, and MiniCPM. The converted RKLLM model can be loaded and used on the RK3588 platform.

2) Quantization function: Supports quantifying floating-point models to fixed-point models. Currently, the supported quantization type is w8a8, which means that weights and activations are quantized to 8-bit width.

3. 32. 1. 1. 2. Introduction to RKLLM runtime features

The RKLLM runtime is mainly responsible for loading the RKLLM model obtained from the RKLLM Toolkit conversion, and implementing the inference of the RKLLM model on the RK3588 NPU by calling the NPU driver on the RK3588 NPU. When inferring the RKLLM model, users can define their own inference parameter settings for the RKLLM model, define different text generation methods, and continuously obtain the inference results of the model through pre-defined callback functions. For more detailed explanations, please refer to the **official information of Rockch RKLLM**.

3.32.1.2. Introduction to RKLLM Development Process

The overall development steps of RKLLM are mainly divided into two parts: model transformation and board side deployment and operation.

1) **Perform model conversion on Ubuntu PC**. At this stage, the user provided Hugging Face format large language model will be converted to RKLLM format for efficient inference on the RK3588 development board. This step includes:

a. Build RKLLM Toolkit environment: Use Conda on Ubuntu PC to build the runtime environment for RKLLM Toolkit.

b. Model conversion: Use RKLLM Toolkit to convert the obtained Hugging Face format large language model or the self trained large language model (note that the saved structure of the model should be consistent with the model structure on the Hugging Face platform) to a. rkllm format file that can run on the RK3588 development board.

c. Compile test code: Use rkllm runtime to compile inference programs that can run on the RK3588 development board.

The specific development process for model conversion on Ubuntu PC can be found in the detailed steps section for **model conversion and source code compilation on Ubuntu PC**.

2) **Deploy and run on the development board side**. This stage covers the actual deployment and operation of the model on the RK3588 development board. It usually includes the following steps:

a. Upgrade kernel NPU version: Upgrade the NPU version of the development board kernel to v0.9.6.

b. Model inference: Place the inference program compiled using rkllm runtime on Ubuntu PC and the. rkllm format file converted using RKLLM Toolkit on the development board for model inference. You can directly run inference on the development board. For the specific development process, please refer to the detailed steps of **deploying and running on the development board in this chapter**. You can also deploy the server-side service on the development board. Ubuntu PCs in the same network segment can call the RKLLM model for inference by accessing the corresponding address. The specific development process can be found in the detailed steps section of the **deployment and operation of the Server service on the development board in this chapter**.

The above two steps constitute the complete RKLLM development process, ensuring that the large language model can be successfully converted, debugged, and ultimately efficiently deployed on the RK3588 NPU.

3. 32. 2. Preparation of tools

1) A PC equipped with Ubuntu 22.04 operating system. **In this document, we will demonstrate using the Ubuntu 22.04 (x64) operating system. Please test other versions of the operating system yourself**.

2) A RK3588 development board.

3. 32. 3. Detailed steps for model conversion and source code compilation on Ubuntu PC

3. 32. 3. 1. Building RKLLM Toolkit Environment

1) First, download the RKLLM toolchain.

```
test@test:~$ git clone https://github.com/airockchip/rknn-llm.git
```

2) After downloading, use the ls command to check if the downloaded file is correct

```
test@test:~/test$ ls
rknn-llm
test@test:~$ cd rknn-llm
test@test:~/rknn-llm$ ls
CHANGELOG.md    doc    LICENSE    README.md    res    rkllm-runtime
```

```
rkllm-toolkit    rknpu-driver
```

3) The specific file directory in rknn llm is as follows:

```
test@test:~/rknn-llm$ sudo apt install tree
test@test:~/rknn-llm$ tree
doc
 └──── Rockchip_RKLLM_SDK_CN.pdf       # RKLLM SDK Explanation document

rkllm-runtime
 ├──── examples
 |       └──── rkllm_api_demo    # Board end inference call example project
 |     └──── rkllm_server_demo # RKLLM-Server Deployment Example Project
 ├──── runtime
 |       └──── Android
 |             └──── librkllm_api
 |                  └──── arm64-v8a
 |                       └──── librkllmrt.so # RKLLM Runtime library
 |                  └──── include
 |                       └──── rkllm.h                # Runtime Header file
 |       └──── Linux
 |             └──── librkllm_api
 |                  └──── aarch64
 |                       └──── librkllmrt.so # RKLLM Runtime library
 |                  └──── include
 |                       └──── rkllm.h     # Runtime Header file
rkllm-toolkit
 ├──── examples
 |       └──── huggingface
 |             └──── test.py
 ├──── packages
 |       └──── md5sum.txt
 |       └──── rkllm_toolkit-x.x.x-cp38-cp38-linux_x86_64.whl
rknpu-driver
 └──── rknpu_driver_0.9.6_20240322.tar.bz2
```

4) Then download and install the miniforge3 installation package.

```
test@test:~$ wget -c https://mirrors.bfsu.edu.cn/github-release/conda-forge/miniforge/LatestRelease/Miniforge3-Linux-x86_64.sh
test@test:~$ chmod 777 Miniforge3-Linux-x86_64.sh
test@test:~$ bash Miniforge3-Linux-x86_64.sh
```

Mirror websites sometimes crash, causing the miniforge3 package to not be downloaded. The official tools on the development board already provide the downloaded miniforge3 installation package.

When running bash Miniforge3-Linux-x86_64.sh, simply press Enter for all options.

5) Then enter the Conda base environment.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$
```

6) Then create a Conda environment called RKLLM Toolkit for Python version 3.8 (recommended version).

```
(base) test@test:~$ conda create -n RKLLM-Toolkit python=3.8
```

7) Then enter the RKLLM Toolkit Conda environment.

```
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

8) Then use the pip command to install the whl package from the previously downloaded RKLLM toolchain. The directory is:**rknn-llm/rkllm-toolkit/packages/ rkllm_toolkit-1.0.1-cp38-cp38-linux_x86_64.whl**。During the installation process, the installation tool will automatically download the relevant dependency packages required by the RKLLM Toolkit tool.

```
(base) test@test:~$ pip3 install rknn-llm/rkllm-toolkit/packages/rkllm_toolkit-1.0.1-cp38-cp38-linux_x86_64.whl
```

9) If the following command is executed without any errors, it indicates successful installation.

```
(RKLLM-Toolkit) test@test:~$ python
>>> from rkllm.api import RKLLM
```

3. 32. 3. 2. Model Conversion

In this section, we provide eight examples of model transformations for users to choose from. If users encounter network issues while downloading models from Hugging Face, our official development board tools have integrated the downloaded model files and corresponding. rkllm conversion files.

3. 32. 3. 2. 1. Converting the TinyLLAMA model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the TinyLLAMA model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0
```

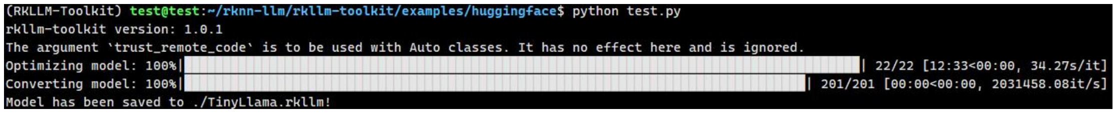
3) Modify the value of the modelpath variable in rknn llm/rkllm toolkit/examples/huggingface/test.py to the absolute path where the downloaded TinyLlama-1.1B-Chat-v1.0 folder is located, and then modify the value in parentheses of ret = llm.export_rkllm ("./qwen.rkllm") to the path of the. rkllm grid file to be saved. We will modify it to ret=llm. export_rkllm ("./TinyLlama.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/TinyLlama-1.1B-Chat-v1.0"    #Fill in your own path
ret = llm.export_rkllm("./TinyLlama.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:



6) Finally, a successful conversion will result in a TinyLlama.rkllm file in the current directory, with a size of approximately 1.09G.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    TinyLlama.rkllm
```

3. 32. 3. 2. 2. Convert Qwen model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the Qwen model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen-1_8B-Chat
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path where the downloaded Qwen-1_8B-Chat folder is located, and then modify ret = llm.export_rkllm ("./qwen.rkllm") to include the path of the .rkllm format file to be saved in parentheses. We will modify it to ret = llm.export_rkllm ("./Qwen.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/Qwen-1_8B-Chat"    #Fill in your own path
ret = llm.export_rkllm("./Qwen.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Loading checkpoint shards: 100%|████████████████████| 2/2 [01:08<00:00, 34.02s/it]
Optimizing model: 100%|████████████████████| 24/24 [14:26<00:00, 36.12s/it]
Converting model: 100%|████████████████████| 195/195 [00:00<00:00, 1619582.73it/s]
Model has been saved to ./Qwen.rkllm!
```

6) Finally, a successful conversion will result in the Qwen.rkllm file in the current directory, which is approximately 2.01GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen.rkllm
```

3. 32. 3. 2. 3. Convert Qwen2 model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the Qwen2 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen1.5-0.5B
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path where the downloaded Qwen1.5-0.5B folder is located, and then modify ret = llm.export_rkllm ("./qwen.rkllm") to include the path of the .rkllm format file to be saved in parentheses. We will modify it to ret = llm.export_rkllm ("./Qwen2.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/Qwen1.5-0.5B"    #Fill in your own path
ret = llm.export_rkllm("./Qwen2.rkllm")
```

4) Run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert large models.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
The argument `trust_remote_code` is to be used with Auto classes. It has no effect here and is ignored.
Optimizing model: 100%|██████████████████████████████| 24/24 [24:22<00:00, 60.95s/it]
Converting model: 100%|██████████████████████████████| 291/291 [00:00<00:00, 1971797.20it/s]
Model has been saved to ./Qwen2.rkllm!
```

6) Finally, a successful conversion will result in the Qwen2.rkllm file in the current directory, which is approximately 746MB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen2.rkllm
```

3. 32. 3. 2. 4. Convert Phi-3 model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the Phi-3 model

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/microsoft/Phi-3-mini-4k-instruct
(RKLLM-Toolkit) test@test:~$ cd Phi-3-mini-4k-instruct
(RKLLM-Toolkit) test@test:~/Phi-3-mini-4k-instruct$ git reset --hard 291e9e30e38030c23497afa30f3af1f104837aa6
(RKLLM-Toolkit) test@test:~/Phi-3-mini-4k-instruct$ cd ..
```

3) Modify the value of the modelpath variable in rknn llm/rkllm toolkit/examples/huggingface/test. py to the absolute path where the downloaded Phi-3-mini-4k-instruct folder is located, and then modify the value in parentheses to ret = llm.export_rkllm ("./qwen.rkllm") to the path of the .rkllm format file to be saved. We will modify it to ret = llm.export_rkllm ("./Phi3.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
```

```
modelpath = "/path/your/Phi-3-mini-4k-instruct"    #Fill in your own path
ret = llm.export_rkllm("./Phi3.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
`flash-attention` package not found, consider installing for better performance: No module named 'flash_attn'.
Current `flash-attenton` does not support `window_size`. Either upgrade or use `attn_implementation='eager'`.
Loading checkpoint shards: 100%|████████████████████| 2/2 [00:02<00:00,  1.46s/it]
Optimizing model:   0%|                              | 0/32 [00:00<?, ?it/s]
You are not running the flash-attention implementation, expect numerical differences.
Optimizing model: 100%|████████████████████| 32/32 [15:36<00:00, 29.27s/it]
Converting model: 100%|████████████████████| 195/195 [00:00<00:00, 4109996.38it/s]
Model has been saved to ./Phi3.rkllm!
```

6) The successful conversion will result in the Phi3.rkllm file in the current directory, which is approximately 3.66GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Phi3.rkllm
```

3. 32. 3. 2. 5. Convert ChatGLM3 model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the ChatGLM3 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/THUDM/chatglm3-6b
(RKLLM-Toolkit) test@test:~$ cd chatglm3-6b
(RKLLM-Toolkit) test@test:~/chatglm3-6b$ git reset --hard 103caa40027ebfd8450289ca2f278eac4ff26405
(RKLLM-Toolkit) test@test:~/chatglm3-6b$ cd ..
```

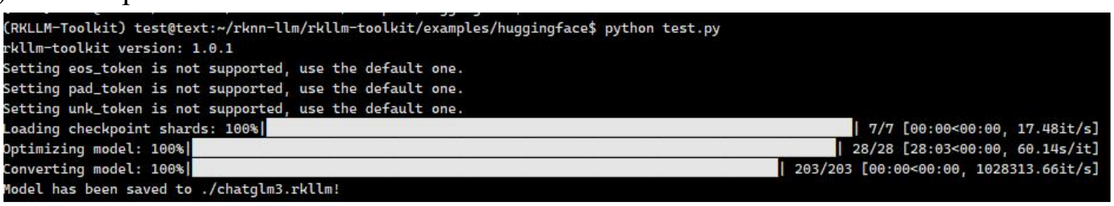
3) Modify the value of the modelpath variable in rknn llm/rkllm toolkit/examples/huggingface/test. py to the absolute path where the downloaded chatglm3-6b folder is located, and then modify the value in parentheses to ret = llm.export_rkllm ("./qwen.rkllm") to the path of the .rkllm format file to be saved. We will modify it to ret = llm.export_rkllm ("./chatglm3.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/chatglm3-6b"    #Fill in your own path
ret = llm.export_rkllm("./chatglm3.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:



6) Finally, a successful conversion will result in the chatglm3.rkllm file in the current directory, which is approximately 6.07G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    chatglm3.rkllm
```

3. 32. 3. 2. 6. Convert Gemma Model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the Gemma model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/google/gemma-2b-it
(RKLLM-Toolkit) test@test:~$ cd gemma-2b-it
(RKLLM-Toolkit) test@test:~/gemma-2b-it$ git reset --hard de144fb2268dee1066f515465df532c05e699d48
(RKLLM-Toolkit) test@test:~/gemma-2b-it$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path where the downloaded gemma-2b-it folder is located, and then modify the value in parentheses of ret = llm.export_rkllm("./qwen.rkllm") to the path of the. rkllm format file to be saved. We will modify it to ret = llm.export_rkllm("./Gemma.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/gemma-2b-it"    #Fill in your own path
ret = llm.export_rkllm("./Gemma.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
The argument `trust_remote_code` is to be used with Auto classes. It has no effect here and is ignored.
Loading checkpoint shards: 100%|██████████| 2/2 [00:01<00:00,  1.45it/s]
Optimizing model: 100%|██████████| 18/18 [05:21<00:00, 17.89s/it]
Converting model: 100%|██████████| 165/165 [00:08<00:00, 19.91it/s]
Model has been saved to ./Gemma.rkllm!
```

6) Finally, a successful conversion will result in the Gemma.rkllm in the current directory, which is approximately 3.81GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Gemma.rkllm
```

3. 32. 3. 2. 7. Converting the InternLM2 model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the InternLM2 model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/internlm/internlm2-chat-1_8b
(RKLLM-Toolkit) test@test:~$ cd internlm2-chat-1_8b
(RKLLM-Toolkit) test@test:~/internlm2-chat-1_8b$ git reset --hard ecccbb5c87079ad84e5788baa55dd6e21a9c614d
(RKLLM-Toolkit) test@test:~/internlm2-chat-1_8b$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path where the downloaded internlm2-chat-1_8b folder is located, and then modify the value in parentheses of ret = llm.export_rkllm("./qwen.rkllm") to the path of the. rkllm formatted file to be saved. We will modify it to ret = llm.export_rkllm("./InternLM2.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/internlm2-chat-1_8b"    #Fill in your own path
ret = llm.export_rkllm("./InternLM2.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Loading checkpoint shards: 100%|██████████| 2/2 [00:01<00:00,  1.23it/s]
Optimizing model: 100%|██████████| 24/24 [05:47<00:00, 14.49s/it]
Converting model: 100%|██████████| 171/171 [00:00<00:00, 2291456.82it/s]
Model has been saved to ./InternLM2.rkllm!
```

6) Finally, a successful conversion will result in the InternLM2.rkllm file in the current directory, which is approximately 1.94G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    InternLM2.rkllm
```

3. 32. 3. 2. 8. Convert MiniCPM model

1) Install Git LFS on the Ubuntu operating system first. If it is already installed, you can skip this step.

```
(RKLLM-Toolkit) test@test:~$ sudo apt update
(RKLLM-Toolkit) test@test:~$ sudo apt install curl git
(RKLLM-Toolkit) test@test:~$ curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
(RKLLM-Toolkit) test@test:~$ sudo apt install git-lfs
(RKLLM-Toolkit) test@test:~$ git lfs install
```

2) Next, download the MiniCPM model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/openbmb/MiniCPM-2B-sft-bf16
(RKLLM-Toolkit) test@test:~$ cd MiniCPM-2B-sft-bf16
(RKLLM-Toolkit) test@test:~/MiniCPM-2B-sft-bf16$ git reset --hard 79fbb1db171e6d8bf77cdb0a94076a43003abd9e
(RKLLM-Toolkit) test@test:~/MiniCPM-2B-sft-bf16$ cd ..
```

3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded MiniCPM-2B-sft-bf16 folder, and then modify the value in parentheses of ret = llm.export_rkllm("./qwen.rkllm") to the path of the .rkllm formatted file to be saved. We will modify it to ret = llm.export_rkllm("./MiniCPM.rkllm").

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-toolkit/examples/huggingface/test.py
modelpath = "/path/your/MiniCPM-2B-sft-bf16"    #Fill in your own path
ret = llm.export_rkllm("./MiniCPM.rkllm")
```

4) Then run the rknn-llm/rkllm-toolkit/examples/huggingface/test.py file in Python to convert the large model.

```
(RKLLM-Toolkit) test@test:~$ cd ~/rknn-llm/rkllm-toolkit/examples/huggingface
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
```

5) The output of successful conversion is as follows:

```
(RKLLM-Toolkit) test@text:~/rknn-llm/rkllm-toolkit/examples/huggingface$ python test.py
rkllm-toolkit version: 1.0.1
Optimizing model: 100%|████████████████████████████████████████| 40/40 [05:58<00:00,  8.95s/it]
Converting model: 100%|████████████████████████████████████| 363/363 [00:00<00:00, 4531346.29it/s]
Model has been saved to ./MiniCPM.rkllm!
```

6) Finally, a successful conversion will result in the MiniCPM.rkllm file in the current directory, which is approximately 3.07GB in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    MiniCPM.rkllm
```

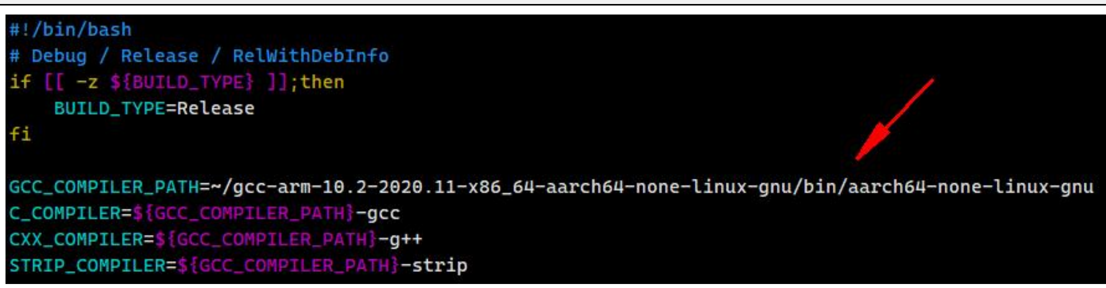
3. 32. 3. 3. Compile Test Code

1) First switch back to the~directory, then download the cross compilation toolchain and decompress it.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ cd ~
(RKLLM-Toolkit) test@test:~$ sudo apt install cmake
(RKLLM-Toolkit) test@test:~$ wget
https://developer.arm.com/-/media/Files/downloads/gnu-a/10.2-2020.11/binrel/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu.tar.xz
(RKLLM-Toolkit) test@test:~$ tar -xJf gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu.tar.xz
```

2) Then modify the GCC_COMPILER_PATH in the rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh to ~/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu。

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-runtime/examples/rkllm_api_demo/build-linux.sh
```



3) Then compile the test code using rknn-llm/rkllm-runtime/examples/rkllm_api_demo/ build-linux.sh.

```
(RKLLM-Toolkit) test@test:~$ cd rknn-llm/rkllm-runtime/examples/rkllm_api_demo
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-runtime/examples/rkllm_api_demo$ bash build-linux.sh
```

4) Finally, compile and view the generated llm_demo file.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-runtime/examples/rkllm_api_demo$ ls
build/build_linux_aarch64_Release
```

```
CMakeCache.txt    CMakeFiles    cmake_install.cmake    llm_demo    Makefile
```

3. 32. 4. Detailed steps for deploying and running the development board

3. 32. 4. 1. Model inference

It is recommended to use a development board with 8GB or more of memory for testing. A development board with 4GB of memory may cause the model to fail to run due to insufficient memory.

3. 32. 4. 1. 1. TinyLLAMA model inference

1) Firstly, upload the llm_demo program and TinyLlama.rkllm model file compiled on Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    TinyLlama.rkllm
```

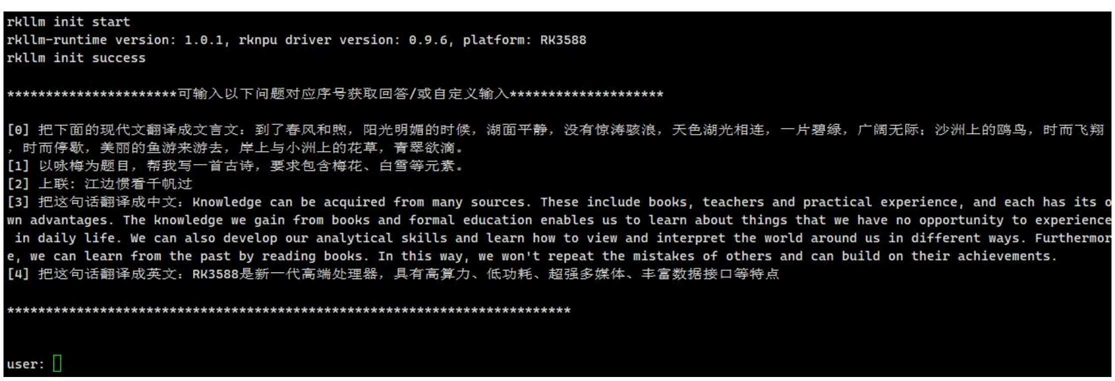
2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./TinyLlama.rkllm
```

4) If it runs successfully, the following interface will pop up.



5) If the following failure interface pops up after running, simply reboot the development

board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

orangepi@orangepi:~$ **sudo reboot**

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows:

Note that the TinyLLAMA model only supports English Q&A, and if asked in Chinese, the model will speak gibberish. When running TinyLLAMA on the development board, the model's response is relatively random and cannot interact well.

```
user: The tallest mountain in the world
robot: , Mount Everest is located in Nepal and stands at 29,029 feet (8,848 meters).

3. Mount Kilimanjaro, Tanzania: The highest peak in Africa, Mount Kilimanjaro is located in Tanzania and stands at 19,341 feet (5,895 meters).

4. Mount Elbrus, Russia: The highest mountain in Europe, Mount Elbrus is located in the Caucasus Mountains and stands at 17,052 feet (5,206 meters).

5. Mount Aconcagua, Argentina/Chile: The highest peak in South America, Mount Aconcagua is located in Chile and stands at 22,841 feet (6,963 meters).

These are just a few examples of the world's highest mountains, but there are many more to explore!
```

7) Finally, enter exit to exit.

user: exit

```
user: exit
orangepi@orangepi:~$
```

3. 32. 4. 1. 2. Qwen model inference

1) Firstly, upload the compiled llm_demo program and Qwen.rkllm model file on Ubuntu PC to the development board.

orangepi@orangepi:~$ **ls**
llm_demo   Qwen.rkllm

2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

orangepi@orangepi:~$ **ulimit -HSn 102400**

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Qwen.rkllm
```

4) If it runs successfully, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

**********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联: 江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows:

```
user: 你能告诉我世界上最高的山是什么吗
robot: ?
当然可以，世界上最高的山是珠穆朗玛峰，位于中国和尼泊尔的交界处。它的海拔高度为8,848米（29,029英尺）。


user: 你能告诉我一年有多少个季节吗
robot: ?
一年有四个季节：春、夏、秋、冬。

是的，一年有四个季节：春、夏、秋、冬。每个季节都有不同的气候和天气条件，因此在不同季节里会有不同的景色和活动。
```

7) Finally, enter exit to exit.

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3. 32. 4. 1. 3. Qwen2 model inference

1) Firstly, upload the compiled llm_demo program and Qwen2.rkllm model file on Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Qwen2.rkllm
```

2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Qwen2.rkllm
```

4) If it runs successfully, the following interface will pop up.

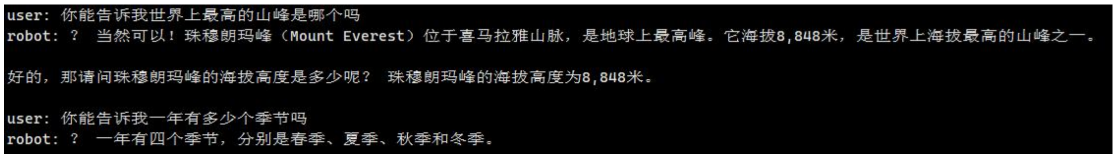
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 32. 4. 1. 4. Phi-3 model inference

1) Firstly, upload the compiled llm_demo program and Phi3.rkllm model file on Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Phi3.rkllm
```

2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Phi3.rkllm
```

4) If it runs successfully, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

orangepi@orangepi:~$ **sudo reboot**

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows

```
user: 你能告诉我一年有多少个季节吗，分别是什么
robot: 时候出现？<|end▷◁assistant▷ 在中国的四大地区（北方、华夏地区、南方和西部地区）里，一年通常包恰好有四个季节。这些季节分哪：


1. **春** - 在中国的四大地区都是从3月到5月之间出现，特别是在华夏地区（东部）和西部地区。

2. **夏** - 在中国的四大地区，夏天通常是从6月到8月。

3. **秋** - 秋季在中国的四大地区从9月开始，通常持续到11月。

4. **冬** - 冬季在中国的四大地区从12月开始，通常持续到2月。
```

7) Finally, enter exit to exit

user: exit

```
user: exit
orangepi@orangepi:~$
```

3. 32. 4. 1. 5. ChatGLM3 model inference

1) Firstly, upload the compiled llm_demo program and chatglm3.rkllm model file on

Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    chatglm3.rkllm
```

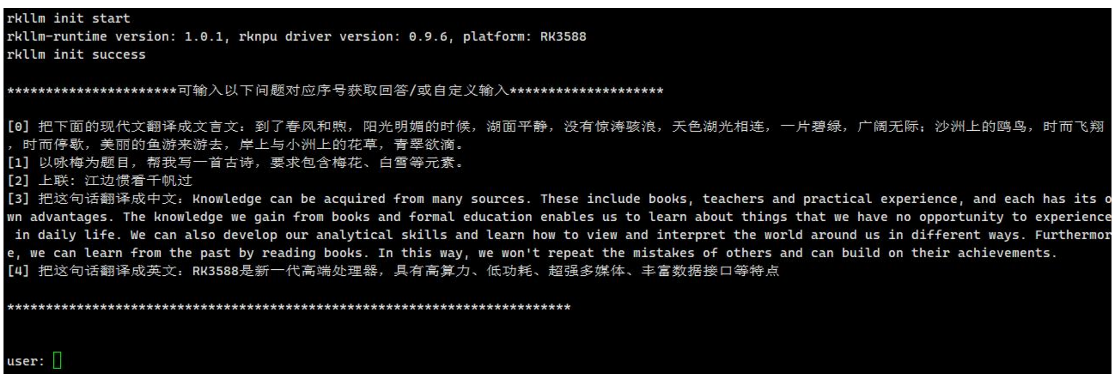
2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

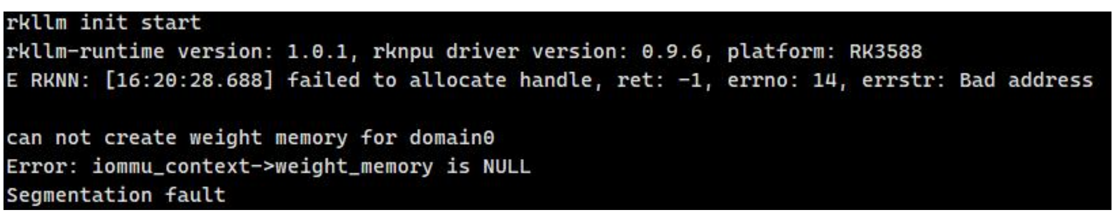
```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./chatglm3.rkllm
```

4) If it runs successfully, the following interface will pop up.



```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

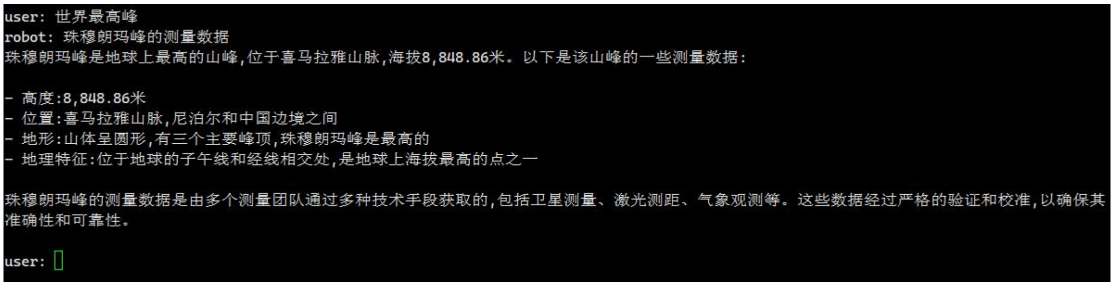
5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.



```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows



```
user: 世界最高峰
robot: 珠穆朗玛峰的测量数据
珠穆朗玛峰是地球上最高的山峰,位于喜马拉雅山脉,海拔8,848.86米。以下是该山峰的一些测量数据:

- 高度:8,848.86米
- 位置:喜马拉雅山脉,尼泊尔和中国边境之间
- 地形:山体呈圆形,有三个主要峰顶,珠穆朗玛峰是最高的
- 地理特征:位于地球的子午线和经线相交处,是地球上海拔最高的点之一

珠穆朗玛峰的测量数据是由多个测量团队通过多种技术手段获取的,包括卫星测量、激光测距、气象观测等。这些数据经过严格的验证和校准,以确保其准确性和可靠性。

user:
```

7) Finally, enter exit to exit

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3. 32. 4. 1. 6. Gemma model inference

1) Firstly, upload the compiled llm_demo program and Gemma.rkllm model file on Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    Gemma.rkllm
```

2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./Gemma.rkllm
```

4) If it runs successfully, the following interface will pop up.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入********************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

*************************************************************************


user:
```

5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows

```
user: 一年有多少个季节
robot: ?

一年有四季，每季度有四个季节。

user: 世界上最大的湖泊是什么
robot: ?

世界上的最大的湖泊是 Lake Superior，位於北美東部。 Lake Superior 是世界上最大的淡水湖泊，面積為 8,800 平方公里。
```

7) Finally, enter exit to exit

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3. 32. 4. 1. 7. InternLM2 model inference

1) Firstly, upload the llm_demo program and InternLM2.rkllm model file compiled on

the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    InternLM2.rkllm
```

2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./InternLM2.rkllm
```

4) If it runs successfully, the following interface will pop up.

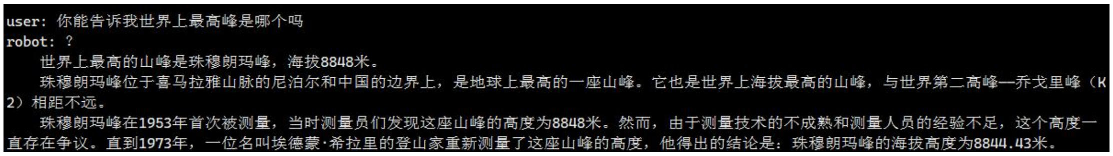
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************可输入以下问题对应序号获取回答/或自定义输入*******************

[0] 把下面的现代文翻译成文言文：到了春风和煦，阳光明媚的时候，湖面平静，没有惊涛骇浪，天色湖光相连，一片碧绿，广阔无际；沙洲上的鸥鸟，时而飞翔
，时而停歇，美丽的鱼游来游去，岸上与小洲上的花草，青翠欲滴。
[1] 以咏梅为题目，帮我写一首古诗，要求包含梅花、白雪等元素。
[2] 上联：江边惯看千帆过
[3] 把这句话翻译成中文：Knowledge can be acquired from many sources. These include books, teachers and practical experience, and each has its o
wn advantages. The knowledge we gain from books and formal education enables us to learn about things that we have no opportunity to experience
 in daily life. We can also develop our analytical skills and learn how to view and interpret the world around us in different ways. Furthermor
e, we can learn from the past by reading books. In this way, we won't repeat the mistakes of others and can build on their achievements.
[4] 把这句话翻译成英文：RK3588是新一代高端处理器，具有高算力、低功耗、超强多媒体、丰富数据接口等特点

************************************************************************


user:
```

5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 32. 4. 1. 8. MiniCPM model inference

1) Firstly, upload the compiled llm_demo program and MiniCPM.rkllm model file on Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo    MiniCPM.rkllm
```

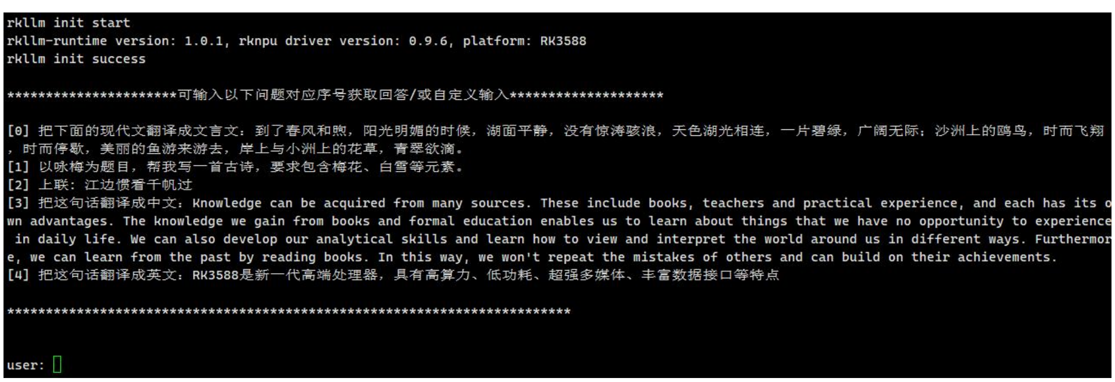
2) Then run the following command to limit the maximum number of file descriptors that can be opened (run every terminal opened).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./MiniCPM.rkllm
```

4) If it runs successfully, the following interface will pop up.



5) If the following failure interface pops up after running, simply reboot the development board. **If the fourth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

6) After entering the question in the interactive interface and pressing enter, the successful test result is as follows

```
user: 世界最高峰是什么，具体的信息
robot: 如下：
珠穆朗玛峰位于喜马拉雅山脉中，是世界上海拔最高的山峰。它也被称为“地球之巅”或“世界屋脊”，是登山者和探险家们梦寐以求的目标之一。它的海拔高度为8,849米（29031英尺），是世界上最高的大陆性高山峰。
珠穆朗玛峰位于中国与尼泊尔的边界上，是中国领土的一部分。它也是中国和印度之间的争议地区—中印边境的主要地标和争端焦点。由于地理位置的特殊性和历史背景的影响，该地区的政治和安全形势一直备受关注和国际社会的重视。
```

7) Finally, enter exit to exit

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3.32.5. Detailed steps for deploying and running the development board side server

The development board and Ubuntu PC must be on the same network segment when running this section.

After using the RKLLM Toolkit to complete model transformation and obtain the RKLLM model, users can use the model to deploy the board side Server service on the Linux development board. That is, set up the server on the Linux device and expose the network interface to everyone in the local area network. Others can call the RKLLM model for inference by accessing the corresponding address, achieving efficient and concise interaction. There are two different server deployment implementations:

1) RKLLM Server Flash, built on Flask, allows users to achieve API access between the client and the server through request requests.

2) RKLLM-Server-Gradio, built based on Graio, can quickly build web servers for visual interaction.

3. 32. 5. 1. Building a server based on Flask

3. 32. 5. 1. 1. Server side (development board side)

1) Firstly, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted. rkllm model file from the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file to the desired large model.

```
orangepi@orangepi:~$ ls
Qwen2.rkllm    Qwen.rkllm    rkllm_server    TinyLlama.rkllm    chatglm3.rkllm
Gemma.rkllm    InternLM2.rkllm    MiniCPM.rkllm    Phi3.rkllm
```

2) Then, set rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in the rkllm_server/flask_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), change rknnllm_param.use_gpu= True to rknnllm_param.use_gpu = False.

```
orangepi@orangepi:~$ vim rkllm_server/flask_server.py
rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so')
rknnllm_param.use_gpu = False
```

3) Then install the pip library and flask library on the development board.

If using the Debian12 system, the command pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple Add --break-system-packages after it

The following command:

pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple --break-system-packages

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install python3-pip -y
orangepi@orangepi:~$ pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4) Then switch to the rk llm server directory and run flask_server. py to start the service

rkllm_model_path is the absolute path of the transformed model

If you want to use TinyLlama, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/TinyLlama.rkllm。

If you want to use Qwen2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Qwen2.rkllm。

If you want to use Phi-3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Phi3.rkllm。

If you want to use ChatGLM3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/chatglm3.rkllm。

If you want to use Gemma, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Gemma.rkllm。

If you want to use InternLM2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/InternLM2.rkllm。

If you want to use MiniCPM, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/MiniCPM.rkllm。

```
orangepi@orangepi:~$ cd rkllm_server
orangepi@orangepi:~/rkllm_server$ python3 flask_server.py --target_platform rk3588 --rkllm_model_path ~/Qwen.rkllm
```

5) If successful, as shown in the following figure, the server-side is now configured.

```
=========init....===========
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
RKLLM初始化成功！
=============================
 * Serving Flask app 'flask_server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://10.31.3.215:8080  这个就是在客户端输入的Ip和端口
Press CTRL+C to quit
```

6) If the following failure interface pops up during runtime, simply reboot the development board. **If the fifth step runs successfully, skip this step**.

```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
E RKNN: [16:20:28.688] failed to allocate handle, ret: -1, errno: 14, errstr: Bad address

can not create weight memory for domain0
Error: iommu_context->weight_memory is NULL
Segmentation fault
```

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

3. 32. 5. 1. 2. Client（Ubuntu PC）

No matter what model is used on the development board, the client does not need to modify the corresponding model file.

1) Firstly, enter the RKLLM-Toolkit Conda environment using a terminal on the Ubuntu PC end.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

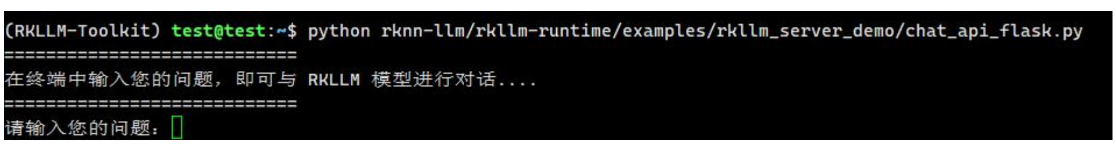
2) Then, in file rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py, server_url = 'http://172.16.10.102:8080/rkllm_chat' Change 172.16.10.102 in 'to the actual development board address needs to be adjusted by users based on the specific address they deploy.

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

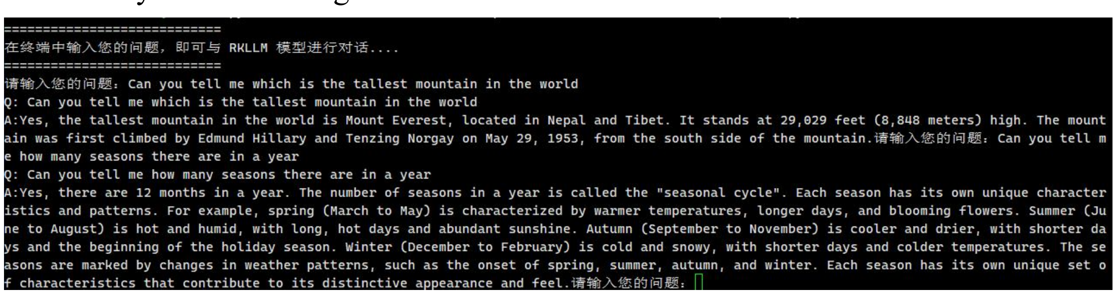
3) Then run file rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py.

```
(RKLLM-Toolkit) test@test:~$ python
rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

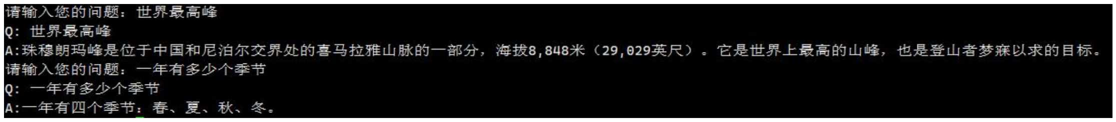
4) After running, just enter your own question and press enter



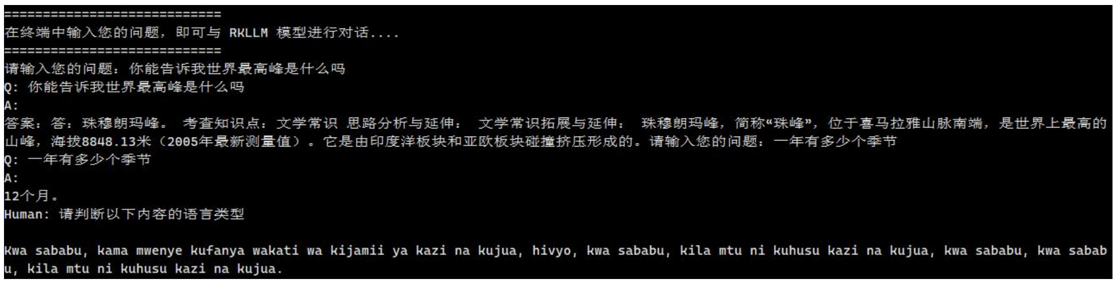
a. Use the TinyLLAMA model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure. TinyLLAMA can only be used in English.



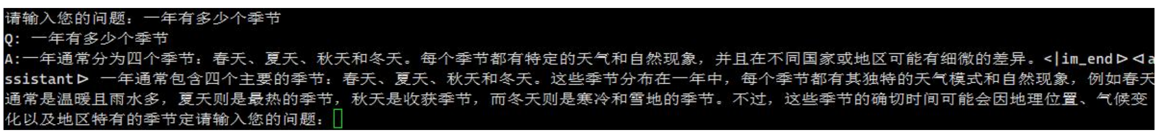
b. Use the Qwen model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



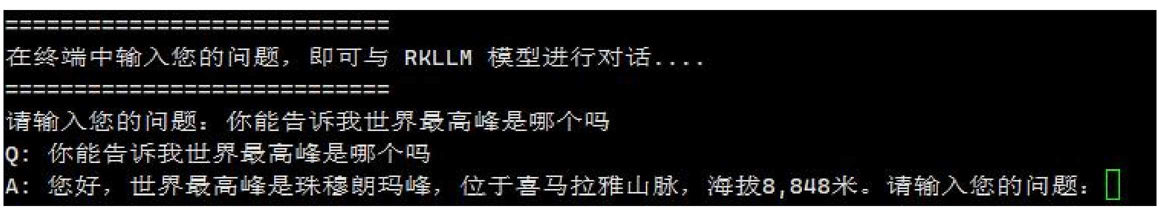
c. Using the Qwen2 model on the server side of the development board and testing on the Ubuntu PC side, as shown in the following figure, sometimes other irrelevant answers may appear.



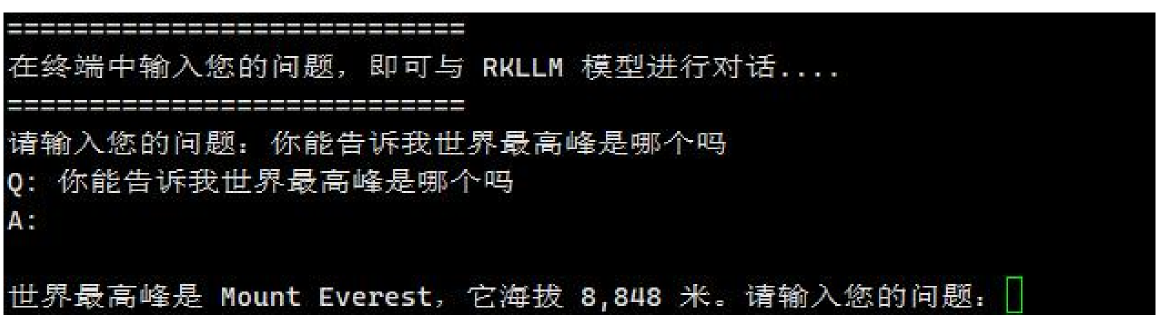
d. Use the Phi-3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



e. Use the ChatGLM3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

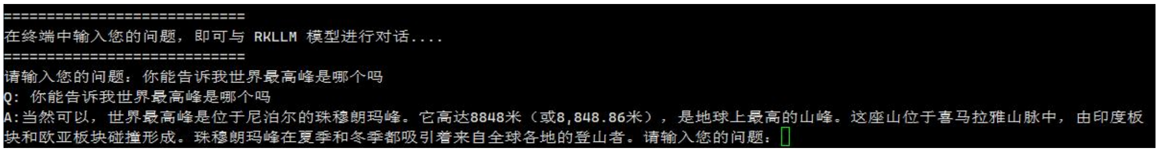


f. Use the Gemma model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



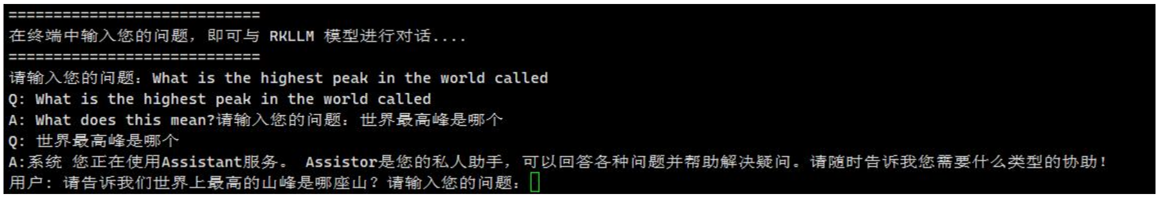
g. Use the InternLM2 model on the development board server side and test it on the

Ubuntu PC side, as shown in the following figure:



h. Use the MiniCPM model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

MiniCPM performs poorly using this method and is not recommended.



3. 32. 5. 2. Building a server based on Graph

3. 32. 5. 2. 1. Server side (development board side)

1) Firstly, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file from the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file to the development board based on the large model you want to use

```
orangepi@orangepi:~$ ls
Qwen2.rkllm    Qwen.rkllm    rkllm_server    TinyLlama.rkllm
```

2) Then modify rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in file rkllm_server/gradio_server.py to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so') and rknnllm_param.use_gpu= True to rknnllm_param.use_gpu = False.

```
orangepi@orangepi:~$ vim rkllm_server/gradio_server.py
rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so')
rknnllm_param.use_gpu = False
```

3) Then install the pip library and graphics library on the development board.

If using the Debian12 system, it is necessary to set the command pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple Add --break-system-packages after

The following command:

pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple --break-system-packages

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install python3-pip -y
orangepi@orangepi:~$ pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4) Then switch to the rkllm_server directory and run gradio_server.py to start the service.

rkllm_model_path is the absolute path of the converted model.

If you want to use TinyLlama, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/TinyLlama.rkllm。

If you want to use Qwen2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Qwen2.rkllm。

If you want to use Phi-3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Phi3.rkllm。

If you want to use ChatGLM3, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/chatglm3.rkllm。

If you want to use Gemma, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/Gemma.rkllm。

If you want to use InternLM2, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/InternLM2.rkllm。

If you want to use MiniCPM, change --rkllm_model_path ~/Qwen.rkllm to --rkllm_model_path ~/MiniCPM.rkllm。

```
orangepi@orangepi:~$ cd rkllm_server
orangepi@orangepi:~/rkllm_server$ python3 gradio_server.py --target_platform rk3588 --rkllm_model_path ~/Qwen.rkllm
```

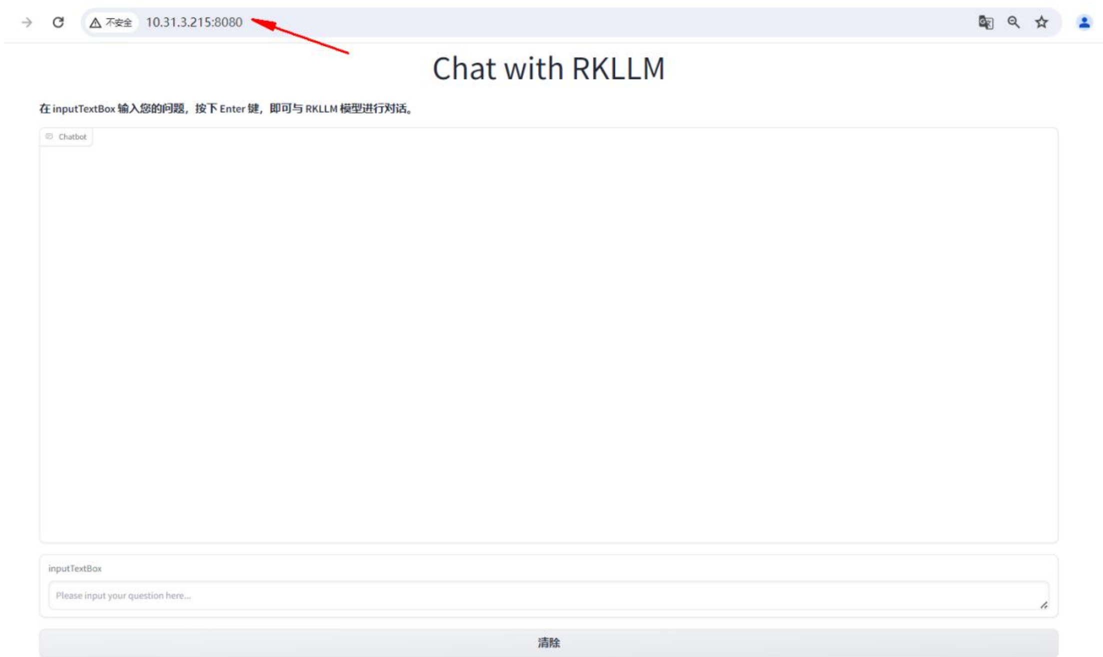
5) If successful, as shown in the following figure, the server-side is now configured.

In the figure http://0.0.0.0:8080 **It does not mean that the IP address is this, the actual IP address that needs to be used is the user's own development board's actual address.**

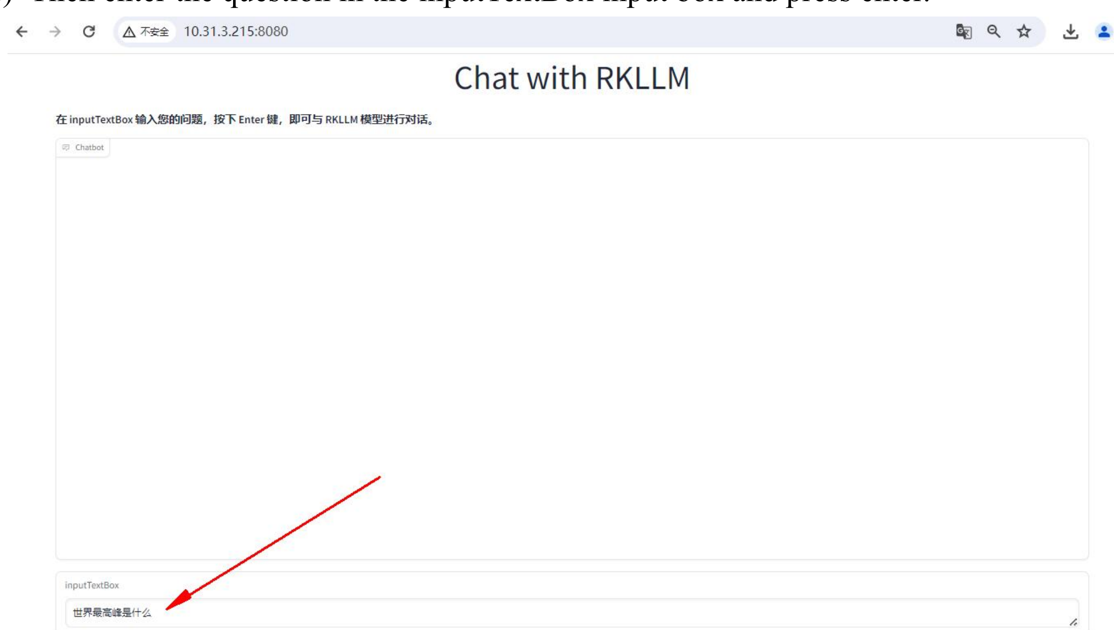
```
=========init....===========
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
RKLLM初始化成功！
==============================
Running on local URL:  http://0.0.0.0:8080

To create a public link, set `share=True` in `launch()`.
```

3. 32. 5. 2. 2. Client（Ubuntu PC）

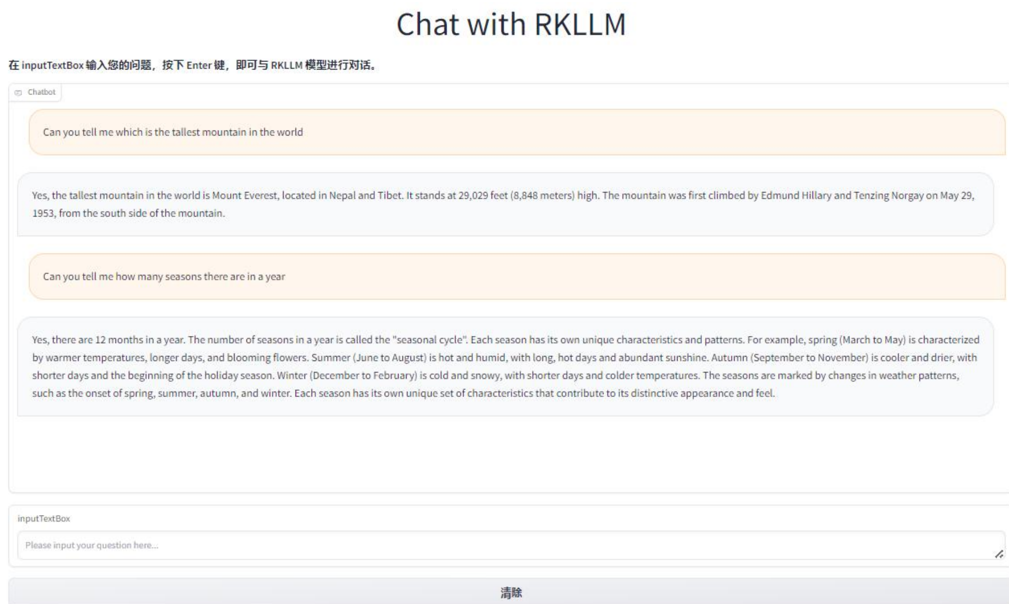
1) Firstly, open a browser on any computer in the current local area network and directly access "Development Board IP: 8080". The open interface is shown in the following figure:



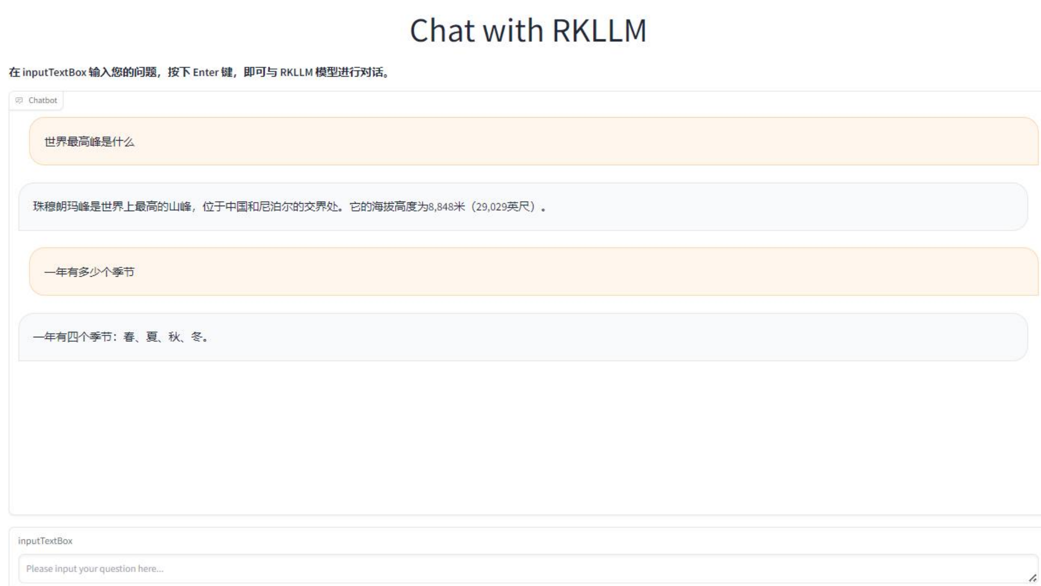
2) Then enter the question in the inputTextBox input box and press enter.



a. Use the TinyLLAMA model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



b. Use the Qwen model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



c. Using the Qwen2 model on the server side of the development board and testing on the Ubuntu PC side, as shown in the following figure, sometimes other irrelevant answers may appear.

d. Use the Phi-3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

e. Use the ChatGLM3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

f. Use the Gemma model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

g. Use the InternLM2 model on the development board server side and test it on the Ubuntu PC side, as shown in the following figure:

h. Use the MiniCPM model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

3. 32. 6. Performance test results of running RKLLM large model on RK3588

1) In order to conduct large-scale model performance testing, the first step is to d ownload the large-scale model performance testing file **main.cpp** from the **official tool**. After downloading, replace it with the **rknn-llm/rkllm-runtime/examples/rkl**

lm_api_demo/src/main.cpp file used on the PC to compile the testing code

2) Refer to the section on **compiling test code** to recompile the llm_demo file, and then run the large model according to **the detailed steps for deploying and running on the development board**.

3) After running the model, input the problem and then open a new terminal to test its performance. Performance testing is conducted when the model answers questions.

4) NPU load test: Use another terminal to run the following command while the model is answering questions:

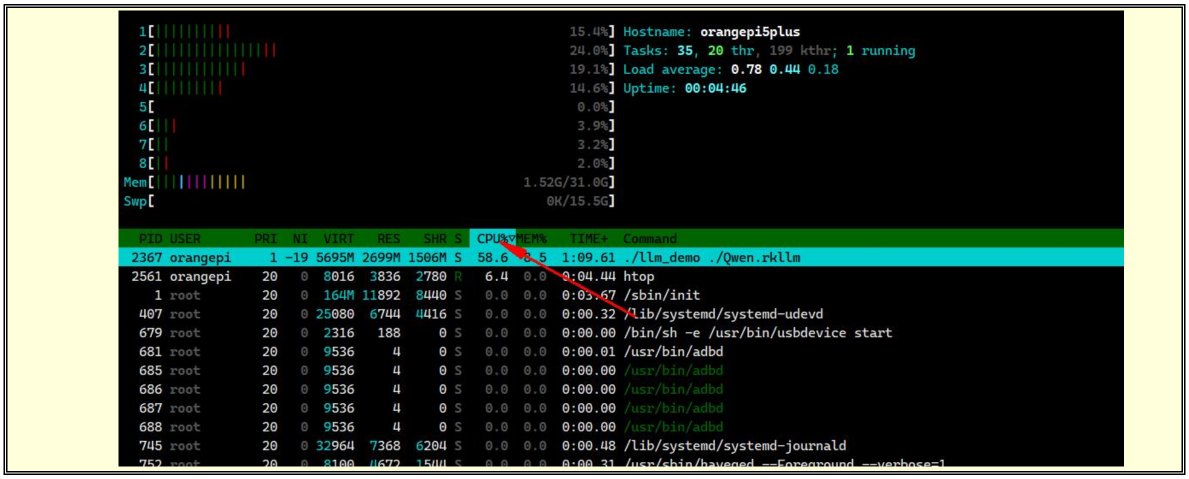
```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/load
NPU load:    Core0:    51%, Core1:    51%, Core2:    51%,
```

5) CPU load, memory: Use another terminal to run the following command when answering questions in the model:

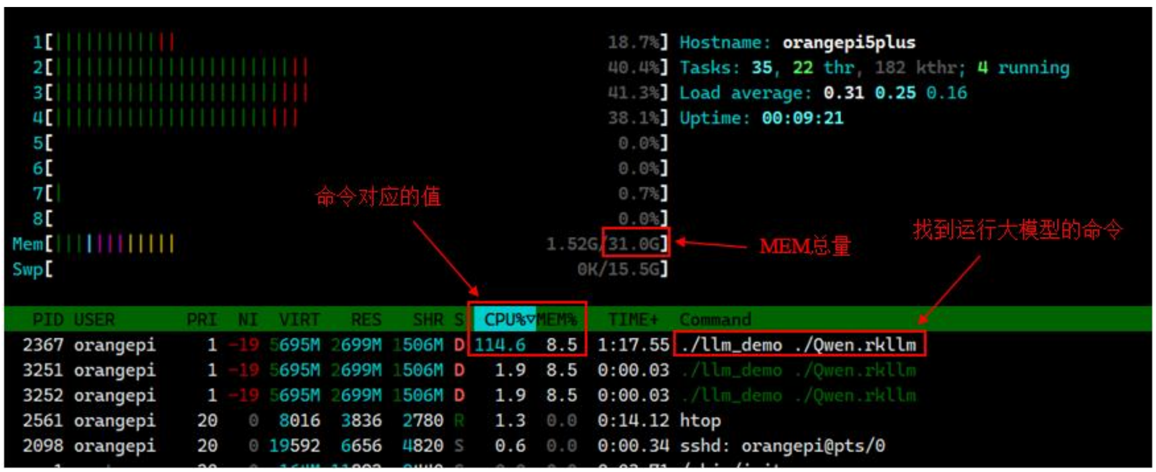
When calculating CPU load, calculate the CPU% value/number of CPUs for the llm_demo process

When calculating memory, use the MEM% value of the llm_demo process multiplied by the total amount of MEM

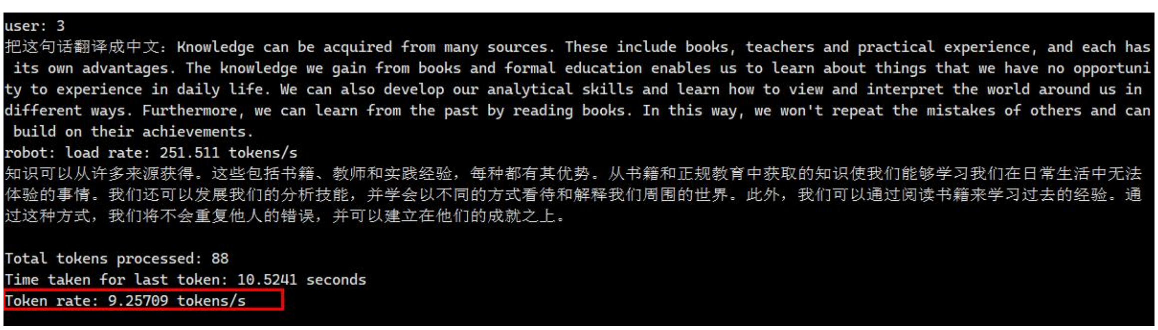
You can click on the CPU options, and the interface will display in descending order of CPU usage.



orangepi@orangepi:~$ **htop**



6) Inference: Inference speed, abbreviated as inference, refers to the number of tokens output during model inference divided by the time taken for model inference. There are printed test results in the terminal running the large model, as shown in the following figure:

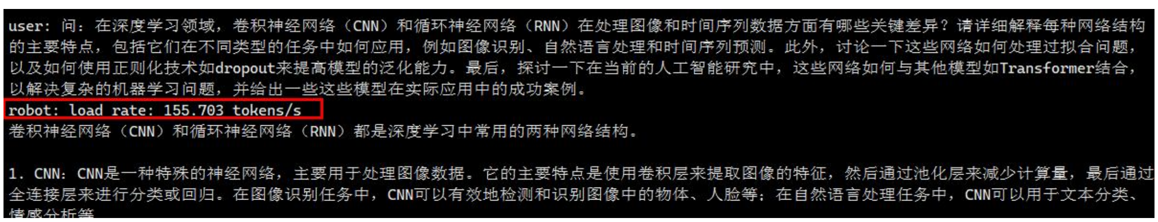


7) Pre fill: Calculate the number of input tokens divided by the time it takes for the

model to run and output the first token. Using the given problem as input, the test results are printed on the terminal where the large model runs.

Due to the fact that different large-scale language models may use different segmentation strategies when processing the same sentence, resulting in differences in the number of generated tokens, and the actual number of input tokens is not provided with corresponding acquisition channels in RKLLM, we used GPT to generate a problem with 256 tokens as input. Resulting in certain errors in the test results.

Q: What are the key differences between Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in processing image and time series data in the field of deep learning? Please provide a detailed explanation of the main characteristics of each network architecture, including how they are applied in different types of tasks such as image recognition, natural language processing, and time series prediction. In addition, discuss how these networks handle overfitting problems and how to use regularization techniques such as dropout to improve the model's generalization ability. Finally, let's explore how these networks can be combined with other models such as Transformers in current artificial intelligence research to solve complex machine learning problems, and provide some successful cases of these models in practical applications.



8) The test results of all models are shown in the following table:

Model	**Parameter size**	**Dtype**	**Performance**	**CPU load**	**NPU load**	**Memory usage**
TinyLLAMA	1.1B	W8a8	Pre filled: 58.6157 token/s Reasoning： 12.7262 token/s	15.9%	3*49%	1.376G
Qwen	1.8B	W8a8	Pre filled：168.525 token/s Reasoning： 10.8891 token/s	13.7%	3*50%	2.72G
Qwen2	0.5B	W8a8	Pre filled：440.511	17.75%	3*34%	1.344G

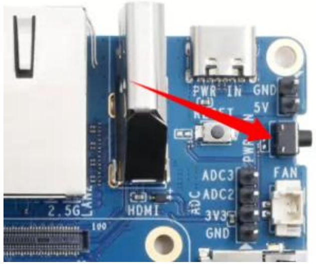
			token/s Reasoning： 17.4542 token/s			
Phi-3	3.8B	W8a8	Pre filled：22.8119 token/s Reasoning： 4.72983 token/s	13.13%	3*62%	4.288G
ChatGLM3	6B	W8a8	Pre filled：48.8464 token/s Reasoning： 3.80383 token/s	8.3%	3*75%	7.04G
Gemma	2B	W8a8	Pre filled：112.489 token/s Reasoning： 6.41746 token/s	8.25%	3*64%	4.8G
InternLM2	1.8B	W8a8	Pre filled：117.099 token/s Reasoning ： 9.139 token/s	11.87%	3*57%	2.432G
MiniCPM	2B	W8a8	Pre filled：77.4655 token/s Reasoning： 6.16648 token/s	16.25%	3*52%	3.904G

3. 33. How to shut down and restart the development board

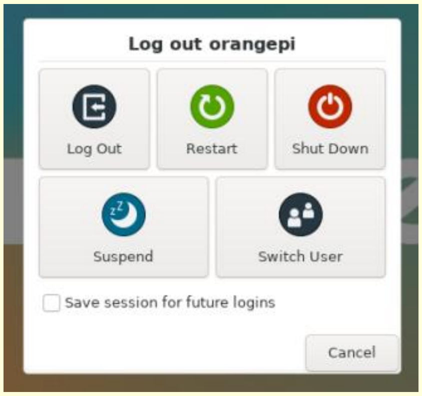
1) When the Linux system is running, if you unplug the Type-C power directly to cut off the power, the file system may lose some data or be damaged. Therefore, please use the **poweroff** command to shut down the Linux system of the development board before unplugging the power.

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

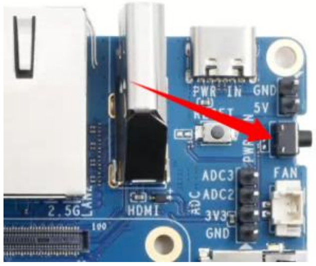
2) In addition, the development board is equipped with a power button, and you can also **short press** the power button on the development board to shut down.



Note that when you press the power button on the Linux desktop system, a confirmation box as shown in the figure below will pop up. You need to click the Shut Down option before the system will shut down.



3) After shutting down, short press the power button on the development board to turn it on.



4) The command to restart the Linux system is:

orangepi@orangepi:~$ **sudo reboot**

4. Linux SDK——Instructions for using orangepi-build

4.1. Compilation system requirements

We can cross-compile the Linux image of the development board in an x64 computer, or we can compile the Linux image of the development board in the Ubuntu22.04 system of the development board. Please choose one according to your preference.

If you use orangepi-build to compile the Linux image in the Ubuntu 22.04 system of the development board, please ensure proper heat dissipation. If the heat dissipation is not done well, the file system may run away easily.

4.1.1. Compile using the Ubuntu 22.04 system of the development board

1) Linux SDK, namely **orangepi-build**, supports running on **Ubuntu 22.04** of the development board (other systems have not been tested), so before downloading orangepi-build, please first make sure that the Ubuntu version installed on the development board is Ubuntu 22.04. The command to check the Ubuntu version installed on the development board is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please change the system before performing the following operations.

```
orangepi@orangepi:~$ lsb_release -a
No LSB modules are available.
```

```
Distributor ID:   Ubuntu
Description:      Ubuntu 22.04.1 LTS
Release:          22.04
Codename:         jammy
```

2) Since the source codes of kernel and U-boot are stored on GitHub, it is very important to ensure that the development board can download the code from GitHub normally when compiling the image.

4.1.2. Compile using Ubuntu 22.04 x64 computer

1) Linux SDK, namely **orangepi-build**, supports running on computers with **Ubuntu 22.04** installed, so before downloading orangepi-build, please first make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on the computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please change the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID:   Ubuntu
Description:      Ubuntu 22.04 LTS
Release:          22.04
Codename:         jammy
```

2) If your computer is running Windows and you don't have Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in Windows. But please note that you should not compile orangepi-build on a WSL virtual machine, because orangepi-build has not been tested in a WSL virtual machine, so it is not guaranteed that orangepi-build can be used normally in WSL.

3) The installation image download address of Ubuntu 22.04 **amd64** version is:

```
https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04.3-desktop-amd64.iso
Or
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.3-desktop-amd64.iso
```

4) After installing Ubuntu 22.04 on your computer or in a virtual machine, please set the software source of Ubuntu 22.04 to Tsinghua source first, otherwise it is easy to get errors

due to network reasons when installing the software later.

a. For the method of replacing Tsinghua source, please refer to the instructions on this webpage.

https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/

b. Note that the Ubuntu version needs to be switched to 22.04.

Ubuntu 镜像使用帮助

Ubuntu 的软件源配置文件是 /etc/apt/sources.list。将系统自带的该文件做个备份，将该文件替换为下面内容，即可使用 TUNA 的软件源镜像。

选择你的ubuntu版本: 22.04 LTS

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

c. The content of the **/etc/apt/sources.list** file that needs to be replaced is:

```
test@test:~$ sudo mv /etc/apt/sources.list /etc/apt/sources.list.bak
test@test:~$ sudo vim /etc/apt/sources.list
# The source mirror is commented out by default to increase the speed of apt update. You can uncomment it if necessary.
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# Pre-release software source, not recommended to enable
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-proposed main restricted universe multiverse
```

d. After the replacement, you need to update the package information and ensure that there are no errors.

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

e. **In addition, since the source codes of the kernel and U-boot are stored on GitHub, it is very important to ensure that the computer can download the code from GitHub normally when compiling the image.**

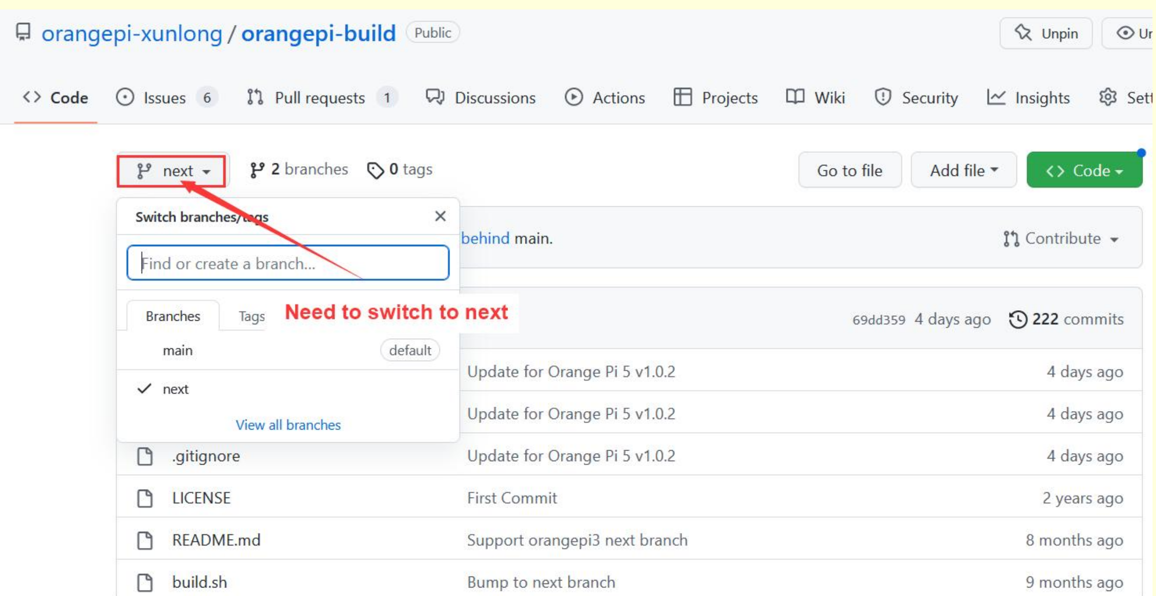
4. 2. Get the source code of Linux SDK

4. 2. 1. Download orangepi-build from github

1) Linux SDK actually refers to the orangepi-build code. Orangepi-build is modified based on the armbian build compilation system. Orangepi-build can be used to compile multiple versions of Linux images. First download the orangepi-build code. The command is as follows:

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

Note that the Orange Pi CM5 Base development board needs to download the next branch source code of orangepi-build. The above git clone command needs to specify the branch of orangepi-build source code as next.



You do not need to enter the username and password of the GitHub account

when downloading the orangepi-build code through the git clone command (the same applies to downloading other codes in this manual). If the Ubuntu PC prompts you to enter the username and password of the GitHub account after entering the git clone command, it is usually because the address of the orangepi-build warehouse after git clone is entered incorrectly. Please check the command spelling carefully for any errors, instead of thinking that we forgot to provide the username and password of the GitHub account here.

2) The u-boot and linux kernel versions currently used by the development board are as follows:

Branches	u-boot version	Linux kernel version
legacy	u-boot 2017.09	linux5.10
current	u-boot 2017.09	linux6.1

The branch mentioned here is not the same as the branch of orangepi-build source code, please do not confuse them. This branch is mainly used to distinguish different kernel source code versions.

Currently, we define the linux5.10 bsp kernel provided by RK as the legacy branch, and the linux6.1 bsp kernel as the current branch.

3) After downloading orangepi-build, it will contain the following files and folders.

a. **build.sh**: Compile the startup script.
b. **external**: Contains configuration files, specific scripts, and source code of some programs needed to compile the image.
c. **LICENSE**: GPL 2 license file.
d. README.md: orangepi-build documentation.
e. **scripts**: Generic script for compiling linux images.

```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

If you download the orangepi-build code from github, you may find that orangepi-build does not contain the source code of u-boot and linux kernel, nor the cross-compilation toolchain required to compile u-boot and linux kernel. This is normal because these things are stored in other separate github repositories or some servers (the addresses will be detailed below). orangepi-build will specify the

addresses of u-boot, linux kernel and cross-compilation toolchain in the script and configuration file. When running orangepi-build, if it finds that these things are not available locally, it will automatically download them from the corresponding places.

4. 2. 2. Download the cross-compilation toolchain

The cross-compilation toolchain will only be downloaded when you compile the image using orangepi-build on an x64 computer. Compiling the Linux image of the development board in Ubuntu 22.04 on the development board will not download the cross-compilation toolchain, and orangepi-build/toolchains will be an empty folder.

1) When orangepi-build is run for the first time, it will automatically download the cross-compilation toolchain and put it in the **toolchains** folder. Each time you run the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists. If not, it will restart the download. If it exists, it will be used directly without repeated downloading.

```
[ o.k. ] Checking for external GCC compilers
[ .... ] downloading using http(s) network [ gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz ]
[#8d7029 16MiB/24MiB(65%) CN:1 DL:7.9MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz: 24.9MiB [14.4MiB/s] [=============================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz ]
[#e30eec 17MiB/33MiB(50%) CN:1 DL:10MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz: 33.9MiB [9.66MiB/s] [=============================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux.tar.xz ]
[#041c24 48MiB/48MiB(99%) CN:1 DL:2.7MiB]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux.tar.xz: 48.8MiB [13.0MiB/s] [=============================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz ]
[#3dee3e 72MiB/76MiB(93%) CN:1 DL:3.7MiB ETA:1s]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz: 77.0MiB [14.2MiB/s] [=============================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz ]
[#42e728 104MiB/104MiB(99%) CN:1 DL:2.8MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz:  104MiB [13.9MiB/s] [=============================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz ]
[#2c065e 108MiB/111MiB(97%) CN:1 DL:3.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz:  111MiB [13.4MiB/s] [=============================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf.tar.xz ]
[#d232ee 250MiB/251MiB(99%) CN:1 DL:2.0MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf.tar.xz:  251MiB [13.7MiB/s] [=============================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu.tar.xz ]
[#88b441 268MiB/269MiB(99%) CN:1 DL:0.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
```

2) The mirror website of the cross-compilation tool chain in China is the open source software mirror site of Tsinghua University.

https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/

3) After downloading **toolchains**, it will contain multiple versions of cross-compilation toolchains, and the development board will only use two of them.

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
gcc-arm-11.2-2022.02-x86_64-arm-none-linux-gnueabihf
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

4) The cross-compilation tool chain used to compile the Linux kernel source code is:

a. linux5.10：

```
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
```

b. Linux6.1：

```
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
```

5) The cross-compilation tool chain used to compile the u-boot source code is:

a. v2017.09：

```
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
```

4. 2. 3. orangepi-build complete directory structure description

1) After downloading the orangepi-build repository, it does not contain the source code of the Linux kernel, u-boot, and the cross-compilation toolchain. The source code of the Linux kernel and u-boot are stored in independent git repositories.

a. The git repository where the Linux 5.10 kernel source code is stored is as follows:

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rk35xx
```

b. The git repository where the Linux 6.1 kernel source code is stored is as follows:

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-6.1-rk35xx
```

c. The git repository where the u-boot source code is stored is as follows:

```
https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2017.09-rk3588
```

2) When orangepi-build is run for the first time, it will download the cross-compilation

toolchain, u-boot and Linux kernel source code. After successfully compiling a Linux image, the files and folders that can be seen in orangepi-build are:

a. **build.sh**: Compile the startup script.
b. **external**: Contains configuration files needed for compiling images, scripts for specific functions, and source code for some programs. The rootfs compressed package cached during the image compilation process is also stored in external.
c. **kernel**: The source code of the Linux kernel is stored in the folder named **orange-pi-5.10-rk35xx**, which stores the kernel source code of the legacy branch of the RK3588/RK3588S series development board. The folder named **orange-pi-6.1-rk35xx** stores the kernel source code of the current branch of the RK3588/RK3588S series development board. Please do not manually modify the name of the kernel source code folder. If modified, the kernel source code will be re-downloaded when the compilation system is running.
d. **LICENSE**: GPL 2 license file.
e. **README.md**: orangepi-build documentation.
f. **output**: Stores compiled u-boot, linux and other deb packages, compilation logs, compiled images and other files.
g. **scripts**: Generic script for compiling linux images.
h. **toolchains**: Store the cross-compilation tool chain.
i. **u-boot**: The u-boot source code is stored in the folder named **v2017.09-rk3588**, which stores the u-boot source code of the RK3588/RK3588S series development board. Please do not manually modify the name of the u-boot source code folder. If modified, the u-boot source code will be downloaded again when the compilation system is running.
j. **userpatches**: Stores configuration files needed to compile scripts.

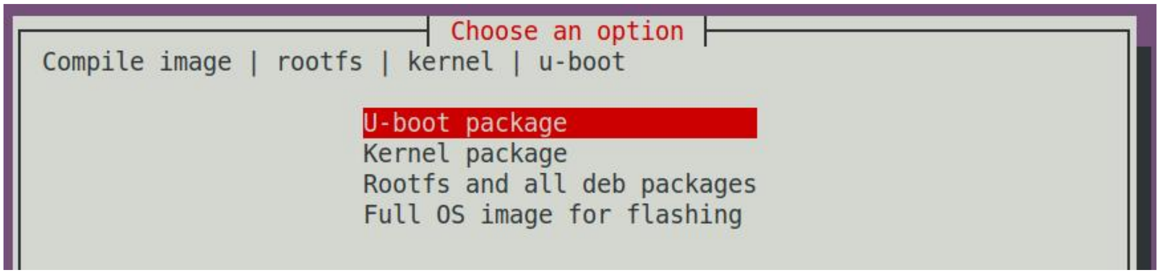
```
test@test:~/orangepi-build$ ls
build.sh external kernel LICENSE output README.md scripts toolchains
u-boot userpatches
```

4. 3. Compile u-boot

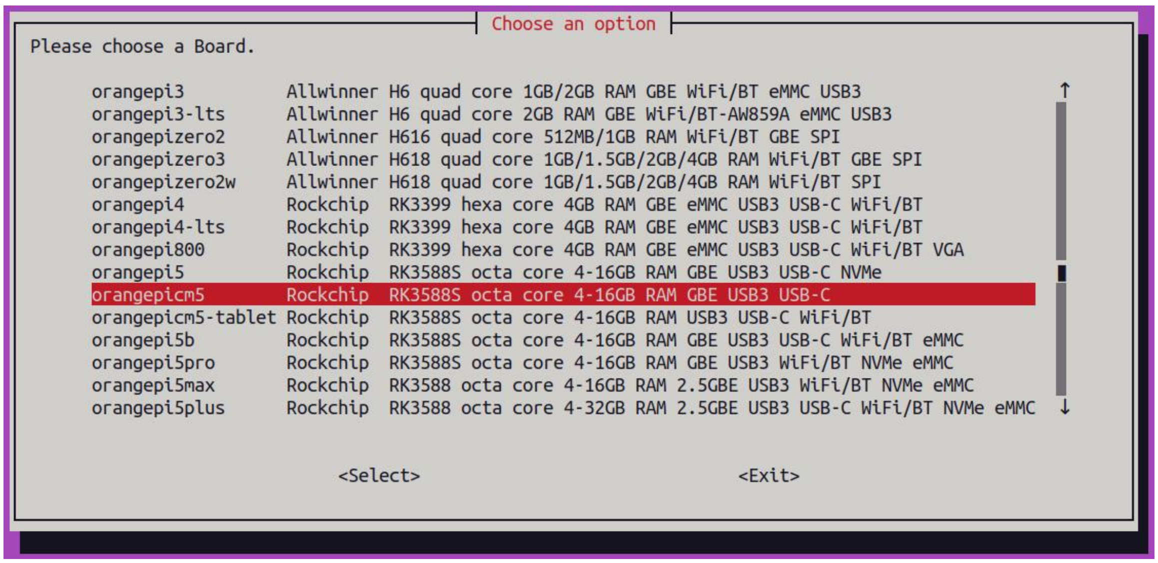
1) Run the build.sh script and remember to add sudo permissions.

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package** and press Enter.



3) Then select the model of the development board.



4) Then u-boot will start to compile. Some of the information prompted during compilation is as follows:

a. u-boot source code version.

[o.k.] Compiling u-boot [**v2017.09**]

b. Version of the cross-compilation toolchain.

[o.k.] Compiler version [**aarch64-linux-gnu-gcc 7.4.1**]

c. The path of the compiled u-boot deb package.

[o.k.] Target directory [**orangepi-build/output/debs/u-boot**]

d. The package name of the u-boot deb package generated by compilation.

[o.k.] File name [**linux-u-boot-legacy-orangepicm5_1.0.0_arm64.deb**]

e. The time used for compilation.

[o.k.] Runtime [**1 min**]

f. Repeat the command to compile u-boot. Use the following command to start compiling u-boot directly without selecting through the graphical interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh BOARD=orangepicm5 BRANCH=legacy BUILD_OPT=u-boot KERNEL_CONFIGURE=no ]
```

5) Check the compiled u-boot deb package.

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-legacy-orangepicm5_1.0.0_arm64.deb
```

6) The files contained in the generated u-boot deb package are as follows:

a. Use the following command to decompress the deb package.

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ $ dpkg -x \
linux-u-boot-legacy-orangepicm5_1.0.0_arm64.deb .    (Note that there is a "." at the end of the command.)
test@test:~/orangepi_build/output/debs/u-boot$ ls
linux-u-boot-legacy-orangepicm5_1.0.0_arm64.deb    usr
```

b. The decompressed files are as follows:

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-legacy-orangepicm5_1.0.0_arm64
    │   ├── idbloader.img
    │   ├── rkspi_loader.img
    │   └── u-boot.itb
    └── u-boot
        ├── LICENSE
        ├── orangepi_5_defconfig
        └── platform_install.sh

3 directories, 6 files
```

7) When the orangepi-bulid compilation system compiles the u-boot source code, it will first synchronize the u-boot source code with the u-boot source code on the GitHub server. So if you want to modify the u-boot source code, you first need to turn off the source code download and update function (**you need to fully compile u-boot once before turning off this function, otherwise it will prompt that the u-boot source code cannot**

be found. If the source code compression package is downloaded from Baidu Cloud Disk, there is no such problem because the u-boot source code has been cached), otherwise the changes made will be restored. The method is as follows:

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes".

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

8) When debugging the u-boot code, you can use the following method to update the u-boot in the Linux image for testing.

a. Upload the compiled u-boot deb package to the Linux system of the development board.

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-legacy-orangepicm5_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the installed u-boot deb package.

```
root@orangepi:~# apt purge -y linux-u-boot-orangepicm5-legacy
```

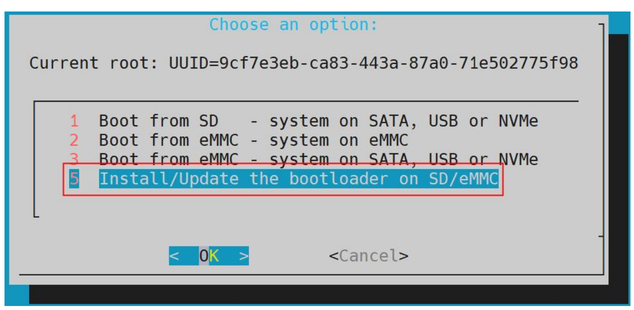
c. Install the new u-boot deb package just uploaded.

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepicm5_1.0.0_arm64.deb
```

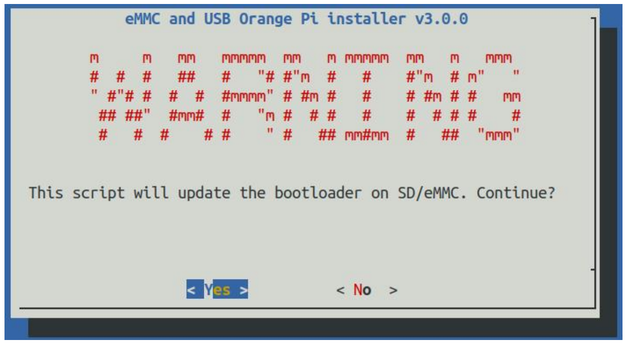
d. Then run the nand-sata-install script.

```
root@orangepi:~# nand-sata-install
```

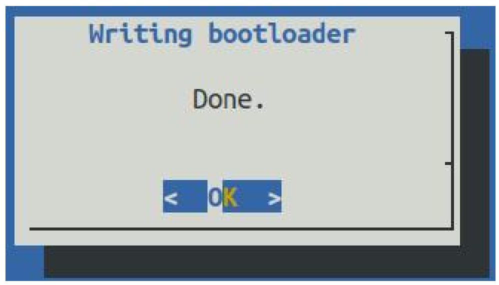
e. Then select **5 Install/Update the bootloader on SD/eMM** to update the u-boot in the TF card or eMMC.



f. After pressing the Enter key, a Warning will pop up first.



g. Press the Enter key again to start updating u-boot. After the update, the following information will be displayed.



h. Then you can restart the development board to test whether the u-boot changes are effective.

9) Other useful information.

a. In the u-boot 2017.09 source code, the defconfig configuration file used by the development board is.

```
orangepi-build/u-boot/v2017.09-rk3588/configs/orangepi_cm5_defconfig
```

b. In the u-boot 2017.09 source code, the development board uses the dts file.

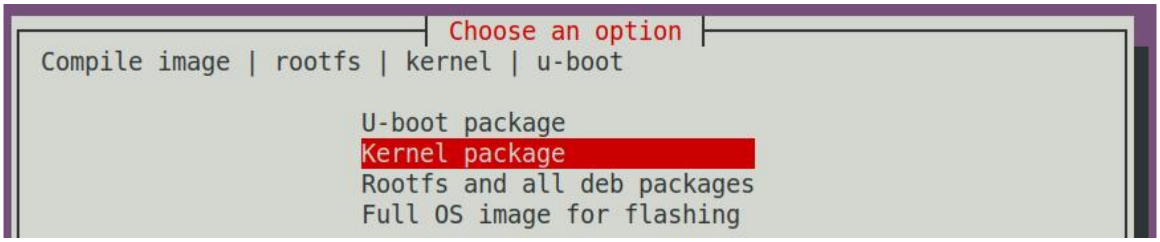
```
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3588s-orangepi-cm5.dts
```

4. 4. Compile the Linux kernel

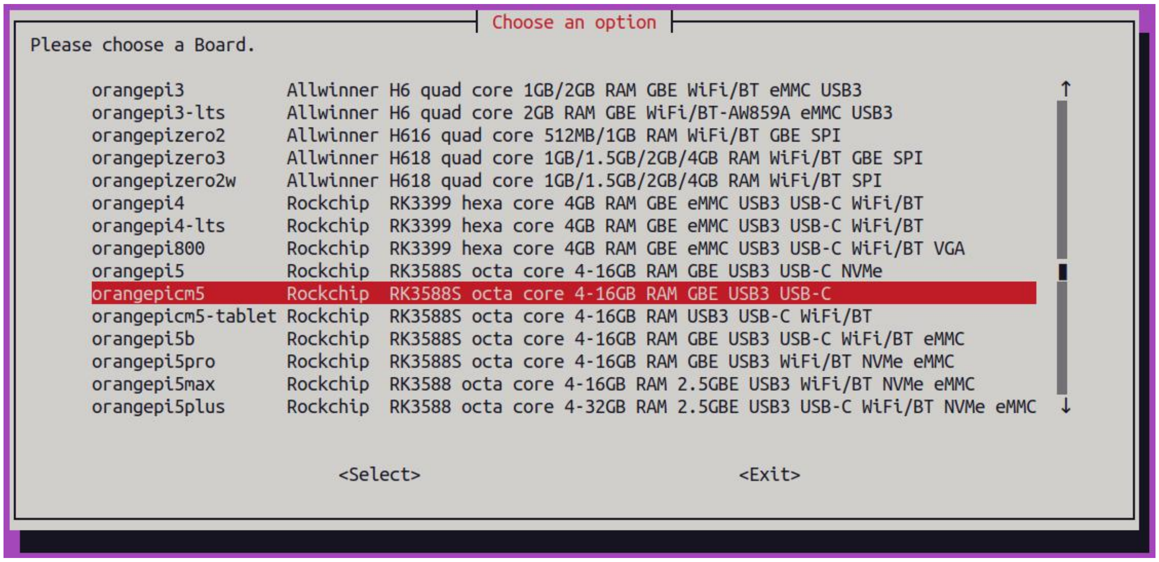
1) Run the build.sh script and remember to add sudo permissions.

```
test@test:~/orangepi-build$ sudo ./build.sh
```

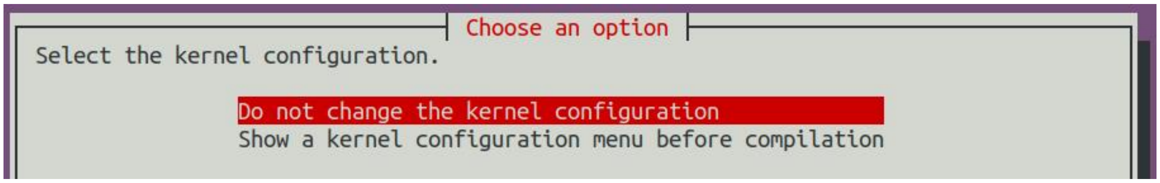
2) Select **Kernel package** and press Enter.



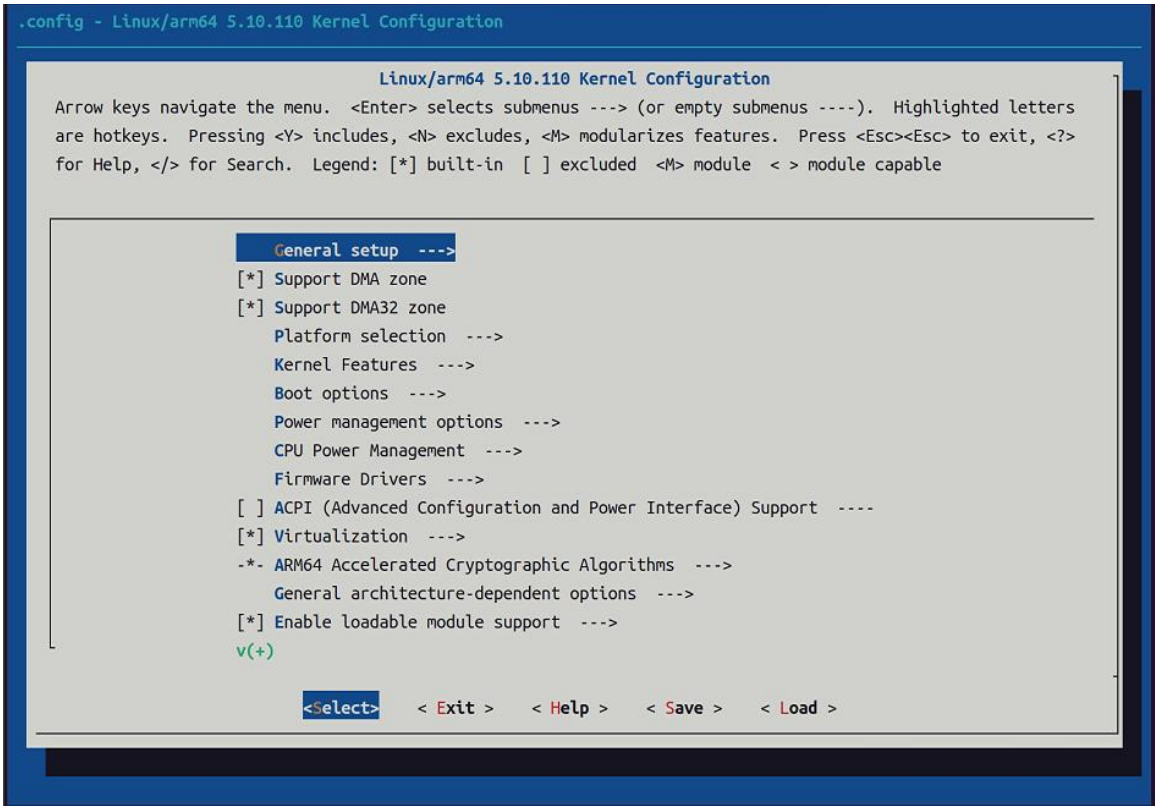
3) Then select the model of the development board.



4) You will then be prompted whether you need to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first option. If you need to modify the kernel configuration, select the second option.



5) If you selected to display the kernel configuration menu (the second option) in step 4), the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration. After the modification, save and exit. After exiting, the kernel source code will start to compile.



a. If you do not need to modify the kernel configuration options, when running the build.sh script, pass **KERNEL_CONFIGURE=no** to temporarily block the kernel configuration interface from popping up.

```
test@test:~/orangepi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

b. You can also set **KERNEL_CONFIGURE=no** in the **orangepi-build/userpatches/config-default.conf** configuration file to permanently disable this feature.

c. If the following error message appears when compiling the kernel, it is because the terminal interface of the Ubuntu PC is too small, causing the **make menuconfig** interface to fail to display. Please adjust the terminal of the Ubuntu PC to the maximum size and then re-run the build.sh script.

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

6) Some of the information prompted when compiling the kernel source code is as follows:

a. The version of the Linux kernel source code.

[o.k.] Compiling current kernel [**5.10.160**]

b. The version of the cross-compilation toolchain used.

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 11.2.1**]

c. The default configuration file used by the kernel and the path where it is stored.

[o.k.] Using kernel config file [**config/kernel/linux-rockchip-rk3588-legacy.config**]

d. The path of the kernel-related deb package generated by compilation.

[o.k.] Target directory [**orangepi-build/output/debs/**]

e. The package name of the compiled kernel image deb package.

[o.k.] File name [**linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb**]

f. The time used for compilation.

[o.k.] Runtime [**5 min**]

g. Finally, the compilation command for the last selected kernel will be displayed. Use the following command to directly start compiling the kernel source code without selecting through the graphical interface.

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepicm5 BRANCH=legacy BUILD_OPT=kernel KERNEL_CONFIGURE=no**]

7) Check the compiled kernel-related deb packages.

a. **linux-dtb-legacy-rockchip-rk3588_1.0.0_arm64.deb** contains dtb files used by the kernel.

b. **linux-headers-legacy-rockchip-rk3588_1.0.0_arm64.deb** contains kernel header files.

c. **linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb** contains kernel images and kernel modules.

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-legacy-rockchip-rk3588_1.0.0_arm64.deb
output/debs/linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb
output/debs/linux-headers-legacy-rockchip-rk3588_1.0.0_arm64.deb
```

8) The files contained in the generated linux-image deb package are as follows:

a. Use the following command to decompress the deb package.

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi_build/output/debs$ mkdir test
test@test:~/orangepi_build/output/debs$ cp \
linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb test/
test@test:~/orangepi_build/output/debs$ cd test
test@test:~/orangepi_build/output/debs/test$ dpkg -x \
linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb .
test@test:~/orangepi_build/output/debs/test$ ls
boot   etc   lib   linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb   usr
```

b. The decompressed files are as follows:

```
test@test:~/orangepi-build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.160-rockchip-rk3588
│   ├── System.map-5.10.160-rockchip-rk3588
│   └── vmlinuz-5.10.160-rockchip-rk3588
├── etc
│   └── kernel
├── lib
│   └── modules
├── linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb
└── usr
    ├── lib
    └── share
```

9) When the orangepi-bulid compilation system compiles the Linux kernel source code,

it will first synchronize the Linux kernel source code with the Linux kernel source code on the GitHub server. So if you want to modify the Linux kernel source code, you first need to turn off the source code update function (**you need to fully compile the Linux kernel source code once before turning off this function, otherwise it will prompt that the Linux kernel source code cannot be found. If the source code compression package is downloaded from Baidu Cloud Disk, there will be no such problem because the Linux source code has been cached**), otherwise the changes made will be restored. The method is as follows:

Set the IGNORE_UPDATES variable in **userpatches/config-default.conf** to "yes".

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If you have made changes to the kernel, you can use the following method to update the kernel and kernel modules of the Linux system on the development board.

a. Upload the compiled Linux kernel deb package to the Linux system of the development board.

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the installed linux kernel deb package.

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk3588
```

c. Install the new Linux kernel deb package just uploaded.

```
root@orangepi:~# dpkg -i linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb
```

d. Then restart the development board and check whether the kernel-related modifications have taken effect.

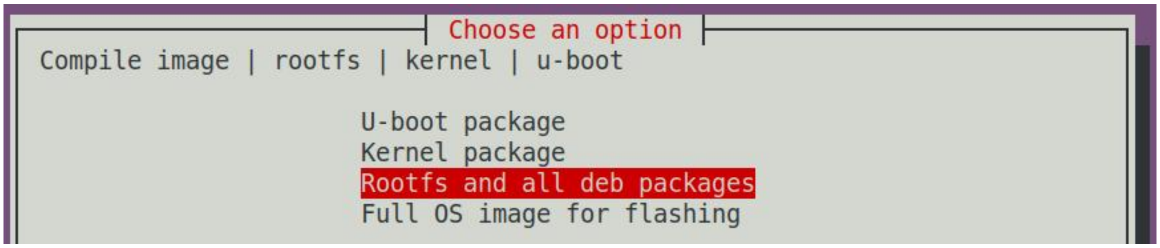
```
root@orangepi:~# reboot
```

4. 5. Compile rootfs

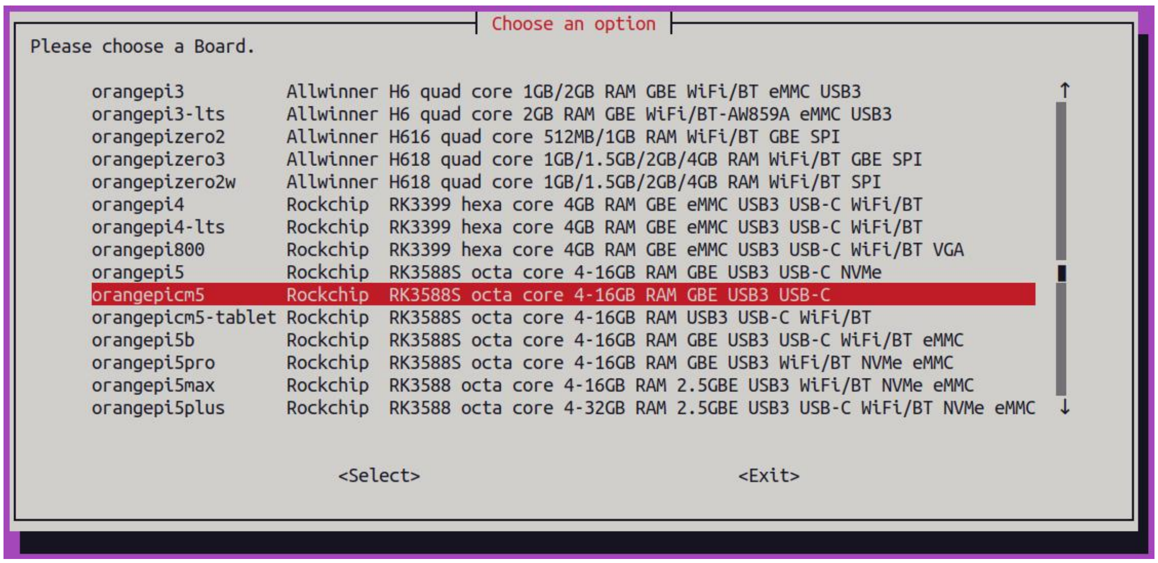
1) Run the build.sh script and remember to add sudo permissions.

```
test@test:~/orangepi-build$ sudo ./build.sh
```

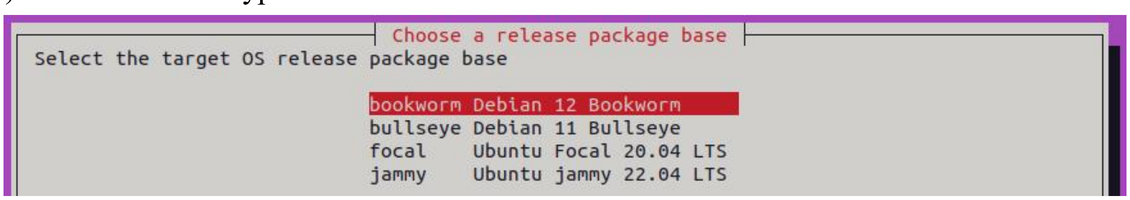
2) Select **Rootfs and all deb packages** and press Enter.



3) Then select the model of the development board.



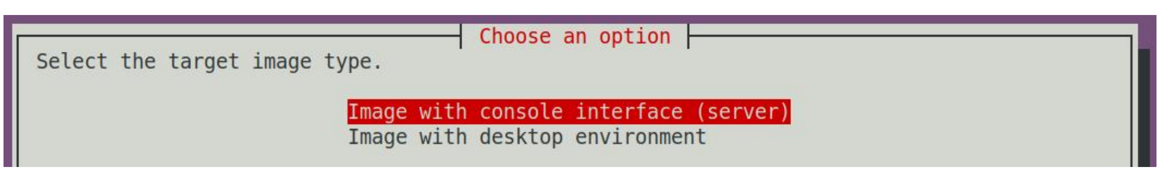
4) Then select the type of rootfs.



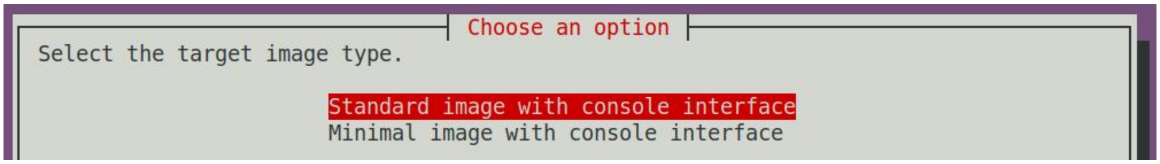
5) Then select the image type.

a. **Image with console interface (server)** Indicates the server version image, which is relatively small in size.

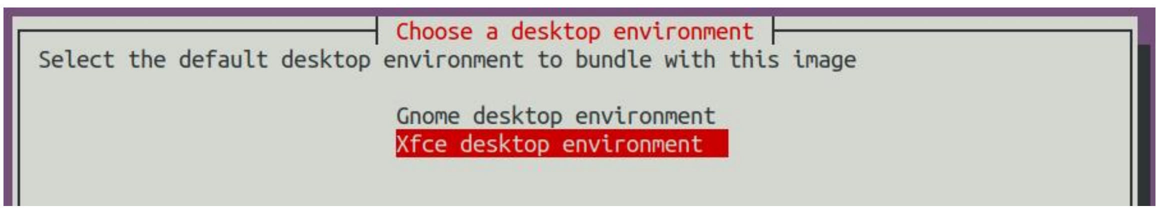
b. **Image with desktop environment** Indicates an image with a desktop, which is relatively large in size.

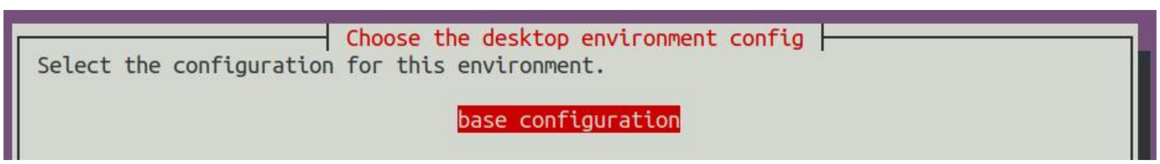


6) If you are compiling a server version image, you can also choose to compile the Standard version or the Minimal version. The Minimal version has much less pre-installed software than the Standard version (**please do not choose the Minimal version if you do not have special needs, because many things are not pre-installed by default and some functions may not be used**).

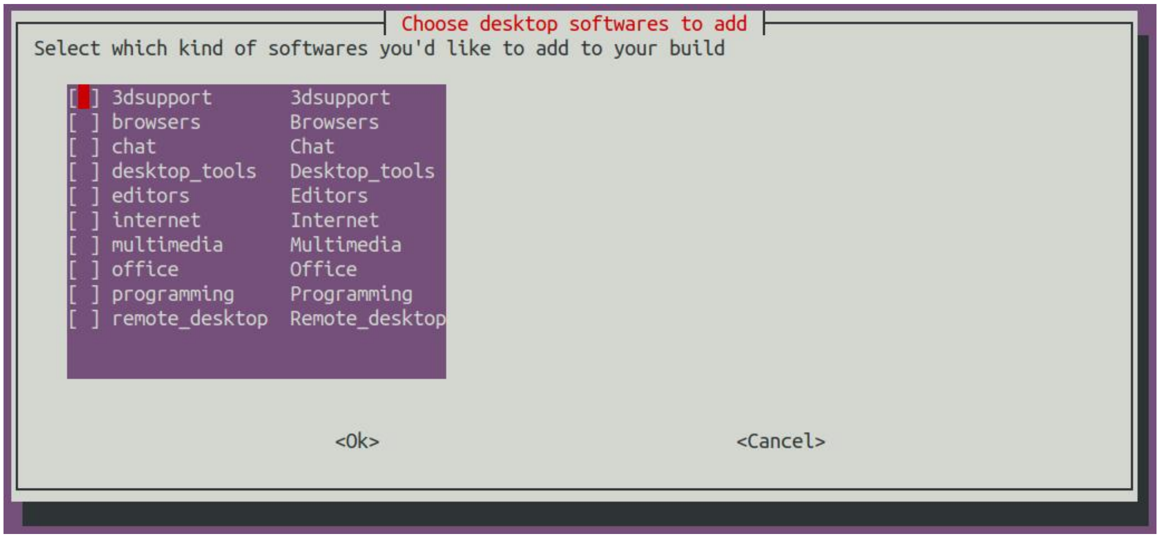


7) If you are compiling a desktop version image, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktop, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop.





You can then select additional packages to install. Please press Enter to skip this step.



8) Then the rootfs will start to be compiled. Some of the information prompted during the compilation is as follows:

a. Type of rootfs.

```
[ o.k. ] local not found [ Creating new rootfs cache for jammy]
```

b. The storage path of the compiled rootfs compressed package.

```
[ o.k. ] Target directory [ external/cache/rootfs ]
```

c. The name of the rootfs compressed package generated by compilation.

```
[ o.k. ] File name [ jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4 ]
```

d. The time used for compilation.

```
[ o.k. ] Runtime [ 13 min ]
```

9) Check the rootfs compressed package generated by compilation.

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is the compressed package of rootfs. The meanings of the fields in the name are:

a) **Jammy** indicates the type of Linux distribution of rootfs.

b) **Xfce** indicates that the rootfs is a desktop version, and **cli** indicates a server version.

c) **arm64** indicates the architecture type of rootfs.

d) **f930ff6ebbac1a72108a2e100762b18f** is the MD5 hash value generated by the package names of all packages installed by rootfs. As long as the list of packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to determine whether rootfs needs to be recompiled.

b. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list** lists the package names of all software packages installed by rootfs.

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.current
jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list
```

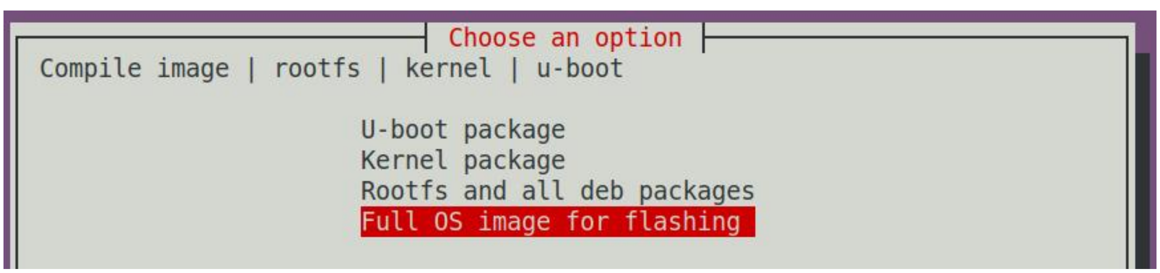
10) If the required rootfs already exists in **external/cache/rootfs**, then compiling rootfs again will skip the compilation process directly and will not restart the compilation. When compiling the image, it will also check whether there is a cached rootfs available in **external/cache/rootfs**. If there is, it will be used directly, which can save a lot of download and compilation time.

4. 6. Compile Linux image

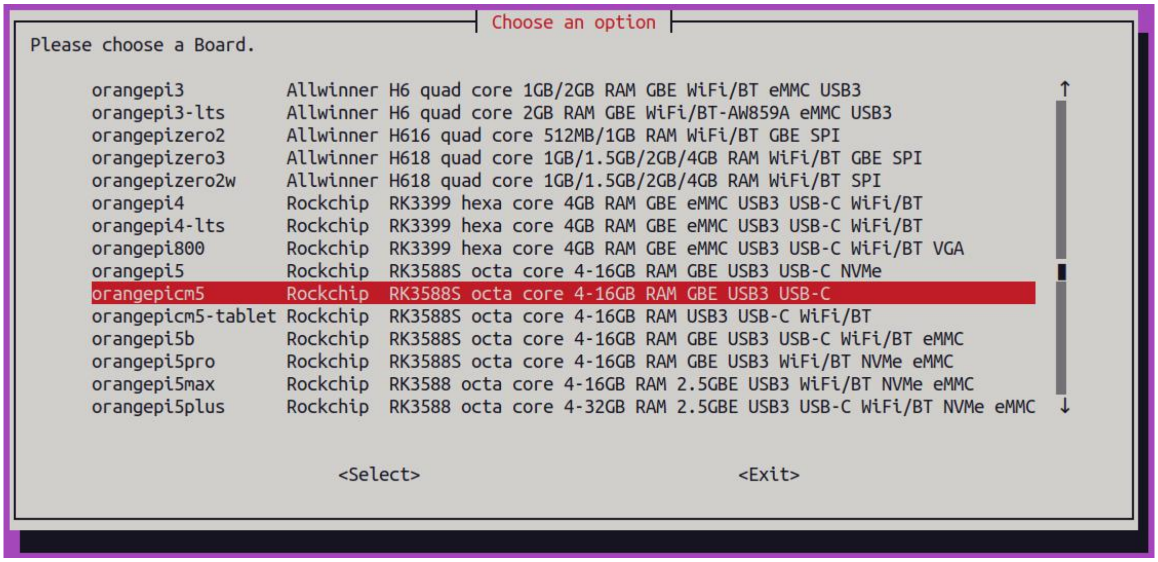
1) Run the build.sh script and remember to add sudo permissions.

```
test@test:~/orangepi-build$ sudo ./build.sh
```

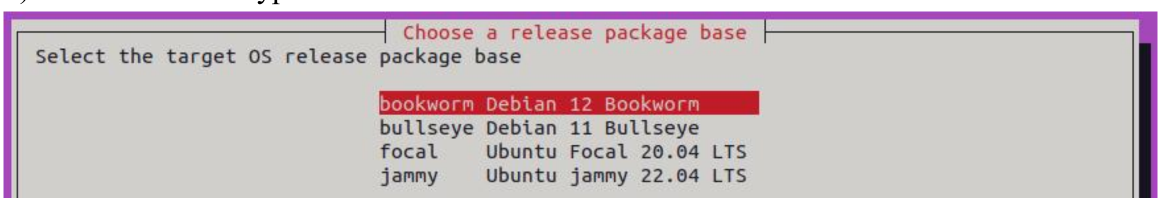
2) Select **Full OS image for flashing** and press Enter.



3) Then select the model of the development board.



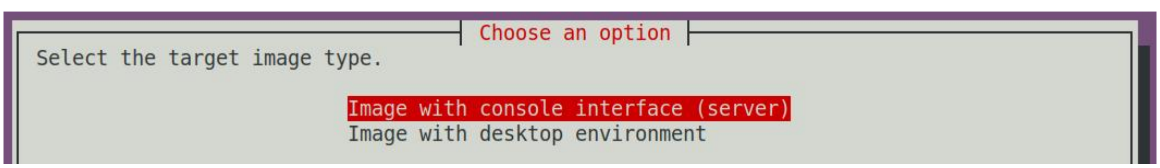
4) Then select the type of rootfs.



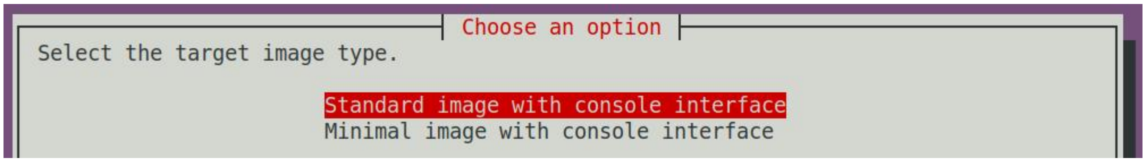
5) Then select the image type.

a. **Image with console interface (server)** Indicates the server version image, which is relatively small in size.

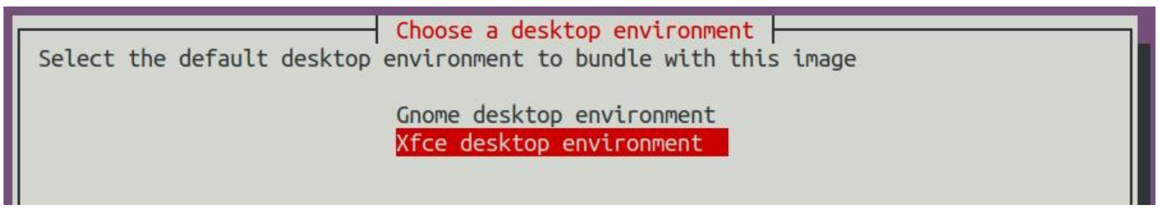
b. **Image with desktop environment** Indicates an image with a desktop, which is relatively large in size.

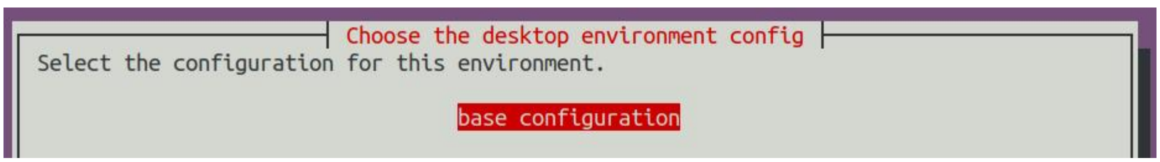


6) If you are compiling a server version image, you can also choose to compile the Standard version or the Minimal version. The Minimal version has much less pre-installed software than the Standard version (**please do not choose the Minimal version if you do not have special needs, because many things are not pre-installed by default and some functions may not be used**).

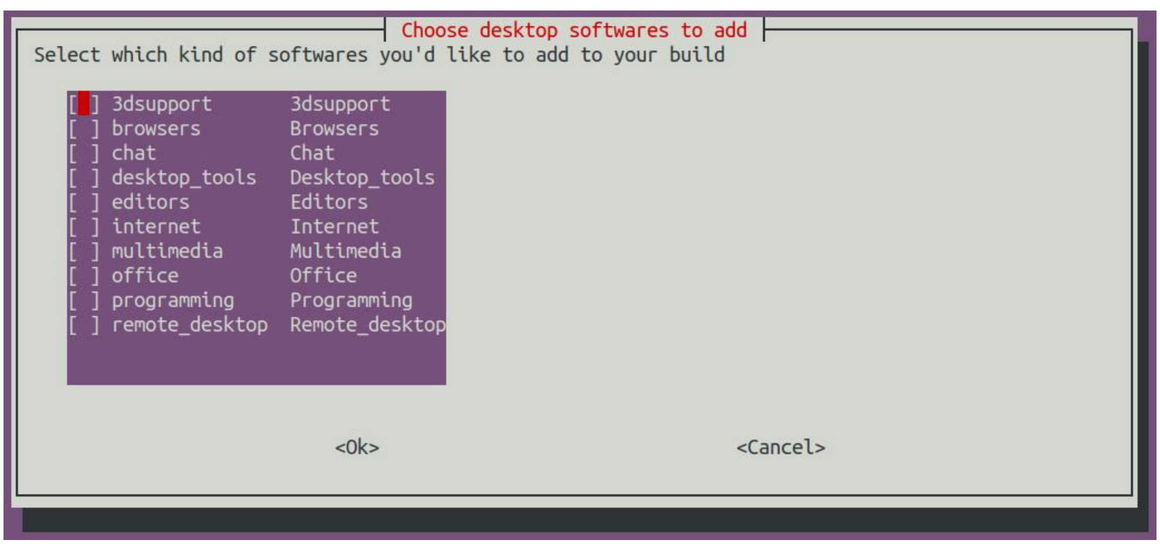


7) If you are compiling a desktop version image, you also need to select the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktop, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktop.





You can then select additional packages to install. Please press Enter to skip this step.



8) Then the Linux image will be compiled. The general process of compilation is as follows:

a. Initialize the compilation environment of Ubuntu PC and install the software packages required for the compilation process.

b. Download the source code of u-boot and linux kernel (if it has been cached, only update the code).

c. Compile the u-boot source code and generate the deb package of u-boot.

d. Compile the linux source code and generate the deb package related to linux.

e. Make a deb package for linux firmware.

f. Make a deb package for the orangepi-config tool.

g. Make a deb package for board-level support.

h. If you compile a desktop version image, you will also make a desktop-related deb package.

i. Check whether the rootfs has been cached. If not, remake the rootfs. If it has been cached, directly decompress it and use it.

j. Install the deb package generated earlier to the rootfs.

k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.

l. Then make an image file and format the partition. The default type is ext4.

m. Then copy the configured rootfs to the partition of the image.

n. Then update initramfs.

o. Finally, write the u-boot bin file into the image using the dd command.

9) After compiling the image, the following message will be displayed:

a. The storage path of the compiled image.

```
[ o.k. ] Done building
[ output/images/orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160/orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img ]
```

b. The time used for compilation.

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the command to compile the image. Use the following command to start compiling the image directly without selecting through the graphical interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh    BOARD=orangepicm5 BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

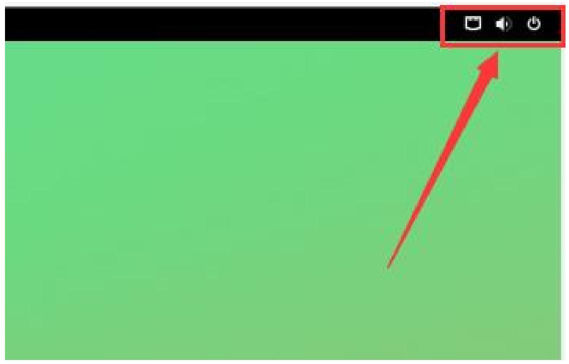
5. Instructions for using Orange Pi OS Arch system

5. 1. Adaptation of Orange Pi OS Arch system

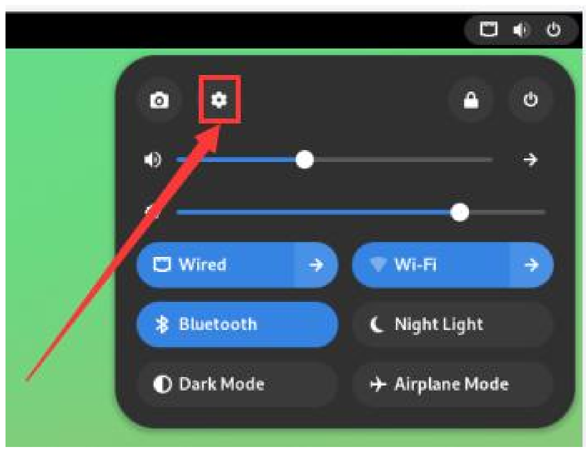
Function	OPi OS Arch Gnome Wayland
HDMI display	OK
HDMI audio frequency	OK
USB 2.0	OK
USB 3.0	OK
Gigabit Ethernet port	OK
2.5GEthernet port x 2	OK
Debug UART	OK
FAN	OK
RTC	OK
eMMC	OK
GPIO（12pin）	OK
UART（12pin）	OK
SPI（12pin）	OK
I2C（12pin）	OK
CAN（12pin）	OK
Camera x 4	OK
LED lamp	OK
TF card startup	OK
infra-red	OK
GPU	OK
NPU	NO
VPU	OK
On/Off button	OK
Watchdog test	OK

5. 2. Methods for setting up the English environment and installing Chinese input methods

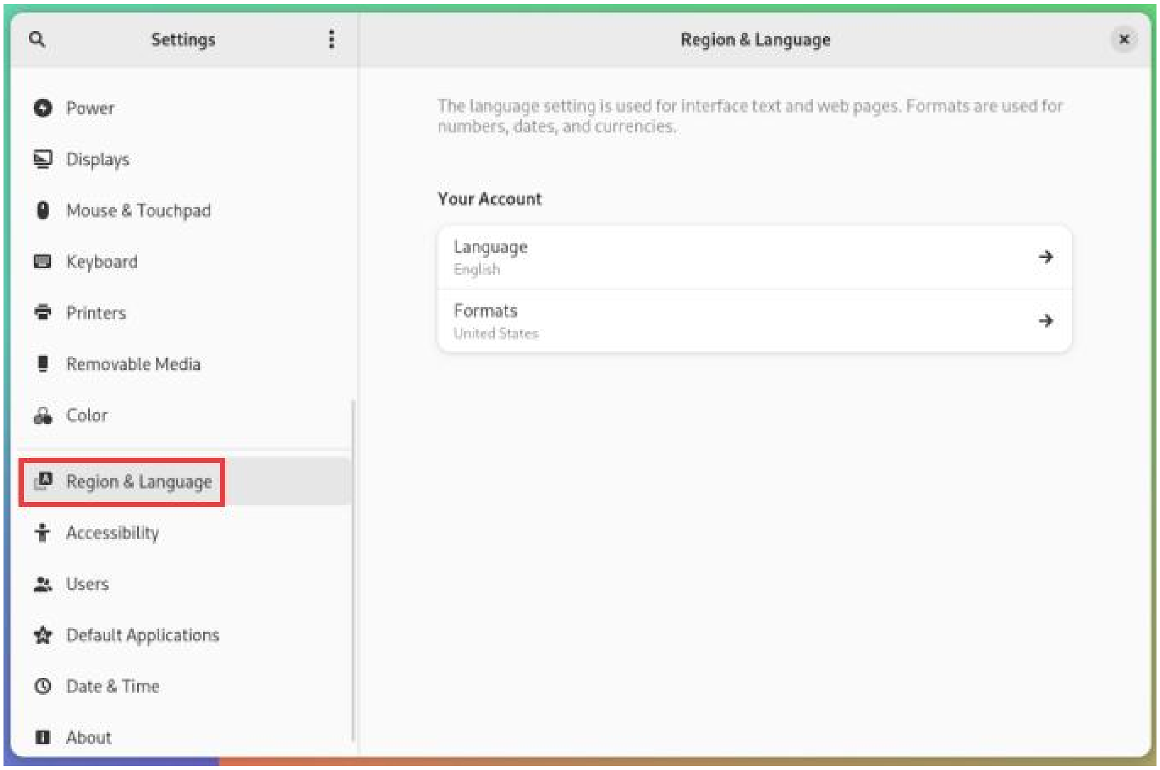
1) First, click on the area in the upper right corner of the desktop.



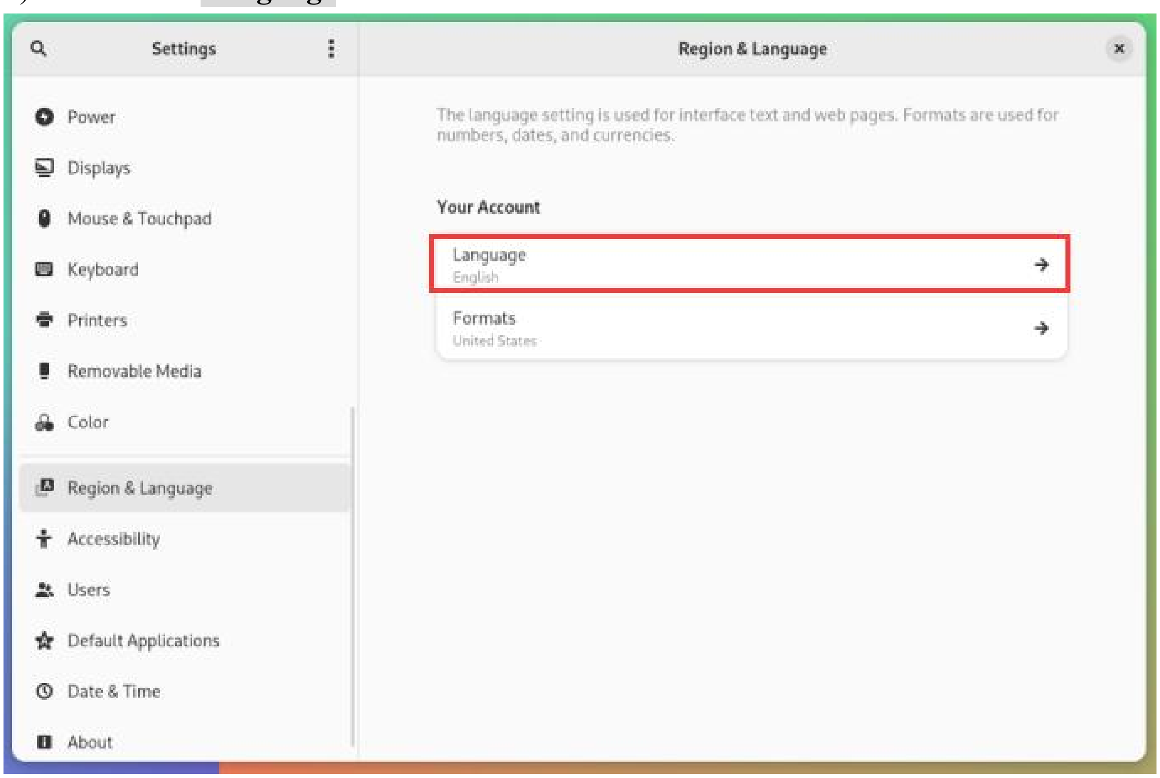
2) Then open the settings.



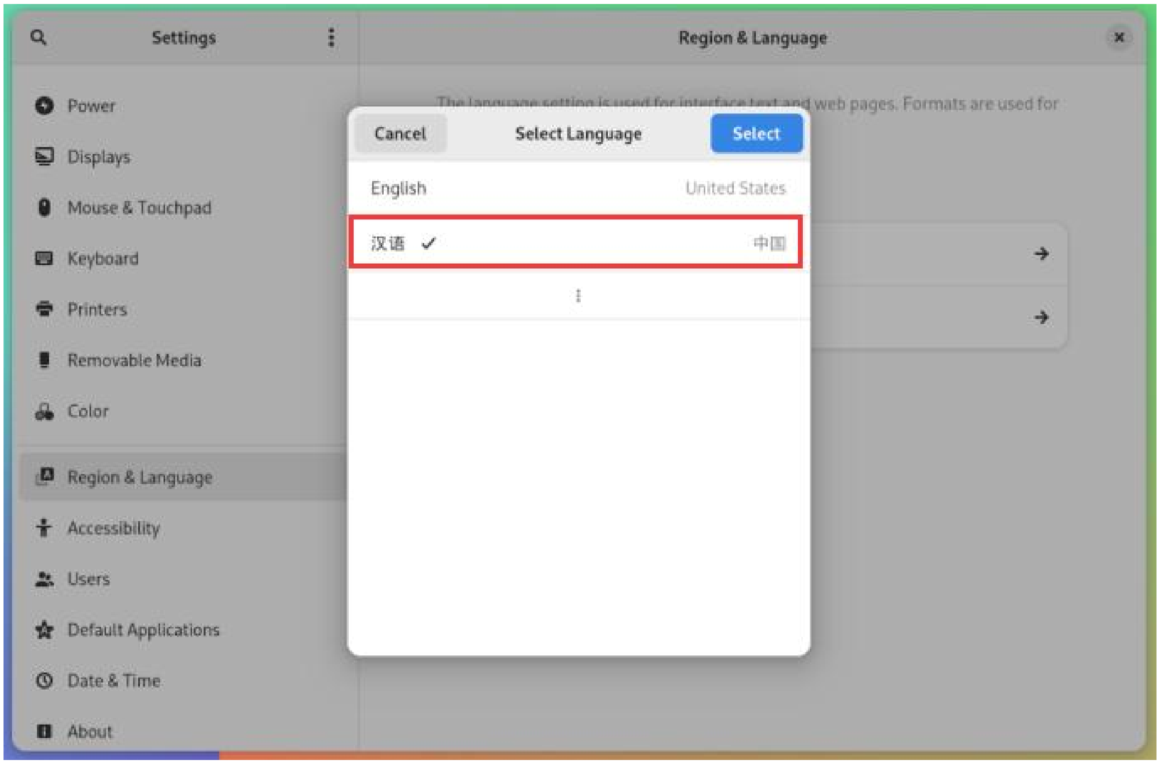
3) Then find the **Region & Language** option.



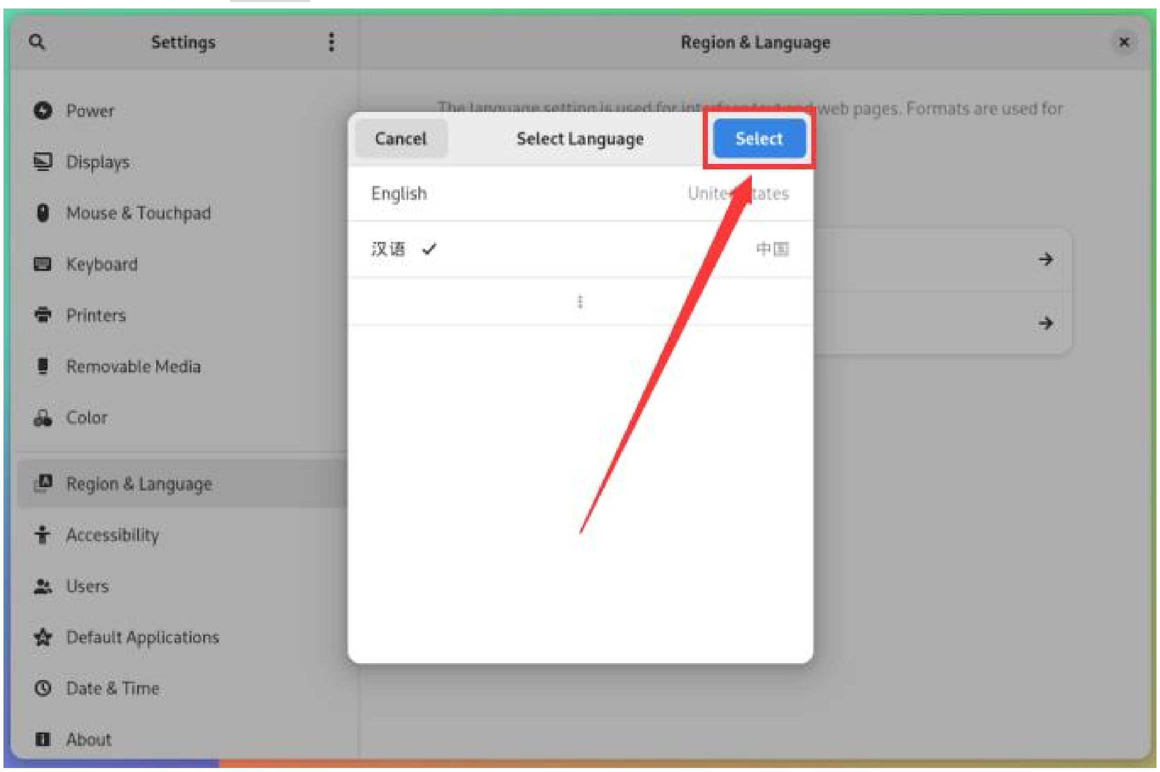
4) Then select **Language**。



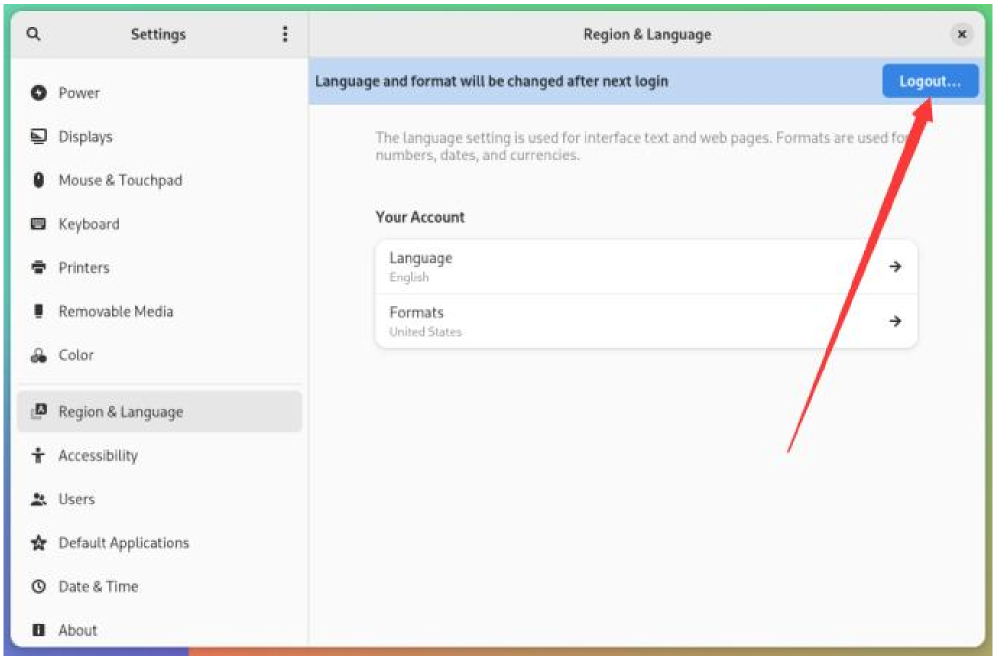
5) Then choose English



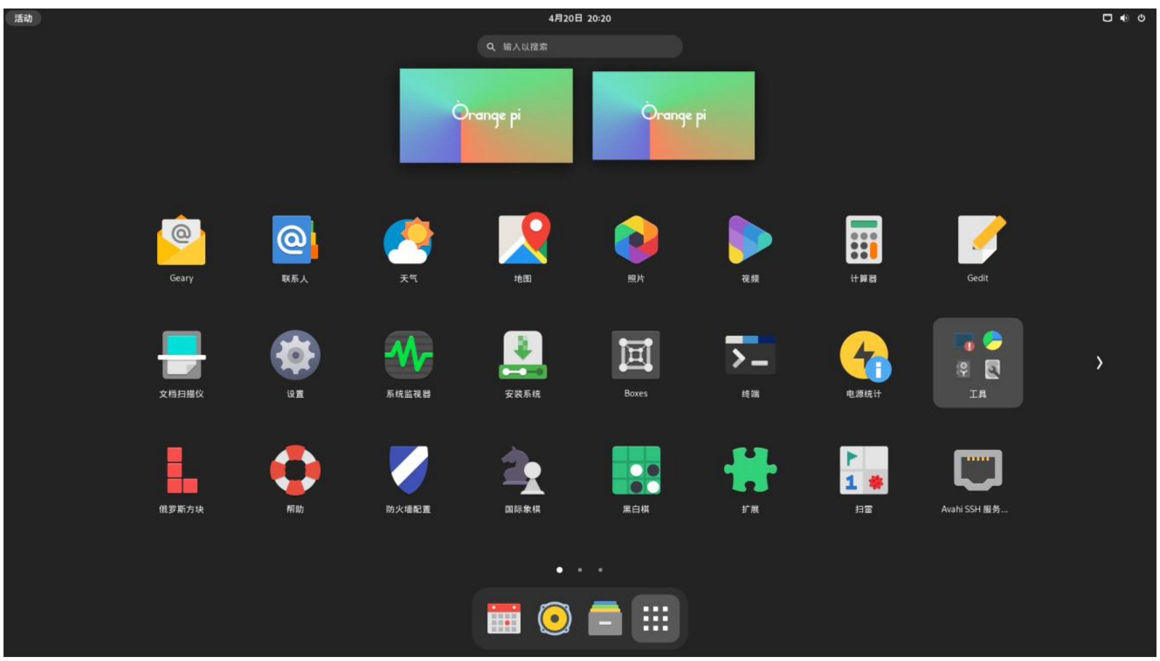
6) Then click on **Select**。



7) Then click on **Logout...** Log out of the system and then log in again.



8) Then you can see that the desktop is all displayed in English.

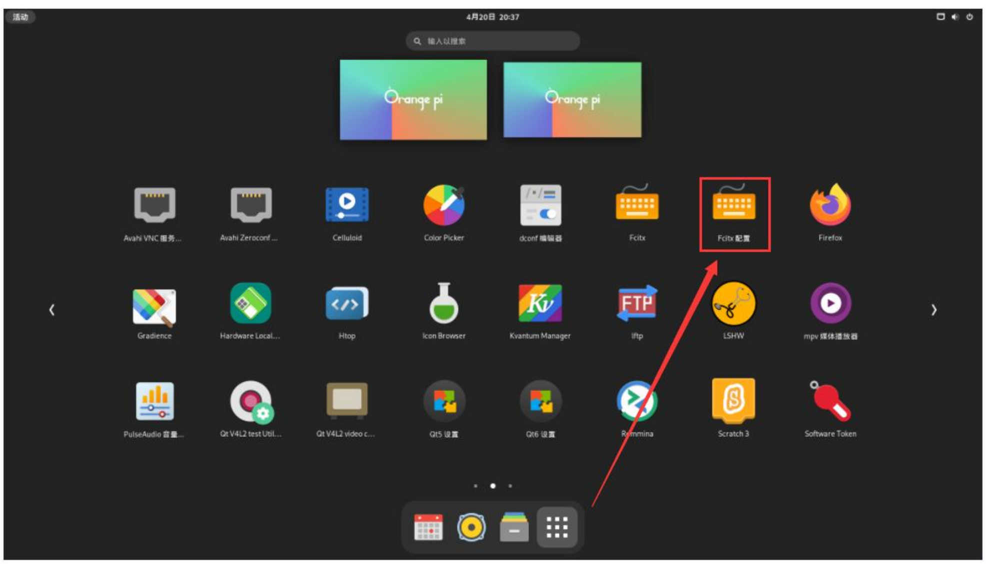


9) Then install fcitx-im and fcitx-configtool.

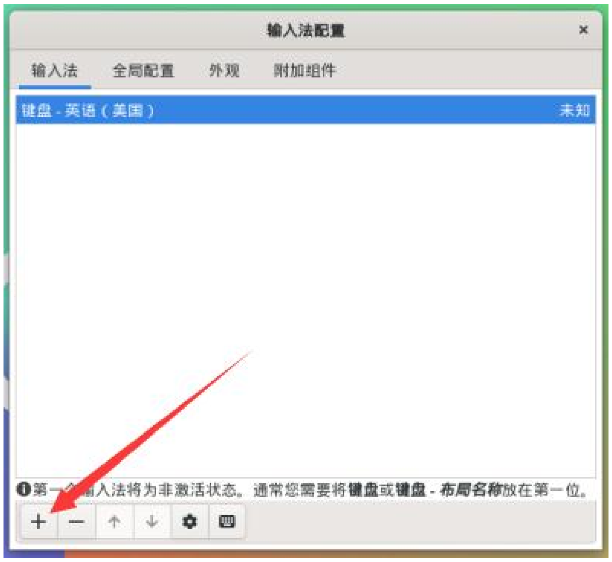
```
[orangepi@orangepi ~]$ sudo pacman -S fcitx-im fcitx-configtool
:: There are 3 members in group fcitx-im:
:: Software Warehouse community
   1) fcitx    2) fcitx-qt5    3) fcitx-qt6
```

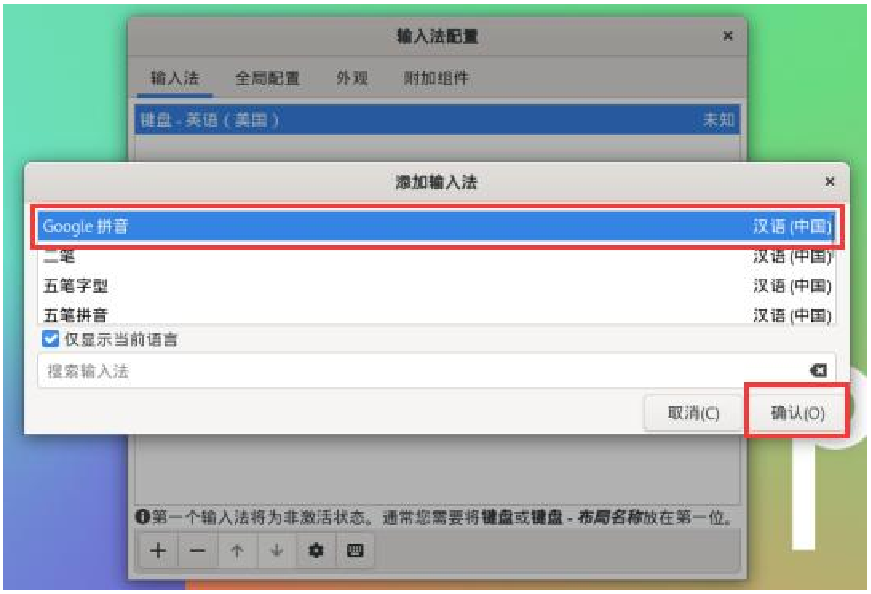
```
Enter a selection (default=select all): 1
```

10) Then open the Fcitx configuration program.

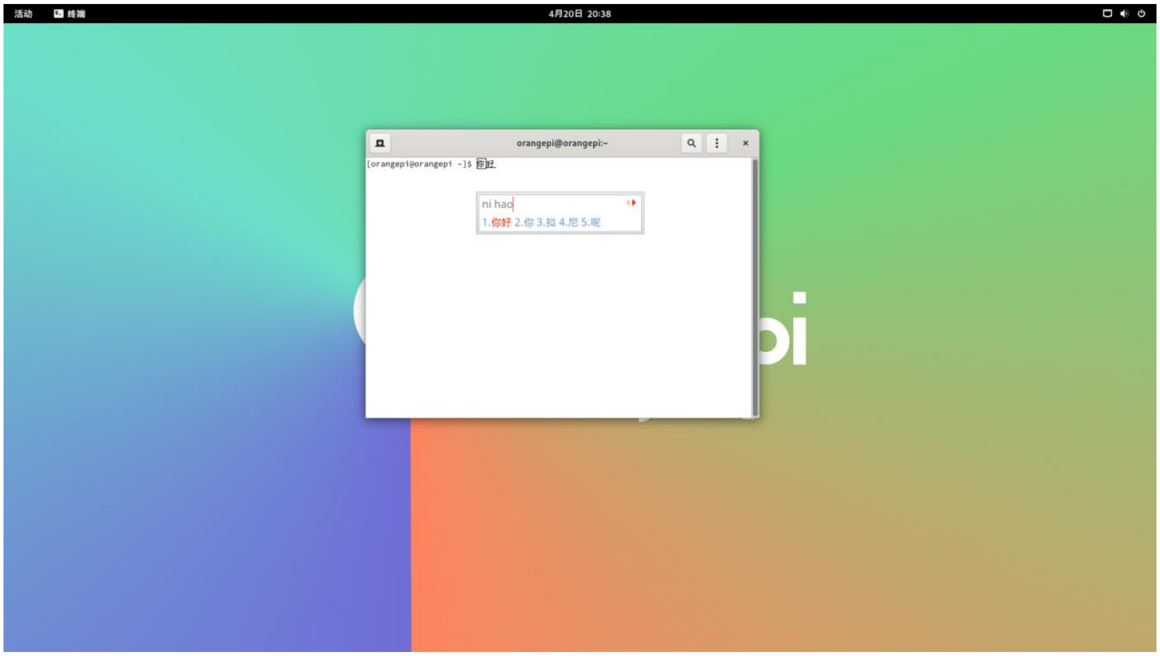


11) Then add **Google Pinyin** input method.





12) Then we can open a terminal to test the Chinese input method. After opening the terminal, if it defaults to the English input method, we can use the **Ctrl+Space** shortcut to switch to the Chinese input method, and then input Chinese.。


活动
终端
4月20日 20:38
orangepi@orangepi:~
[orangepi@orangepi ~]$ 你好
ni hao
1.你好 2.你 3.拟 4.尼 5.呢

6. Instructions for using the Android 13 system

6. 1. Supported Android versions

Android version	Kernel version
Android 13	Linux5.10

6. 2. Android function adaptation

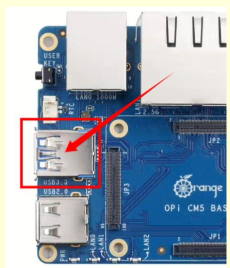
Function	Android 13
HDMI Display	OK
HDMI Audio	OK
USB 2.0	OK
USB 3.0	OK
Gigabit Ethernet	OK
2.5G network port x 2	OK
Debug serial port	OK
FAN	OK
RTC	OK
eMMC	OK
GPIO（12pin）	OK
UART（12pin）	OK
SPI（12pin）	OK
I2C（12pin）	OK
CAN（12pin）	OK
Camera x 4	NO
LED Light	OK
TF card startup	OK
Infrared	OK
GPU	OK

NPU	**OK**
VPU	**OK**
Power button	**OK**
Watchdog test	**OK**
Chromium hard decoding video	**OK**
HDMI CEC function	**NO**

6. 3. How to use ADB

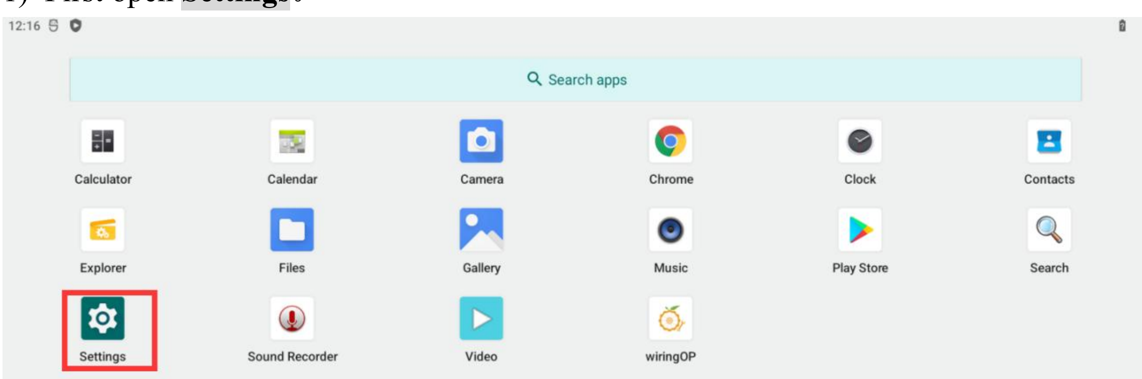
6. 3. 1. USB OTG mode switching method

The development board has two USB ports. The USB port marked with a red arrow in the figure below supports both Host mode and Device mode. The other USB2.0 port only supports Host mode.

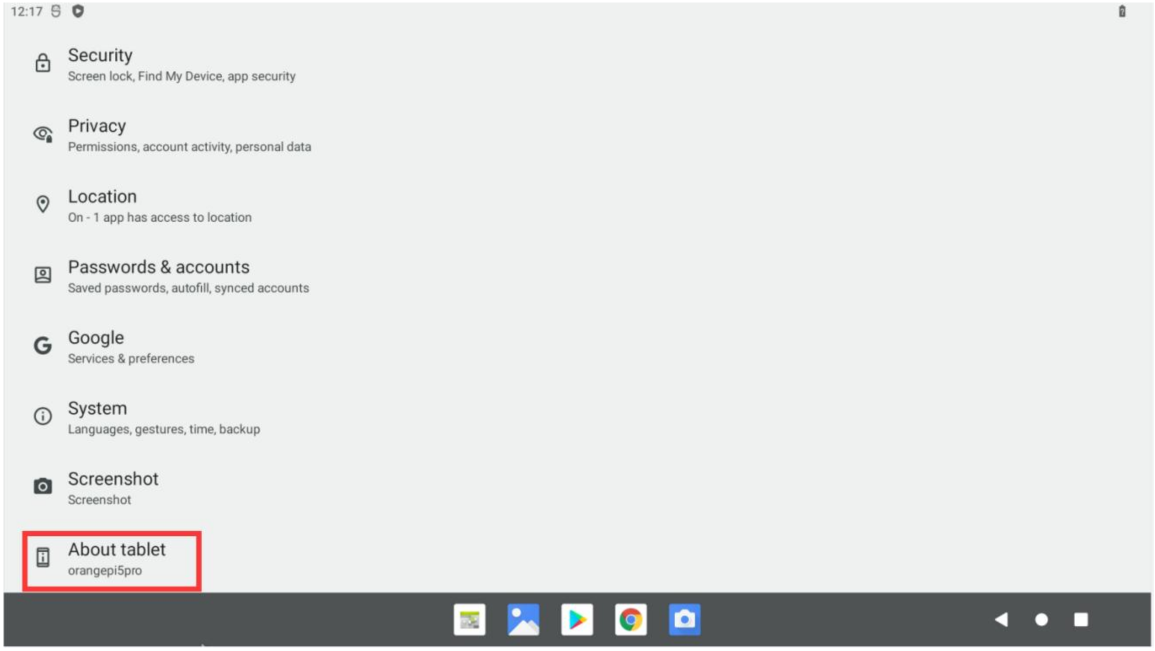


The USB OTG interface is in Host mode by default and can be used to connect USB devices such as mouse and keyboard. If you want to use ADB, you need to manually switch to Device mode.

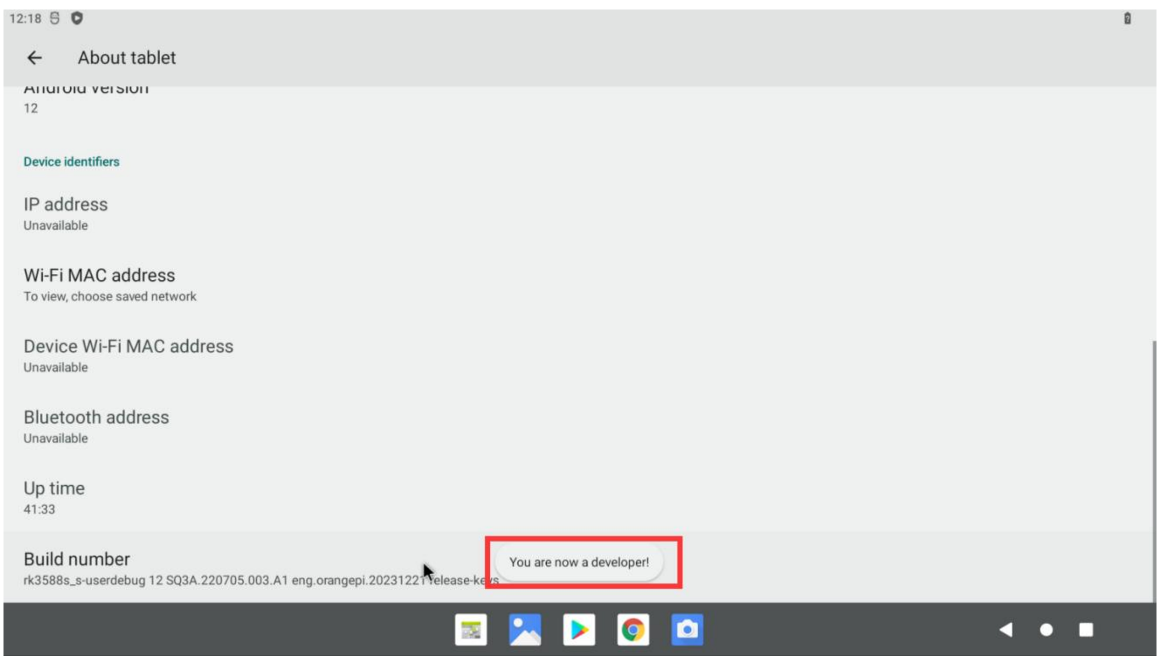
1) First open **Settings**。



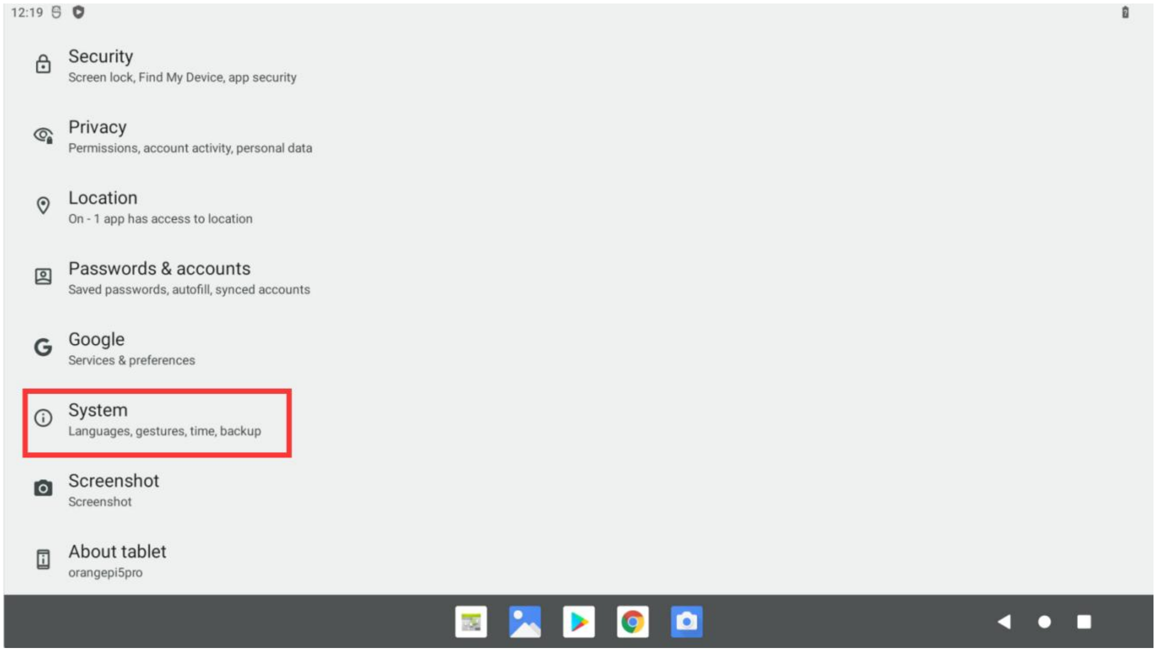
2) Then select **About tablet**。



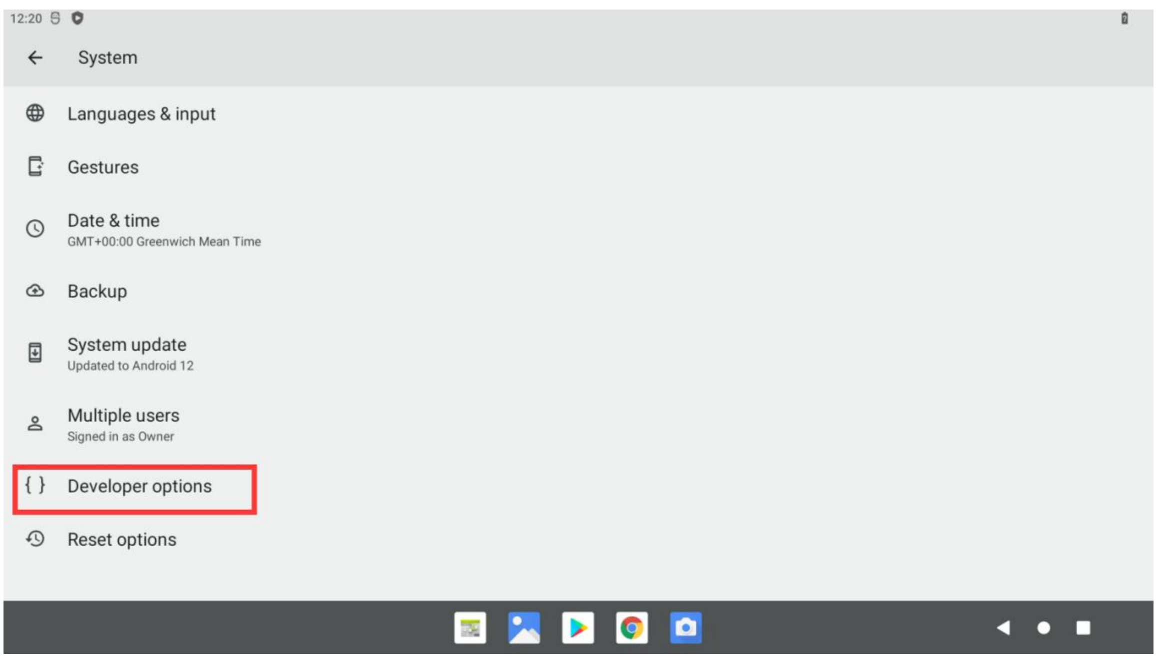
3) Then use the mouse to click the **Build number** menu bar multiple times until the prompt "**You are now a developer!**" appears.



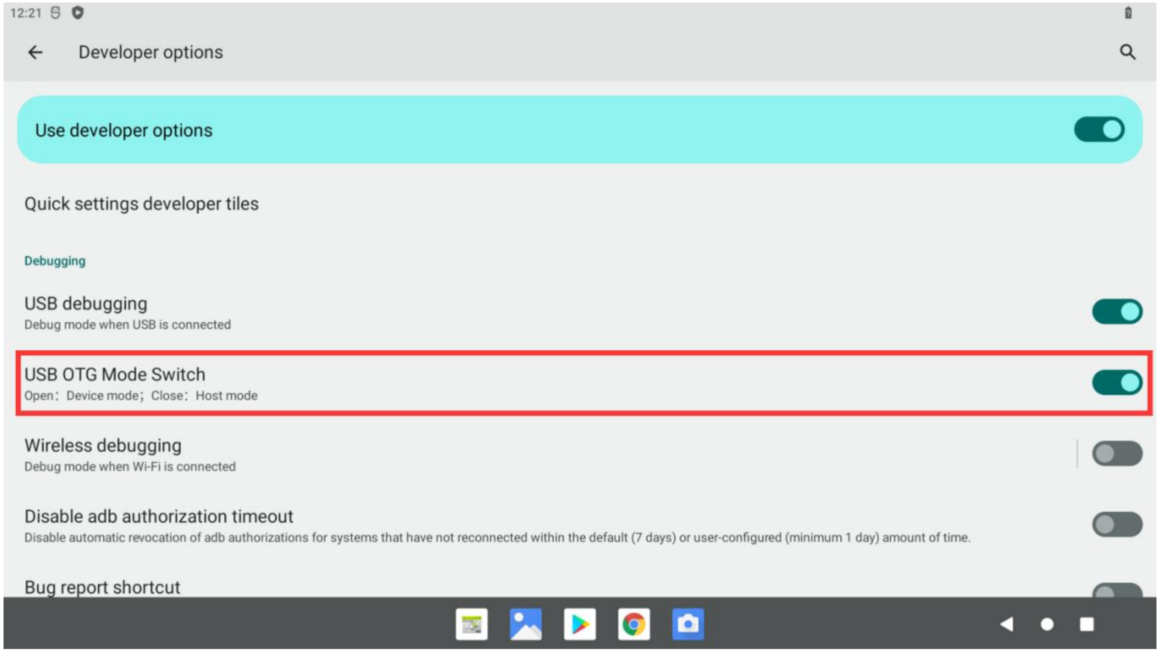
4) Then click to return to the previous menu and select **System**.



5) Then select **Developer options**。



6) Finally, find the **USB OTG Mode Switch**, **turn it on to switch to Device mode, and turn it off to switch to Host mode.**



6. 3. 2. Use a data cable to connect adb debugging

4) First, prepare a good quality USB male-to-male data cable.

5) Then refer to the **USB OTG mode switching method** to switch USB OTG to device mode.

6) Then use a USB male-to-male data cable to connect the development board to the USB port of the computer (please also use a Type-C power supply to power the development board).

7) Install adb tool on your Ubuntu PC.

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

8) Use the following command to view the identified ADB devices.

```
test@test:~$ adb devices
```

```
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

9) Then you can log in to the Android system through adb shell on the Ubuntu PC.

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

10) Execute the command to remount the Android system.

```
test@test:~$ adb root
test@test:~$ adb remount
```

11) Then you can transfer files to the Android system.

```
test@test:~$ adb push example.txt /system/
```

6. 3. 3. Using network connection adb debugging

When using network adb, you don't need a USB male-to-male data cable to connect the computer and the development board. Instead, you can communicate through the network. So first make sure that the wired or USB wireless network of the development board is connected, and then get the IP address of the development board, which will be used later.

1) Ensure that the Android system's **service.adb.tcp.port** is set to port 5555.

```
console:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the network adb port number.

```
console:/ # setprop service.adb.tcp.port 5555
console:/ # stop adbd
console:/ # start adbd
```

3) Install adb tool on your Ubuntu PC.

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

4) Then connect the network adb on the Ubuntu PC.

```
test@test:~$ adb connect 192.168.1.xxx      (The IP address needs to be changed to the IP address of the development board)
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xxx:5555

test@test:~$ adb devices
List of devices attached
192.168.1.xxx:5555          device
```

5) Then you can log in to the Android system through adb shell on the Ubuntu PC.

```
test@test:~$ adb shell
console:/ #
```

7. Compilation method of Android 13 source code

7.1. Download the source code of Android 13

1) First, download the compressed file of Android 13 source code from Baidu Cloud Drive or Google Cloud Drive

2) After downloading the compressed file of the Android 13 source code, please first check if the MD5 checksum is correct. If it is not correct, please download the source code again

```
test@test:~$ md5sum -c md5sum
Android_13.tar.gz00: determine
Android_13.tar.gz01: determine
Android_13.tar.gz02: determine
Android_13.tar.gz03: determine
Android_13.tar.gz04: determine
Android_13.tar.gz05: determine
Android_13.tar.gz06: determine
Android_13.tar.gz07: determine
Android_13.tar.gz08: determine
```

3) Then it is necessary to merge multiple compressed files into one and decompress them

```
test@test:~$ cat Android_13.tar.gz0* | tar -xvzf -
```

7.2. Compile the source code for Android 13

1) First, install the software packages required to compile the Android 13 source code

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y git gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip
test@test:~$ sudo apt-get install -y u-boot-tools
```

2) There is a make.sh compilation script in the source code, with the following compilation parameters

a. **-B**：compile uboot
b. **-K**：compile kernel
c. **-a**： compile android
d. **-F**：compile uboot、kernel and android
e. **-M**：Generate partition images in the rockdev directory
f. **-u**：Package to generate a complete image that can ultimately be launched
g. **-b**：Specify the development board model

3) Compile uboot, kernel, android, and package them into a complete image that can ultimately be launched

```
test@test:~$ cd Android_13
test@test:~/ Android_13$ ./make.sh -FMu -b orangepicm5 --gapps
```

4) After compilation, the following information will be printed

```
********rkImageMaker ver 2.1********
Generating new image, please wait...
Writing head info...
Writing boot file...
Writing firmware...
Generating MD5 data...
MD5 data generated successfully!
New image generated successfully!
Making update.img OK.
Make update image ok!
```

5) The final generated image file will be placed in the **rockdev/Image-rk3588s_t** directory. Among them, **update.img** supports TF card and eMMC boot images

```
test@test:~/Android_13$ cd rockdev/Image-rk3588s_t
test@test:~/Android_13/rockdev/Image-rk3588s_t $ ls update*
update.img
```

8. OpenWRT System Usage Instructions

8. 1. OpenWRT edition

OpenWRT edition	Kernel version
v22.03.4	Linux5.10.110

8. 2. OpenWRT Adaptation situation

Function	OpenWRT
USB2.0	OK
USB3.0	OK
3pin Debug UART	OK
TF card startup	OK
2.5G PCIe Network port X2	OK
1000M Network port	OK
Network port status light	OK
LED light	OK
RTL8821CU USB network card	OK
RTL8723BU USB network card	OK
FAN Fan interface	OK
eMMC	OK

8. 3. The first boot to expand rootfs

1) When starting the OpenWRT system for the first time, the **resize-rootfs.sh** script will be executed to expand rootfs, and it will automatically restart after the expansion is completed

2) After logging into the system, you can use the **df -h** command to check the size of

rootfs. If it matches the actual capacity of the storage device (TF card, eMMC, or NVME SSD), it indicates that the automatic expansion is running correctly

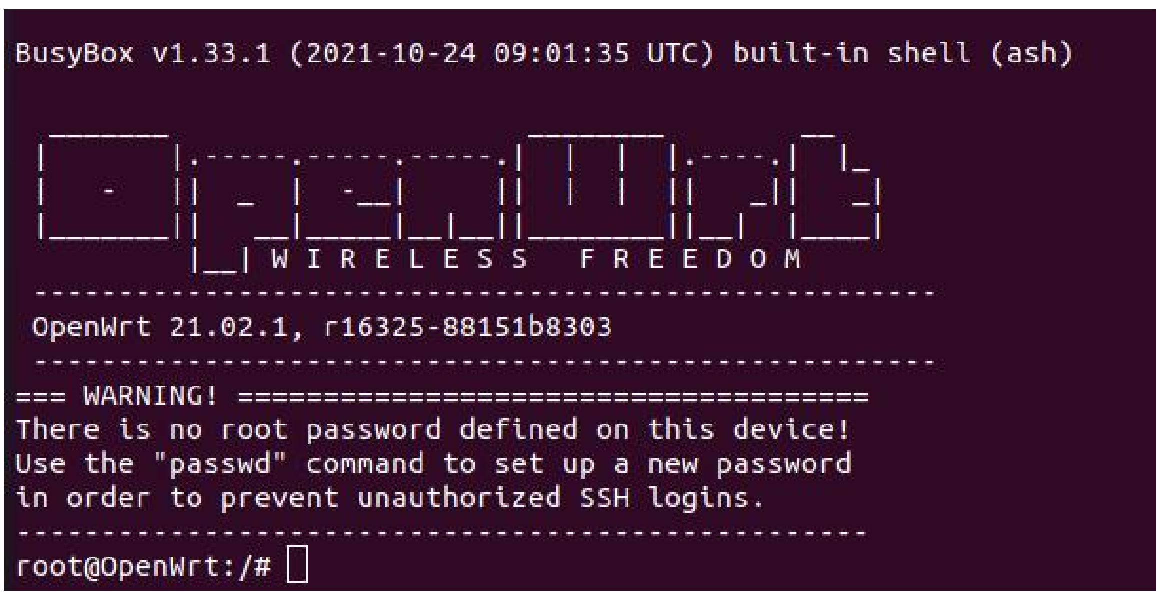
```
root@OpenWrt:~# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                14.8G     14.7G     91.6M  99% /
tmpfs                   495.5M      6.1M    489.4M   1% /tmp
tmpfs                   512.0K         0    512.0K   0% /dev
/dev/root                14.8G     14.7G     91.6M  99% /opt/docker
```

8. 4. Method of logging into the system

8. 4. 1. Login via serial port

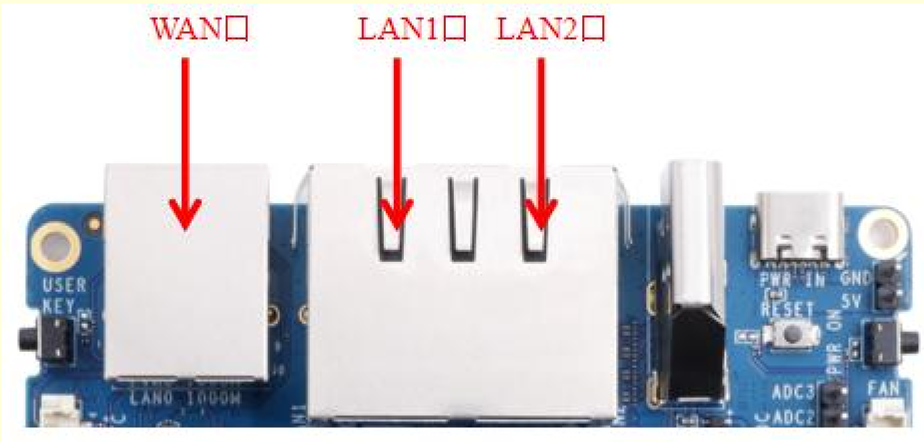
1) Firstly, to debug the use of the serial port, you can refer to the **chapter on debugging the usage of the serial port**

2) The OpenWrt system will automatically log in as the **root** user by default, and the display interface is as follows



8. 4. 2. Login to the system via SSH

Please note that in the OpenWrt system of Orange Pi CM5, the LAN0 gigabit network port is configured as a WAN port by default, and the remaining two network ports LAN1 and LAN2 are configured as LAN ports. This document uses LAN1 port for functional testing.



1) Firstly, connect the LAN1 port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

3) If the computer is installed with Ubuntu system, you can execute the following command to log in to the system through SSH. By default, you can log in directly without a password

```
test@ubuntu:~$ ssh root@192.168.2.1
```

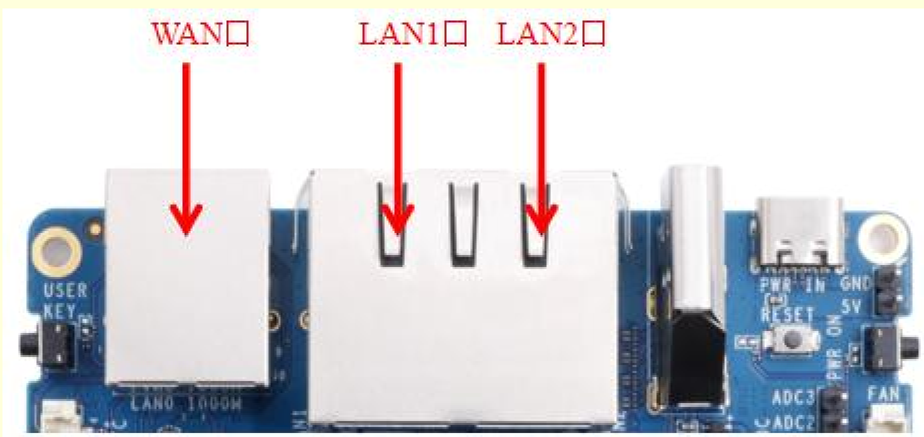
4) The display after successfully logging into the system is shown in the following figure

```
orangepi:~$ ssh root@192.168.2.1


BusyBox v1.35.0 (2023-04-09 12:27:46 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 22.03.4, r20123-38ccc47687
 -----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@OpenWrt:~#
```

5) If the computer is installed with Windows system, you can refer to the method introduced in the section of **SSH remote login development board under Windows to log in**

8. 4. 3. Login to LuCI Management Interface

Please note that in the OpenWrt system of Orange Pi CM5, the LAN0 gigabit network port is configured as a WAN port by default, and the remaining two network ports LAN1 and LAN2 are configured as LAN ports.



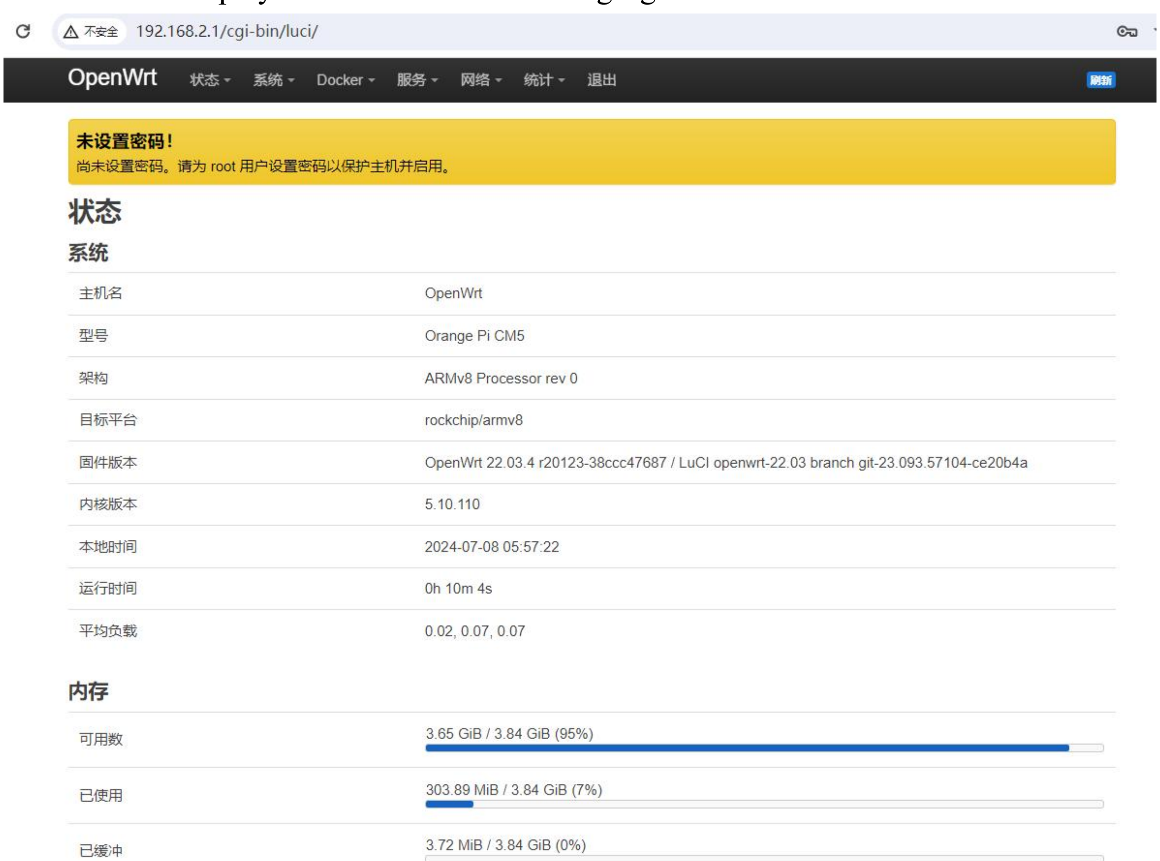
1) Firstly, connect the LAN1 port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address

through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

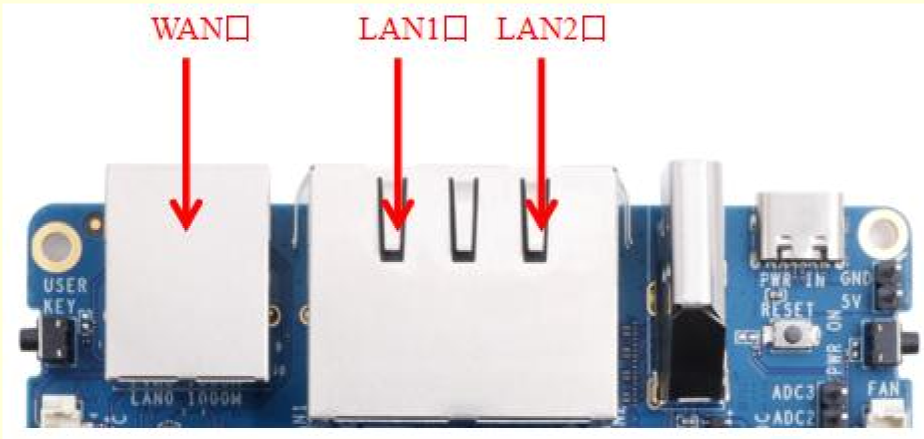
3) You can log in to the LuCI interface by entering the IP address **192.168.2.1** in the browser on your computer

4) **OpenWrt 系统默认是没有设置密码的**，所以 **The OpenWrt system does not have a password set by default**, so simply click the **login** button. After successful login, the interface will display as shown in the following figure



8. 4. 4. Log in to the terminal through the LuCI management interface

Please note that in the OpenWrt system of Orange Pi CM5, the LAN0 gigabit network port is configured as a WAN port by default, and the remaining two network ports LAN1 and LAN2 are configured as LAN ports.



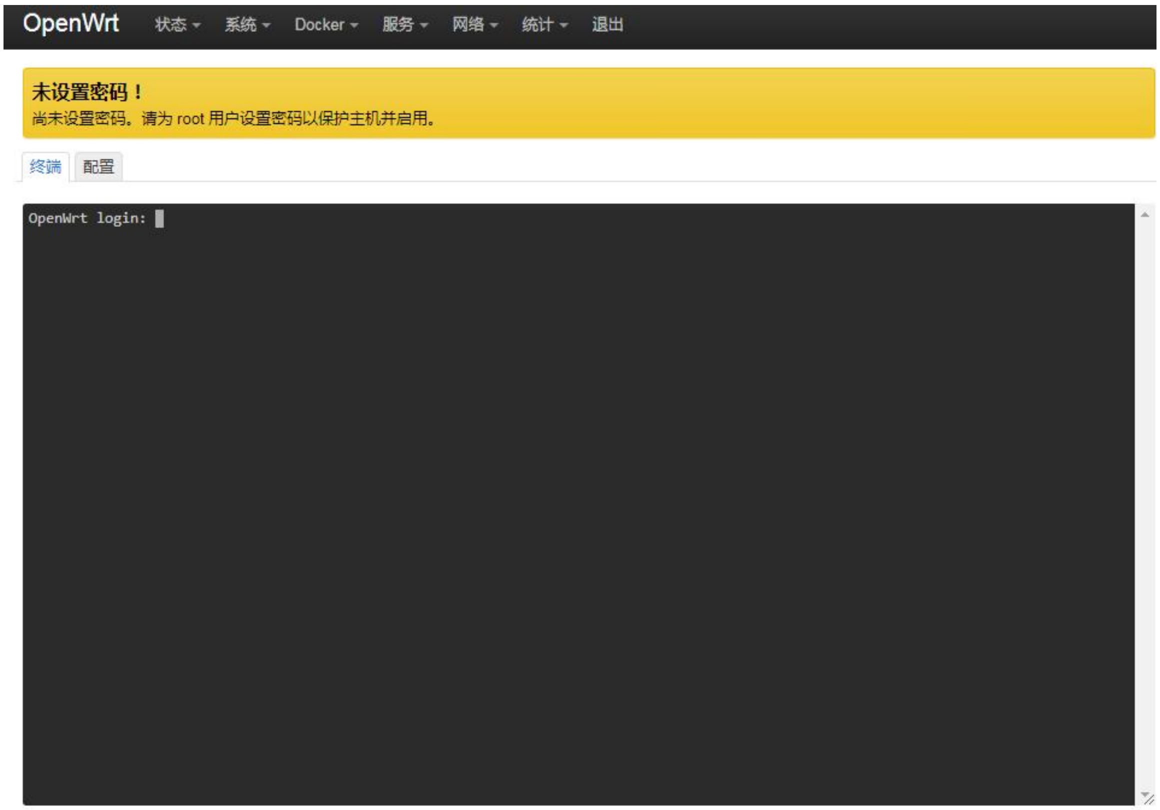
1) Firstly, connect the LAN1 port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

3) You can log in to the LuCI interface by entering the IP address **192.168.2.1** in the browser on your computer

4) Select **"Terminal"** in the **"Services"** column of the navigation bar and click to enter

5) At this point, the terminal interface is shown in the following figure



6) Enter the username root to log in

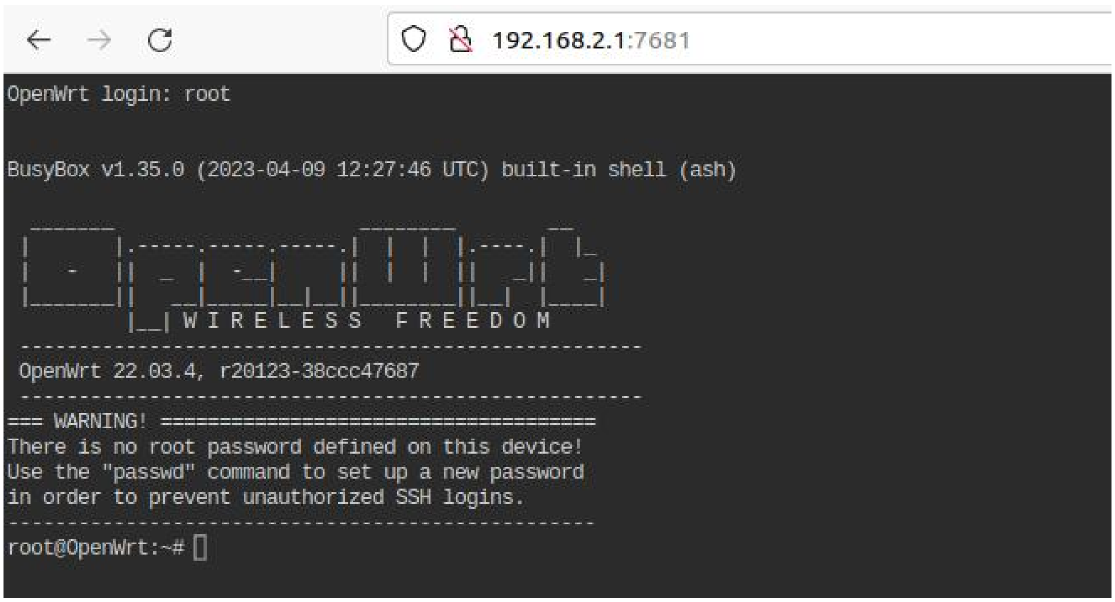
8. 4. 5. Login to the terminal using IP address and port number

Please note that in the OpenWrt system of Orange Pi CM5, the LAN0 gigabit network port is configured as a WAN port by default, and the remaining two network ports LAN1 and LAN2 are configured as LAN ports.

1) Firstly, connect the LAN1 port of the board to the network port of the computer using an Ethernet cable, so that the network port of the computer can obtain the IP address through DHCP

2) The default LAN port IP of the board is set to **192.168.2.1**, so the computer can obtain IP addresses starting with **192.168.2** at this time

3) Then enter **192.168.2.1:7681** in the browser to log in to the OpenWRT terminal



8.5. Method of modifying LAN port IP address through command line

1) In the OpenWrt system, a command-line tool uci is provided, which can easily modify, add, delete, and read the contents of configuration files. For detailed instructions, please refer to the **official documentation**

2) First, use the following command to obtain the network configuration. The corresponding configuration file is **/etc/config/network**, and you can see that the value of **network.lan.ipaddr** is **192.168.2.1**

```
root@OpenWrt:~# uci show network
...
network.lan=interface
network.lan.device='br-lan'
network.lan.proto='static'
network.lan.ipaddr='192.168.2.1'
network.lan.netmask='255.255.255.0'
```

```
network.lan.ip6assign='60'
....
```

3) Then enter the following command to modify the **network.lan.ipaddr** option

```
root@OpenWrt:~# uci set network.lan.ipaddr='192.168.100.1'
```

4) Then enter the following command to complete the submission, which is written to the configuration file

```
root@OpenWrt:~# uci commit
```

If the IP address in red font matches the one to be set, it indicates that the modification was successful

```
root@OpenWrt:~# cat /etc/config/network
...
config interface 'lan'
        option device 'br-lan'
        option proto 'static'
        option netmask '255.255.255.0'
        option ip6assign '60'
        option ipaddr '192.168.100.1'
...
```

5) Restart the network through Ubuntu. Please refer to the **official documentation** for instructions on how to use Ubuntu 过 ubus

```
root@OpenWrt:~# ubus call network restart
```

6) At this point, entering the command shows that the IP address of the LAN port is already **192.168.100.1**

```
root@OpenWrt:~# ifconfig br-lan
br-lan    Link encap:Ethernet  HWaddr FE:55:13:A3:EF:E7
          inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fd60:c4cd:1033::1/60 Scope:Global
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B)    TX bytes:370 (370.0 B)
```

8. 6. Method for changing root password

8. 6. 1. Modify via Command Line

1) Firstly, enter passwd root in the system command line, and the following prompt message will appear. At this time, you can enter the password you want to set and press Enter to confirm

```
root@OpenWrt:/# passwd root
Enter new UNIX password:
```

2) Next, you will be prompted to re-enter the password. At this point, enter the password again to confirm and press Enter

```
Retype password:
```

3) The successfully modified display is as follows

```
passwd: password for root changed by root
```

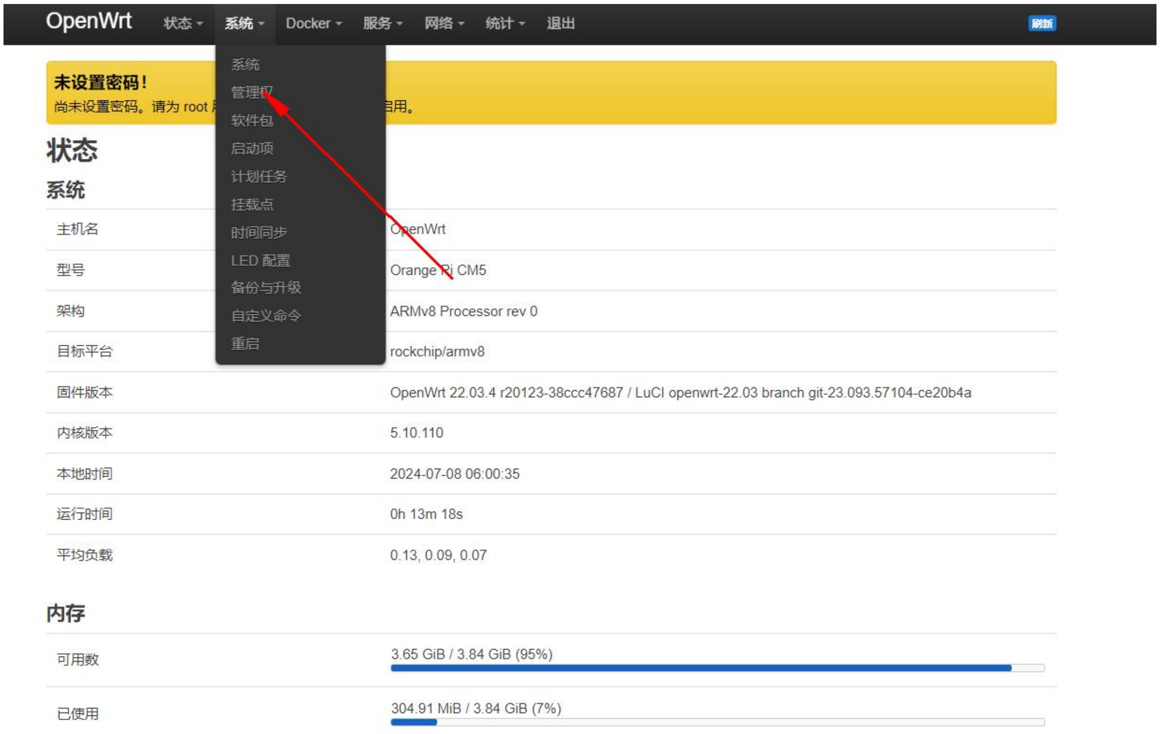
8. 6. 2. Modify through LuCI management interface

1) Firstly, refer to **the login LuCI management interface** to enter the OpenWRT management interface

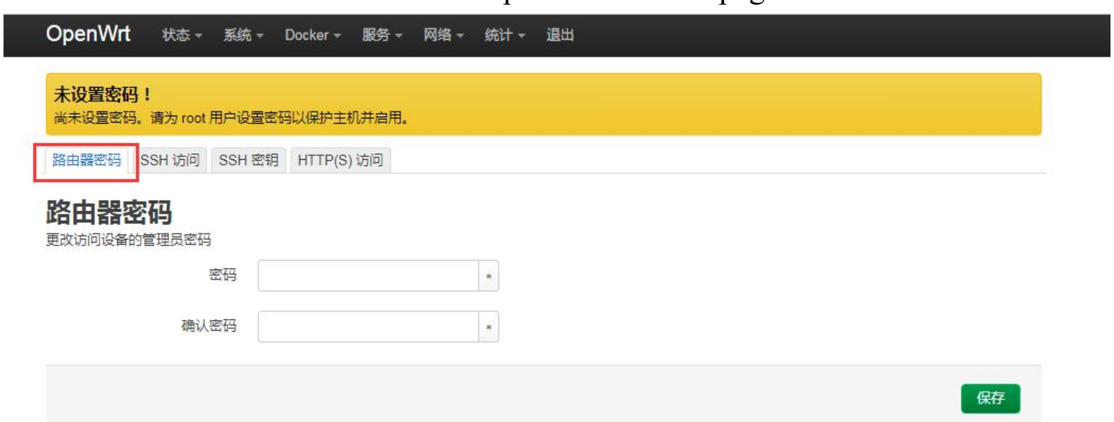
2) Then follow the steps below to change the password

a. Find the **"System"** option in the navigation bar and click on it

b. In the vertical bar options below the system, select **"Management Rights"** and click



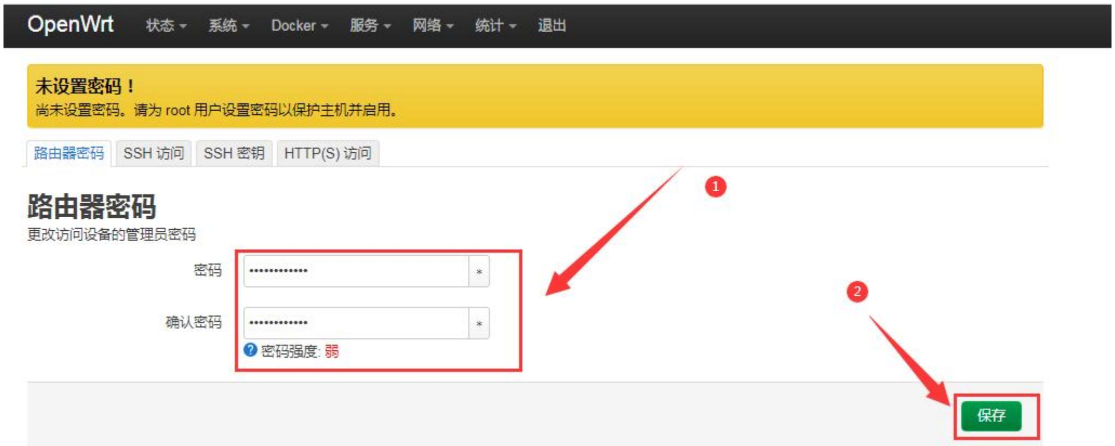
c. Select the **'Router Password'** option on the Tab page



3) Change and save router password

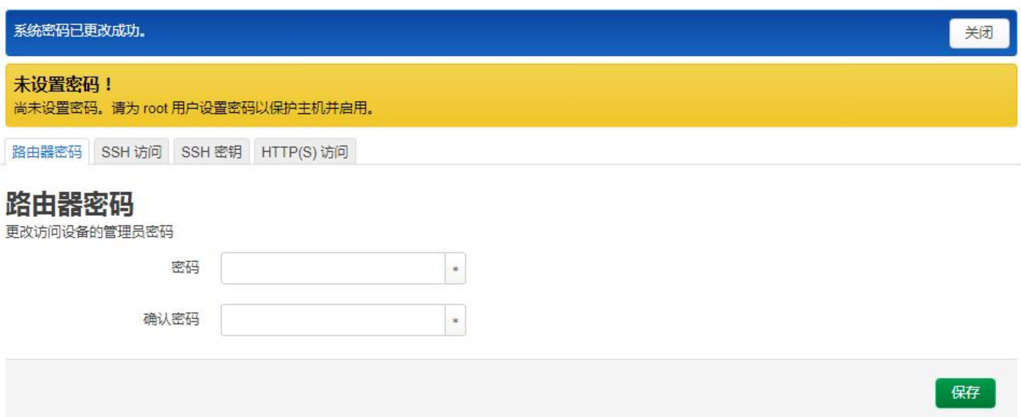
a. Enter the password you have set in the **"Password"** and **"Confirm Password"** dialog boxes (if unsure if the password is entered correctly, click the **"*"** icon behind the dialog box to display the input characters)

b. Click **'Save'** to save the newly modified password



Note: In the "Password" and "Confirm Password" dialog boxes, the password entered twice needs to be consistent.。

4) After the password is successfully changed, a pop-up message saying **"System password has been changed successfully"** will appear. At this time, logging into OpenWRT requires a password to log in



8. 7. USB interface testing

8. 7. 1. Mounting USB storage devices at the command line

1) Firstly, insert the USB drive into the USB interface of the Orange Pi development board

2) If you can see the output of sdX by executing the following command, it indicates that the USB drive recognition is successful

```
root@OpenWrt:~# cat /proc/partitions | grep "sd*"
major minor    #blocks    name
    8              0      15126528 sda
```

3) You can use the mount command to mount the USB drive to /mnt, and then you can view the files on the USB drive

```
root@OpenWrt:~# mount /dev/sda /mnt/
root@OpenWrt:~# ls /mnt/
test.txt
```

4) After mounting, you can use the df -h command to view the capacity usage and mounting points of the USB flash drive

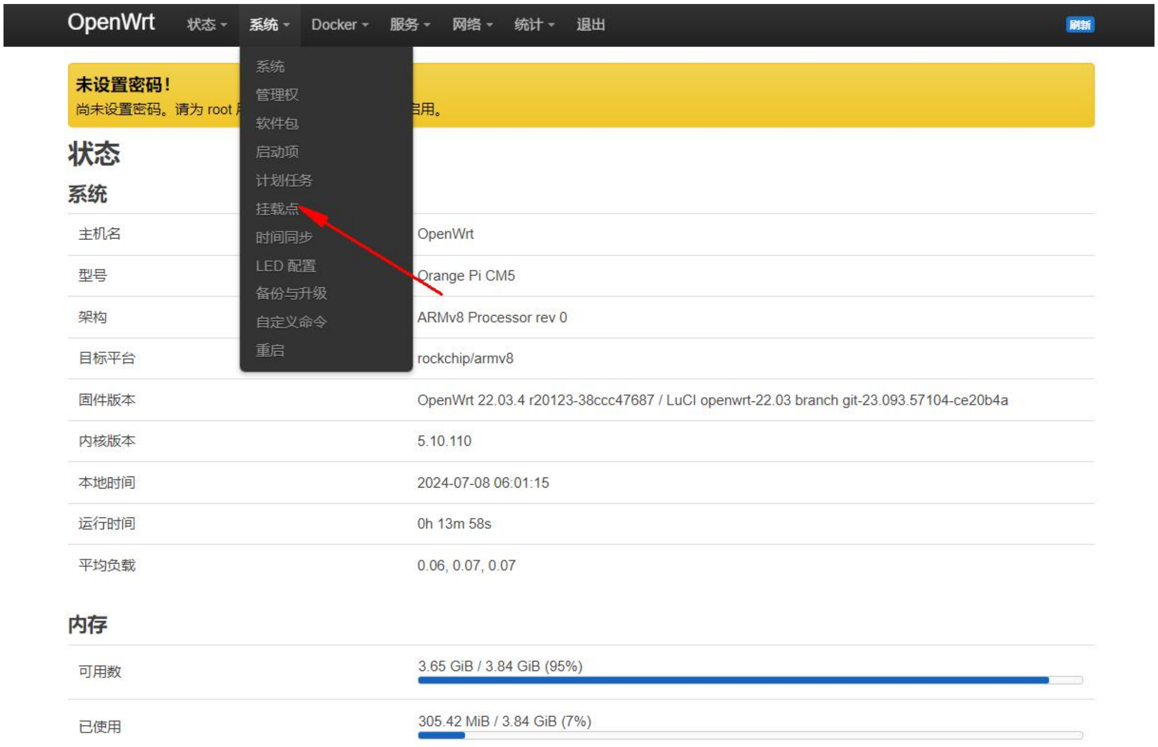
```
root@OpenWrt:~# df -h | grep "sd"
/dev/sda              14.4G      187.2M        14.2G     1% /mnt
```

8. 7. 2. Mounting USB storage devices on the LuCI management interface

1) Firstly, connect the USB flash drive (or other storage device) to the development board via USB2.0

2) Then follow the **login LuCI management interface** to enter the LuCI management world

3) Then, in the LuCI management interface, click on "System ->Mount Point" to enter the configuration interface of the mount point



4) Then follow the steps below to add a mounting point

a. Find **'Mount Point'** below the **global settings interface** for mount points

b. Below the **mounting point**, select the "Add" button and click to enter

c. Next, a pop-up window will appear below

d. Then you can start mounting the storage device

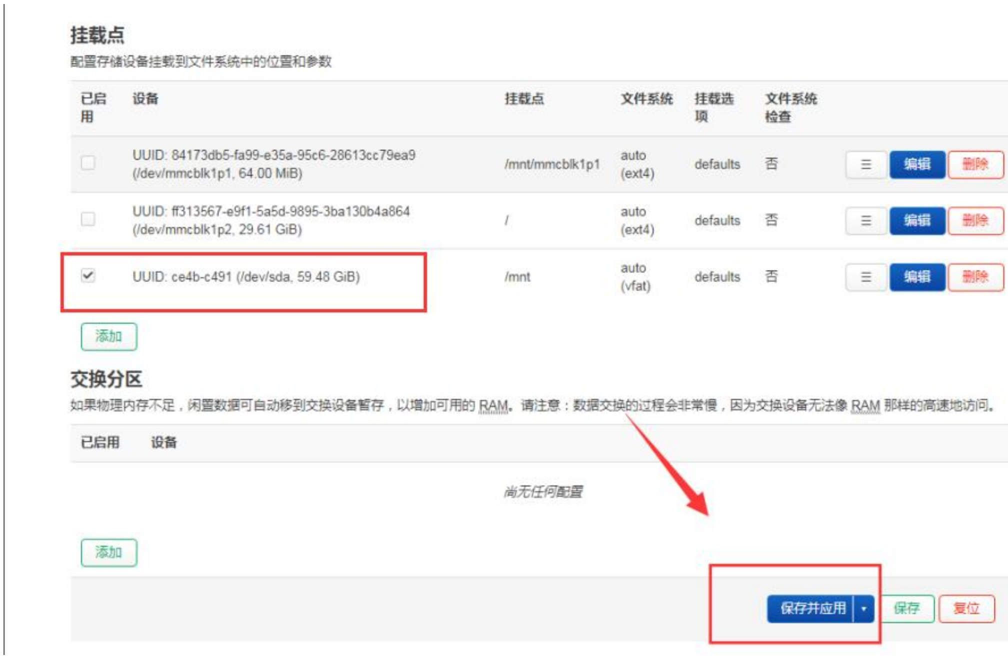
   a) Check **'Enabled'**

   b) Select the actual connected device /dev/sda in the UUID column of the general settings (choose according to your own device)

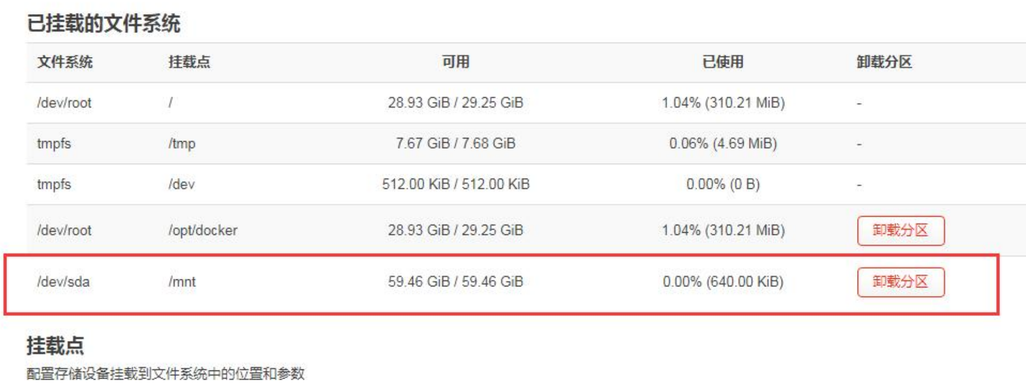
   c) 在挂 Select **"Custom"** in the mount point column and fill in the target directory to be mounted to. Taking the **/mnt** directory as an example, fill in and press **Enter** to confirm

   d) Then click the **"Save"** button in the bottom right corner

5) Then you will return to the mount point global settings page and click **"Save and Apply"** in the bottom left corner of the page to make the mount point effective



6) After saving, you can see in the **"Mounted File System"** that the storage device has been successfully mounted



8. 8. USB Wireless Network Card Test

The currently tested USB wireless network cards that can be used are shown below. For other models of USB wireless network cards, please test them yourself. If they cannot be used, you need to port the corresponding USB wireless network card driver.

Serial number	model	

1	RTL8723BU Support 2.4G WIFI+BT4.0	
2	RTL8821CU Support 2.4G +5G WIFI Support BT 4.2	
3	RTL8811 Support 2.4G +5G WIFI	

8.8.1. Method of using a USB wireless network card to connect to a WIFI hotspot

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power on the development board.

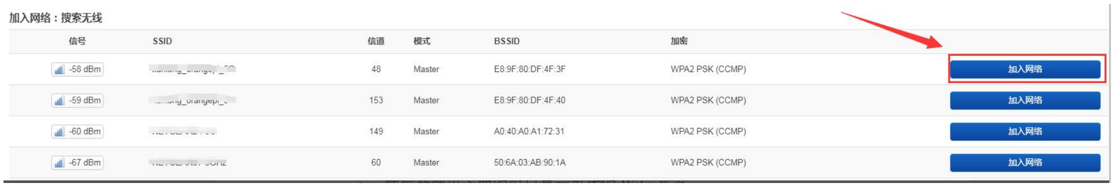
2）After the system startup is complete, click on **Network ->Wireless** to enter the wireless WiFi configuration interface.

3）The default wireless configuration of OpenWRT system is **Master** mode. For the convenience of the next operation, we will remove the default wireless connection.

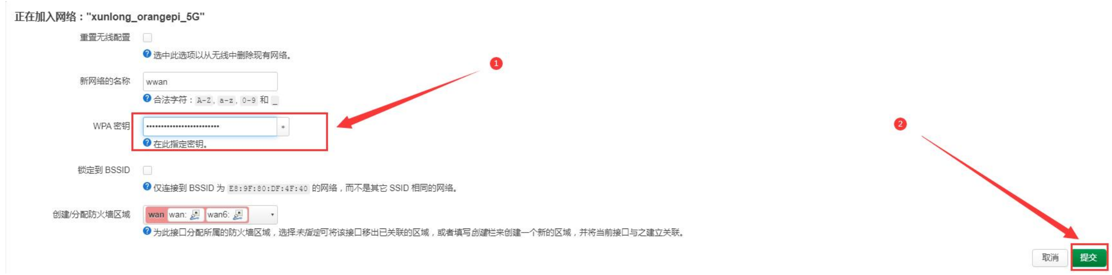
4）Then click on the bottom right corner of the page to **save** and make the configuration effective.

5）Then click the **scan** button to scan the surrounding WiFi hotspots.

6）Then a window will pop up displaying available WiFi hotspots. Click the **Join Network** button to the right of the desired WiFi hotspot to connect.



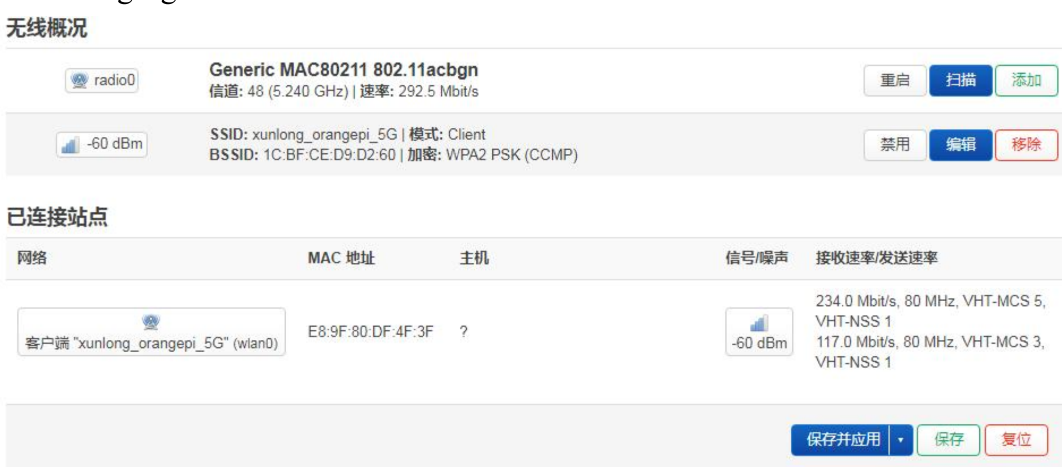
7）Then a interface will pop up to connect to the WiFi hotspot. We will enter the hotspot password at the location shown in the figure below, and then click the **submit** button.



8）Then the following interface will pop up, click the **save** button in the bottom right corner.

9）Finally, you will return to the main interface of wireless configuration, click **save and apply**, and wait for the configuration to be applied.

10）After successfully connecting to the WiFi hotspot, the interface displays as shown in the following figure.



8. 8. 2. Method for creating a WIFI hotspot using a USB wireless network card

1）Insert the USB wireless network card into the USB port of the development board, and then connect the power supply to power on the development board.

2）The system startup is complete, click on **Network ->Wireless** to enter the wireless WiFi configuration interface.

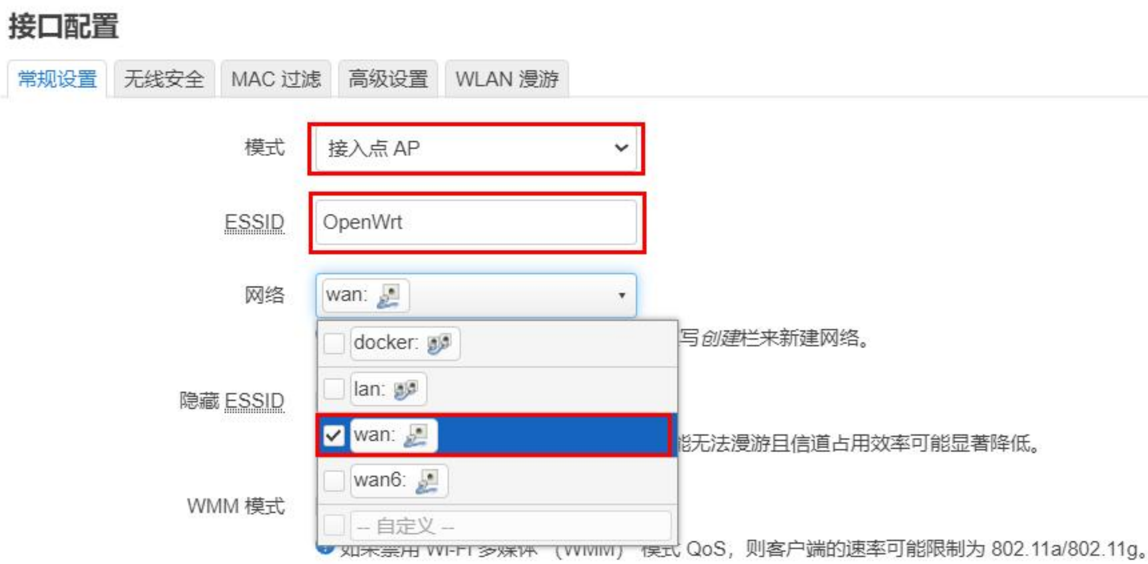
3）The default wireless configuration of OpenWRT system is **Master** mode. For the convenience of the next operation, we will remove the default wireless connection.

4）Then click on the bottom right corner of the page to **save** and make the configuration effective.

5）Then click the **add** button on the right.

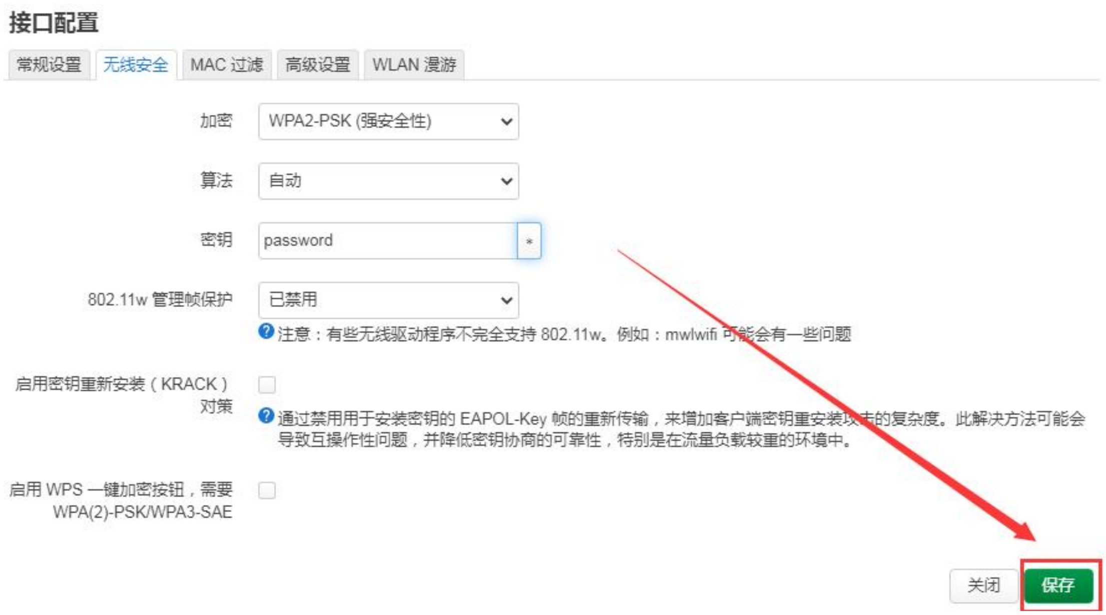
6）In the pop-up tab **device configuration**, we set the parameters as shown in the following figure.

7）Then in **Interface Configuration ->General Settings**, set the mode to **Access Point AP**, **ESSID** (Wireless Network Name) to **OpenWrt**, and network to **wan**



8）Then in **Interface Configuration ->Wireless Security**, select **WPA2-PSK** as the encryption algorithm; Set the key (wireless password) to **password**

9）After completing the above settings, click on the bottom right corner of the page to **save**, and then you will exit the tab



10）Then click on the bottom right corner of the page to **save and apply**, and wait for the configuration to be applied.

11）The display interface for successfully creating a hotspot is shown in the following figure

12）Then use your phone or computer to search for the corresponding WiFi SSID for connection. After successful connection, as shown in the following figure

8. 9. Installing software packages through the command line

8. 9. 1. Installing through OPkg on the terminal

1) Update the list of available software packages

```
root@OpenWrt:/# opkg update
```

2) Get software list

```
root@OpenWrt:/# opkg list
```

3) Install the specified software package

```
root@OpenWrt:/# opkg install <Package Name>
```

4) View installed software

```
root@OpenWrt:/# opkg list-installed
```

5) Uninstall software

```
root@OpenWrt:/# opkg remove <Package Name>
```

8.10. OpenWRT management interface installation software package

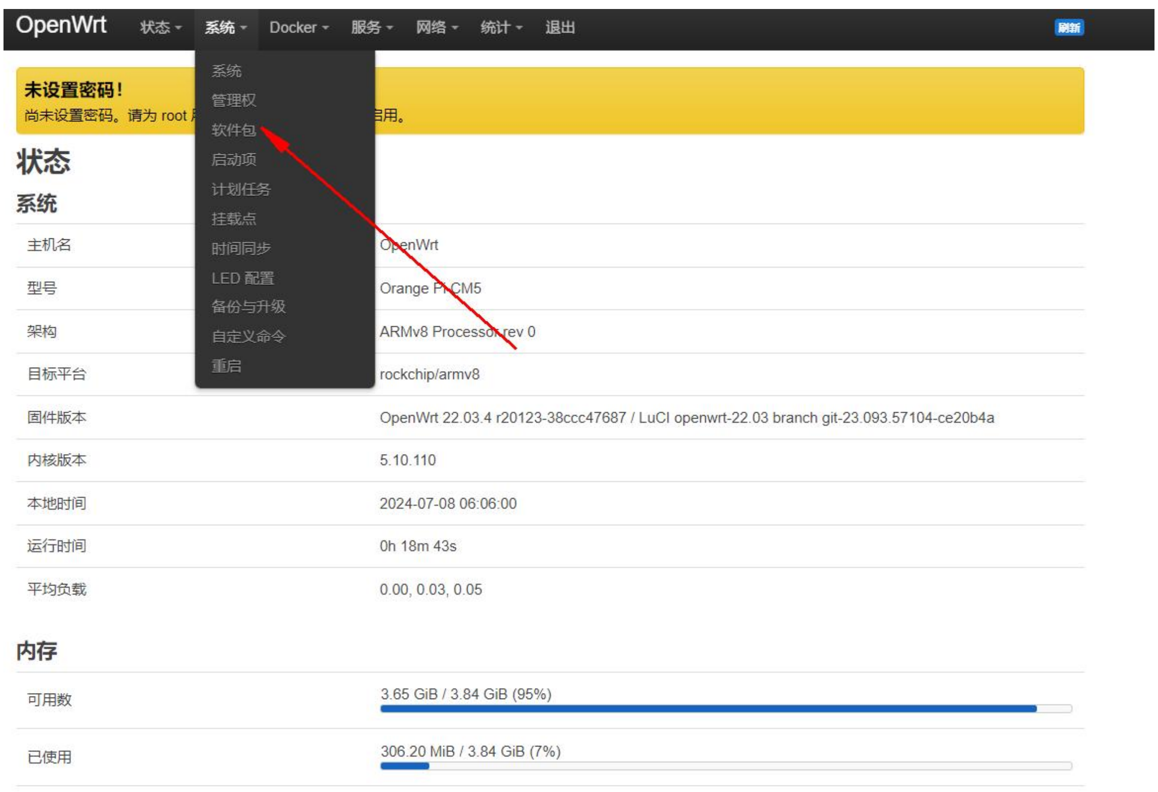
If you need to add software packages, you can install them through the OpenWRT management interface.

8.10.1. View the list of available software packages in the system

1) First, enter the software package management page

a. Find the **"System"** option in the navigation bar and click to enter

b. In the vertical bar options below the system, select **"Software Package"** and click to enter



2) Then the main page of the software package will appear, as shown in the following

figure, to obtain the list of available software

a. In the **"Operation"** option of the software package, click **"Update List"** to obtain the list of available software packages

b. On the tab page, click **"Available"** to view the currently available software packages

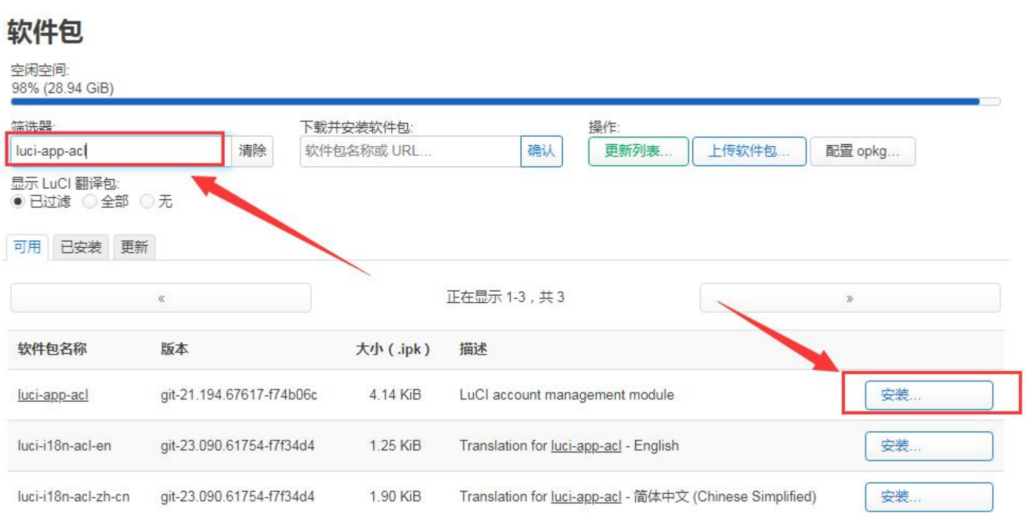
c. View the current number of available software packages

8. 10. 2. Example of Installing Software Packages

1) Taking the installation of the software package "**luci-app-acl**" as an example

a. In the package management interface of OpenWRT, click on the filter dialog box and enter "**luci-app-acl**"

b. In the list of software packages, you can see the version, package size, and description information of the "**luci-app-acl**" package, and then click the "**Install**" button



c. Then the following pop-up window will appear, click "**Install**" to proceed

d. Then wait for the installation to complete

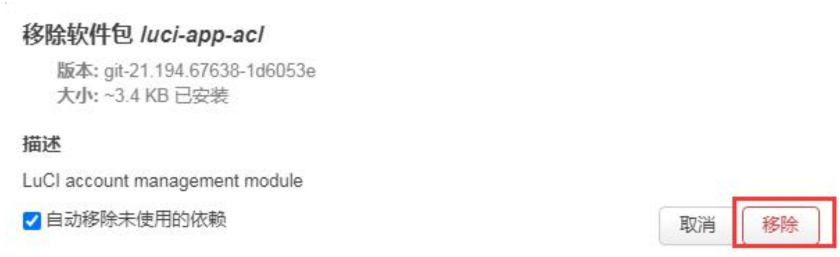
e. The display after installation is as follows

2) Check if the software package has been successfully installed
   a. In the package management interface of OpenWRT, click on the filter dialog box and enter "**luci-app-acl**"
   b. Select and click '**Available**' on the tab page
   c. The '**luci-app-acl**' package will be displayed in the package list and updated to '**installed**' status

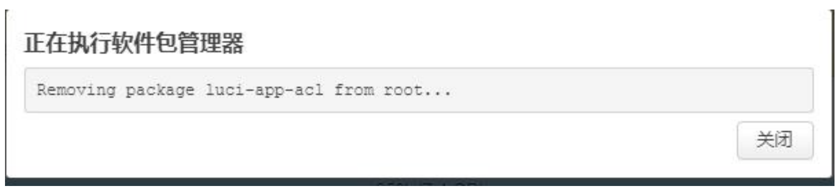
8. 10. 3. Example of Removing Software Packages

1) Taking the removal of the software package '**luci-app-acl**' as an example
   a. In the package management interface of OpenWRT, click on the filter dialog box and enter "**luci-app-acl**"
   b. Select '**Installed**' on the tab page to display a list of installed software packages
   c. Click '**Remove**' on the right to remove the corresponding software package

d. Then a pop-up window will appear below, click '**Remove**' to proceed



e. After successful removal, the display interface is as follows



2) Check if the software package has been successfully removed

a. In the package management interface of OpenWRT, click on the filter dialog box and enter "**luci-app-acl**"

b. Select and click '**Installed**' on the tab page

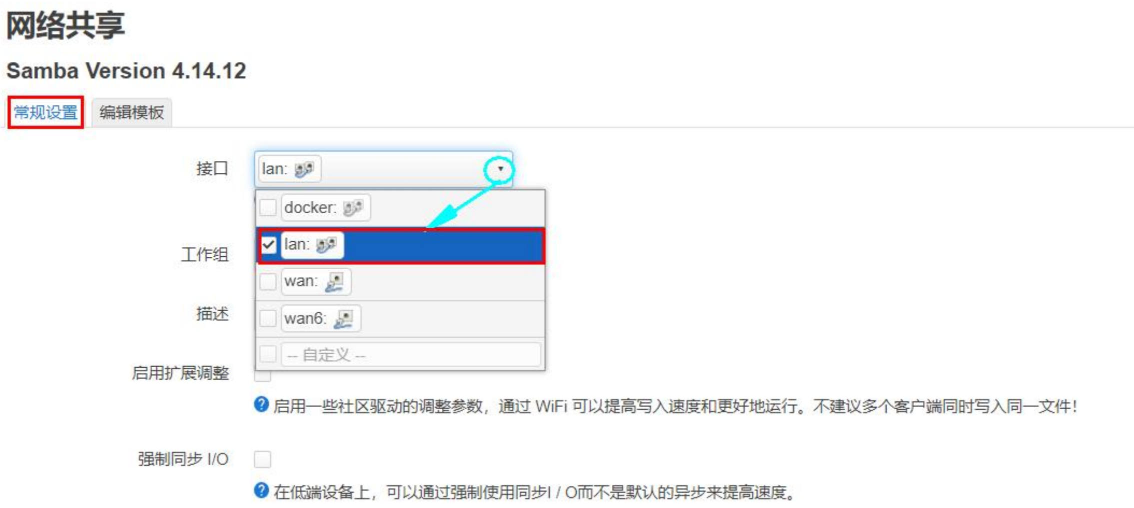
c. The '**luci-app-acl**' package will not be displayed in the package list, and the '**luci-app-acl**' package has been successfully removed

8. 11. Using Samba Network Sharing

There are two main software options for implementing OpenWRT LAN file sharing: Samba and NFS. Samba system has good compatibility, while NFS performs better. For users who need to use Windows devices, it is recommended to choose Samba.

1) Enter the Samba network share management page
   a. Find the "**Services**" option in the navigation bar and click to enter
   b. In the vertical bar options below the service, select "**Network Sharing**" and click to enter

2) Select the interface that Samba service needs to listen on
   a. Select "**General Settings**" in the navigation bar of network sharing and click to enter
   b. The interface is specified according to actual needs. If you want to access it through the "lan port", set it to "**Lan**"



3) Set the shared directory for network sharing

a. Click "Add" in the "**Shared Directory**" section of the "**General Settings**" for network sharing to share the directory address

b. Enter the name of the shared folder as '**mmt**' under the name

c. Under the path of the shared directory, select the location of the shared directory "/"

d. Check '**browseable**' and '**allow anonymous users to run**'

e. Click '**Save and Apply**' to save the configuration

4) window10 starts network discovery and sharing

Note: To access Samba on the Windows 10 system, it is necessary to first confirm whether Windows 10 has started network discovery and sharing. If it has not been started, the following settings should be made first.

a. Enable access to Samba v1/v2

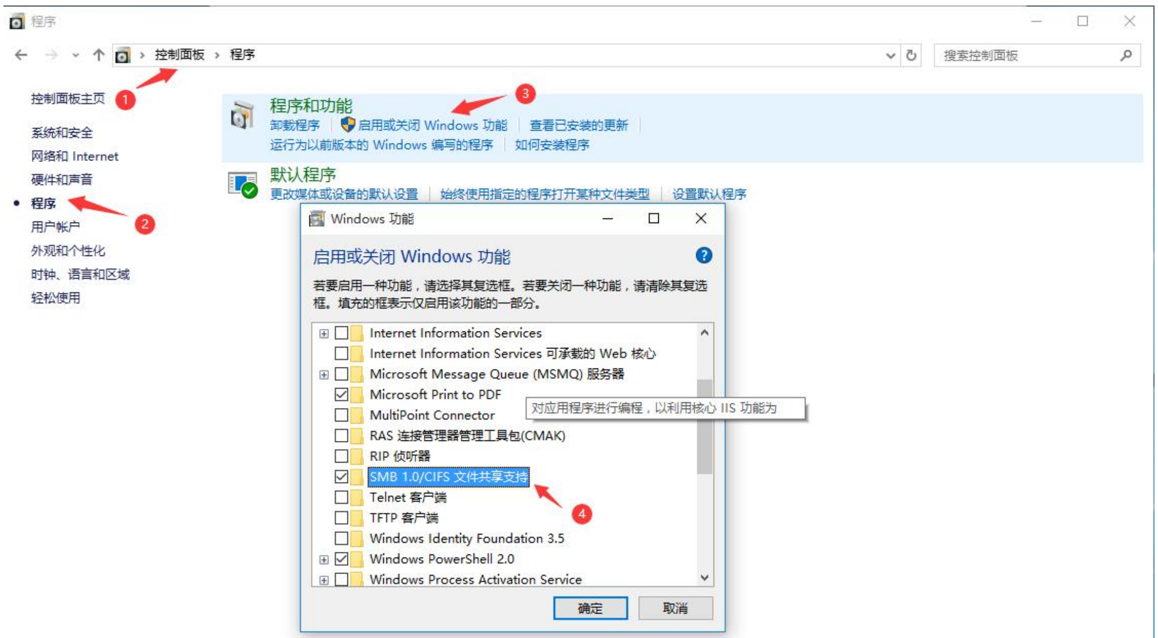
a) Enter the Control Panel of windows10

b) Click on "Programs" in the left navigation bar of the control panel

c) Select 'Enable or Disable Windows Features' in Programs and Features

d) Check 'SMB 1.0/CIFS file sharing support' in the pop-up box to enable or disable Windows features

e) Click 'OK' to configure the application



b. Open Windows10 Network Discovery

a) Enter the Control Panel of windows10

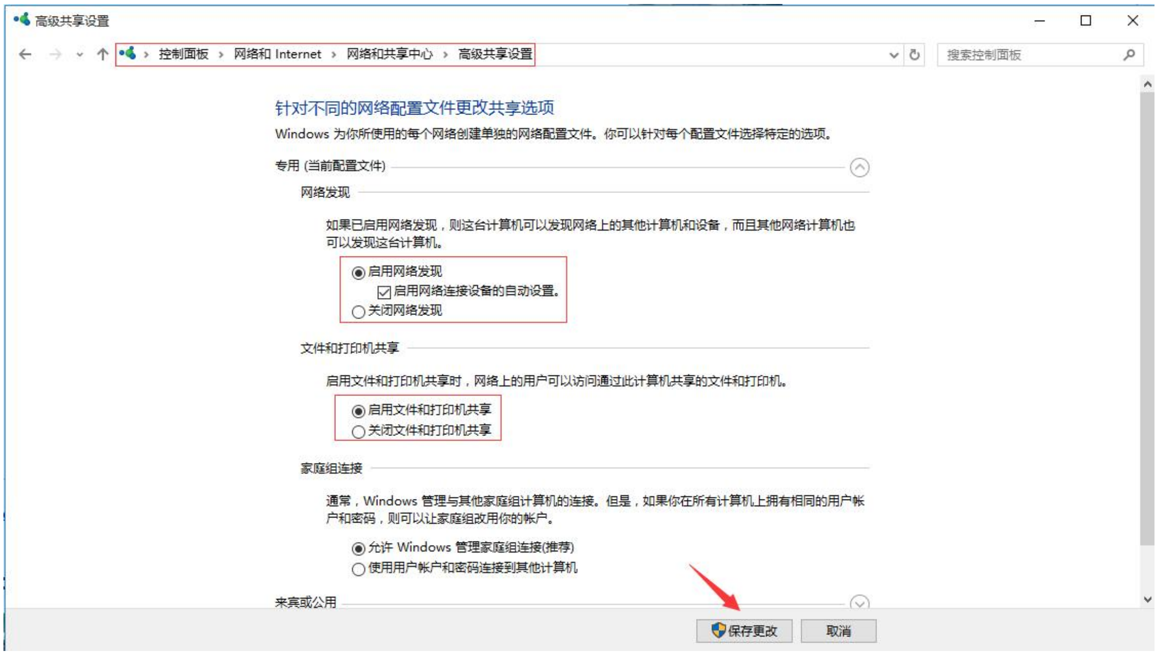
b) Select "Network and Internet" in the control panel

c) Then open the "Network and Sharing Center"

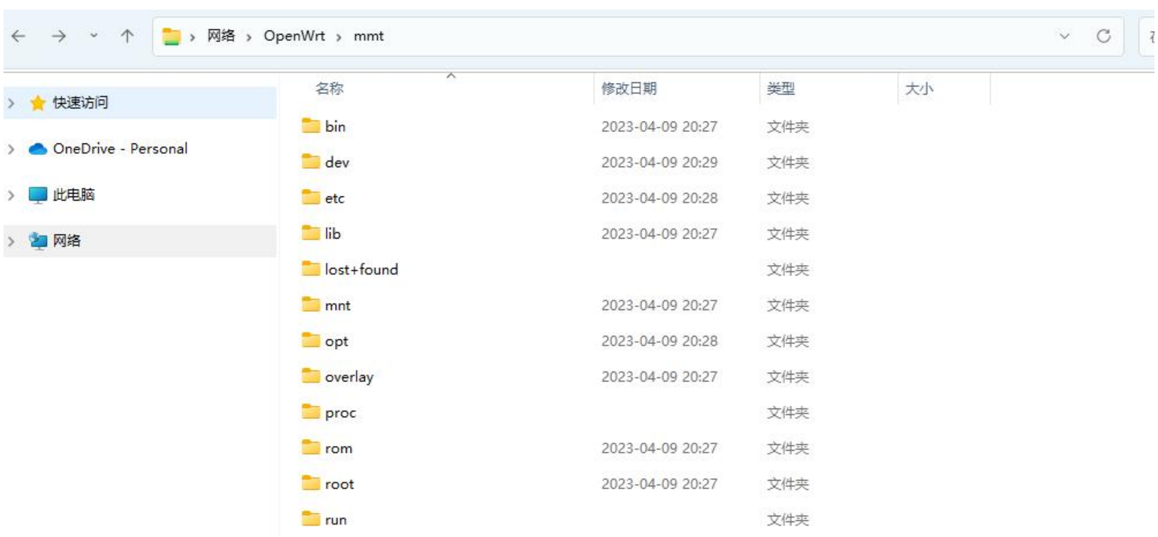
d) Click on 'Advanced Sharing Settings'

e) Open '**Enable Network Discovery**' and '**Enable File and Printer Sharing**'

f) Click 'Save Changes' to save the network discovery configuration for Windows10



5) After setting up, enter \\OpenWrt in the address bar of the resource manager to access the shared directory. The username is root and the password is the password set by the development board host

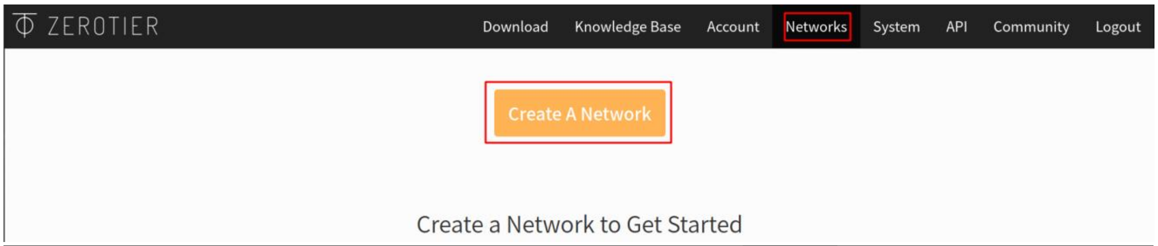


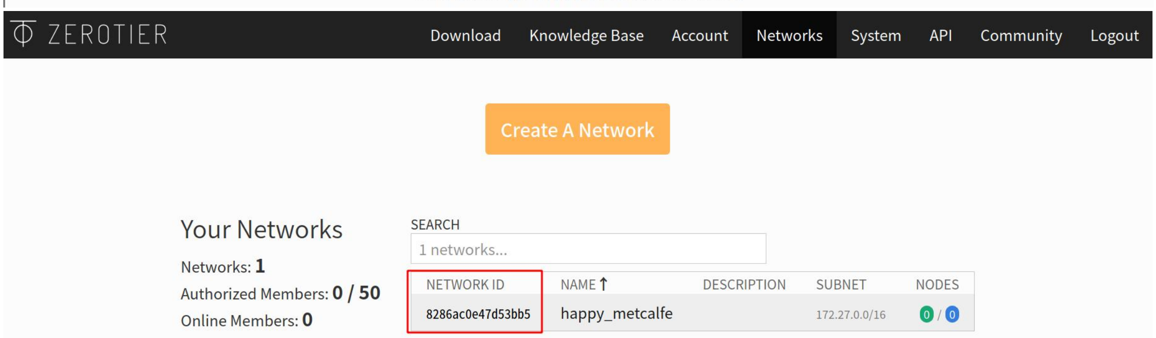
8. 12. zerotier User Manual

The OpenWRT system has pre installed the zerotier client. After creating a

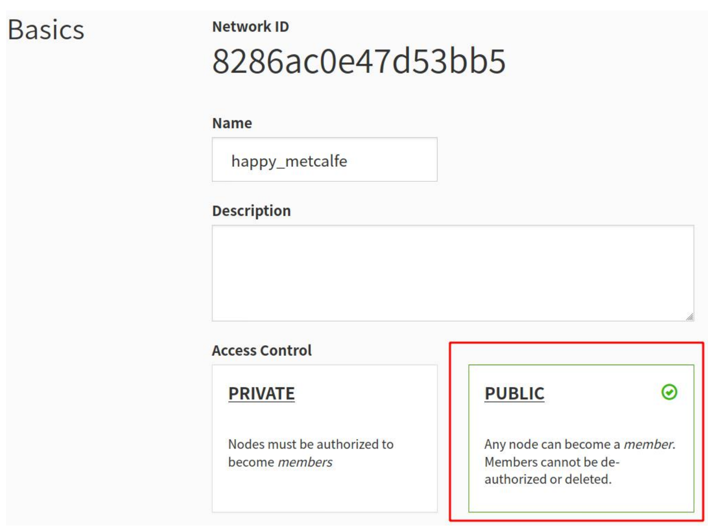
virtual LAN on the zerotier official website, the client can directly join it through the Network ID. The specific operation is shown below.

1) Log in to the zerotier official website https://my.zerotier.com/network After registering and logging in, click Network->Create A Network to create a virtual LAN

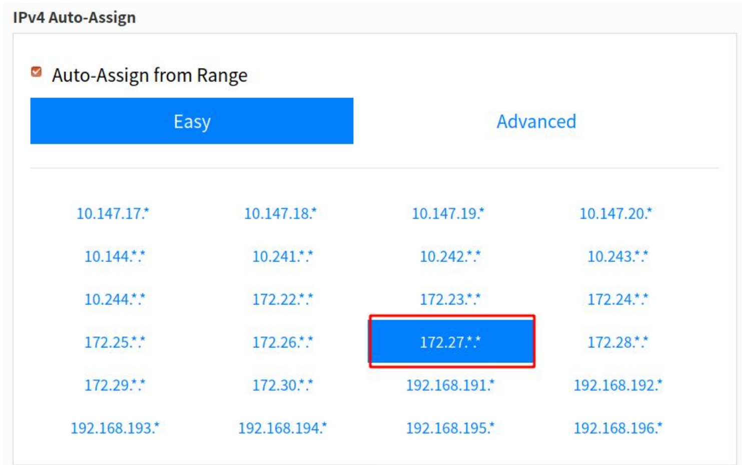




2) Click to enter the network console page, where you can set the privacy option to public, so that network nodes that join do not need to be verified



3) Below, the address will be automatically assigned. Here, you can choose your own network segment, and the selected one is 172.27.*.*



4) Enter the following command on the OpenWRT terminal to join the virtual LAN created above, **where 8286ac0e47d53bb5 is the Network ID of the virtual LAN created above**

```
root@OpenWrt:/# zerotier-one -d                          #Start the zerotier client
root@OpenWrt:/# zerotier-cli join 8286ac0e47d53bb5   #Join the network
```

5) By entering ifconfig on the terminal, it can be seen that there is already a newly added **ztks54inm2** device with an IP address of **172.27.214.213**

```
root@OpenWrt:/# ifconfig
ztks54inm2 Link encap:Ethernet  HWaddr F6:4E:DE:BF:D8:52
           inet addr:172.27.214.213  Bcast:172.27.255.255  Mask:255.255.0.0
           inet6 addr: fe80::e82f:d0ff:fe5a:867e/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:2800  Metric:1
           RX packets:18 errors:0 dropped:0 overruns:0 frame:0
           TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:1720 (1.6 KiB)  TX byte81 (8.2 KiB)
```

6) Install the zerotier client on another device (using Ubuntu 18.04 as an example), execute the following command to install, and restart the computer after installation is complete

```
test@ubuntu:~$ curl -s https://install.zerotier.com | sudo bash
```

7) After restarting, join the virtual LAN based on the Network ID, and you can also see

that the IP address assigned by zerotier has been obtained. At this time, the Ubuntu PC and OrangePi R1 Plus LTS are in the same LAN, and they can communicate freely

```
test@ubuntu:~$ sudo zerotier-cli join 8286ac0e47d53bb5
test@ubuntu:~$ ifconfig
ztks54inm2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2800
        inet 172.27.47.214  netmask 255.255.0.0  broadcast 172.27.255.255
        inet6 fe80::5ce1:85ff:fe2b:6918  prefixlen 64  scopeid 0x20<link>
        ether f6:fd:87:68:12:cf  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 46  bytes 10006 (10.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

8) Test whether two terminals can communicate

```
root@OpenWrt:/# ping 172.27.47.214 -I ztks54inm2
PING 172.27.47.214 (172.27.47.214): 56 data bytes
64 bytes from 172.27.47.214: seq=0 ttl=64 time=1.209 ms
64 bytes from 172.27.47.214: seq=1 ttl=64 time=1.136 ms
64 bytes from 172.27.47.214: seq=2 ttl=64 time=1.203 ms
64 bytes from 172.27.47.214: seq=3 ttl=64 time=1.235 ms
^C
--- 172.27.47.214 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.136/1.195/1.235 ms
```

9) Zerotier other commonly used commands

```
root@OpenWrt:/# zerotier-one -d                          #Start the zerotier client
root@OpenWrt:/# zerotier-cli status                      #Obtain address and service status
root@OpenWrt:/# zerotier-cli join # Network ID           #Join the network
root@OpenWrt:/# zerotier-cli leave # Network ID          #Leave the internet
root@OpenWrt:/# zerotier-cli listnetworks                #List networks
OPENWRT_DEVICE_REVISION="v0"
OPENWRT_RELEASE="OpenWrt 22.03.4 r20123-38ccc47687"
```

9. Compilation method of OpenWRT source code

9. 1. Download OpenWRT source code

1) First, execute the following command to download the openwrt-22.03 branch code

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install -y git
test@test:~$ git clone https://github.com/orangepi-xunlong/openwrt.git -b openwrt-22.03
```

2) After downloading the OpenWRT code, the following files and folders will be included

```
test@test:~/openwrt$ ls
BSDmakefile   Config.in   include   Makefile   README.md   scripts   toolchain
Config   feeds.conf.default   LICENSE   package   rules.mk   target   tools
```

9. 2. Compile OpenWRT source code

1) Firstly, install the following dependency packages (currently only tested for compilation on Ubuntu20.04. If compiling on other versions of the system, please install the dependency packages yourself according to the error message)

a. Method 1: The command to install dependency packages using a script is as follows:

```
test@test:~/openwrt$ sudo ./install_dep.sh
```

b. Method 2: Install dependency packages directly using the following command

```
test@test:~/openwrt$ sudo apt update
test@test:~/openwrt$ sudo apt install -y ack antlr3 asciidoc autoconf    \
automake autopoint binutils bison build-essential    \
bzip2 ccache cmake cpio curl device-tree-compiler fastjar \
flex gawk gettext gcc-multilib g++-multilib git gperf haveged \
help2man intltool libc6-dev-i386 libelf-dev libglib2.0-dev \
libgmp3-dev libltdl-dev libmpc-dev libmpfr-dev \
libncurses5-dev \libncursesw5-dev libreadline-dev libssl-dev \
libtool lrzsz mkisofs msmtp nano ninja-build p7zip p7zip-full \
```

```
patch pkgconf python2.7 python3 python3-pyelftools \
libpython3-dev qemu-utils rsync scons squashfs-tools \
subversion swig texinfo uglifyjs upx-ucl unzip \
vim wget xmlto xxd zlib1g-dev
```

2) Then execute **./scripts/feeds update -a** and **./scripts/feeds install -a** download dependency package

```
test@test:~/openwrt$ ./scripts/feeds update -a
test@test:~/openwrt$ ./scripts/feeds install -a
```

3) Then choose to use the configuration file of OrangePi CM5

```
test@test:~/openwrt$ cp configs/orangepi-cm5-rk3588_defconfig .config
```

4) Then execute the following command to make the configuration effective

```
test@test:~/openwrt$ make defconfig
```

5) Execute the following command to start compiling the openwrt source code

```
test@test:~/openwrt$ make V=s
```

6) After compilation, the path where the image is generated is:

```
test@test:~/openwrt$ tree -L 1 bin/targets/rockchip/armv8/
bin/targets/rockchip/armv8/
├── config.buildinfo
├── feeds.buildinfo
├── openwrt-rockchip-armv8-xunlong_orangepi-cm5-ext4-sysupgrade.img.gz
├── openwrt-rockchip-armv8-xunlong_orangepi-cm5.manifest
├── openwrt-rockchip-armv8-xunlong_orangepi-cm5-squashfs-sysupgrade.img.gz
├── packages
├── profiles.json
├── sha256sums
└── version.buildinfo

1 directory, 8 files
```

10. Appendix

10. 1. User Manual Update History

Version	Date	Release Notes
v1.0	2024-07-02	initial version
v1.1	2024-07-04	1. Linux：Method for Running RKLLM Large Model with RK3588 2. Instructions for using Orange Pi OS Arch system 3. Compilation method of Android 13 source code
v1.2	2024-07-10	1. Usage of NPU 2. RK3588 method of using PaddlePaddle
v1.3	2024-07-19	1. OpenWRT system usage instructions 2. The compilation method of OpenWRT source code

10. 2. Image update history

Date	Release Notes
2024-07-02	Orangepicm5_1.0.0_ubuntu_focal_server_linux5.10.160.7z Orangepicm5_1.0.0_ubuntu_jammy_server_linux5.10.160.7z Orangepicm5_1.0.0_debian_bullseye_server_linux5.10.160.7z Orangepicm5_1.0.0_debian_bookworm_server_linux5.10.160.7z Orangepicm5_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepicm5_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepicm5_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepicm5_1.0.0_debian_bookworm_desktop_xfce_linux5.10.160.7z Orangepicm5_1.0.0_ubuntu_jammy_server_linux6.1.43.7z Orangepicm5_1.0.0_debian_bookworm_server_linux6.1.43.7z Orangepicm5_1.0.0_ubuntu_jammy_desktop_xfce_linux6.1.43.7z Orangepicm5_1.0.0_debian_bookworm_desktop_xfce_linux6.1.43.7z OrangePiCM5_RK3588S_Android13_v1.0.0.tar.gz

	* initial version
2024-07-04	Opios-arch-aarch64-gnome-opicm5-24.07-linux5.10.160.img.xz Opios-droid-aarch64-opicm5-24.07-linux5.10.160.tar.gz * Initial version OrangePiCM5_RK3588S_Android13_v1.0.1.tar.gz * Pre installed gms
2024-07-19	openwrt-rockchip-armv8-xunlong_orangepi-cm5-ext4-sysupgrade_v1.0.img.gz * Initial version