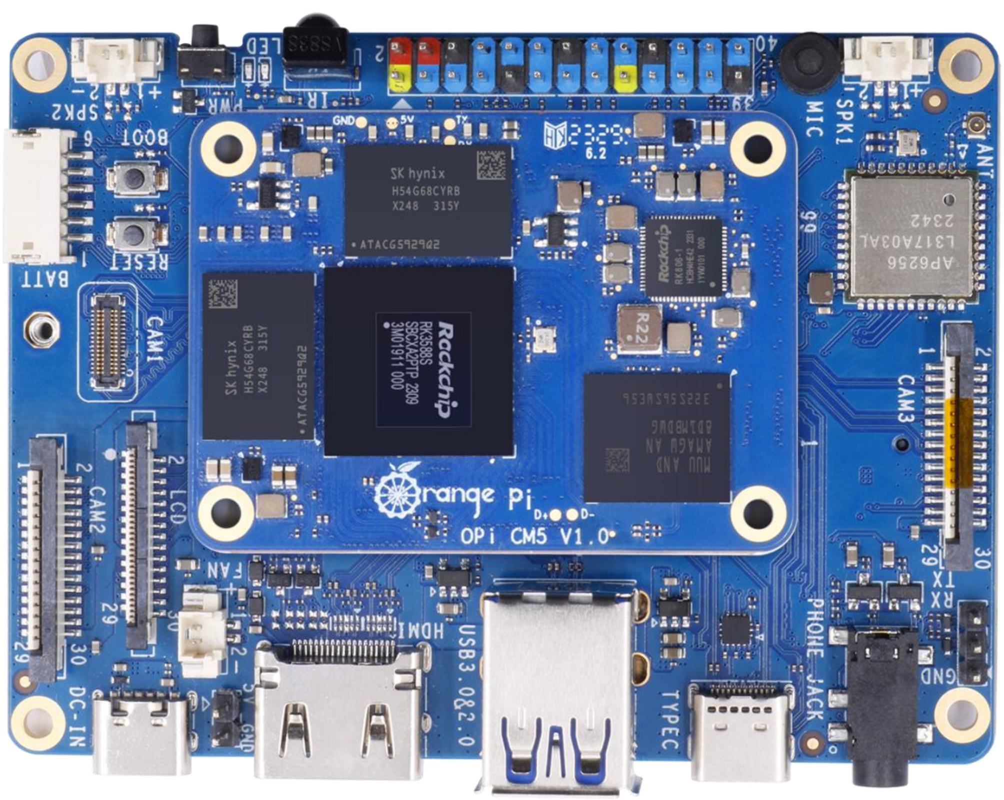
Orange Pi CM5 Base Tablet User Manual



Catalogue

1. Basic features of Orange Pi CM5 Base Tablet

1. 1. What is Orange Pi CM5 Base Tablet

Orange Pi CM5 core board adopts the Rockchip RK3588S new generation eight core 64 bit ARM processor, specifically the quad core A76 and quad core A55, using Samsung's 8nm LP process technology. The large core frequency can reach up to 2.4GHz, integrated with ARM Mali-G610 MP4 GPU, embedded with high-performance 3D and 2D image acceleration modules, and built-in AI accelerator NPU with up to 6 Tops computing power. It has 4GB/8GB/16GB (LPDDR5) memory and up to 8K display processing capability. In addition, the core board is also equipped with onboard eMMC, with optional capacities of 32GB/64GB/128GB/256GB.

Orange Pi CM5 Base Tablet motherboard offers a wide range of interfaces, including HDMI output Wi-Fi, Bluetooth, M.2 PCIe 2.0x1, MIPI CSI, MIP DSI, USB 2.0, USB 3.1 interfaces, and 26 pin extension pins. It can be widely used in high-end tablets, edge computing, artificial intelligence, cloud computing AR/VR、 In the fields of intelligent security and smart home, covering various industries of AIoT.

Orange Pi CM5 Base Tablet supports Orange Pi OS, the official operating system developed by Orange Pi, as well as Android 13, Debian11, Debian12, Ubuntu 20.04, and Ubuntu 22.04 operating systems.

1. 2. Purpose of Orange Pi CM5 Base Tablet

We can use it to achieve:

- A Linux desktop computer.
- Android tablet.
- Android game consoles, etc.

Of course, there are many other features as well. With a powerful ecosystem and various expansion accessories, Orange Pi can help users easily achieve delivery from creativity to prototype to mass production. It is an ideal creative platform for makers, dreamers, and hobbyists.

1. 3. Hardware Characteristics of Orange Pi CM5 Core Board

Hardware specifications of OPi CM5 core board	
Master 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 computing power, supporting INT4/INT8/INT16 mixed operations
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)), Includes 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 SDIO 3.0 or RGMII I2C、I2S、UART、SPI、CAN、PWM、PDM、GPIO etc POWER_ON、RESET、MASKROM、RECOVERY and other buttons

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	

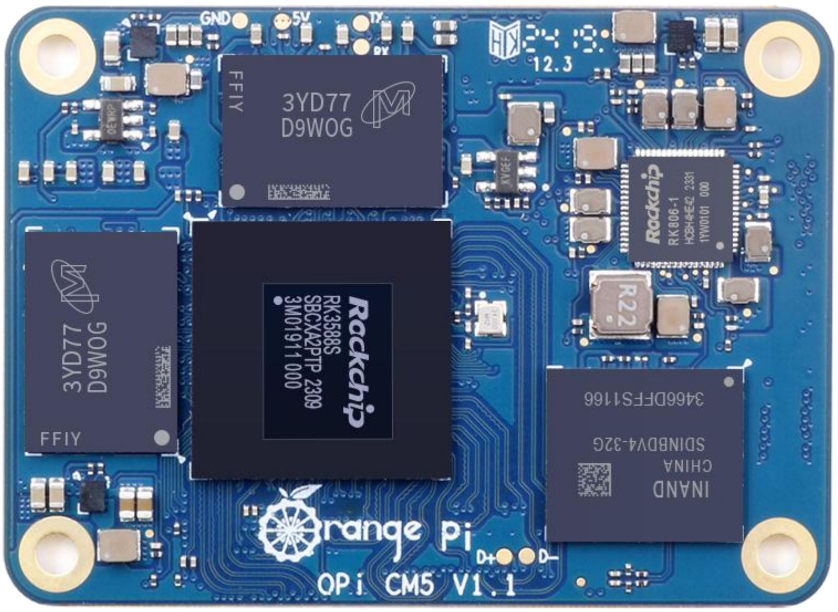
1.4. Hardware characteristics of Orange Pi CM5 Base Tablet motherboard

Orange Pi CM5 base-tablet hardware features	
Board to board connector	3 * 100PIN, 0.4mm PIN spacing, connector model: df40c-100ds-0.4v
Storage	• MicroSD (TF) slot • M.2 M-KEY slots: Supports NVMe PCIe 2.0 or SATA SSD
USB	• Type-C USB3.0 • USB3.0+USB2.0
Video output	• HDMI 2.1， Highest support 8K@60Hz • DP 1.4 ALT 4Lane (Shared TYPE-C interface） • MIPI DPHY TX 4 Lane
Camera	• 1*MIPI CSI 4 Lane • 2*MIPI DPHY RX 2 Lane
Audio	• 3.5mm headphone jack audio input/output • HDMI audio output • Onboard MIC audio input • 2 * speaker socket, specification 2PIN 1.25mm
WIFI+BT	• module： AP6256 • Onboard WI-FI5+BT 5.0
26PI Nextension interface	• Double row pin specifications: 2.54mm spacing • Supports UART, PWM, I2C, SPI, GPIO and other functions
key	1*MASKROM、1*RESET and 1*PWR
power supply	Supports Type-C power supply, 5V@5A
Battery holder	Battery holder: board to wire connection, 6PIN, 1.5mm spacing, connected to a single lithium battery or lithium polymer battery,

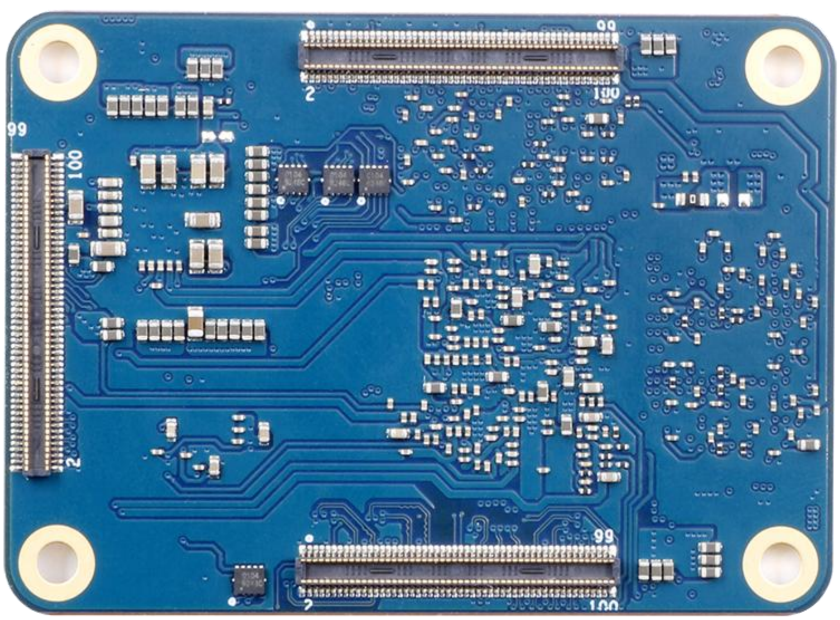
	maximum charging current 5A
RTC	Reserved solder pad, capable of soldering 3V spare RTC battery
Key interface	6PIN FPC socket, PIN distance 0.5mm
sensor interface	6PIN FPC socket, PIN distance 0.5mm
LED	Red power indicator light and green status indicator light
FAN	5V 2PIN 1.25mm spacing
SPK	2*2PIN 1.25mm spacing
Debug UART	3Pin debugging serial port, PIN distance 2.54mm
PCB	Length: 90mm, width: 66mm, thickness:1.6mm
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:

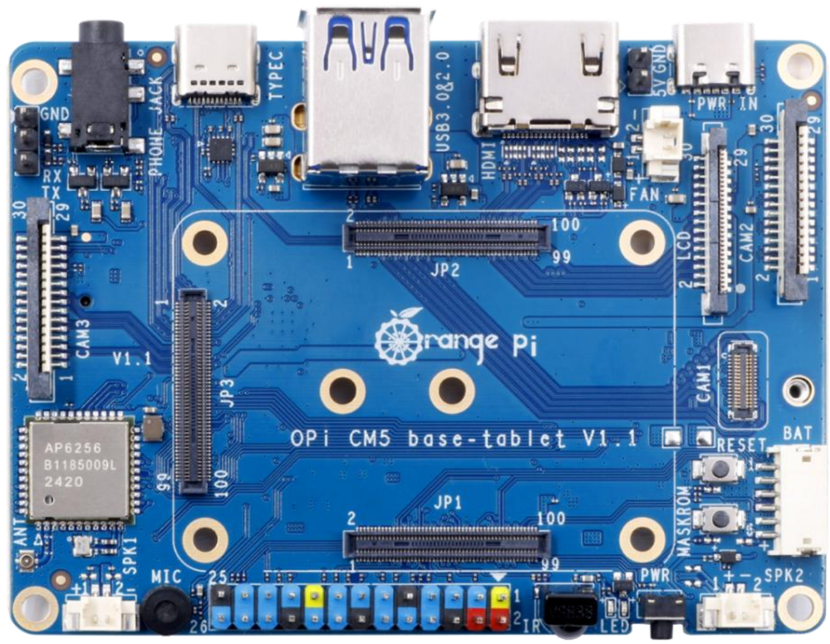


Bottom view:

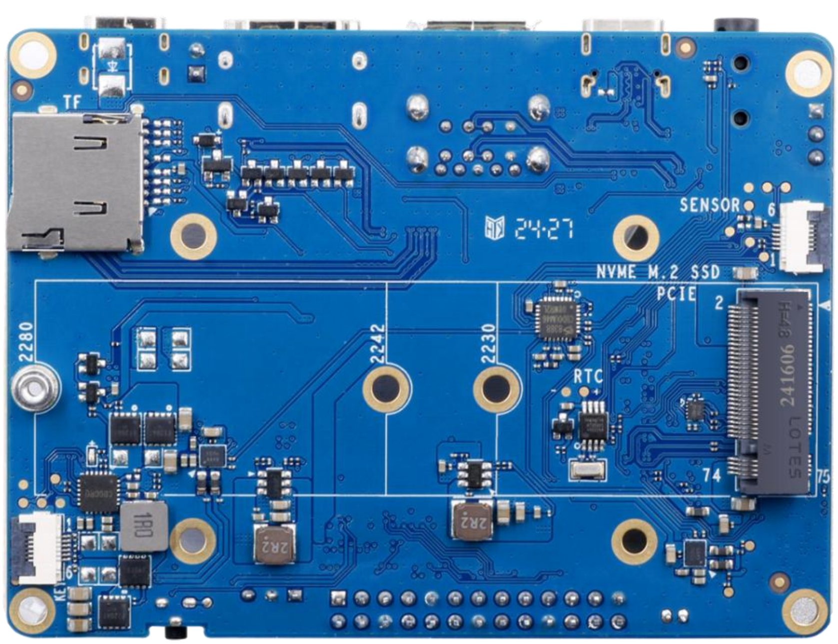


1.6. Orange Pi CM5 Base Tablet Top View and Bottom View

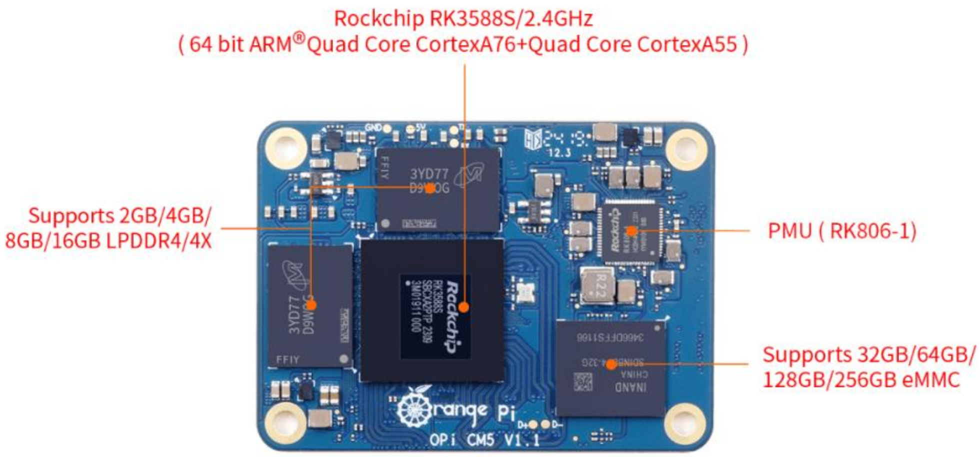
top view:



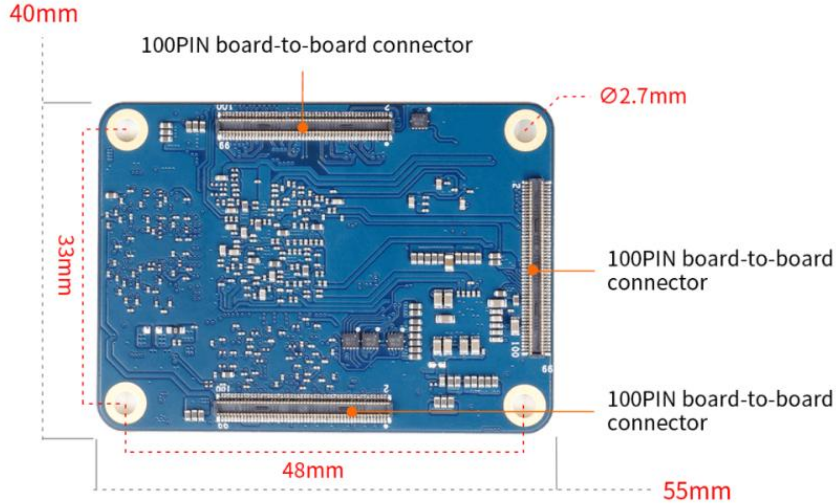
Bottom view:



1.7. Interface Details of Orange Pi CM5 Core Board

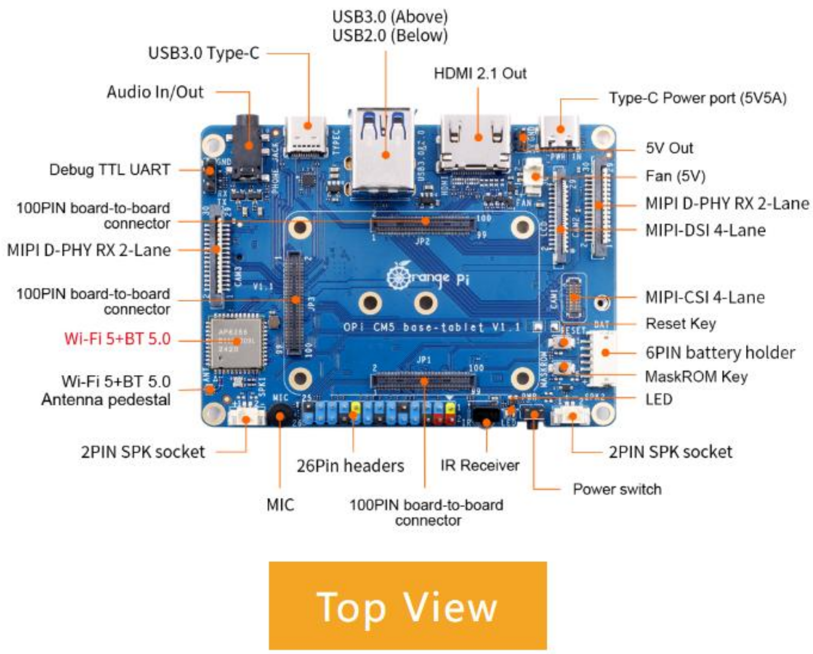


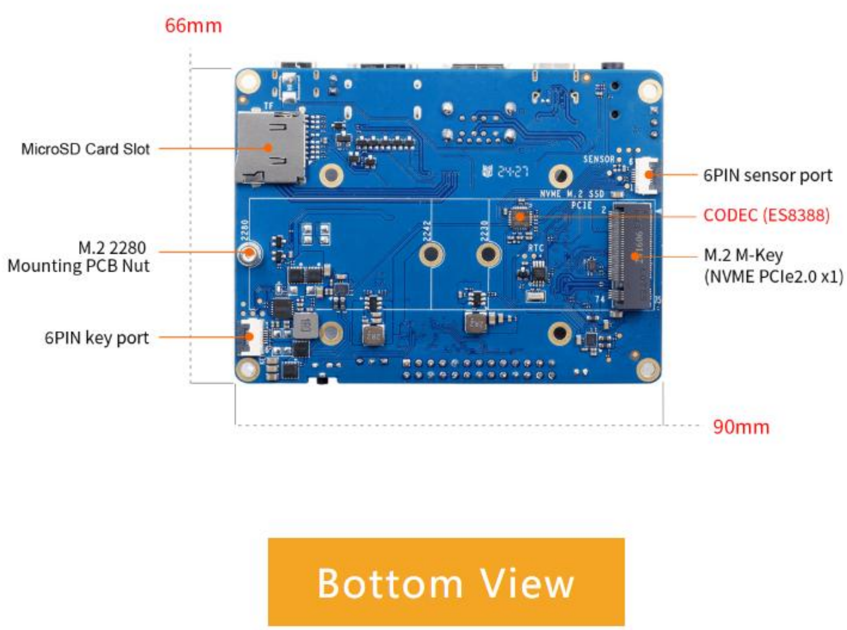
Top View

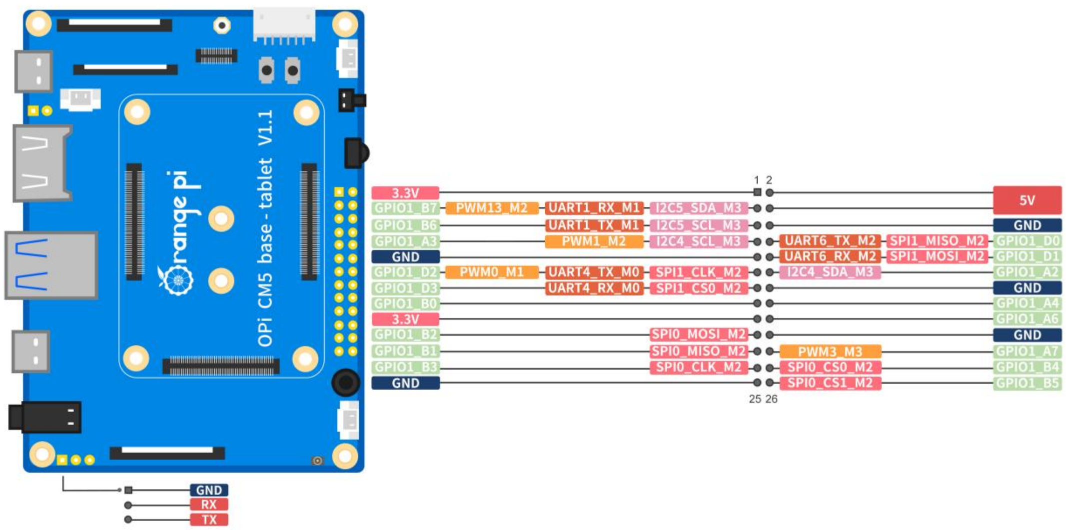


Bottom View

1. 8. Orange Pi CM5 Base Tablet motherboard interface details diagram







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) 10.1-inch MIPI screen, used to display the system interface of the development board (this screen includes an adapter board, which is compatible with OPi5Plus/OPi5/OPiCM5BaseTablet).

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

There are two identical Type-C interfaces on the motherboard, the one on the right is the power interface, and the one on the left does not have power supply function. Please do not connect them incorrectly.

The Type-C power interface of the development board does not support PD negotiation function and only supports a 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.

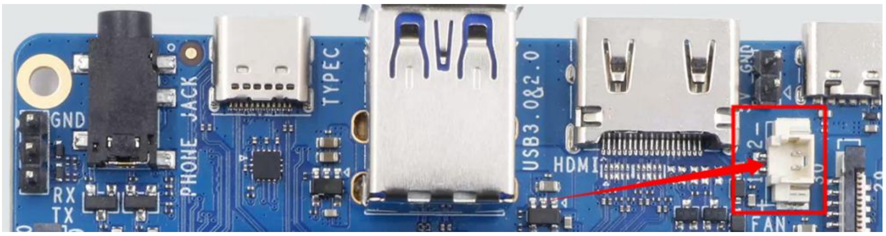
8) USB camera.

9) Infrared remote control.

Note that the 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.

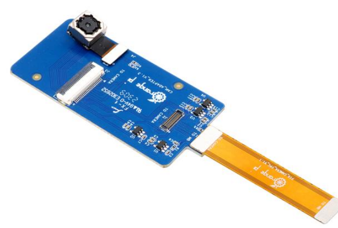


11) Type-C to HDMI cable, connect the development board to an HDMI monitor or TV for display through the Type-C interface.

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

13) OV13850 camera with 13 million MIPI interface.

14) OV13855 camera with 13 million MIPI interface.

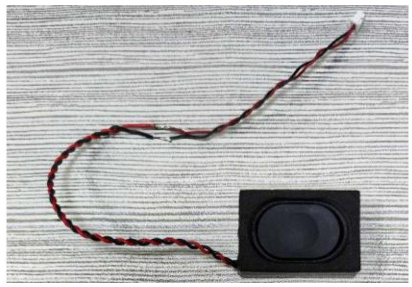


15) Type-C interface data cable, used for burning images, using ADB and other functions.

16) The NVMe SSD solid state drive with M.2 M-KEY 2280 specification has a PCIe interface specification of PCIe 2.0x1.

17) M.2 M-KEY 2280 specification SATA SSD solid state drive.

18) Speaker, interface is 2pin, with a spacing of 1.25mm.



The positions of the two speaker interfaces on the base plate are shown in the following figure:

19) 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.

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

1	Ubuntu22.04 PC	Optional, used for compiling Linux source code
2	Windows PC	Used for burning Android and Linux images

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

1) The download link for the Chinese version of the materials is：

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

2) The download link for the English version of the material is：

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

3) The information mainly includes：

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

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

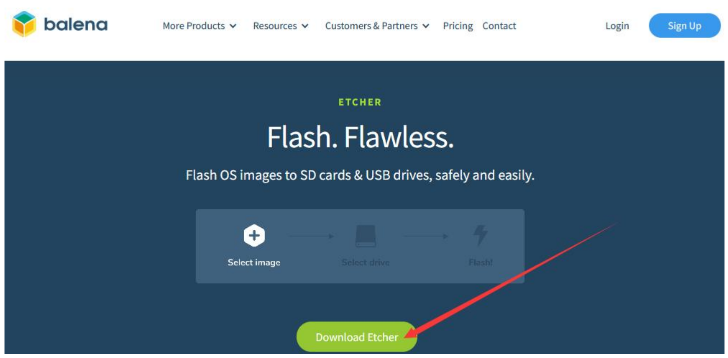
2) Then use the card reader to insert the TF card into the computer

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

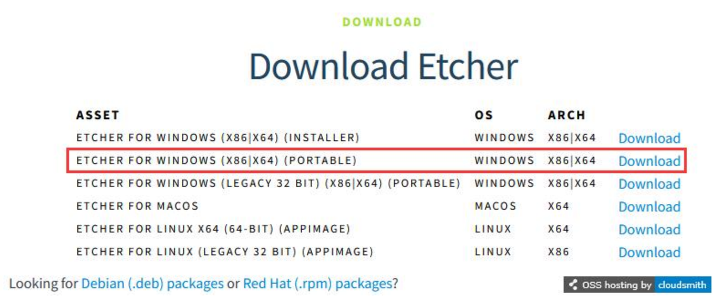
4) Then download the Linux image burning software - **balenaEtcher**, from:

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

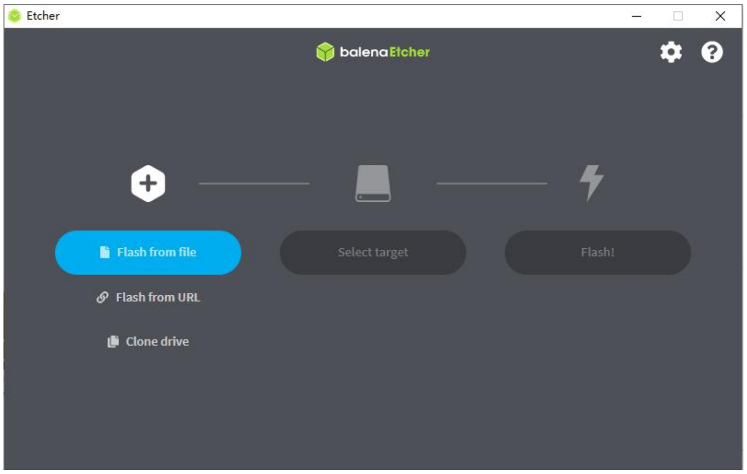
5) After entering the BalenaEtcher download page, clicking the green download button will redirect you to the software download location



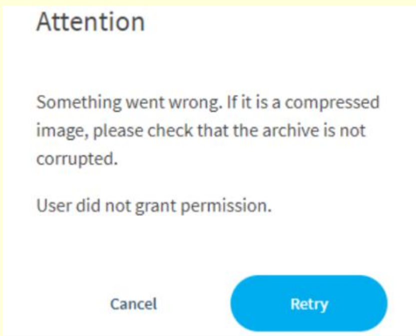
6) Then you can choose to download the Portable version of BalenaEtcher software. The Portable version does not need to be installed, and can be opened by double clicking to use it



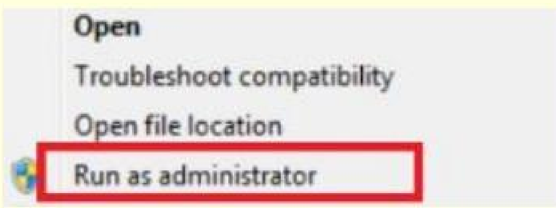
7) If you are downloading a version of BalenaEtcher that requires installation, please install it before using it. If you download the Portable version of balenaEtcher, simply double-click to open it. The interface of balenaEtcher after opening is shown in the following figure:



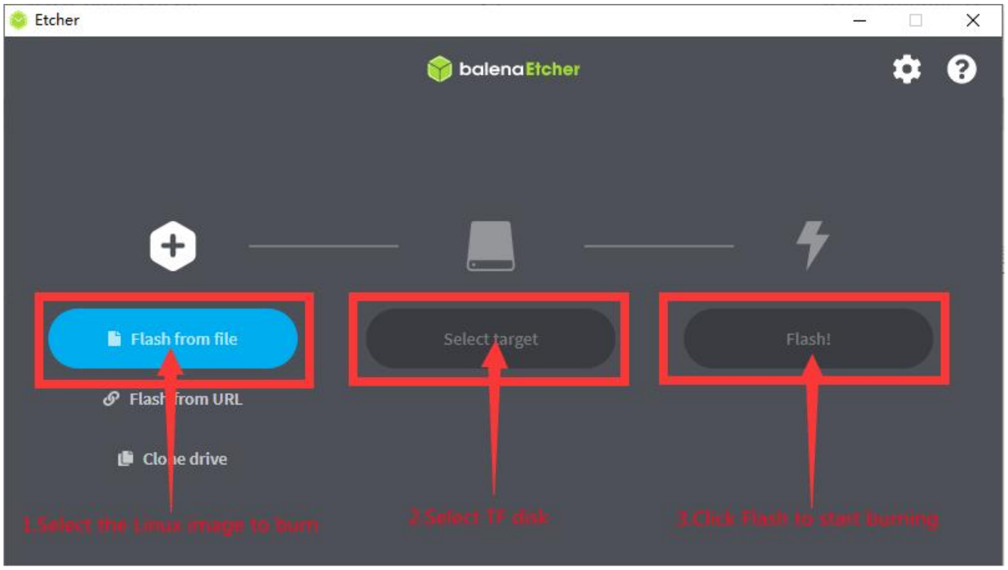
If the following error is prompted when opening balenaEtcher:



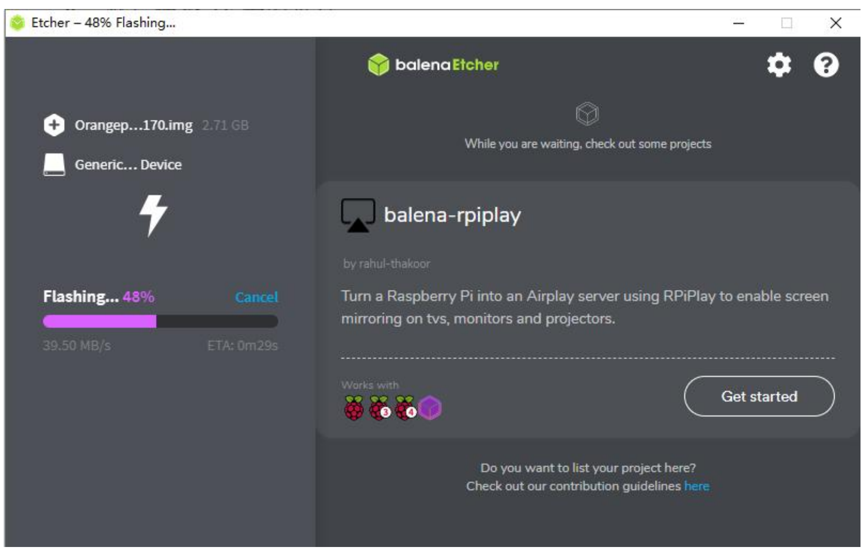
Please select balenaEtcher, right-click, and then select Run as administrator.



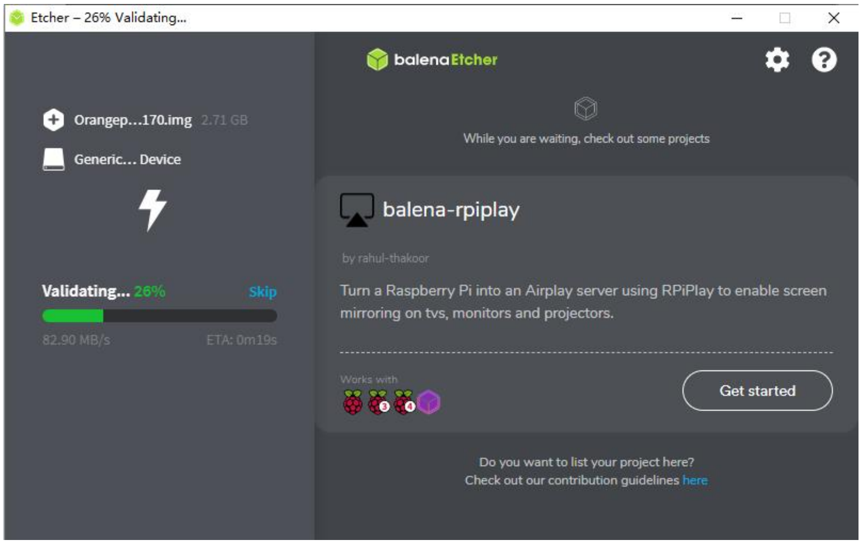
8) The specific steps to use balenaEtcher to burn the Linux image are aa follow
   a. First select the path of the Linux image file to burned
   b. Then select the drive letter of the TF card
   c. Finally, click Flash to start burning the Linux image to the TF card



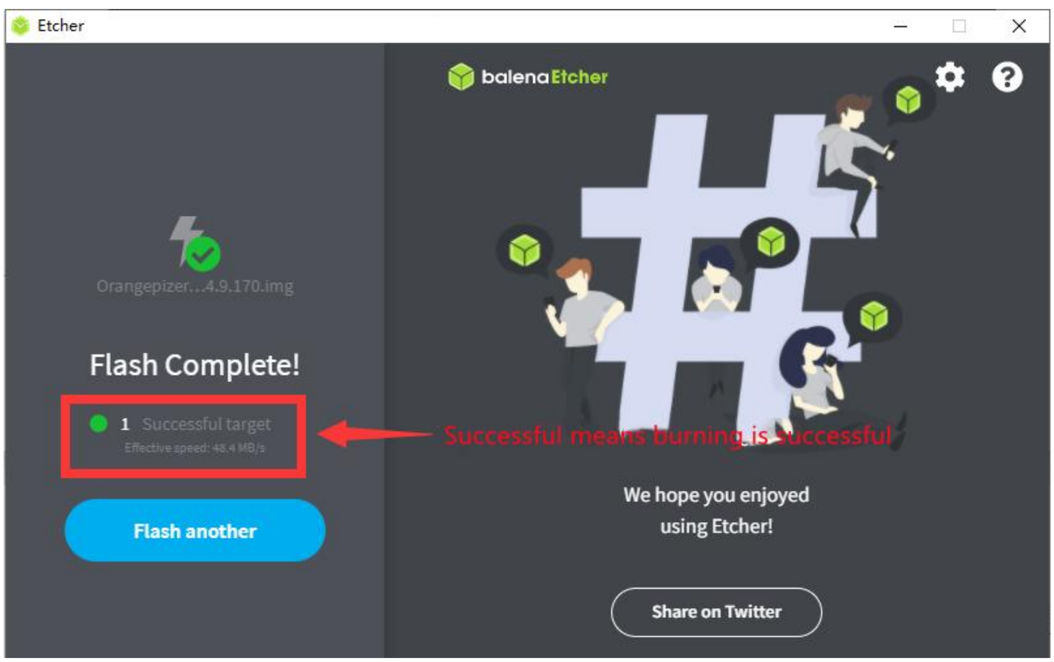
9) The interface displayed in the process of burning the Linux image by balenaEtcher is shown in the figure below, and the progress bar displays purple, indicating that the Linux image is being burned into the TF card



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

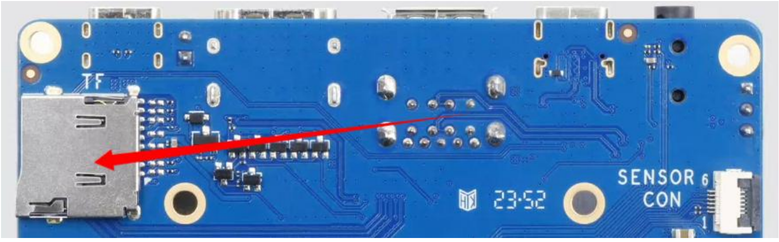


2. 3. 2. Method of burning Linux images to TF card using RKDevTool

1) Firstly, it is necessary to prepare a high-quality Type-C interface data cable.

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

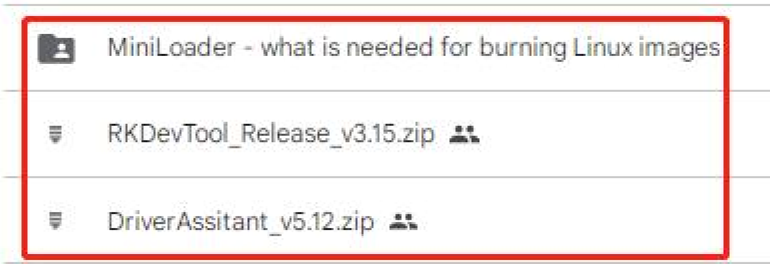
3) Then insert the TF card into the card slot of the development board



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



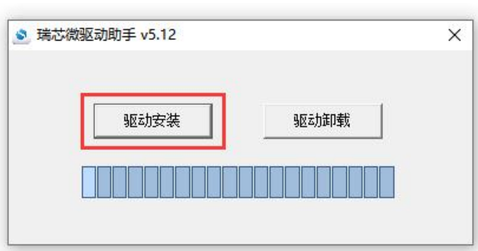
Note that the "MiniLoader - what is needed for burning Linux images" folder is hereinafter referred to as the MiniLoader folder.

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

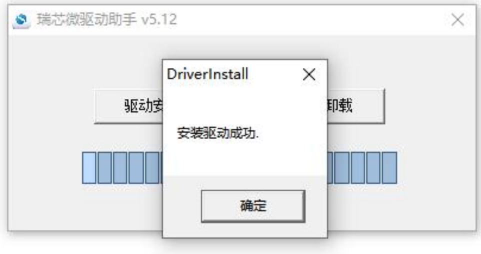
6) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**. zip, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

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.

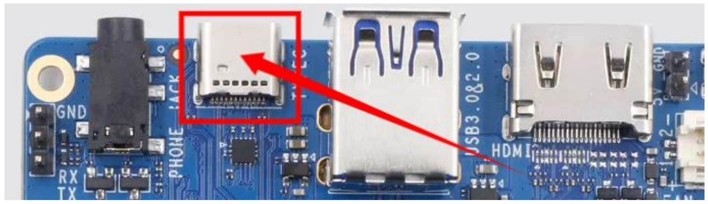


8) Then decompress **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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 onto the TF card.

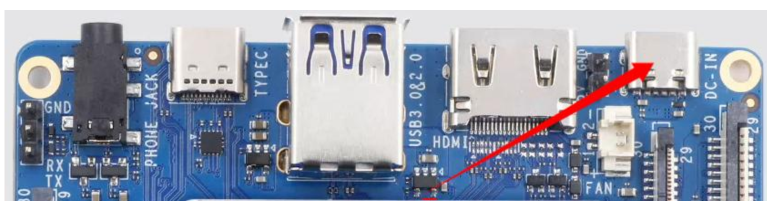
a. Firstly, connect the motherboard to the Windows computer via a Type-C data cable. The location of the Type-C interface on the development board is shown in the following figure:



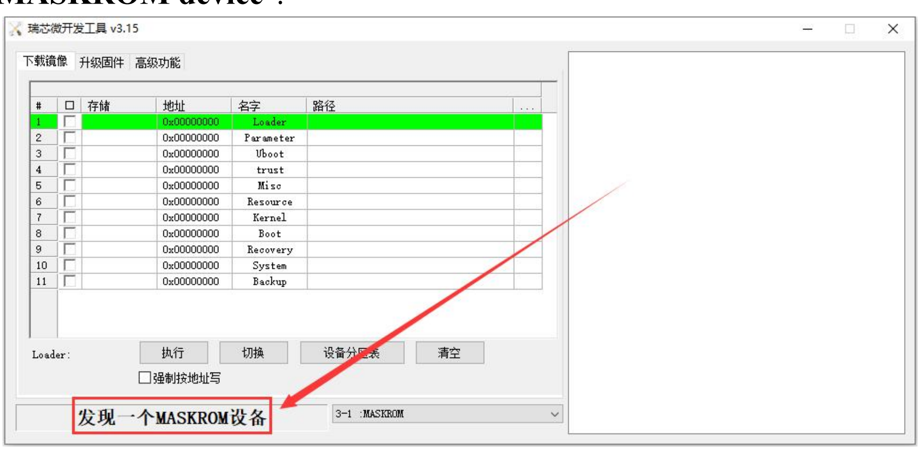
b. Ensure that the development board is not connected to a Type-C power supply.

c. Then hold down the MaskROM button on the development board and do not release it. The position of the MaskROM button on the development board is shown in the following figure:

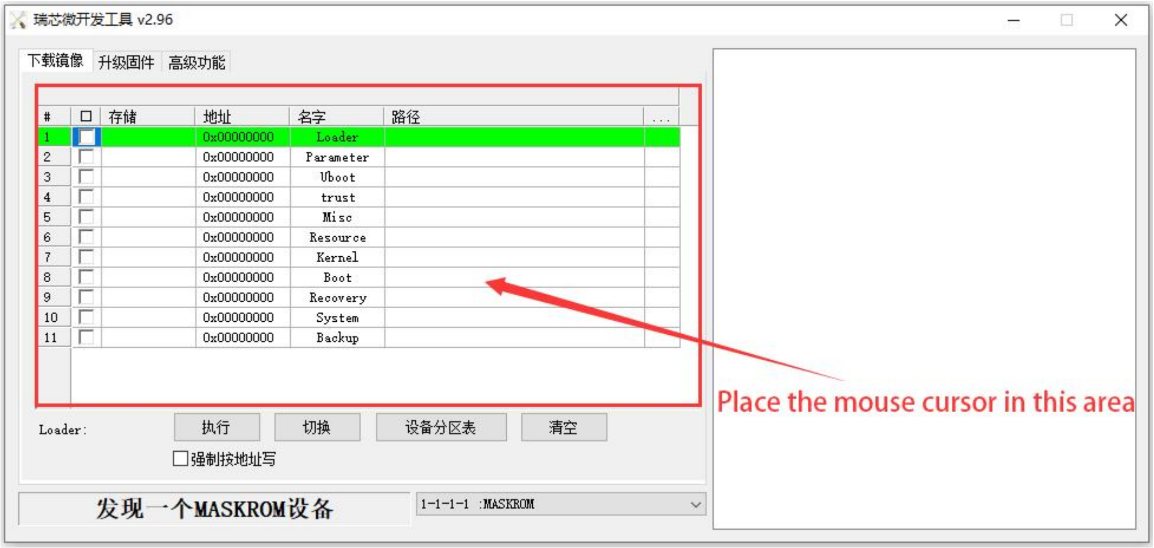
d. Then connect the Type-C interface power to the development board, power it on, and then release the MaskROM button. The location of the Type-C power interface is as follows:



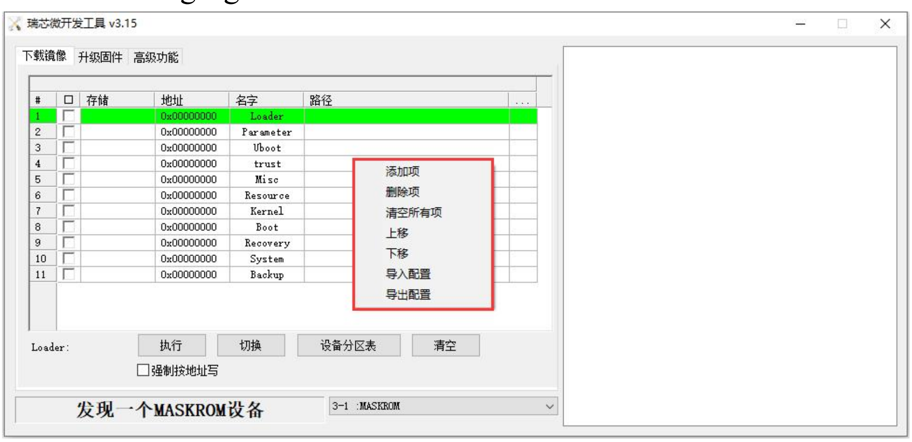
e. If the previous steps are successful, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**Found a MASKROM device**".



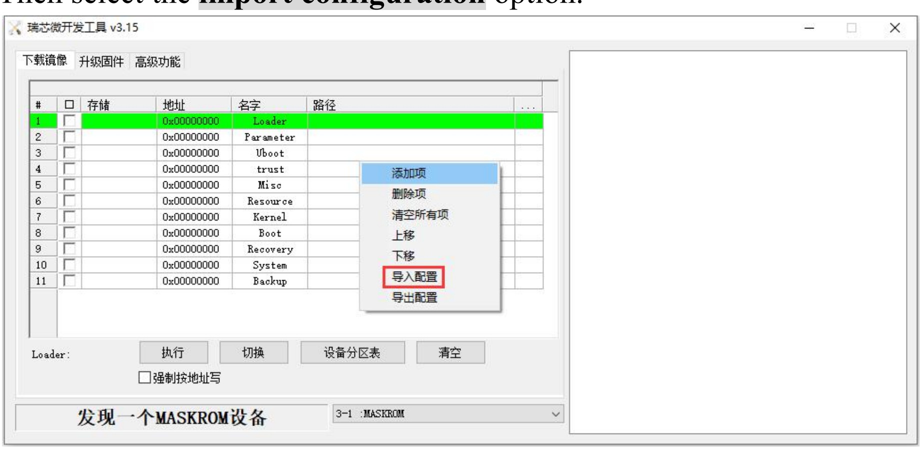
f. Then place the mouse cursor in the area below.



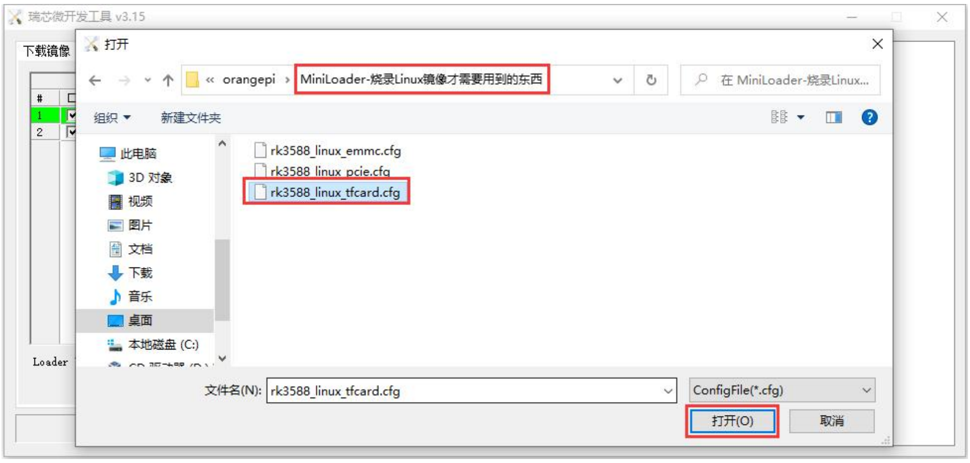
g. Then, clicking the right mouse button will bring up the selection interface shown in the following figure.



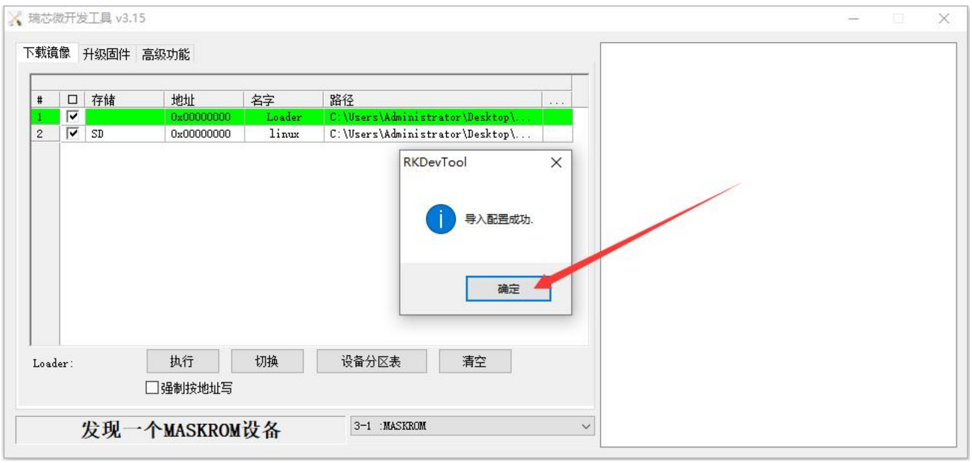
h. Then select the **import configuration** option.



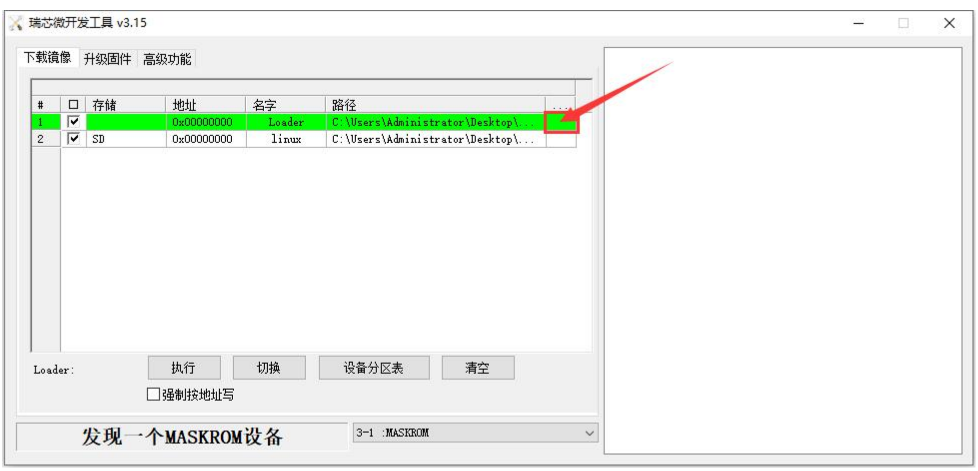
i. Then select the **rk3588_linux_tfcard.cfg** configuration file from the **MiniLoader** folder downloaded earlier, and click **open**.



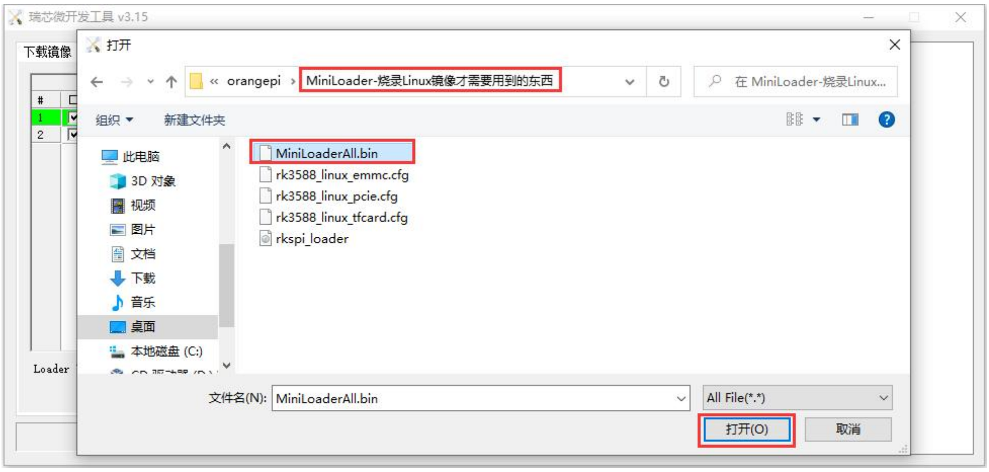
j. Then click **OK**.



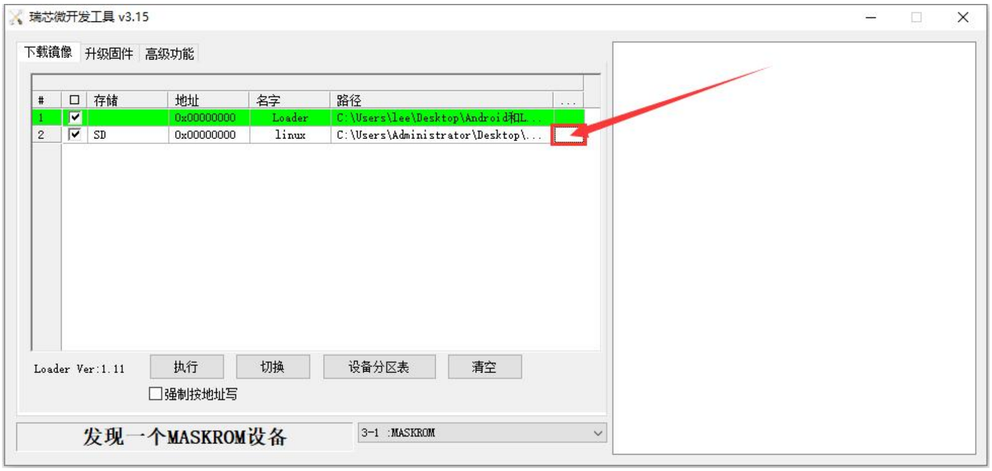
k. Then click on the location shown in the figure below.



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

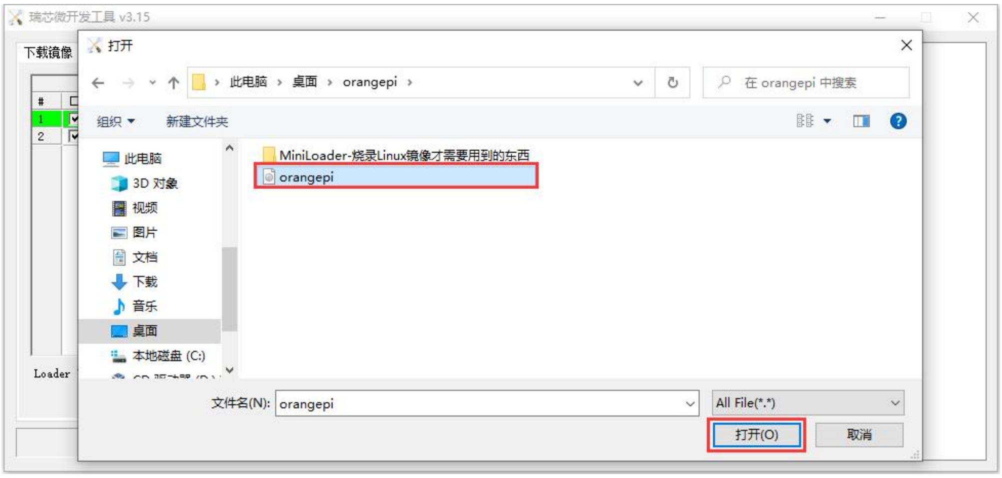


m. Then click on the location shown in the figure below.

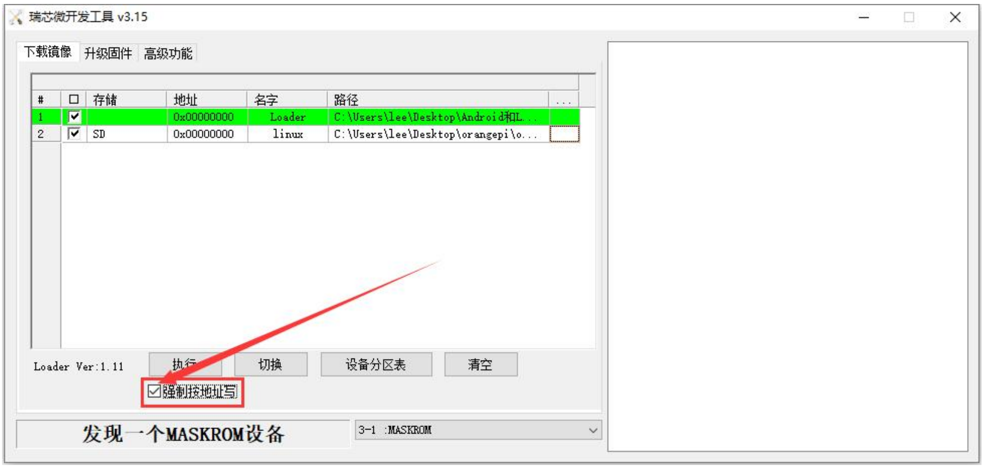


n. Then select the path to the Linux image you want to burn, and click **Open**.

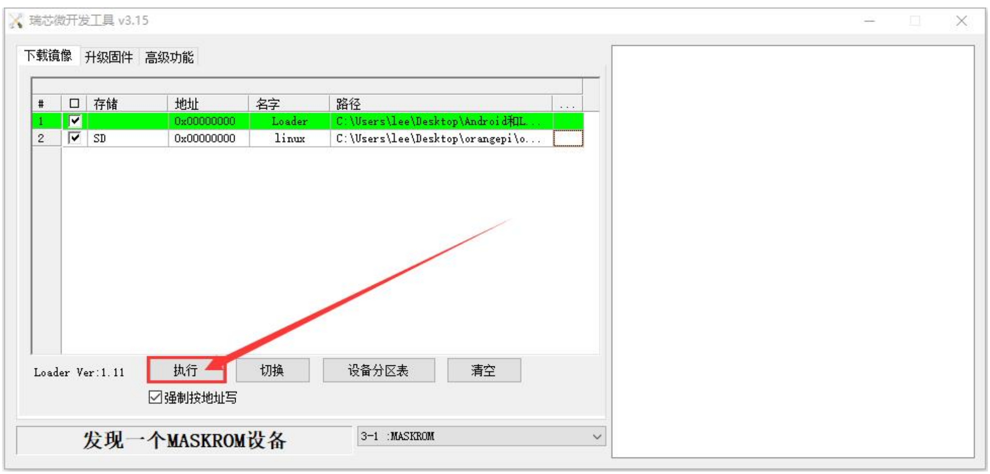
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



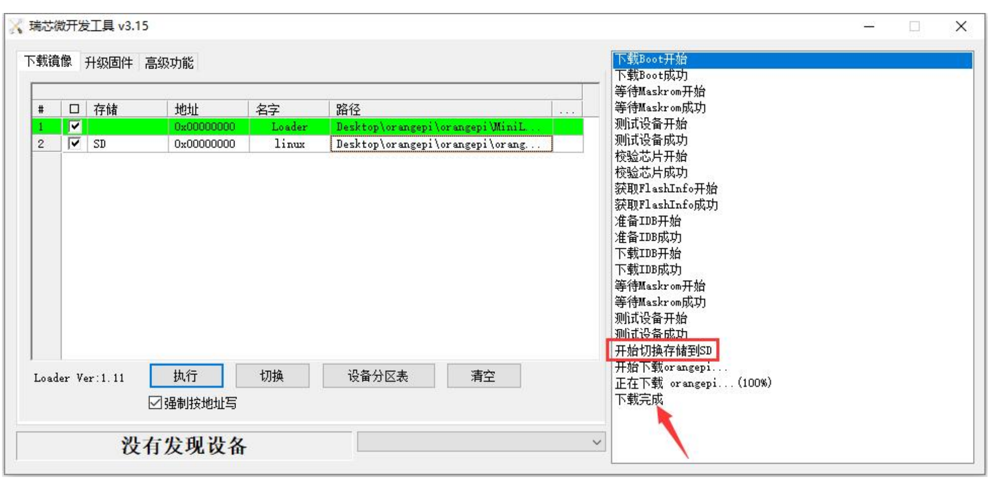
o. Then please check the option to **force writing by address**.



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.

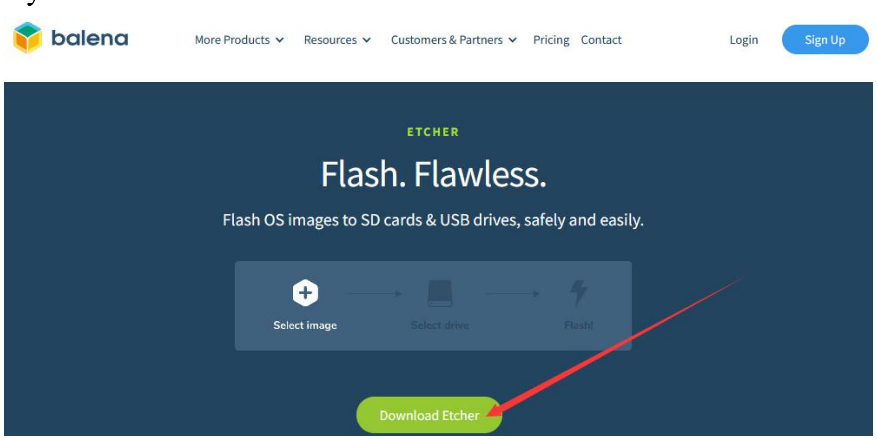
1) First prepare a TF card with a capacity of 16GB or more. The transmission speed of the TF card must be **class 10** or above. It is recommended to use a TF card of SanDisk and other brand

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

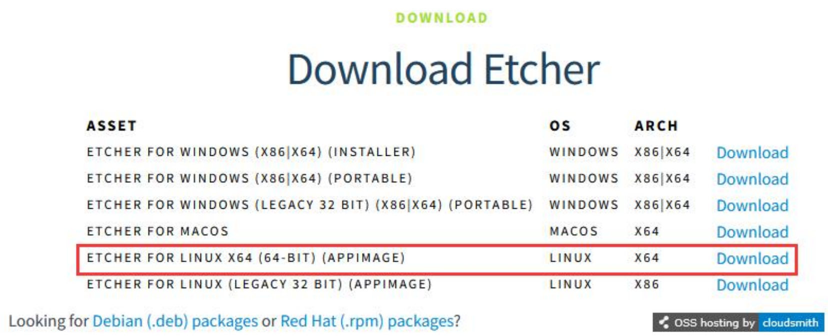
3) Download the balenaEtcher software, the download address is:

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-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
test@test:~$ ls orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.*
orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z
orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.sha     #Checksum file
orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img     #image 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-tablet-24.03-linux-6.1.43-ext4.img.gz
test@test:~$ ls openwrt-aarch64-opicm5-tablet-24.03-linux-6.1.43-ext4.img
openwrt-aarch64-opicm5-tablet-24.03-linux-6.1.43-ext4.img     #image file
```

7) After decompressing the image, you can first use the **sha256sum -c *.sha** command to calculate whether the checksum is correct. If the prompt is **successful**, it 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

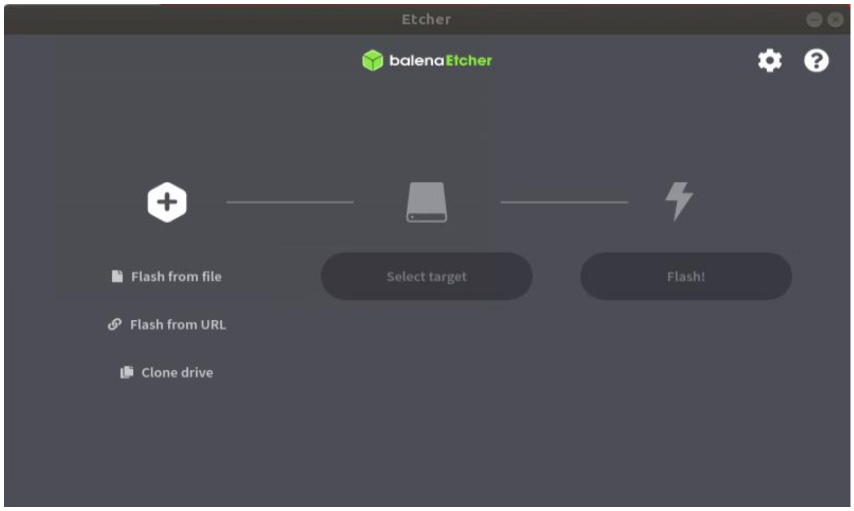
```
test@test:~$ sha256sum -c *.sha
orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img: OK
```

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-tablet-24.03-linux-6.1.43-ext4.img.gz.sha
```

```
openwrt-aarch64-opicm5-tablet-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

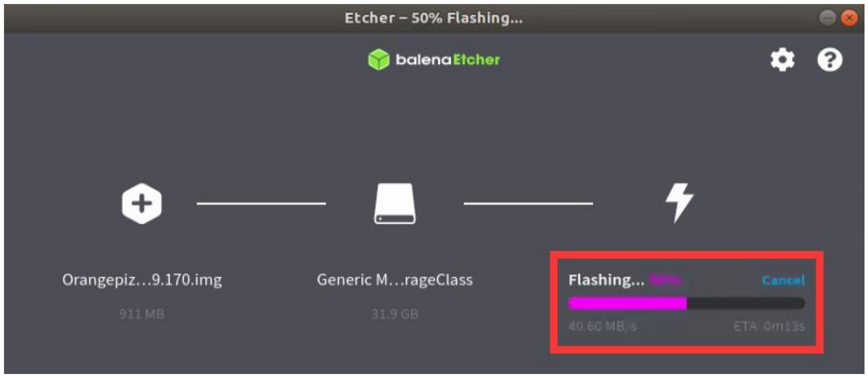


9) The specific steps to use balenaEtcher to burn the Linux image are as follows

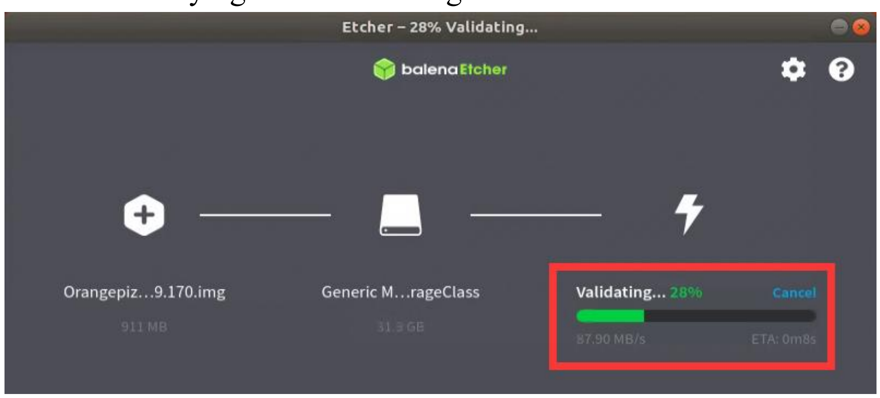
a. First select the path of the Linux image file to be burned
b. Then select the drive letter of the TF Card
c. Finally, click Flash to start burning the Linux image to the TF Card

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

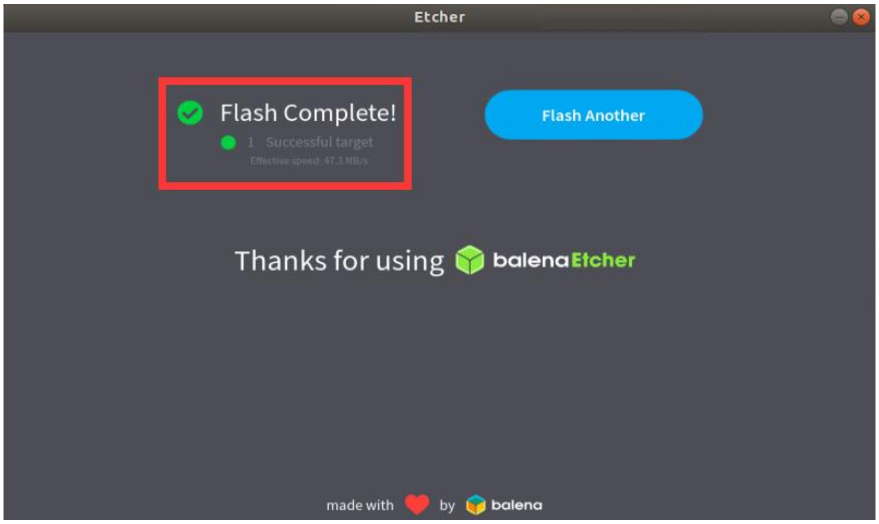
purple to indicate that the Linux image is being burned to the TF card.



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.



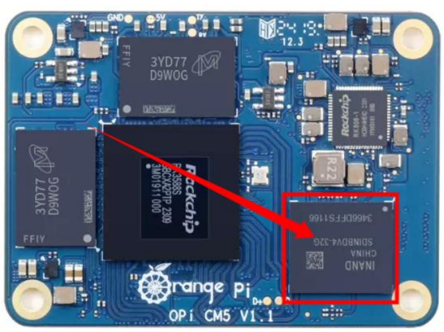
2. 5. The method of burning Linux images into eMMC

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:

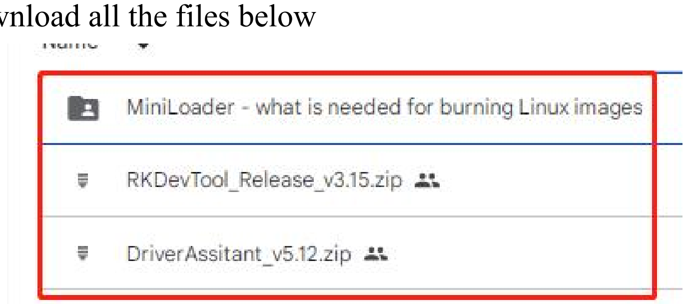


2) Firstly, it is necessary to prepare a high-quality Type-C interface data cable.

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



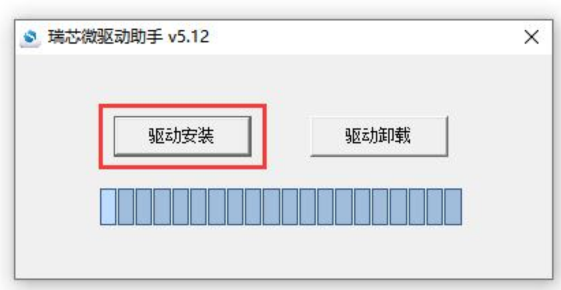
Note that the "MiniLoader - what is needed for burning Linux images" folder is hereinafter referred to as the MiniLoader folder.

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

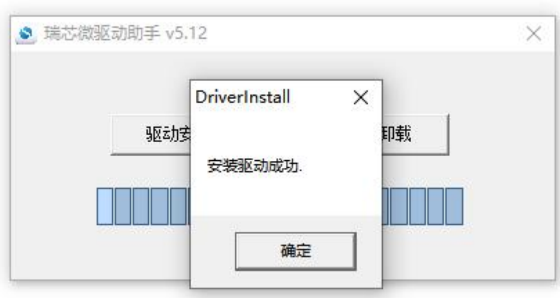
5) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it

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

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

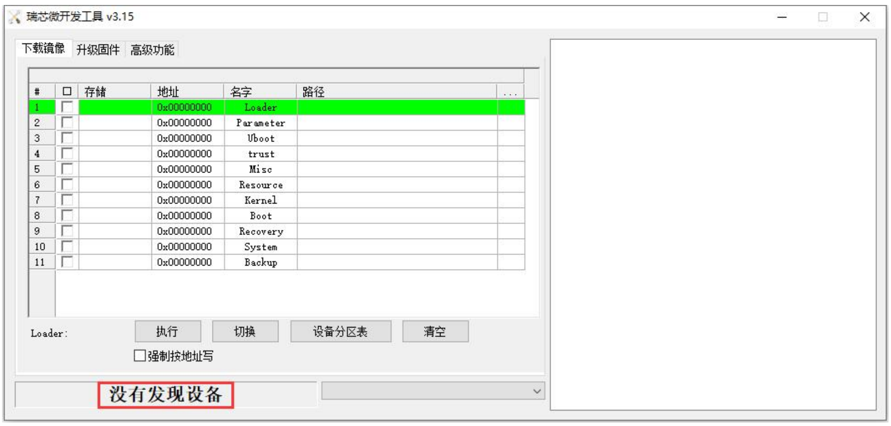


b. After waiting for a while, a pop-up window will prompt "**driver installed successfully**", and then click the "**OK**" button.



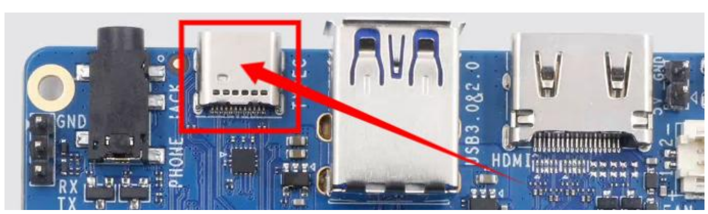
7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it

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 the Linux image into eMMC.

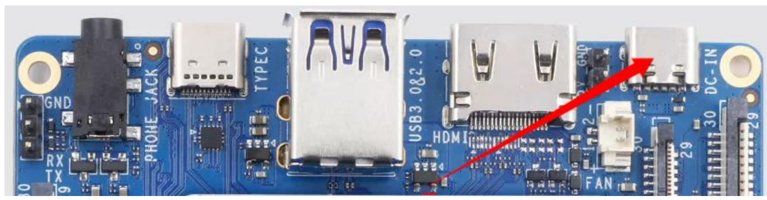
a. Firstly, connect the development board to the Windows computer via a Type-C data cable. The location of the Type-C interface on the development board is shown in the following figure:



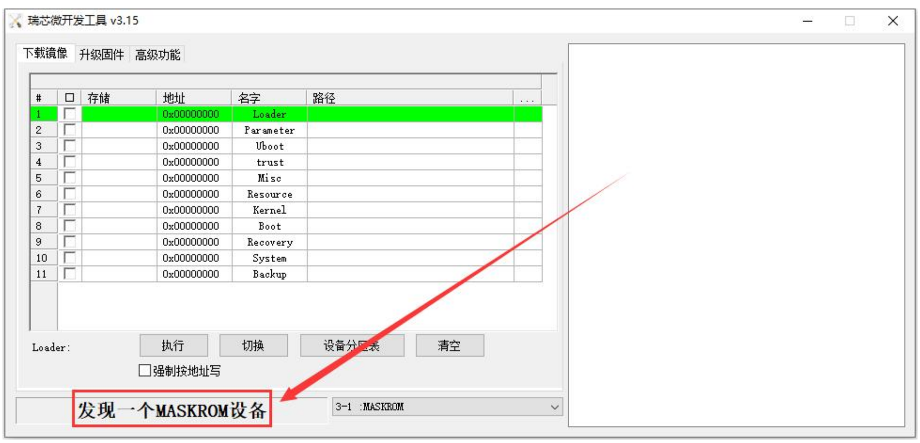
b. Ensure that the development board is not inserted with a TF card or connected to a power source.

c. Then hold down the MaskROM button on the development board and do not release it. The position of the MaskROM button on the development board is shown in the following figure:

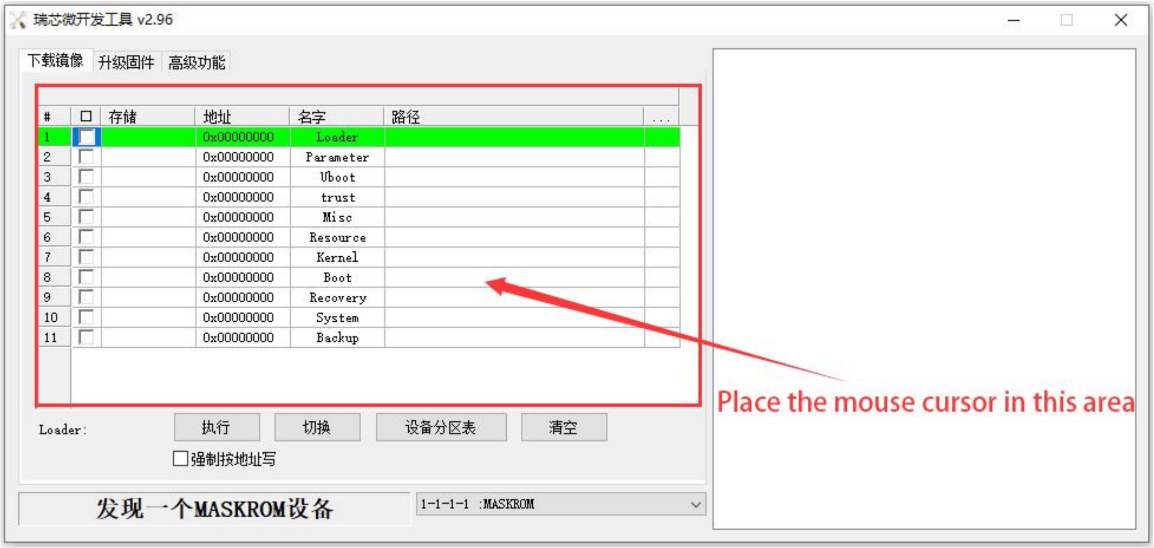
d. Then connect the Type-C interface power to the development board, power it on, and then release the MaskROM button.



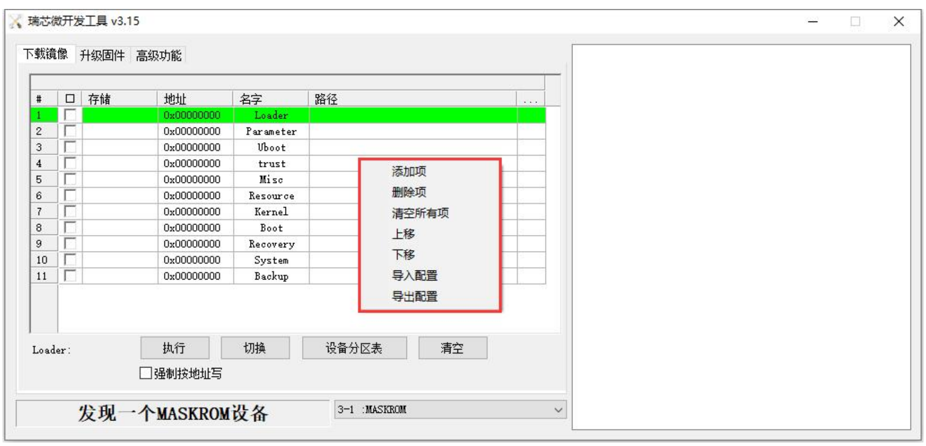
e. If the previous steps are successful, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**Found a MASKROM device**".



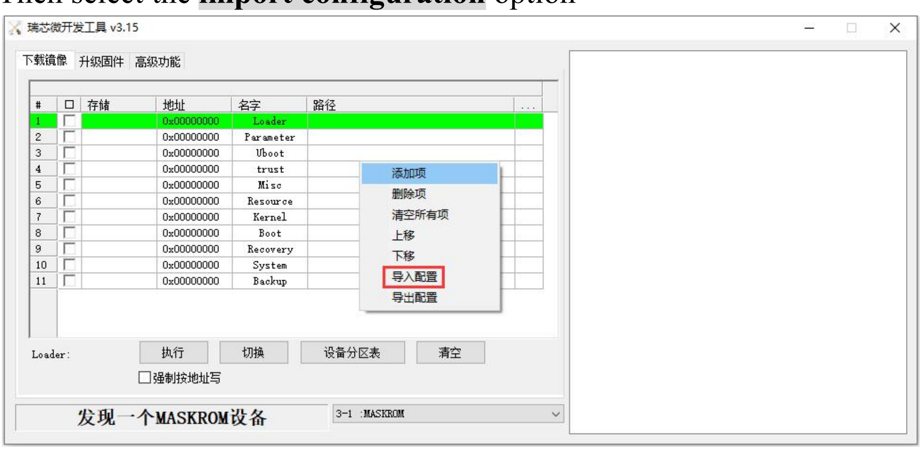
f. Then place the mouse cursor in the area below



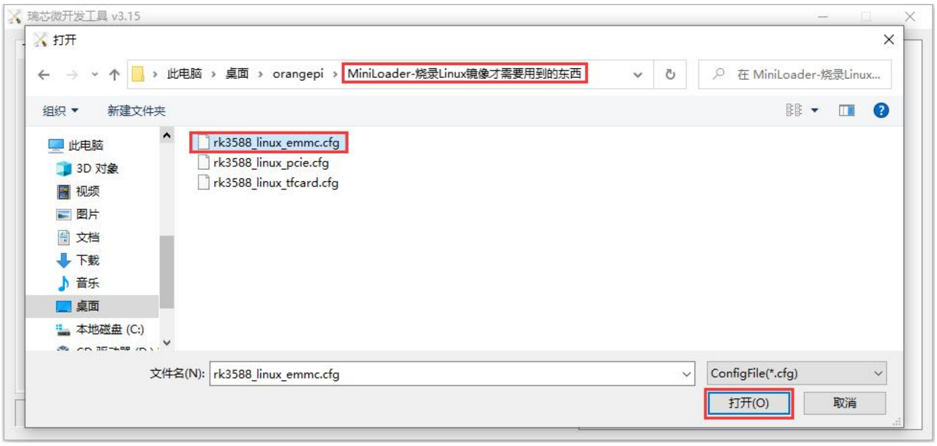
g. Then, clicking the right mouse button will pop up the selection interface shown in the following figure



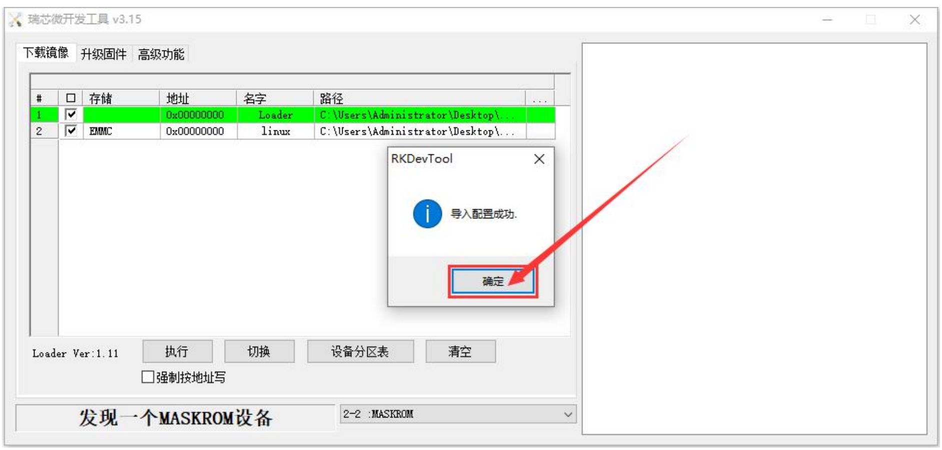
h. Then select the **import configuration** option



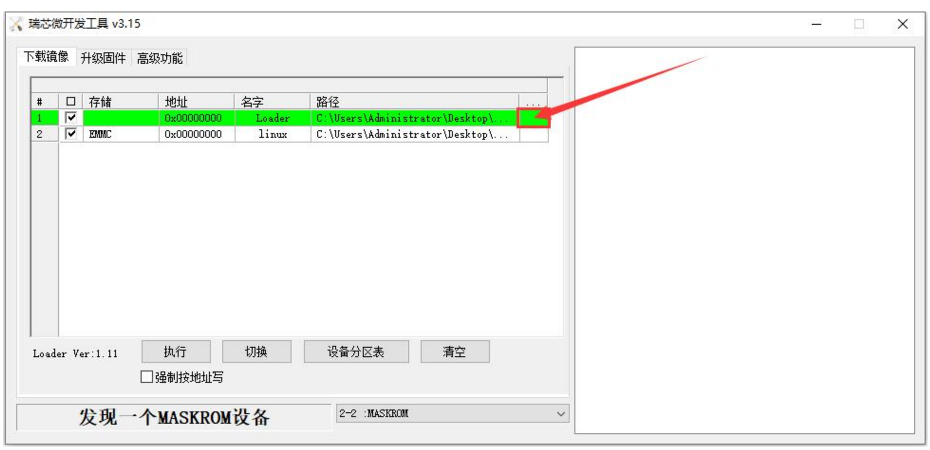
i. Then select the **rk3588_linux_emmc.cfg** configuration file from the **MiniLoader** folder downloaded earlier, and click to **open**



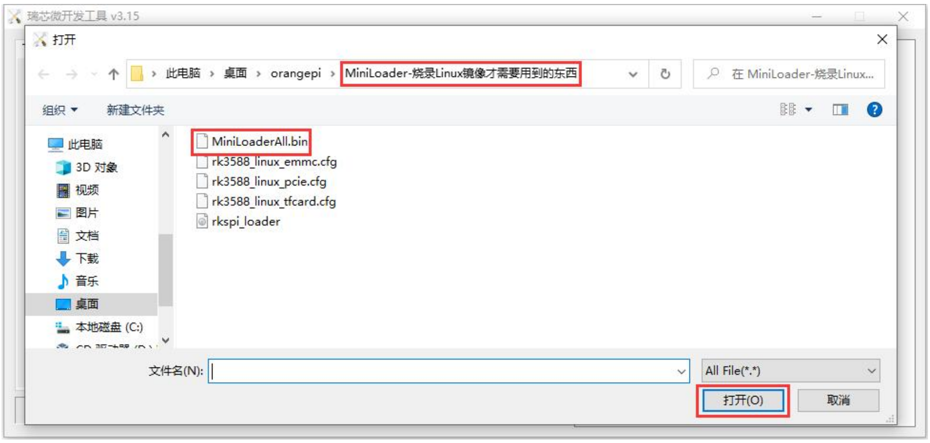
j. Then click **OK**



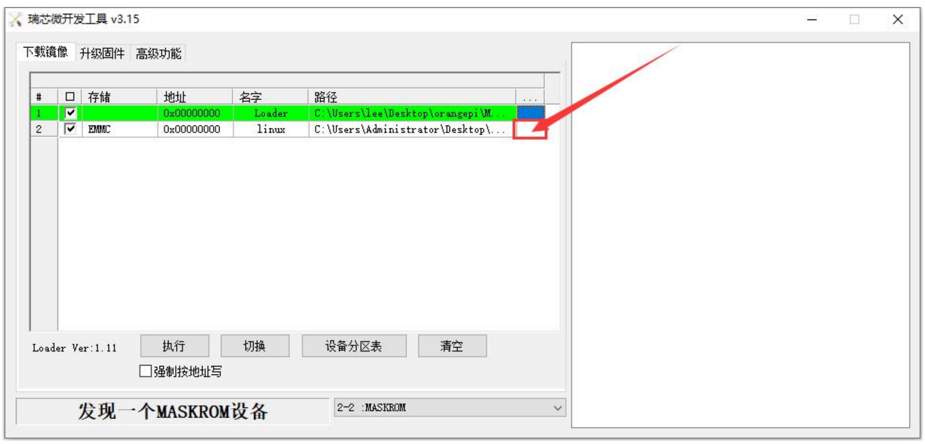
k. Then click on the location shown in the following image



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

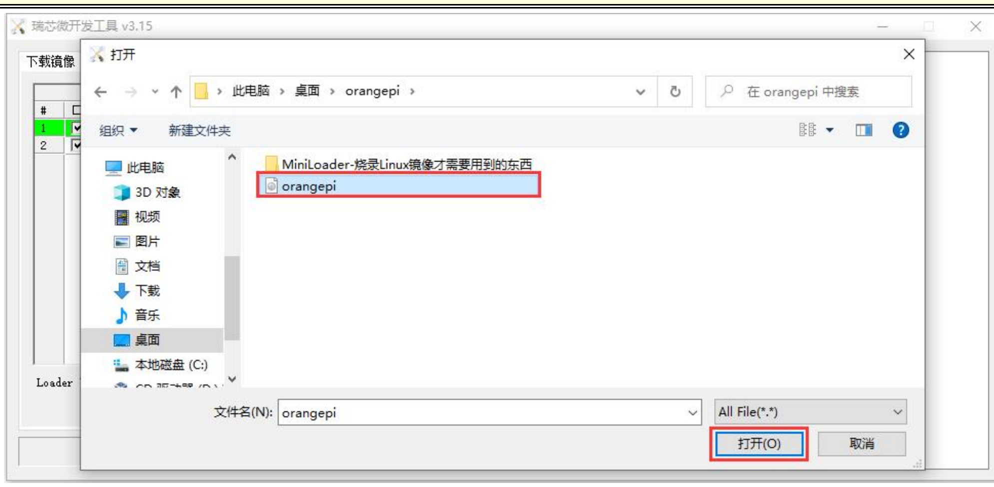


m. Then click on the location shown in the following image

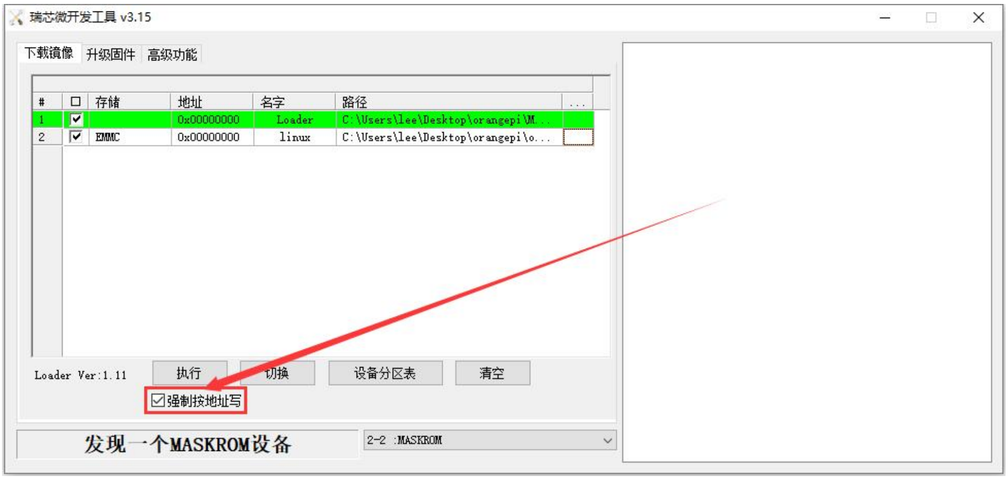


n. Then select the path to the Linux image you want to burn, and click **open**

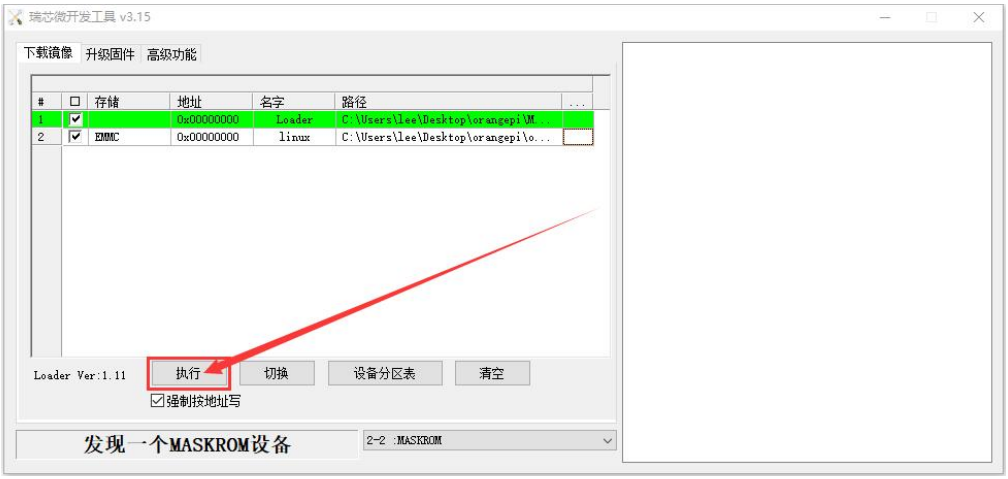
Before burning the image, it is recommended to rename the Linux image to orangepi.img or other shorter names, so that the percentage of burning progress can be seen when burning the image.



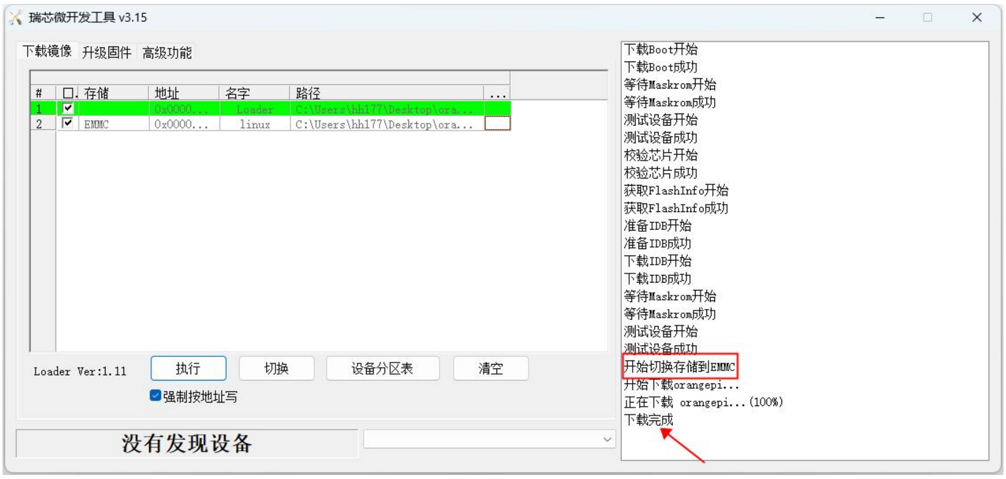
o. Then please check the option to **force writing by address**



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

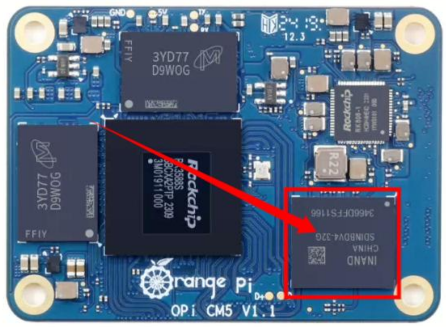


r. After burning the Linux image into eMMC, the Linux system will automatically start.

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-tablet_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, let's 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-tablet_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 image to TF card

2. 6. 1. Method of burning using RKDevTool

1) Firstly, it is necessary to prepare a high-quality Type-C interface data cable.

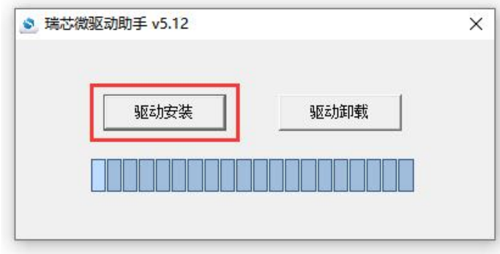
2) Then download the Ruixin micro driver **DriverAssitant_v5.12.zip** and the burning tool **RKDevTool_Release_v3.15.zip** from **Orange Pi's data download page**.

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

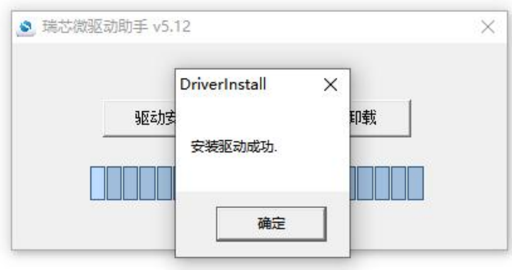
4) Then use the decompression software to unzip **DriverAssitant_v5.12.zip**, and then find the **DriverInstall.exe** executable file in the unzipped folder and open it.

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.

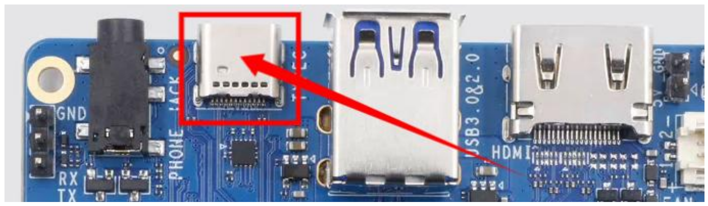


6) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. You can find **RKDevTool** in the unzipped folder and open it.

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 onto the TF card.

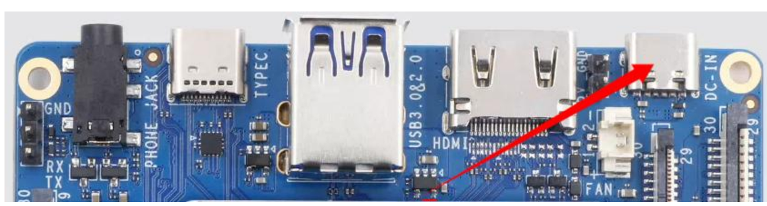
a. Firstly, connect the development board to the Windows computer via a Type-C data cable. The location of the Type-C interface on the development board is shown in the following figure.



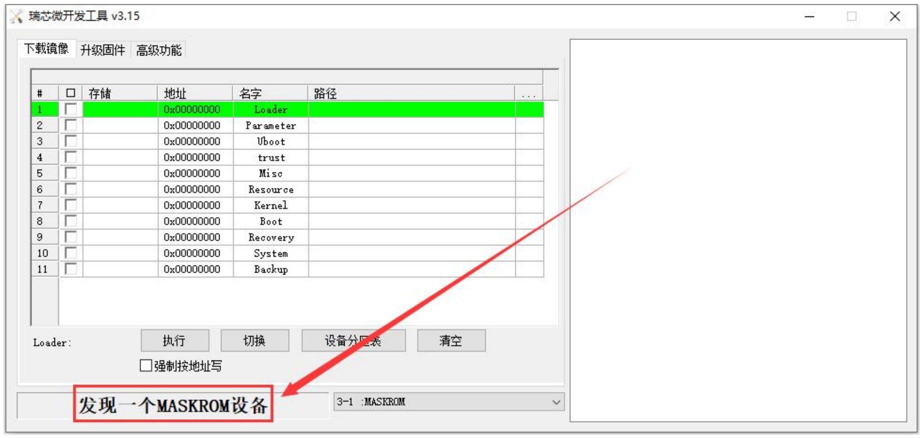
b. Then insert the TF card into the development board and ensure that the board is not connected to a power source.

c. Then hold down the MaskROM button on the development board and do not release it. The position of the MaskROM button on the development board is shown in the following figure:

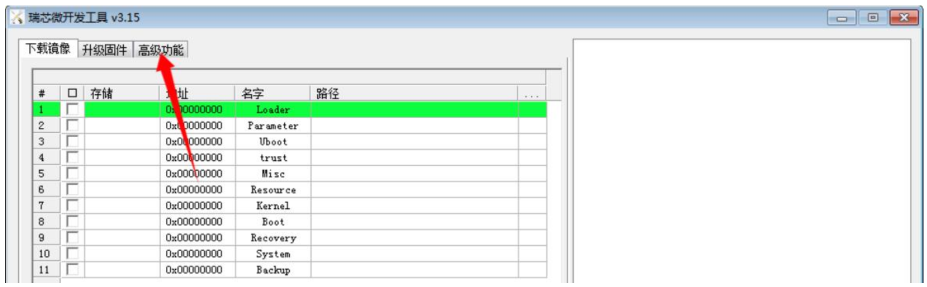
d. Then connect the Type-C interface power to the development board, power it on, and then release the MaskROM button.



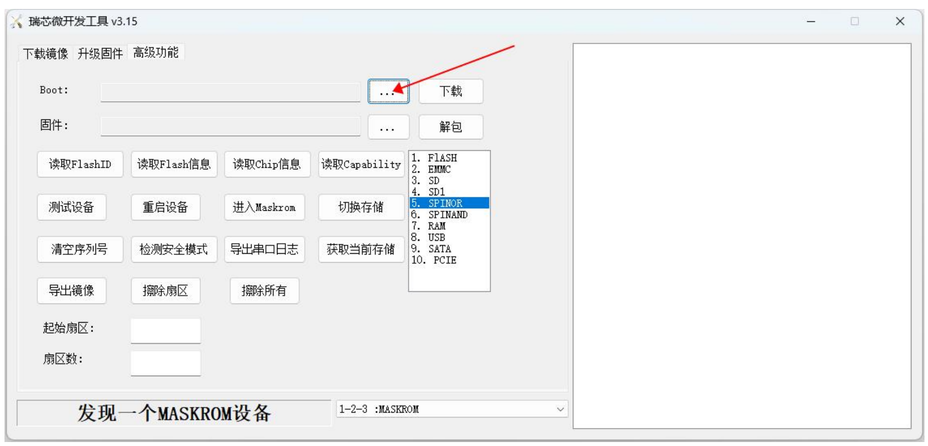
e. If the previous steps go well, the development board will enter **MASKROM** mode, and the interface of the burning tool will prompt "**A MASKROM device was found**"



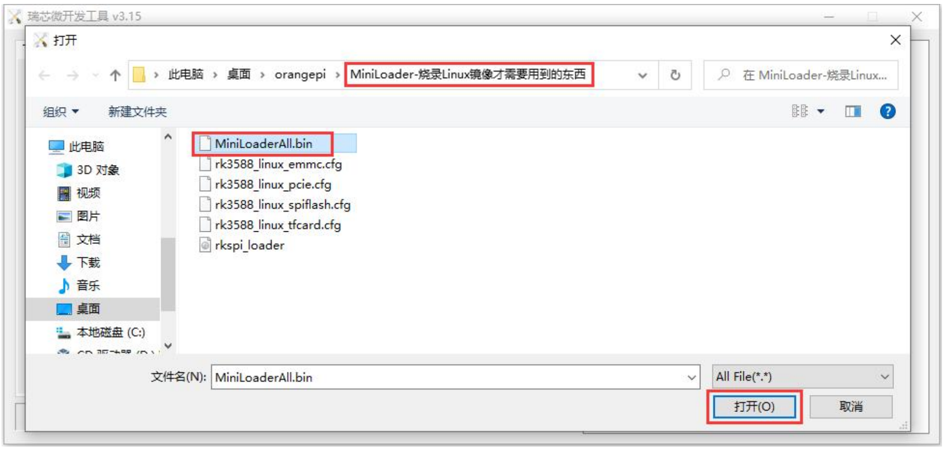
f. Then please select **Advanced Features**



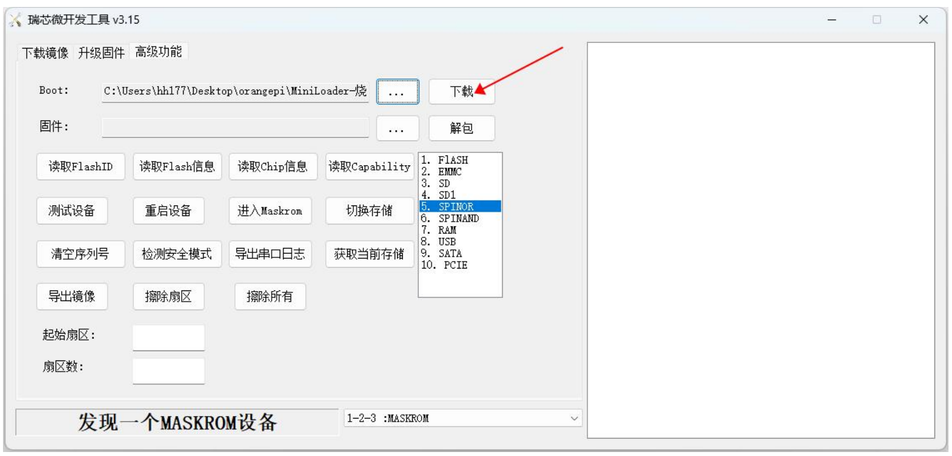
g. Then click the location shown in the picture below



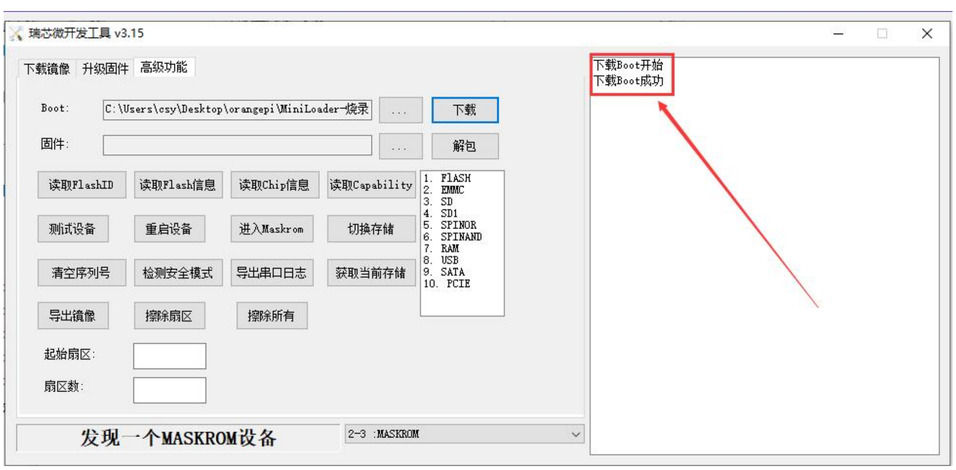
h. Then select **MiniLoaderAll.bin** in the **MiniLoader**r folder downloaded earlier, and then click Open



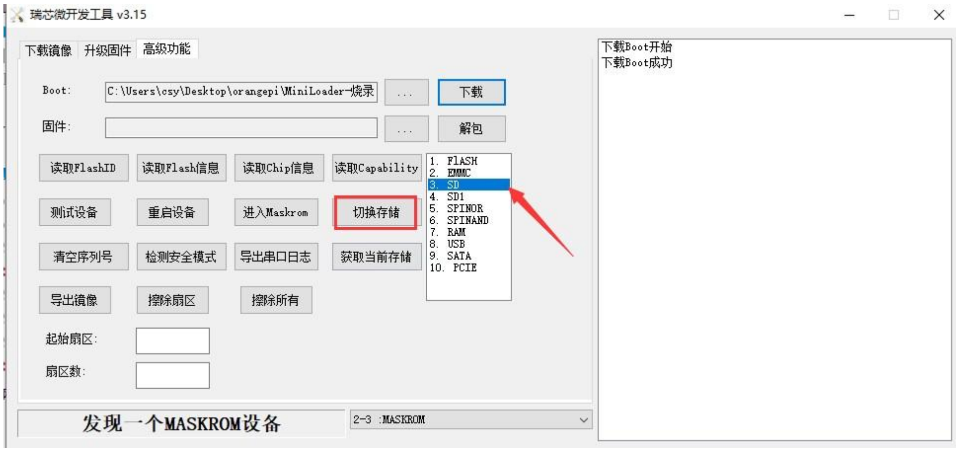
i. Then click **Download**



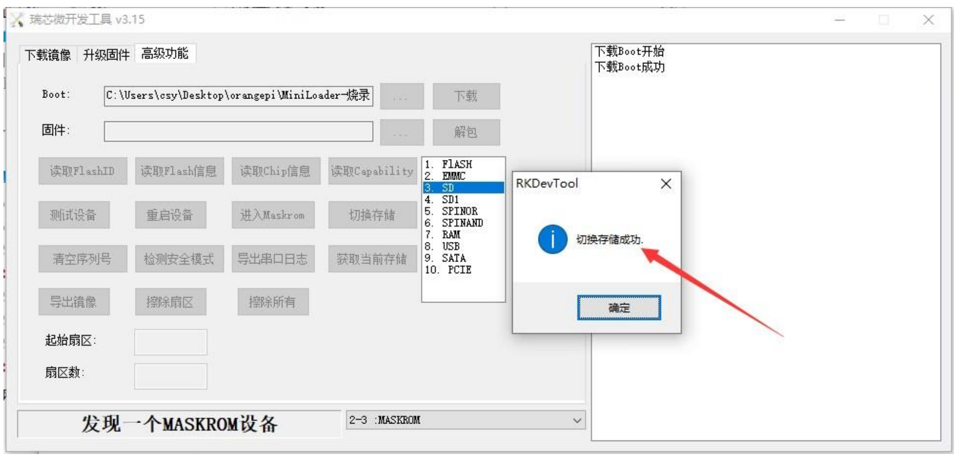
j. After downloading **MiniLoaderAll.bin**, the display is as shown below



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.

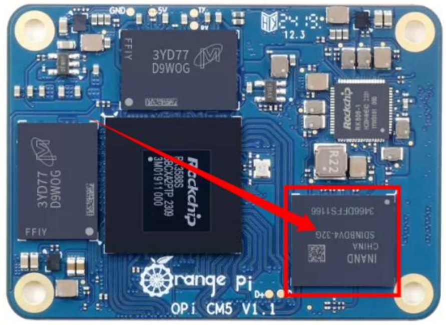
o. Finally, click the "**Upgrade**" button to start burning. The log during the burning process is as shown below. After the burning is completed, the Android system will automatically start.

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:

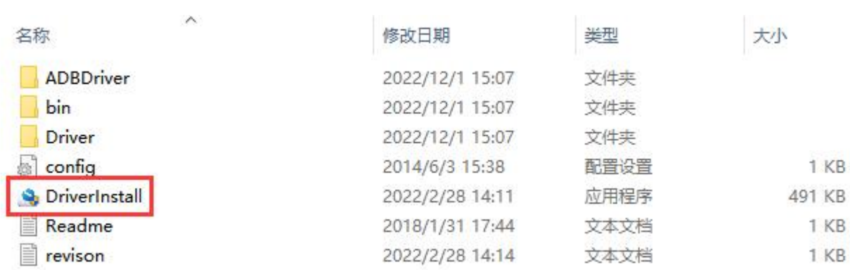


2) We also need to prepare a high-quality Type-C interface data cable.

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

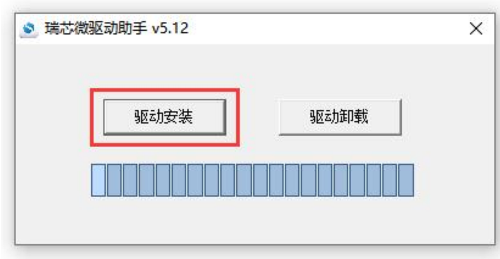
4) Then download the Android image from **Orange Pi's download page**.

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.

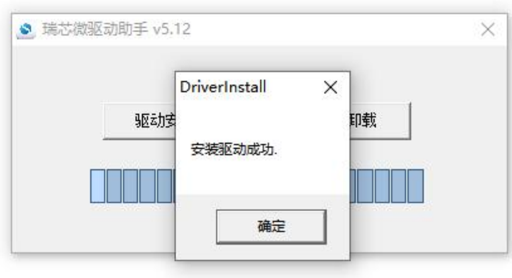


6) After opening **DriverInstall.exe** the steps to install the Rockchip driver are as follows:

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.

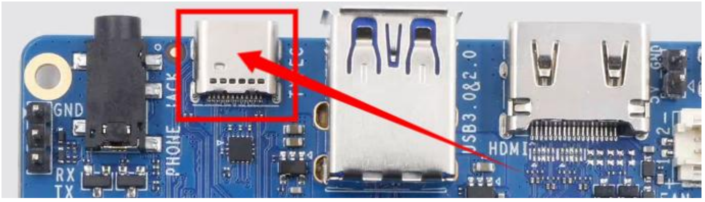


7) Then unzip **RKDevTool_Release_v3.15.zip**. This software does not need to be installed. Just find **RKDevTool** in the unzipped folder and open it.

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 into eMMC.

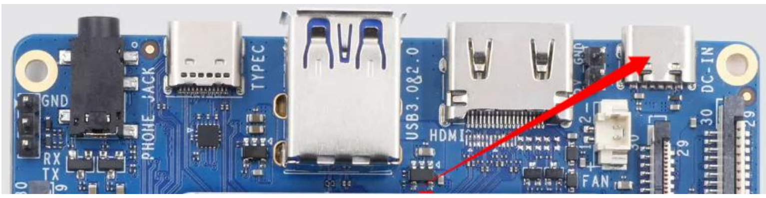
a. Firstly, connect the development board to the Windows computer via a Type-C data cable. The location of the Type-C interface on the development board is shown in the following figure:



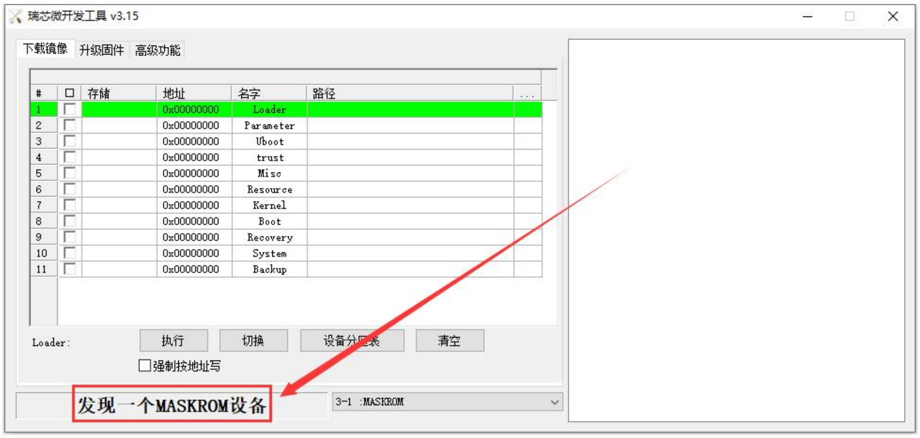
b. Ensure that the development board is not connected to a power source or inserted with a TF card.

c. Then hold down the MaskROM button on the development board and do not release it. 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 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 debugging serial port on the motherboard is shown in the following figure:

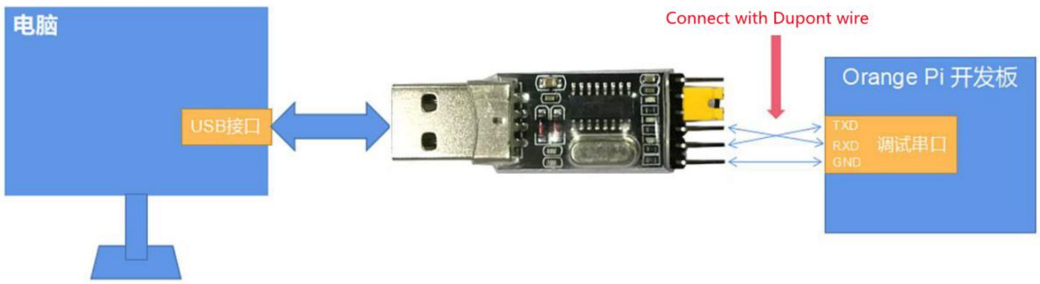
3) The GND, TX and RX pins of the USB to TTL module need to be connected to the debug serial port of the development board via a Dupont cable.

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
```

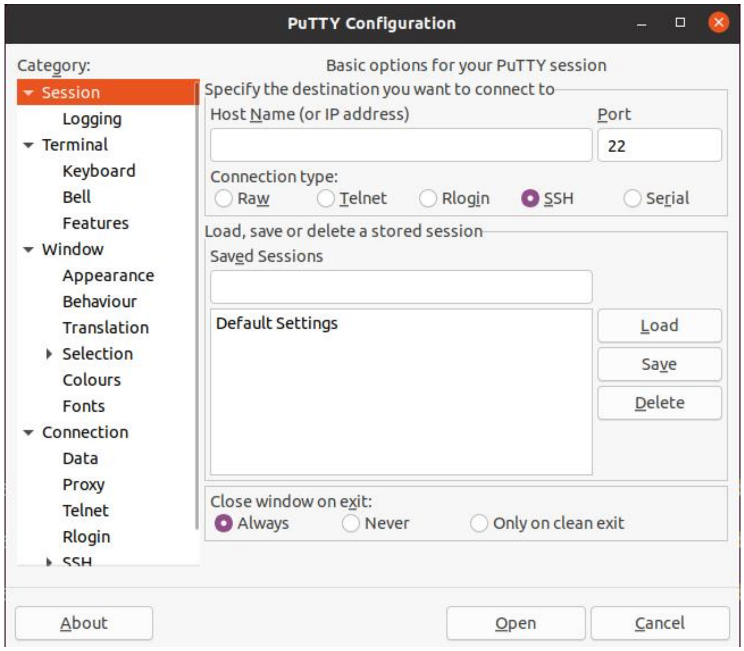
2) Then install putty on your Ubuntu PC using the command below.

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

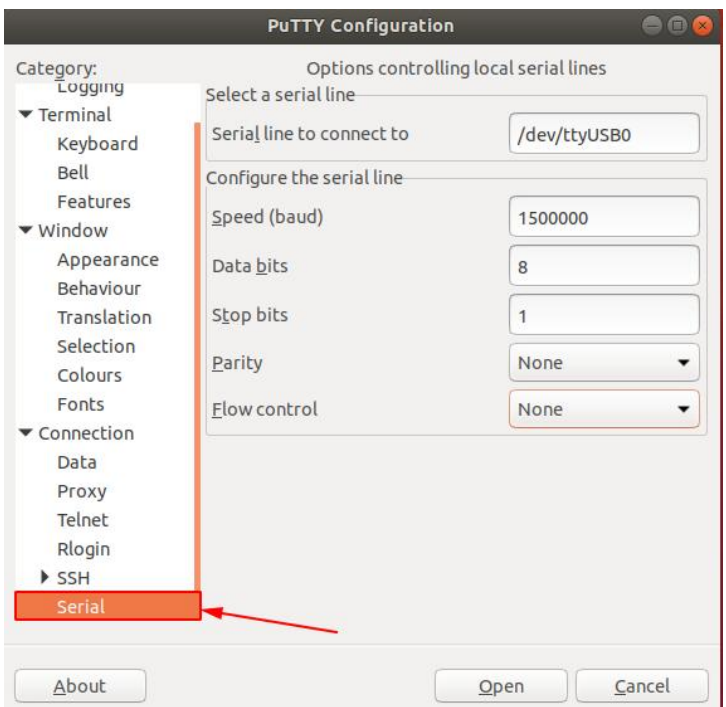
3) Then run Putty, **remember to add sudo privileges**.

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

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



5) First select the serial port settings interface.

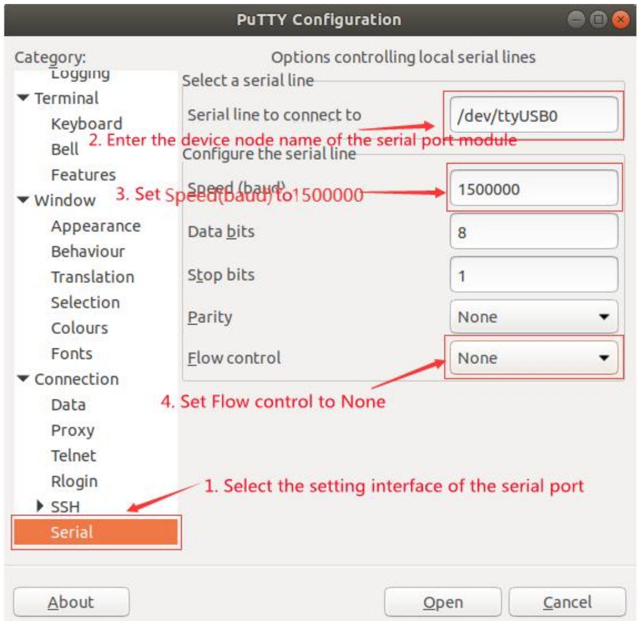


6) Then set the serial port parameters.

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).

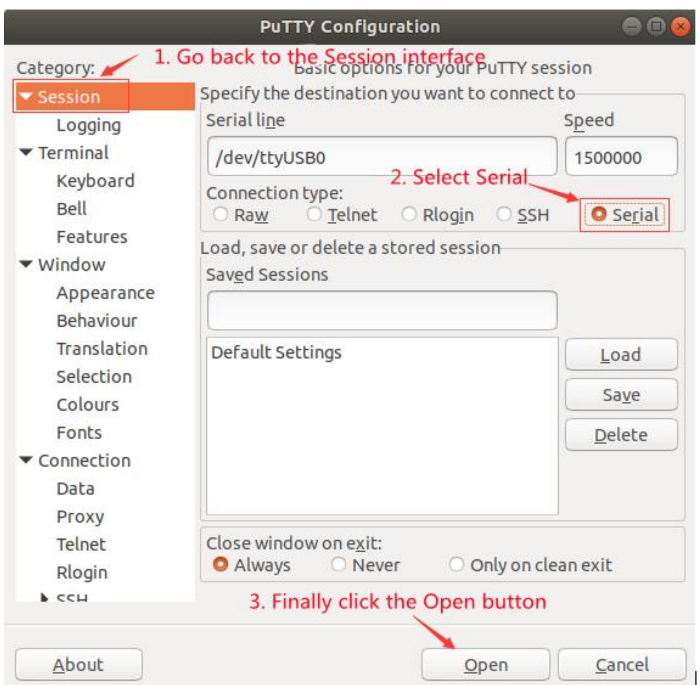
c. Set Flow control to None.



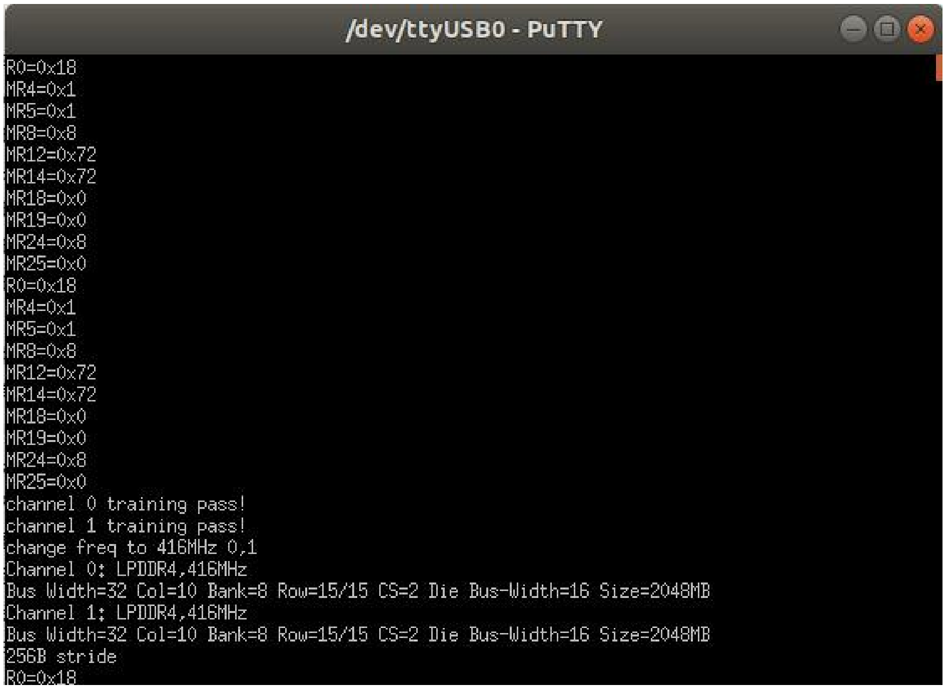
7) After completing the settings on the serial port settings interface, return to the Session interface.

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

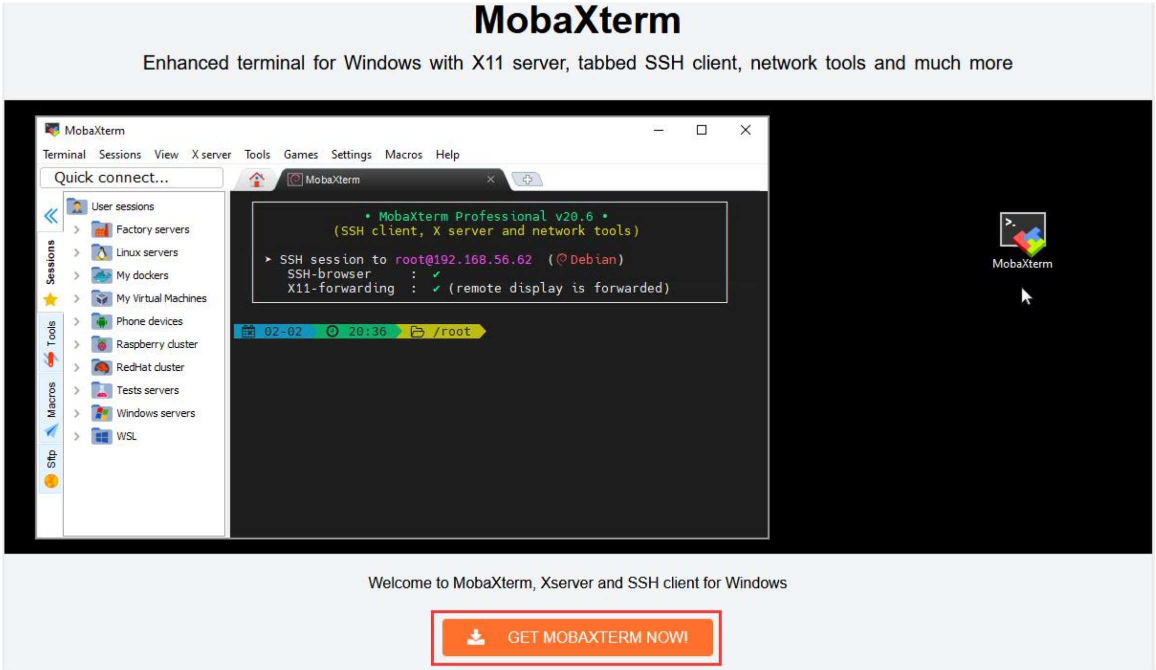
There are many serial port debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without purchasing a serial number.

1) Download MobaXterm.

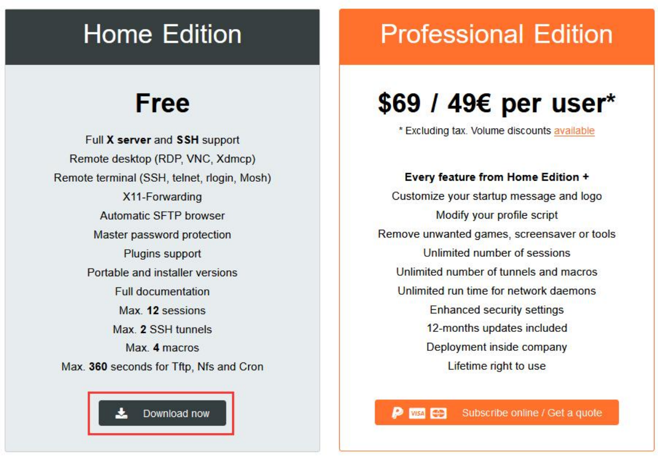
a. Download MobaXterm from the following URL:

https://mobaxterm.mobatek.net

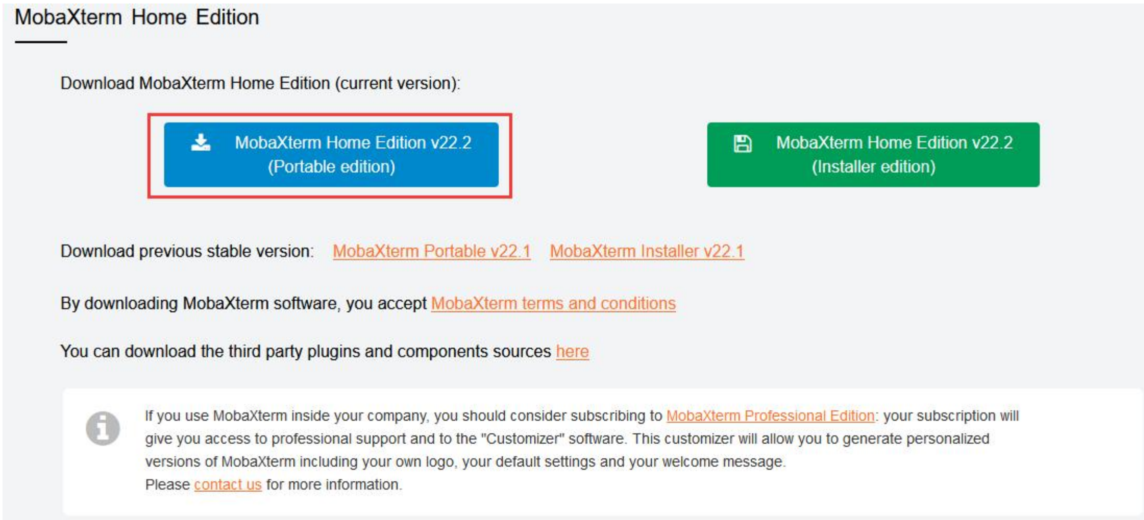
b. Go to the MobaXterm download page and click **GET XOBATERM NOW!**.



c. Then choose to download the Home version.



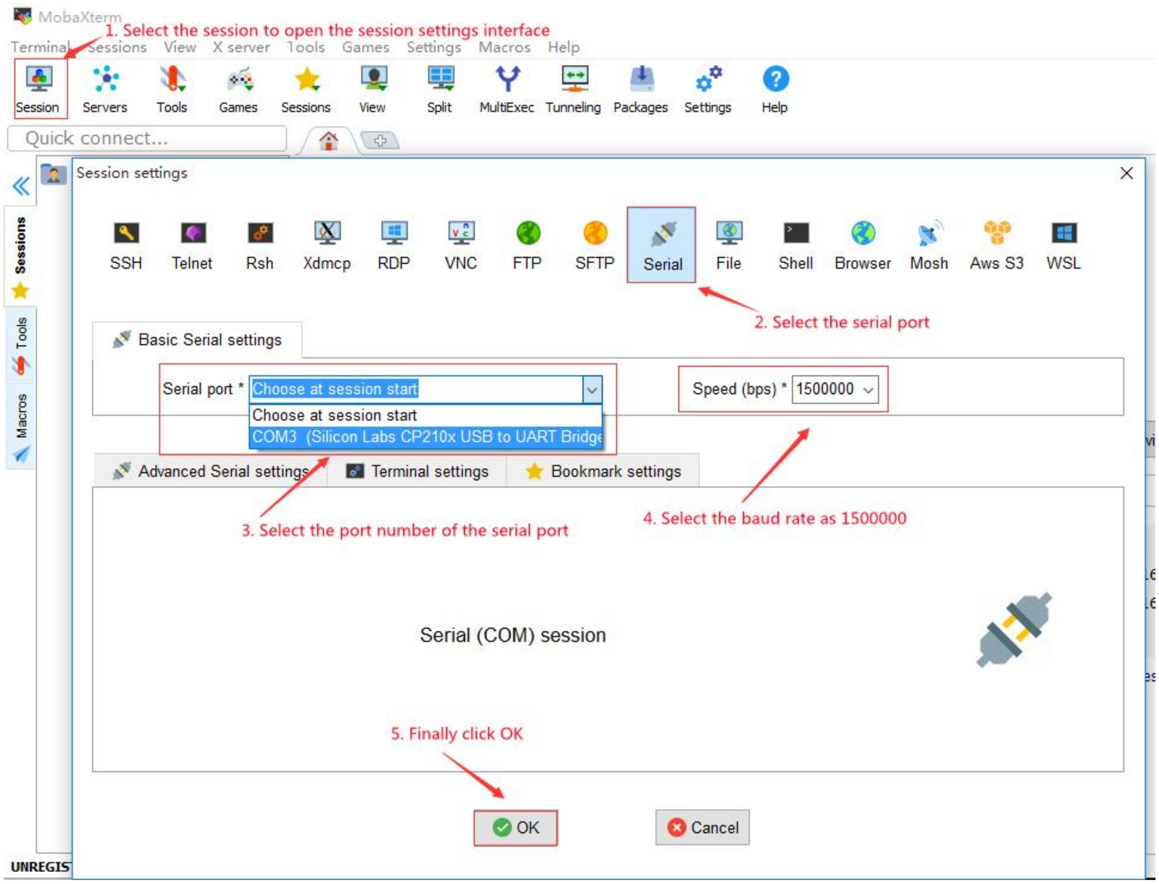
d. Then select the Portable version. After downloading, you don't need to install it, you can just open it and use it.



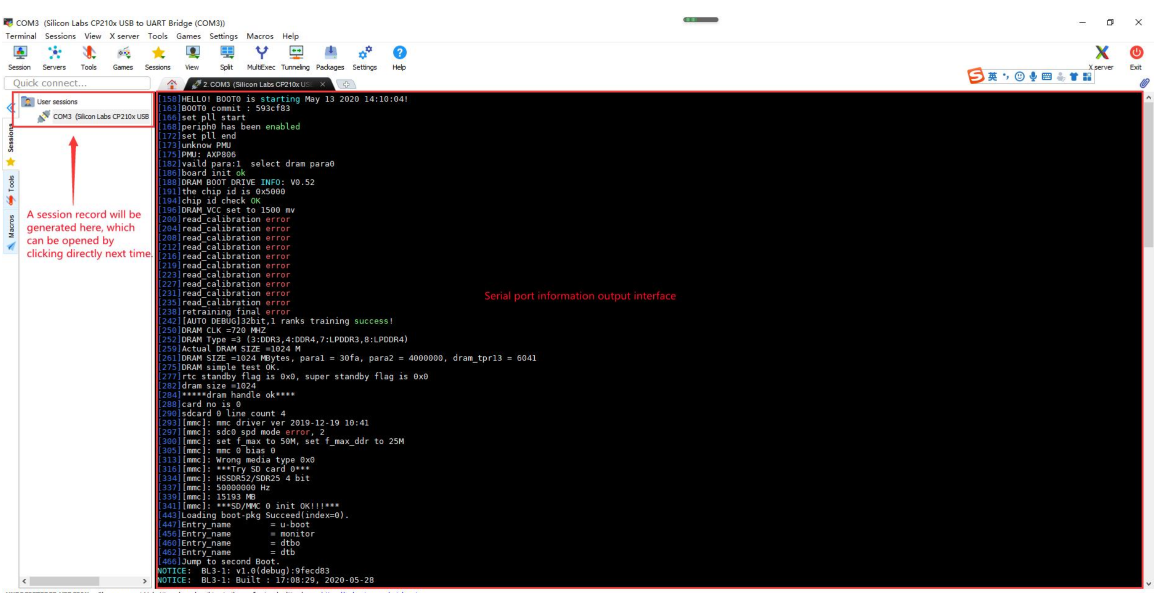
2) After downloading, use decompression software to decompress the downloaded compressed package to get the executable software of MobaXterm, and then double-click to open it.

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

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



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. Ubuntu/Debian Server and Xfce Desktop System User Manual

The content of this chapter is based on the Linux server version image and the XFCE desktop version image.

If you are using the Ubuntu 22.04 Gnome image, please first refer to the instructions in the Ubuntu 22.04 Gnome Wayland desktop system user manual chapter,

The content that does not exist in the Ubuntu 22.04 Gnome Wayland desktop system usage instructions chapter can be referred to in this chapter's instructions, but some details may vary, please pay special attention to this.

If you are using an OPi OS Arch image, please refer to the Orange Pi OS Arch system user manual chapter.

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
WIFI	OK	OK	OK	OK

Bluetooth	**OK**	**OK**	**OK**	**OK**
Debug UART	**OK**	**OK**	**OK**	**OK**
FAN	**OK**	**OK**	**OK**	**OK**
EMMC startup	**OK**	**OK**	**OK**	**OK**
GPIO（26pin）	**OK**	**OK**	**OK**	**OK**
UART（26pin）	**OK**	**OK**	**OK**	**OK**
SPI（26pin）	**OK**	**OK**	**OK**	**OK**
I2C（26pin）	**OK**	**OK**	**OK**	**OK**
PWM（26pin）	**OK**	**OK**	**OK**	**OK**
Camera1	**OK**	**OK**	**OK**	**OK**
Camera2	**OK**	**OK**	**OK**	**OK**
Camera3	**OK**	**OK**	**OK**	**OK**
LCD display	**OK**	**OK**	**OK**	**OK**
LCD Touch	**OK**	**OK**	**OK**	**OK**
Onboard MIC	**OK**	**OK**	**OK**	**OK**
Headphone playback	**OK**	**OK**	**OK**	**OK**
Earphone recording	**OK**	**OK**	**OK**	**OK**
Speaker x 2	**OK**	**OK**	**OK**	**OK**
LED lights	**OK**	**OK**	**OK**	**OK**
Type-C to USB 3.0	**OK**	**OK**	**OK**	**OK**
Type-C interface DP display	**OK**	**OK**	**OK**	**OK**
Type-C interface DP audio	**OK**	**OK**	**OK**	**OK**
TF card startup	**OK**	**OK**	**OK**	**OK**
NVMe SSD recognition	**OK**	**OK**	**OK**	**OK**
SATA SSD recognition	**OK**	**OK**	**OK**	**OK**
infrared reception	**OK**	**OK**	**OK**	**OK**
GPU	**OK**	**OK**	**OK**	**OK**
NPU	**OK**	**OK**	**OK**	**OK**
VPU	**OK**	**OK**	**OK**	**OK**
Power on/off button	**OK**	**OK**	**OK**	**OK**
Watch dog test	**OK**	**OK**	**OK**	**OK**
Chromium Hardcore Video	**OK**	**OK**	**OK**	**OK**

3. 3. Linux 6.1 System Adaptation Status

Function	Debian12	Ubuntu22.04
HDMI Display	OK	OK
HDMI Audio	OK	OK
USB 2.0	OK	OK
USB 3.0	OK	OK
WIFI	OK	OK
Bluetooth	OK	OK
Debug UART	OK	OK
FAN	OK	OK
EMMC startup	OK	OK
GPIO（26pin）	OK	OK
UART（26pin）	OK	OK
SPI（26pin）	OK	OK
I2C（26pin）	OK	OK
PWM（26pin）	OK	OK
Camera1	OK	OK
Camera2	OK	OK
Camera3	OK	OK
LCD Display	OK	OK
LCD Touch	OK	OK
Onboard MIC	OK	OK
Headphone playback	OK	OK
Earphone recording	OK	OK
Speaker x 2	OK	OK
LED lights	OK	OK
Type-C to USB 3.0	OK	OK
Type-C interface DP display	OK	OK
Type-C interface DP audio	OK	OK
TF card startup	OK	OK
NVMe SSD recognition	OK	OK
SATA SSD recognition	OK	OK
Battery	OK	OK

Infrared	**OK**	**OK**
GPU	**OK**	**OK**
NPU	**OK**	**OK**
VPU	**OK**	**OK**
Power on/off button	**OK**	**OK**
Watch dog test	**OK**	**OK**
Chromium Hardcore 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

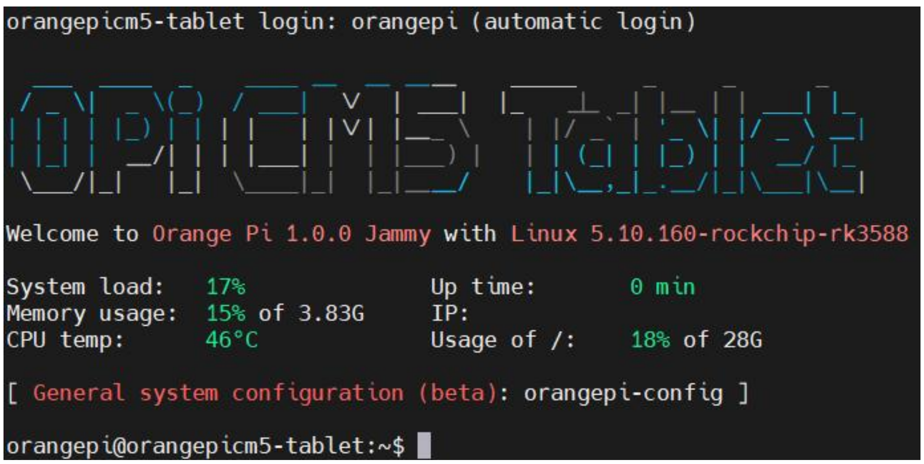
Note that when entering the password, the specific content of the entered

password will not be displayed on the screen. Please do not assume that there is any malfunction. After entering, simply press Enter.

When the password prompt is incorrect or there is a problem with the SSH connection, please note that as long as you are using the Linux image provided by Orange Pi, do not suspect that the password is incorrect, but instead look for other reasons.

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**.



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.

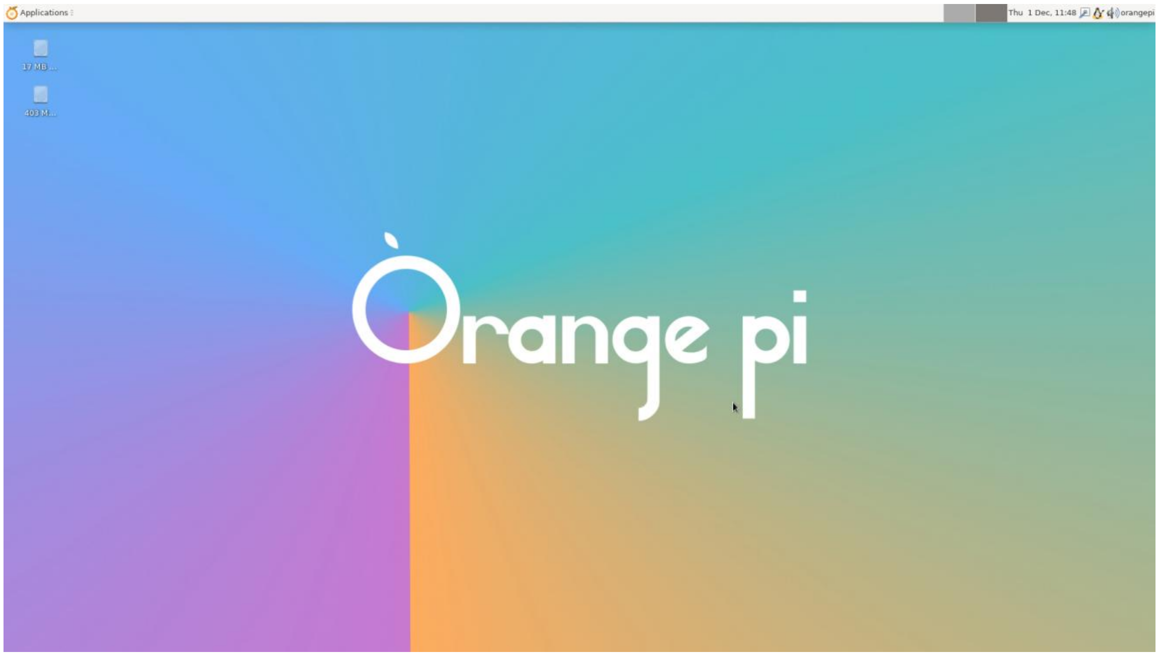
```
orangepi@orangepi:~$ sudo auto_login_cli.sh -d
```

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 starts, it will automatically log in to the desktop without entering a password.



2) Running the following command can prevent the desktop version system from automatically logging into the desktop

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

3) Then restart the system and a login dialog box will appear. At this time, you need to enter your **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. Test instructions for onboard LED lights

1) There is a red light and a green light on the bottom plate, located as shown in the following figure:

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

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

4) The method for setting the green light on/off and flashing is as follows:

Please note that the following operations should be performed under the root user.

a. First, enter the directory for green light settings.

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

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

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

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

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

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

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

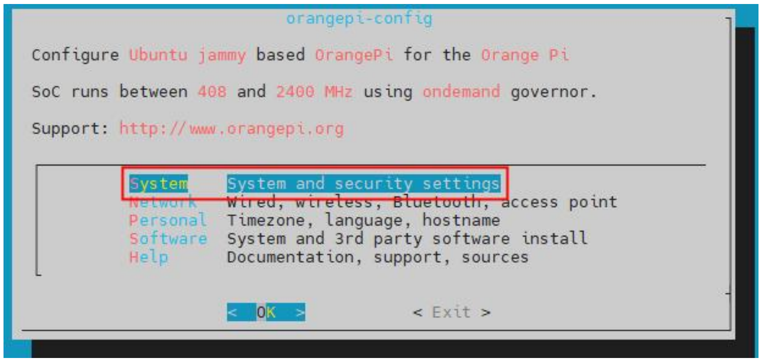
5) If the green LED light does not need to flash after booting up, you can use the

following method to turn off the green light.

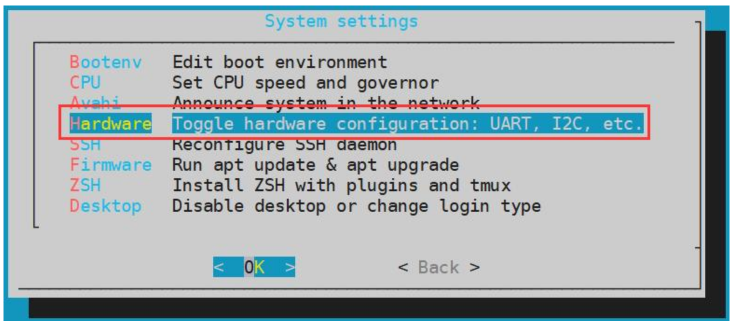
a. First, run **orangepi-config**. Ordinary users should remember to add **sudo** privileges.

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

b. Then select **System**。

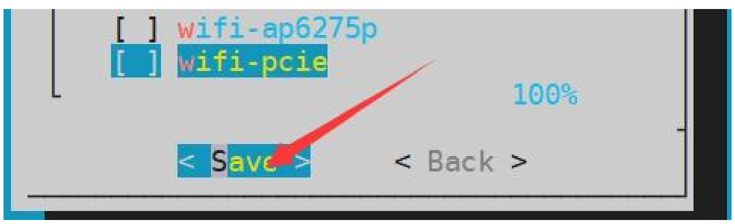


c. Then select **Hardware**。

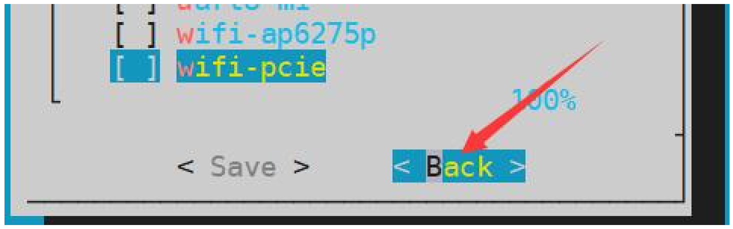


d. Then use the directional keys on the keyboard to locate the position shown in the figure below, and use a **space** to select the **opicm5-tablet-disable-leds** configuration

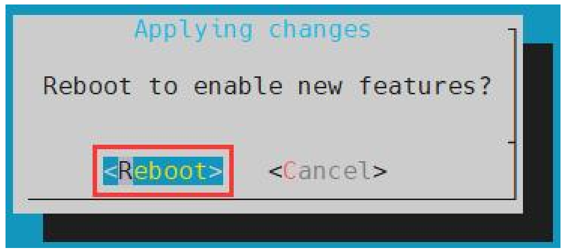
e. Then select **<Save>** to save.



f. Then select **<Back>**.



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



h. After restarting, you can see that only the red light is constantly on on the development board, and the green light will not flash anymore.

3. 7. Network Connection Testing

3. 7. 1. WIFI Connection Test

Please do not connect to WIFI by modifying the /etc/network/interfaces configuration file, as there may be issues with using WIFI networks in this way.

3. 7. 1. 1. Server version image connects to WIFI through commands

When the development board is not connected to Ethernet or an HDMI monitor, and only a serial port is connected, it is recommended to use the commands demonstrated in this section to connect to the WIFI network. Because nmtui can only display characters in some serial port software (such as Minicom) and cannot display graphical interfaces properly. Of course, if the development board is connected to an Ethernet or HDMI display, the commands demonstrated in this section can also be used to connect to a WIFI network.

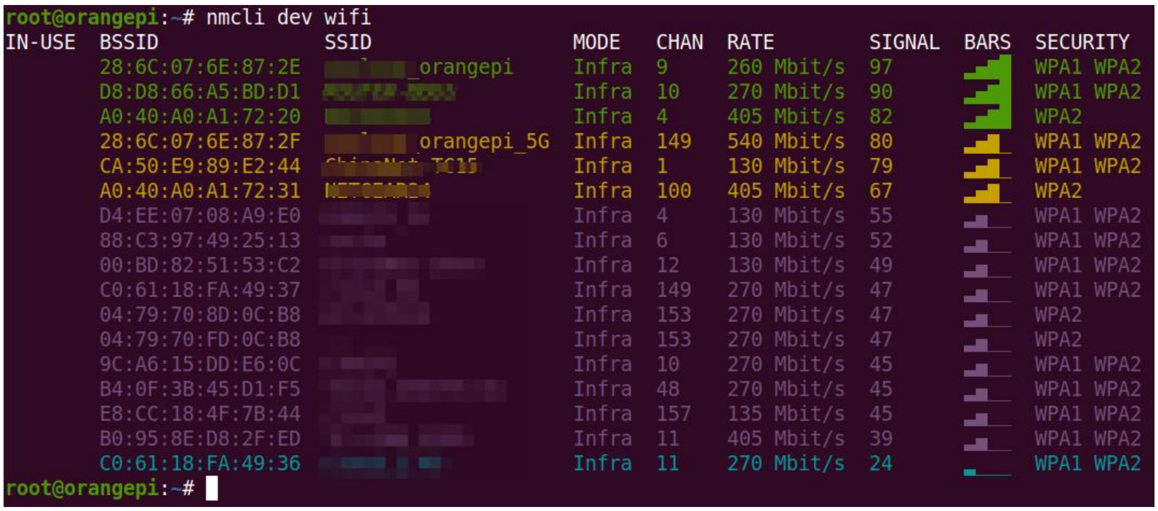
1) There are two ways to log in to the Linux system first:

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

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

2) Firstly, use the **nmcli dev wifi** command to scan the surrounding WIFI hotspots.

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



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

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

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

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

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

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

5) Use the **ping** command to test the connectivity of the WiFi network. The **ping** command can be interrupted by 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
```

3.7.1.2. Server version image connects to WIFI through graphical means

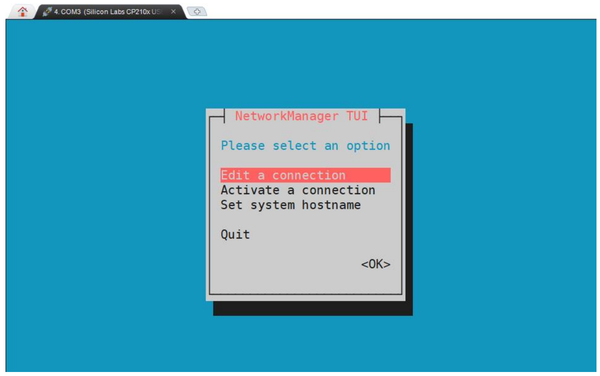
1) There are two ways to log in to the Linux system first:

   a. If the development board is connected to debug the serial port, you can log in to the Linux system using a serial port terminal (please use MobaXterm for serial port software, as the graphical interface cannot be displayed using Minicom).
   b. If the development board is connected to an HDMI monitor, you can log in to the Linux system through the HDMI display terminal.

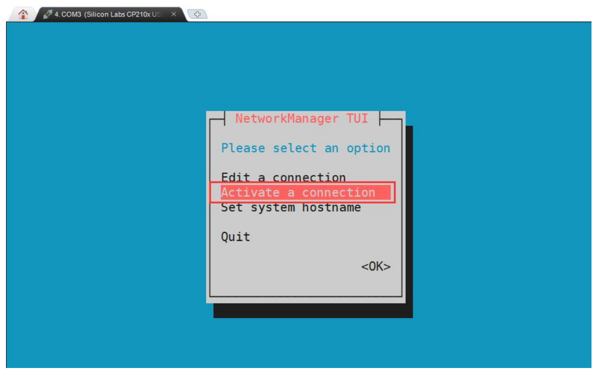
2) Then enter the nmtui command in the command line to open the wifi connection interface.

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

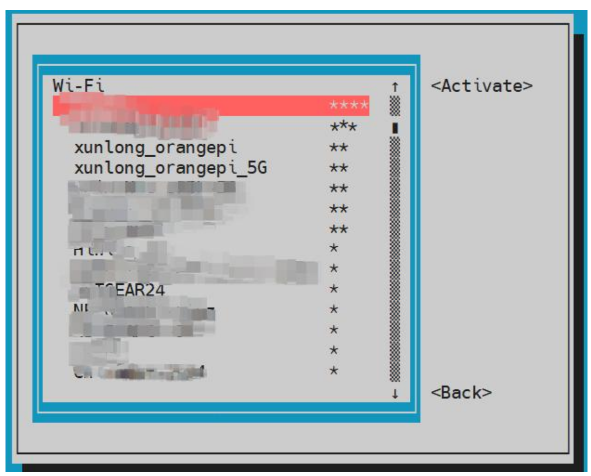
3) The interface opened by entering the nmtui command is shown below：



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

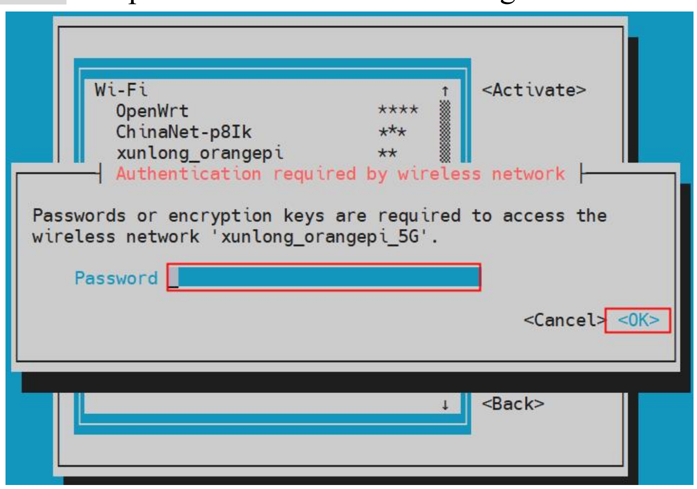


5) Then you can see all the searched WIFI hotspots.

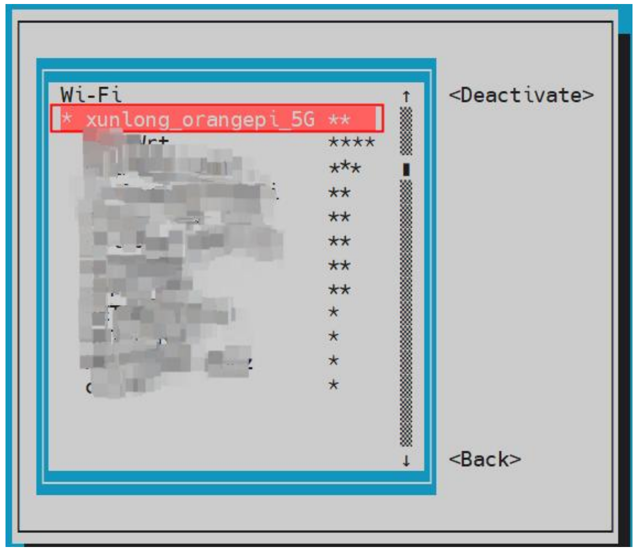


6) Select the WIFI hotspot you want to connect to, then use the Tab key to move the cursor to **Activate** and press Enter.

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
2: 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
```

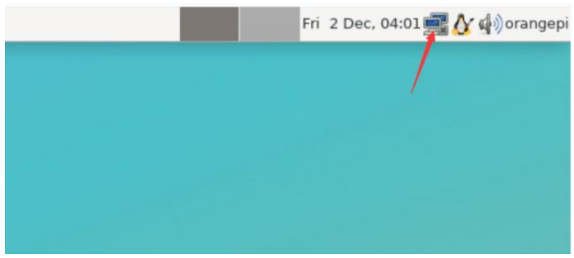
10) Use the **ping** command to test the connectivity of the WiFi network. The **ping** command can be interrupted by pressing the **Ctrl+C** shortcut key.

```
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
```

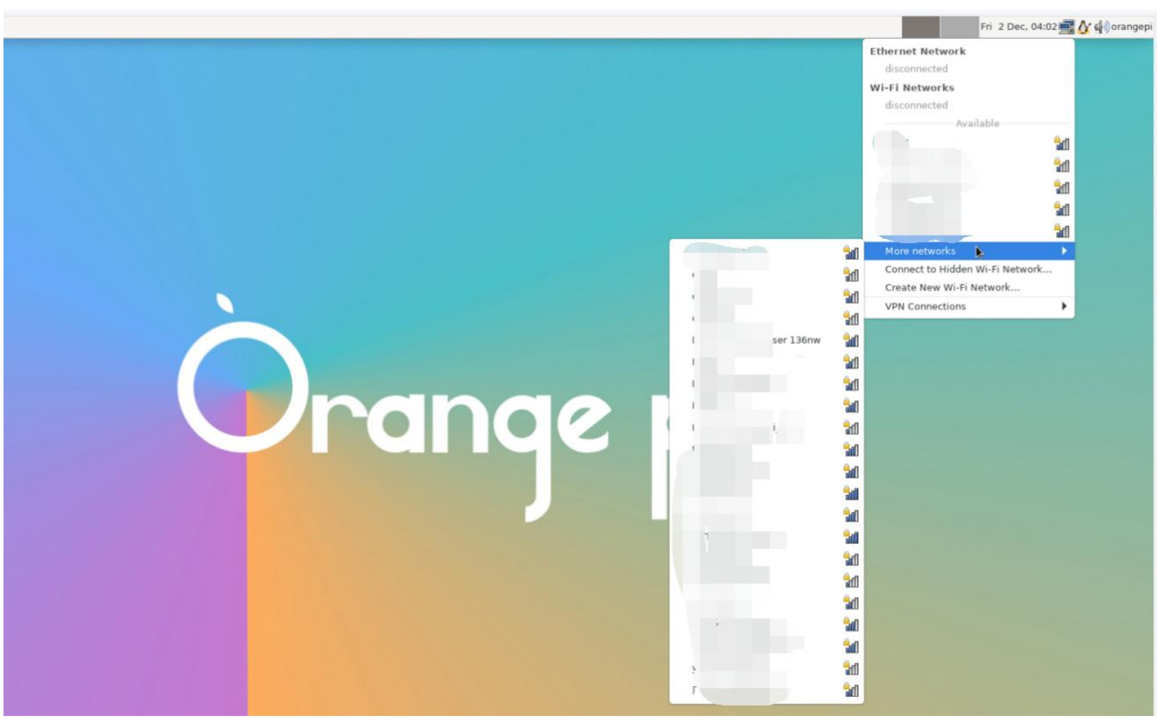
```
64 bytes from 182.92.236.130 (182.92.236.130): icmp_seq=5 ttl=52 time=48.8 ms
^C
--- www.orangepi.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 41.321/44.864/48.834/2.484 ms
```

3.7.1.3. Testing method for desktop version image

1) Click on the network configuration icon in the upper right corner of the desktop.



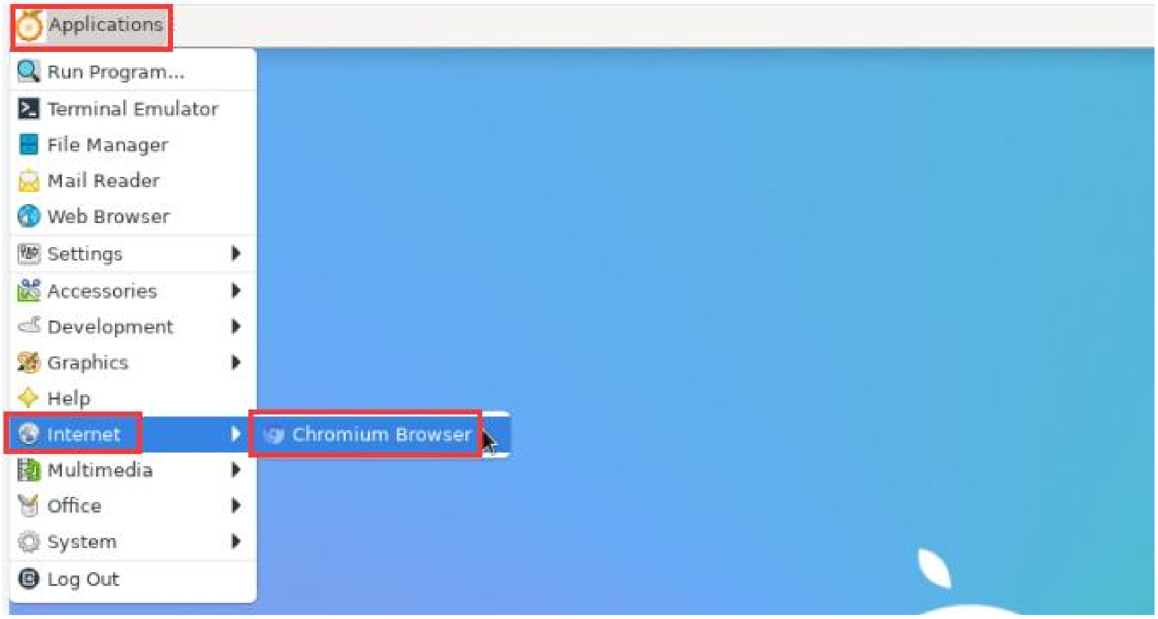
2) Click **More networks** in the pop-up drop-down box to see all scanned WIFI hotspots, and then select the WIFI hotspot you want to connect to.



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. 8. SSH remote login Single-board Computer

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 Single-board computer under Ubuntu

1) Get the IP address of the development board.

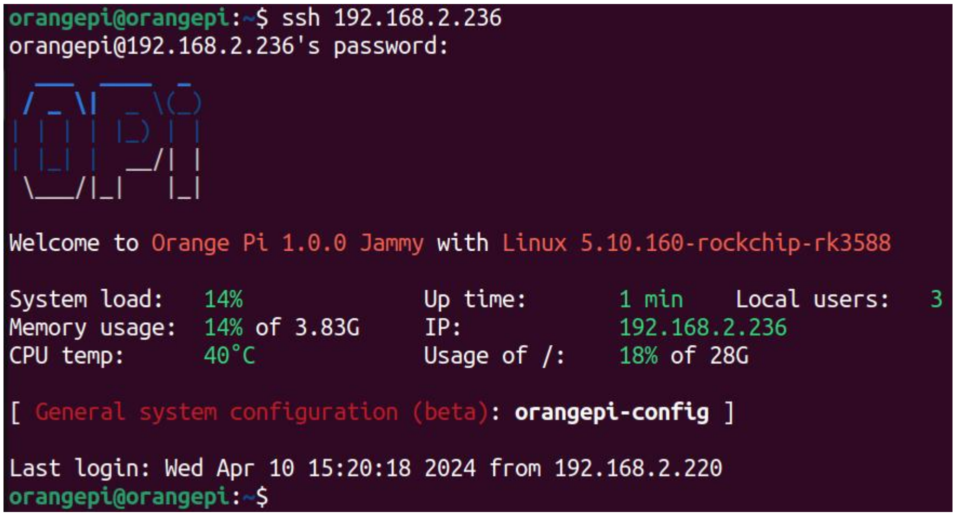
2) Then you can log in to the Linux system remotely through the ssh command.

```
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）
```

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.

If the connection is refused, as long as you are using the image provided by Orange Pi, please do not doubt whether the password orangepi is incorrect, 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**

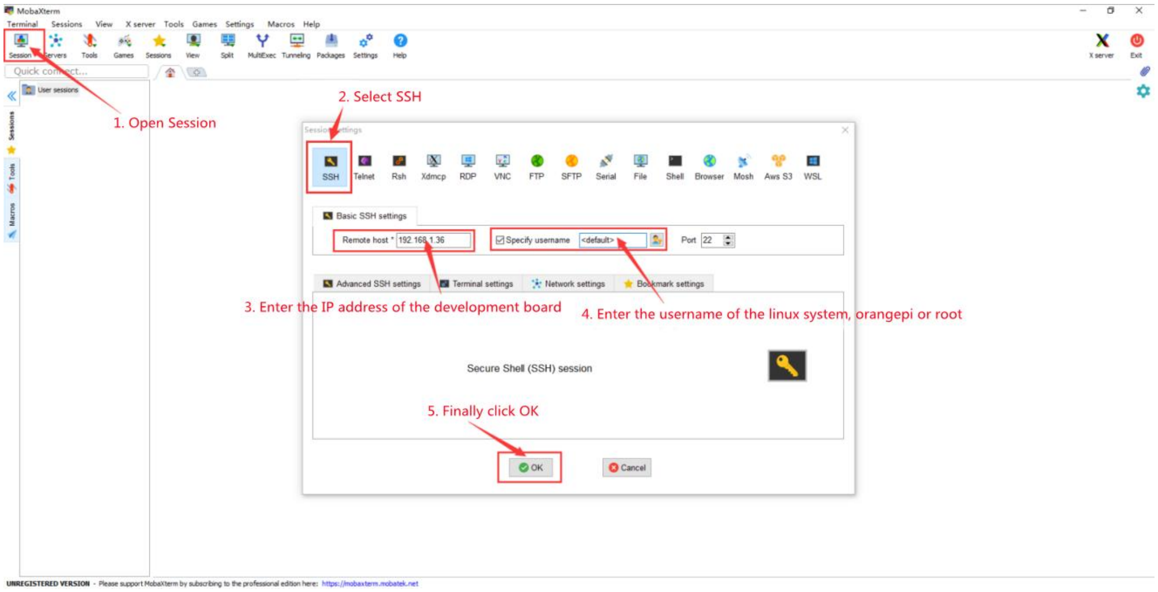
If it still doesn't work, please re-burn the system and try again.

3.8.2. SSH remote login Single-board Computer under Windows

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

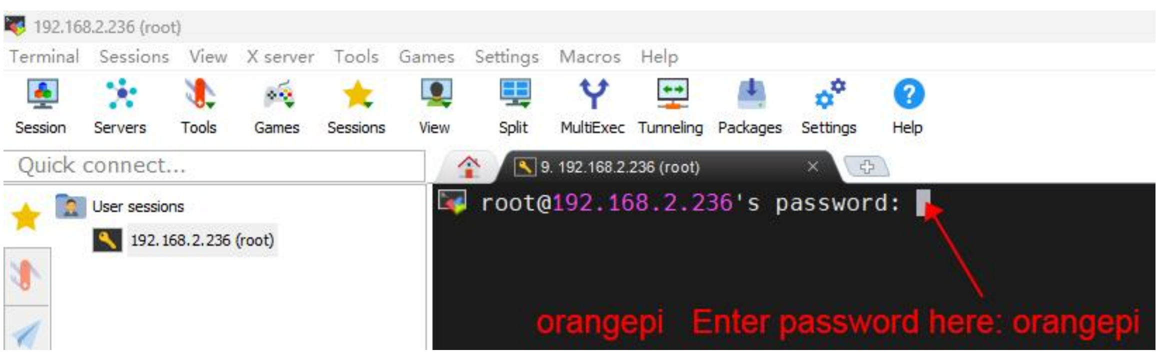
2) Under Windows, you can use MobaXterm to remotely log in to the development board. First, create a new ssh session.

a. Open **Session**.
b. 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. Click **OK**.

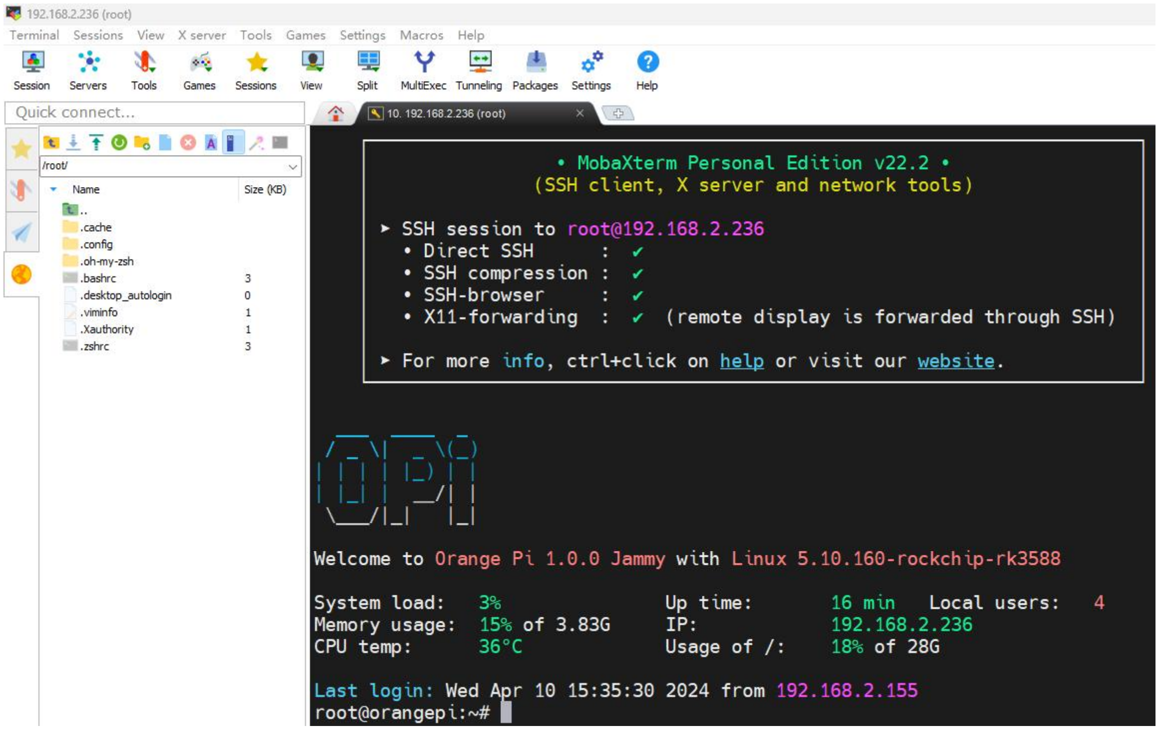


3) You will then be prompted to enter a password. The default password for both root and orangepi users is orangepi.

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@orangepi:/#      <--- 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:

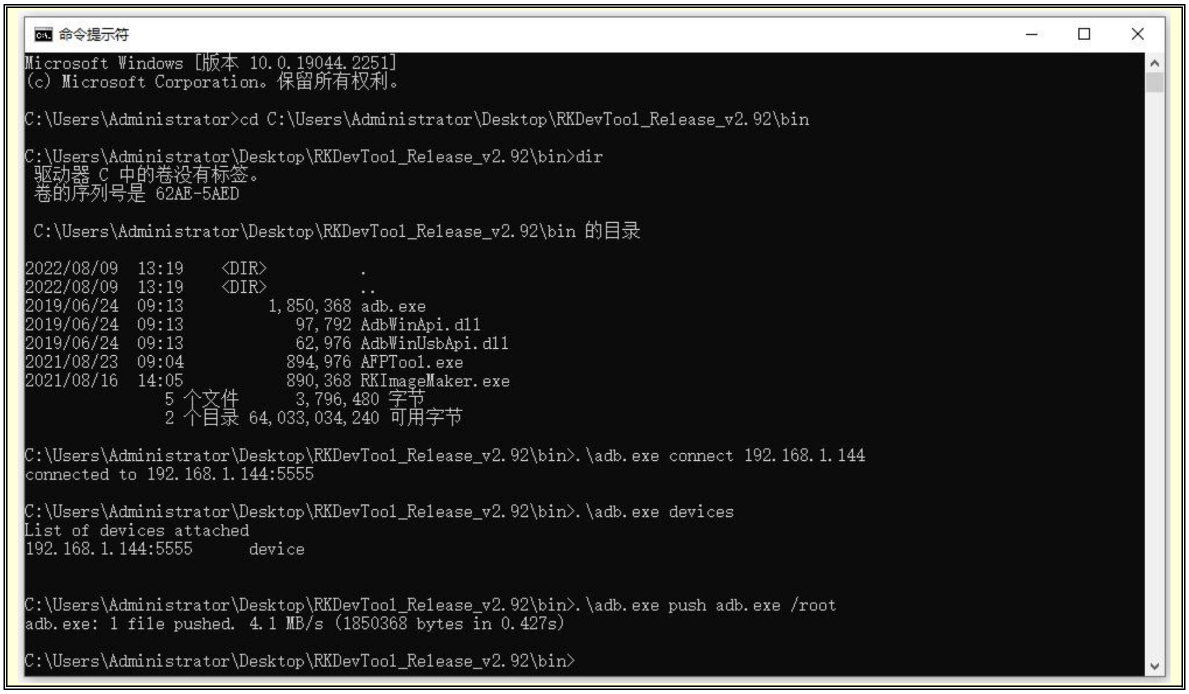
```
test@test:~$ adb push filename /root
filename: 1 file pushed. 3.7 MB/s (1075091 bytes in 0.277s)
```

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.

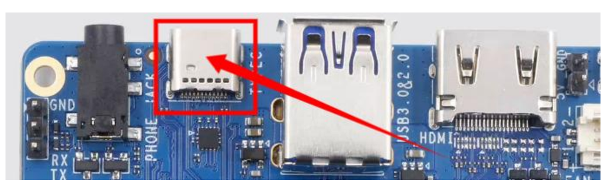
An example using adb in Windows is shown below:



3.9.2. Connect adb using Type-C data cable

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

2) Then connect the development board to the Ubuntu PC via a Type-C data cable. The location of the Type-C interface on the development board is shown in the figure below:



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

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

If the set_device.sh script does not exist in the Linux system, 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
```

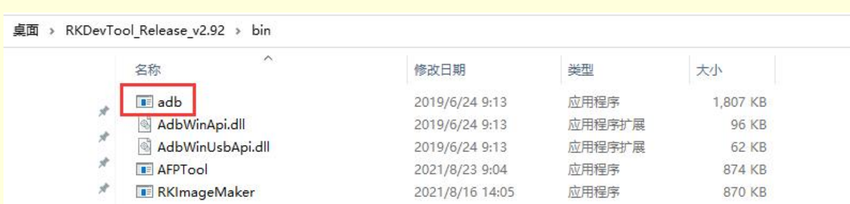
7) Then use the following command to log in to the Linux system of the development board.

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

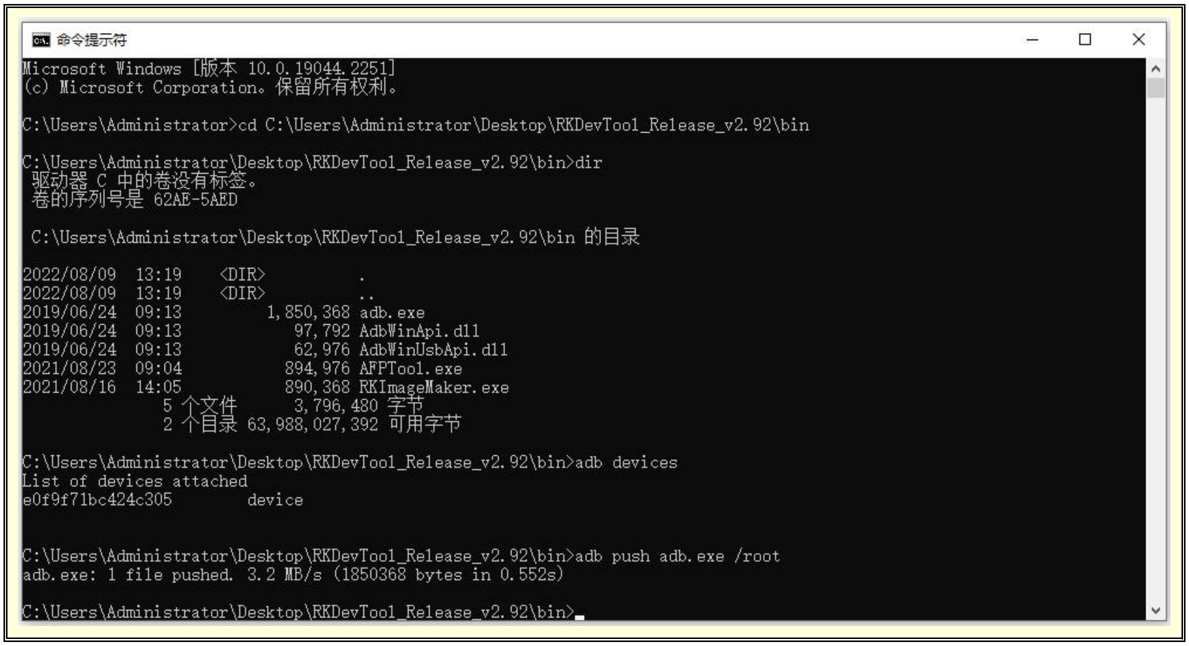
8) The command to upload files to the development board using adb is as follows:

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

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



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

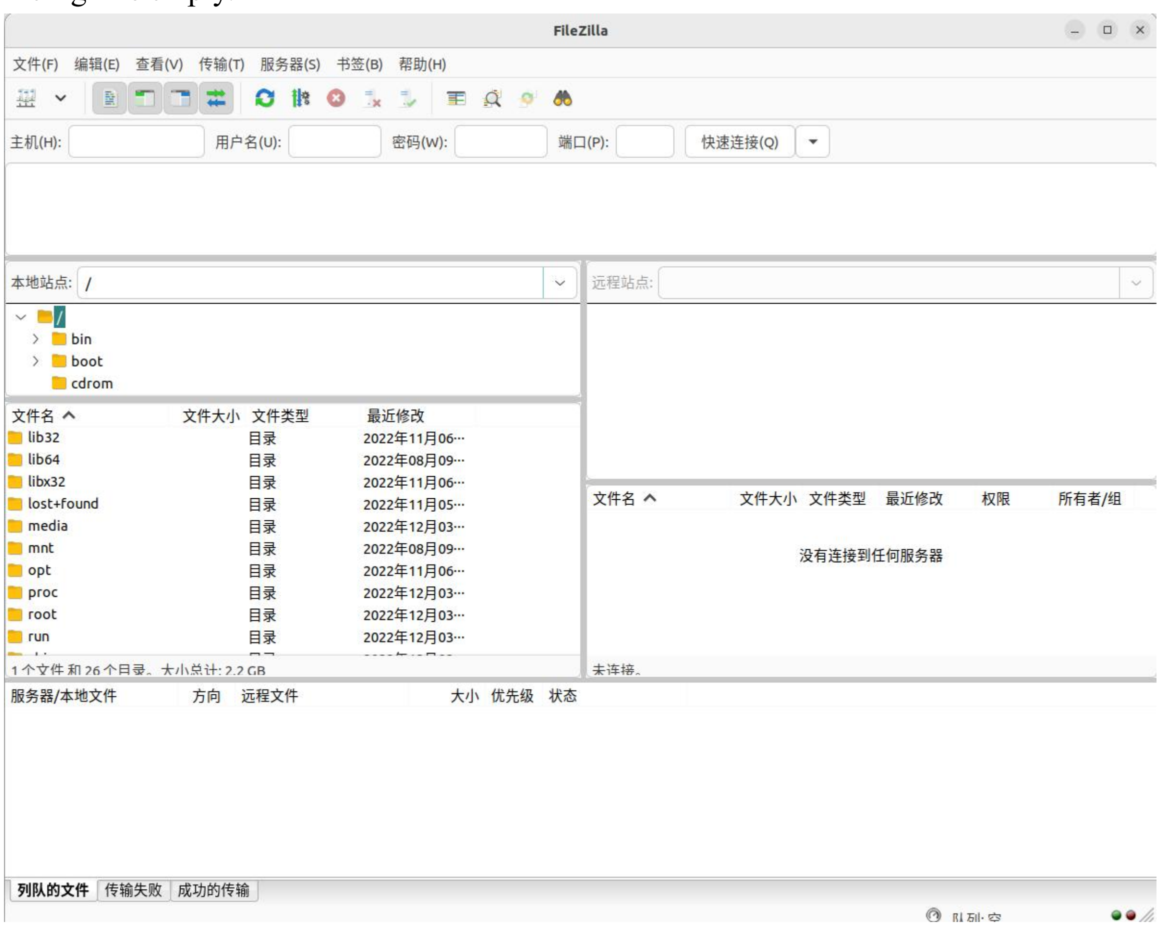
1) First install filezilla in your Ubuntu PC.

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

2) Then open filezilla using the command below.

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

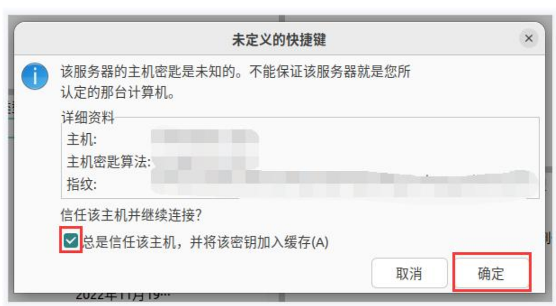
3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.



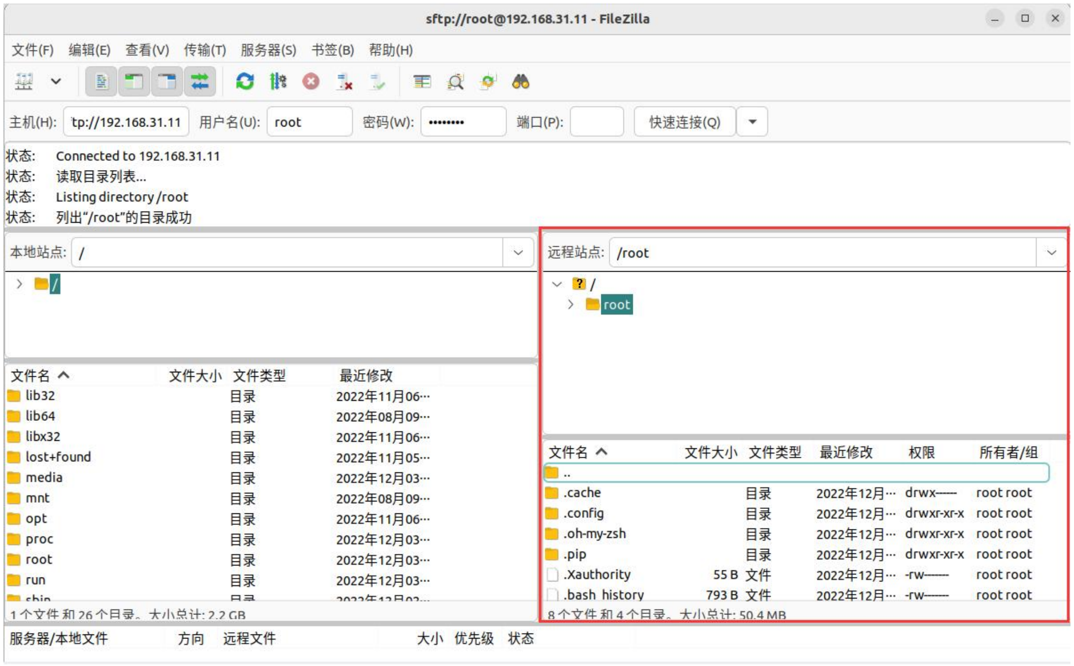
4) The method of connecting the development board is shown in the figure below:

5) Then select Save Password and click OK.

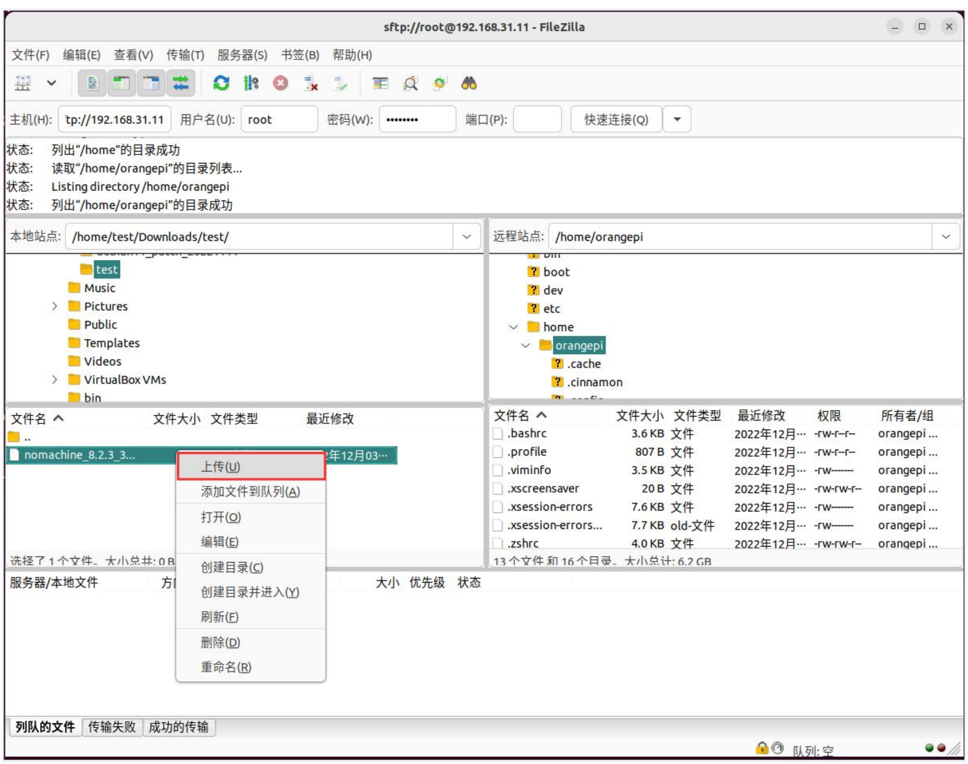
6) Then select Always trust this host and click OK.



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.

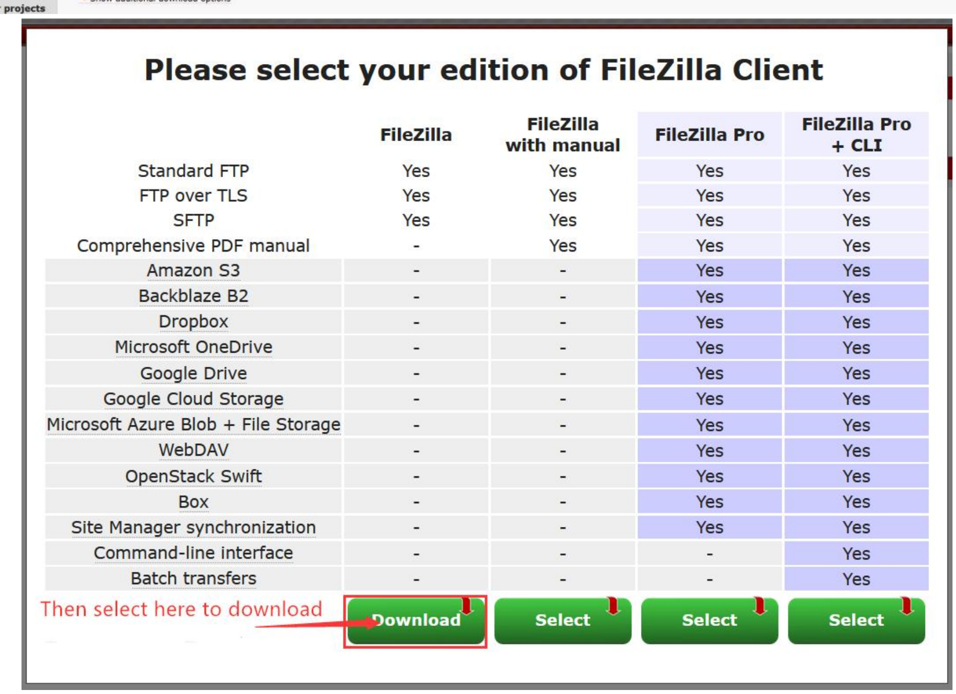
10) The method for uploading a folder is the same as that for uploading a file, so I will not go into details here.

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



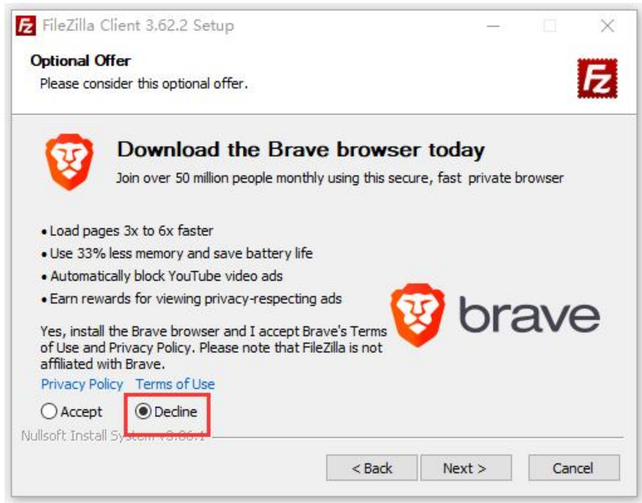
Please select your edition of FileZilla Client

	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

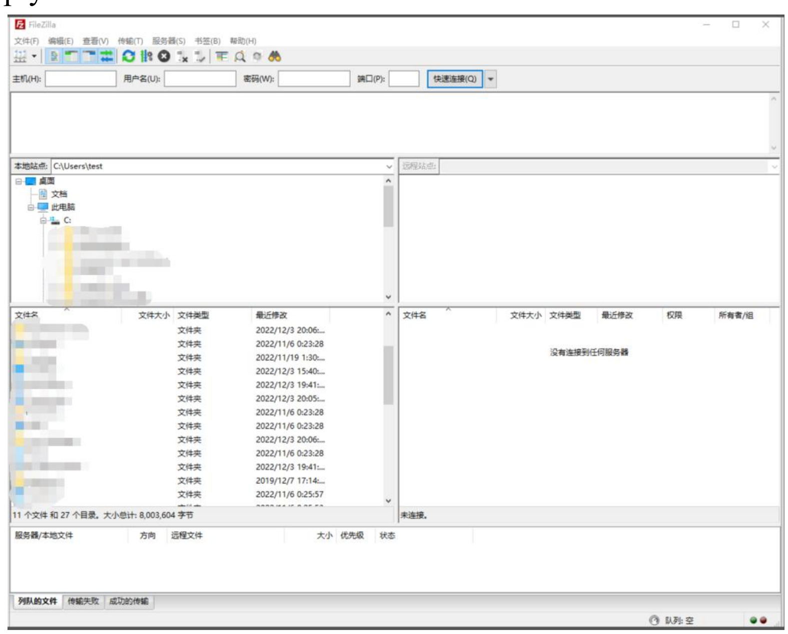
2) The downloaded installation package is as shown below, then double-click to install directly.

FileZilla_Server_1.5.1_win64-setup.exe

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

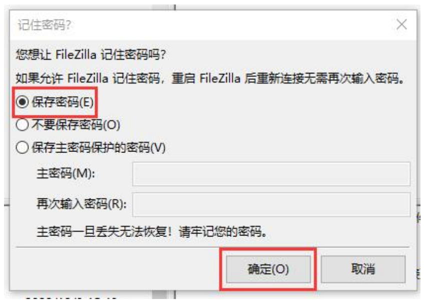


3) The interface after opening filezilla is as shown below. At this time, the remote site on the right is empty.



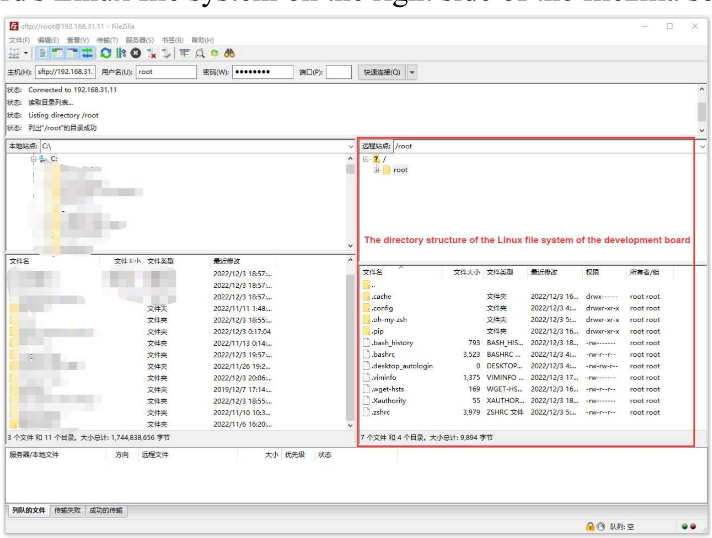
4) The method of connecting the development board is shown in the figure below:

5) Then select Save Password and click OK.

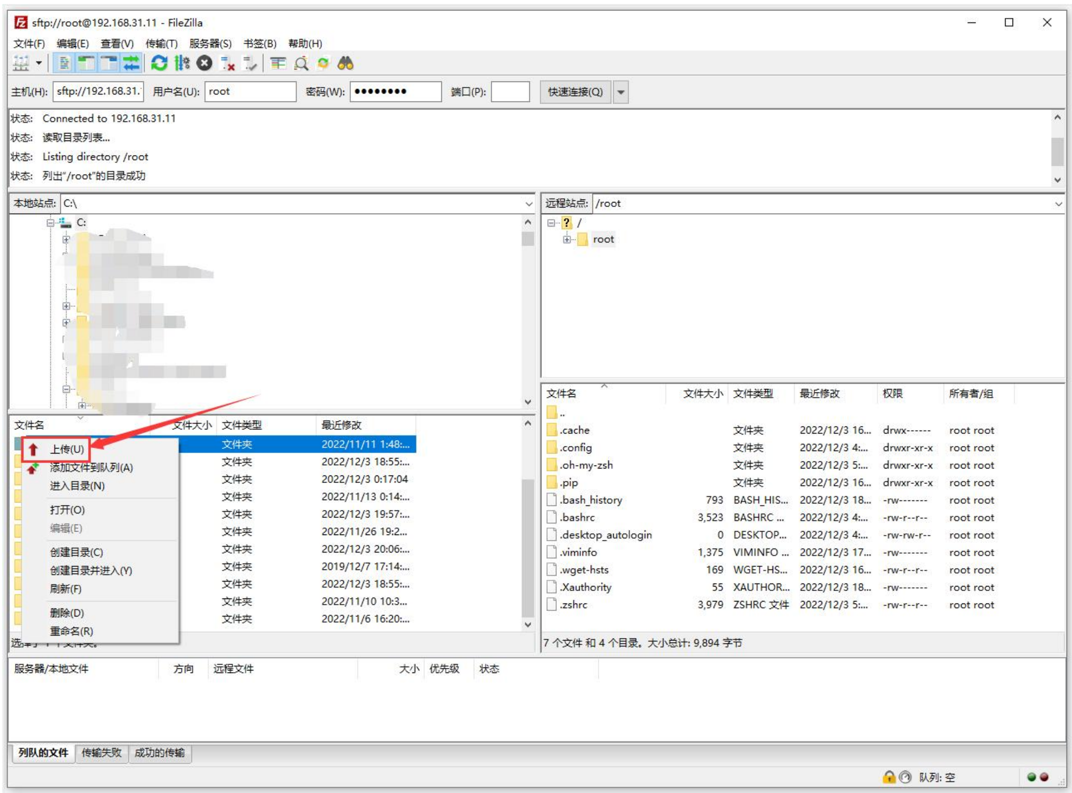


6) Then select Always trust this host and click OK.

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 HDMI to HDMI cable to connect Orange Pi development board and HDMI display

2) After starting the Linux system, if the HDMI monitor has image output, it means that the HDMI interface is working 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 to VGA display test

1) First, you need to prepare the following accessories.

a. HDMI to VGA converter.

b. A VGA cable.

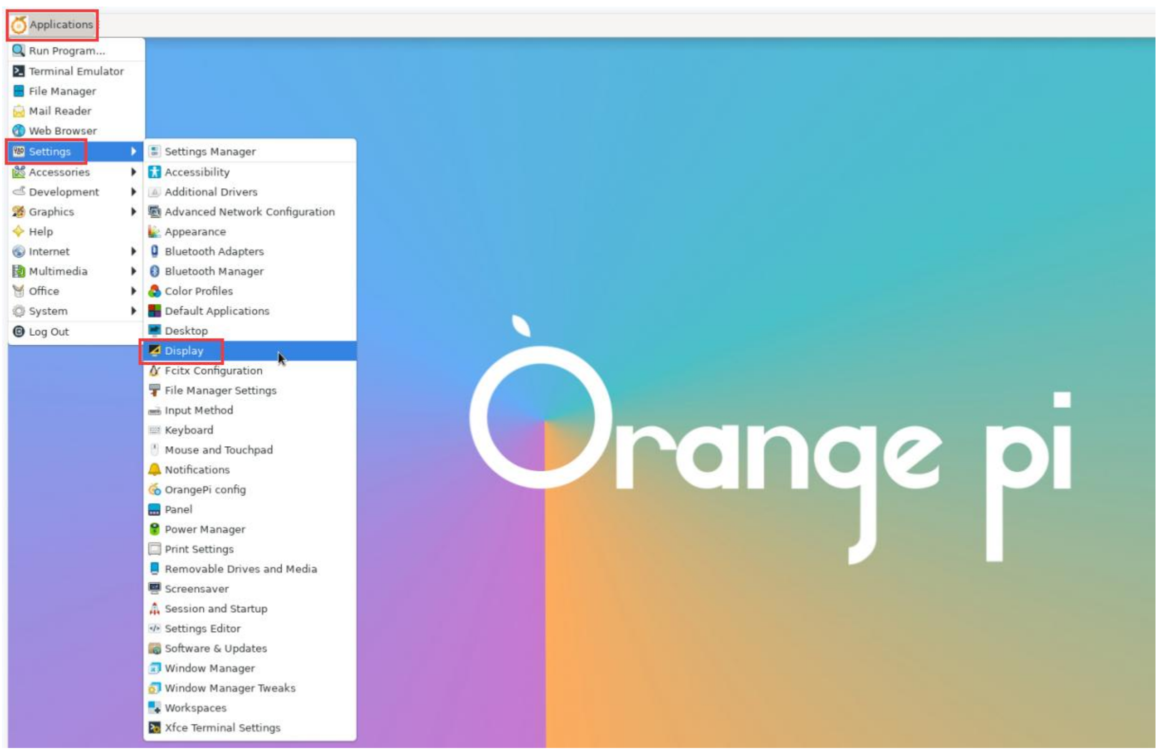
c. A monitor or TV that supports VGA interface.

2) One end of the VGA cable needs to be connected to the monitor or TV, and the other end of the VGA cable needs to be connected to the VGA port of the HDMI to VGA converter. Finally, the HDMI port of the HDMI to VGA converter needs to be plugged into the HDMI port of the development board. If everything is normal, you can see the display on the monitor after the development board is started.

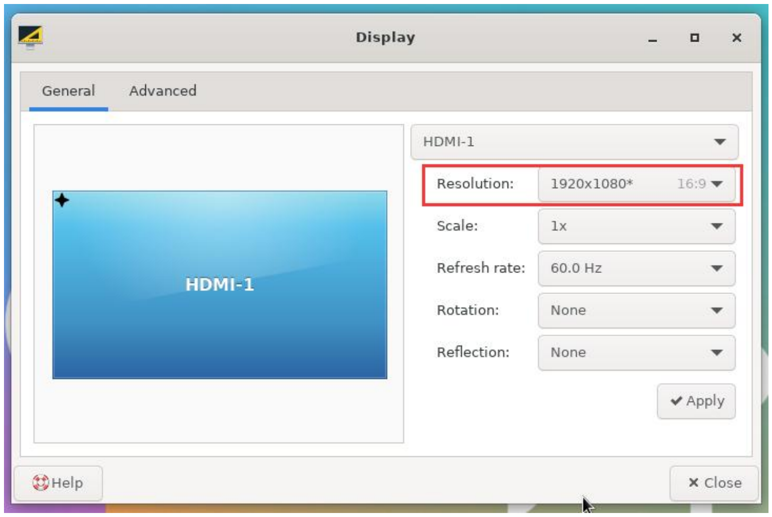
When using HDMI to VGA display, the development board and the Linux system of the development board do not need to do any settings, as long as the HDMI interface of the development board can display normally. So if there is a problem with the test, please check whether there is a problem with the HDMI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

3. 11. 3. HDMI resolution setting method

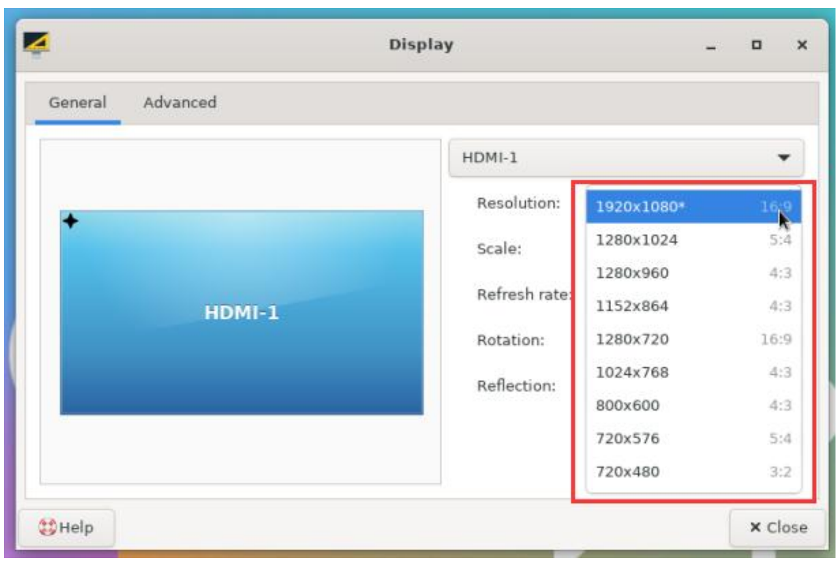
1) First, open Display in Settings.



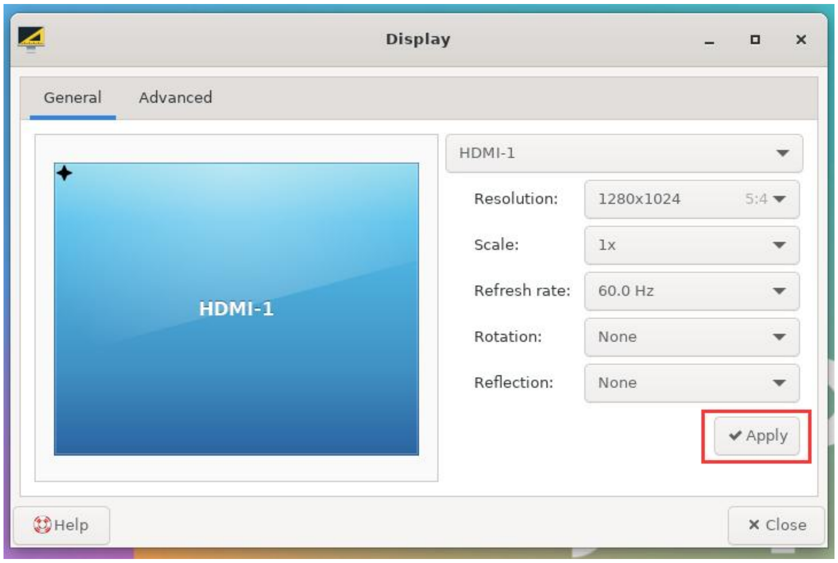
2) Then you can see the current resolution of the system.



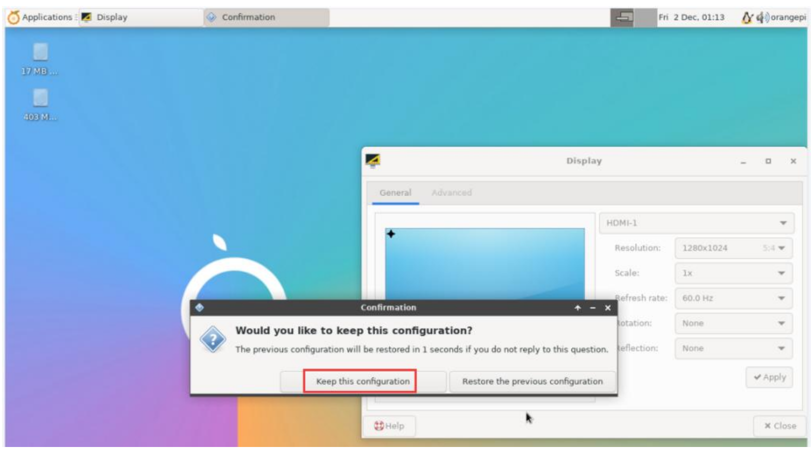
3) Click the drop-down box of Resolution to see all the resolutions currently supported by the monitor.



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



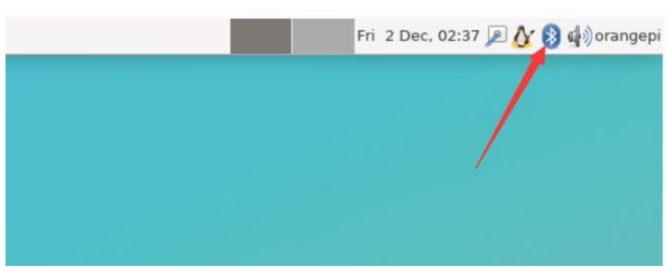
5) After the new resolution is set, select **Keep the configuration**.



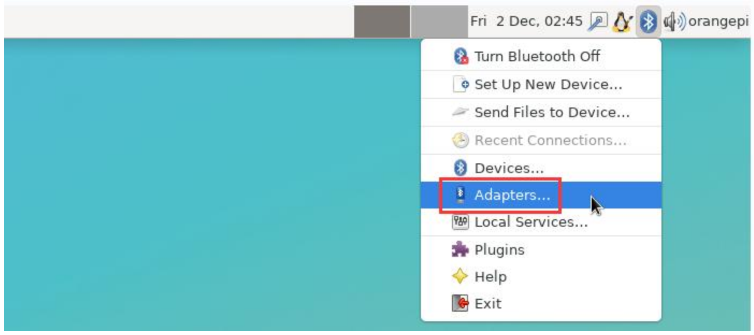
3. 12. How to use Bluetooth

3. 12. 1. Testing methods for desktop images

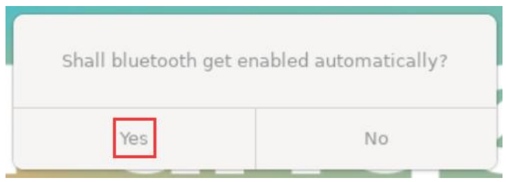
1) Click the Bluetooth icon in the upper right corner of the desktop.



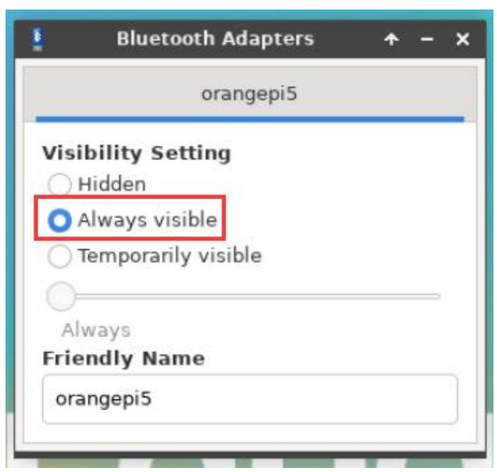
2) Then select the adapter.



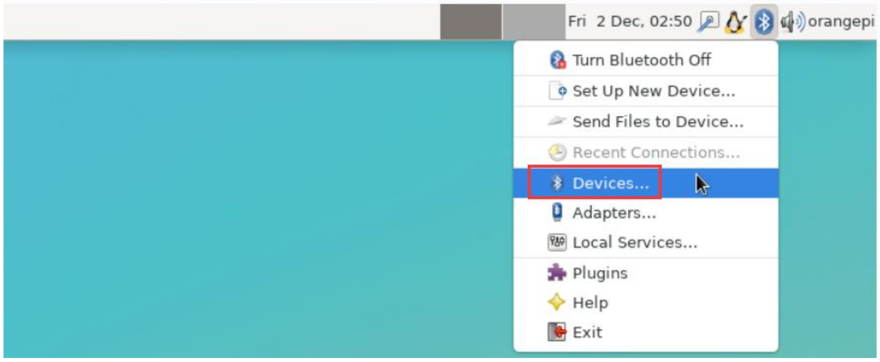
3) If the following interface is displayed, select **Yes**.



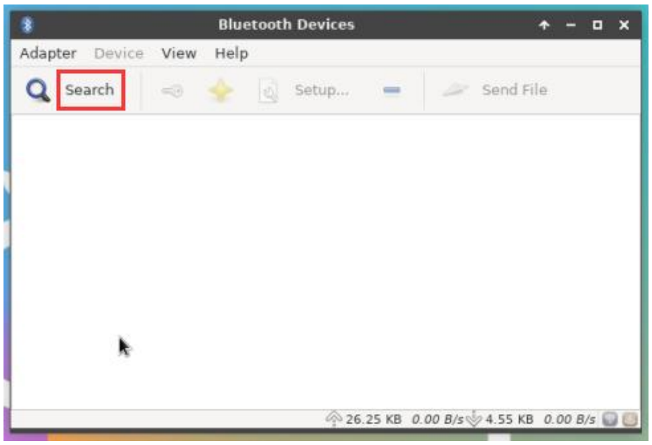
4) Then set the **Visibility Setting** to **Always visible** in the Bluetooth adapter settings interface, and then turn it off.



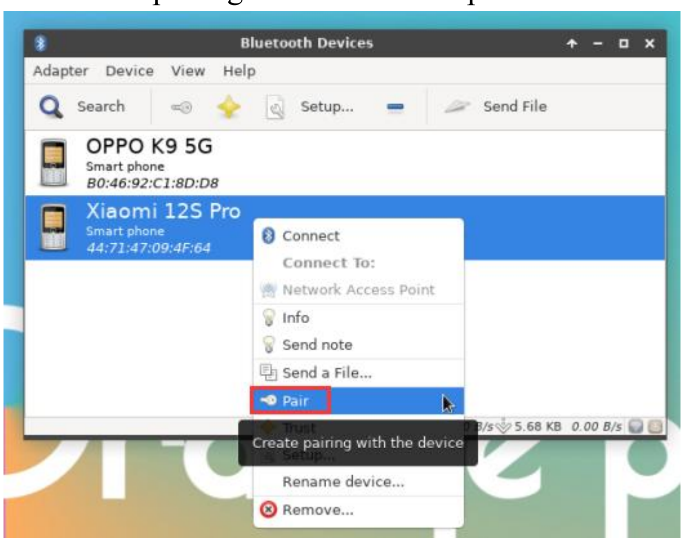
5) Then open the configuration interface of the Bluetooth device.



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

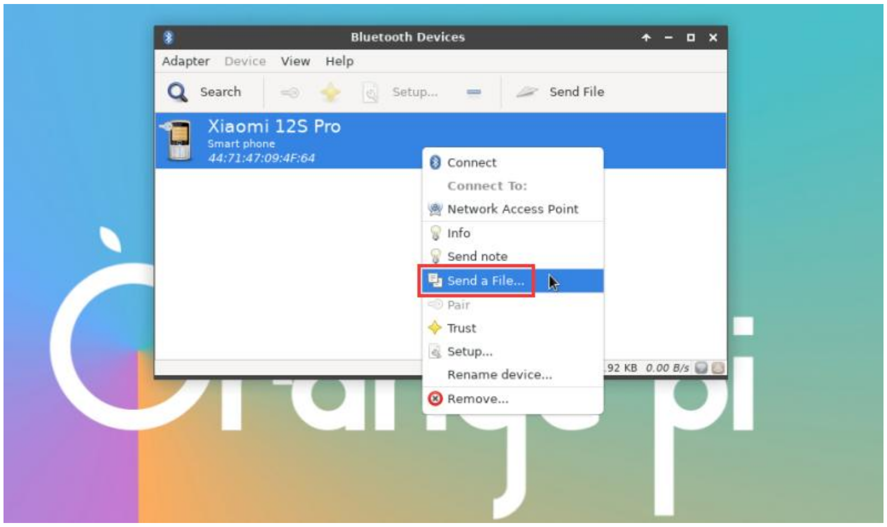


7) Then select the Bluetooth device you want to connect to, and then right-click the mouse to pop up the operation interface for this Bluetooth device. Select Pair to start pairing. Here we demonstrate pairing with an Android phone.

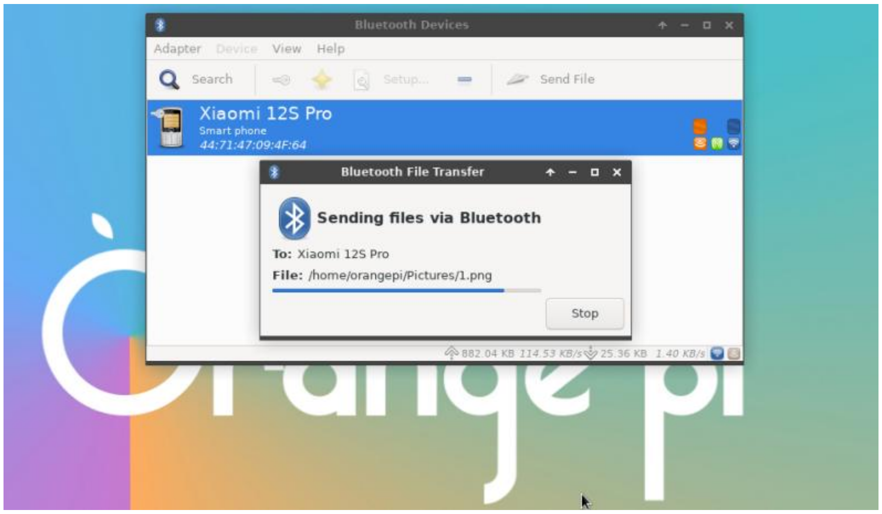


8) During pairing, a pairing confirmation box will pop up in the upper right corner of the desktop. Select **Confirm** to confirm. Confirmation is also required on the phone.

9) 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.



10) 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. Connecting USB mouse or keyboard test

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

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

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

3. 13. 2. Connecting USB storage device test

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

2) Execute the following command if you can see the output of sdX indicates that the USB flash drive is recognized successfully.

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

3) Using the mount command you can mount the USB flash drive into **/mnt**and then you can view the files on the USB flash drive.

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

4) After mounting, you can check the capacity usage and mount point of the USB flash drive by **df -h** command.

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

3. 13. 3. USB camera test

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

2) With v4l2-ctl command, you can see the device node information of 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
```

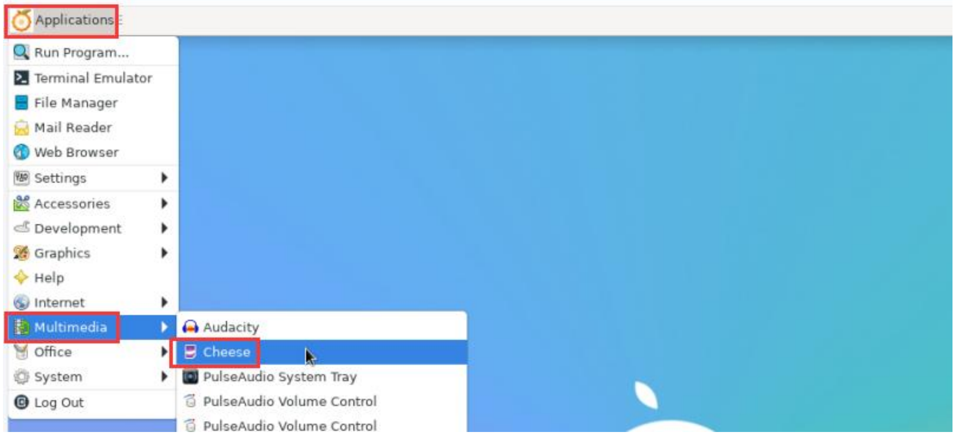
Note that the l in v4l2 is the lowercase letter l, not the number 1.

Also the serial number of video may not always be video0, please refer to what you actually see.

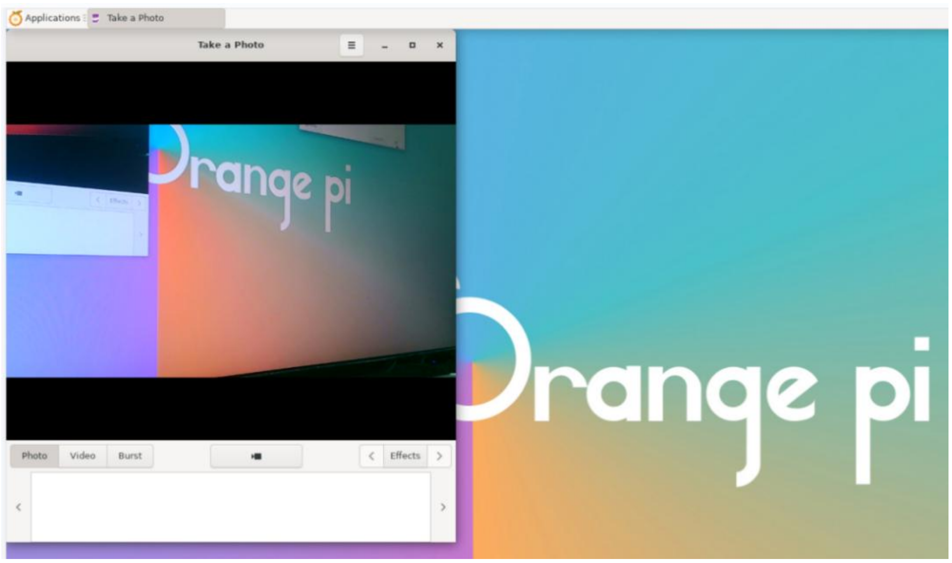
3) In the desktop system you can use Cheese to open the USB camera directly, Cheese open method is shown below:

If the system does not have cheese pre-installed, please use the following command to install it:

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



Cheese's interface after opening the USB camera is shown below:



4) Ways to test USB camera with 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 pictures.

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

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

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

d) 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 the linux system, you can use the scp command to transfer the captured picture to the Ubuntu PC for mirroring and viewing after taking the picture.

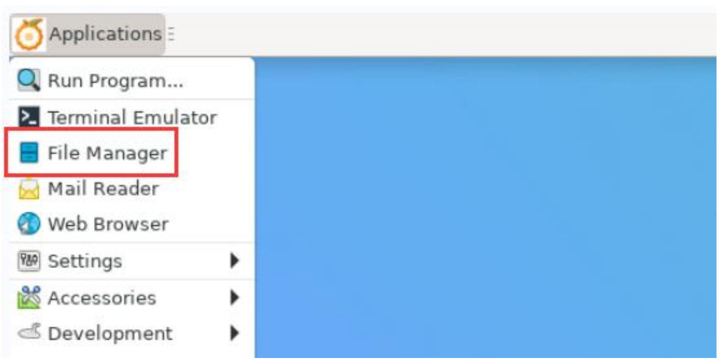
```
orangepi@orangepi:~$ scp image.jpg test@192.168.1.55:/home/test （Modify the IP address and path according to the actual situation）
```

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

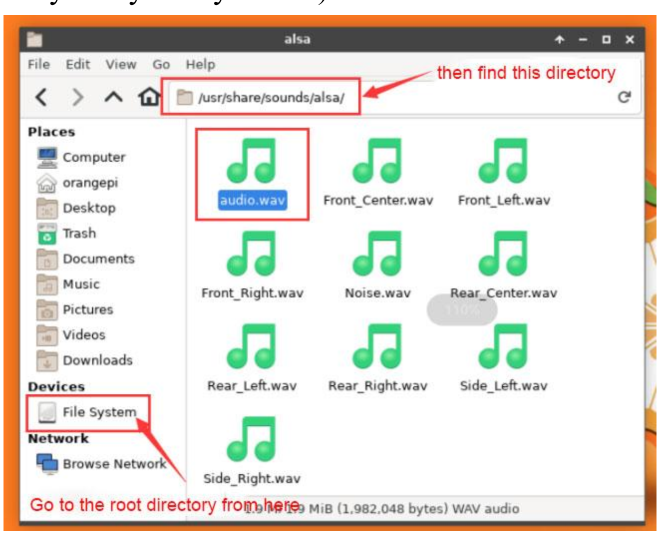
3. 14. Audio testing

3. 14. 1. Testing Audio Methods on a Desktop System

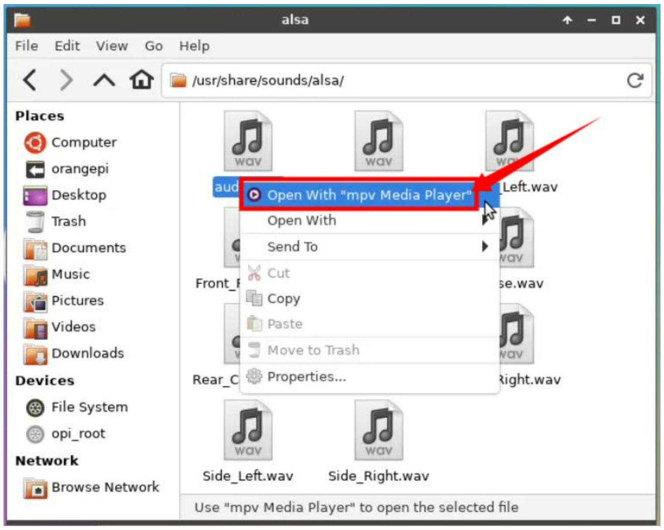
1) First open the file manager.



2) Then find the following file (if you don't have this audio file in your system, you can upload an audio file to your system yourself).

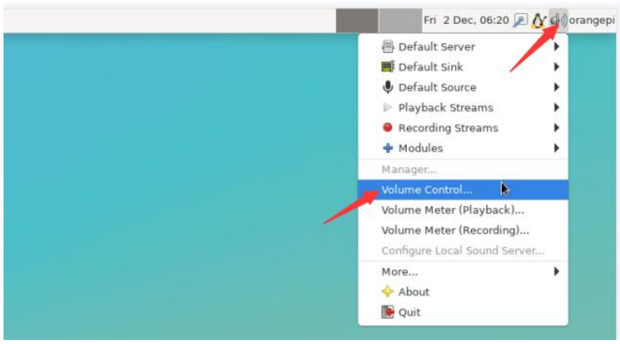


3) Then select the audio.wav file, right click and select open with vlc or mpv to start playing.

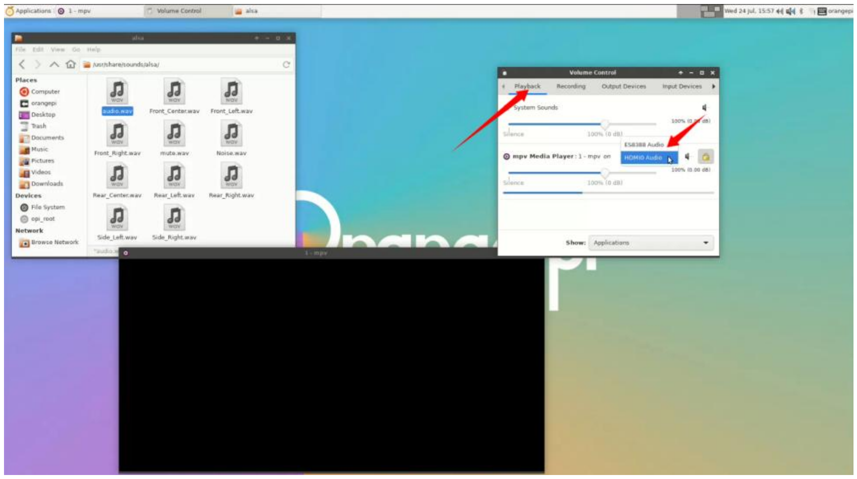


4) Method of switching between different audio devices such as HDMI playback and headphone playback.

a. First open the volume control interface.



b. When playing audio, the options of audio devices that can be used by the playback software will be displayed in **Playback**, as shown in the following figure, where you can set which audio device you need to play to.

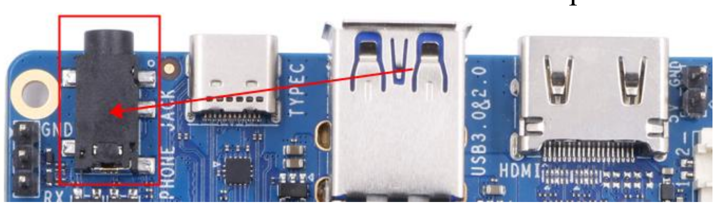


If you don't see the audio device option for HDMI, try unplugging the HDMI cable.

3. 14. 2. Using Commands to Play Audio

3. 14. 2. 1. Headphone jack playback audio test

1) First insert the headset into the headset hole of the development board.



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

```
orangepi@orangepi:~$ aplay -l
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
card 1: rockchiphdmi0 [rockchip-hdmi0], device 0: rockchip-hdmi0 i2s-hifi-0 [rockchip-hdmi0 i2s-hifi-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
card 2: rockchipes8388 [rockchip,es8388], device 0: dailink-multicodecs ES8323.3-0011-0 [dailink-multicodecs ES8323.3-0011-0]
```

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

3) Then use the **aplay**command to play the audio file that comes with the system, if the headset can hear the sound that the hardware can be used normally.

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

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

1) First connect the Orange Pi development board to a TV using an HDMI to HDMI cable (other HDMI displays need to make sure they can play audio).

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

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

3) Then use the **aplay** command to play the system comes with audio files, if the HDMI monitor or TV can hear the sound that the hardware can be used normally.

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

3. 14. 2. 3. DP Audio Playback Tests

1) First connect the Orange Pi development board to your TV using a Type-C to HDMI cable (other HDMI displays need to make sure they can play audio).

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

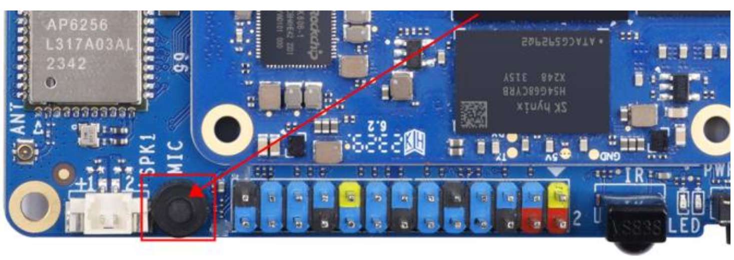
```
orangepi@orangepi:~$ aplay -l
card 0: rockchipdp0 [rockchip-dp0], device 0: rockchip-dp0 spdif-hifi-0 [rockchip-dp0 spdif-hifi-0]
  Subdevices: 0/1
  Subdevice #0: subdevice #0
......
```

3) Then use the**aplay** command to play the system comes with audio files, if the HDMI monitor or TV can hear the sound that the hardware can be used normally.

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

3. 14. 3. Using Commands to Test Recordings

1) There is an on-board MIC on the development board at the location shown below:



2) Running the **test_record.sh main** command will record a piece of audio through the on-board MIC and play it back to the HDMI and headphones.

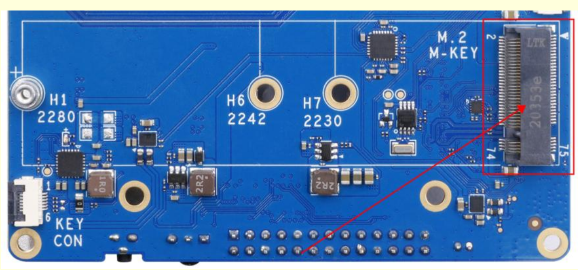
```
orangepi@orangepi:~$ test_record.sh main
Start recording: /tmp/test.wav
Recording WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Start playing
Playing WAVE '/tmp/test.wav' : Signed 16 bit Little Endian, Rate 44100 Hz, Stereo
Playing WAVE '/tmp/test.wav' : Signed 16 bit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

3) In addition to the on-board MIC, we can also record audio through headphones with MIC function. After plugging the headset with MIC function into the development board, running the **test_record.sh headset** command will record a piece of audio through the headset, and then play it back to HDMI and headset.

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

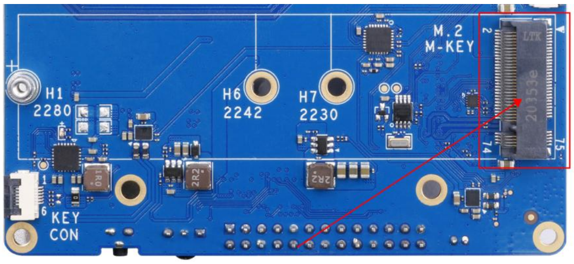
3. 15. Using SATA SSD's

The m.2 interface shown below can use both nvme ssd and sata ssd. since the pcie2.0 controller and the sata controller are one or the other, you can only turn on one of the configurations at the same time. the linux image released by Orange Pi turns on the pcie configurations by default, so it can only recognize nvme ssd by default. if you want to use sata ssd, you need to turn on the corresponding configurations to do so. you want to use sata ssd, you need to open the corresponding configuration.



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

2) Then insert the SSD into the M.2 port of the development board and secure it.

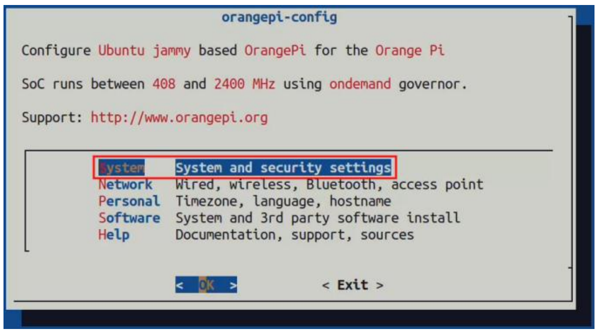


3) sata ssd usage is currently mainly as an extended storage device.

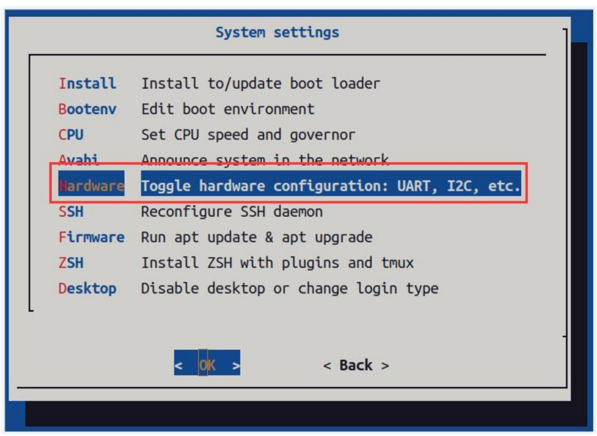
4) Then run **orangepi-config**, ordinary users remember to add **sudo** privileges.

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

5) Then select **System**.



6) Then select **Hardware**.

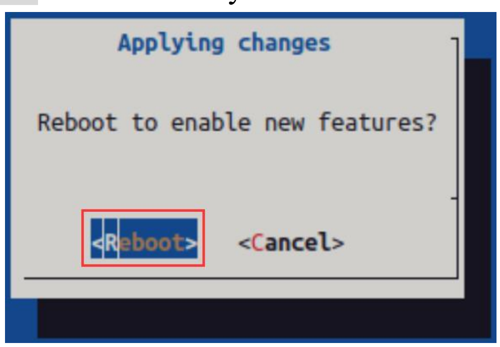


7) Then use the arrow keys on your keyboard to locate **ssd-sata0**, and then use the spacebar to select the

8) Then select **<Save>** to save.

9) Then select **<Back>**.

10) Then select **<Reboot>** to reboot the system to make the configuration take effect.



The above setup will end up in/boot/orangepiEnv.txt with the overlays=ssd-sata0 line of configuration. You can check this first after setting it up. If this line does not exist, then there is something wrong with the setup.

If you find it troublesome to use orangepi-config, you can also open /boot/orangepiEnv.txt and add the line configuration of overlays=ssd-sata0 as well.

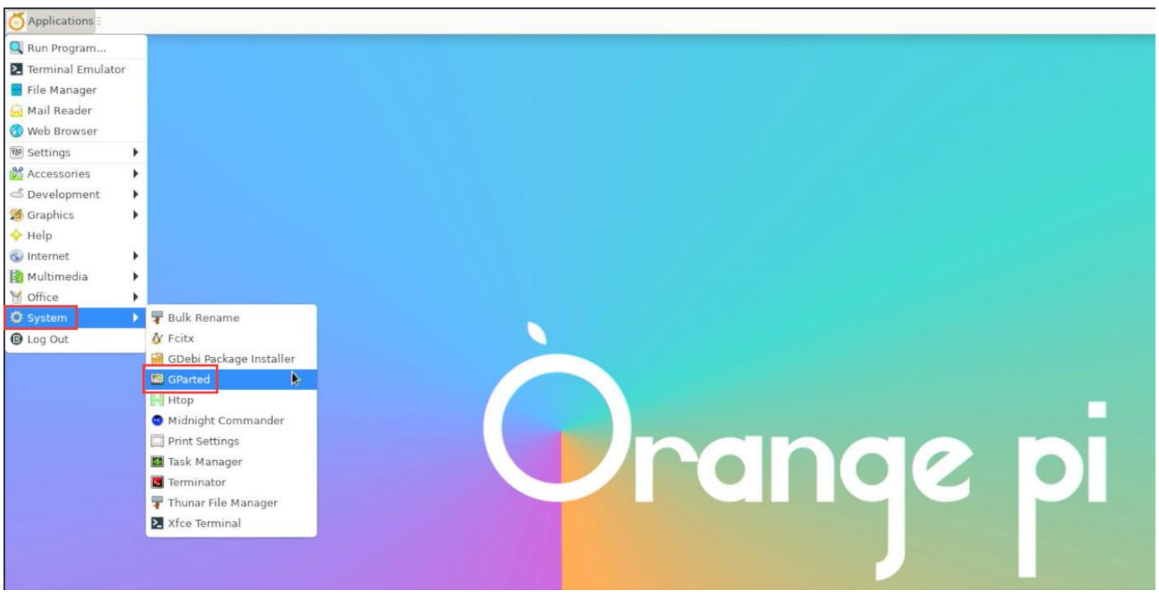
orangepi@orangepi:~$ **cat /boot/orangepiEnv.txt | grep ssd**
overlays=ssd-sata0

11) If everything is fine, use **sudo fdisk -l** command after system reboot to see the information of sata ssd.

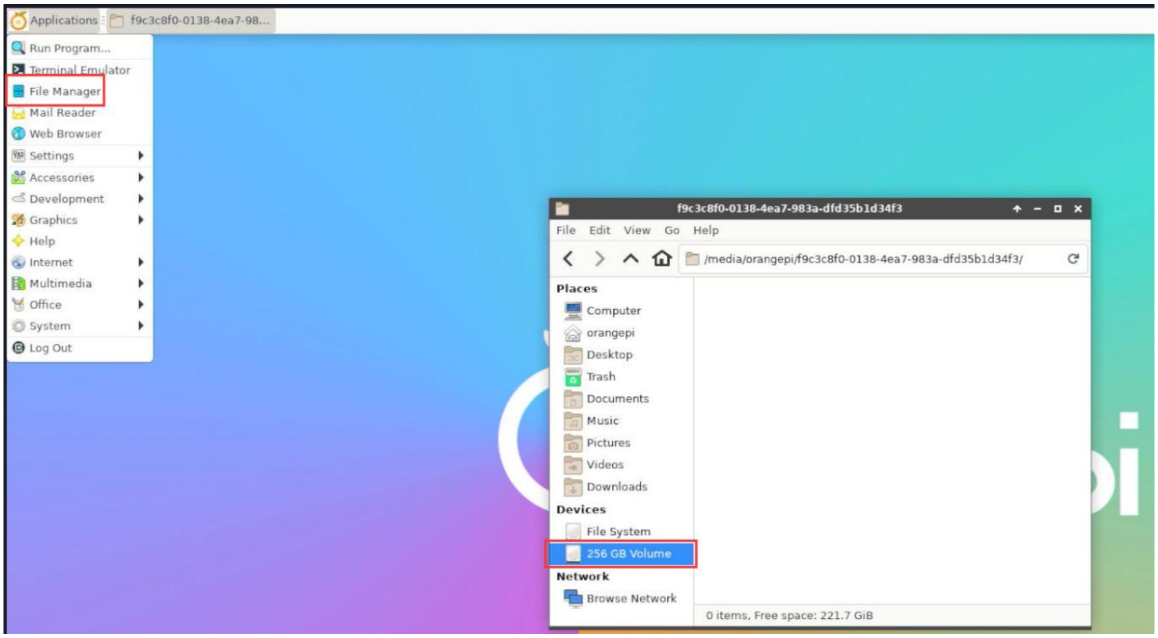
orangepi@orangepi:~$ **sudo fdisk -l**

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

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

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



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



14) The mount command can be used in the server version of the system to **mount** sata ssd to the desired directory.

```
orangepi@orangepi:~$ sudo mount /dev/sda1 /mnt
orangepi@orangepi:~$ df -h
Filesystem          Size   Used Avail Use% Mounted on
udev                 3.8G   8.0K   3.8G    1% /dev
tmpfs                769M   1.4M   768M    1% /run
/dev/mmcblk1p2        29G   5.9G    23G   21% /
tmpfs                3.8G      0   3.8G    0% /dev/shm
tmpfs                5.0M   4.0K   5.0M    1% /run/lock
tmpfs                3.8G    16K   3.8G    1% /tmp
/dev/mmcblk1p1       256M    90M   166M   36% /boot
/dev/zram1           194M    27M   154M   15% /var/log
tmpfs                769M    60K   769M    1% /run/user/1000
/dev/sda1            234G    28K   222G    1% /mnt
```

3. 16. Temperature sensors

1) 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)
```

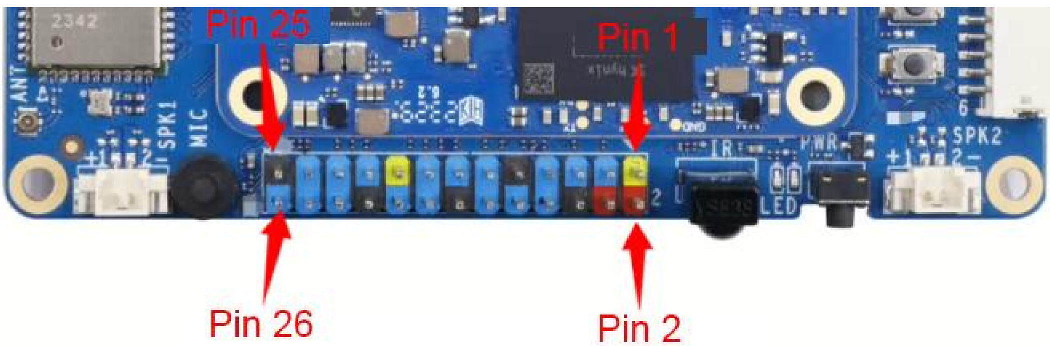
2) The command to view the current temperature of nvme ssd SSD is:

```
orangepi@orangepi:~$ sudo smartctl -a /dev/nvme0 | grep "Temperature:"
```

Temperature:	**40 Celsius**

3. 17. 26 Pin Interface Pin Description

1) Please refer to the following diagram for the order of the 26 Pin interface pins on the Orange Pi CM5 Base Tablet base board.



2) The functions of the 26 Pin interface pins on the Orange Pi CM5 Base Tablet base board are shown in the table below:

a. Below is the complete 26 Pin pinout.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

b. The table below is a graph of the left half of the complete table above, so you can see it more clearly.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
			3.3V		1
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7
			GND		9
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13
			GPIO1_B0	40	15
			3.3V		17
		SPI0_MOSI_M2	GPIO1_B2	42	19
		SPI0_MISO_M2	GPIO1_B1	41	21
		SPI0_CLK_M2	GPIO1_B3	43	23
			GND		25

c. The table below is a graph of the right half of the complete table above to give a clearer view.

引脚序号	GPIO序号	GPIO	复用功能	复用功能
2		5V		
4		5V		
6		GND		
8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
12	34	GPIO1_A2	I2C4_SDA_M3	
14		GND		
16	36	GPIO1_A4		
18	38	GPIO1_A6		
20		GND		
22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

The pwm's in the table above are labeled with the base address of the corresponding register, which is useful when looking at which pwmchip in /sys/class/pwm/ corresponds to which pwm pin in the 26-pin pinout.

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

3. 18. Method of installing wiringOP

Note that wiringOP is pre-installed in the linux image released by Orange Pi. Unless there is an update to the wiringOP code, there is no need to re-download,

compile and install it, just use it directly.

The path to the compiled deb package of wiringOP in orangepi-build is: orangepi-build/external/cache/debs/arm64/wiringpi_x.xx.deb

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

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |    OUT | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |    OUT | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |    OUT | 1 |  7 || 8  | 1 | OUT    | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | OUT    | RXD.6    | 4   | 57   |
 |   58 |   5 |    RXD.4 |    OUT | 1 | 11 || 12 | 1 | OUT    | GPIO1_A2 | 6   | 34   |
 |   59 |   7 |    TXD.4 |    OUT | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |    OUT | 1 | 15 || 16 | 1 | OUT    | GPIO1_A4 | 9   | 36   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | OUT    | GPIO1_A6 | 10  | 38   |
 |   42 |  11 | SPI0_TXD |    OUT | 1 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |    OUT | 1 | 21 || 22 | 1 | OUT    | PWM3     | 13  | 39   |
 |   43 |  14 | SPI0_CLK |    OUT | 1 | 23 || 24 | 1 | OUT    | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | OUT    | SPI0_CS1 | 16  | 45   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
orangepi@orangepicm5-tablet:~$
```

1) Download the code for wiringOP.

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

Note that you need to download the code for the wiringOP next branch, please don't miss the -b next argument.

If you have problems downloading the code from GitHub, you can directly use the wiringOP source code that comes with the Linux image, stored at /usr/src/wiringOP.

2) Compile and install wiringOP.

```
orangepi@orangepi:~$ cd wiringOP
```

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

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

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |    OUT | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |    OUT | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |    OUT | 1 |  7 || 8  | 1 | OUT    | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | OUT    | RXD.6    | 4   | 57   |
 |   58 |   5 |    RXD.4 |    OUT | 1 | 11 || 12 | 1 | OUT    | GPIO1_A2 | 6   | 34   |
 |   59 |   7 |    TXD.4 |    OUT | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |    OUT | 1 | 15 || 16 | 1 | OUT    | GPIO1_A4 | 9   | 36   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | OUT    | GPIO1_A6 | 10  | 38   |
 |   42 |  11 | SPI0_TXD |    OUT | 1 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |    OUT | 1 | 21 || 22 | 1 | OUT    | PWM3     | 13  | 39   |
 |   43 |  14 | SPI0_CLK |    OUT | 1 | 23 || 24 | 1 | OUT    | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | OUT    | SPI0_CS1 | 16  | 45   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
orangepi@orangepicm5-tablet:~$
```

3.19. 26pin interface GPIO, I2C, UART, SPI and PWM tests

3.19.1. 26pin GPIO port test

The Orange Pi is shipped with a pre-installed linux program called blink_all_gpio, which sets all 17 GPIO ports in the 26-pin to toggle high and low constantly.

After running the blink_all_gpio program, when using a multimeter to measure the level of the GPIO port, you will find that the GPIO pin will keep switching between 0 and 3.3v. Using this program we can test if the GPIO port is working properly.

The way to run the blink_all_gpio program is shown below:

```
orangepi@orangepi:~$ sudo blink_all_gpio          #Remember to add sudo privileges
[sudo] password for orangepi:                     #Here you need to enter the password
```

1) There are a total of 17 GPIO ports available in the 26pin of the development board,

and the following is an example of how to set the high and low levels of the GPIO ports on pin 7 - corresponding to GPIO1_A3 - corresponding to the wPi serial number 2-. -As an example, we will demonstrate how to set the high and low levels of the GPIO port.

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |     IN | 0 |  7 || 8  | 1 | IN     | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | IN     | RXD.6    | 4   | 57   |
 |   58 |   5 |    RXD.4 |     IN | 1 | 11 || 12 | 0 | IN     | GPIO1_A2 | 6   | 34   |
 |   59 |   7 |    TXD.4 |  ALT11 | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |     IN | 1 | 15 || 16 | 0 | IN     | GPIO1_A4 | 9   | 36   |
```

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

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

3) Then set the GPIO port output low level, after the setup you can use a multimeter to measure the value of the pin's voltage, if it is 0v, it means that the setup low level is successful.

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

Using gpio readall you can see that the value (V) of pin 7 has changed to 0.

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |    OUT | 0 |  7 || 8  | 1 | IN     | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | IN     | RXD.6    | 4   | 57   |
```

4) Then set the GPIO port output high level, after the setup you can use a multimeter to measure the value of the pin's voltage, if it is 3.3v, it means that the setup high level is successful.

root@orangepi:~/wiringOP# **gpio write 2 1**

Using gpio readall you can see that the value (V) of pin 7 has changed to 1.

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |    OUT | 1 |  7 || 8  | 1 | IN     | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | IN     | RXD.6    | 4   | 57   |
```

5) The setting method for other pins is similar, just modify the serial number of wPi to the corresponding serial number of the pin.

3. 19. 2. 26pin GPIO port pull-up and pull-down resistor setting method

1) The following is an example of how to set the pull-up and pull-down resistors of the GPIO port on pin 7 - corresponding to the GPIO as GPIO1_A3 - corresponding to the wPi serial number of 2. .

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |     IN | 0 |  7 || 8  | 1 | IN     | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | IN     | RXD.6    | 4   | 57   |
 |   58 |   5 |    RXD.4 |     IN | 1 | 11 || 12 | 0 | IN     | GPIO1_A2 | 6   | 34   |
 |   59 |   7 |    TXD.4 |  ALT11 | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |     IN | 1 | 15 || 16 | 0 | IN     | GPIO1_A4 | 9   | 36   |
```

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

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

3) After setting to input mode, execute the following command to set the GPIO port to pull-up mode.

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

4) Then input the following command to read the level of GPIO port, if the level is 1, it means the pull-up mode is set successfully.

```
root@orangepi:~/wiringOP# gpio read 2
1
```

5) Then execute the following command to set the GPIO port to pull-down mode.

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

6) Then enter the following command to read the level of GPIO port, if the level is 0, it means the pull-down mode is set successfully.

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

3. 19. 3. 26pin I2C test

1) As shown in the table below, the available i2c for the Orange Pi CM5 Base Tablet are two sets of i2c buses, i2c4 and i2c5.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

2) The corresponding pins of the two I2C buses in 26pin are shown in the table below:

I2C bus	SDA corresponds to 26pin	SCL corresponds to 26pin	The dtbo counterpart configuration
I2C4_M3	Pin 12	Pin 7	i2c4-m3
I2C5_M3	Pin 3	Pin 5	i2c5-m3

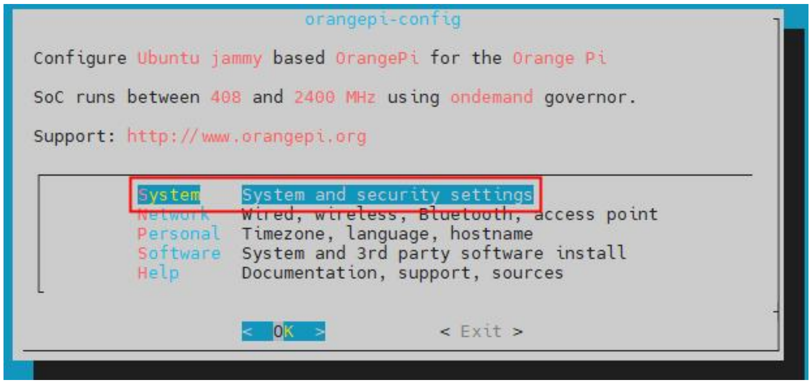
3) In linux, the I2C buses in 26 Pin are all turned off by default and need to be turned on manually before they can be used.

The detailed steps are shown below:

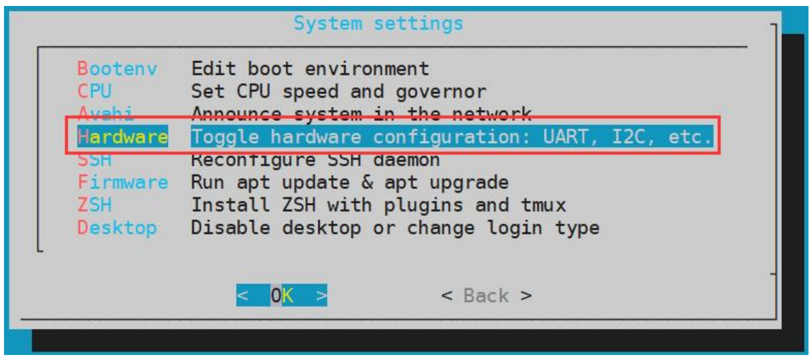
a. First run **orangepi-config**, ordinary users remember to add **sudo** privileges.

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

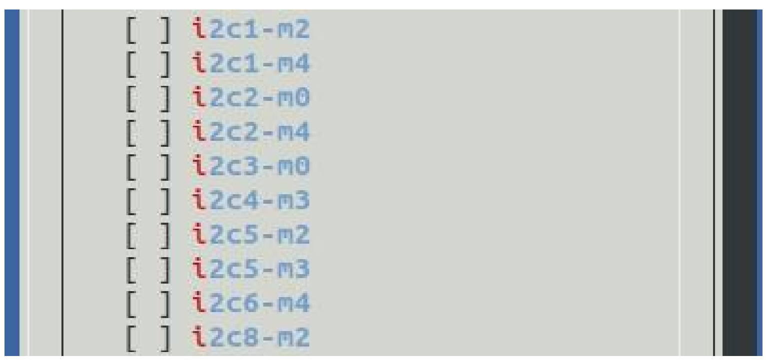
b. Then select **System**.



c. Then select **Hardware**.

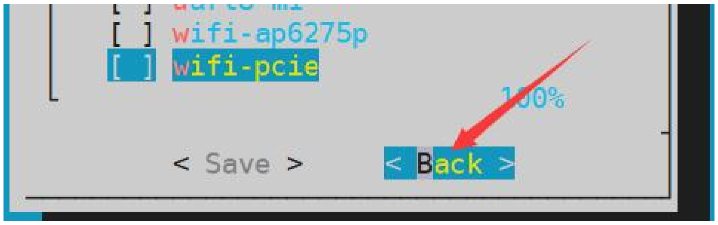


d. Then use the arrow keys of the keyboard to locate the position shown in the figure below, and then use the space to select the configuration of the I2C you want to open.

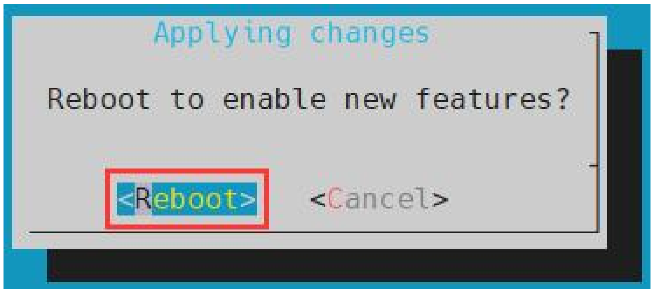


e. Then select**<Save>** to save.

f. Then select **<Back>**.



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



4) After booting the linux system, first confirm the existence of device nodes that need to use I2C under **/dev**.

```
orangepi@orangepi:~$ ls /dev/i2c-*
```

5) Then connect an i2c device to the i2c pin on the 26pin connector.

6) Then use the **i2cdetect -y** command If the address of the connected i2c device can be detected, it means that the i2c can be used normally.

```
orangepi@orangepi:~$ sudo i2cdetect -y 4        #i2c4 Command
orangepi@orangepi:~$ sudo i2cdetect -y 5        #i2c5 Command
```

3. 19. 4. 26pin UART test

1) From the table below, the uart available for Orange Pi CM5 Base Tablet are uart1, uart4, uart6 and uart7, four sets of uart bus.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

2) The corresponding pins of the four UART buses in 26pin are shown in the table below:

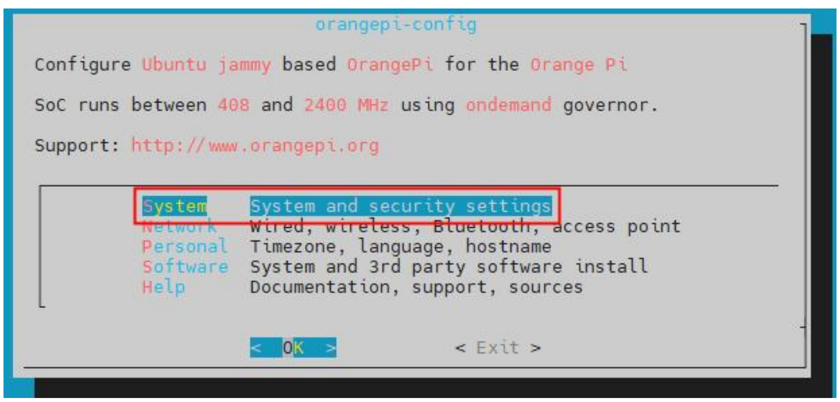
UART bus	RX corresponds to 26pin	TX corresponds to 26pin	dtbo corresponding configuration
UART1_M1	3pin	5pin	uart1-m1
UART4_M0	13pin	11pin	uart4-m0
UART6_M2	10pin	8pin	uart6-m2
UART7_M2	24pin	26pin	uart7-m2

3) In linux system, the UARTs in 26 Pin are all turned off by default and need to be turned on manually to be used. Detailed steps are shown below:

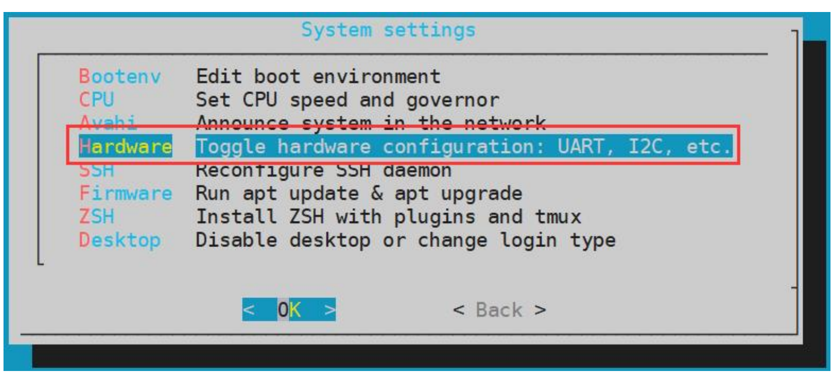
a. First run **orangepi-config**, ordinary users remember to add **sudo** privileges.

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

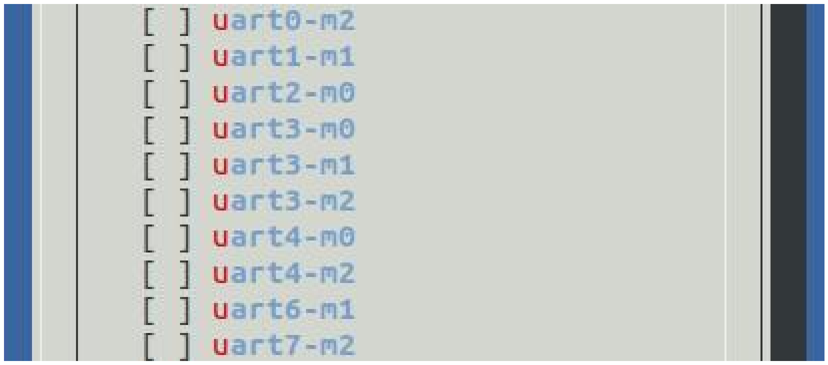
b. Then select **System**.



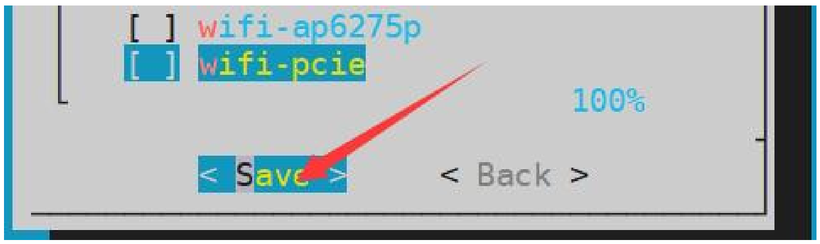
c. Then select **Hardware**.



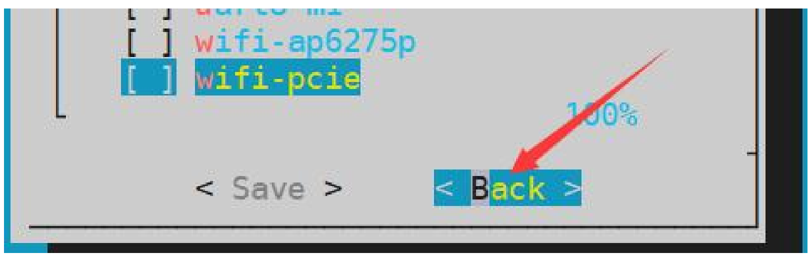
d. Then use the arrow keys of your keyboard to locate the position shown in the figure below, and then use space to select the configuration of the UART you want to open.



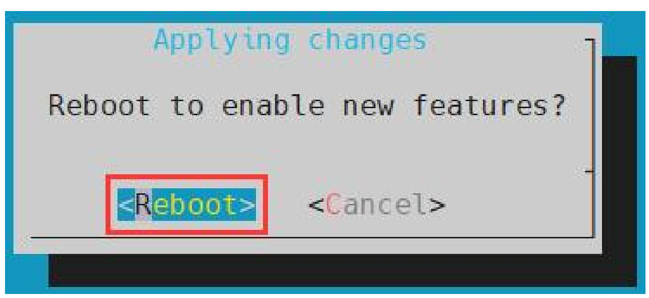
e. Then select **<Save>** to save.



f. Then select **<Back>**.



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



4) After entering the linux system, first confirm whether there is a device node corresponding to uart under /dev.

```
orangepi@orangepi:~$ ls /dev/ttyS*
```

5) Then start testing the uart interface, first use the DuPont cable to short the rx and tx of the uart interface to be tested.

6) Use the **gpio serial** command to test the loopback function of the serial port as shown below, if you can see the printout below, it means that the serial port communication is normal (ttySX needs to be replaced with the node name of the corresponding uart, please do not copy it).

```
orangepi@orangepi:~$ sudo gpio serial /dev/ttySX
[sudo] password for orangepi:    #Enter your password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

3. 19. 5. 26pin SPI test

1) From the table below, the available spi for Orange Pi CM5 Base Tablet are spi0 and spi1.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

2) The corresponding pins of SPI0 and SPI4 in 26pin are shown in the table below.

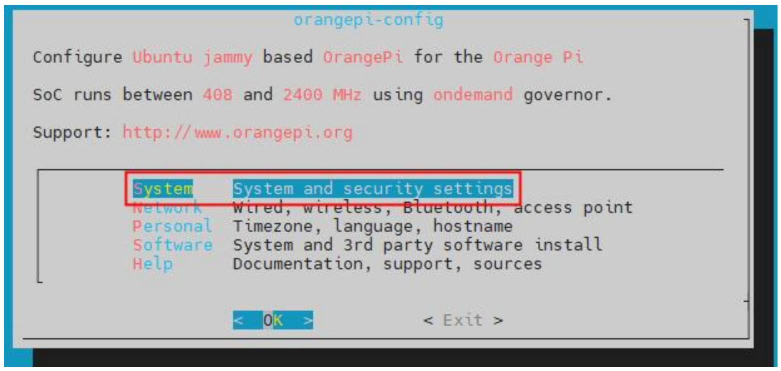
	SPI1_M2 corresponds to 26pin	**SPI0_M2 corresponds to 26pin**
MOSI	**Pin 10**	**Pin 19**
MISO	**Pin 8**	**Pin 21**
CLK	**Pin 11**	**Pin 23**
CS0	**Pin 13**	**Pin 24**
CS1	**NONE**	**Pin 26**
dtbo configuration	**spi1-m2-cs0-spidev**	**spi0-m2-cs0-spidev** **spi0-m2-cs1-spidev** **spi0-m2-cs0-cs1-spidev**

3) In linux system, SPI in 26 Pin are closed by default, you need to open them manually to use them. Detailed steps are shown below:

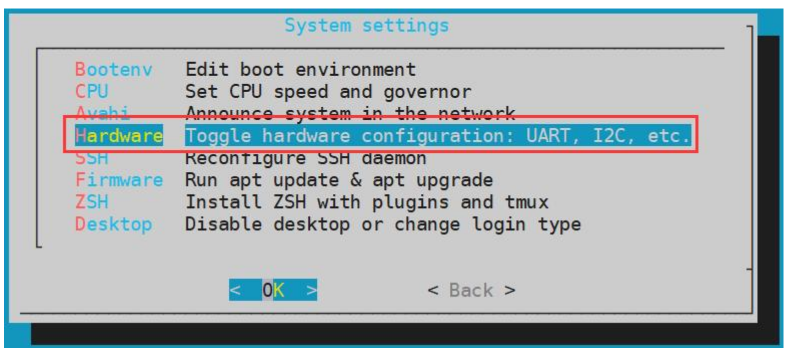
a) First run **orangepi-config**, normal users remember to add **sudo** privileges.

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

b) Then select **System**.

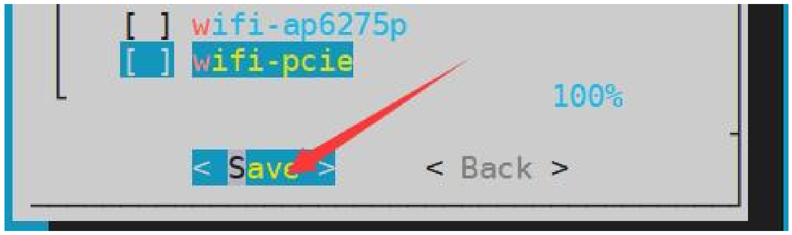


c) Then select **Hardware**.

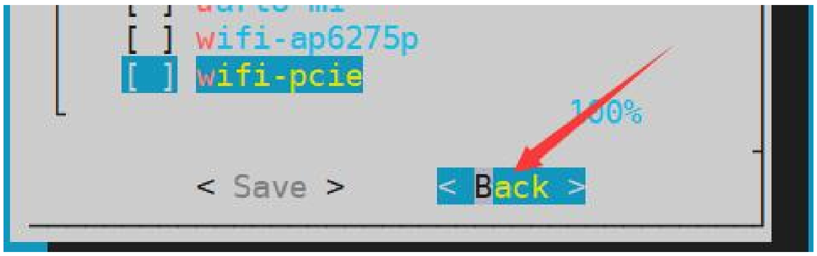


d) Then use the arrow keys of your keyboard to locate the position shown in the figure below, and then use space to select the configuration of the SPI you want to open.

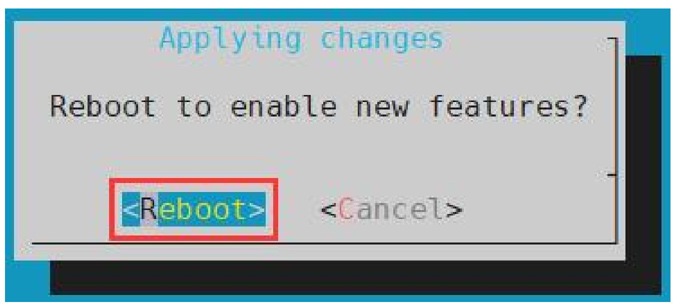
e) Then select **<Save>** to save.



f) Then select **<Back>**.



g) Then select **<Reboot>** to reboot the system to make the configuration take effect.



4) After reboot, enter the system to check whether the device node of **spidevx.x** exists in the linux system first, if it exists, it means that the SPI has been set up and can be used directly.

```
orangepi@orangepi:~$ ls /dev/spidev*
/dev/spidev0.0    /dev/spidev0.1    /dev/spidev1.0
```

Above is the result shown after opening spi1-m2-cs0-spidev and spi0-m2-cs0-cs1-spidev.

5) First do not short SPI0 or SPI1 mosi and miso two pins, run spidev_test output results are shown below, you can see that the TX and RX data is not consistent.

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev0.0
OR
```

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev0.1
OR
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF  | ................................
```

6) Then short SPI0 or SPI1 mosi and miso two pins and then run spidev_test output is as follows, you can see the same data sent and received.

```
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev0.0
OR
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev0.1
OR
orangepi@orangepi:~$ sudo spidev_test -v -D /dev/spidev1.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
```

3. 19. 6. Testing PWM with /sys/class/pwm

1) From the table below, the available pwm for Orange Pi CM5 Base Tablet are pwm0, pwm1, pwm3, and pwm13.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

2) The corresponding pins of PWM in 26pin are shown in the table below:

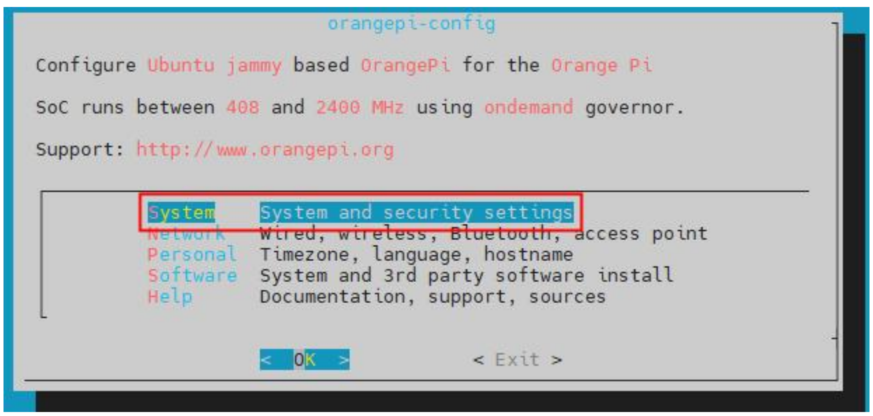
PWM bus	Corresponding to 26pin	Dtbo corresponding configuration
PWM0_M1	**11pin**	**pwm0-m1**
PWM1_M2	**7pin**	**pwm1-m2**
PWM3_M3	**22pin**	**pwm3-m3**
PWM13_M2	**3pin**	**pwm13-m2**

3) In linux system, the PWM in 26 Pin are all turned off by default, you need to turn them on manually to use them. Detailed steps are shown below:

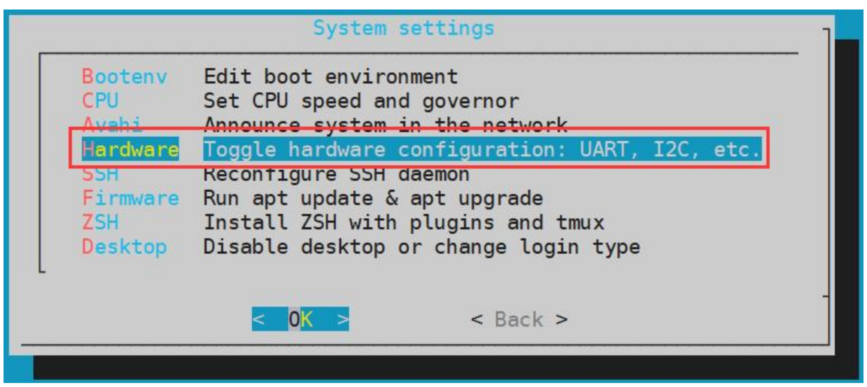
a. First run **orangepi-config**, ordinary users remember to add **sudo** privileges.

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

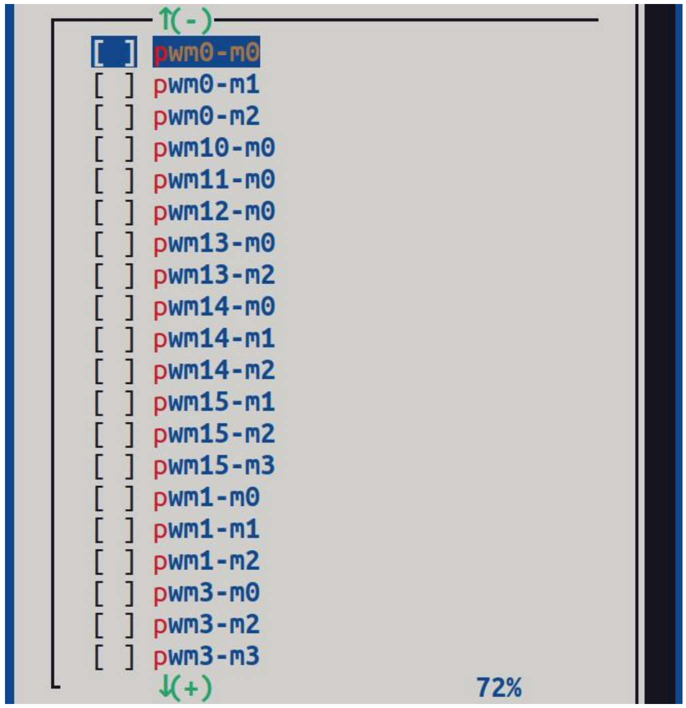
b. Then select **System**.

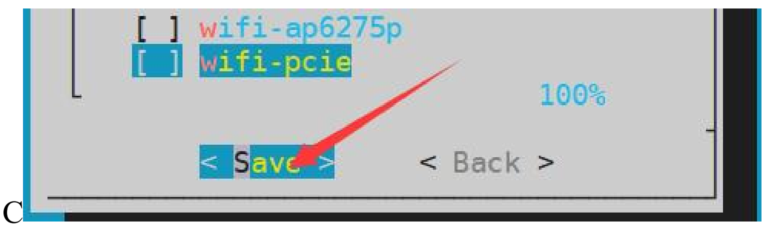


c. Then select **Hardware**.



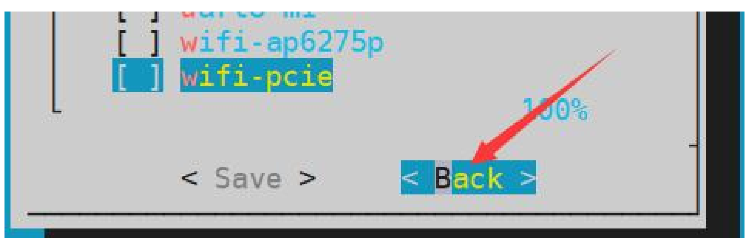
d. Then use the arrow keys of the keyboard to locate the position shown in the figure below, and then use the space to select the configuration of the PWM you want to open.



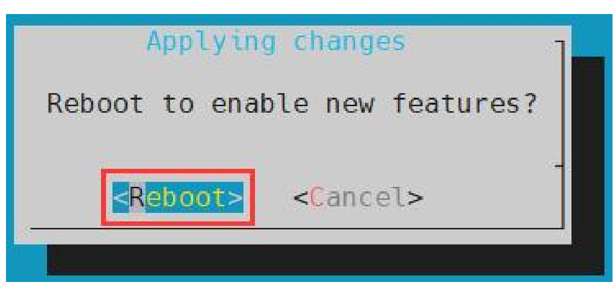


C

e. Then select **<Back>**.



f. Then select **<Reboot>** to reboot the system to make the configuration take effect.



4) When open a pwm, in **/sys/class/pwm/** there will be an additional pwmchipX (X is a

specific number), for example, after opening pwm13, view /sys/class/pwm/ under the pwmchipX will change from two to three.

```
orangepi@orangepi:~$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

5) Which pwmchip above corresponds to pwm13? Let's check the output of the ls **ls /sys/class/pwm/ -l** command as follows:

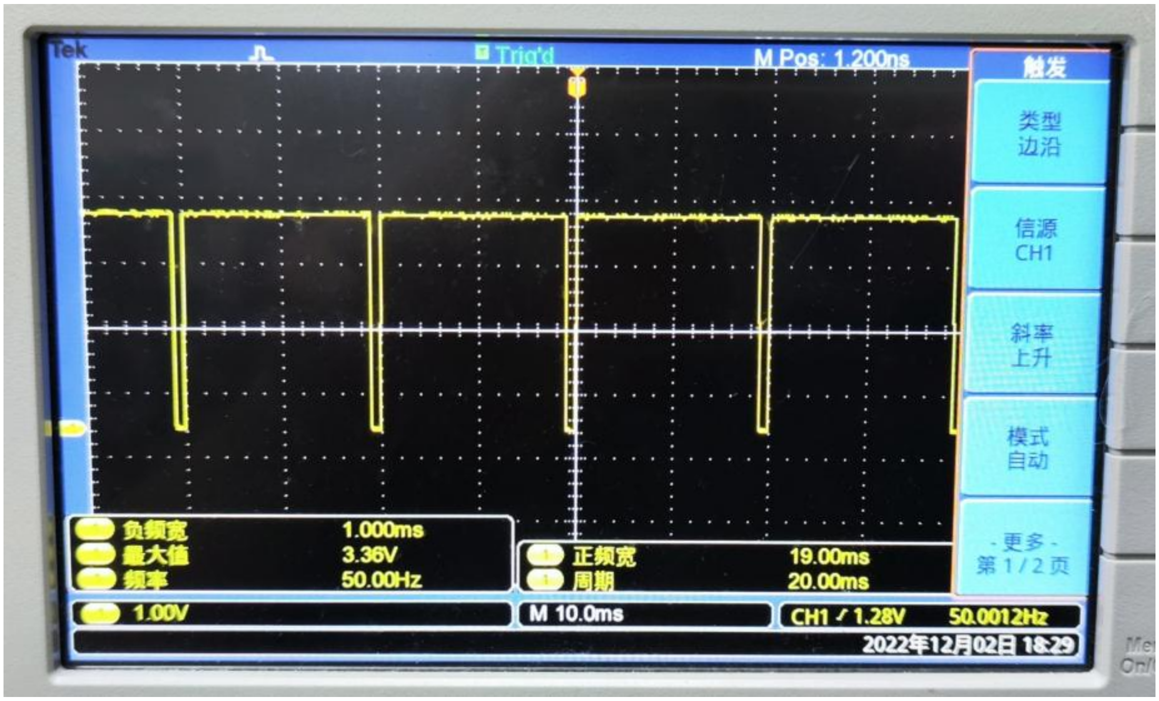
```
orangepi@orangepi:~$ ls /sys/class/pwm/ -lh
total 0
lrwxrwxrwx 1 root root 0 Jan  1  1970 pwmchip0 -> ../../devices/platform/febf0010.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0 Jan  1  1970 pwmchip1 -> ../../devices/platform/febf0020.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0 Jan  1  1970 pwmchip2 -> ../../devices/platform/febf0030.pwm/pwm/pwmchip2
orangepi@orangepi:~$
```

6) Then from the following table, we can see that the base address of the pwm13 register is febf0010, and then look at the output of the ls /sys/class/pwm/ -l command, we can see that febf0010.pwm is linked in pwmchip0, so pwm13 corresponds to pwmchip as pwmchip0.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号
			3.3V		1
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7
			GND		9
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13
			GPIO1_B0	40	15
			3.3V		17
		SPI0_MOSI_M2	GPIO1_B2	42	19
		SPI0_MISO_M2	GPIO1_B1	41	21
		SPI0_CLK_M2	GPIO1_B3	43	23
			GND		25

7) Then you can make pwm13 output a 50Hz square wave by using the following command (please switch to root user first, then execute the following command).

```
root@orangepi:~# echo 0 > /sys/class/pwm/pwmchip0/export
root@orangepi:~# echo 20000000 > /sys/class/pwm/pwmchip0/pwm0/period
root@orangepi:~# echo 1000000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle
root@orangepi:~# echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable
```



8) The test method for pwm13 demonstrated above is similar for all other pwm tests.

3. 20. wiringOP-Python Installation and Usage

wiringOP-Python is a library for the Python language version of wiringOP, which is used to manipulate hardware resources such as GPIOs, I2Cs, SPIs and UARTs of the development board in Python programs.

Also note that all the commands below operate under the root user.

3. 20. 1. wiringOP-Python installation methods

1) First install the dependency packages.

```
root@orangepi:~# sudo apt-get update
root@orangepi:~# sudo apt-get -y install git swig python3-dev python3-setuptools
```

2) Then use the following command to download the wiringOP-Python source code.

Note that the git clone--recursive command below automatically downloads the wiringOP source code, since wiringOP-Python is dependent on wiringOP. Please make sure that the download process is not reporting errors due to network problems.

If you have problems downloading the code from GitHub, you can directly use the wiringOP-Python source code that comes with the Linux image, which is stored at **/usr/src/wiringOP-Python**.

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

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

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

4) Then enter the following command, if there is a help message output, it means wiringOP-Python installation is successful, press q to exit the help message interface.

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

NAME
    wiringpi

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

5) The steps to test whether wiringOP-Python is successfully installed at the python command line are shown below:

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

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

b. Then import the wiringpi python module.

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

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

```
>>> help(wiringpi)
Help on module wiringpi:

NAME
    wiringpi

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

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

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

>>>
```

3. 20. 2. 26pin GPIO port test

Like wiringOP, wiringOP-Python also allows you to determine which GPIO pin to operate by specifying the wPi number. Because there is no command to view the wPi number in wiringOP-Python, you can only use the gpio command in wiringOP to view the correspondence between the board's wPi number and the physical pins.

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |    OUT | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |    OUT | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |    OUT | 1 |  7 || 8  | 1 | OUT    | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | OUT    | RXD.6    | 4   | 57   |
 |   58 |   5 |    RXD.4 |    OUT | 1 | 11 || 12 | 1 | OUT    | GPIO1_A2 | 6   | 34   |
 |   59 |   7 |    TXD.4 |    OUT | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |    OUT | 1 | 15 || 16 | 1 | OUT    | GPIO1_A4 | 9   | 36   |
 |      |     |     3.3V |        |   | 17 || 18 | 1 | OUT    | GPIO1_A6 | 10  | 38   |
 |   42 |  11 | SPI0_TXD |    OUT | 1 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |    OUT | 1 | 21 || 22 | 1 | OUT    | PWM3     | 13  | 39   |
 |   43 |  14 | SPI0_CLK |    OUT | 1 | 23 || 24 | 1 | OUT    | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | OUT    | SPI0_CS1 | 16  | 45   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
orangepi@orangepicm5-tablet:~$
```

1) The following is an example of how to set the high and low levels of the GPIO port on pin 7 -- corresponding to GPIO 1_A3 -- corresponding to wPi serial number 2.

```
orangepi@orangepicm5-tablet:~$ gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   47 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   46 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |     IN | 0 |  7 || 8  | 1 | IN     | TXD.6    | 3   | 56   |
 |      |     |      GND |        |   |  9 || 10 | 1 | IN     | RXD.6    | 4   | 57   |
 |   58 |   5 |    RXD.4 |     IN | 1 | 11 || 12 | 0 | IN     | GPIO1_A2 | 6   | 34   |
 |   59 |   7 |    TXD.4 |  ALT11 | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |     IN | 1 | 15 || 16 | 0 | IN     | GPIO1_A4 | 9   | 36   |
```

2) The procedure for testing directly with commands is shown below:

a. First set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the pin's corresponding wPi, and the second parameter is the mode of the GPIO.

root@orangepi:~/wiringOP-Python# **python3 -c "import wiringpi; **
**from wiringpi import GPIO; wiringpi.wiringPiSetup() ; **
wiringpi.pinMode(2, GPIO.OUTPUT) ; "

b. Then set the GPIO port output low level, after the setup you can use a multimeter to measure the value of the pin's voltage, if it is 0v, it means that the setup low level is successful.

root@orangepi:~/wiringOP-Python# **python3 -c "import wiringpi; **

```
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.LOW)"
```

c. Then set the GPIO port output high level, after the setup you can use a multimeter to measure the value of the voltage of the pin, if it is 3.3v, it means that the setup of the high level is successful.

```
root@orangepi:~/wiringOP-Python# python3 -c "import wiringpi; \
 from wiringpi import GPIO; wiringpi.wiringPiSetup() ;\
wiringpi.digitalWrite(2, GPIO.HIGH)"
```

3) The steps to test in python3 command line are shown below:

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

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

b. Then import the wiringpi python module.

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

c. Then set the GPIO port to output mode, where the first parameter of the **pinMode** function is the serial number of the pin's corresponding wPi, and the second parameter is the mode of the GPIO.

```
>>> wiringpi.wiringPiSetup()
0
>>> wiringpi.pinMode(2, GPIO.OUTPUT)
```

d. Then set the GPIO port output low level, after the setup you can use a multimeter to measure the value of the pin's voltage, if it is 0v, it means that the setup low level is successful.

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

e. Then set the GPIO port output high level, after the setup you can use a multimeter to measure the value of the pin's voltage, if it is 3.3v, it means that the setup high level is successful.

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

4) wiringOP-Python in python code to set the GPIO high and low levels of the method can refer to the **blink.py** test program in the examples, **blink.py** test program will set the development board 26 pin in all the GPIO port voltage constantly high and low changes.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# ls blink.py
```

```
blink.py
root@orangepi:~/wiringOP-Python/examples# python3 blink.py
```

3. 20. 3. 26pin I2C test

1) As shown in the table below, the available i2c for the Orange Pi CM5 Base Tablet are two sets of i2c buses, i2c4 and i2c5.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

2) The corresponding pins of the two I2C buses in 26pin are shown in the table below:

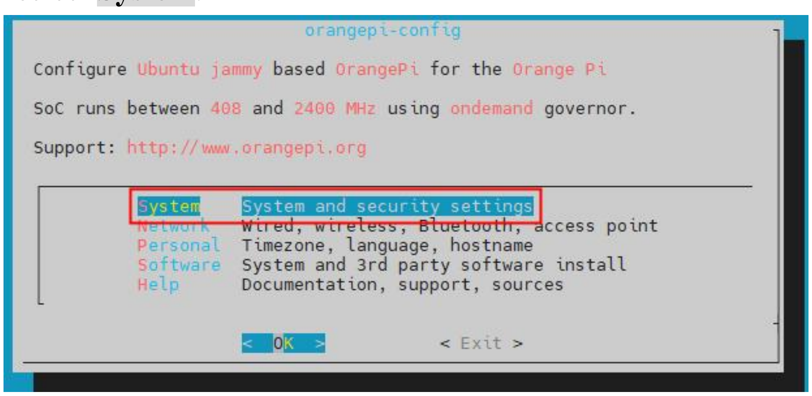
I2C bus	SDA corresponds to 26pin	SCL corresponds to 26pin	The dtbo counterpart configuration
I2C4_M3	Pin 12	Pin 7	i2c4-m3
I2C5_M3	Pin 3	Pin 5	i2c5-m3

3) In linux system, the I2C bus in 26 Pin are closed by default, you need to open it manually to use it. Detailed steps are shown below:

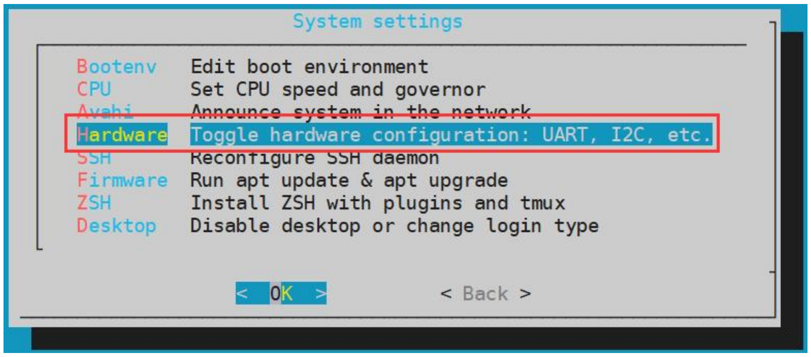
a. First run **orangepi-config**, ordinary users remember to add **sudo** privileges.

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

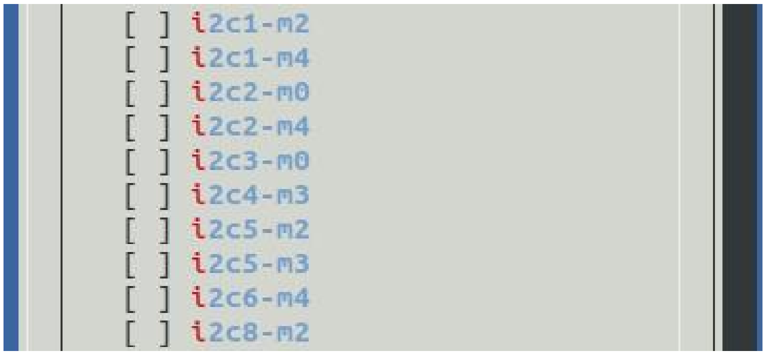
b. Then select **System**.



c. Then select **Hardware**.

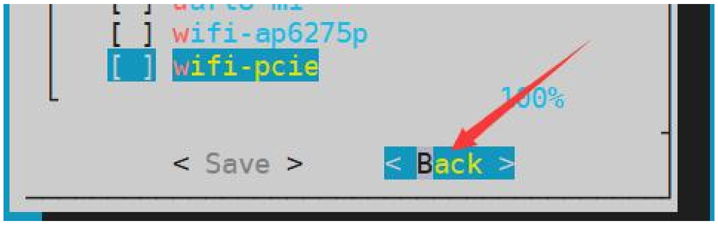


d. Then use the arrow keys of the keyboard to locate the position shown in the figure below, and then use the space to select the configuration of the I2C you want to open.

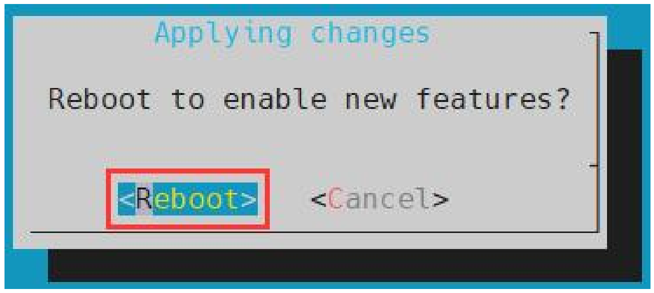


e. Then select **<Save>** to save.

f. Then select **<Back>**.



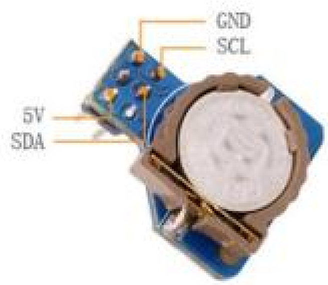
g. Then select **<Reboot>** to reboot the system to make the configuration take effect.



4) After booting the linux system, first confirm the existence of device nodes that need to use I2C under **/dev**.

```
orangepi@orangepi:~$ ls /dev/i2c-*
```

5) Then connect an i2c device on the i2c pin of the 26pin connector, here take the ds1307 RTC module as an example



6) Then use the **i2cdetect -y** command if you can detect the address of the connected i2c device, it means that i2c can be used normally

```
orangepi@orangepi:~$ sudo i2cdetect -y 4        #i2c4 Command
orangepi@orangepi:~$ sudo i2cdetect -y 5        #i2c5 Command
```

7) Then you can run the **ds1307.py** test program in **examples** to read the RTC time

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device "/dev/i2c-4"
Thu 2023-01-05    14:57:55
Thu 2023-01-05    14:57:56
Thu 2023-01-05    14:57:57
^C
exit
```

3. 20. 4. 26pin UART test

1) From the table below, the uart available for Orange Pi CM5 Base Tablet are uart1, uart4, uart6 and uart7, four sets of uart bus.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

2) The corresponding pins of the four UART buses in 26pin are shown in the table below:

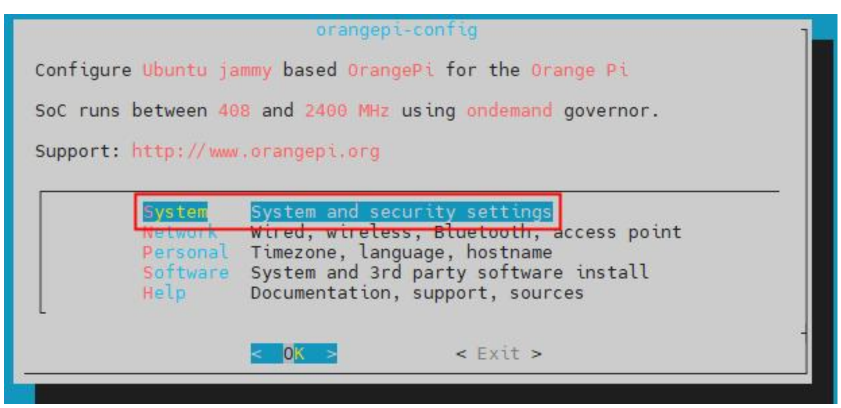
UART BUS	RX corresponds to 26pin	TX corresponds to 26pin	dtbo corresponding configuration
UART1_M1	Pin 3	Pin 5	uart1-m1
UART4_M0	Pin 13	Pin 11	uart4-m0
UART6_M1	Pin 10	Pin 8	uart6-m2
UART7_M2	Pin 24	Pin 26	uart7-m2

3) In Linux system, the UART in 26 Pin is closed by default and needs to be opened manually before it can be used. The detailed steps are as follows:

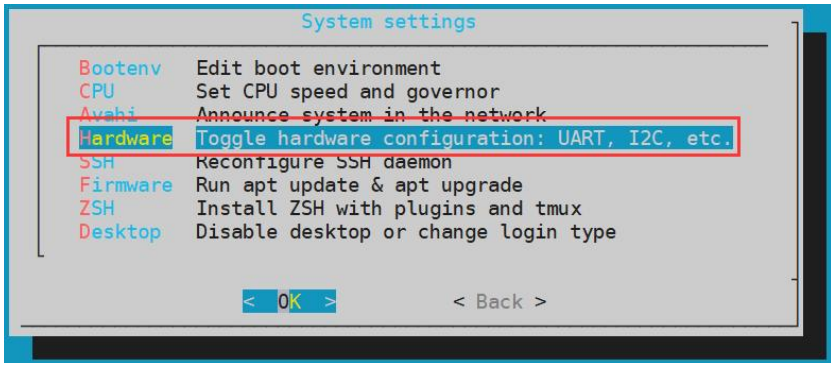
a. First run **orangepi-config**. Remember to add **sudo** permissions as a normal user.

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

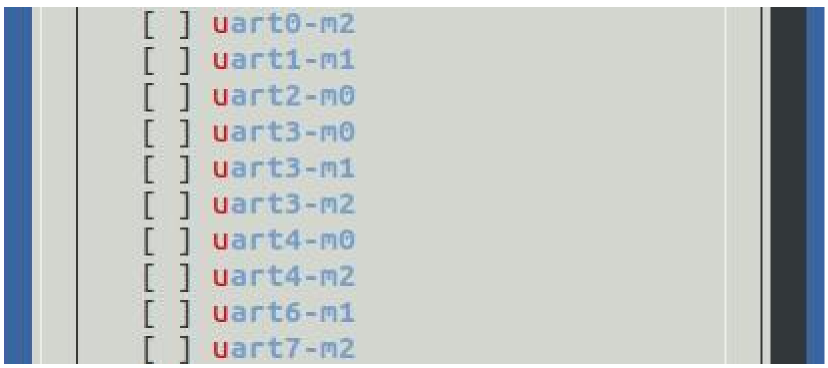
b. Then select **System**.



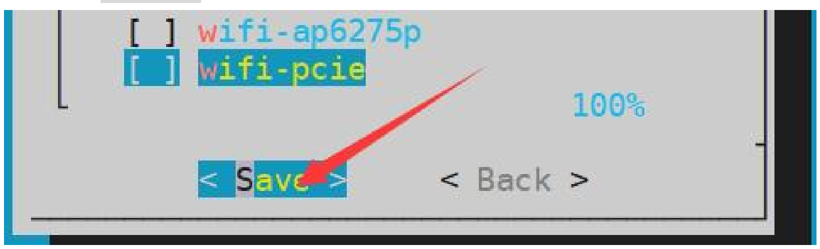
c. Then select **Hardware**.



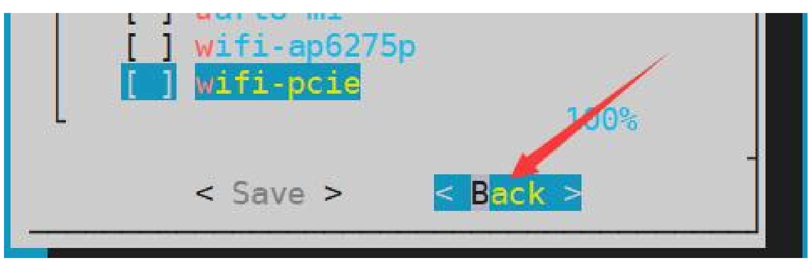
d. Then use the arrow keys on your keyboard to locate the position shown in the figure below, and then use the spacebar to select the UART configuration you want to open.



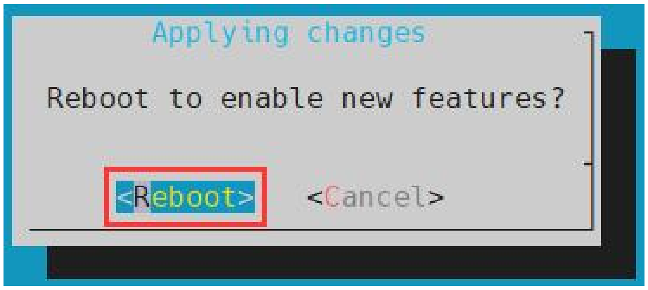
e. Then select **<Save>** to save.



f. Then select **<Back>**.



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



4) After entering the Linux system, first confirm whether there is a device node corresponding to uart under /dev.

```
orangepi@orangepi:~$ ls /dev/ttyS*
```

5) Then start testing the UART interface. First use the Dupont line to short-circuit the rx and tx of the UART interface to be tested.

6) Use the **serialTest.py** program in examples to test the loopback function of the serial port as shown below. If you can see the following print, it means that the serial port communication is normal.

/dev/ttySX needs to be replaced with the specific uart device node number.

```
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device "/dev/ttySX"

Out:    0: ->    0
Out:    1: ->    1
Out:    2: ->    2
Out:    3: ->    3
Out:    4:^C
exit
```

3. 20. 5. 26pin SPI test

1) As shown in the figure below, the available spis for Orange Pi CM5 Base Tablet are spi1 and spi4.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

2) The corresponding pins of SPI0 and SPI4 in 26 pins are shown in the following table.

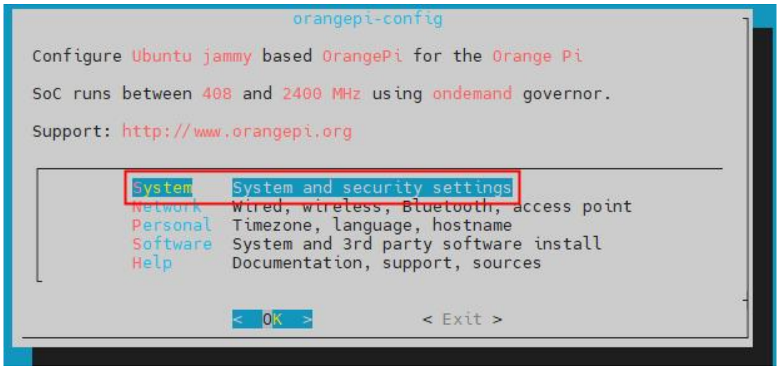
	SPI1_M2 corresponds to 26pin	SPI4_M2 corresponds to 26pin
MOSI	PIN 10	PIN 19
MISO	PIN 8	PIN 21
CLK	PIN 11	PIN 23
CS0	PIN 13	PIN 24
CS1	无	PIN 26
dtbo configuration	spi1-m2-cs0-spidev	spi4-m2-cs0-spidev spi4-m2-cs1-spidev spi4-m2-cs0-cs1-spidev

3) In Linux system, the SPI in 26 Pin is closed by default and needs to be opened manually before it can be used. The detailed steps are as follows:

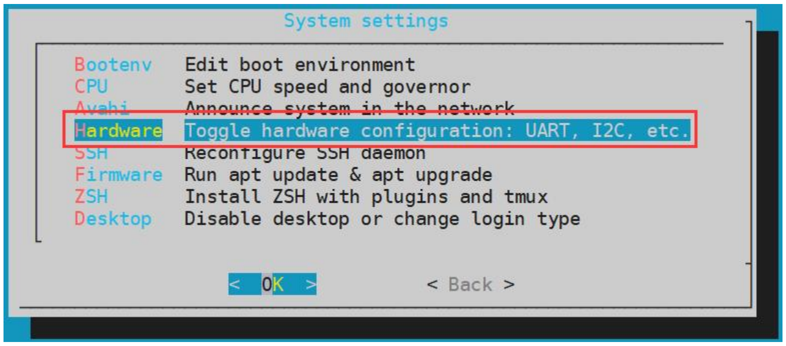
a) First run **orangepi-config**. Remember to add **sudo** privileges as a normal user.

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

b) Then select **System**.



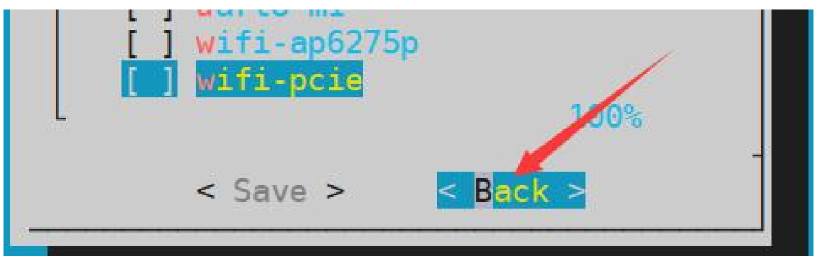
c) Then select **Hardware**.



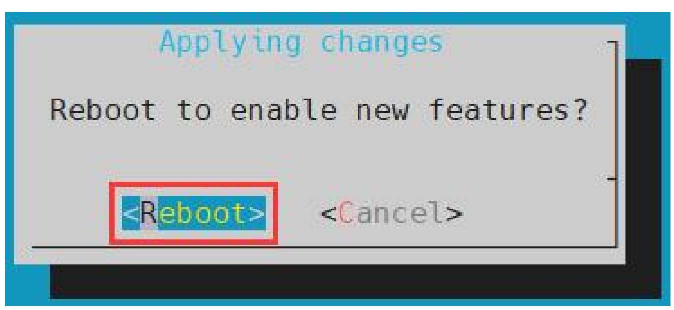
d) Then use the arrow keys on your keyboard to locate the position shown in the figure below, and then use the spacebar to select the SPI configuration you want to open.

e) Then select **<Save>** to save.

f) Then select **<Back>**.



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



4) After restarting, enter the system and check whether there is a device node of **spidevx.x** in the Linux system. If it exists, it means that SPI has been set up and can be used directly.

```
orangepi@orangepi:~$ ls /dev/spidev*
```

```
/dev/spidev1.0    /dev/spidev4.0    /dev/spidev4.1
```

The above is the result after opening spi1-m2-cs0-spidev and spi4-m2-cs0-cs1-spidev.

5) Then you can use the **spidev_test.py** program in the examples to test the SPI loopback function. The **spidev_test.py** program needs to specify the following two parameters:

a. **--channel**: Specify the SPI channel number

b. **--port**: Specify the SPI port number.

6) Do not short the mosi and miso pins of SPI. The output of running spidev_test.py is as shown below. You can see that the data of TX and RX are inconsistent.

The x after the --channel and --port parameters needs to be replaced with the specific SPI channel number and SPI port number.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py --channel x --port x
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev0.0
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF F0 0D    |......@..........|
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF    |................|
```

7) Then use Dupont wire to short the SPI TXD (pin 19 in the 26-pin interface) and RXD (pin 21 in the 26-pin interface) pins and run spidev_test.py. The output is as follows. You can see that the data sent and received are the same, indicating that the SPI loopback test is normal.

The x after the --channel and --port parameters needs to be replaced with the specific SPI channel number and SPI port number.

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py --channel x --port x
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev0.0
```

```
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF F0 0D    |......@..........|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF F0 0D    |......@..........|
```

3. 21. 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 is fed successfully.

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

3. 22. 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. 23. 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:~$ sudo 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. 24. How to download and install the arm64 version of balenaEtcher

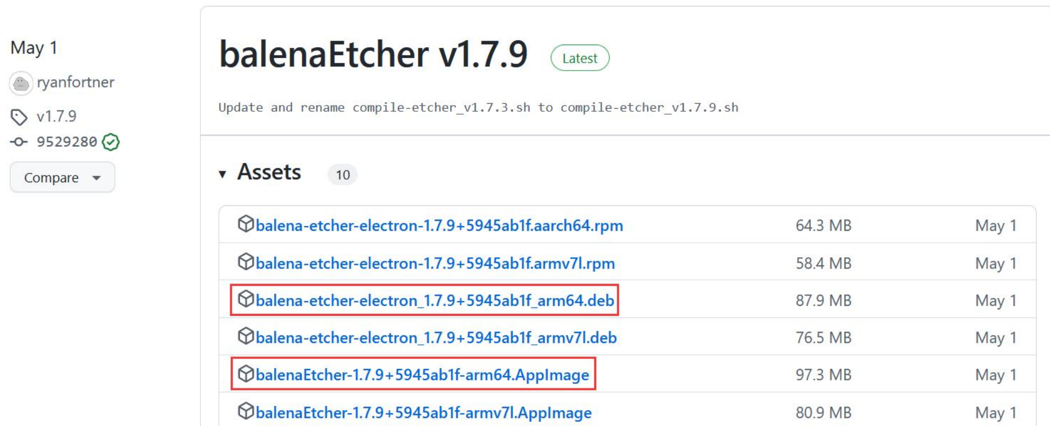
1) The download address of balenaEtcher arm64 version is:

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

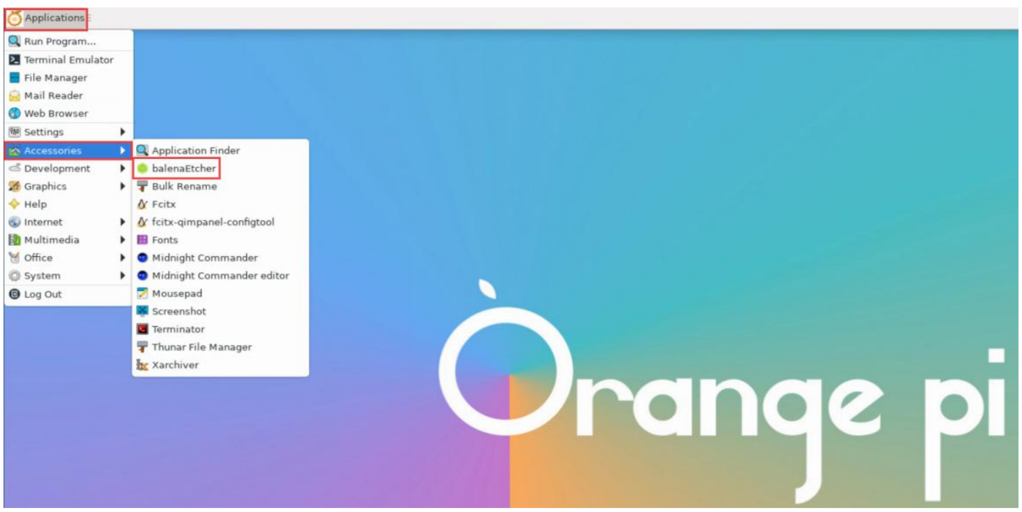


2) How to install and use the deb version of balenaEtcher:

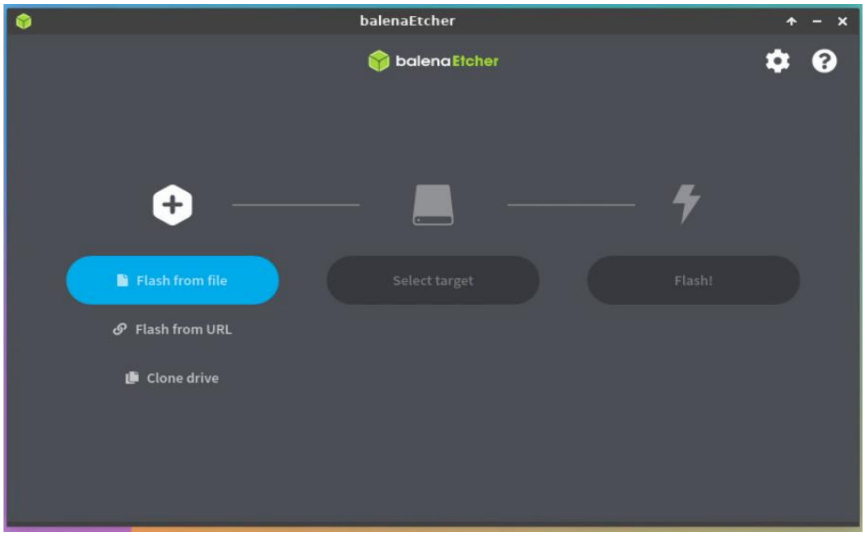
a. deb version of balenaEtcher installation command is as follows:

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:

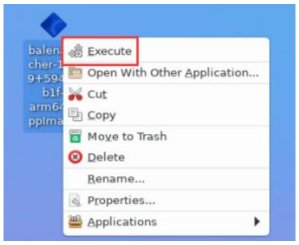


3) How to use the AppImage version of balenaEtcher:

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. 25. 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 https://www.bt.cn/new/index.html)

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

Debian11 > Ubuntu 22.04 > Debian12

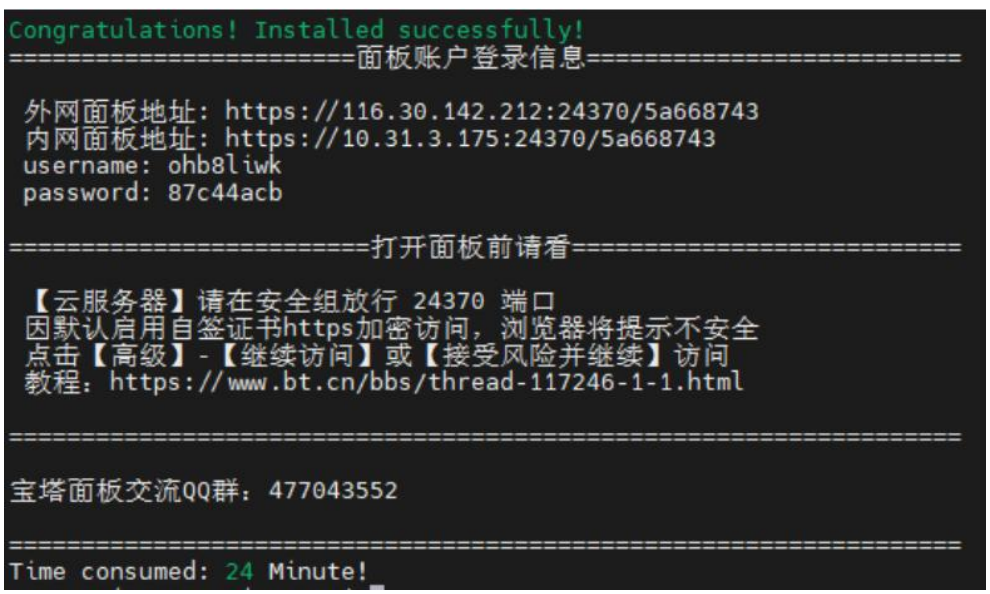
2) Then enter the following command in the Linux system to start the installation of the pagoda.

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

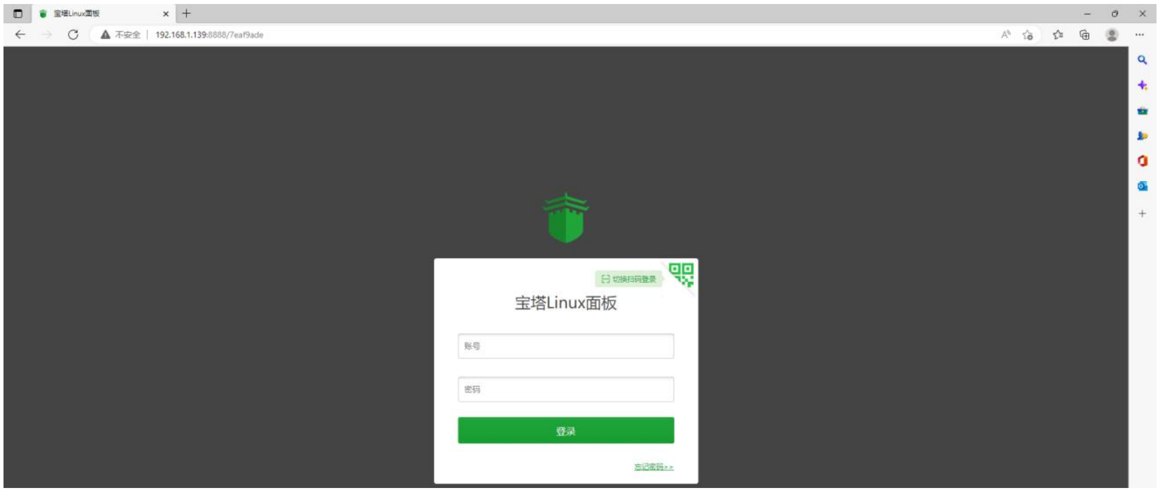
3) Then the Baota installation program will prompt whether to install **Bt-Panel** to the **/www** folder, just enter y.

```
+----------------------------------------------------------------------
| 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.

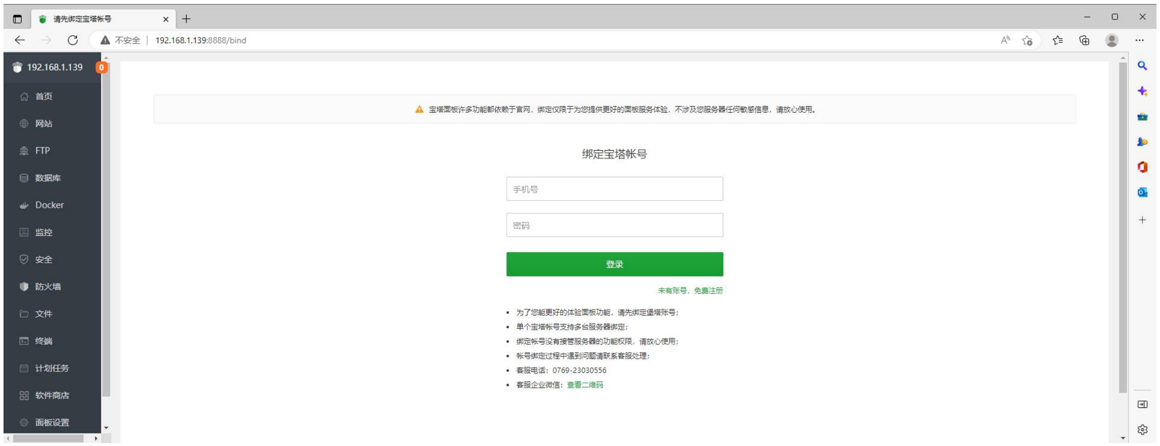


5) At this time, enter the panel address shown above in the browser to open the login interface of the Baota Linux panel, and then enter the username and password shown in the above picture in the corresponding positions to log in to Baota.

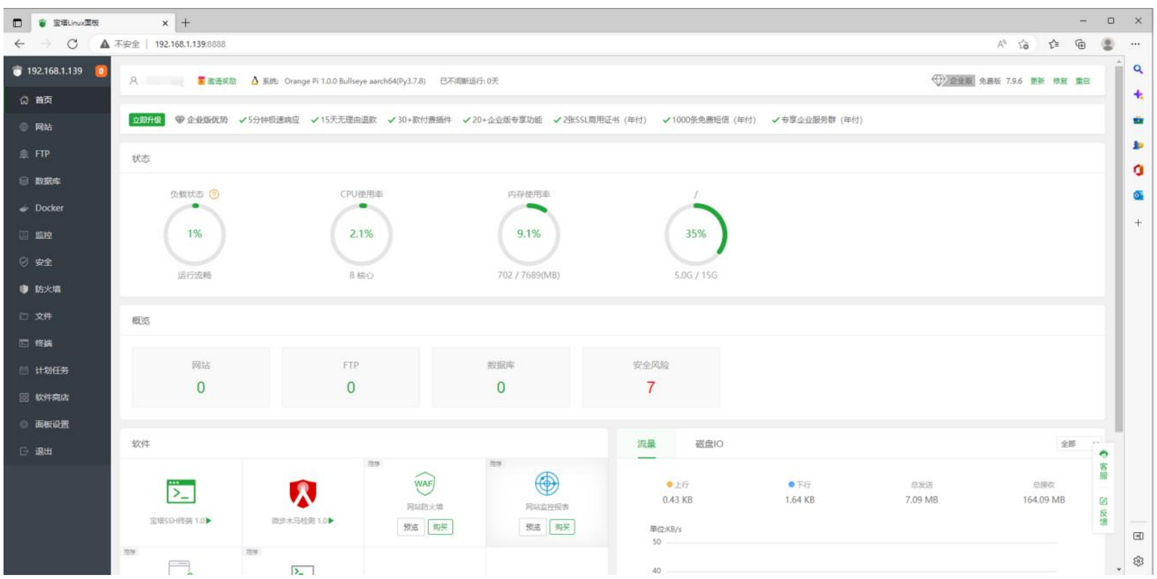


6) After successfully logging into the pagoda, the welcome interface below will pop up. First, please read the user instructions in the middle and drag them to the bottom. Then you can select "I have agreed and read the User Agreement", and then click "Enter Panel" to enter the pagoda.

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.

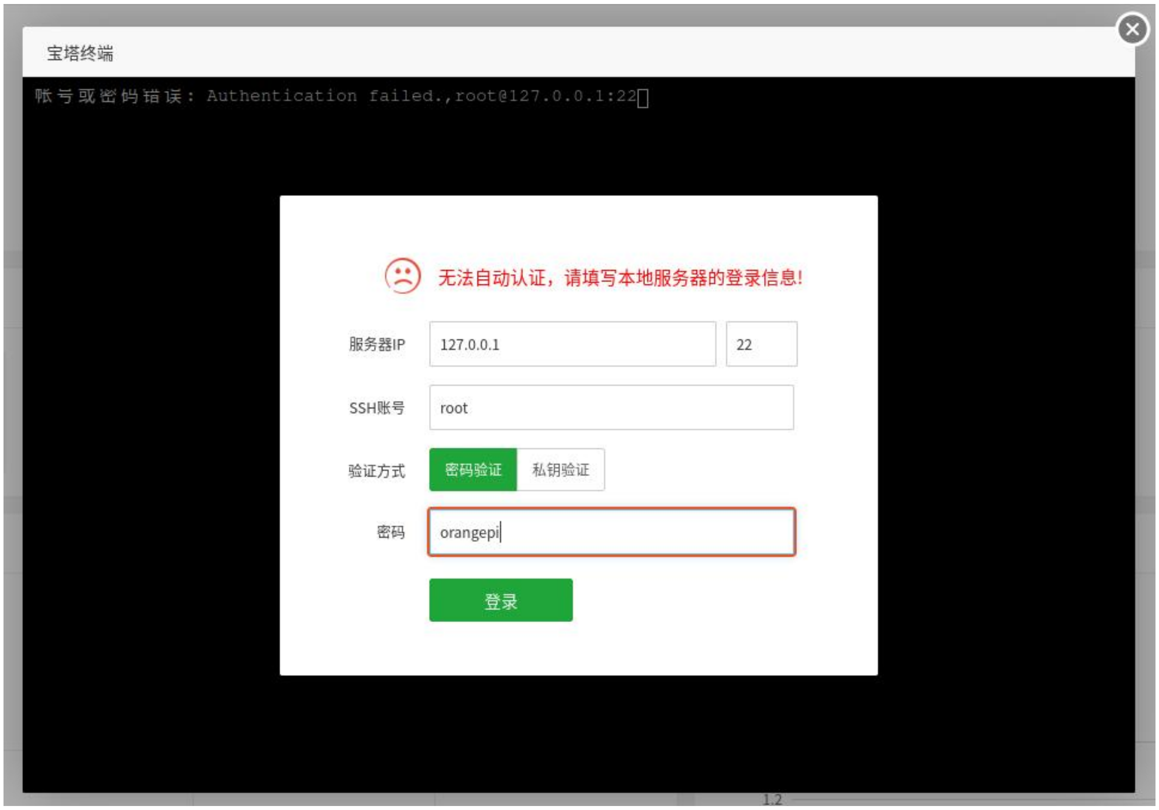


8) The final interface is shown in the figure below. You can intuitively see some status information of the development board Linux system, such as load status, CPU usage, memory usage, and storage space usage.

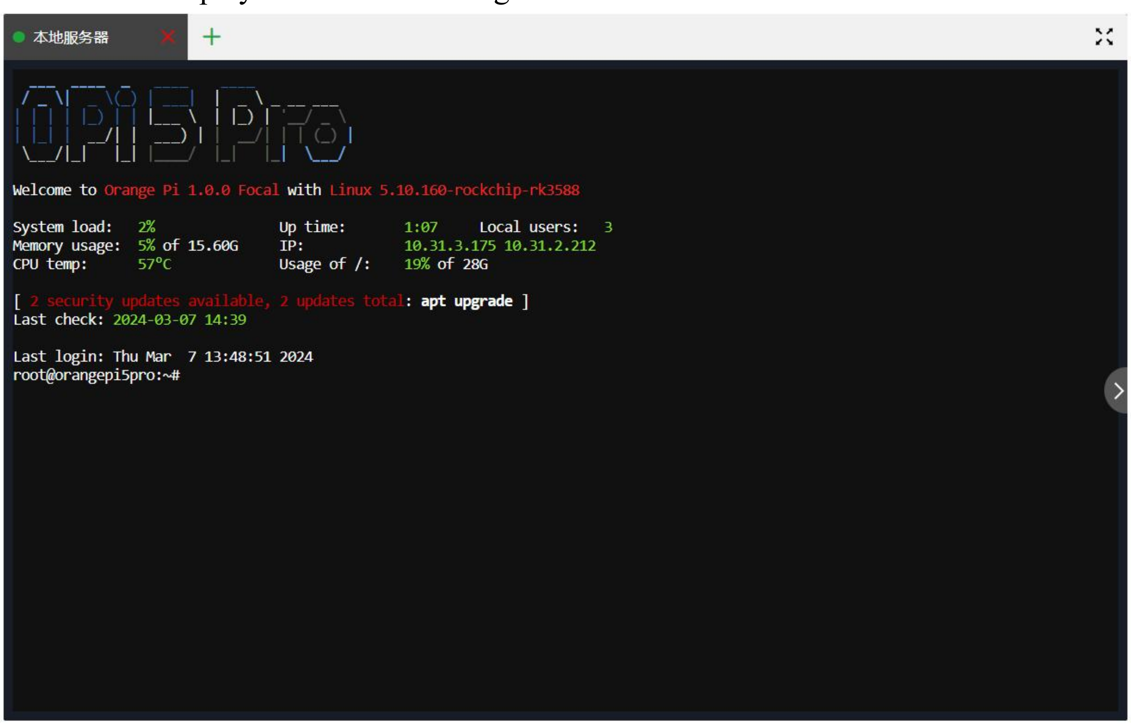


9) Test the SSH terminal login of Baota.

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).

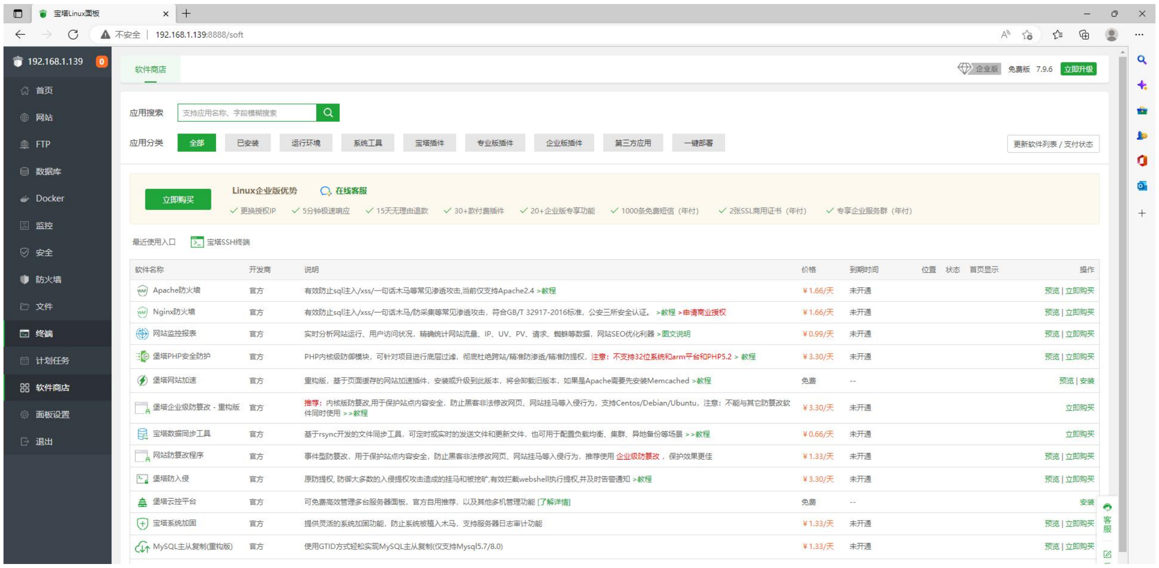


b. The display after successful login is as shown below:



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 on your own.

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

3. 26. 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.

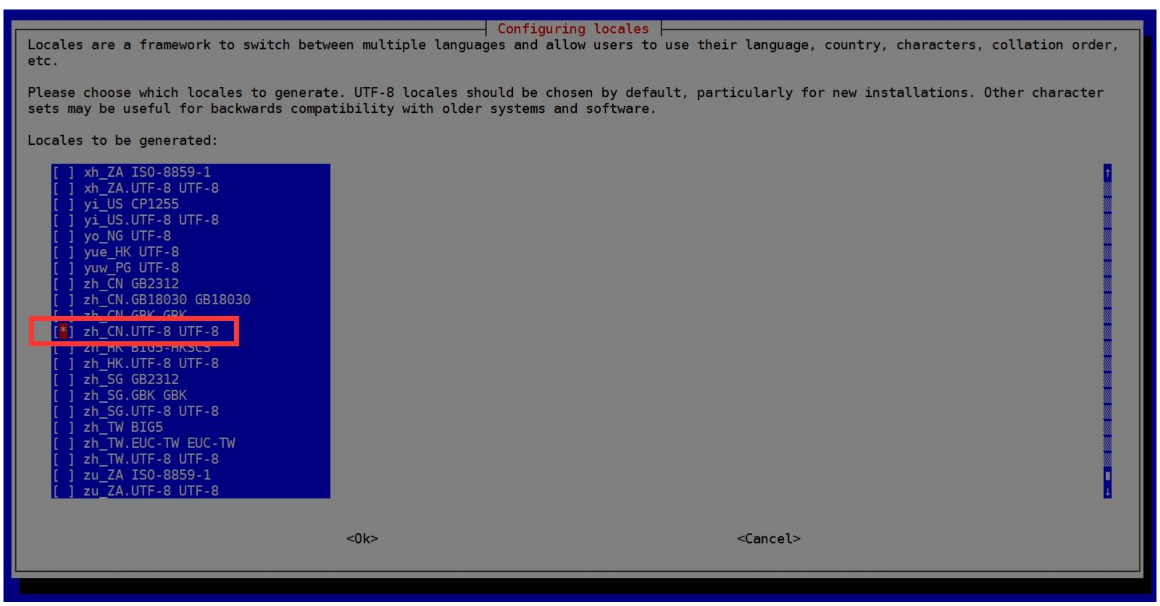
3. 26. 1. Debian system installation method

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

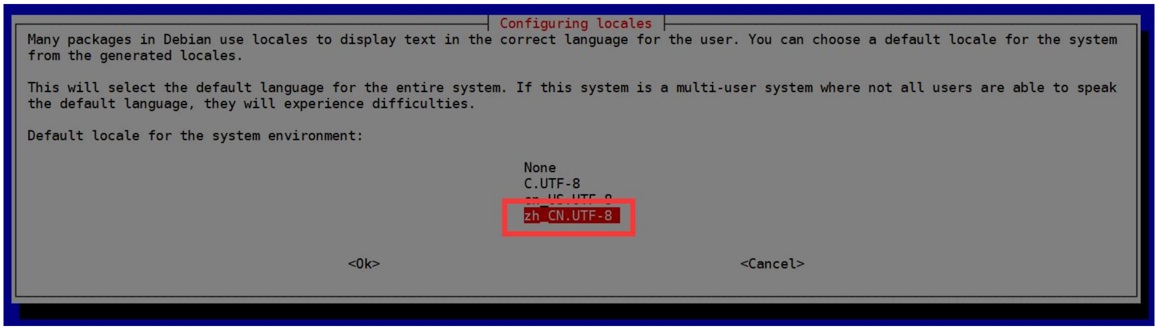
a. Enter the following command to start configuring **locale**.

```
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).



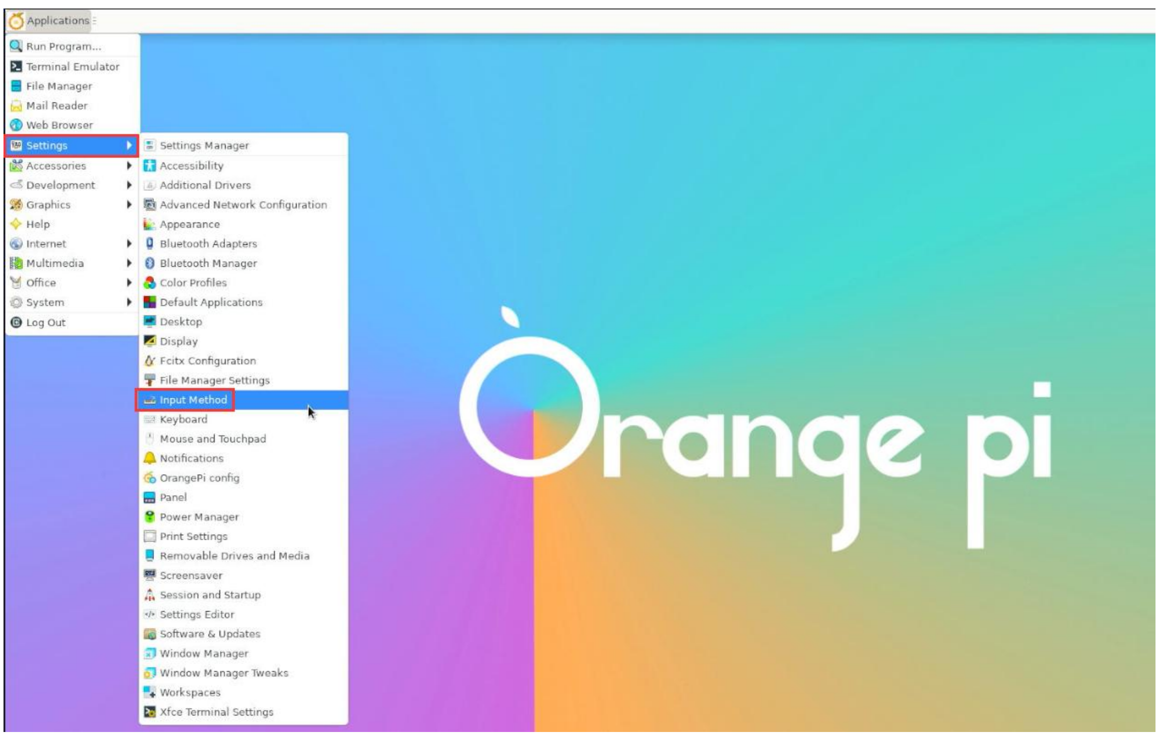
c. Then set the default **locale** to **zh_CN.UTF-8**.



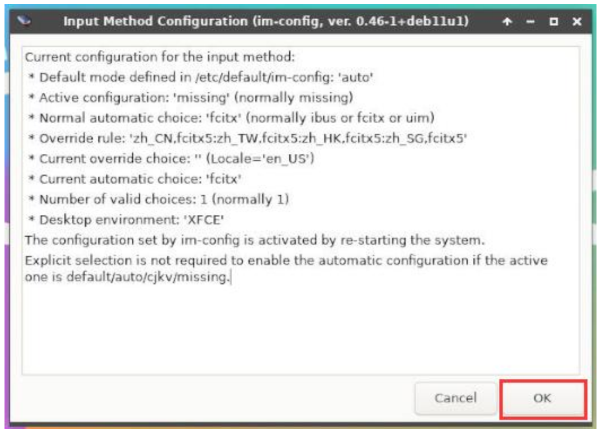
d. After exiting the interface, the **locale** setting will begin. The output displayed on the command line is as follows:

```
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.
```

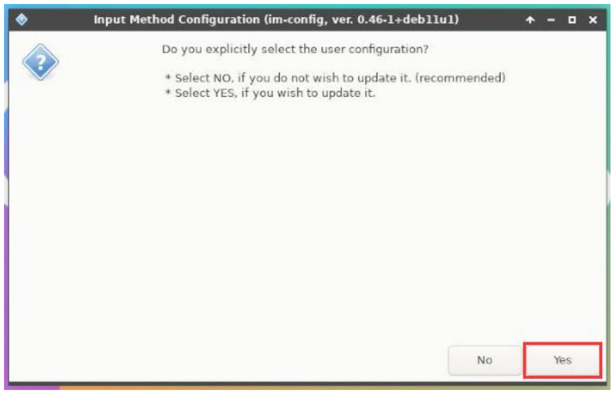
2) Then open **Input Method**.



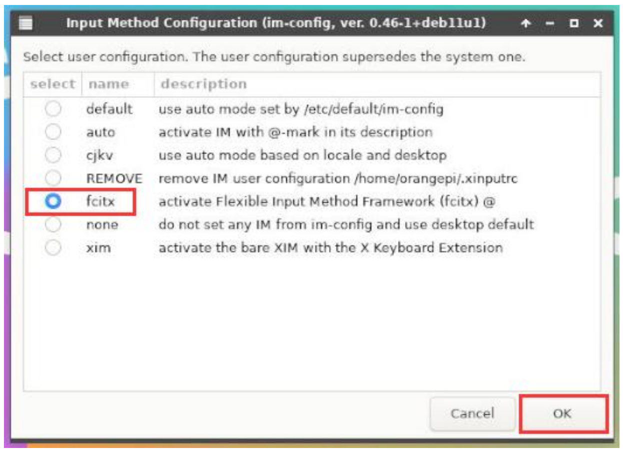
3) Then select **OK**.



4) Then select **Yes**.



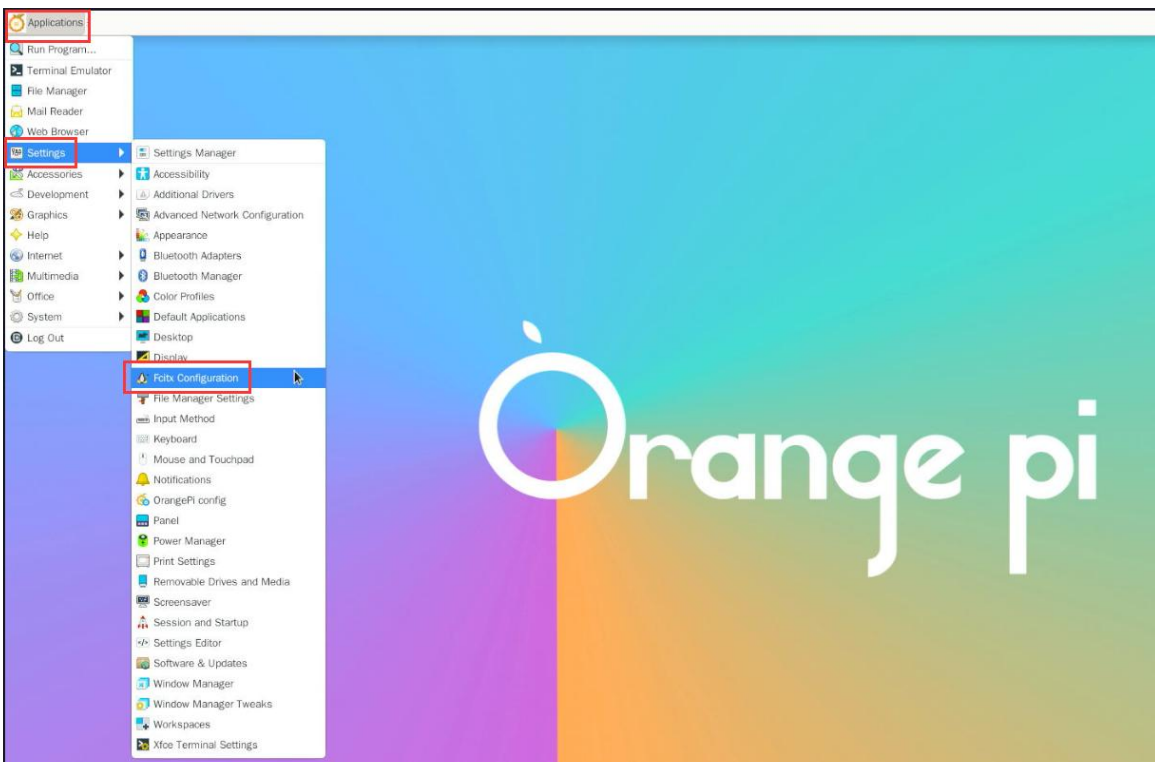
5) Then select **fcitx**.



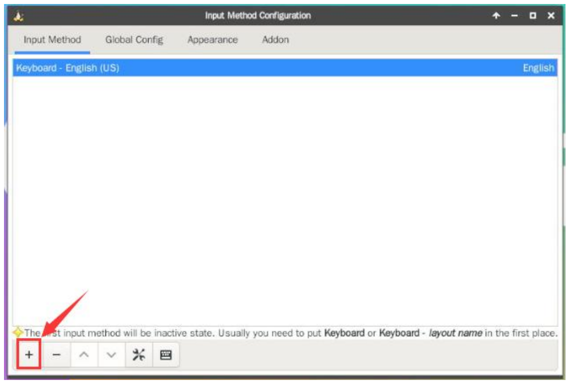
6) Then select **OK**.

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

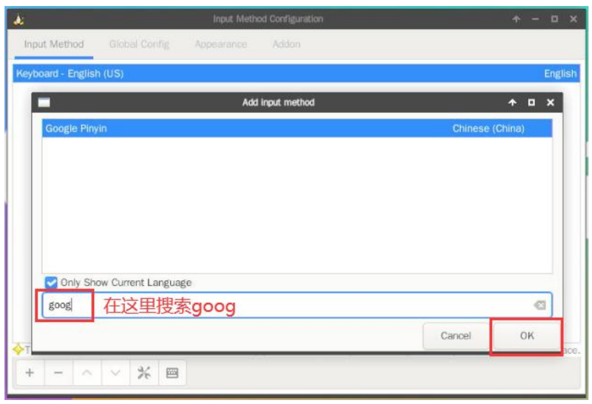
8) Then open **Fcitx configuration**.



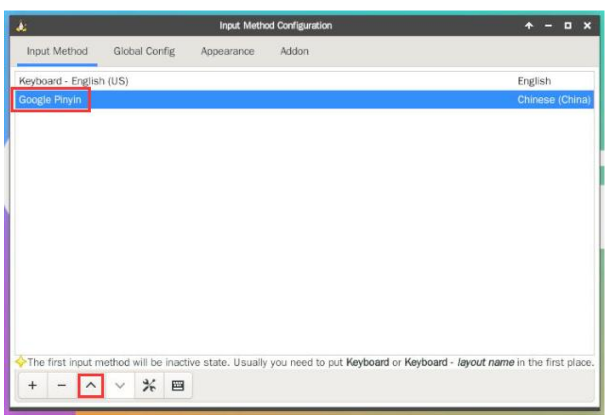
9) Then click the + sign in the location shown in the picture below.

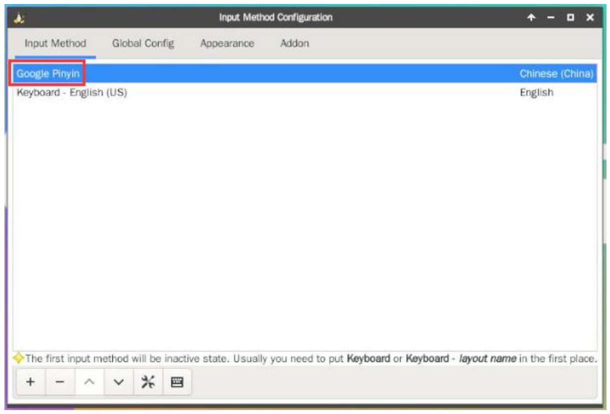


10) Then search **Google Pinyin** and click **OK**.



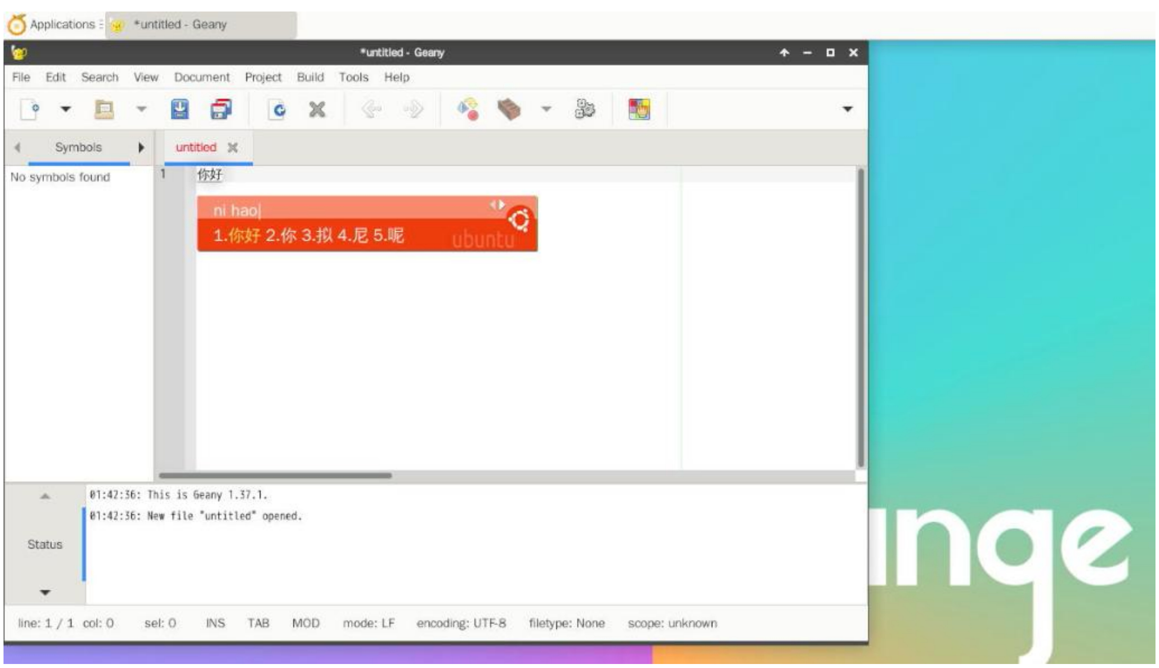
11) Then put **Google Pinyin** at the front.





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

13) The Chinese input method test is as follows:

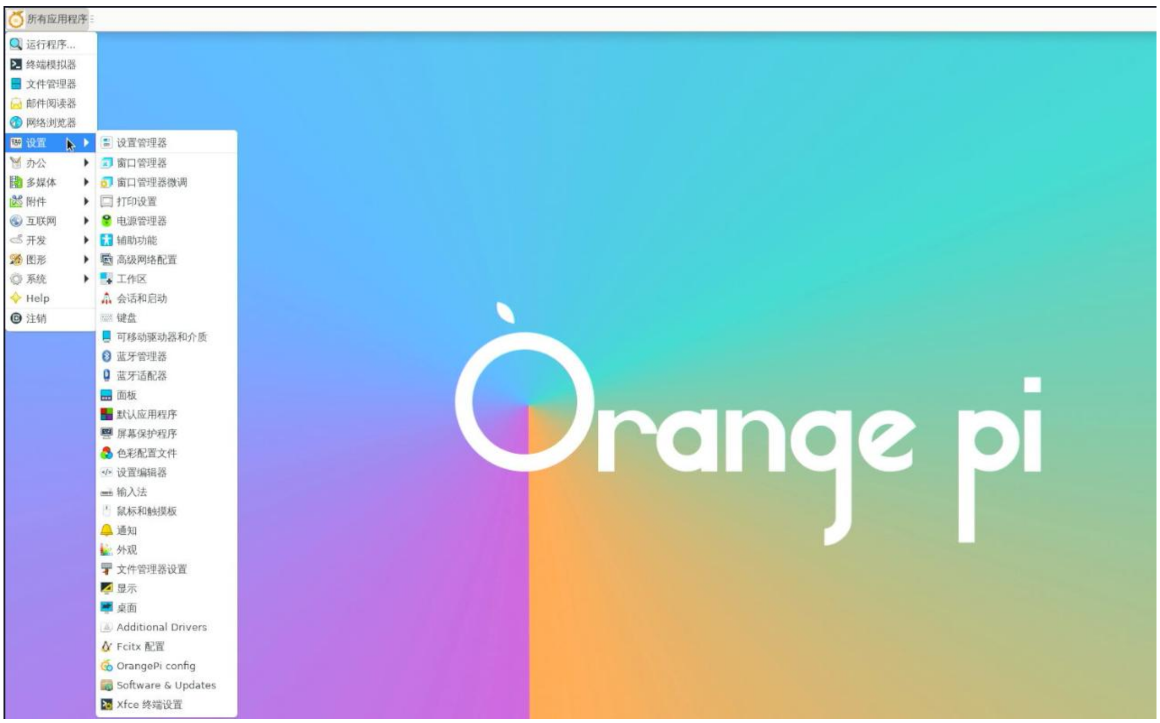


14) Use the **Ctrl+Space** shortcut key to switch between Chinese and English input methods.

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**.

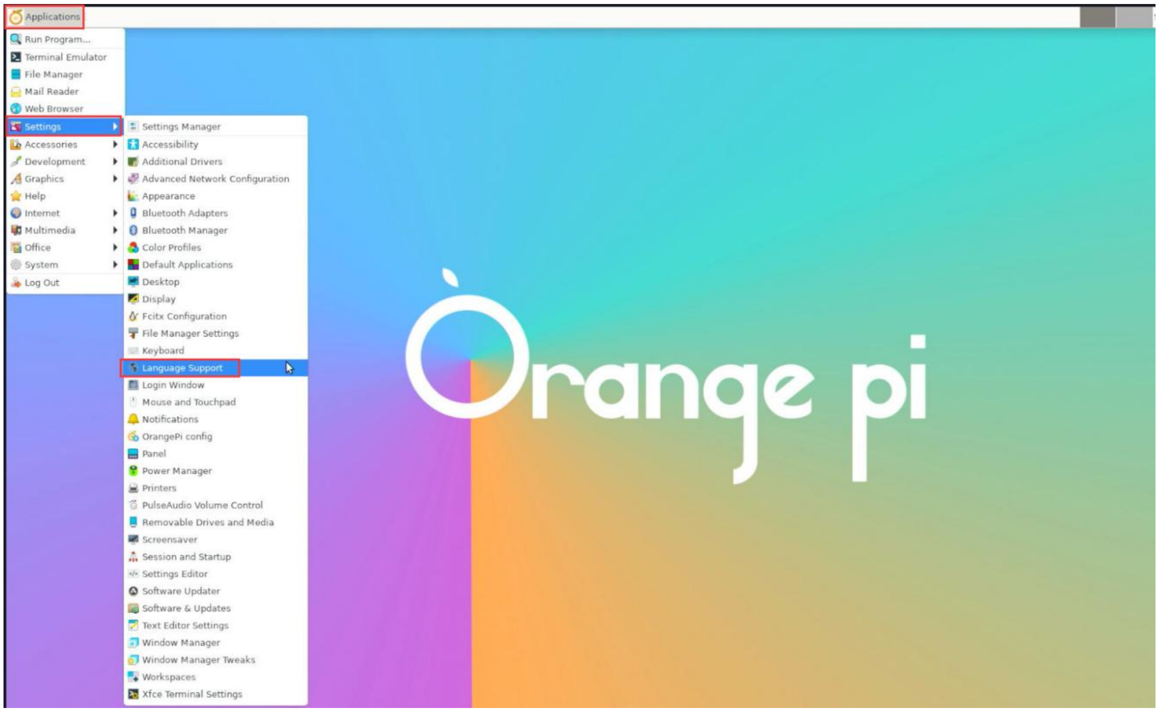
```
orangepi@orangepi:~$ sudo vim /etc/default/locale
#  File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
LANGUAGE=zh_CN.UTF-8
```

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

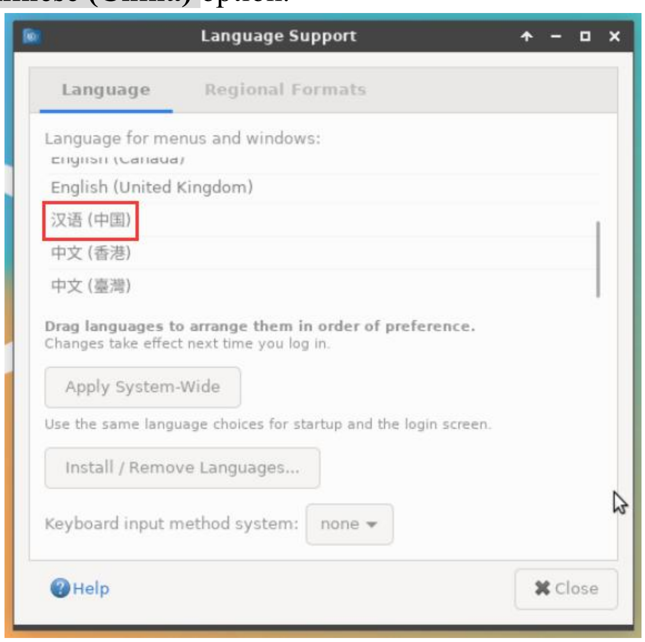


3. 26. 2. How to install Ubuntu 20.04 system

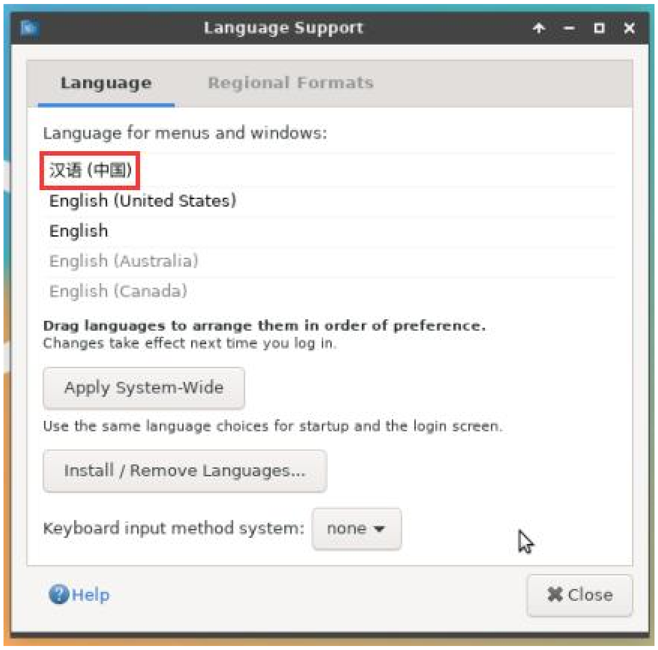
1) First open **Language Support**。



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

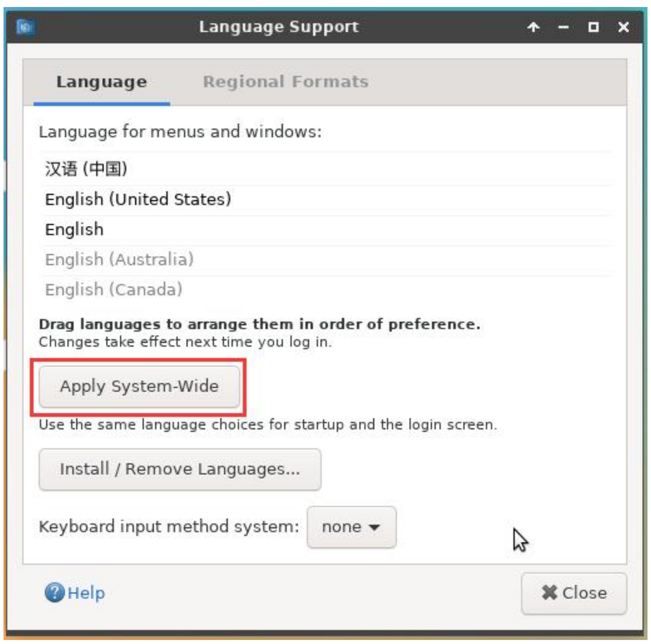


3) Then use the left mouse button to select **Chinese (China)** and hold it down, then drag it upwards to the starting position. The display after dragging is as shown in the figure below:



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.

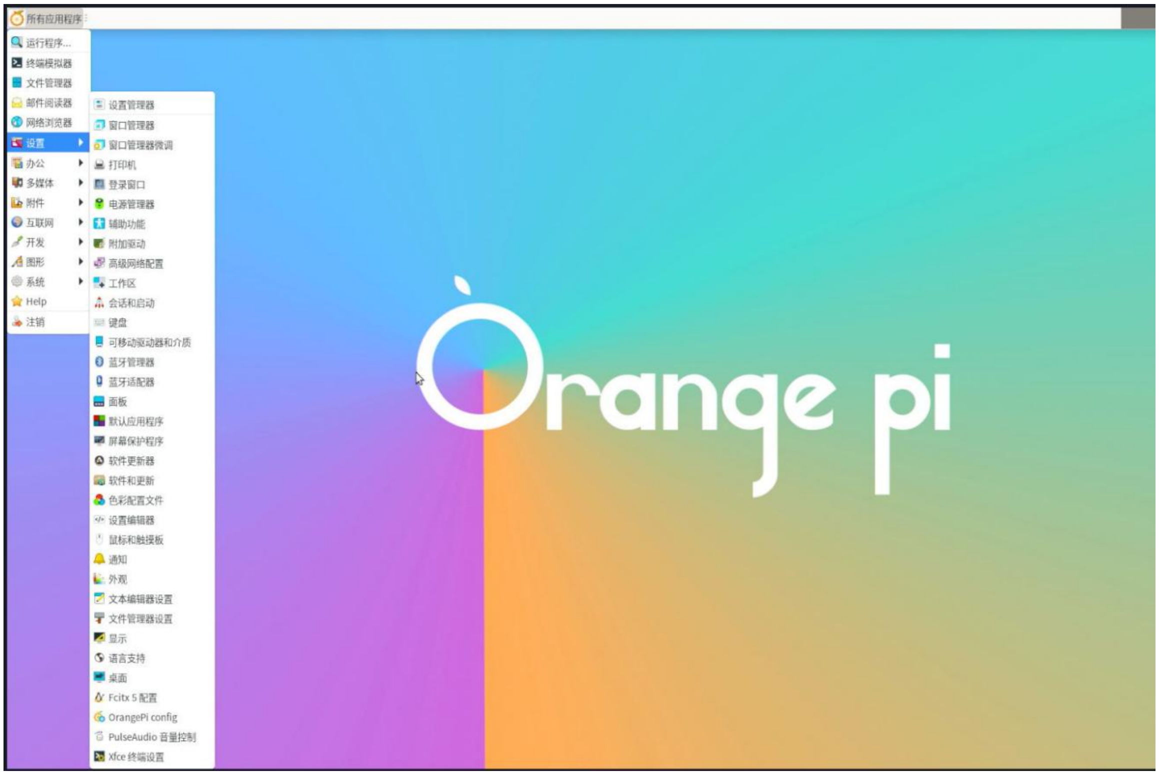


5) Then set the **Keyboard input method system** to **fcitx**.

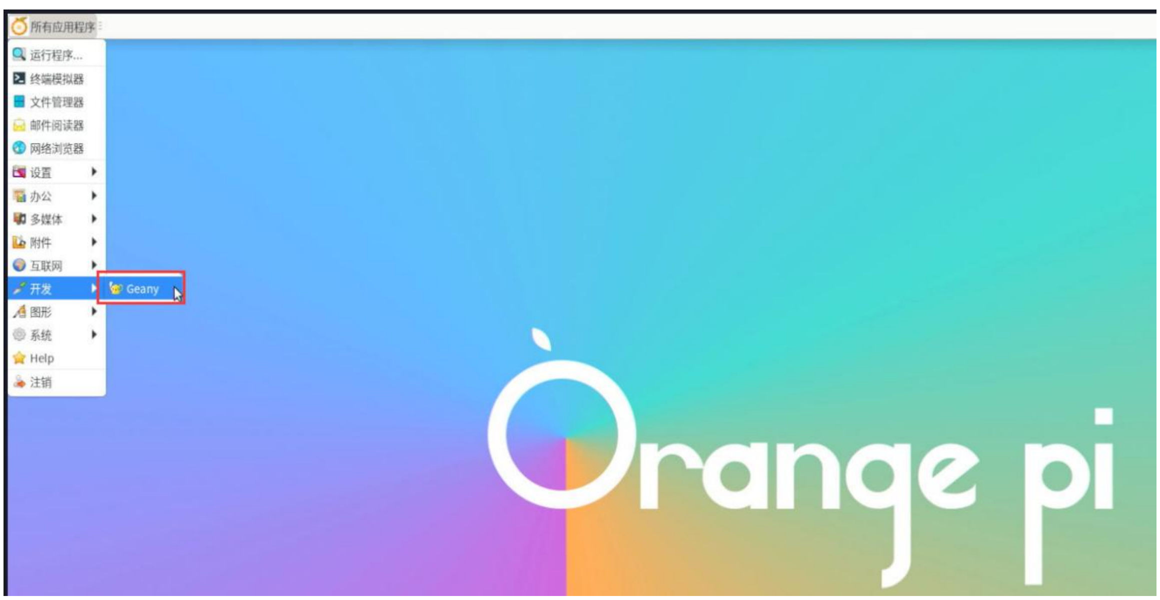
6) Then restart the Linux system to make the configuration take effect.

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.

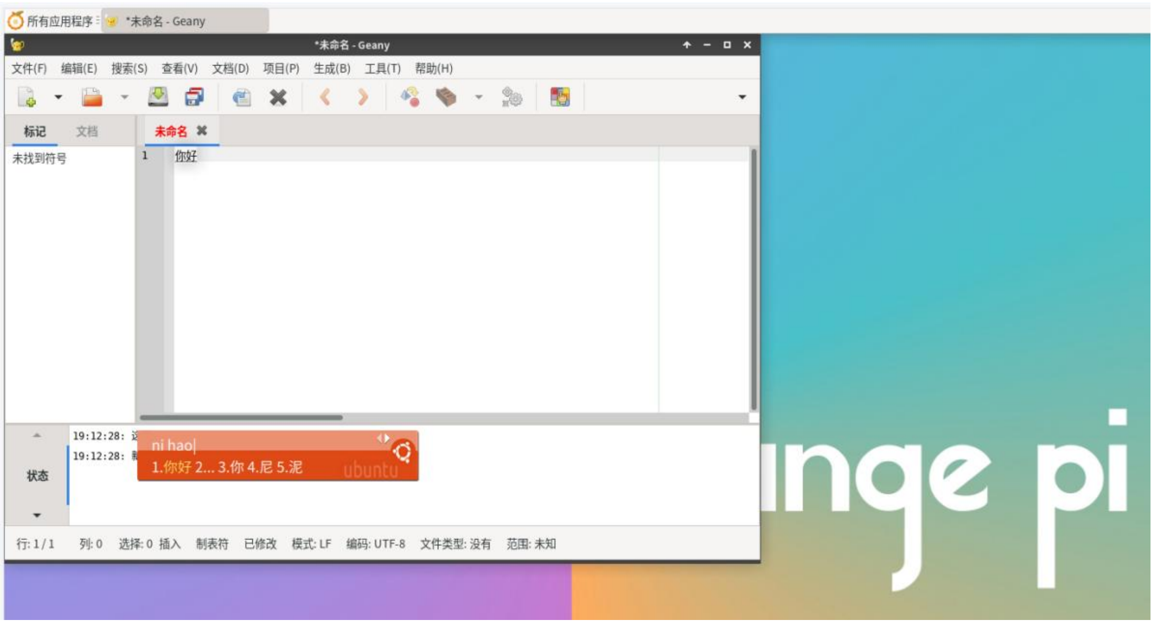
8) Then you can see that the desktop is displayed in Chinese.



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

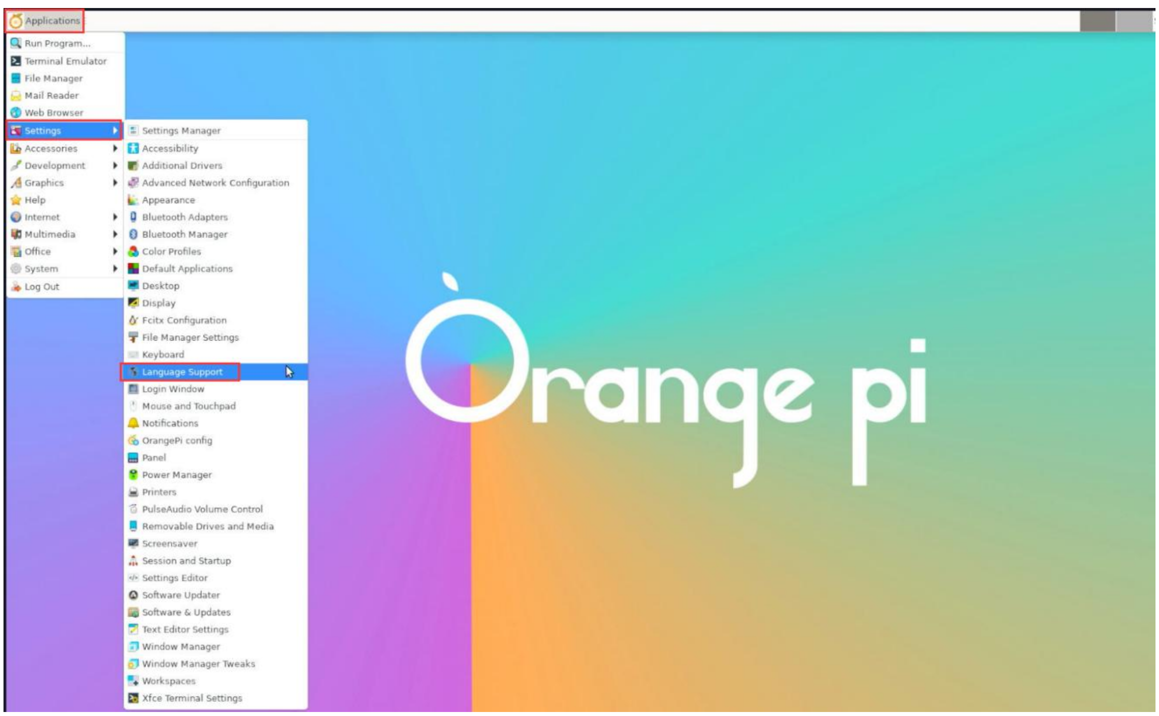


10) After opening **Geany**, the default input method is still English. We can switch to Chinese input method using the **Ctrl+Space** shortcut key, and then we can input Chinese.

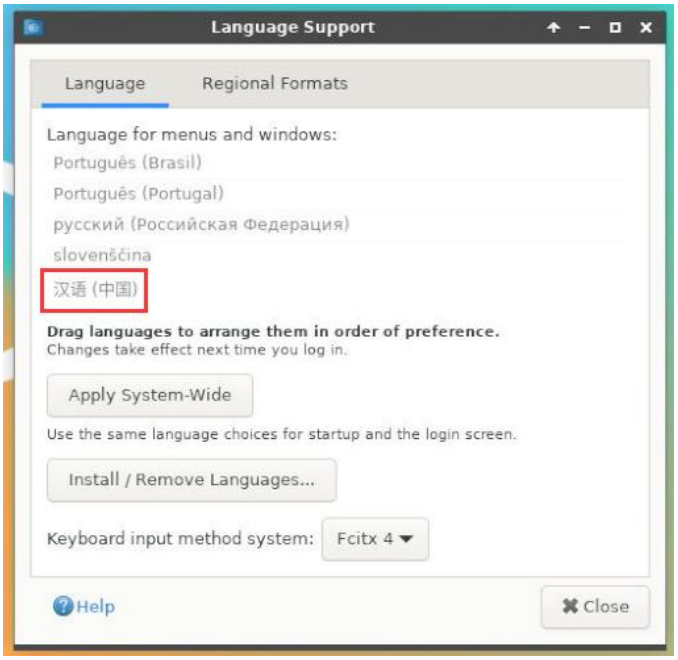


3. 26. 3. Installation method for Ubuntu 22.04 system

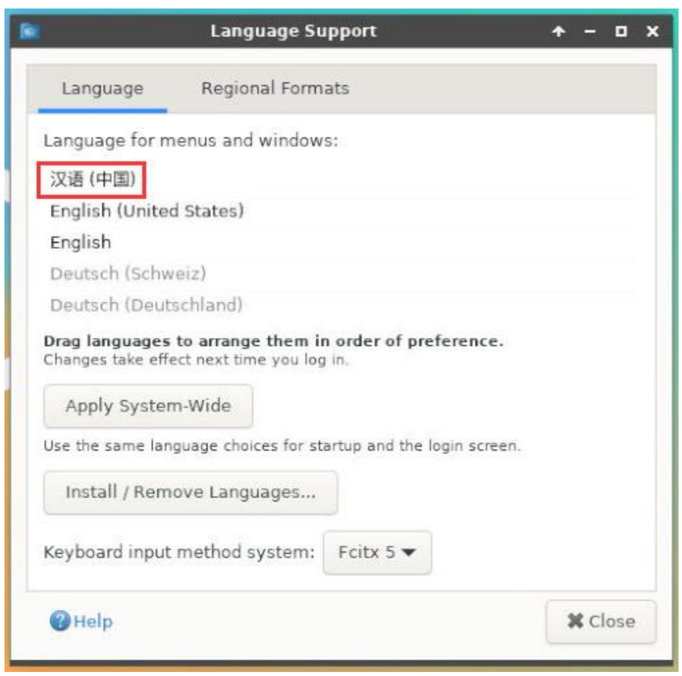
1) First open **Language Support**。



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

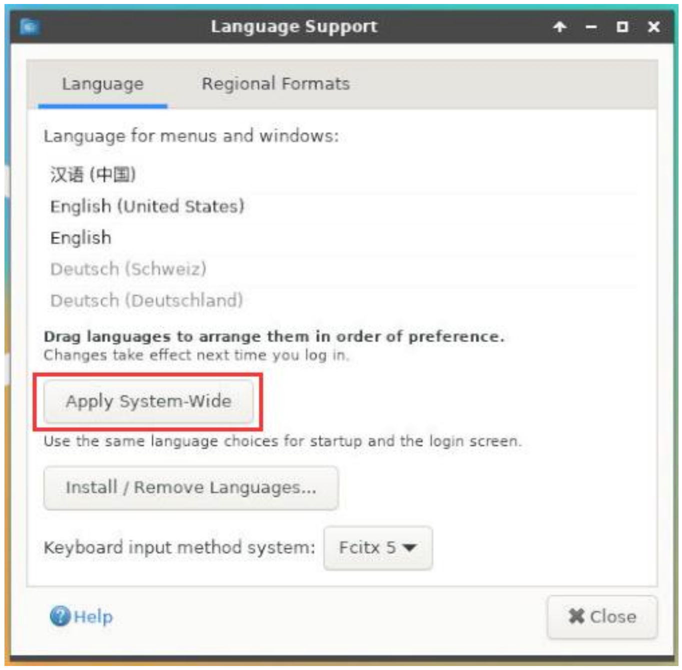


3) Then use the left mouse button to select **Chinese (China)** and hold it down, then drag it upwards to the starting position. The display after dragging is as shown in the figure below:



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

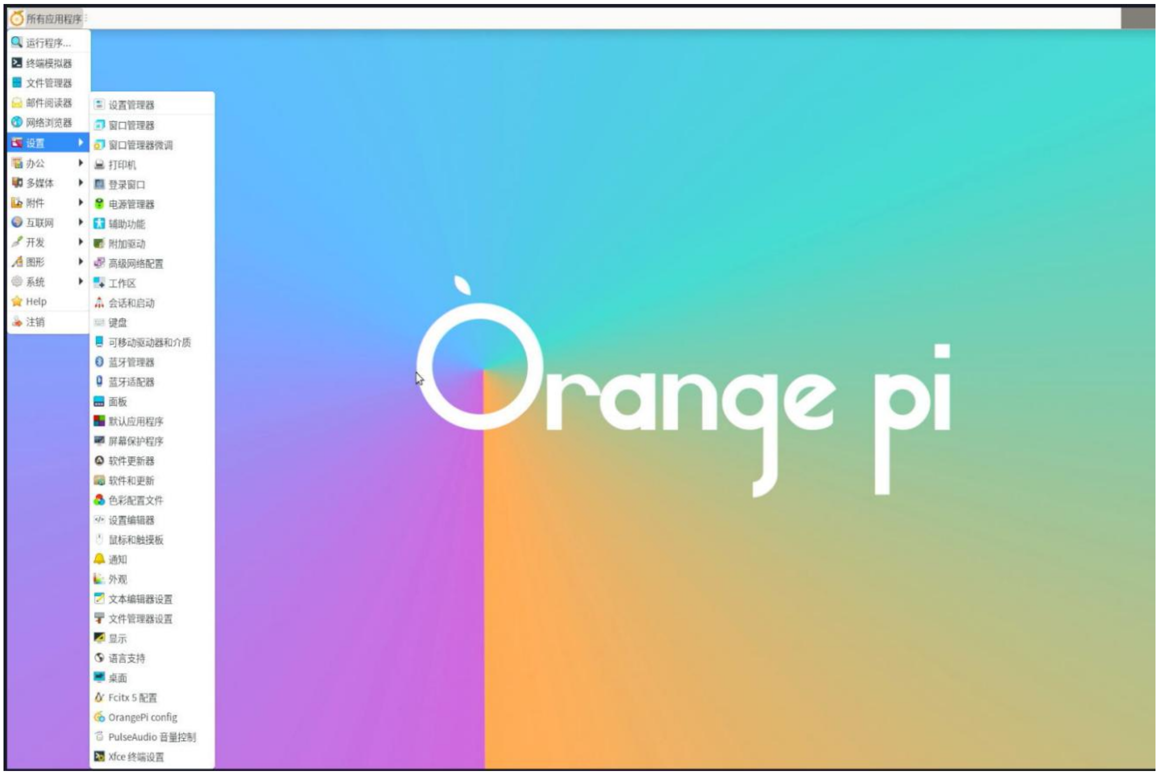
4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system.



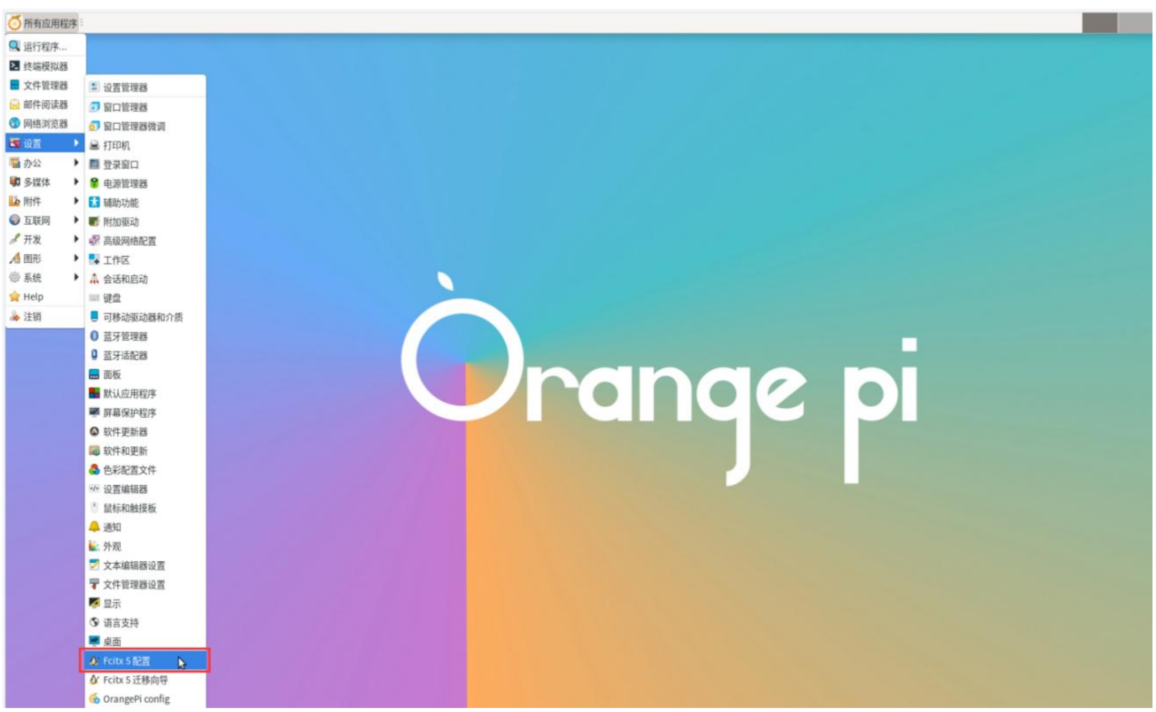
5) Then restart the Linux system to make the configuration take effect.

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.

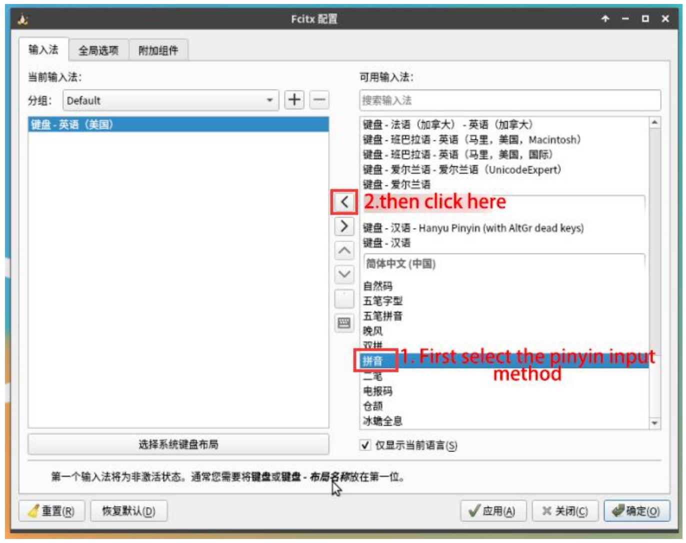
7) Then you can see that the desktop is displayed in Chinese.



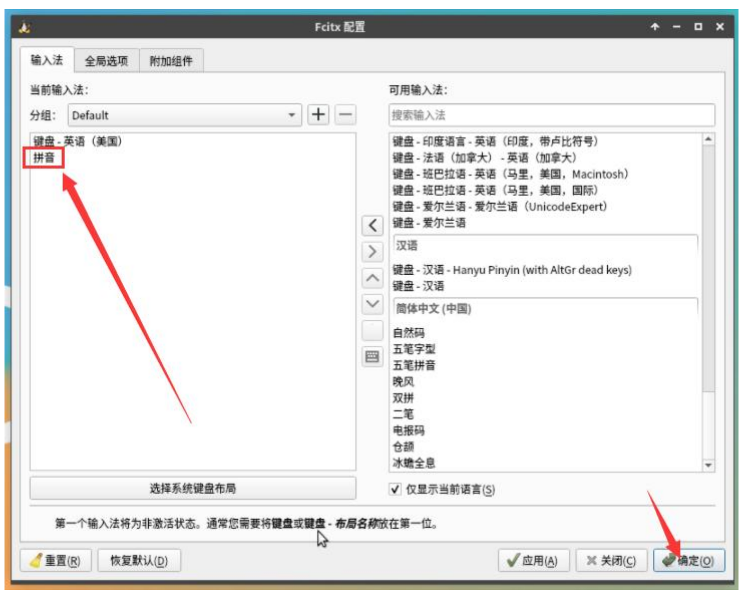
8) Then open the Fcitx5 configuration program.



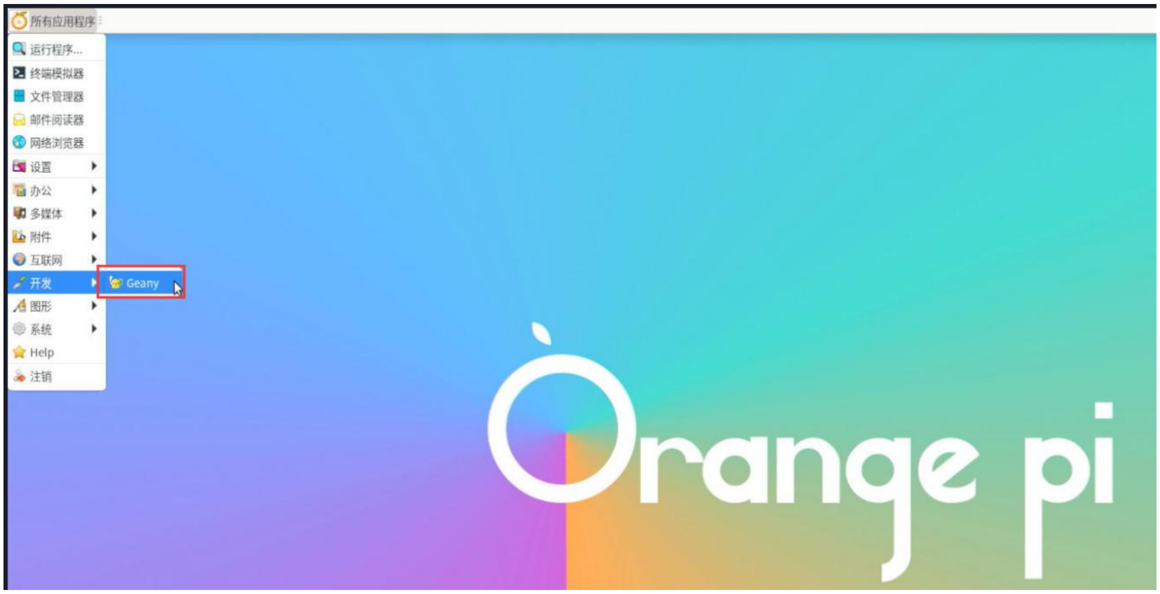
9) Then select Pinyin input method.



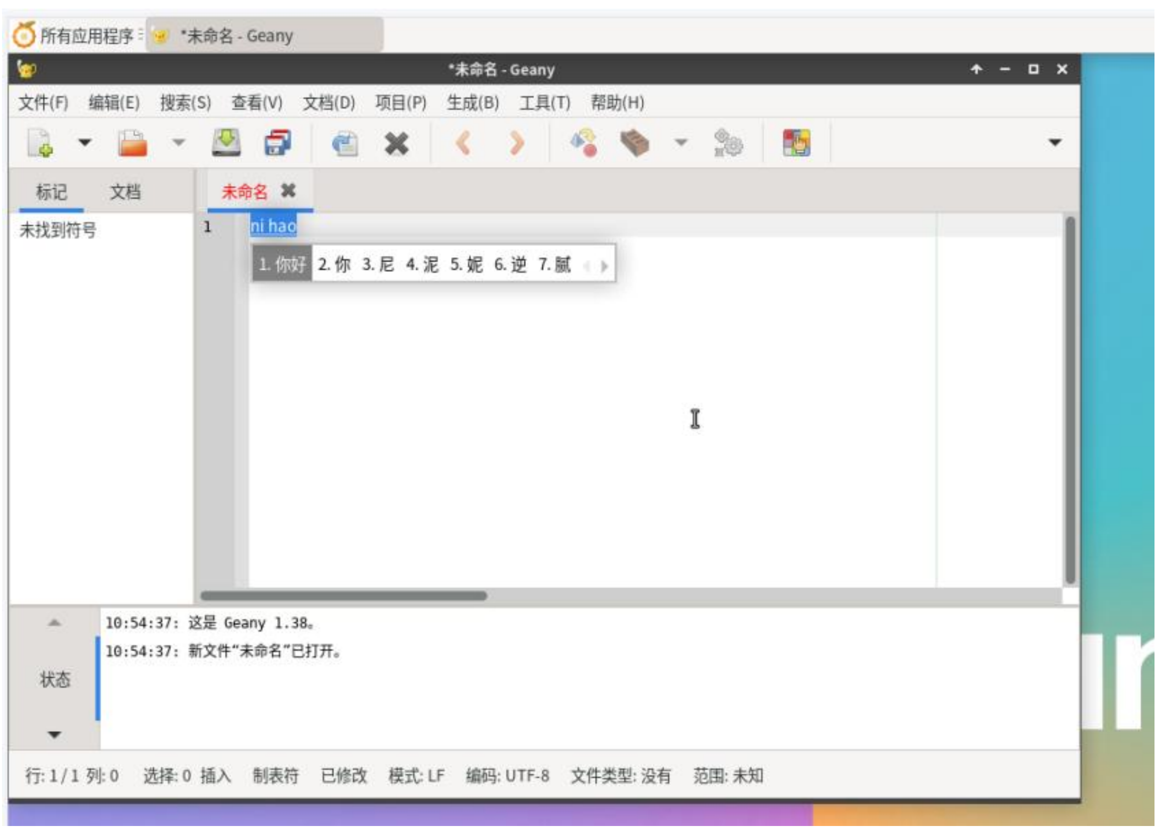
10) The interface after selection is as shown below, then click OK.



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. 27. 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. 27. 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.

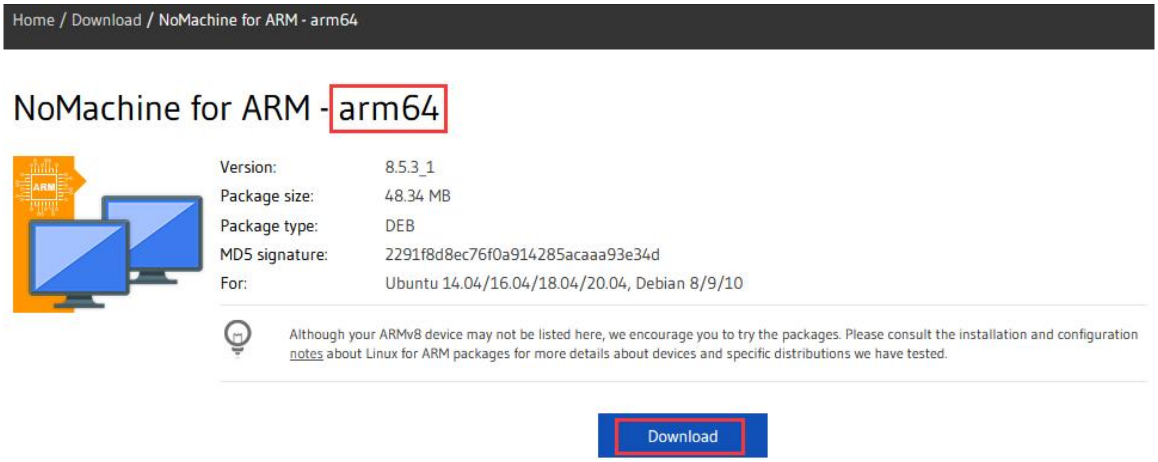
Before operation, please make sure that the Windows computer and the development board are in the same LAN and can log in to the Ubuntu or Debian system of the development board normally through SSH.

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

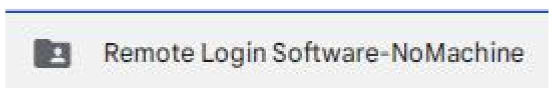
Note that this download link may change, please look for the Armv8/Arm64 version of the deb package.

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

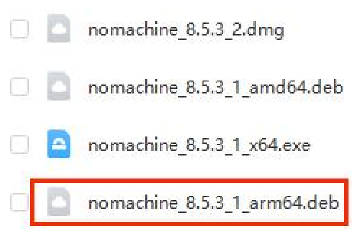


b. In addition, you can also download the **NoMachine** installation package in the **official tool.**

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

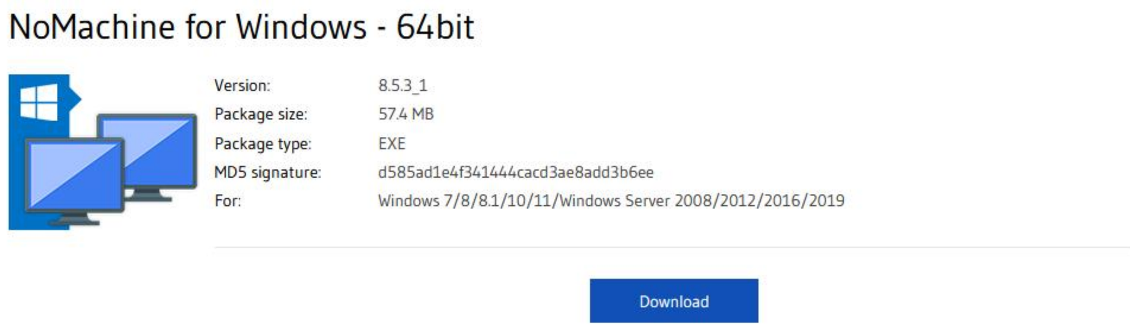
d. Then use the following command to install **NoMachine** in the Linux system of the development board.

```
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:

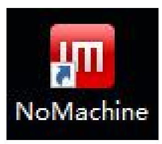
Note that this download link may change.

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

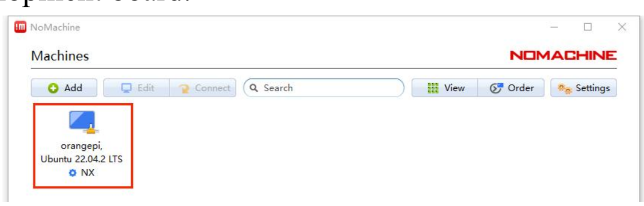


3) Then install NoMachine in Windows. **After the installation, please restart your computer.**

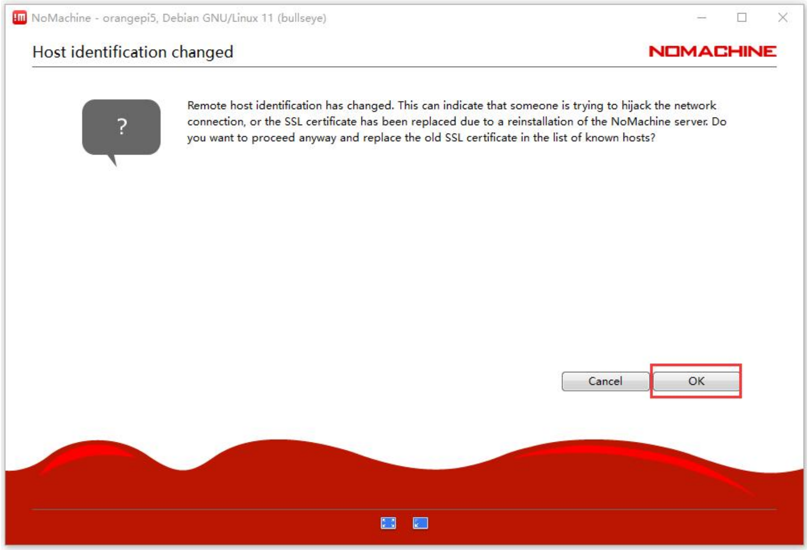
4) Then open **NoMachine** in Windows.



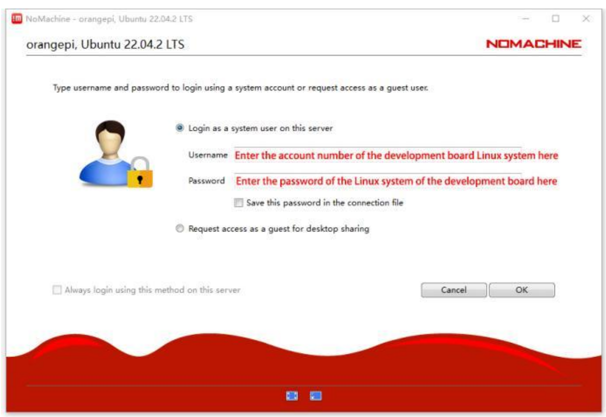
5) After NoMachine is started, it will automatically scan other devices with NoMachine installed in the LAN. After entering the main interface of NoMachine, you can see that the development board is already in the list of connectable devices. Then click the location shown in the red box in the figure below to start logging into the Linux system desktop of the development board.



6) Then click **OK**。

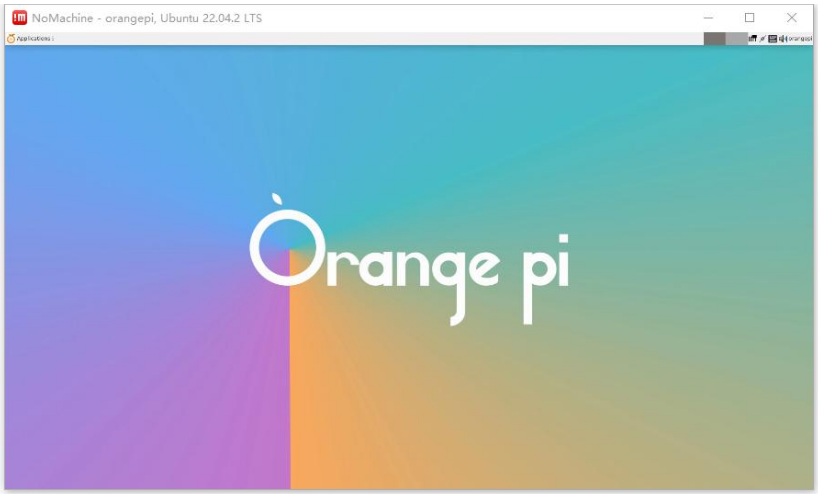


7) Then enter the user name and password of the development board Linux system in the corresponding position in the figure below, and click **OK** to start logging in.



8) Then click OK in the following interfaces.

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



3. 27. 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.

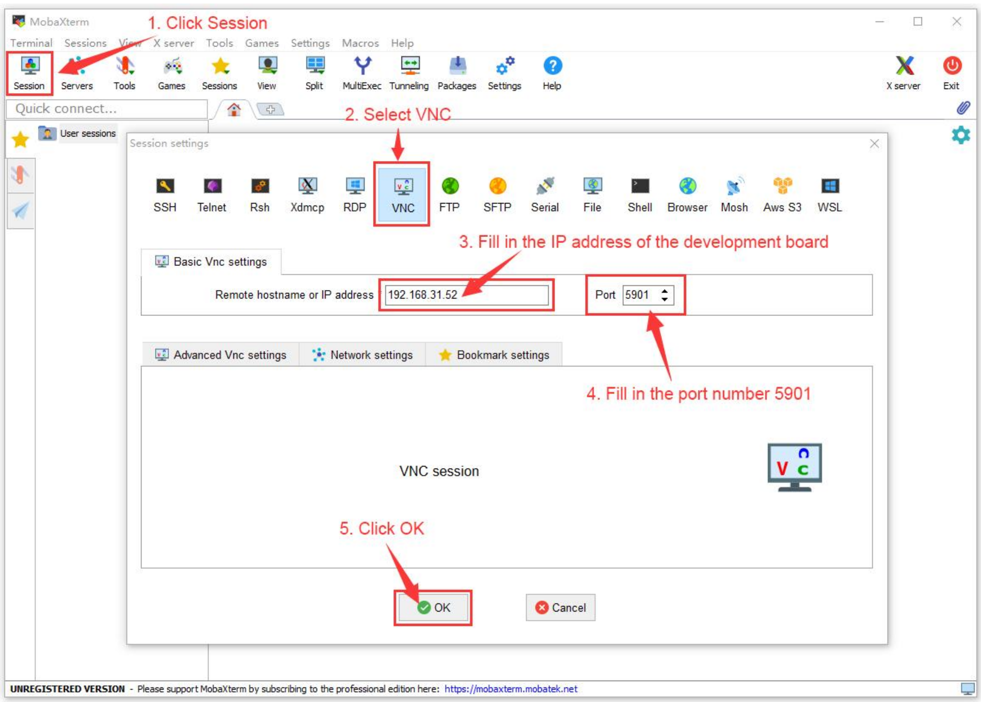
1) First run the **set_vnc.sh** script to set up vnc, **remember to add sudo permissions.**

```
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-tablet:1

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

Killing Xtightvnc process ID 3047
```

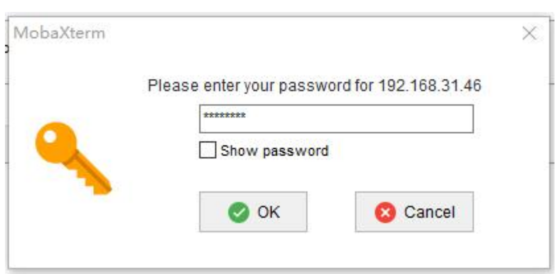
```
New 'X' desktop is orangepicm5-tablet:1

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

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

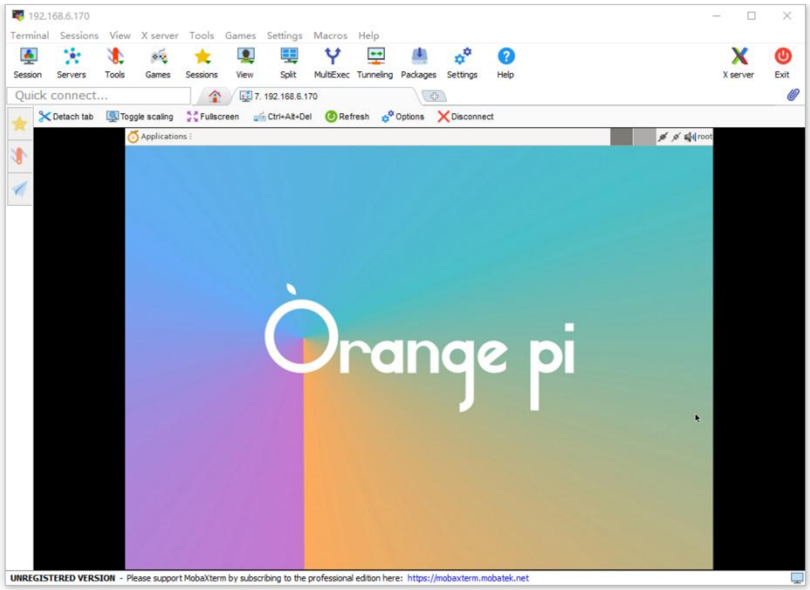
a. First click Session, then select VNC, then fill in the IP address and port of the development board, and finally click OK to confirm.



b. Then enter the VNC password set previously.



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



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

3. 28. 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 version of a.gcc is 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 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. 28. 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 version of a.gcc is as follows:

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

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

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

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

        return 0;
}
```

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

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

2) Debian Bookworm has 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. 28. 3. Ubuntu Focal System

1) Ubuntu Focal has the gcc compilation tool chain installed by default, which allows you to 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 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 Python 3 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. 28. 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 11.4.0-1ubuntu1~22.04) 11.4.0
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 3.10.12 (main, Jul 29 2024, 16:56:48) [GCC 11.4.0] 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!
```

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 18.0.2-ea 2022-07-19
OpenJDK Runtime Environment (build 18.0.2-ea+9-Ubuntu-222.04)
OpenJDK 64-Bit Server VM (build 18.0.2-ea+9-Ubuntu-222.04, 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. 29. How to install QT

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
```

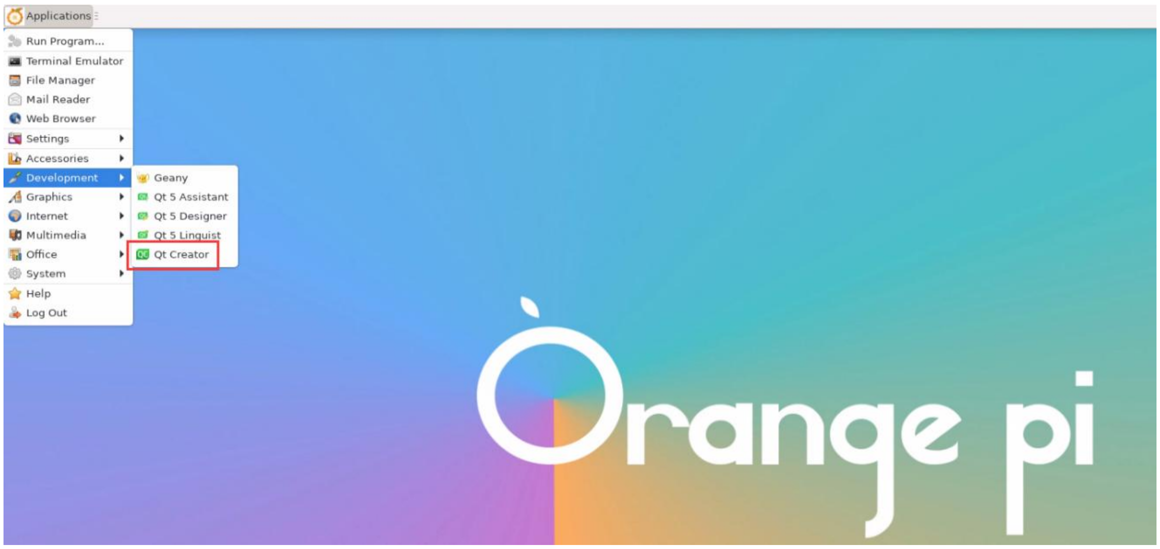
c. The QT version that comes with Debian11 is **5.15.2**

```
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.**



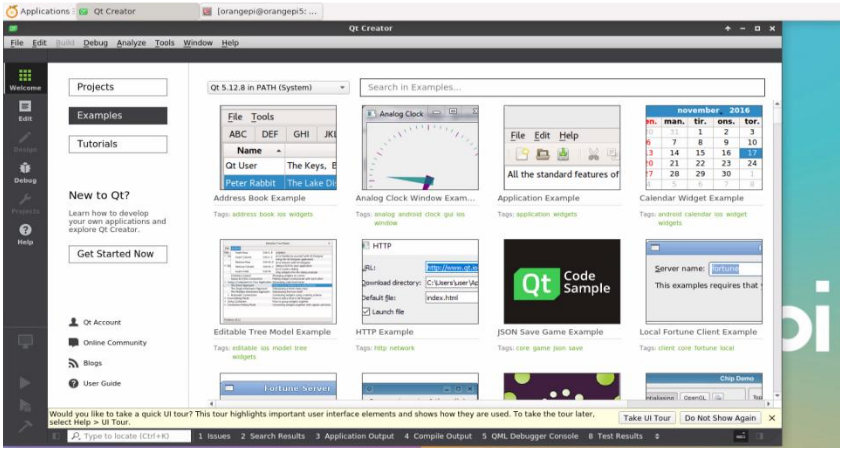
You can also use the following command to open QT Creator.

```
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.

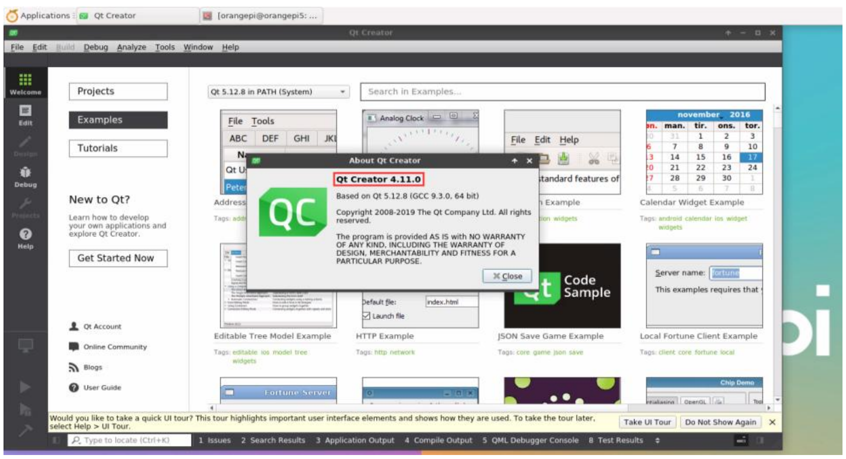
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip
libGL error: failed to create dri screen
libGL error: failed to load driver: rockchip

4) The interface after QT Creator is opened is as follows:

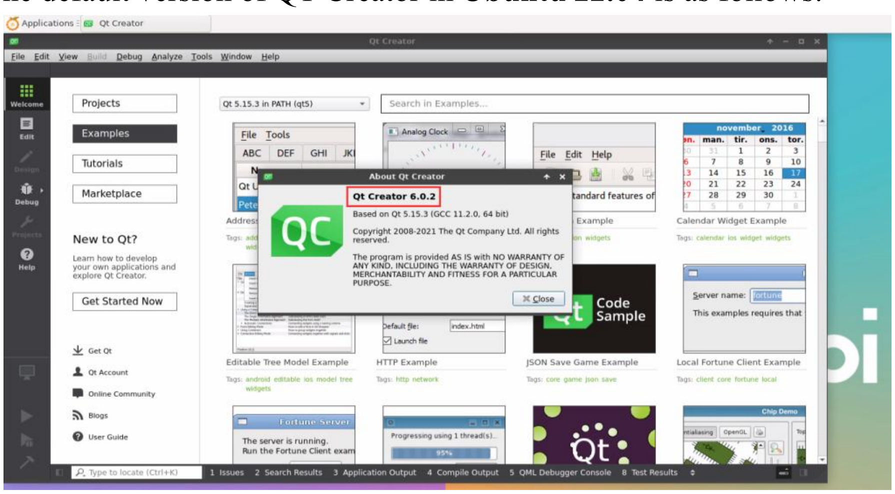


5) The version of QT Creator is as follows:

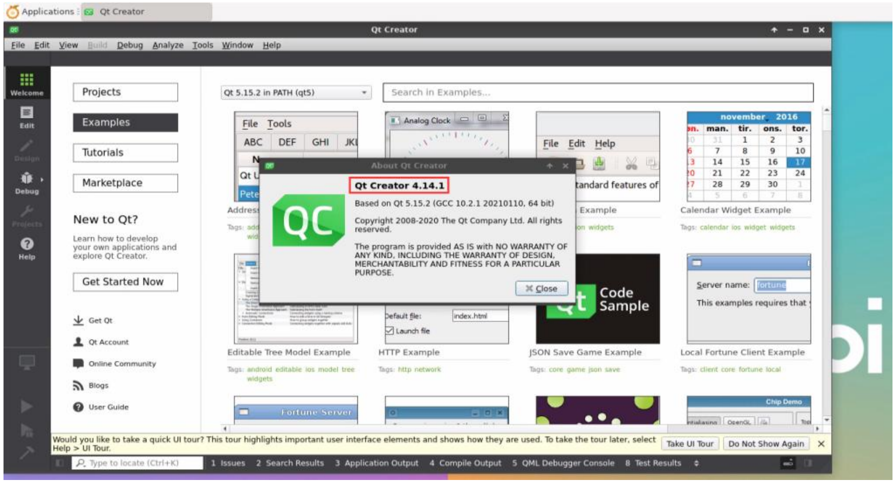
a. The default version of QT Creator in **Ubuntu 20.04** is as follows:



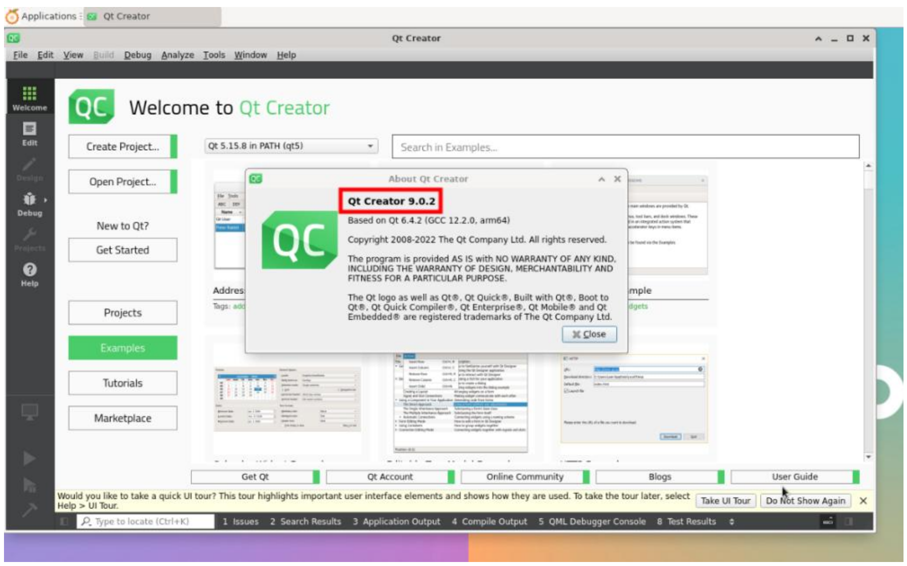
b. The default version of QT Creator in **Ubuntu 22.04** is as follows:



c. The default version of QT Creator in **Debian 11** is as follows:

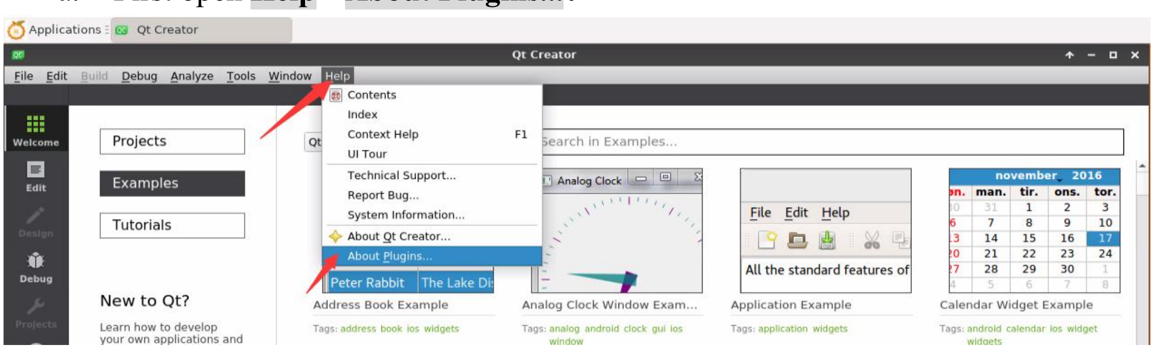


d. The default version of QT Creator in **Debian 12** is as follows:

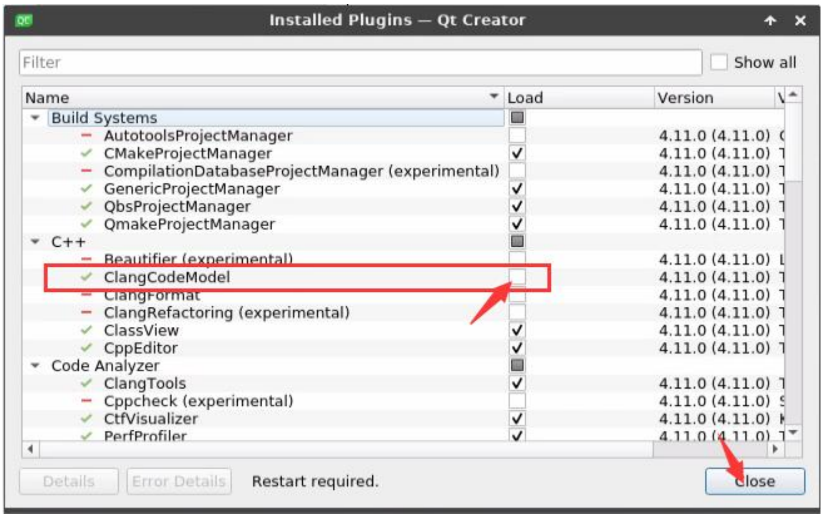


6) Then set up QT.

a. First open **Help**->**About Plugins...**。



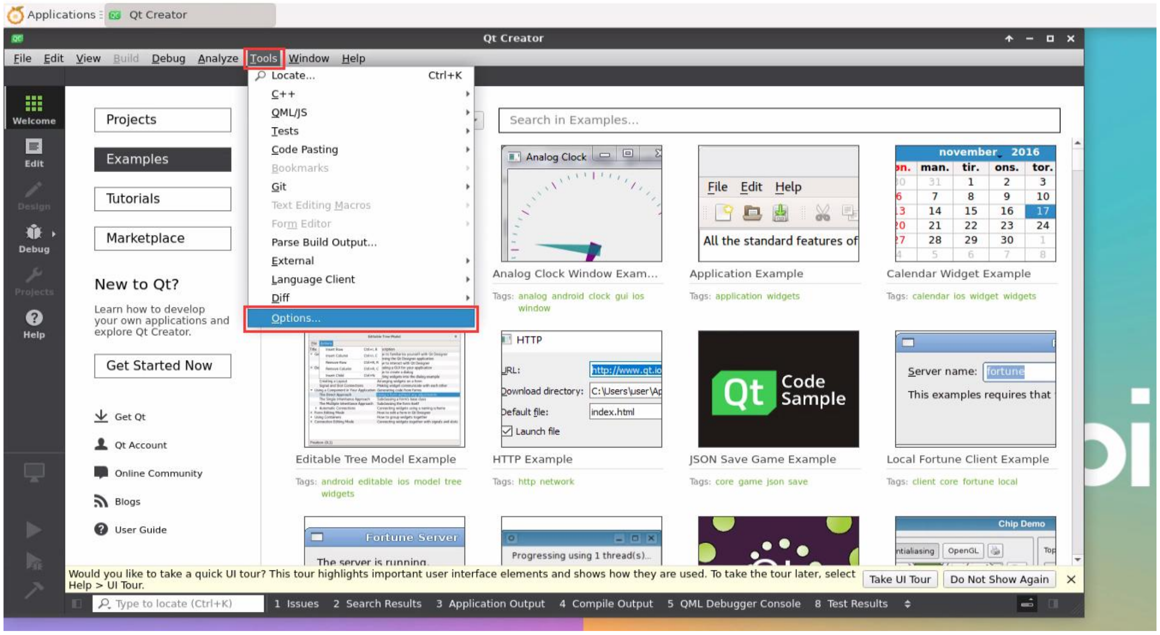
b. Then uncheck **ClangCodeModel**.

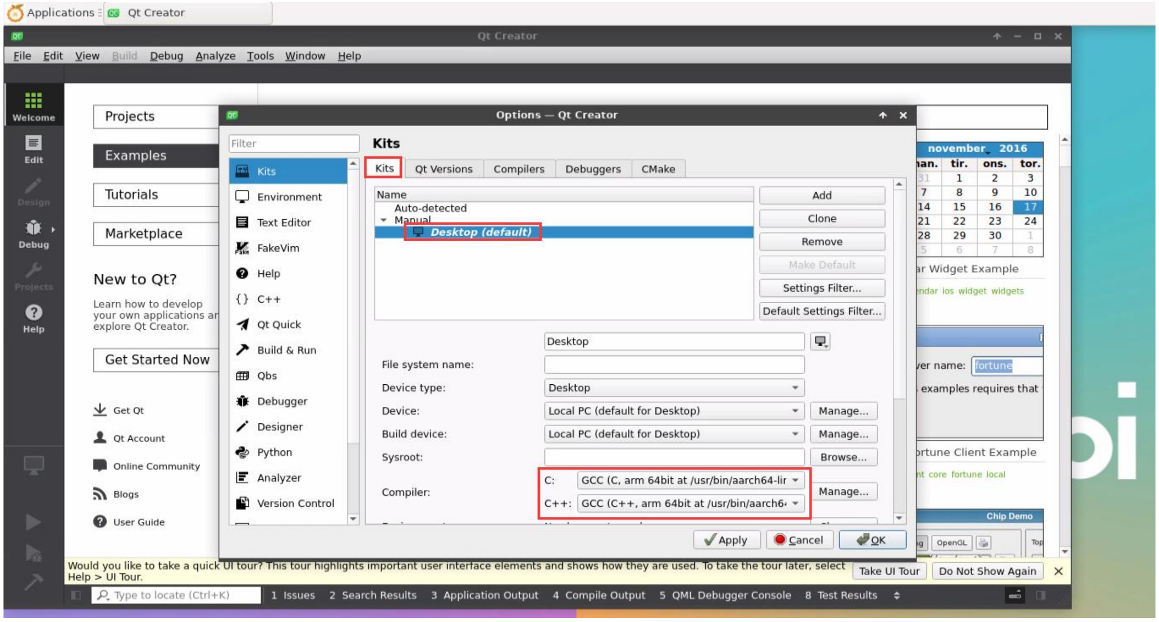


c. **After setting, you need to restart QT Creator.**

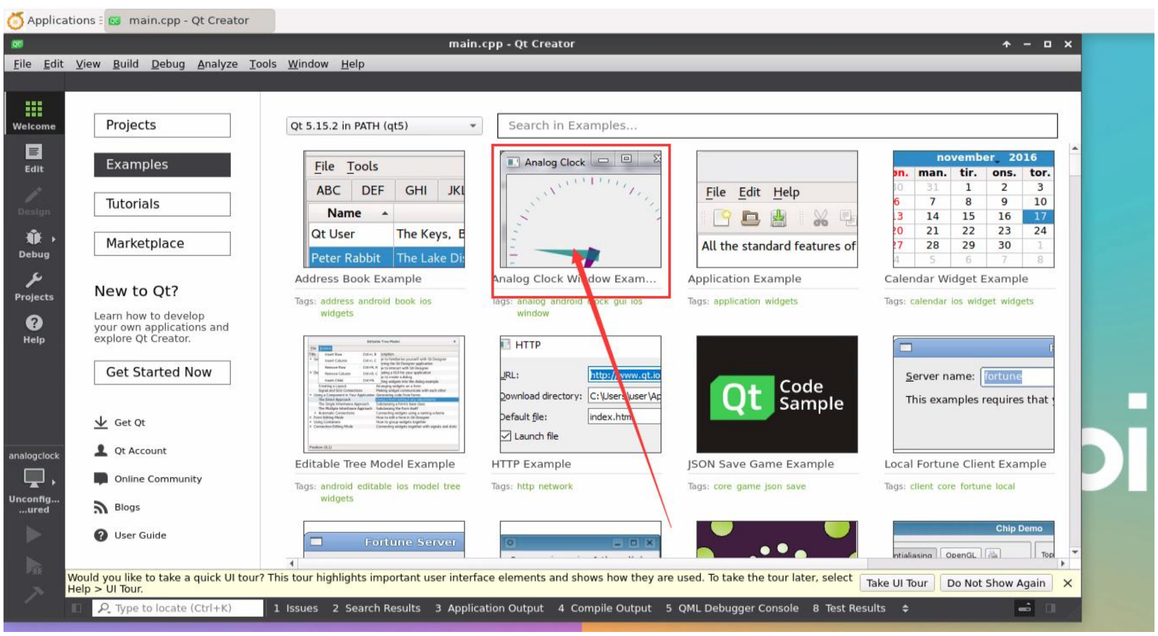
d. Then make sure that QT Creator uses the GCC compiler. If it defaults to Clang, change it to GCC.

For Debian 12, please skip this step.

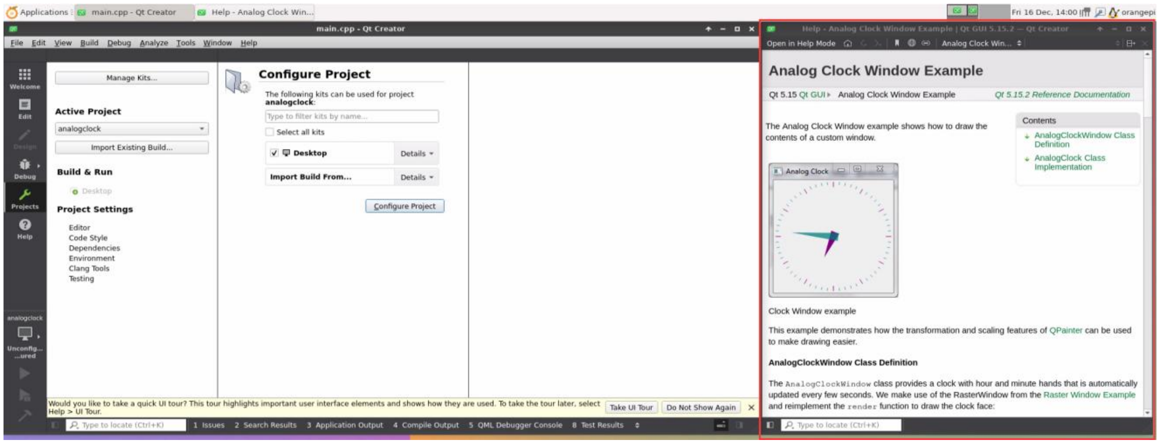




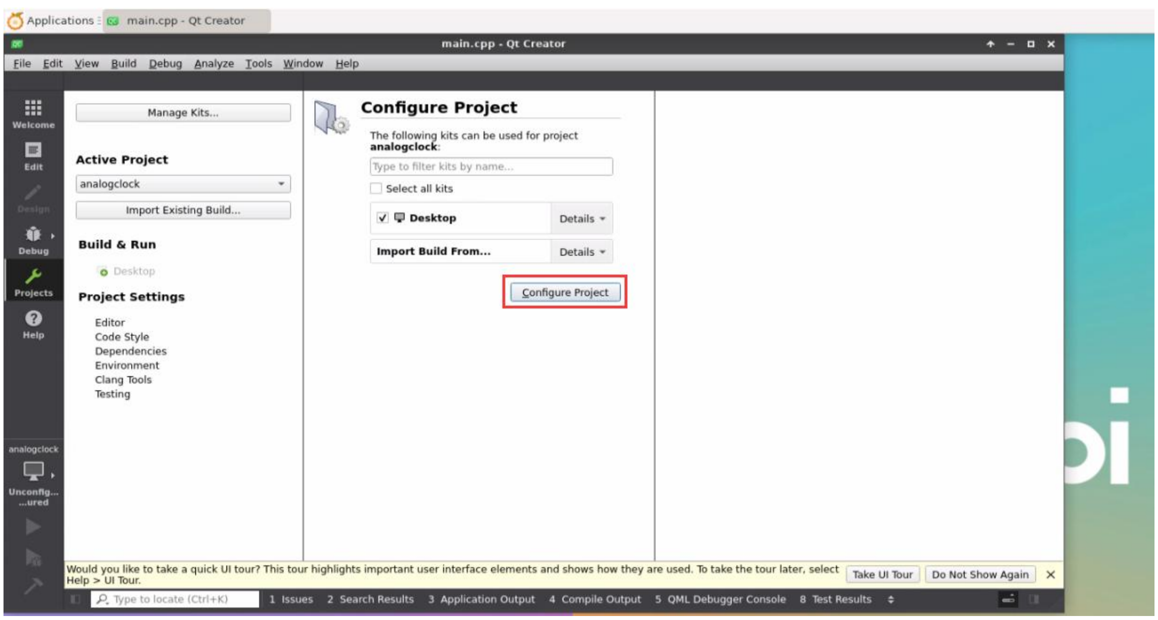
7) Then you can open a sample code.



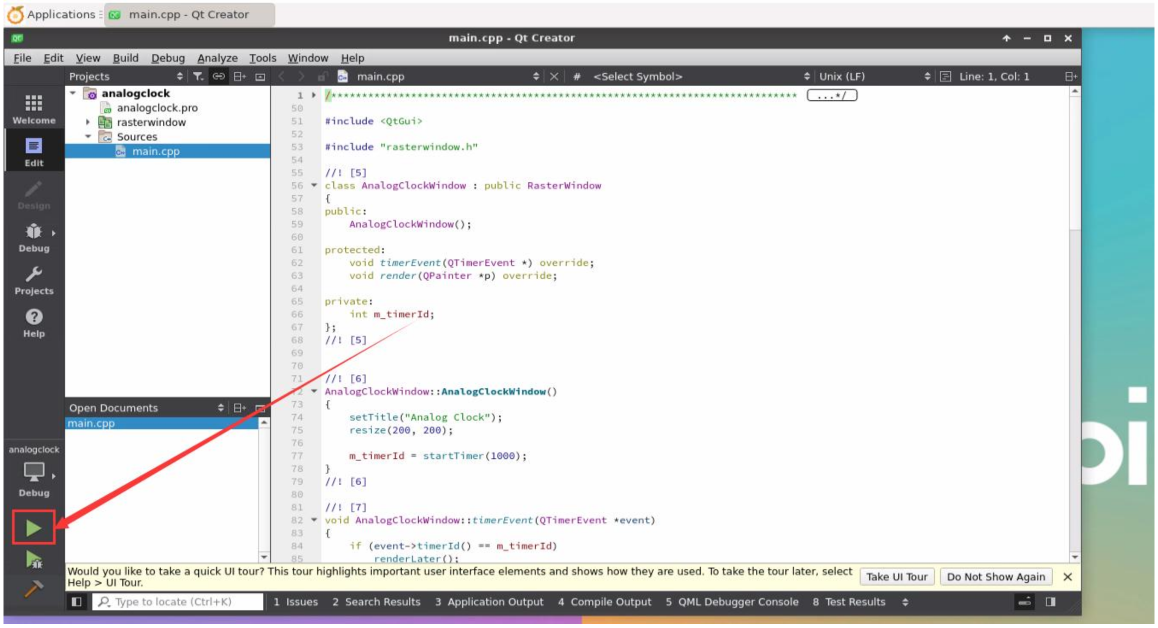
8) Clicking on the sample code will automatically open the corresponding documentation. You can read the instructions carefully.



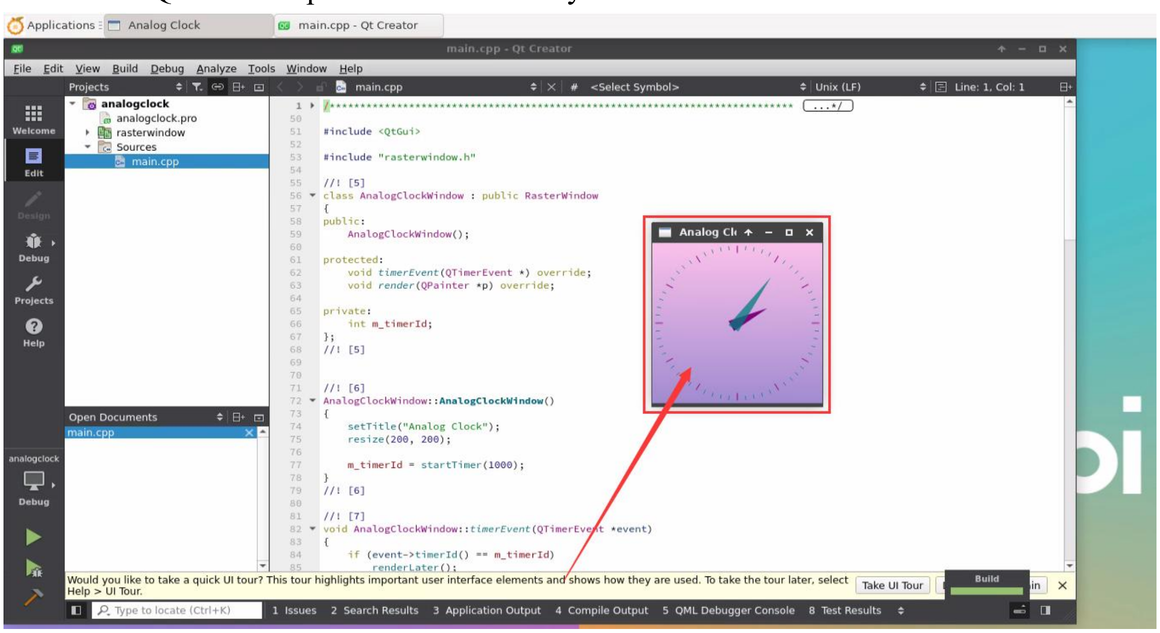
9) Then click Next **Configure Project**。



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. 30. ROS installation method

3. 30. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The currently active versions of ROS 1 are as follows, and the recommended versions are **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-tablet:~$ **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

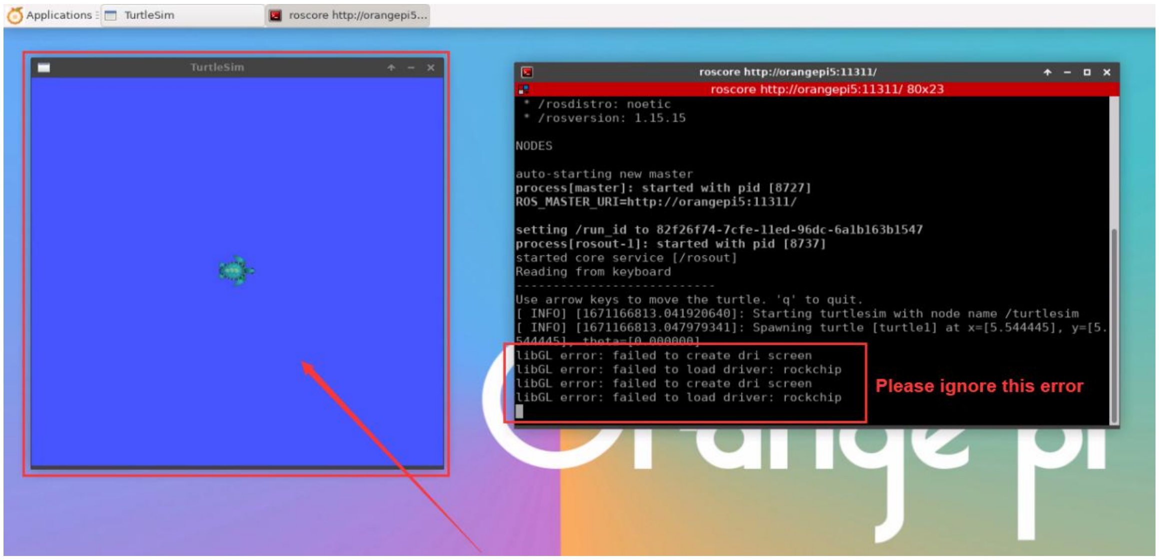
orangepi@orangepi:~$ **source /opt/ros/noetic/setup.bash**
orangepi@orangepi:~$ **sudo rosdep init**
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

```
            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
```

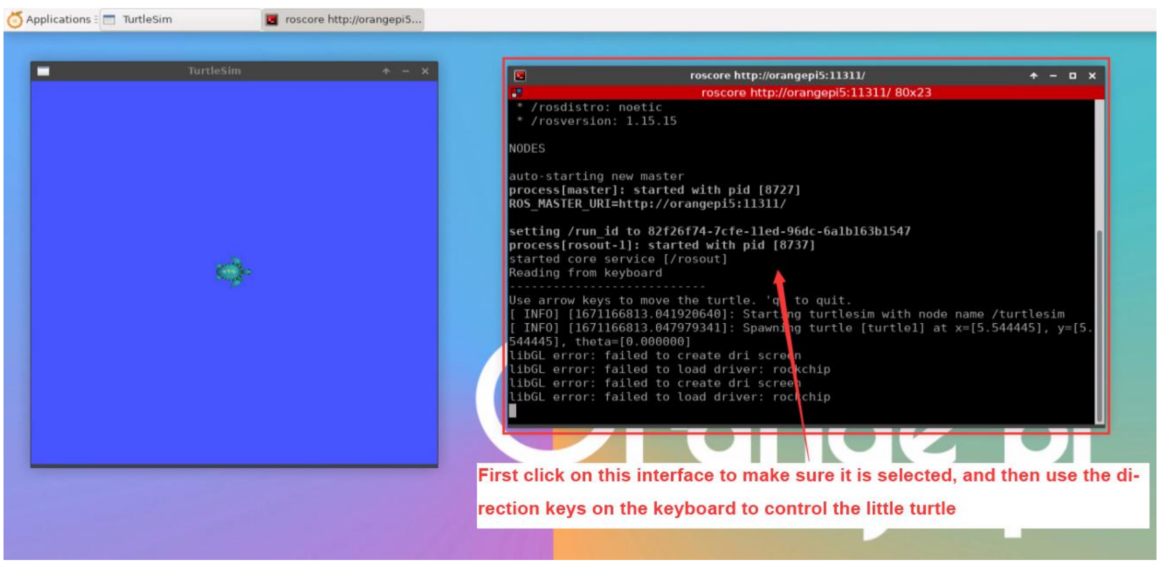
6) Then open a command line terminal window on the **desktop**, and use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally.

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

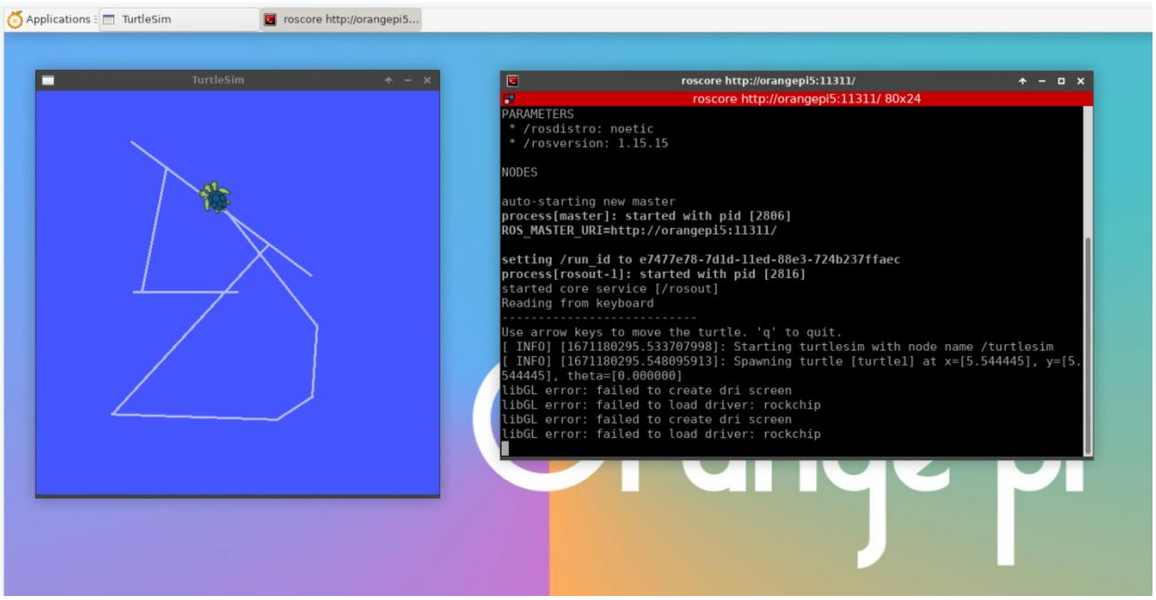
7) After running the **test_ros.sh** script, a small turtle will pop up as shown in the figure below.



8) Then please keep the terminal window you just opened on top.

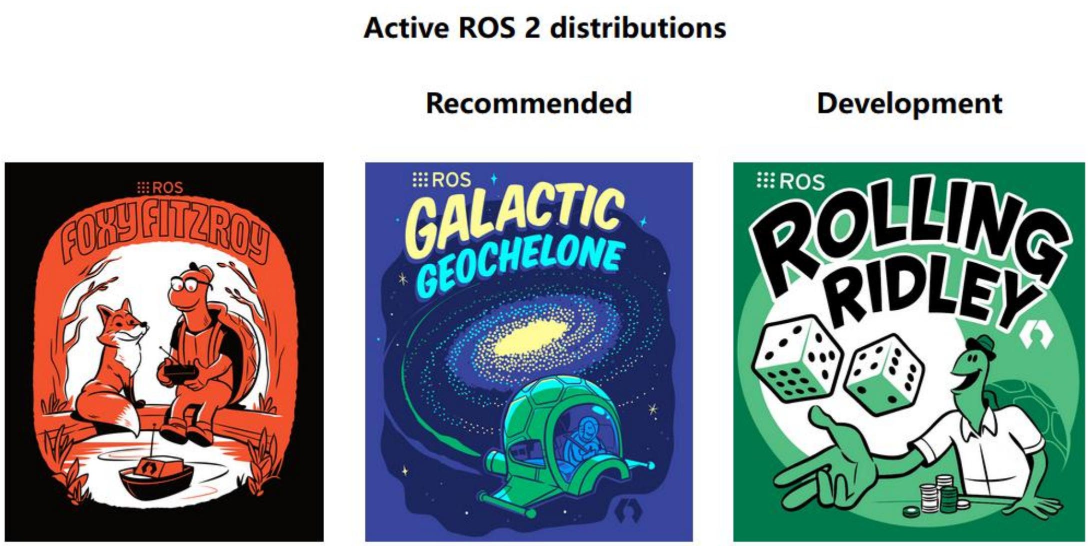


9) At this time, press the direction keys on the keyboard to control the little turtle to move up, down, left and right.



3. 30. 2. How to install ROS 2 Galactic on Ubuntu 20.04

1) The currently active versions of ROS 2 are shown below. 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.
```

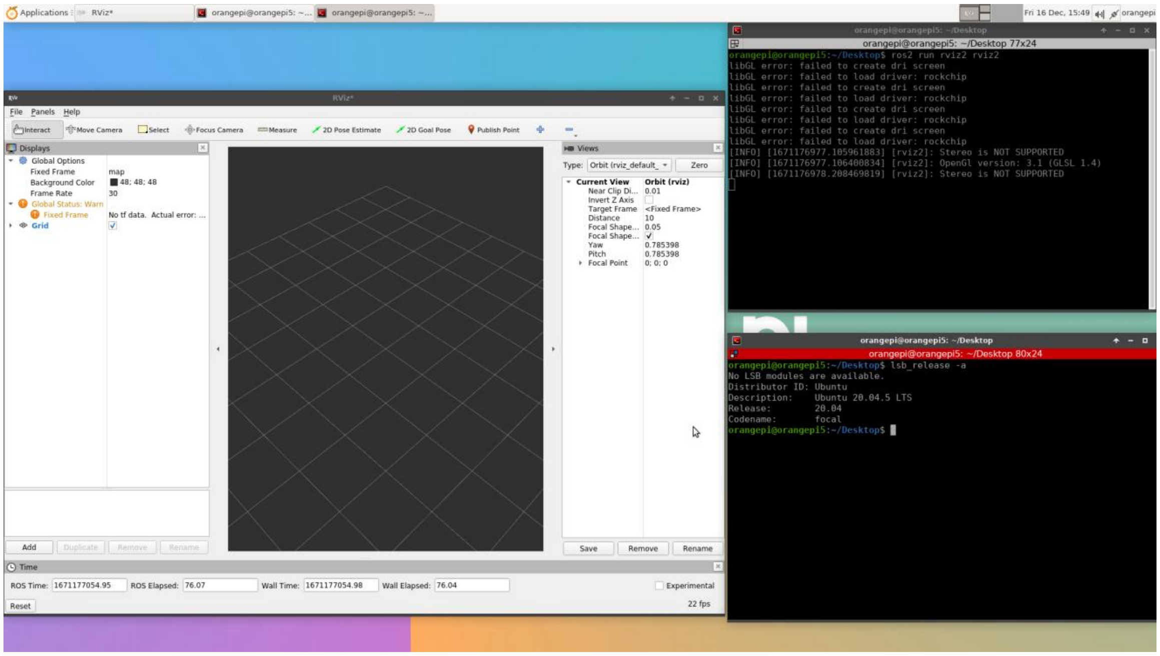
6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally.

```
orangepi@orangepicm5-tablet:~$ 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. 30. 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
```

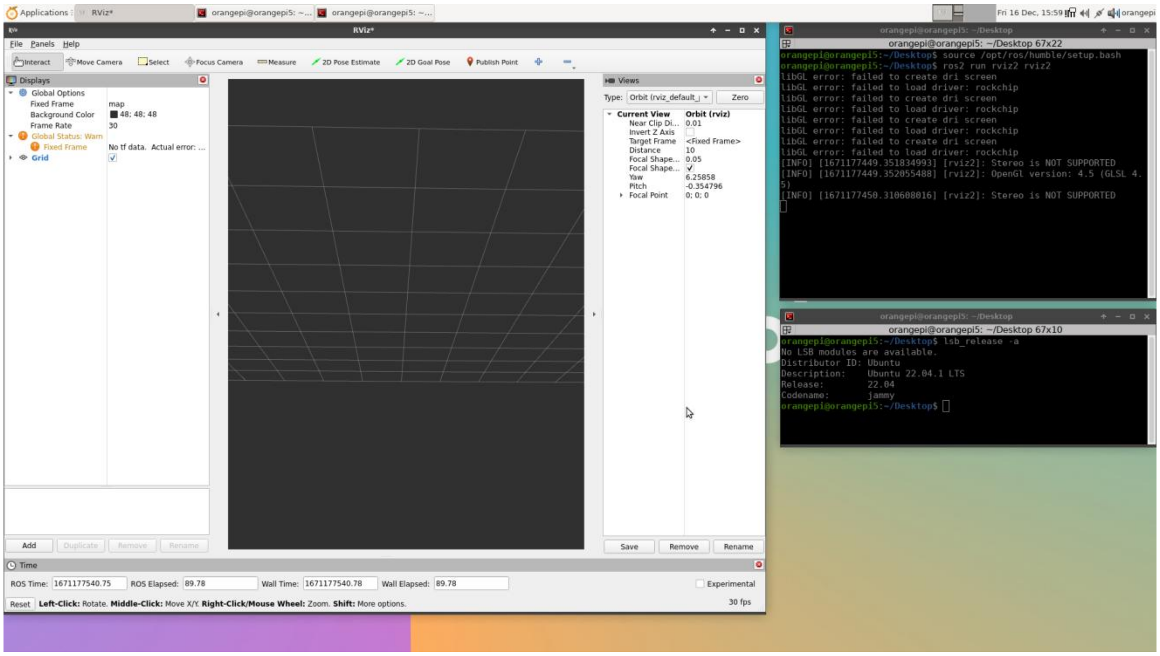
```
daemon        Various daemon related sub-commands
doctor        Check ROS setup and other potential issues
interface     Show information about ROS interfaces
launch        Run a launch file
lifecycle     Various lifecycle related sub-commands
multicast     Various multicast related sub-commands
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-tablet:~$ 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 documentation.

http://docs.ros.org/en/humble/index.html
http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

3. 31. 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");
```

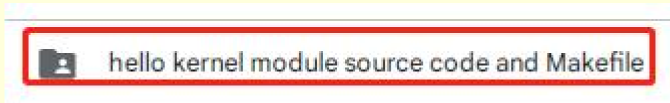
b. Then write the Makefile file to compile the hello kernel module as follows:

```
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. 32. How to use the 10.1-inch MIPI LCD screen

3. 32. 1. 10.1 inch MIPI screen assembly method

1) First prepare the necessary accessories.

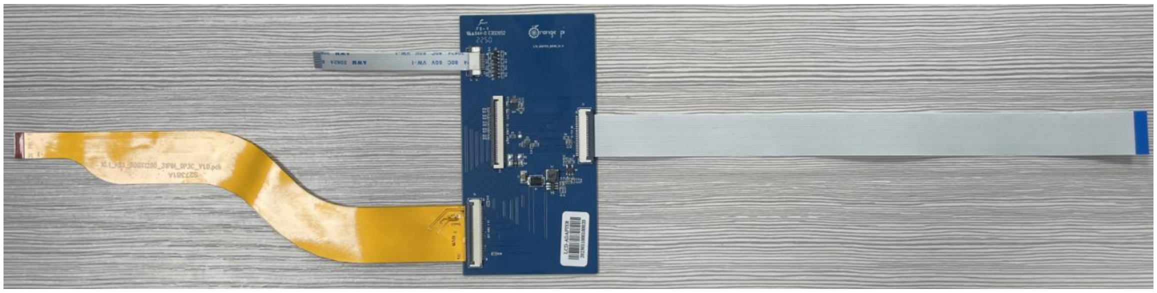
   a. 10.1 inch MIPI LCD display + touch screen.

   b. Screen adapter board + 31pin to 26pin cable.

   c. 30pin MIPI cable.

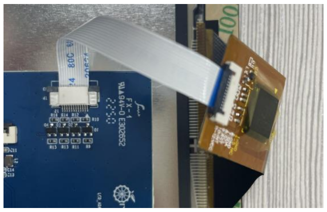
   d. 12pin touch screen cable.

2) Connect the 12-pin touch screen cable, 31-pin to 26-pin cable, and 30-pin MIPI cable to the screen adapter board as shown below. Note that **the blue insulation side of the touch screen cable should face down**, and the insulation sides of the other two cables should face up. If they are connected incorrectly, there will be no display or touch failure.

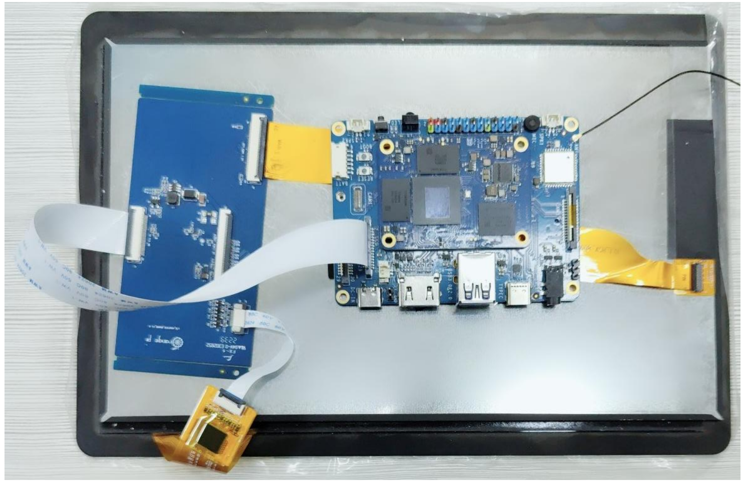


3) Place the adapter board with the connected cables on the MIPI LCD screen as shown below, and connect the MIPI LCD screen and the adapter board via a 31-pin to 26-pin cable.

4) Then connect the touch screen and the adapter board through the 12-pin touch screen cable, paying attention to the direction of the insulating surface.



5) Finally, connect it to the LCD interface of the development board via a 30-pin MIPI cable.



3. 32. 2. How to open the 10.1-inch MIPI LCD screen configuration

1) The Linux image does not have the mipi lcd screen configuration turned on by default. If you need to use the mipi lcd screen, you need to turn it on manually.

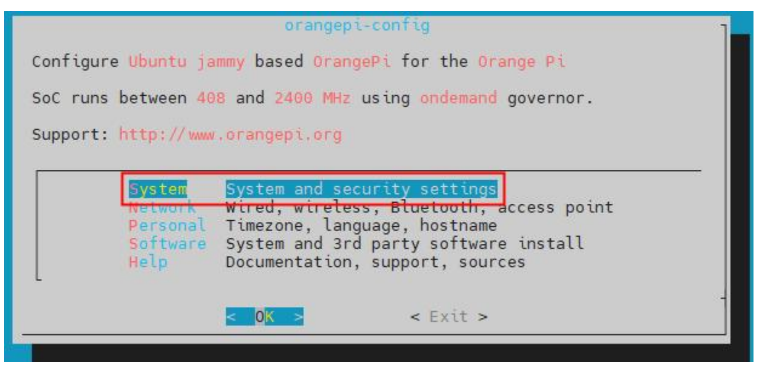
2) The location of the interface of the mipi lcd screen on the development board is shown in the figure below:

3) The steps to open the mipi lcd configuration are as follows:

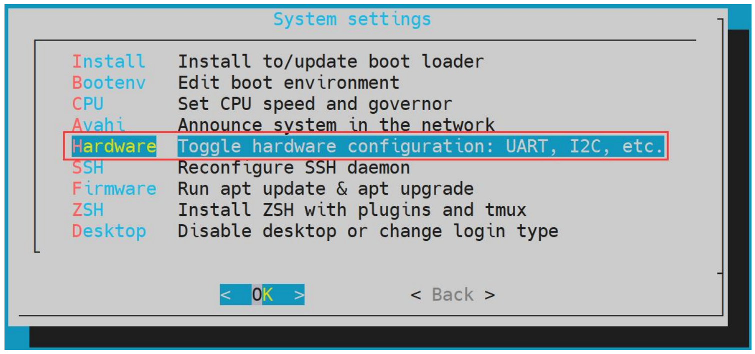
a. First run **orangepi-config.** Remember to add **sudo** permissions as a normal user.

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

b. Then select **System**。



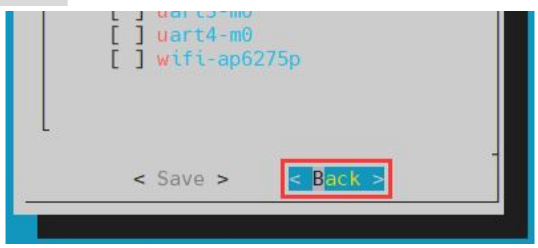
c. Then select **Hardware**。



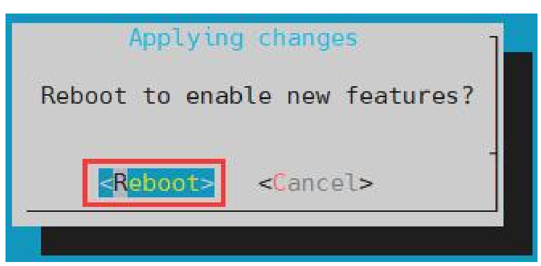
d. Then use the arrow keys on your keyboard to locate **opicm5-tablet-lcd**, and then use the **spacebar** to select it.

e. Then select <Save> to save

f. Then select **<Back>**。



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



The above configuration will eventually add overlays=opicm5-tablet-lcd to /boot/orangepiEnv.txt. You can check it after setting it. If this line does not exist,

then there is a problem with the configuration.

If you find it troublesome to use orangepi-config, you can also use the vim editor to open /boot/orangepiEnv.txt and add the line overlays=opicm5-tablet-lcd.

```
orangepi@orangepi:~$ cat /boot/orangepiEnv.txt | grep "lcd"
overlays=opicm5-tablet-lcd        #Example Configuration
```

4) After startup, you can see the LCD screen display as follows (the default is vertical):

3. 32. 3. How to rotate the display direction of the server version image

1) Add **extraargs=fbcon=rotate: direction to be rotated** in **/boot/orangepiEnv.txt** to set the display direction of the server version of Linux system. The number after **fbcon=rotate:** can be set to:

a. 0: Normal screen (portrait by default)
b. 1: Rotate 90 degrees clockwise
c. 2: Flip 180 degrees
d. 3: Rotate 270 degrees clockwise

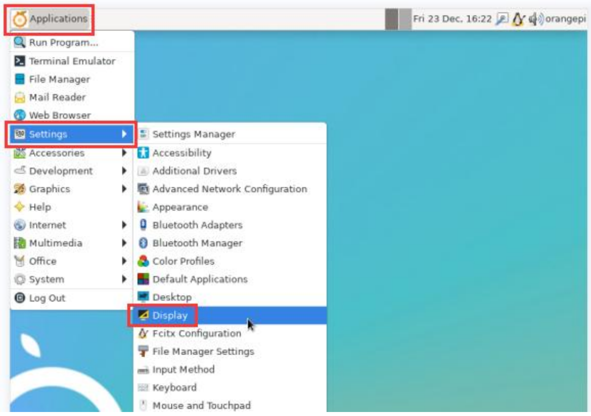
```
orangepi@orangepi:~$ sudo vim /boot/orangepiEnv.txt
overlays=opicm5-tablet-lcd
extraargs=cma=128M fbcon=rotate:3
```

Note that if there is a line of extraargs=cma=128M in /boot/orangepiEnv.txt by default, you can add the line fbcon=rotate:3 after extraargs=cma=128M (separated by a space).

2) Then **restart** the Linux system and you will see that the direction of the LCD screen display has been rotated.

3. 32. 4. How to mirror the display and touch direction on the desktop version

1) First open the **Display** settings in the Linux system.



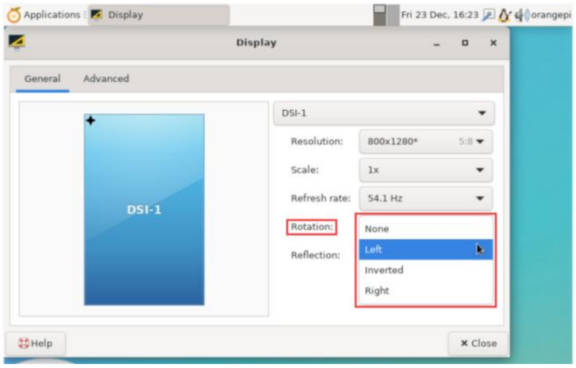
2) Then select the direction you want to rotate in **Rotation.**

a. **None**: No rotation

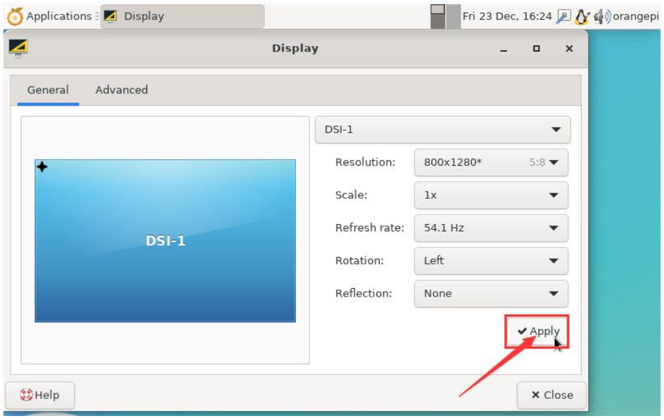
b. **Left**: Rotate 90 degrees left

c. **Inverted**: Flip upside down, equivalent to rotating 180 degrees

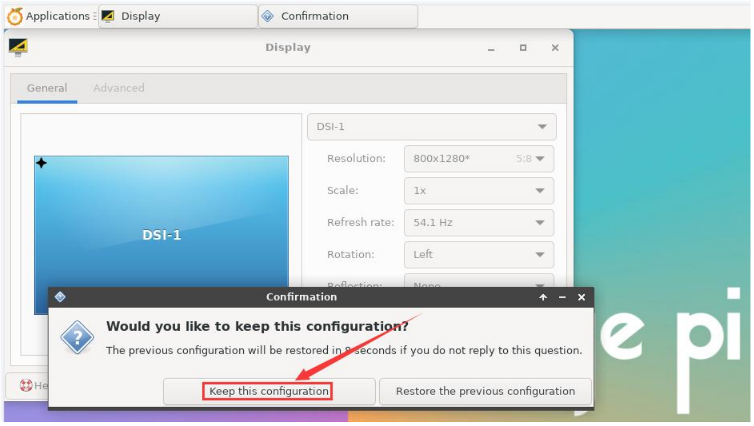
d. **Right**: Rotate 90 degrees to the right



3) Then click **Apply**。



4) Then select **Keep this configuration**。



5) At this point the screen display has been rotated, and you can then close the **Display** program.

6) The above steps will only select the display direction, but will not rotate the touch direction. You can use the **set_lcd_rotate.sh** script to rotate the touch direction. After setting this script, it will automatically restart, and then you can test whether the touch can be used normally.

a. **None**: No rotation

```
orangepi@orangepi:~$ set_lcd_rotate.sh none
```

b. **Left**: Rotate 90 degrees to the left

```
orangepi@orangepi:~$ set_lcd_rotate.sh left
```

c. **Inverted**: Flip upside down, equivalent to rotating 180 degrees

```
orangepi@orangepi:~$ set_lcd_rotate.sh inverted
```

d. **Right**: Rotate 90 degrees to the right

```
orangepi@orangepi:~$ set_lcd_rotate.sh right
```

The set_lcd_rotate.sh script does four main things:

1. **Rotate the direction of the framebuffer display**
2. **Rotate the touch direction**
3. **Turn off power logo**
4. **Restart the system**

Rotating the touch direction is achieved by adding Option "TransformationMatrix" "x x x x x x x x x" to /usr/share/X11/xorg.conf.d/40-libinput.conf, where "x x x x x x x x x" has different configurations for different directions.

7) Touch to rotate the reference.

```
https://wiki.ubuntu.com/X/InputCoordinateTransformation
```

3. 33. 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:~$ sudo 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:~$ sudo 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
```

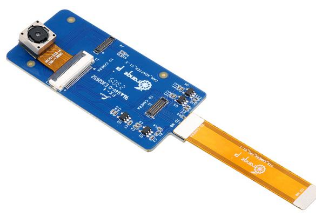
5) After replacing the boot logo image, you need to run the following command to make it take effect.

```
orangepi@orangepi:~$ sudo update-initramfs -u
```

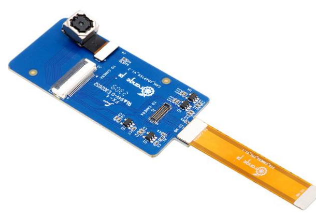
3. 34. Test Methods for OV13850 and OV13855 MIPI Cameras

Currently the development board supports two MIPI cameras, OV13850 and OV13855. The specific pictures are as follows:

a. 13MP OV13850 camera with MIPI interface.



b. 13MP OV13855 camera with MIPI interface.



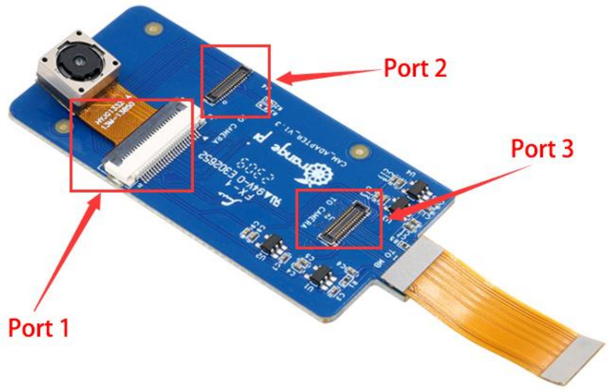
The adapter board and FPC cable used by OV13850 and OV13855 cameras are the same, but the two cameras are connected to the adapter board in different positions. The FPC cable is shown in the figure below. Please note that the FPC cable has a direction. The end marked with **TO MB** needs to be plugged into the camera interface of the development board, and the end marked with **TO CAMERA** needs to be plugged into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board. Only one can be connected at a time, as shown in the following figure:

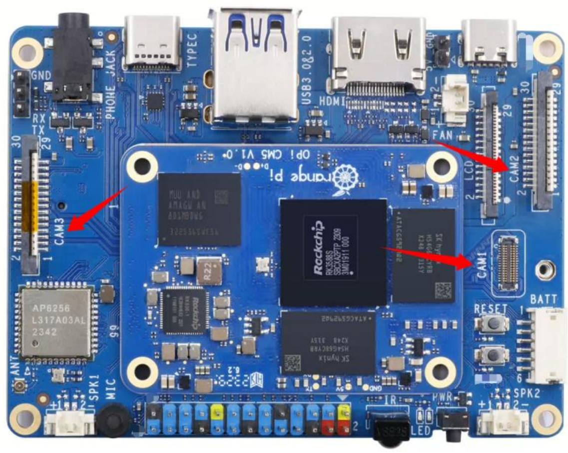
a. **Interface 1 is connected to the OV13850 camera**

b. **Interface 2 connects to the OV13855 camera**

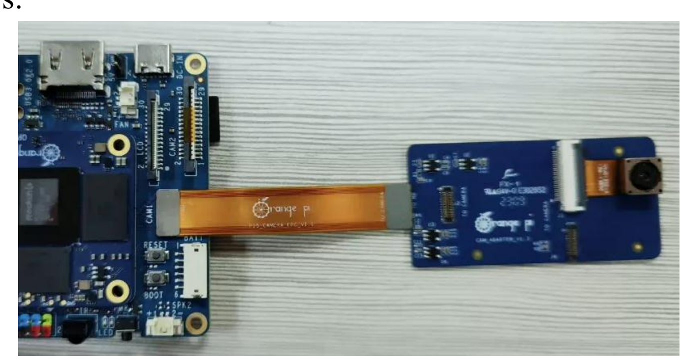
c. Interface 3 is not used, just ignore it



There are 3 camera interfaces on the Orange Pi CM5 Base Tablet development board, and only CAM1 can be used to connect to the OV13850 or OV13855 camera. We define the positions of Cam1, Cam2, and Cam3 as shown in the following figure:



The method of inserting the camera into the Cam1 interface of the development board is as follows:

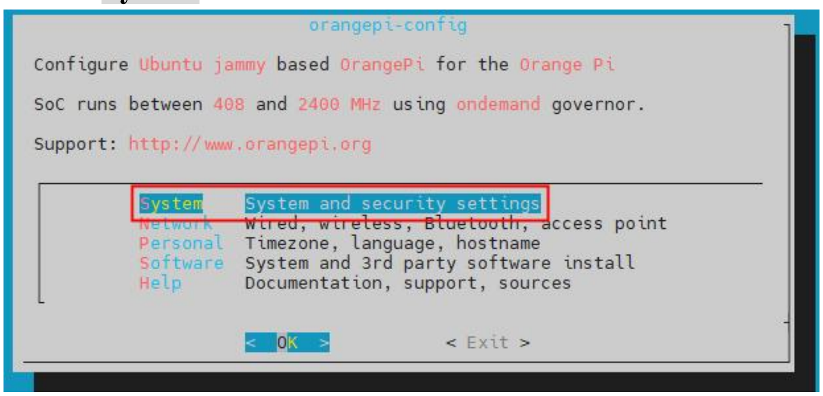


After connecting the camera to the development board, we can use the following method to test the camera:

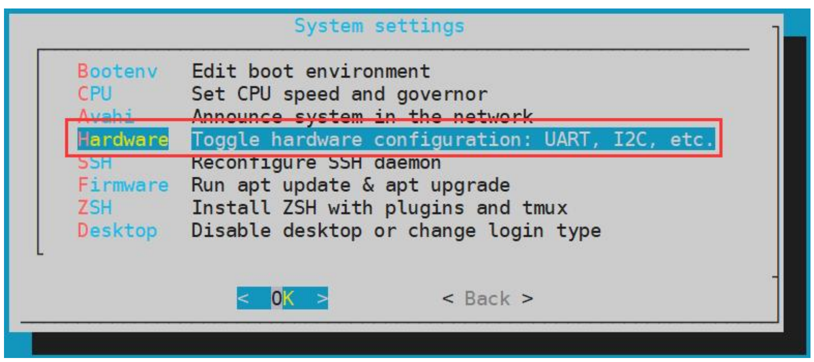
a. First run **orangepi-config.** Remember to add **sudo** permissions as a normal user.

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

b. Then select **System**



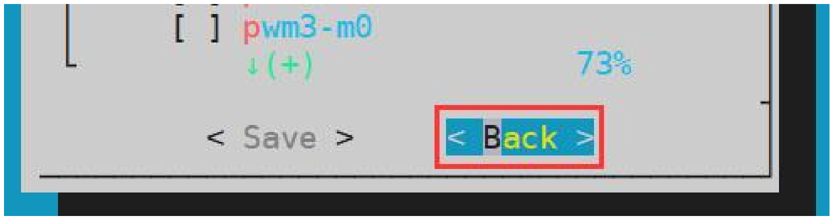
c. Then select **Hardware**



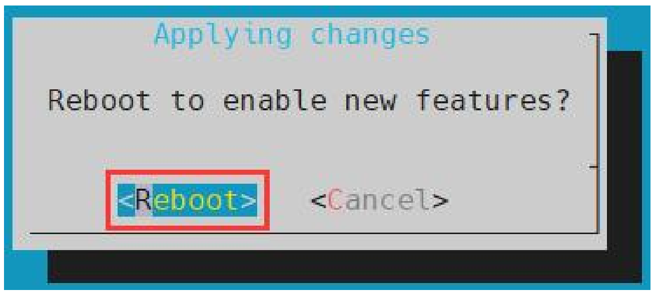
d. Then use the arrow keys on the keyboard to locate the position shown in the figure below, and then use the **space bar** to select the camera you want to open. **opicm5-tablet-cam1** means using the ov13850 or ov13855 camera in the Cam1 interface of the development board.

e. Then select **<Save>** to save.

f. Then select **<Back>**



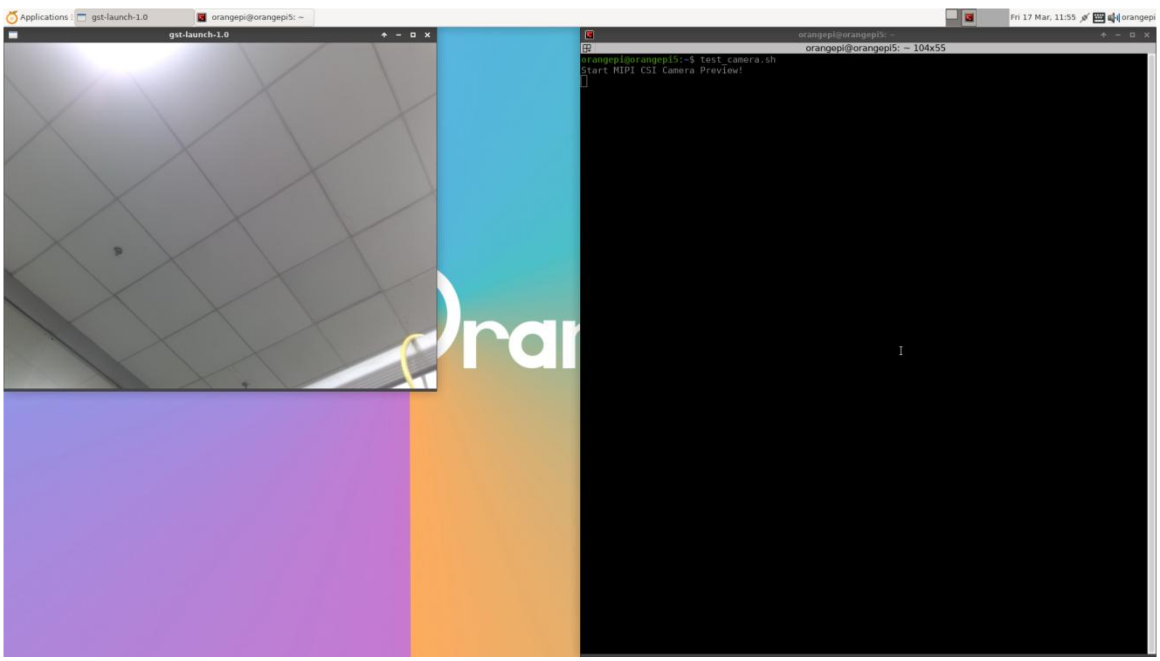
g. Then select **<Reboot>** to restart the system to make the configuration take effect.



h. Then open a terminal in the desktop system and run the following script.

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

i. Then you can see the camera preview.



3. 35. How to use ZFS file system

3. 35. 1. How to install ZFS

Before installing zfs, please make sure that the Linux image you are using is the

latest version. In addition, if zfs has already been installed on the system, there is no need to install it again.

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 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.

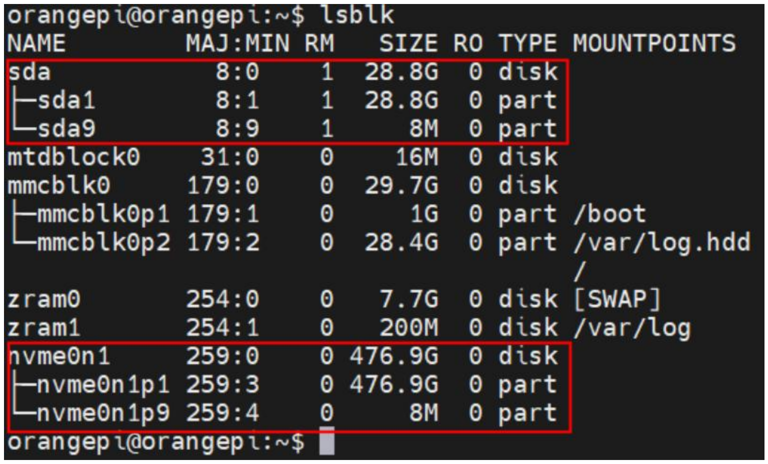
```
orangepi@orangepi:~$ sudo apt install -y zfsutils-linux zfs-dkms
```

3. 35. 2. How to create a ZFS pool

ZFS is based on storage pools. We can add multiple physical storage devices to a pool and then allocate storage space from this pool.

The following content is demonstrated based on the development board connected to an NVMe SSD and a USB flash drive.

1) First, we can use the **lsblk** command to view all storage devices on the development board. The current development board is connected to an NVMe SSD and a USB flash drive. The output is as follows:

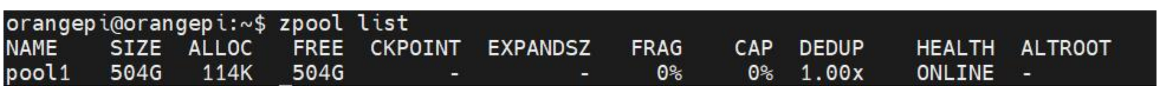


```
orangepi@orangepi:~$ lsblk
NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda             8:0    1  28.8G  0 disk
├─sda1          8:1    1  28.8G  0 part
└─sda9          8:9    1     8M  0 part
mtdblock0      31:0    0    16M  0 disk
mmcblk0       179:0    0  29.7G  0 disk
├─mmcblk0p1   179:1    0     1G  0 part /boot
└─mmcblk0p2   179:2    0  28.4G  0 part /var/log.hdd
                                        /
zram0         254:0    0   7.7G  0 disk [SWAP]
zram1         254:1    0   200M  0 disk /var/log
nvme0n1       259:0    0 476.9G  0 disk
├─nvme0n1p1   259:3    0 476.9G  0 part
└─nvme0n1p9   259:4    0     8M  0 part
orangepi@orangepi:~$
```

2) Then enter the following command to create a ZFS pool containing two storage devices: NVMe SSD and USB flash drive.

orangepi@orangepi:~$ **sudo zpool create -f pool1 /dev/nvme0n1 /dev/sda**

3) Then use the **zpool list** command to see that the system has created a ZFS pool named **pool1**, and the size of the ZFS pool pool1 is the size of the NVME SSD plus the size of the USB flash drive.



```
orangepi@orangepi:~$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G   114K   504G        -         -     0%     0%  1.00x    ONLINE  -
```

4) Then execute **df -h** to see that **pool1** is mounted to the **/pool1** directory.

```
orangepi@orangepi:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           1.6G   18M  1.6G   2% /run
/dev/mmcblk0p2   29G  6.0G   22G  22% /
tmpfs           7.7G   46M  7.7G   1% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           7.7G  944K  7.7G   1% /tmp
/dev/mmcblk0p1 1022M  115M  908M  12% /boot
/dev/zram1      188M  4.5M  169M   3% /var/log
tmpfs           1.6G   80K  1.6G   1% /run/user/1000
pool1           489G  9.3M  489G   1% /pool1
```

5) Use the following command to see that the file system type of pool1 is zfs.

```
orangepi@orangepi:~$ mount | grep pool1
pool1 on /pool1 type zfs (rw,xattr,noacl)
```

6) We can then test copying a file to the ZFS pool.

```
orangepi@orangepi:~$ sudo cp -v /usr/local/test.mp4 /pool1/
'/usr/local/test.mp4' -> '/pool1/test.mp4'
```

3. 35. 3. Test ZFS data deduplication function

1) The data deduplication function of ZFS is disabled by default. We need to execute the following command to enable it.

```
orangepi@orangepi:~$ sudo zfs set dedup=on pool1
```

2) Then do a simple test. First enter pool1 and then execute the following command to generate a random file of 1G.

```
orangepi@orangepi:~$ cd /pool1/
root@orangepi:/pool1$ sudo dd if=/dev/urandom of=test.1g bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 5.04367 s, 213 MB/s
```

3) Then use the following command to copy 1000 copies of a random file of 1G in size.

```
root@orangepi:/pool1$ for ((i=0; i<1000; i++)); do sudo cp test.1g $i.test.1g; done
```

4) Then use **du -lh** to see that there is a total of 1002G of data in the pool. However, in fact, the size of the ZFS pool is only **504GB** (the total capacity of the SSD + USB drive), which cannot accommodate such a large amount of data.

```
root@orangepi:/pool1$ du -lh
1002G
```

5) Then use the **zpool list** command to see that only 1.01G is actually occupied. Because these 1001 files are duplicated, the data deduplication function is effective.

```
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  1.01G   503G        -         -     0%     0%  6.00x    ONLINE  -
```

3. 35. 4. Testing ZFS data compression

1) Because the disk space saved by compression varies depending on the stored data, we choose to compress relatively large plain text files for compression testing and execute

the following command to package the **/var/log/** and **/etc/** directories into a tar package.

```
orangepi@orangepi:~$ cd /pool1/
root@orangepi:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

2) Then, using the **ls -lh** command, you can see that the file size and the space occupied in the ZFS pool are both **27M**.

```
orangepi@orangepi:/pool1$ ls -lh
total 27M
-rw-r--r-- 1 root root 27M Jun  1 14:46 text.tar
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  26.7M   504G        -         -     0%     0%  1.00x    ONLINE  -
orangepi@orangepi:/pool1$
```

3) Then we enable compression in the ZFS pool pool1.

```
root@orangepi:/pool1$ sudo zfs set compression=lz4 pool1
```

4) Then execute the following command again to package the **/var/log/** and **/etc/** directories into a tarball.

```
root@orangepi:/pool1$ sudo tar -cf text.tar /var/log/ /etc/
```

5) At this time, you can see that the size of the **text.tar** file is still 27M, but it only occupies 9.47M of space in the ZFS pool, indicating that the file is compressed.

```
orangepi@orangepi:/pool1$ ls -lh
total 9.2M
-rw-r--r-- 1 root root 27M Jun  1 14:54 text.tar
orangepi@orangepi:/pool1$ zpool list
NAME    SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  ALTROOT
pool1   504G  9.47M   504G        -         -     0%     0%  1.00x    ONLINE  -
```

3. 36. 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.

3. 36. 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.

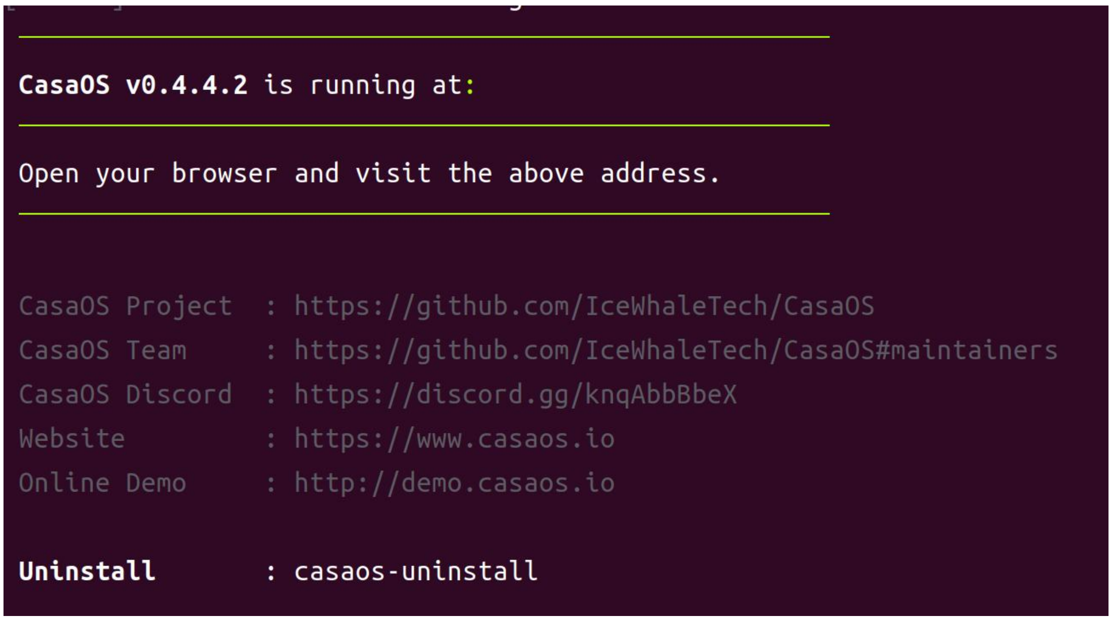
```
orangepi@orangepi:~$ docker --version
```

```
Docker version 27.1.1, build 6312585          # Ubuntu Jammy System Output
```

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
```

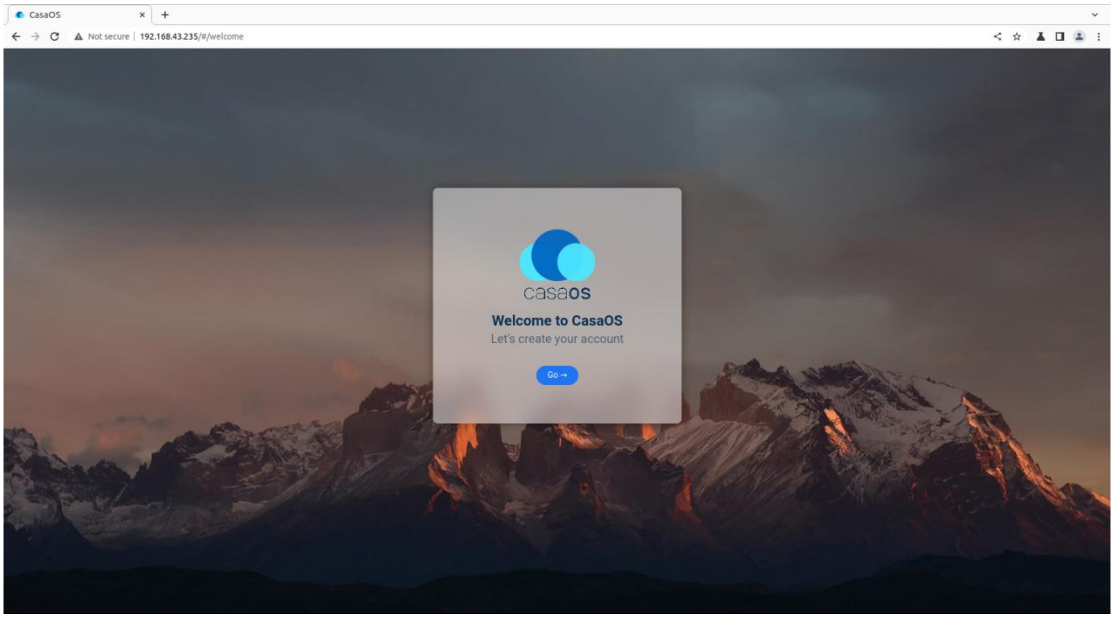
3) When you see the following print information output in the terminal, it means that CasaOS has been installed.



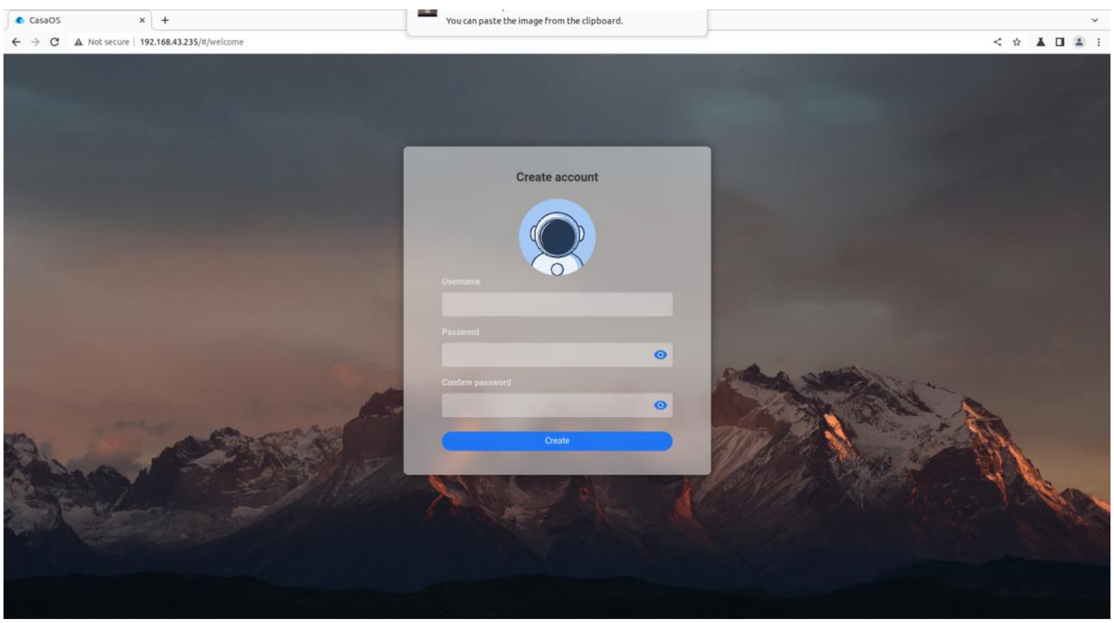
3. 36. 2. How to use CasaOS

1) After installing CasaOS, enter **http://development board's IP address** in the browser to open CasaOS.

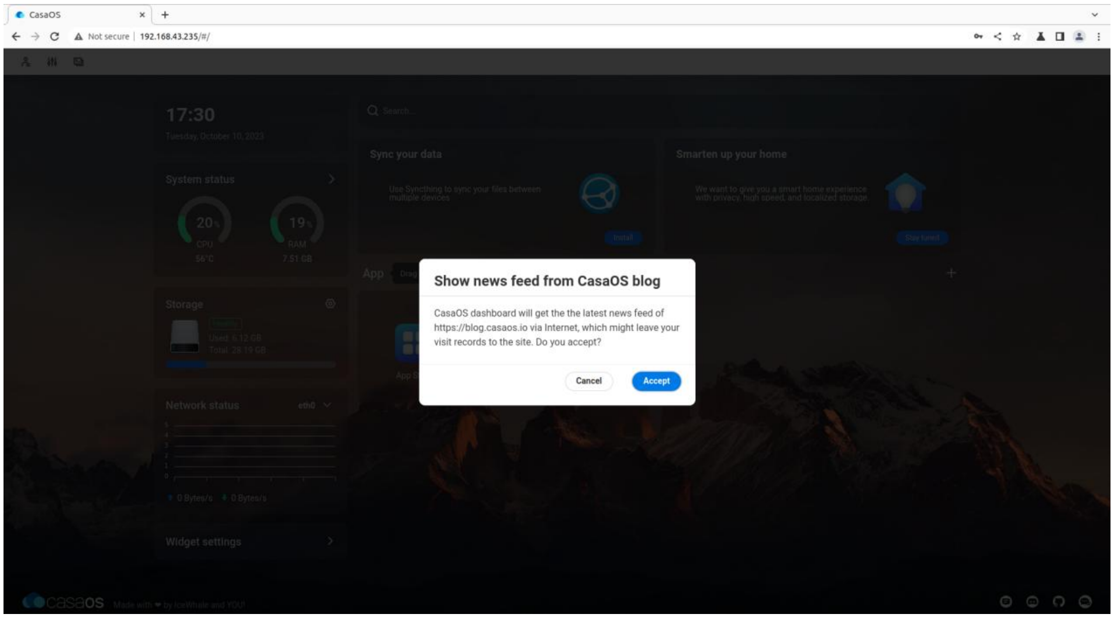
2) After opening CasaO, the welcome interface below will pop up. Click "Go" to proceed to the next step.



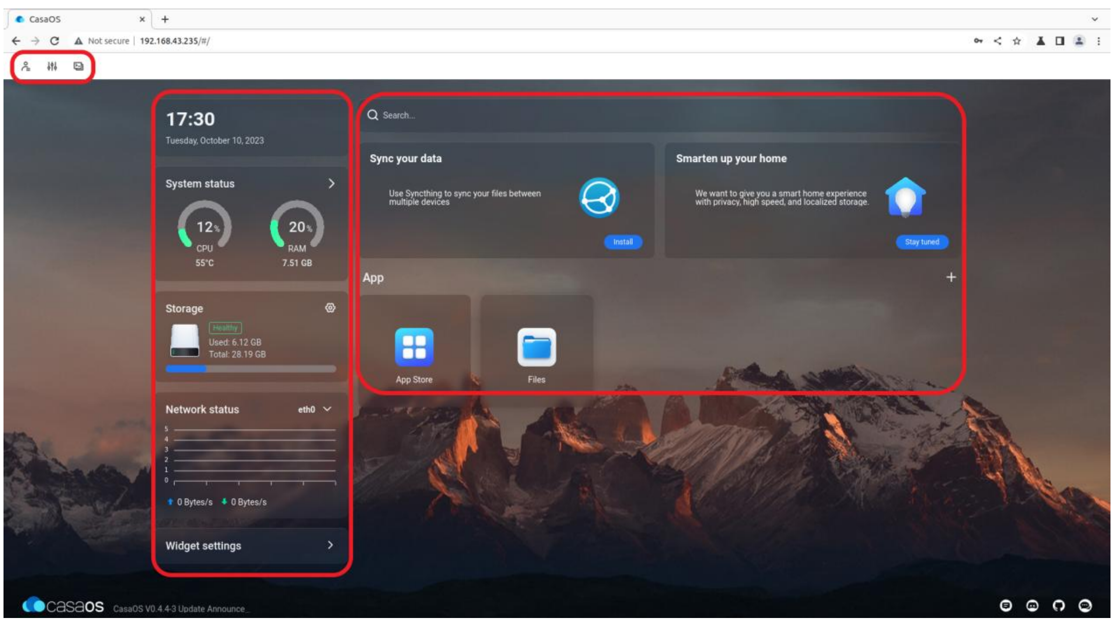
3) When you log in to CasaOS for the first time, the login interface is the interface for setting the account and password. When you log in again in the future, only the interface for entering the account and password will appear. After setting the account and password, click "Create" to proceed to the next step.



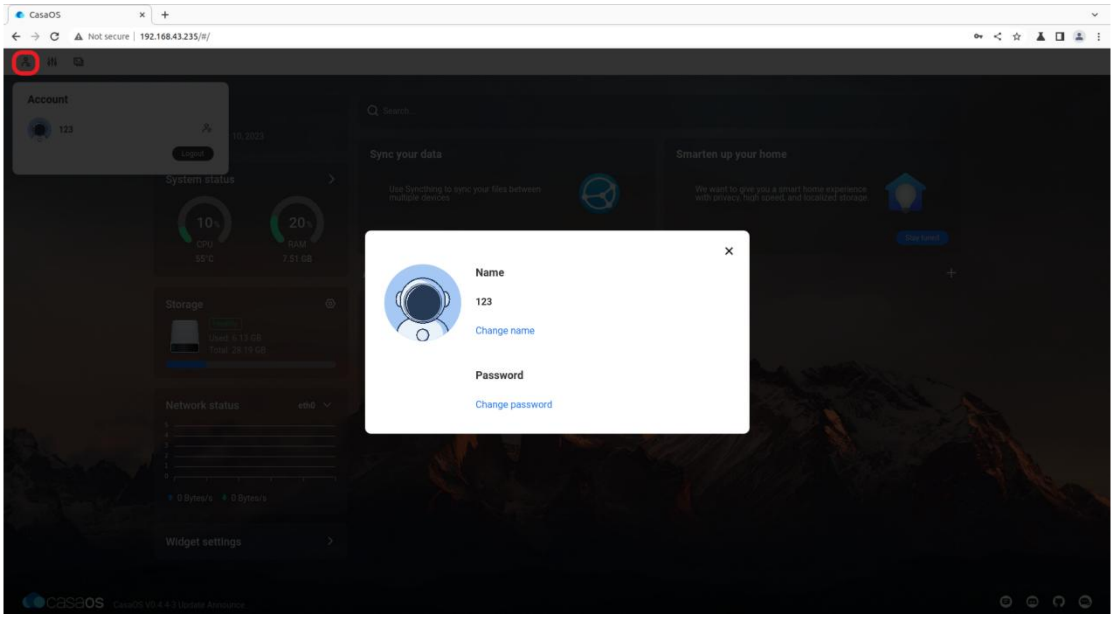
4) In the following interface, just click “Accept” to proceed to the next step.



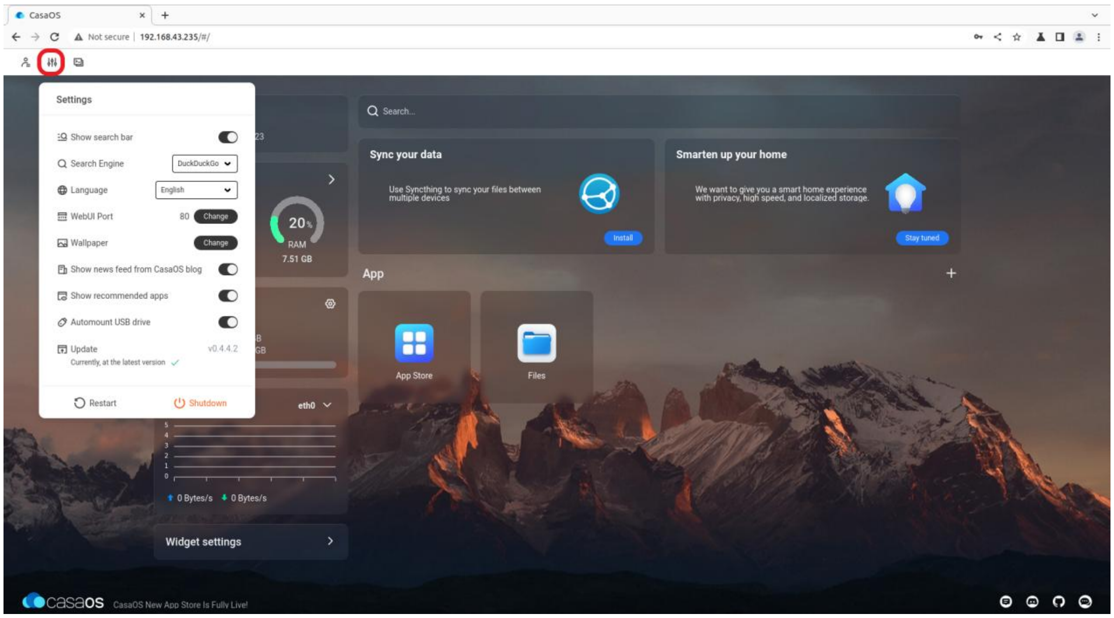
5) Now you have entered the main page of CasaOS. There are three icons in the upper left corner for function settings. On the left is the performance panel, which can display the current time and the status information of CPU, RAM, storage, and network. On the right is the function panel with functions such as search, application recommendation, application store, and file management.



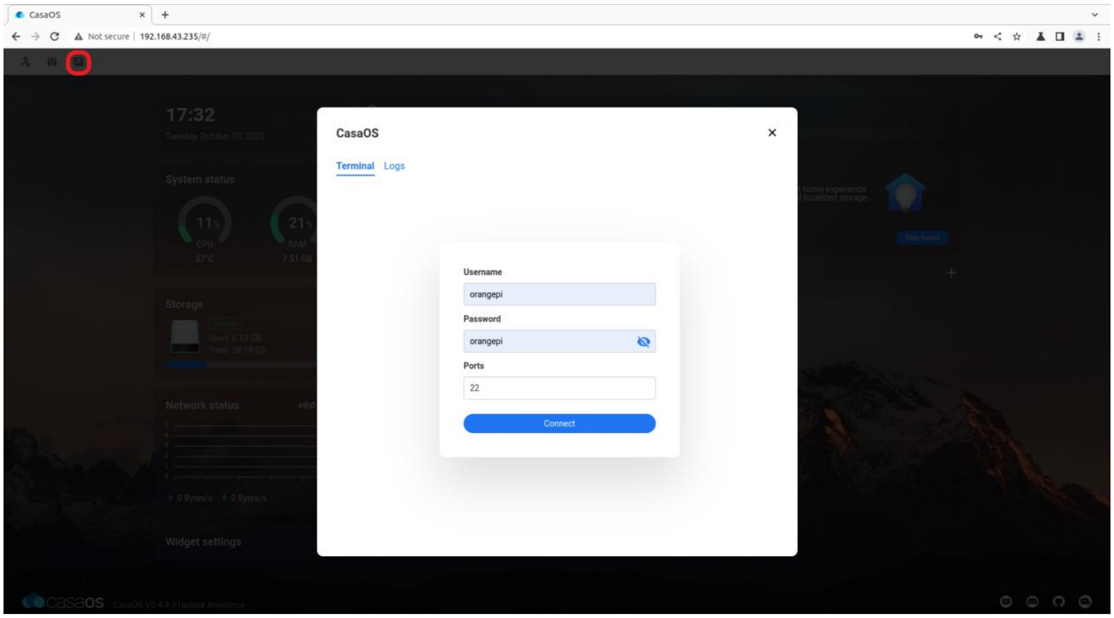
6) You can click the first icon in the upper left corner to modify your account and password.



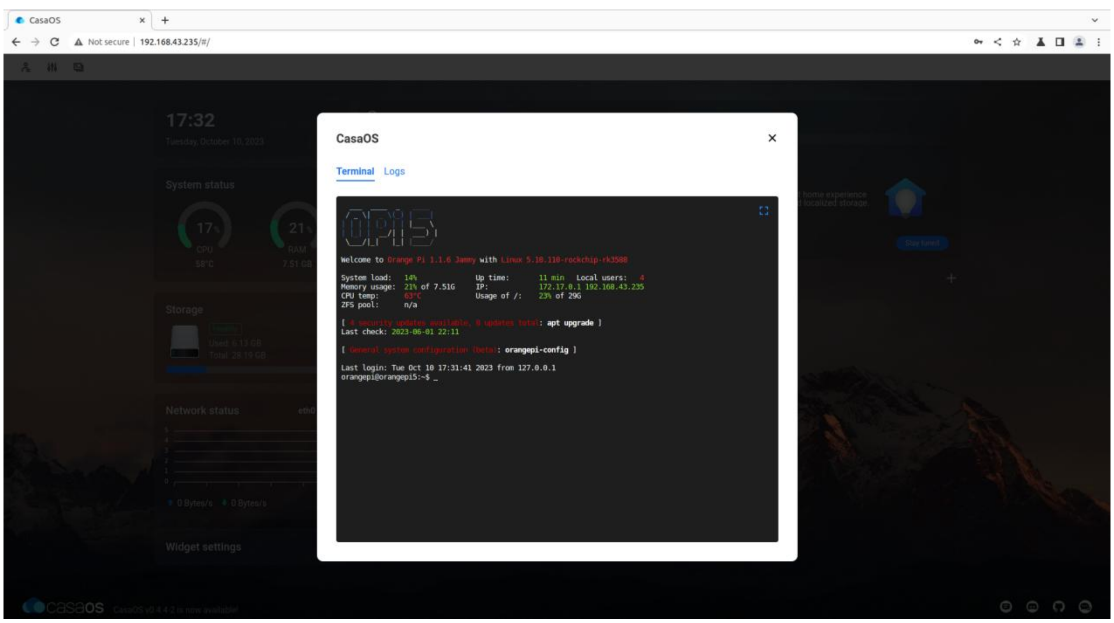
7) You can click the second icon to set basic functions.



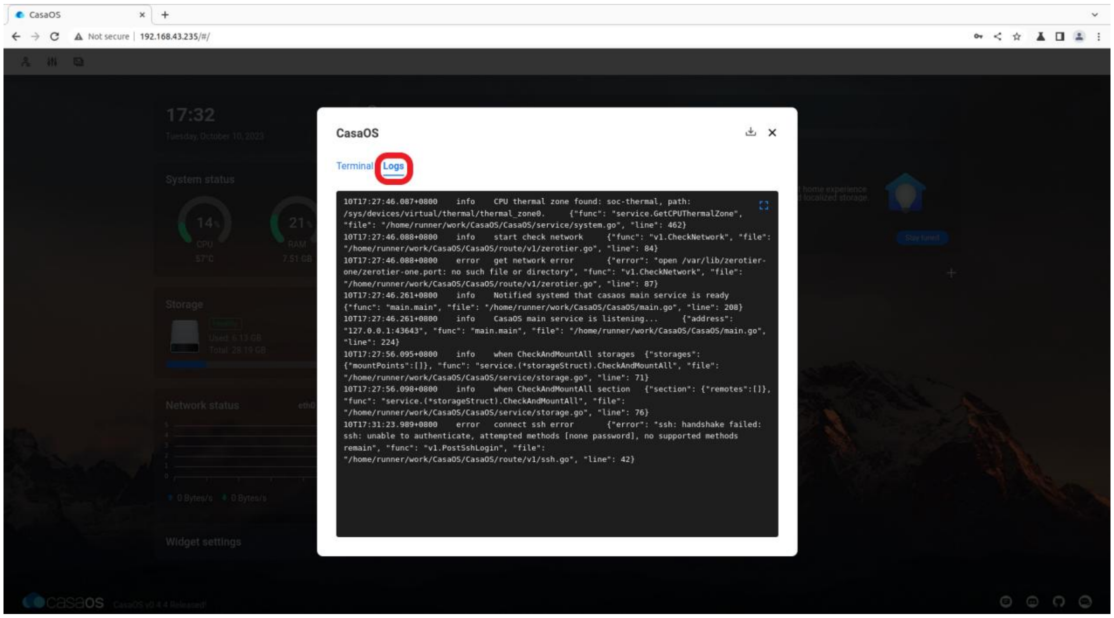
8) The third icon in the upper left corner has two main functions, namely switching to command line mode and printing log information. When switching to command line mode, you need to enter the account and password. The account and password here refer to the account and password of the development board Linux system. The default port system selects 22.



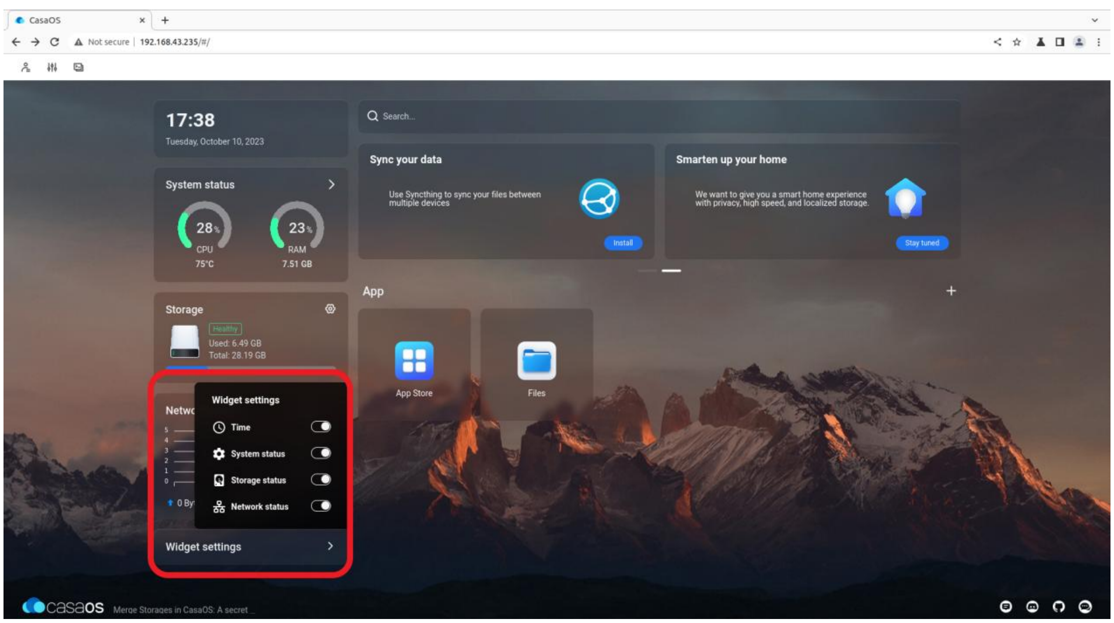
9) Then click "Connect" to enter the command line interface:



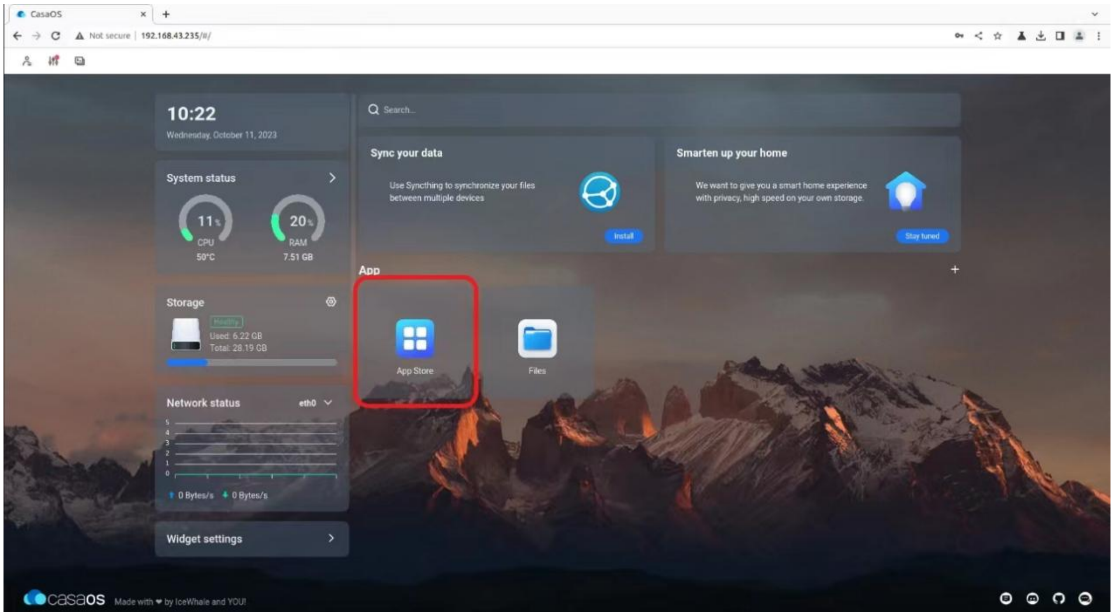
10) Another function under the third icon is to print the CasaOS log. Click "Logs" to enter. The interface is as follows:



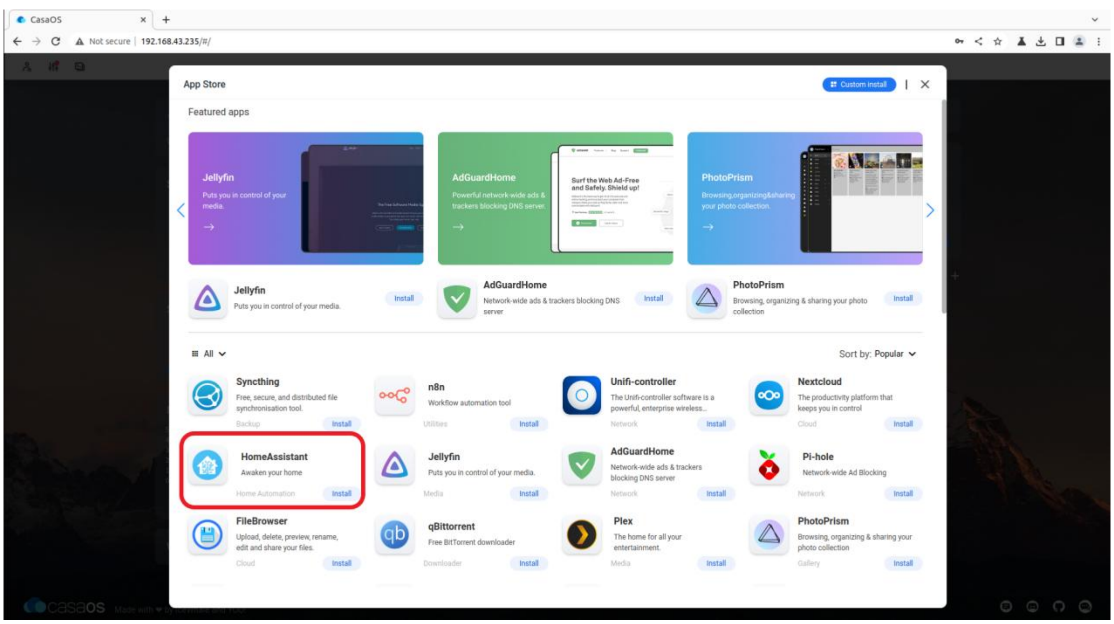
11) Click "Widget settings" in the lower left corner to set whether to display the performance panel widget on the main page.



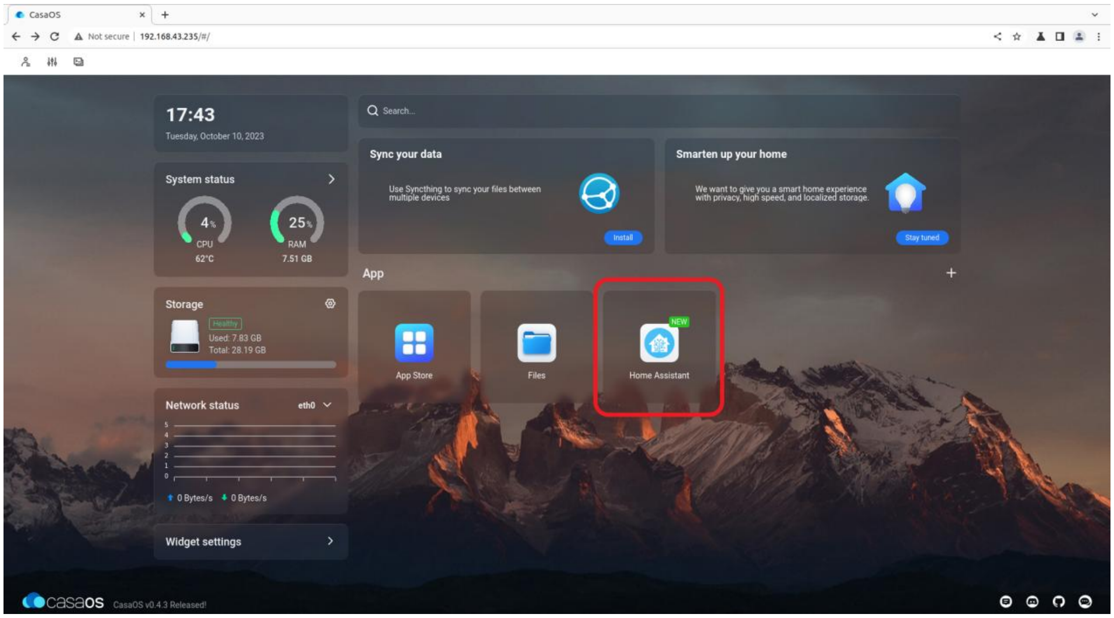
12) Click "APP Store" on the main interface to open the App Store. Currently, there are more than 70 APPs available in the App Store.



13) Here we take Home Assistant as an example for downloading. Find Home Assistant in the APP Store and click the corresponding "install".

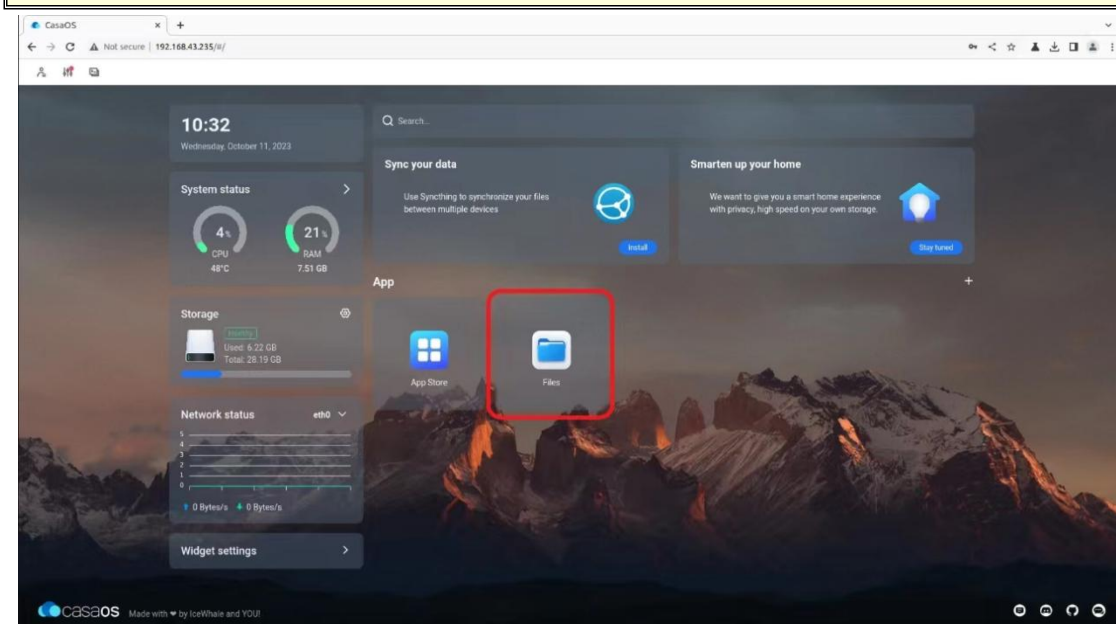


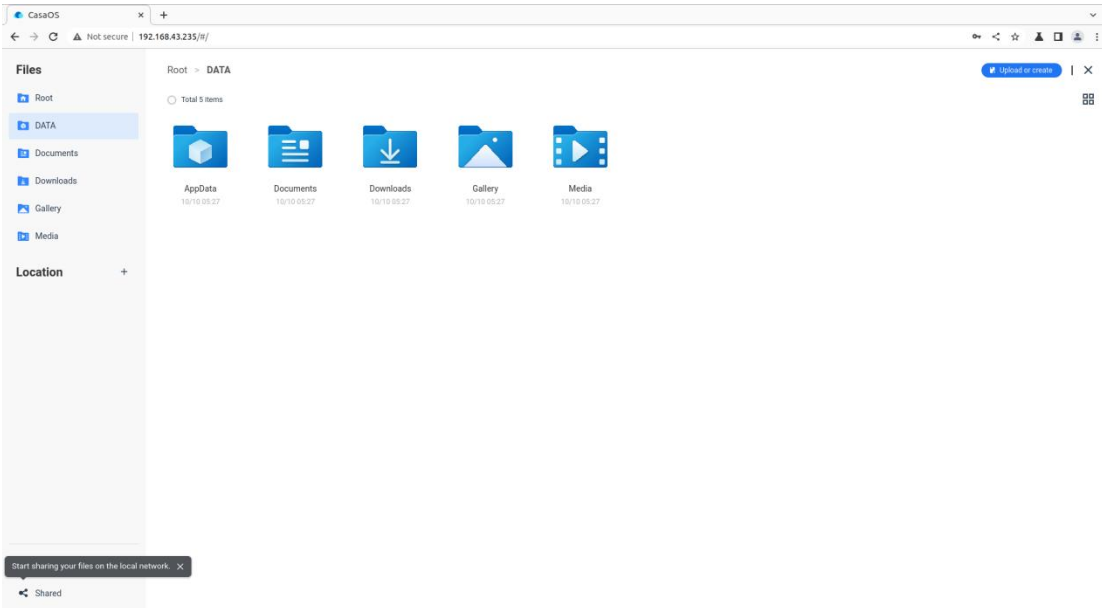
14) After the download is complete, HostAssitant will appear on the main page.



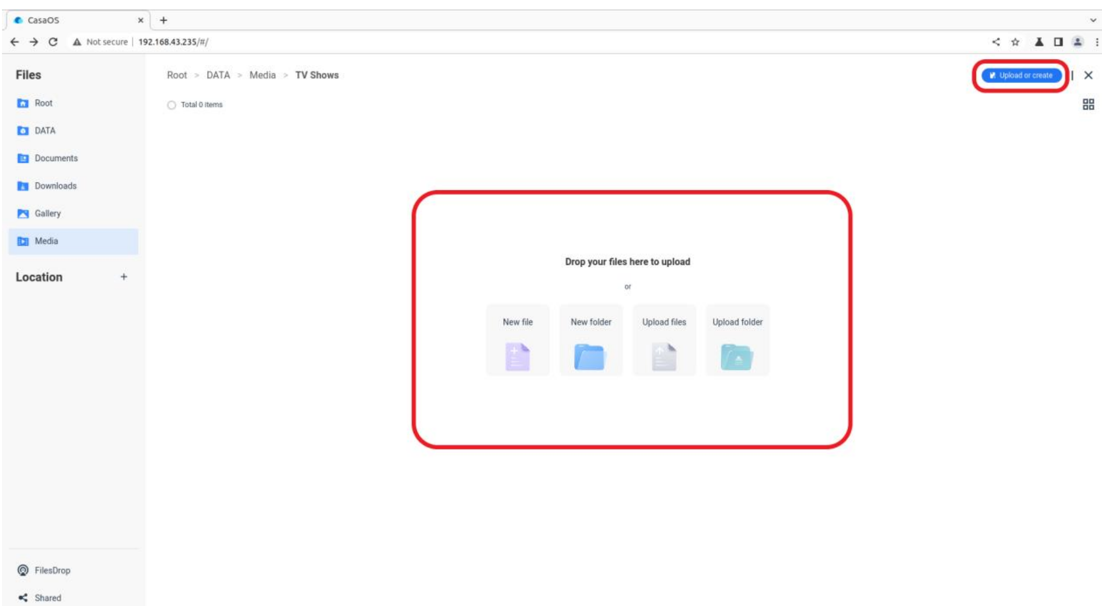
15) Click "Files" in the main interface to open the file system that comes with CasaOS, and then you can upload and save files.

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-tablet:~$ **casaos-uninstall**

3. 37. Methods of using NPU

3. 37. 1. Prepare tools

1) A PC with Ubuntu 20.04 operating system

According to the official documentation of RKNN-Toolkit2, the operating systems supported by the current version of RKNN-Toolkit2 are as follows:

a. Ubuntu18.04 (x64)

b. Ubuntu20.04 (x64)

c. Ubuntu22.04 (x64)

In this document, we use the Ubuntu 20.04 (x64) operating system for demonstration. Please test other versions of the operating system yourself.

2) An Orange Pi development board with Debian 11 installed

3) A Type-C data cable for using the adb function

3. 37. 2. Install RKNN-Toolkit2 on Ubuntu PC

Toolkit2 is a development kit used on the Ubuntu PC platform. Users can use the Python interface provided by the tool to easily complete functions such as model conversion, reasoning, and performance evaluation.

1) On the Ubuntu PC, open a command line window and enter the following commands to install python3 and pip3

```
test@test:~$ sudo apt-get install python3 python3-dev python3-pip
```

2) You can use the following command to view the installed version of python3

```
test@test:~$ python3 --version
```

```
Python 3.8.10
```

3) Then enter the following command to install the dependency package of RKNN-Toolkit2

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install libxslt1-dev zlib1g-dev libglib2.0 \
 libsm6 libgl1-mesa-glx libprotobuf-dev gcc
```

4) Then enter the following command to download the 1.5.2 version of RKNN-Toolkit2

```
test@test:~$ git clone https://github.com/airockchip/rknn-toolkit2 -b v1.5.2
```

5) Then enter the following command to install the corresponding version of Python3 dependency packages. This command will use pip3 to install the dependencies listed in the file requirements_cp38-1.5.2.txt. If the dependencies are not fully installed, 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 software package. After the installation is complete, you can use RKNN-Toolkit2

```
test@test:~$ pip3 install rknn-toolkit2/packages/rknn_toolkit2-1.5.2+b642f30c-cp38-cp38-linux_x86_64.whl
```

3. 37. 3. Model conversion and model inference using RKNN-Toolkit2

RKNN-Toolkit2 supports converting Caffe, TensorFlow, TensorFlow Lite, ONNX, DarkNet, PyTorch and other models into RKNN models, and then ru nning the RKNN model on the Ubuntu PC through simulation or using the N PU of the development board for inference.

Relevant examples are provided in the example folder of RKNN-Toolkit2 to help users better understand how to operate. We take the ONNX model wi th yolov5 function as an example.

3. 37. 3. 1. Simulate the model on Ubuntu PC

RKNN-Toolkit2 is equipped with a built-in simulator, which allows users t o simulate the inference process of the model on Rockchip NPU on Ubuntu P C.

In this way, model conversion and inference can be completed on the Ubuntu PC, helping users test and verify their models faster.

1) First switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

2) Then run the test.py script, which first converts the yolov5s_relu.onnx model into an RKNN model that can be run on the simulator, and then uses the simulator to simulate and run the model to perform inference on the bus.jpg image in the current directory.

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

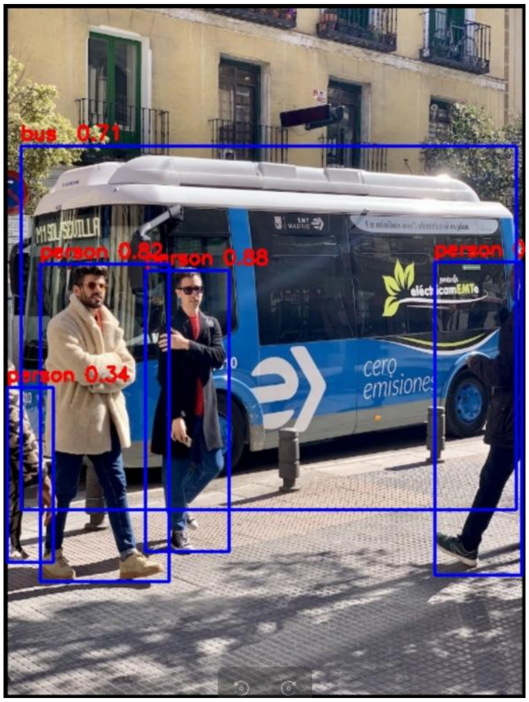
3) After the test.py script runs successfully, you will see the following print information, indicating that the model successfully detected four people and a bus in the bus.jpg image

```
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 inference picture 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. 37. 3. 2. Using the NPU of the development board to run the model on Ubuntu PC

RKNN-Toolkit2 provides users with a Python interface for using the development board's NPU for reasoning through adb, allowing users to use the development board's NPU to run models for reasoning on an Ubuntu PC.

In this way, the Ubuntu PC can use the machine learning library provided by Python to optimize and adjust the model according to the actual effect of the model running on the NPU of the development board.

3. 37. 3. 2. 1. Connect adb using Type-C cable

Use adb to operate the development board on the Ubuntu PC. For the usage of adb, please refer to the instructions in the section **How to use ADB**.

3. 37. 3. 2. 2. Update rknn_server and librknnrt.so of the development board

librknnrt.so is a board-side runtime library.

rknn_server is a background proxy service running on the development board. It is used to receive the protocol transmitted from the PC via USB, then execute the corresponding interface in the board runtime library and return the result to the PC.

1) First, enter the following command on the Ubuntu PC to download the 1.5.2 version of RKNPU2

```
test@test:~$ git clone https://github.com/rockchip-linux/rknpu2 -b v1.5.2
```

2) Then enter the following command on the Ubuntu PC to update the rknn_server of the development board through the adb tool

```
test@test:~$ adb push rknpu2/runtime/RK3588/Linux/rknn_server/aarch64/usr/bin/* /usr/bin
```

3) Then enter the following command on the Ubuntu PC to update the librknnrt.so library of the development board through the 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 of 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 has been opened, and the operating environment of the development board has been set up.

```
root@orangepi:/# pgrep rknn_server
```

```
3131
```

3. 37. 3. 2. 3. Modify the parameters in the example

1) On the Ubuntu PC, you can use the following command to view the device ID of the development board connected to the Ubuntu PC. This ID will be used below.

```
test@test:~$ adb devices
List of devices attached
4f9f859e5a120324    device
```

2) Switch to the rknn-toolkit2/examples/onnx/yolov5 directory

```
test@test:~$ cd rknn-toolkit2/examples/onnx/yolov5/
```

3) Use vim editor to modify the test.py file

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ vim test.py
```

4) In the test.py file, we need to modify the following content:

a. In the preprocessing configuration, change the target platform to rk3588, so that the model conversion results in an RKNN model suitable for the NPU of 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 the description of the target platform and device ID. The target platform is rk3588, and the device ID is the device ID of the development board obtained through adb. The operation of running the model for inference will be 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 the modification is completed, save and exit

3. 37. 3. 2. 4. Run the example on Ubuntu PC

1) Enter the following command to run the test.py script. The script first converts the yolov5s_relu.onnx model to the RKNN model, and then loads the model to the NPU of the development board to perform inference on the out.jpg image in the current directory.

```
test@test:~/rknn-toolkit2/examples/onnx/yolov5$ python3 test.py
```

2) In the printed information, we can see that the Ubuntu PC uses the NPU of the development board to run the model for inference through the adb tool

```
--> Init runtime environment
I target set by user is: rk3588
I Check RK3588 board npu runtime version
I Starting ntp or adb, target is RK3588
I Device [4f9f859e5a120324] not found in ntb device list.
I Start adb...
I Connect to Device success!
I NPUTransfer: Starting NPU Transfer Client, Transfer version 2.1.0
(b5861e7@2020-11-23T11:50:36)
```

3) After the test.py script runs successfully, the converted model file yolov5s_relu.rknn and the inference image result result.jpg are saved in the current directory

4) The results of the operation are the same as those in the section **Simulating the model on the Ubuntu PC**.

3. 37. 4. Call the C interface to deploy the RKNN model to the development board to run

RKNPU2 provides a C programming interface for chip platforms with Rockchip NPU, which can help users deploy RKNN models exported using RKNN-Toolkit2 and accelerate the implementation of AI applications.

In the example folder of RKNPU2, examples of deploying RKNN models with different functions to the development board are provided. We take the deployment of the RKNN model with yolov5 function to the RK3588 Debian 11 platform as an example.

3. 37. 4. 1. Download cross-compilation tools

Since the development board runs on Linux, you need to use the gcc cross compiler to compile. It is recommended to use the gcc version of gcc-9.3.0-x86_64_arrch64-linux-gnu

Enter the following command to download this version of gcc. After downloading, you will get a folder named gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu

```
test@test:~$ git clone https://github.com/airockchip/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
```

3. 37. 4. 2. Modify the compiler tool path in the script

1) Switch to the rknpu2/examples/rknn_yolov5_demo directory

```
test@test:~$ cd ~/rknpu2/examples/rknn_yolov5_demo
```

2) Use the vim editor to modify the contents of the build-linux_RK3588.sh file.

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ vim build-linux_RK3588.sh
```

3) In the build-linux_RK3588.sh file, we need to change the value of the variable TOOL_CHAIN to the path of the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder. In this way, when running the build-android_RK3588.sh script, the cross-compilation tool in the gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu folder will be used for compilation

```
TARGET_SOC="rk3588"
GCC_COMPILER=aarch64-linux-gnu

export TOOL_CHAIN=~/gcc-buildroot-9.3.0-2020.03-x86_64_aarch64-rockchip-linux-gnu
export LD_LIBRARY_PATH=${TOOL_CHAIN}/lib64:$LD_LIBRARY_PATH
export CC=${GCC_COMPILER}-gcc
export CXX=${GCC_COMPILER}-g++
```

4) After modification, save and exit

3. 37. 4. 3. Compile rknn_yolov5_demo

1) Run build-linux_RK3588.sh, which generates a program suitable for the RK3588 development board through cross-compilation and can run the RKNN model for inference on it

```
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, there will be an additional folder named install in the current directory. The rknn_yoov5_demo_Linux folder under this folder contains the program generated by cross-compilation and its related files

```
test@test:~/rknpu2/examples/rknn_yolov5_demo$ ls install
rknn_yolov5_demo_Linux
```

3. 37. 4. 4. Deploy rknn_yolov5_demo to the development board

On the Ubuntu PC, you can use the following command to upload the rknn_ yolov5_demo_Linux folder to the development board through the adb tool to depl oy 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
```

3.37.4.5. Run rknn_yolov5_demo on the development board

1) Enter the file system of the development board through adb shell on the 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 to perform inference. In the following command, the program uses the yolov5s-640-640.rknn model to perform inference on the bus.jpg image. The entire running process will be completed on the development board

```
root@orangepi:/data/rknn_yolov5_demo_Linux# ./rknn_yolov5_demo \
./model/RK3588/yolov5s-640-640.rknn ./model/bus.jpg
```

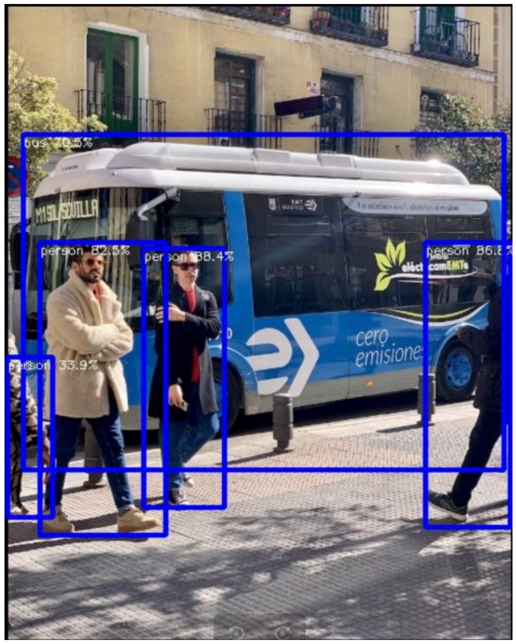
4) After the run is completed, the inference result out.jpg image is saved in the current directory

```
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 the out.jpg image using the adb tool, and then view it using an image viewer:

```
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. 38. How RK3588 uses Baidu Feijiang

Use Baidu Feijiang on the rk3588 development board, including converting the pdmodel model to the rknn model on the PC side and deploying the rknn model using the FastDeploy deployment tool developed by Baidu Feijiang on the board side. The following content is implemented in an environment where the PC system is Ubuntu22.04 and the board system is Debian 11. Please test other environments yourself.

3. 38. 1. Ubuntu PC environment setup

The tools and uses that need to be installed on the Ubuntu PC are as follows

Tool Name	Use
Anaconda3	Used to create and manage Python environments
Paddle2ONNX	Used to convert the pdmodel model to the ONNX model
RKNN-Toolkit2	Used to convert ONNX model to RKNN model

3. 38. 1. 1. Install Anaconda3 on PC

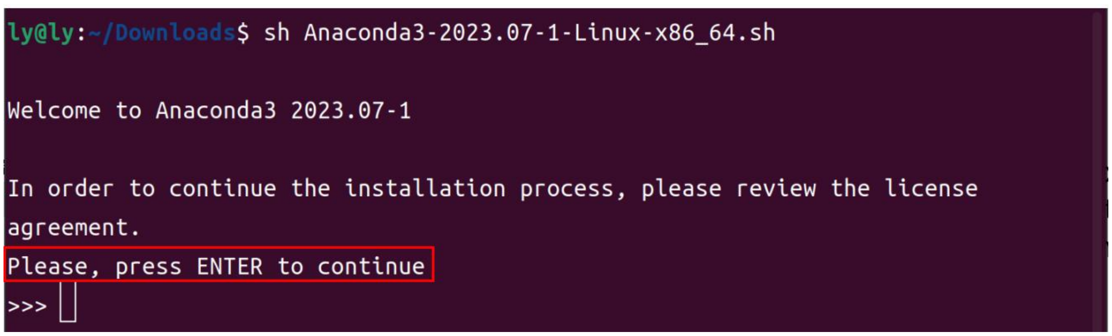
1）Open the browser on the Ubuntu PC and enter the following URL in the address bar to download and install the Anaconda3 script. After the download is complete, you will get the **Anaconda3-2023.07-1-Linux-x86_64.sh** file

https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-x86_64.sh

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）The installation script will then output the following prompt message, at this time click Enter to continue the installation



4）After pressing the Enter key, some introduction information about Anaconda3 will appear. Keep pressing 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）The installation script will then prompt you to accept the license terms. Enter yes and press Enter.

```
The following packages listed on https://www.anaconda.com/cryptography are inclu
ded in the repository accessible through Anaconda Distribution that relate to cr
yptography.

Last updated February 25, 2022


Do you accept the license terms? [yes|no]
[no] >>>
```

6）The installation script will then prompt 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 you see the following print in the terminal, it means 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. 38. 1. 2. Install RKNN-Toolkit2 on PC

1）Open the terminal on the Ubuntu PC and use the Anaconda3 tool to create a Python 3.8 environment

(base)test@test:~$ **conda create -n fastdeploy python=3.8**

2）Activate the python3.8 environment just created

(base)test@test:~$ **conda activate fastdeploy**

3）Then install pip3 development tools and package management tools

(fastdeploy)test@test:~$ **sudo apt-get install python3-dev python3-pip**

4）Then install the dependency package of 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 you need to install numpy==1.16.6 first

```
(fastdeploy)test@test:~$ pip install numpy==1.16.6
```

6）Install git tool

```
(fastdeploy)test@test:~$ sudo apt install git
```

7）Then execute the following command to download RKNN-Toolkit2. After the download is complete, you will get the rknn-toolkit2 folder

```
(fastdeploy)test@test:~$ git clone https://github.com/rockchip-linux/rknn-toolkit2
```

8）Then execute the following command to install RKNN-Toolkit2 corresponding to python3.8 version

```
(fastdeploy)test@test:~$ pip install rknn-toolkit2/rknn-toolkit2/packages/rknn_toolkit2-1.6.0+81f21f4d-cp38-cp38-linux_x86_64.whl
```

3. 38. 1. 3. Install Paddle2ONNX on PC

You can execute the following command to install paddlepaddle and paddle2onnx

```
(fastdeploy)test@test:~$ pip install paddlepaddle
(fastdeploy)test@test:~$ pip install paddle2onnx
```

3. 38. 2. Board environment construction

The tools and uses that need to be installed on the board are as follows

Tool Name	use
Anaconda3	Used to create and manage Python environments
rknpu2	Basic driver for rknpu2
FastDeploy	After compilation, you can get the FastDeploy inference library

3. 38. 2. 1. Install Anaconda3 on the board

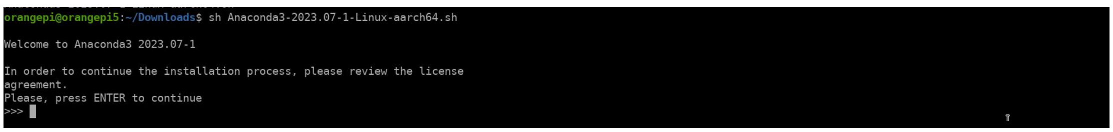
1）Open the browser on the board and enter the following URL in the address bar to download and install the Anaconda3 script. After the download is complete, you will get the **Anaconda3-2023.07-1-Linux-aarch64.sh** file

```
https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2023.07-1-Linux-aarch64.sh
```

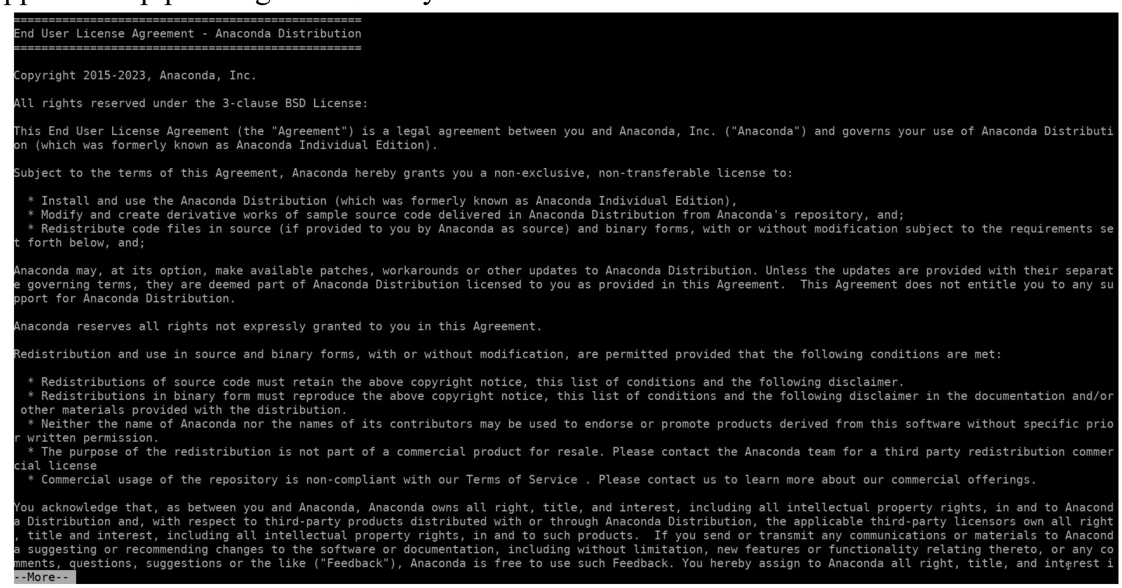
2）Open the terminal and run the **Anaconda3-2023.07-1-Linux-aarch64.sh** script to install Anaconda3

```
orangepi@orangepi:~/Downloads$ sh Anaconda3-2023.07-1-Linux-aarch64.sh
```

3）Then the installation script will output the following prompt information, click Enter to continue the installation



4）After pressing the Enter key, some introduction information about Anaconda3 will appear. Keep pressing the " ↓ " key.



5）The installation script will then prompt you to accept the license terms. Enter yes and press Enter.

```
The following packages listed on https://www.anaconda.com/cryptography are included in the repository accessible through Anaconda Distribution that relate to
cryptography.

Last updated February 25, 2022


Do you accept the license terms? [yes|no]
[no] >>>
```

6）The installation script will then prompt 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 you see the following print in the terminal, it means 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 in the terminal and it says the command does not exist, you need to modify the ~/.bashrc file

orangepi@orangepi:~$ **vi ~/.bashrc**

10）Add the following code to the end of the ~/.bashrc file

```
export PATH=/home/orangepi/anaconda3/bin:$PATH
```

11）Then enter the following command in the terminal to make the changes take effect

orangepi@orangepi:~$ **source ~/.bashrc**

12）Then enter the following command in the terminal to initialize conda

(base)orangepi@orangepi:~$ **conda init bash**

13）Then close the current terminal and reopen a new terminal. You can now use the conda command normally.

3. 38. 2. 2. Install rknpu2 driver on the board

1）Open the terminal on the board and use the Anaconda3 tool to create a Python version 3.9 environment

```
(base)orangepi@orangepi:~$ conda create -n fastdeploy python=3.9
```

2）Activate the python3.9 environment just created

```
(base)orangepi@orangepi:~$ conda activate fastdeploy
```

3）Download the rknpu2_device_install_1.4.0.zip file via wget

```
(fastdeploy)orangepi@orangepi:~$ wget https://bj.bcebos.com/fastdeploy/third_libs/rknpu2_device_install_1.4.0.zip
```

4）Then execute the following command to decompress rknpu2_device_install_1.4.0.zip. After decompression, you will get the rknpu2_device_install_1.4.0 folder and the __MACOSX folder

```
(fastdeploy)orangepi@orangepi:~$ unzip rknpu2_device_install_1.4.0.zip
```

5）Switch to the rknpu2_device_install_1.4.0 directory

```
(fastdeploy)orangepi@orangepi:~$ cd rknpu2_device_install_1.4.0/
```

6）There is a rknn_install_rk3588.sh script in this directory. Run the script to complete the installation of the board-side rknpu2 driver.

```
(fastdeploy)orangepi@orangepi:~/rknpu2_device_install_1.4.0$ sudo bash rknn_install_rk3588.sh
```

3. 38. 2. 3. Compile FastDeploy C++ SDK on the board

1）The cmake command is needed when compiling. You can execute the following command to install the cmake tool

```
(fastdeploy)orangepi@orangepi:~$ sudo apt-get install -y cmake
```

2）Then download the FastDeploy SDK. After the command is executed, you will get the FastDeploy folder

```
(fastdeploy)orangepi@orangepi:~$ git clone https://github.com/PaddlePaddle/FastD
```

```
eploy.git
```

3）Switch to the FastDeploy directory

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy
```

4）Create a compilation directory build and switch to the build directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy$ mkdir build && cd build
```

5）Before compiling, you need to use cmake to configure the project information to be compiled. After executing the following command, there will be some more files in the current directory, including the Makefile file used for compilation

```
(fastdeploy)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

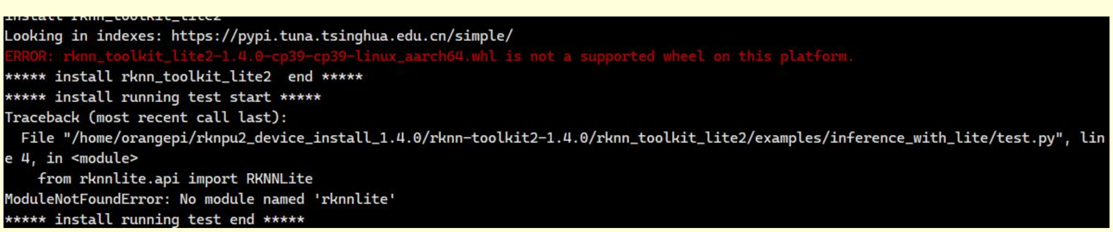
```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make -j8
```

7）After the compilation is complete, use the following command to install the compiled files to the specified path

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ make install
```

8）After the compilation is completed, you will get the fastdeploy-0.0.3 folder. In this folder, there is a script file fastdeploy_init.sh for configuring environment variables. After using this script to configure environment variables, you can use some compiled library files.

There may be errors as shown in the following pictures. Just ignore them. They will not affect subsequent operations.



```
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple/
ERROR: rknn_toolkit_lite2-1.4.0-cp39-cp39-linux_aarch64.whl is not a supported wheel on this platform.
***** install rknn_toolkit_lite2  end *****
***** install running test start *****
Traceback (most recent call last):
  File "/home/orangepi/rknpu2_device_install_1.4.0/rknn-toolkit2-1.4.0/rknn_toolkit_lite2/examples/inference_with_lite/test.py", lin
e 4, in <module>
    from rknnlite.api import RKNNLite
ModuleNotFoundError: No module named 'rknnlite'
***** install running test end *****
```

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/build$ source fastdeploy-0.0.3/fastdeploy_init.sh
```

3. 38. 3. Example of deploying a model using FastDeploy

The ResNet50_vd model is a model used for target classification. The following uses the ResNet50_vd model as an example to illustrate the process of using FastDeploy to deploy the pdmodel model.

3. 38. 3. 1. Ubuntu PC model conversion

1）Open the terminal on the PC and activate the python3.8 environment created using Anaconda3

```
test@test:~$ conda activate fastdeploy
```

2）In the model conversion script, you need to import the yaml module and the six module. You can execute the following command to install them.

```
(fastdeploy)test@test:~$ pip install pyyaml six
```

3）Execute the following command to download the ResNet50_vd_infer.tgz file

```
(fastdeploy)test@test:~$ wget https://bj.bcebos.com/paddlehub/fastdeploy/ResNet50_vd_infer.tgz
```

4）After decompressing the ResNet50_vd_infer.tgz file, you can get the ResNet50_vd_infer folder, which contains the pdmodel model file inference.pdmodel and other related files.

```
(fastdeploy)test@test:~$ tar -xvf ResNet50_vd_infer.tgz
```

5）You can use the following command to convert the pdmodel model to an onnx model through paddle2onnx. After executing the command, there will be an additional onnx model file ResNet50_vd_infer.onnx in the ResNet50_vd_infer folder.

```
(fastdeploy)test@test:~$ paddle2onnx --model_dir ResNet50_vd_infer \
--model_filename inference.pdmodel \
--params_filename inference.pdiparams \
--save_file ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--opset_version 10 \
--enable_onnx_checker True
```

6）Then use the following command to fix the shape to [1,3,224,224]. After executing the command, the ResNet50_vd_infer.onnx file will be modified.

```
(fastdeploy)test@test:~$ python -m paddle2onnx.optimize --input_model \
ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--output_model ResNet50_vd_infer/ResNet50_vd_infer.onnx \
--input_shape_dict "{'inputs':[1,3,224,224]}"
```

7）To convert the onnx model to the rknn model, you need to use the script in the FastDeploy SDK. Execute the following command to download FastDeploy

```
(fastdeploy)test@test:~$ git clone https://github.com/PaddlePaddle/FastDeploy.git
```

8）Then transfer the ResNet50_vd_infer folder to the corresponding directory of FastDeploy

```
(fastdeploy)test@test:~$ mv ResNet50_vd_infer \
FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

9）Switch to the directory where the model conversion is performed

```
(fastdeploy)test@test:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/
```

10）Execute the following command to convert the onnx model to the rknn model, and finally get the rknn model file in the ResNet50_vd_infer directory ResNet50_vd_infer_rk3588_unquantized.rknn

```
(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, the rknn model file used is named ResNet50_vd_infer_rk3588.rknn, so you need 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. 38. 3. 2. Board-side model deployment

1）Open the terminal on the board and activate the python3.9 environment created previously using Anaconda3

```
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 sample directory for deploying the ResNet50 model in FastDeploy

```
(fastdeploy)orangepi@orangepi:~$ cd FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp
```

4）Create a directory structure under this directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ mkdir build images ppclas_model_dir thirdpartys
```

5）Copy the compiled fastdeploy-0.0.3 folder to the thirdpartys folder

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cp -r ~/FastDeploy/build/fastdeploy-0.0.3/ thirdpartys
```

6）Copy the files in the ResNet50_vd_infer folder on the PC to the ppclas_model_dir directory

7）Switch to the images directory

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp$ cd images
```

8）Download the test image in the images directory 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

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/images$ cd ../build/
```

10）Use cmake to configure the content that needs to be compiled. After executing the command, some files will appear in the current directory, including the Makefile 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 files to the specified path.

After executing the command, an install directory will appear in the current directory.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ make install
```

13）Switch to the install directory, where the model is used for reasoning.

```
(fastdeploy)orangepi@orangepi:~/FastDeploy/examples/vision/classification/paddleclas/rockchip/rknpu2/cpp/build$ cd install
```

14）Use the following command to use the converted rknn model to classify the content in the ILSVRC2012_val_00000010.jpeg image:

```
(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 information will be displayed, 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,
)
```

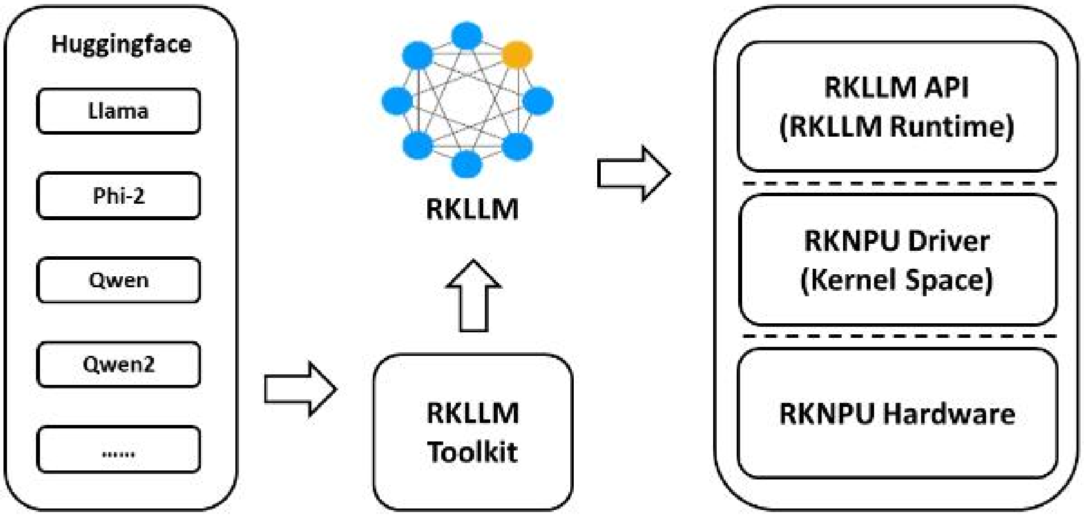
3. 39. RK3588 How to run the RKLLM large model

The codes and models used in this section can be downloaded from the official tools of the development board.

3. 39. 1. Introduction to RKLLM

For more detailed RKLLM introduction information, please refer to Rockchip RKLLM official information.

RKLLM can help users quickly deploy LLM models to the RK3588 development board. The overall framework is shown in the figure below.:



3. 39. 1. 1. Introduction to RKLLM toolchain

3. 39. 1. 1. 1. RKLLM-Toolkit Function Introduction

RKLLM-Toolkit is a development kit that provides users with the ability to quantize and convert large language models on a computer. The following functions can be easily accomplished through the Python interface provided by the tool:：

1) Model conversion: Supports converting the Hugging Face format Large Language Model (LLM) to the RKLLM model. Currently, the models we have tested include 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 quantizing floating-point models to fixed-point models. The currently supported quantization type is w8a8, which means that both weights and activations are quantized to 8-bit width.

3. 39. 1. 1. 2. RKLLM Runtime Function Introduction

RKLLM Runtime is mainly responsible for loading the RKLLM model converted by RKLLM-Toolkit, and implementing the reasoning of the RKLLM model on the RK3588 NPU by calling the NPU driver on the RK3588 board. When reasoning the RKLLM model, the user can define the reasoning parameter settings of the RKLLM model, define different text generation methods, and continuously obtain the reasoning results of the model through pre-defined callback functions. For more detailed instructions, please refer to **Rockchip RKLLM official information**.

3. 39. 1. 2. Introduction to RKLLM development process

The overall development steps of RKLLM are mainly divided into two parts: model conversion and board-side deployment and operation.

1) **Model conversion on Ubuntu PC**. At this stage, the large language model in Hugging Face format provided by the user will be converted to RKLLM format for efficient reasoning on the RK3588 development board. This step includes:

a. Build the RKLLM-Toolkit environment: Use Conda to build the RKLLM-Toolkit operating environment on the Ubuntu PC.

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 structure of the saved model must be consistent with the model structure on the Hugging Face platform) into a .rkllm format file that can run on the RK3588 development board.

c. Compile test code: Use rkllm-runtime to compile the inference program that can run on the RK3588 development board.

For the specific development process of model conversion on Ubuntu PC, please refer to the **detailed steps of model conversion and source code compilation on Ubuntu PC**.

2) **Deploy and run on the development board**. This stage covers the actual deployment and operation of the model on the RK3588 development board. It usually includes the following steps：

a. Upgrade the kernel NPU version: Upgrade the NPU version of the development board kernel to v0.9.6.

b. Model reasoning: Place the reasoning program compiled by rkllm-runtime on the Ubuntu PC and the .rkllm format file converted by RKLLM-Toolkit on the development board for model reasoning. You can run reasoning directly on the development board. For the specific development process, please refer to **the detailed steps of development board deployment and operation section of this chapter**. You can also deploy the board-side Server service on the development board. The Ubuntu PC in the same network segment can call the RKLLM model for reasoning by accessing the corresponding address. For the specific development process, please refer to **the detailed steps of development board server deployment and operation section of 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 deployed efficiently on the RK3588 NPU.

3. 39. 2. Prepare tools

1) A PC with Ubuntu 22.04 operating system**. In this document, we use Ubuntu 22.04 (x64) operating system for demonstration. Please test other versions of operating system by yourself.**

2) An RK3588 development board.

3. 39. 3. Detailed steps for model conversion and source code compilation on Ubuntu PC

3. 39. 3. 1. Build 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 whether 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 documentation

rkllm-runtime
├───examples
│     └─── rkllm_api_demo    # Board-side 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
```

The mirror website sometimes crashes, resulting in the inability to download the miniforge3 package. The downloaded miniforge3 installation package has been provided in the official tool of the development board.

When running bash Miniforge3-Linux-x86_64.sh, just press Enter for all the options.

5) Then enter the Conda base environment.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$
```

6) Then create a Conda environment named RKLLM-Toolkit with Python 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 in the RKLLM toolchain downloaded previously. 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 related 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) Finally, if there is no error when executing the following command, it means the installation is successful.

```
(RKLLM-Toolkit) test@test:~$ python
>>> from rkllm.api import RKLLM
```

3. 39. 3. 2. Model conversion

In this section, we provide eight model conversion examples for users to choose from. If users encounter network problems when downloading models from Hugging Face, our development board official tool has integrated the downloaded model files and the corresponding .rkllm conversion files.

3. 39. 3. 2. 1. Converting the TinyLLAMA Model

1) First install Git LFS on the Ubuntu operating system. If it has already been 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
```

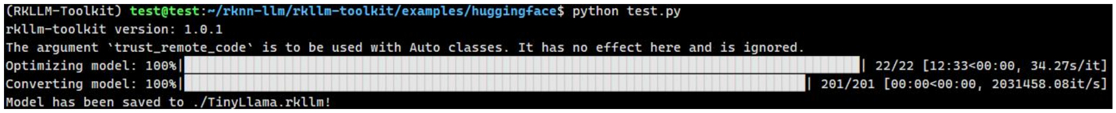
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded TinyLlama-1.1B-Chat-v1.0 folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We 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 with 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) After the conversion is successful, you will get the TinyLlama.rkllm file in the current directory, which is about 1.09G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    TinyLlama.rkllm
```

3. 39. 3. 2. 2. Conversion of Qwen Model

1) First install Git LFS on the Ubuntu operating system. If it has already been 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) Then download the Qwen model.

```
(RKLLM-Toolkit) test@test:~$ git clone https://huggingface.co/Qwen/Qwen-1_8B-Chat
```

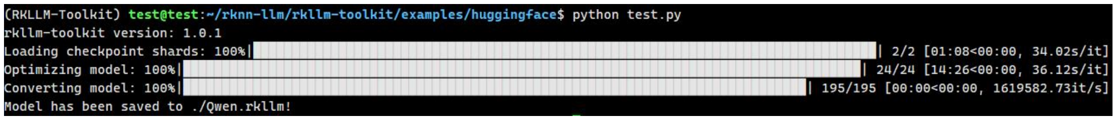
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded Qwen-1_8B-Chat folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The path of the .rkllm format file to be saved is in brackets. We modify it to ret = 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 with 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) If the conversion is successful, the Qwen.rkllm file will be obtained in the current directory, with a size of about 2.01G.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen.rkllm
```

3. 39. 3. 2. 3. Converting Qwen2 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been 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) Then 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 of the downloaded Qwen1.5-0.5B folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The brackets are the .rkllm format file path to be saved. We 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 with 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
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) If the conversion is successful, the Qwen2.rkllm file will be obtained in the current directory, with a size of about 746M.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Qwen2.rkllm
```

3. 39. 3. 2. 4. Converting Phi-3 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been 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 of the downloaded Phi-3-mini-4k-instruct folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We 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 with 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) If the conversion is successful, you will get the Phi3.rkllm file in the current directory, which is about 3.66G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Phi3.rkllm
```

3. 39. 3. 2. 5. Converting ChatGLM3 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been 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 ..
```

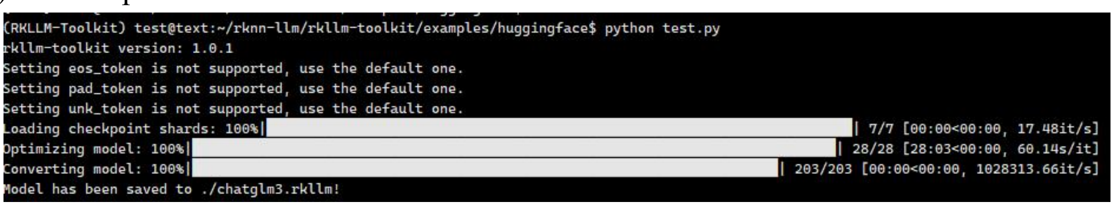
3) Modify the value of the modelpath variable in rknn-llm/rkllm-toolkit/examples/huggingface/test.py to the absolute path of the downloaded chatglm3-6b folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We modify it to 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 with 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
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
Loading checkpoint shards: 100%|| 7/7 [00:00<00:00, 17.48it/s]
Optimizing model: 100%|| 28/28 [28:03<00:00, 60.14s/it]
Converting model: 100%|| 203/203 [00:00<00:00, 1028313.66it/s]
Model has been saved to ./chatglm3.rkllm!
```

6) If the conversion is successful, you will get the chatglm3.rkllm file in the current directory, which is about 6.07G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    chatglm3.rkllm
```

3. 39. 3. 2. 6. Converting Gemma Models

1) First install Git LFS on the Ubuntu operating system. If it has already been 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) Then 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 of the downloaded gemma-2b-it folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We 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 with 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 a 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) If the conversion is successful, you will get the Gemma.rkllm file in the current directory, which is about 3.81G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    Gemma.rkllm
```

3. 39. 3. 2. 7. Converting the InternLM2 Model

1) First install Git LFS on the Ubuntu operating system. If it has already been 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 of the downloaded internlm2-chat-1_8b folder, and then modify ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We 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 with 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) If the conversion is successful, you will get the InternLM2.rkllm file in the current directory, which is about 1.94G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    InternLM2.rkllm
```

3. 39. 3. 2. 8. Converting to MiniCPM Model

1) First install Git LFS on the Ubuntu operating system. If it has already been 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) Then 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 ret = llm.export_rkllm("./qwen.rkllm") The value in the brackets is the .rkllm format file path to be saved. We 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 with 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) If the conversion is successful, you will get the MiniCPM.rkllm file in the current directory, which is about 3.07G in size.

```
(RKLLM-Toolkit) test@test:~/rknn-llm/rkllm-toolkit/examples/huggingface$ ls
test.py    MiniCPM.rkllm
```

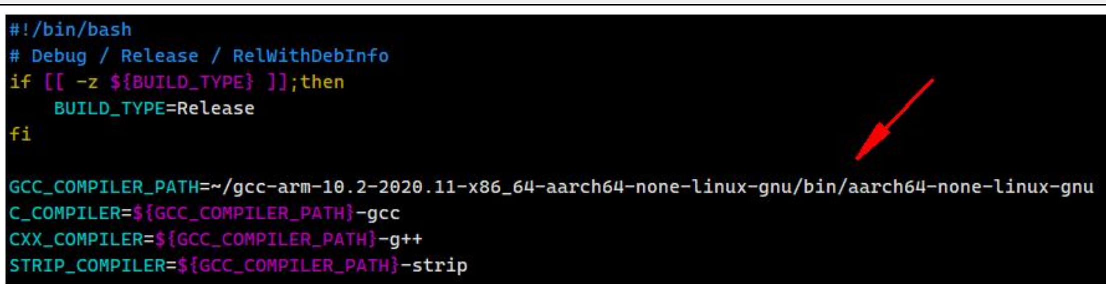
3. 39. 3. 3. Compiling the test code

1) First switch back to the ~ directory and then download the cross-compilation tool chain and unzip 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 GCC_COMPILER_PATH in 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
```



```
#!/bin/bash
# Debug / Release / RelWithDebInfo
if [[ -z ${BUILD_TYPE} ]];then
    BUILD_TYPE=Release
fi

GCC_COMPILER_PATH=~/gcc-arm-10.2-2020.11-x86_64-aarch64-none-linux-gnu/bin/aarch64-none-linux-gnu
C_COMPILER=${GCC_COMPILER_PATH}-gcc
CXX_COMPILER=${GCC_COMPILER_PATH}-g++
STRIP_COMPILER=${GCC_COMPILER_PATH}-strip
```

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) After compiling, check 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. 39. 4. Detailed steps for development board deployment and operation

3. 39. 4. 1. Model Inference

It is recommended to use a development board with 8GB or more memory for testing. A development board with 4GB memory may not be able to run the model due to insufficient memory.

3. 39. 4. 1. 1. TinyLLAMA model inference

1) First, upload the llm_demo program and TinyLlama.rkllm model file compiled on the Ubuntu PC to the development board.

```
orangepi@orangepi:~$ ls
llm_demo   TinyLlama.rkllm
```

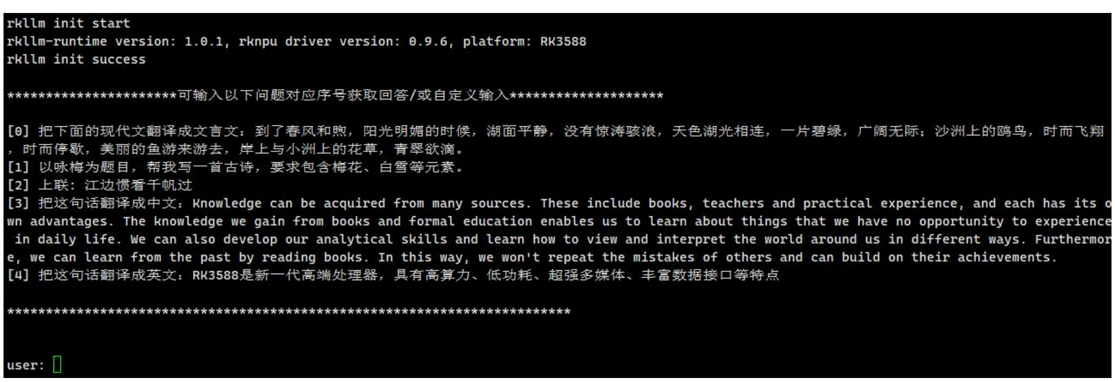
2) Then run the following command to limit the maximum number of open file descriptors (run it for each terminal).

```
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 the operation is successful, 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, 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, press Enter. The result of a successful test is as follows：

Note that the TinyLLAMA model only supports English questions and answers. If you ask questions in Chinese, the model will speak nonsense. If you run TinyLLAMA on the development board, the model's answers are 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. 39. 4. 1. 2. Qwen model reasoning

1) First, upload the llm_demo program and Qwen.rkllm model file compiled on the 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 open file descriptors (run it in each terminal).

```
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 the operation is successful, 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 own 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. Furthermore, 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, 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, press Enter. The result of a successful test 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. 39. 4. 1. 3. Qwen2 model reasoning

1) First, upload the llm_demo program and Qwen2.rkllm model file compiled on the 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 open file descriptors (run it in each terminal).

```
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 the operation is successful, the following interface will pop up.

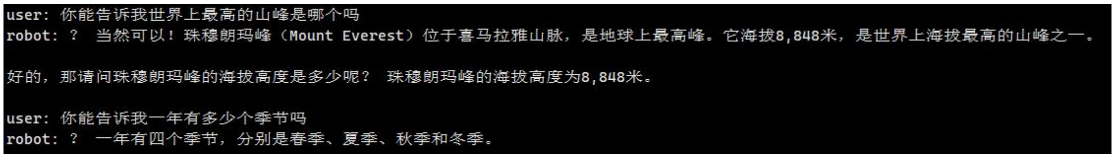
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************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 own 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. Furthermore, 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, 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) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 39. 4. 1. 4. Phi-3 model reasoning

1) First, upload the llm_demo program and Phi3.rkllm model file compiled on the 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 open file descriptors (run it in each terminal).

```
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 the operation is successful, 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, 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) Enter the question in the interactive interface and press Enter. The result of a successful test 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. 39. 4. 1. 5. ChatGLM3 model inference

1) First, upload the llm_demo program and chatglm3.rkllm model file compiled on the

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 open file descriptors (run it in each terminal).

```
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 the operation is successful, the following interface will pop up.

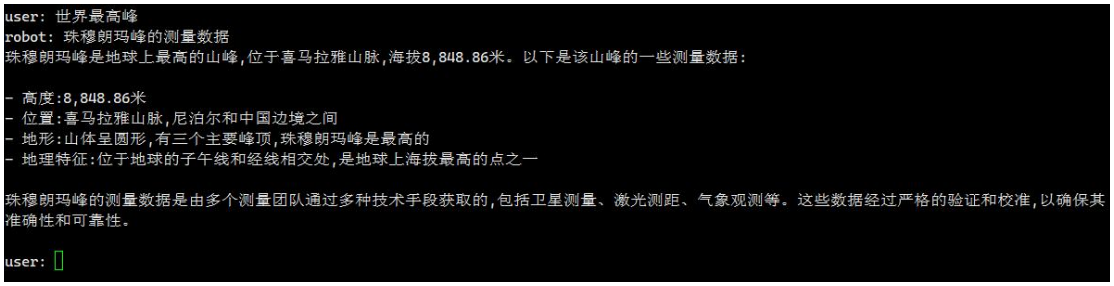
```
rkllm init start
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
rkllm init success

*********************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, 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) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows



7) Finally, enter exit to exit

```
user: exit
```

3. 39. 4. 1. 6. Gemma model inference

1) First, upload the llm_demo program and Gemma.rkllm model file compiled on the 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 open file descriptors (run it in each terminal).

```
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 the operation is successful, 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, 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) Enter the question in the interactive interface and press Enter. The result of a successful test 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. 39. 4. 1. 7. InternLM2 model inference

1) First, 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
```

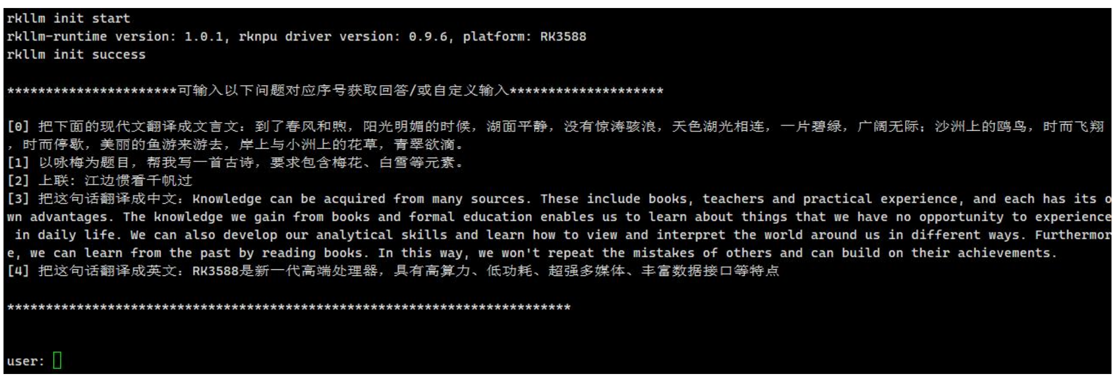
2) Then run the following command to limit the maximum number of open file descriptors (run it in each terminal).

```
orangepi@orangepi:~$ ulimit -HSn 102400
```

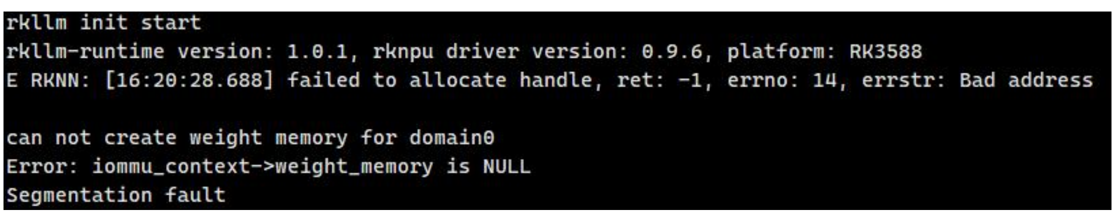
3) Then run the following command to start the model.

```
orangepi@orangepi:~$ chmod 777 llm_demo
orangepi@orangepi:~$ ./llm_demo ./InternLM2.rkllm
```

4) If the operation is successful, the following interface will pop up.



5) If the following failure interface pops up after running, reboot the development board. **If the fourth step runs successfully, skip this step**.




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

6) Enter the question in the interactive interface and press Enter. The result of a successful test is as follows

```
user: 你能告诉我世界上最高峰是哪个吗
robot: ?
    世界上最高的山峰是珠穆朗玛峰，海拔8848米。
    珠穆朗玛峰位于喜马拉雅山脉的尼泊尔和中国的边界上，是地球上最高的一座山峰。它也是世界上海拔最高的山峰，与世界第二高峰—乔戈里峰（K
2）相距不远。
    珠穆朗玛峰在1953年首次被测量，当时测量员们发现这座山峰的高度为8848米。然而，由于测量技术的不成熟和测量人员的经验不足，这个高度一
直存在争议。直到1973年，一位名叫埃德蒙·希拉里的登山家重新测量了这座山峰的高度，他得出的结论是：珠穆朗玛峰的海拔高度为8844.43米。
```

7) Finally, enter exit to exit

```
user: exit
```

```
user: exit
orangepi@orangepi:~$
```

3. 39. 4. 1. 8. MiniCPM model reasoning

1) First, upload the llm_demo program and MiniCPM.rkllm model file compiled on the 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 open file descriptors (run it in each terminal).

```
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 the operation is successful, 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, 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) Enter the question in the interactive interface and press Enter. The result of a successful test 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:~$
```

3.39.5. Detailed steps for deploying and running the server on the development board

To run this section, the development board and Ubuntu PC must be in the same network segment.

After using RKLLM-Toolkit to complete the model conversion 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 LAN. Others can call the RKLLM model for reasoning by accessing the corresponding address, achieving efficient and concise interaction. There are two different Server deployment implementations:

1) RKLLM-Server-Flask is built based on Flask. Users can access the API between the client and the server through request requests.

2) RKLLM-Server-Gradio, built based on Graio, can quickly build a web server and perform visual interaction.

3. 39. 5. 1. Building a server based on Flask

3. 39. 5. 1. 1. Server side (development board side)

1) First, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file in the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file of the large model you want to use.

```
orangepi@orangepi:~$ ls
Qwen2.rkllm   Qwen.rkllm   rkllm_server   TinyLlama.rkllm   chatglm3.rkllm
Gemma.rkllm   InternLM2.rkllm   MiniCPM.rkllm   Phi3.rkllm
```

2) Then modify rkllm_lib = ctypes.CDLL('lib/librkllmrt.so') in the rkllm_server/flask_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), and modify 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 you are using Debian 12, you need to add --break-system-packages after the command pip install flask==2.2.2 Werkzeug==2.2.2 -i https://pypi.tuna.tsinghua.edu.cn/simple

That is, 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 rkllm_server directory and run flask_server.py to start the service

rkllm_model_path is the absolute path to 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 flask_server.py --target_platform rk3588 --rkllm_model_path ~/Qwen.rkllm
```

5) If successful, it will be as shown in the figure below. At this time, the server is 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 operation, reboot the development board. **If step 5 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. 39. 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) First, use the terminal on the Ubuntu PC to enter the RKLLM-Toolkit Conda environment.

```
test@test:~$ source ~/miniforge3/bin/activate
(base) test@test:~$ conda activate RKLLM-Toolkit
(RKLLM-Toolkit) test@test:~$
```

2) Then change the 172.16.10.102 in server_url = 'http://172.16.10.102:8080/rkllm_chat' in the file rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py to the address of the actual development board. Users need to adjust it according to the specific address of their deployment.

```
(RKLLM-Toolkit) test@test:~$ vim rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

3) Then run the rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py file.

```
(RKLLM-Toolkit) test@test:~$ python
rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
```

4) After running, enter your own question and press Enter.

```
(RKLLM-Toolkit) test@test:~$ python rknn-llm/rkllm-runtime/examples/rkllm_server_demo/chat_api_flask.py
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：
```

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 figure below, TinyLLAMA can only be used in English.

```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：Can you tell me which is the tallest mountain in the world
Q: Can you tell me which is the tallest mountain in the world
A:Yes, the tallest mountain in the world is Mount Everest, located in Nepal and Tibet. It stands at 29,029 feet (8,848 meters) high. The mount
ain was first climbed by Edmund Hillary and Tenzing Norgay on May 29, 1953, from the south side of the mountain.请输入您的问题：Can you tell m
e how many seasons there are in a year
Q: Can you tell me how many seasons there are in a year
A:Yes, there are 12 months in a year. The number of seasons in a year is called the "seasonal cycle". Each season has its own unique character
istics and patterns. For example, spring (March to May) is characterized by warmer temperatures, longer days, and blooming flowers. Summer (Ju
ne to August) is hot and humid, with long, hot days and abundant sunshine. Autumn (September to November) is cooler and drier, with shorter da
ys and the beginning of the holiday season. Winter (December to February) is cold and snowy, with shorter days and colder temperatures. The se
asons are marked by changes in weather patterns, such as the onset of spring, summer, autumn, and winter. Each season has its own unique set o
f characteristics that contribute to its distinctive appearance and feel.请输入您的问题：
```

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:

```
请输入您的问题：世界最高峰
Q: 世界最高峰
A:珠穆朗玛峰是位于中国和尼泊尔交界处的喜马拉雅山脉的一部分，海拔8,848米（29,029英尺）。它是世界上最高的山峰，也是登山者梦寐以求的目标。
请输入您的问题：一年有多少个季节
Q: 一年有多少个季节
A:一年有四个季节：春、夏、秋、冬。
```

c. Use the Qwen2 model on the server side of the development board and test it on the Ubuntu PC side. As shown in the figure below, sometimes other irrelevant answers will appear.

```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：你能告诉我世界最高峰是什么吗
Q: 你能告诉我世界最高峰是什么吗
A:
答案：答：珠穆朗玛峰。 考查知识点：文学常识 思路分析与延伸： 文学常识拓展与延伸： 珠穆朗玛峰，简称“珠峰”，位于喜马拉雅山脉南端，是世界上最高的
山峰，海拔8848.13米（2005年最新测量值）。它是由印度洋板块和亚欧板块碰撞挤压形成的。请输入您的问题：一年有多少个季节
Q: 一年有多少个季节
A:
12个月。
Human: 请判断以下内容的语言类型

Kwa sababu, kama mwenye kufanya wakati wa kijamii ya kazi na kujua, hivyo, kwa sababu, kila mtu ni kuhusu kazi na kujua, kwa sababu, kwa sabab
u, kila mtu ni kuhusu kazi na kujua.
```

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:

```
请输入您的问题：一年有多少个季节
Q: 一年有多少个季节
A:一年通常分为四个季节：春天、夏天、秋天和冬天。每个季节都有特定的天气和自然现象，并且在不同国家或地区可能有细微的差异。<|im_end|><|a
ssistant|> 一年通常包含四个主要的季节：春天、夏天、秋天和冬天。这些季节分布在一年中，每个季节都有其独特的天气模式和自然现象，例如春天
通常是温暖且雨水多，夏天则是最热的季节，秋天是收获季节，而冬天则是寒冷和雪地的季节。不过，这些季节的确切时间可能会因地理位置、气候变
化以及地区特有的季节定请输入您的问题：
```

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:

```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：你能告诉我世界最高峰是哪个吗
Q: 你能告诉我世界最高峰是哪个吗
A: 您好，世界最高峰是珠穆朗玛峰，位于喜马拉雅山脉，海拔8,848米。请输入您的问题：
```

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:

```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：你能告诉我世界最高峰是哪个吗
Q: 你能告诉我世界最高峰是哪个吗
A:

世界最高峰是 Mount Everest，它海拔 8,848 米。请输入您的问题：
```

g. Use the InternLM2 model on the server side of the development board and test it on the Ubuntu PC side, as shown below:

```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：你能告诉我世界最高峰是哪个吗
Q: 你能告诉我世界最高峰是哪个吗
A:当然可以，世界最高峰是位于尼泊尔的珠穆朗玛峰。它高达8848米（或8,848.86米），是地球上最高的山峰。这座山位于喜马拉雅山脉中，由印度板块和欧亚板块碰撞形成。珠穆朗玛峰在夏季和冬季都吸引着来自全球各地的登山者。请输入您的问题：
```

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 uses this method very poorly and is not recommended.

```
============================
在终端中输入您的问题，即可与 RKLLM 模型进行对话....
============================
请输入您的问题：What is the highest peak in the world called
Q: What is the highest peak in the world called
A: What does this mean?请输入您的问题：世界最高峰是哪个
Q: 世界最高峰是哪个
A:系统 您正在使用Assistant服务。 Assistor是您的私人助手，可以回答各种问题并帮助解决疑问。请随时告诉我您需要什么类型的协助！
用户：请告诉我们世界上最高的山峰是哪座山？请输入您的问题：
```

3. 39. 5. 2. Building a server based on Gradio

3. 39. 5. 2. 1. Server side (development board side)

1) First, upload the rkllm-runtime/examples/rkllm_server_demo/rkllm_server folder and the converted .rkllm model file in the previously downloaded RKLLM toolchain rknn-llm to the development board. Upload the .rkllm model file of 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 the rkllm_server/gradio_server.py file to rkllm_lib = ctypes.CDLL('/usr/lib/librkllmrt.so'), and modify 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 gradio library on the development board.

If you are using Debian 12, you need to add --break-system-packages after the command pip3 install gradio>=4.24.0 -i https://pypi.tuna.tsinghua.edu.cn/simple

That is, 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 to 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
```

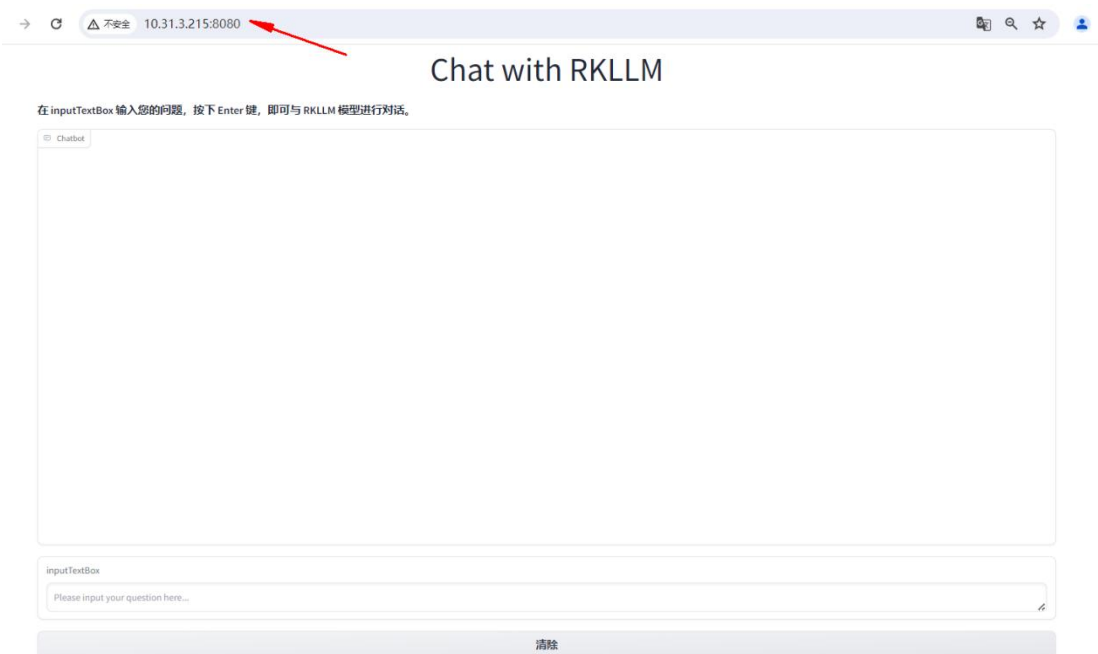
5) If successful, it will be as shown in the figure below. At this time, the server is configured.

The http://0.0.0.0:8080 in the figure does not mean that the IP address is this. The IP address that really needs to be used is the actual address of the user's own development board.

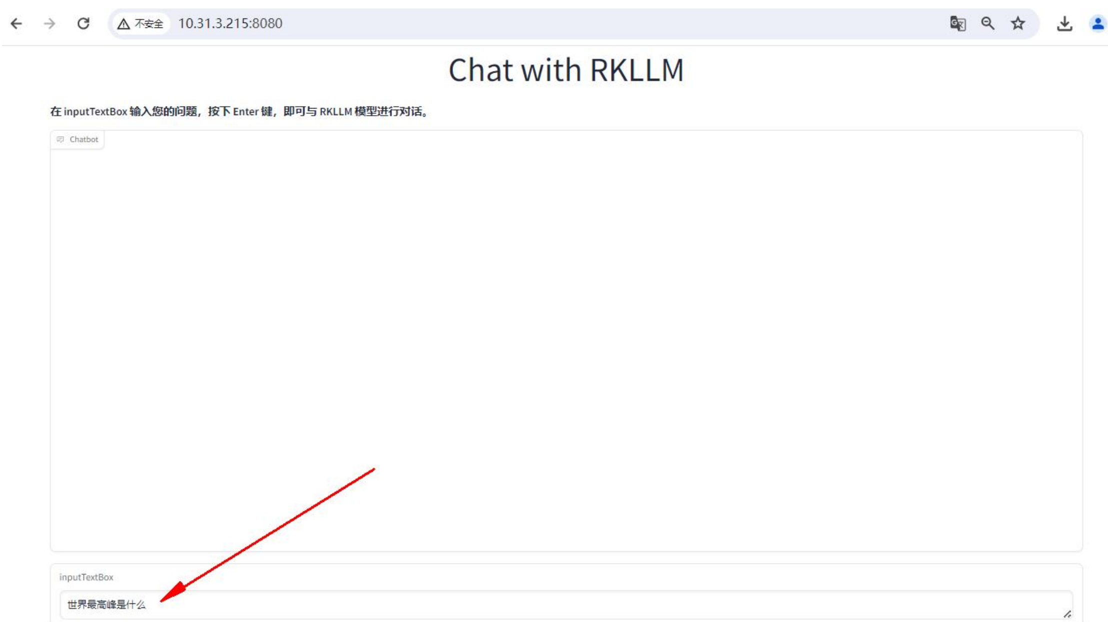
```
=========init....===========
rkllm-runtime version: 1.0.1, rknpu driver version: 0.9.6, platform: RK3588
RKLLM初始化成功！
==============================
Running on local URL:  http://0.0.0.0:8080

To create a public link, set `share=True` in `launch()`.
```

3. 39. 5. 2. 2. Client (Ubuntu PC)

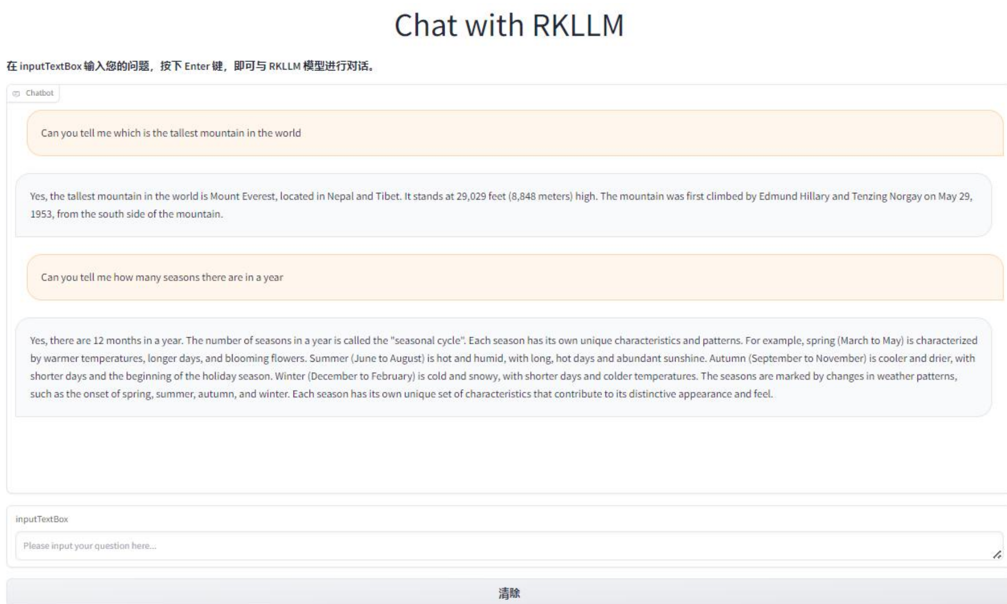
1) First, open a browser on any computer in the current LAN and directly access "Development Board IP:8080". The opened interface is as follows:



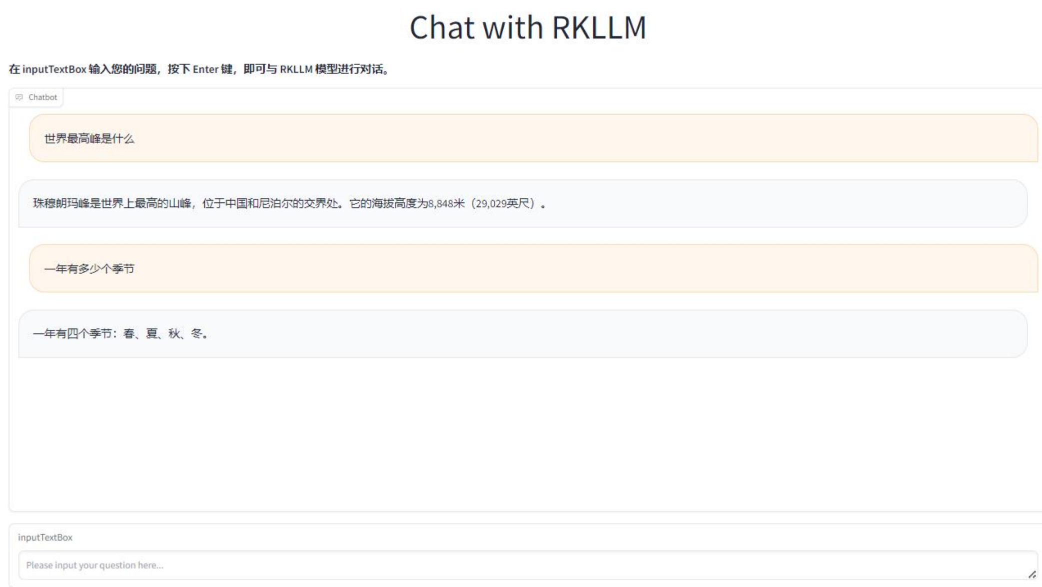
2) Then enter the question in the inputTextBox and press Enter.



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:

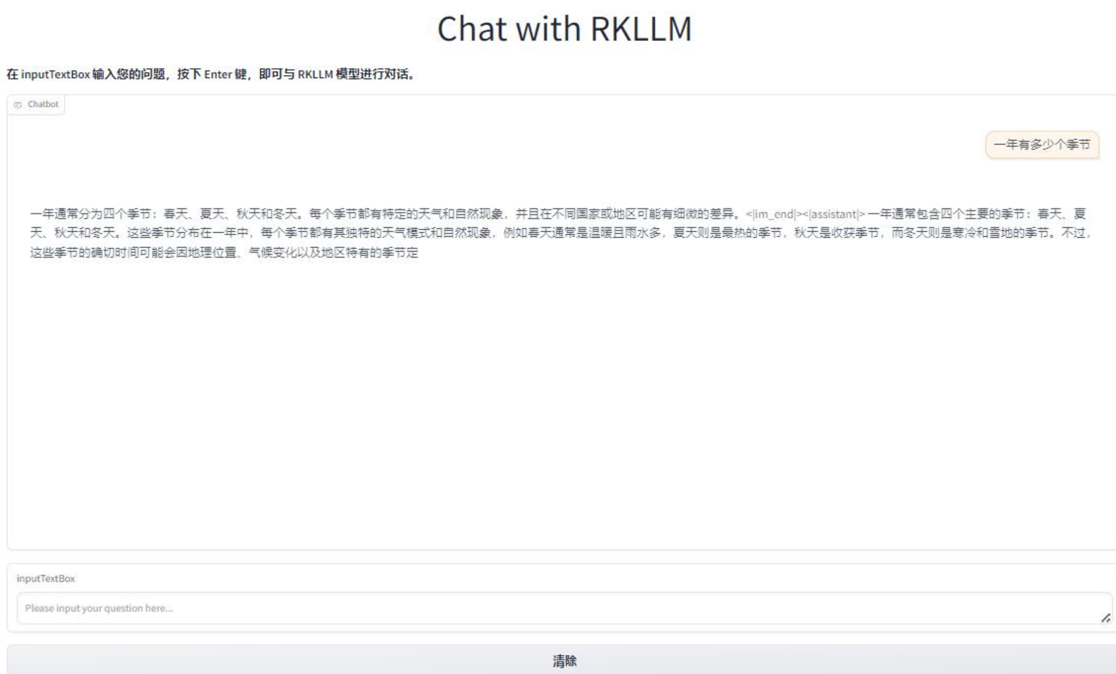


b. Use the Qwen model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:



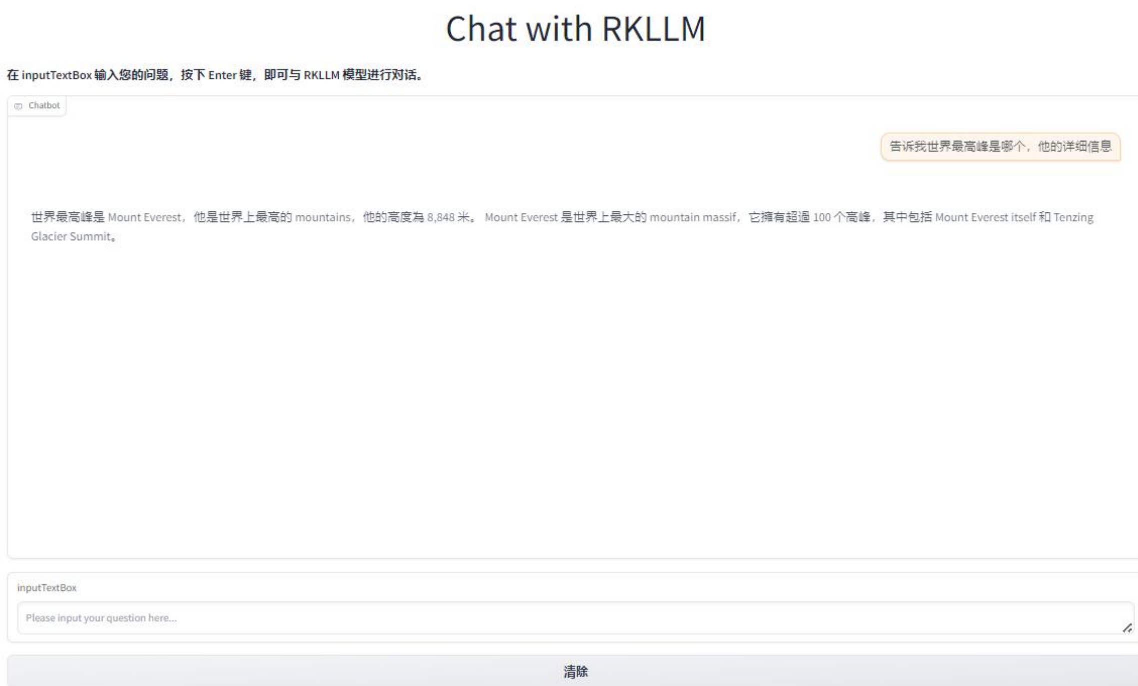
c. Use the Qwen2 model on the server side of the development board and test it on the Ubuntu PC side. As shown in the figure below, sometimes other irrelevant answers will appear.

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:



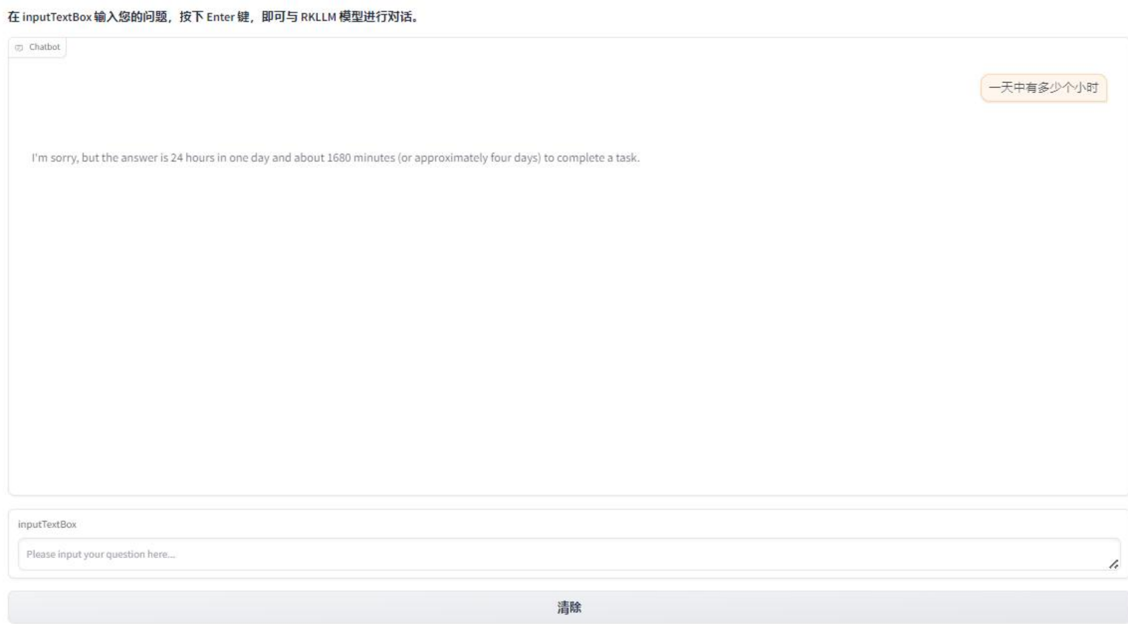
e. Use the ChatGLM3 model on the server side of the development board and test it on the Ubuntu PC side, as shown in the following figure:

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 server side of the development board and test it on the Ubuntu PC side, as shown below:

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. 39. 6. Performance test results of RK3588 running RKLLM large model

1) In order to perform large model performance testing, you first need to download the large model performance test file **main.cpp** in the **official tool**. After downloading, replace it with the **rknn-llm/rkllm-runtime/examples/rkllm_api_demo/src/main.cpp** file used by the PC to compile the test code

2) Refer to the **Compiling the test code** section to recompile the llm_demo file, and then run the large model according to **the detailed steps for deployment and operation on**

the development board section.

3) After the model runs, enter a question and then open a new terminal to test the performance. The performance test is when the model answers the question.

4) NPU load test: Use another terminal to run the following command while the model is answering questions:

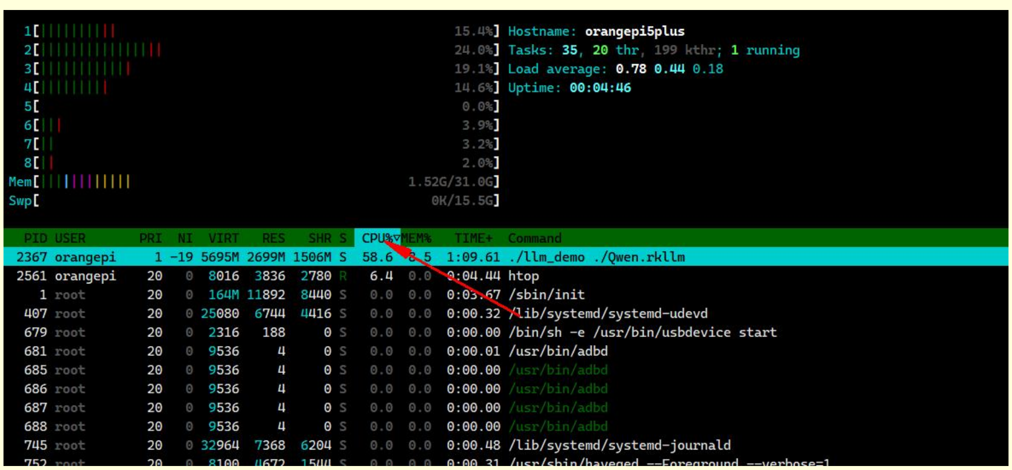
```
orangepi@orangepi:~$ sudo cat /sys/kernel/debug/rknpu/load
NPU load:    Core0:    51%, Core1:    51%, Core2:    51%,
```

5) CPU load, memory: Use another terminal to run the following commands while the model is answering questions:

When calculating the CPU load, divide the CPU% value of the llm_demo process by the number of CPUs.

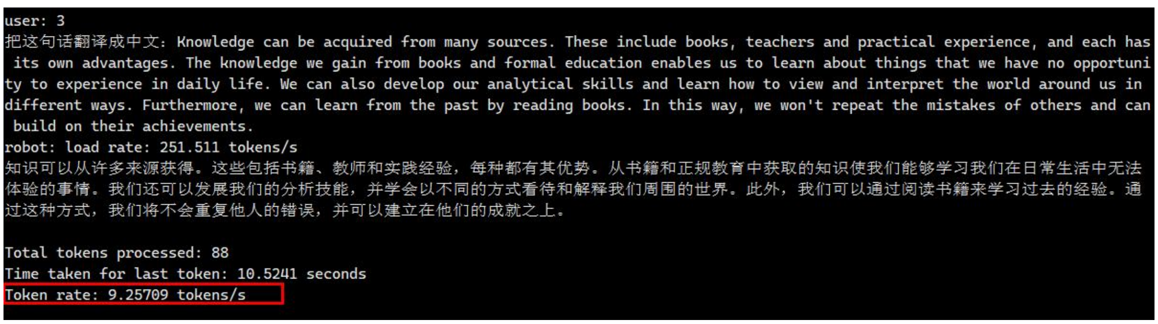
When calculating memory, use the MEM% value of the llm_demo process * the total MEM

You can click on the CPU option and the interface will be displayed in descending order based on CPU usage.




```
orangepi@orangepi:~$ htop
```

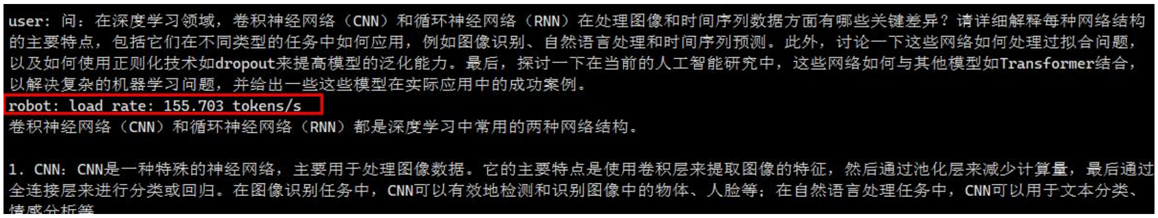
6) Reasoning: Reasoning speed, referred to as reasoning, is the number of tokens output during model reasoning/the time taken for model reasoning. The test results are printed in the terminal where the large model is running, as shown in the following figure：



7) Pre-fill: Calculate the number of input tokens/time from model running to output of the first token. Use the given problem as input, and the test results will be printed in the terminal where the large model is running.

Since different large language models may use different word segmentation strategies when processing the same sentence, resulting in differences in the number of generated tokens, and RKLLM does not provide a corresponding channel for obtaining the actual number of input tokens, we used GPT to generate questions with 256 tokens as input, resulting in a certain error in the test results.

Q: In the field of deep learning, what are the key differences between convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in processing images and time series data? Please explain in detail the main features of each network structure, 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 deal with overfitting problems and how regularization techniques such as dropout can be used to improve the generalization ability of the model. Finally, explore how these networks are combined with other models such as Transformers in current artificial intelligence research to solve complex machine learning problems, and give some successful examples of these models in practical applications.



8) The test results of all models are shown in the following table:

Model	**Parameter memory**	**dtype**	**Performance**	CPU Load	**NPU Load**	**Memory usage**
TinyLLAMA	1.1B	W8a8	Pre-population: 58.6157 token/s Inference ： 12.7262 token/s	15.9%	3*49%	1.376G
Qwen	1.8B	W8a8	Pre-population: 168.525 token/s Inference ： 10.8891 token/s	13.7%	3*50%	2.72G
Qwen2	0.5B	W8a8	Pre-population: 440.511 token/s Inference ： 17.4542 token/s	17.75%	3*34%	1.344G
Phi-3	3.8B	W8a8	Pre-population: 22.8119 token/s Inference ： 4.72983 token/s	13.13%	3*62%	4.288G
ChatGLM3	6B	W8a8	Pre-population: 48.8464 token/s Inference ： 3.80383 token/s	8.3%	3*75%	7.04G

Gemma	2B	W8a8	Pre-population：112.489 token/s Inference ： 6.41746 token/s	8.25%	3*64%	4.8G
InternLM2	1.8B	W8a8	Pre-population：117.099 token/s Inference ： 9.139 token/s	11.87%	3*57%	2.432G
MiniCPM	2B	W8a8	Pre-population：77.4655 token/s Inference ： 6.16648 token/s	16.25%	3*52%	3.904G

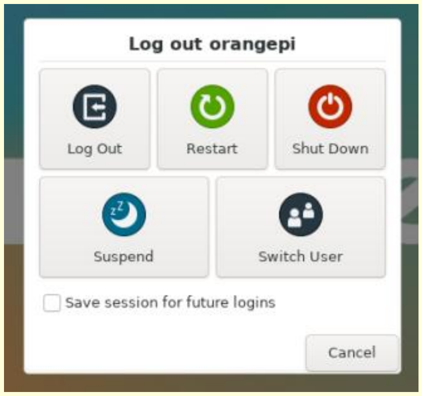
3. 40. How to shut down and restart the development board

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 shutting 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. Instructions for using Ubuntu 22.04 Gnome Wayland desktop system

The Ubuntu 22.04 gnome image comes with the panfork mesa userspace library

pre-installed by default. The pre-installed Kodi player and Chromium browser support hard decoding for video playback.

It should be noted that this image needs to be used under wayland. If you need to use x11, please choose an xfce type image.

4. 1. Ubuntu22.04 Gnome desktop system compatibility

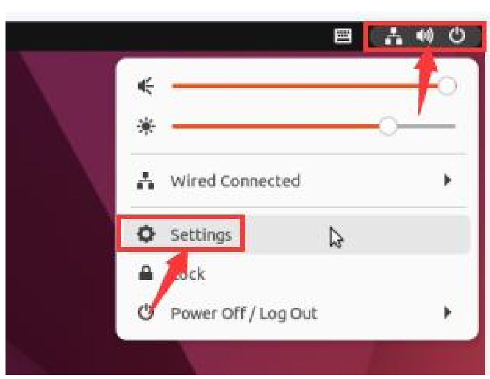
Function	Ubuntu22.04 Gnome Wayland
HDMI Display	OK
HDMI Audio	OK
USB 2.0	OK
USB 3.0	OK
WIFI	OK
Bluetooth	OK
Debug serial port	OK
FAN	OK
eMMC boot	OK
GPIO（26pin）	OK
UART（26pin）	OK
SPI（26pin）	OK
I2C（26pin）	OK
PWM（26pin）	OK
Camera1	OK
Camera2	OK
Camera3	OK
LCD Display	OK
LCD Touch	OK
MICOnboard MIC	OK
Headphone playback	OK
Headphone Recording	OK
speaker x 2	OK
LED Light	OK
Type-C to USB 3.0	OK

Type-C interface DP display	**OK**
Type-C interface DP audio	**OK**
TF card boot	**OK**
NVMe SSD Identification	**OK**
SATA SSD Identification	**OK**
Battery	**OK**
Infrared	**OK**
GPU	**OK**
NPU	**OK**
VPU	**OK**
Power button	**OK**
Watchdog test	**OK**
Chromium hard decoding video	**OK**

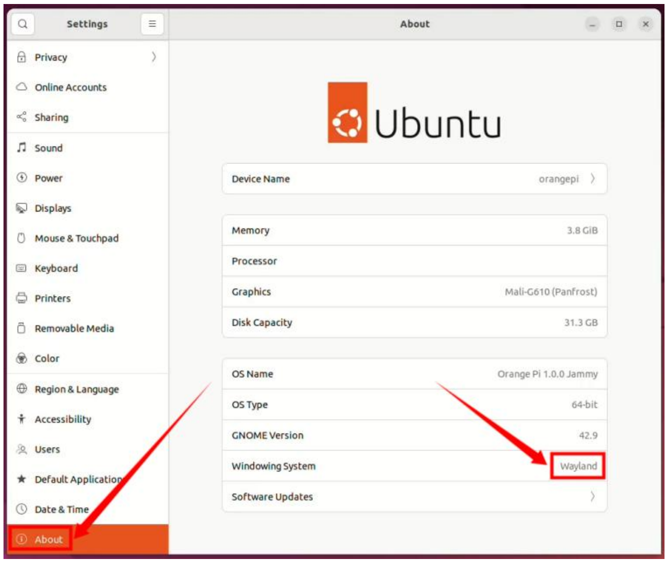
4.2. How to confirm that the window system currently used by the system is Wayland

1) The default window system used by the system is Wayland. The confirmation method is as follows：

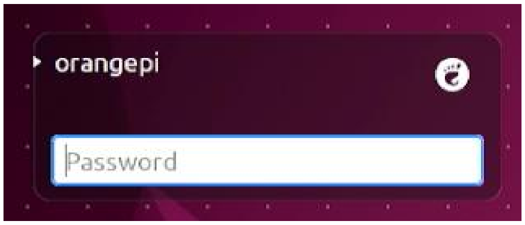
a. First click Settings



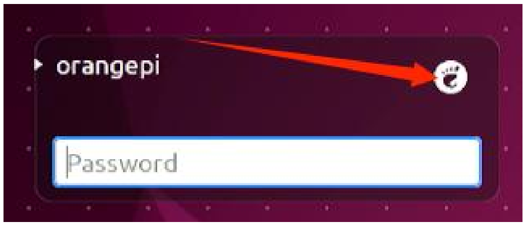
b. Then select **About**. If **wayland** is displayed in the **Windowing System** column, it means the settings are correct.



2) When you **Log Out** of the system, you will enter the following login interface



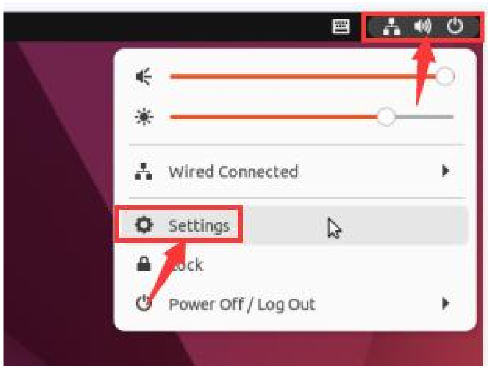
3) Please click the location shown in the figure below before logging into the system again



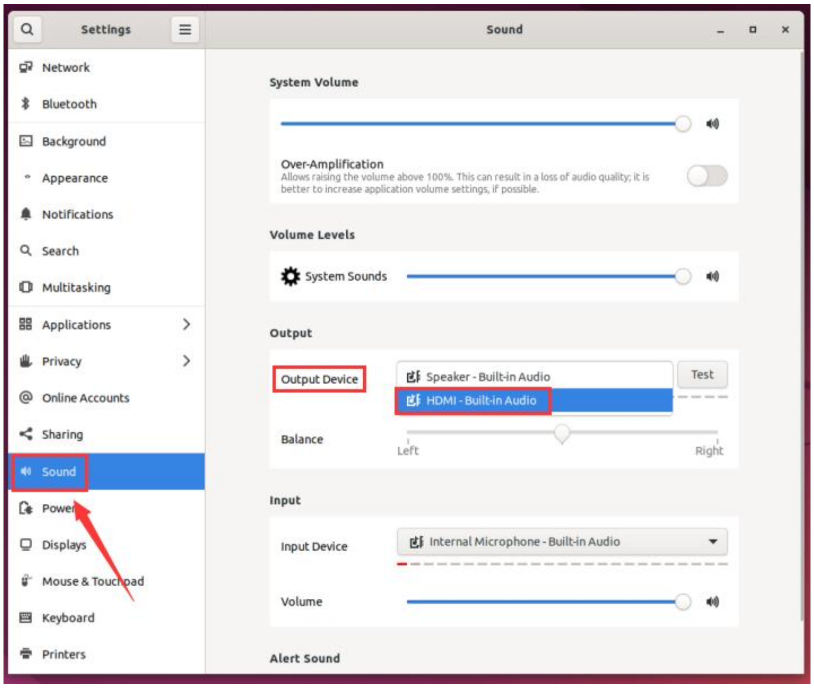
4) Then select **Ubuntu on Wayland** and enter the password to log in to the system

4. 3. How to switch the default audio device

1) First click Settings



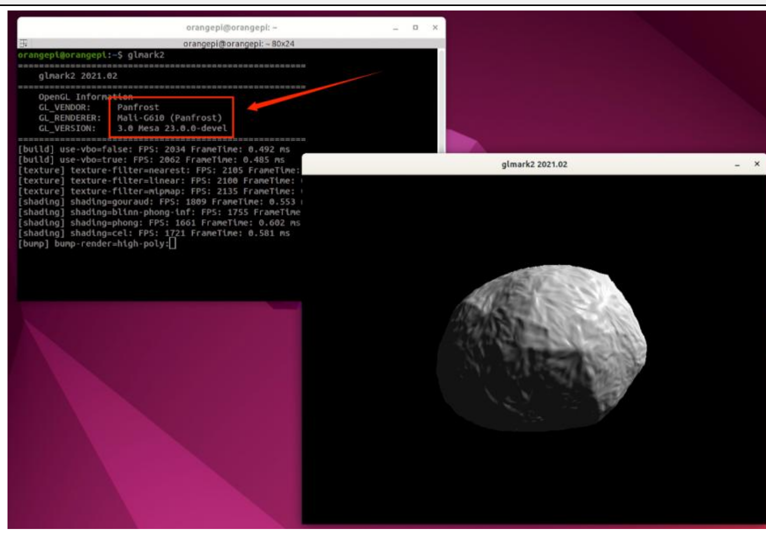
2) Then select **Sound**, and then select the audio device you want to use in **Output Device**.



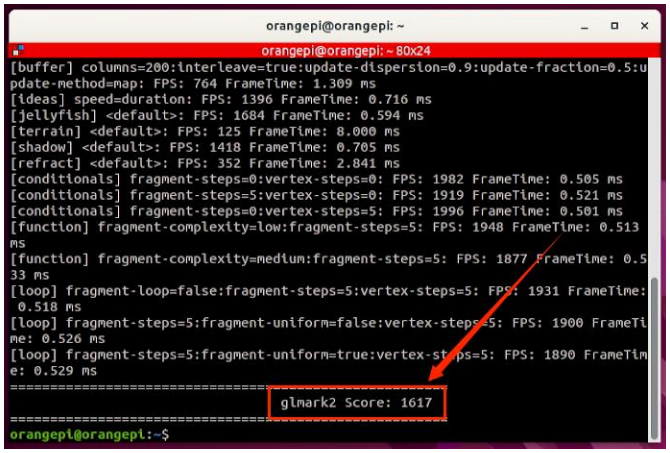
4. 4. GPU Testing Method

1) Open a terminal on the desktop and enter the **glmark2** command. If you can see **Panfrost** after **GL_VERDOR**, it means that the GPU is used.

```
orangepi@orangepi:~$ glmark2
```

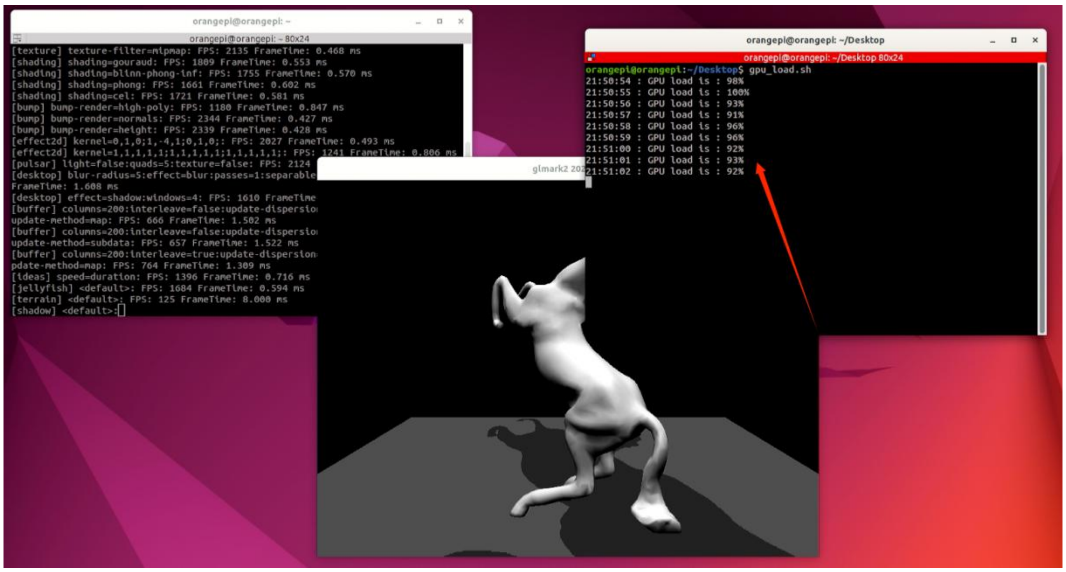


2) The glmark2 test score is usually over 1000 points



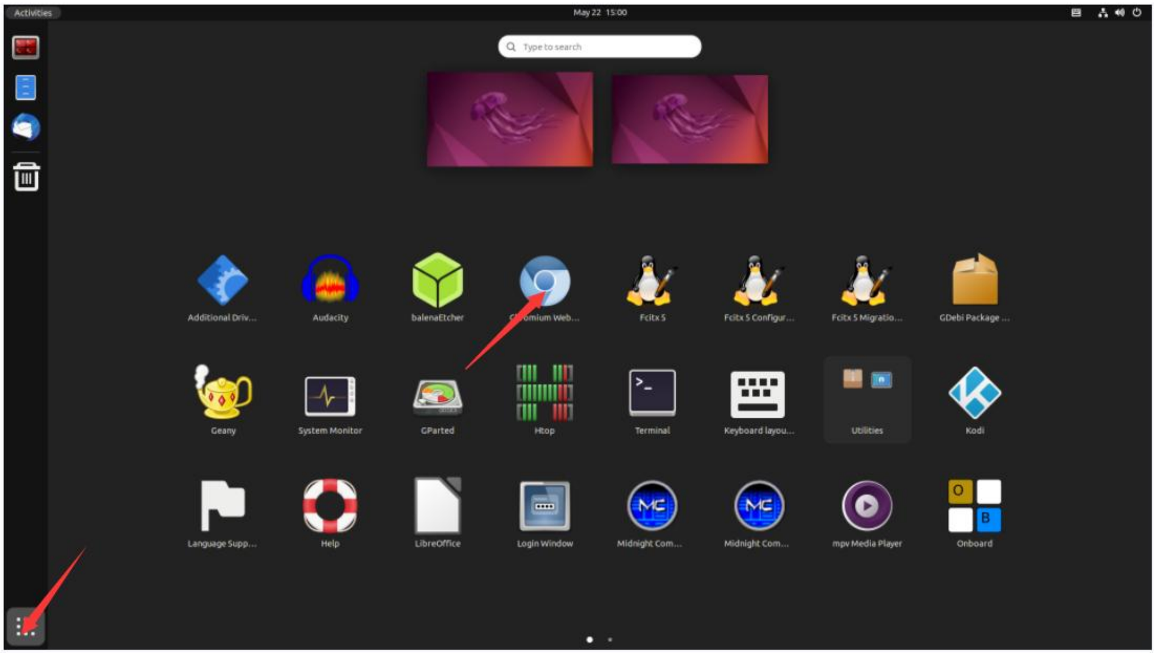
3) Run the **gpu_load.sh** script to view the current GPU load

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

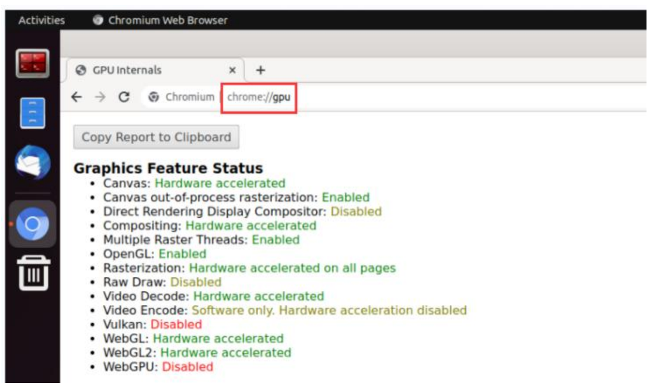


4.5. Testing method of Chromium browser hardware decoding video playback

1) First open the Chromium browser



2) Then enter **chrome://gpu** in the Chromium browser to check the support of GPU and video decoding

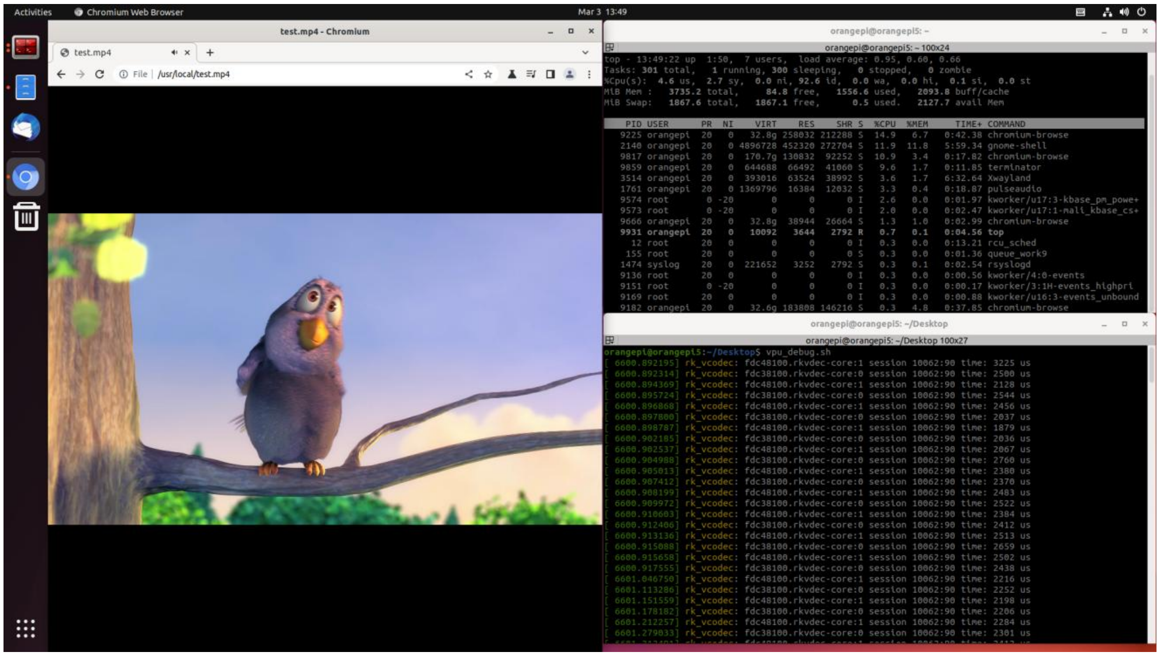


3) Then you can open a video website to play a video file, or enter the following path name in the browser to play a test video file that comes with the system

```
/usr/local/test.mp4
```

4) When playing the video, you can run the **vpu_debug.sh** script in the terminal. If there is a printout in the lower right corner of the figure below, it means that the hardware is used to decode the video.

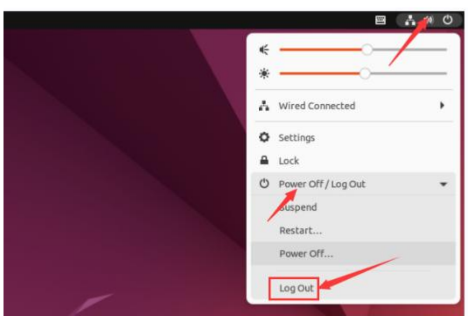
```
orangepi@orangepi:~$ vpu_debug.sh
```

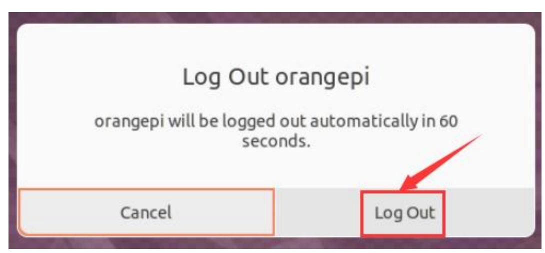


4. 6. Testing method of Kodi hard decoding video playback

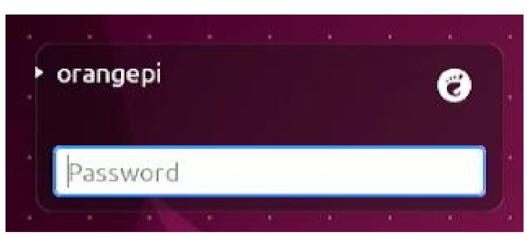
Note that there will be problems when opening Kodi directly in the Wayland desktop. Please strictly follow the method below to open Kodi.

1) First log out of the system

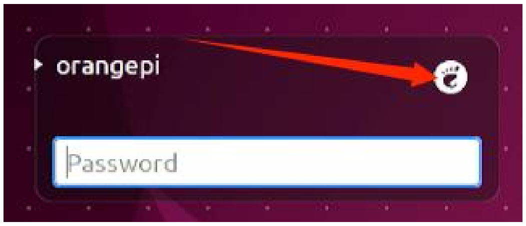




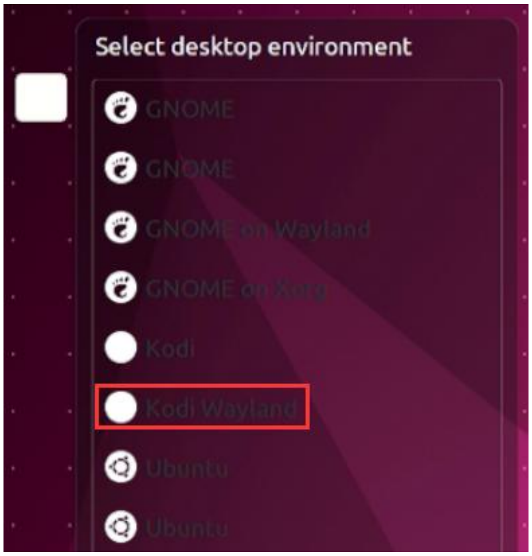
2) When you log out of the system, you will enter the following login interface



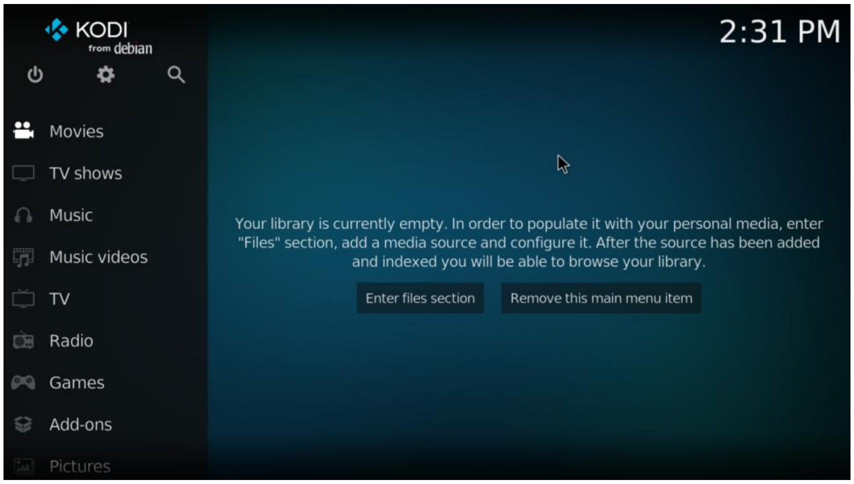
3) Then click the location shown in the picture below



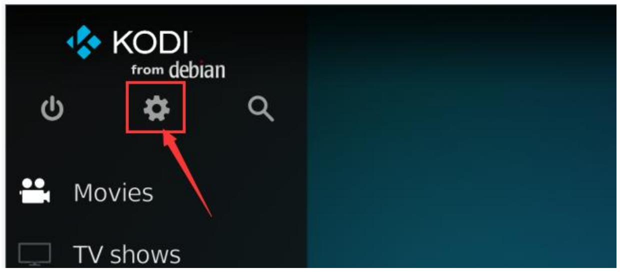
4) Then select **Kodi Wayland** and enter the password to log in to the system



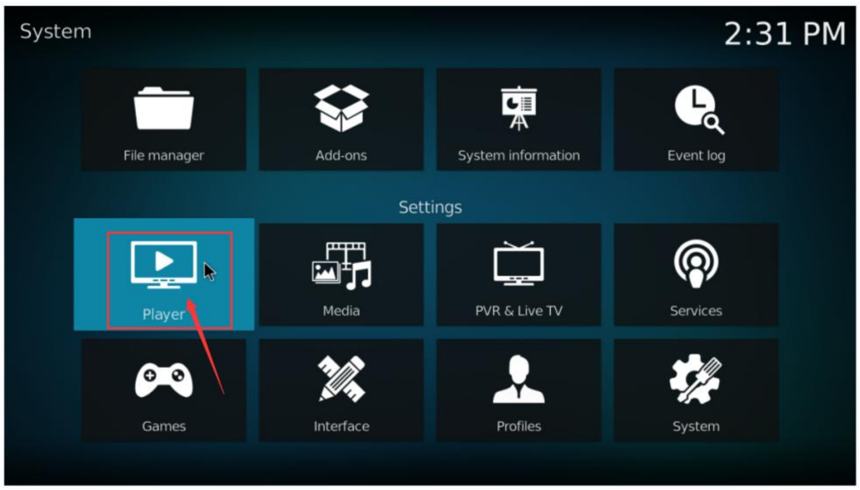
5) The interface of Kodi after opening is as follows



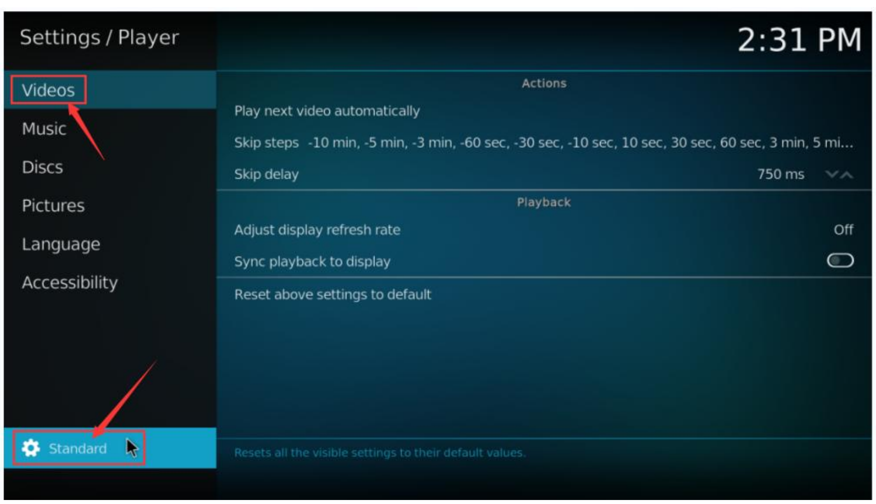
6) Then click Settings



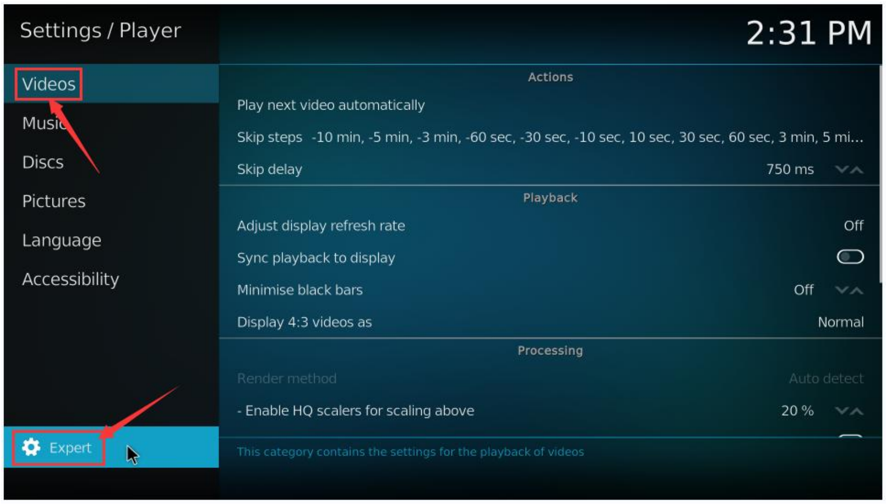
7) Then select **Player**



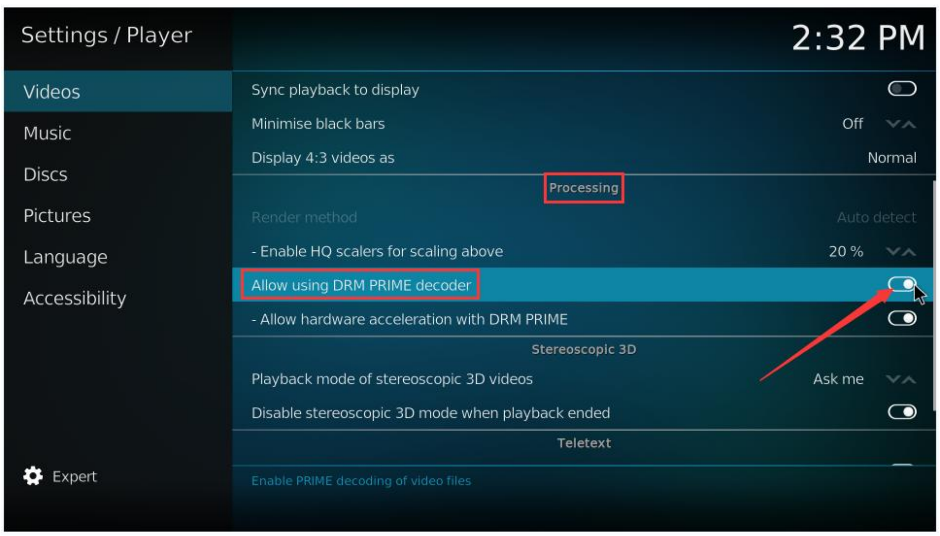
8) Then select **Videos** and click **Standard** in the lower left corner



9) Click twice to switch to **Expert** mode, as shown in the figure below

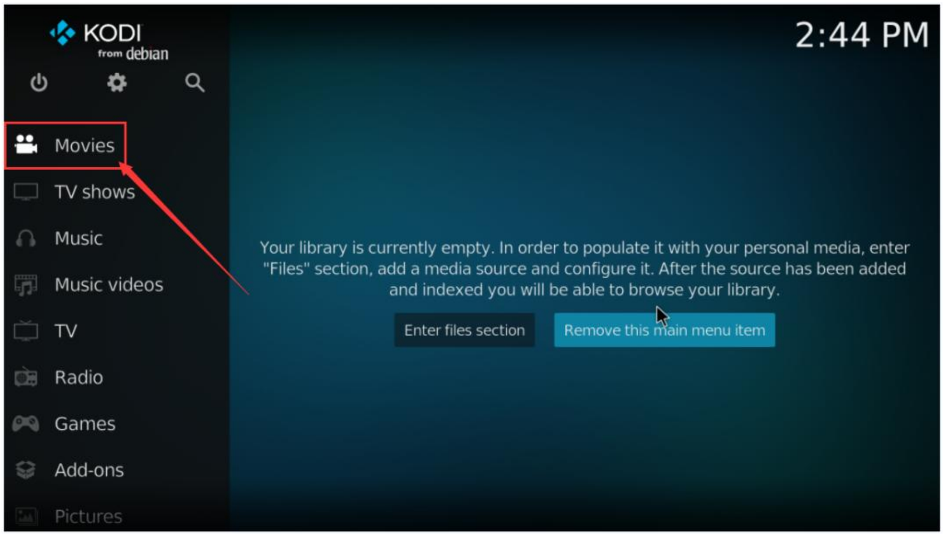


10) Then turn on **Allow using DRM PRIME decoder** in **Processing** settings

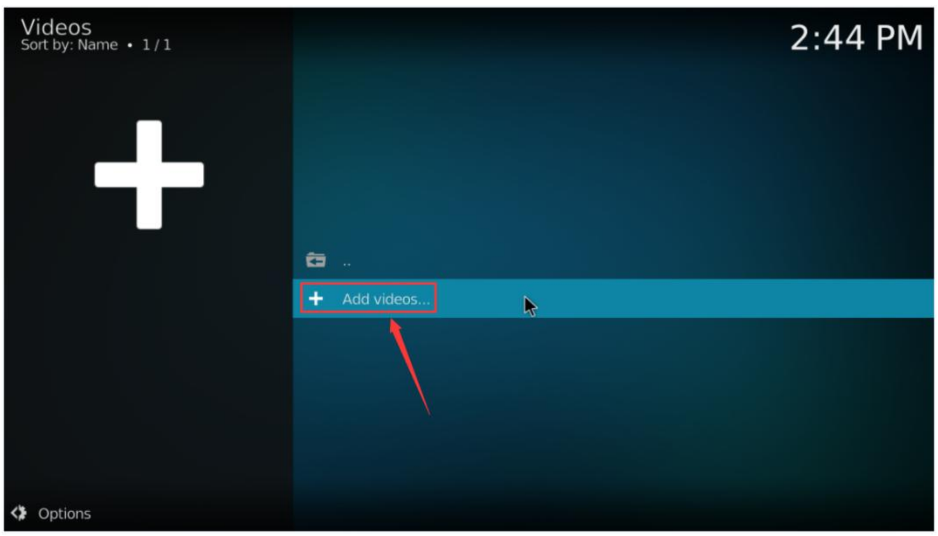


11) Then let's import a test video that comes with the system for testing. You can also upload the video you want to play to the system and then import it to play.

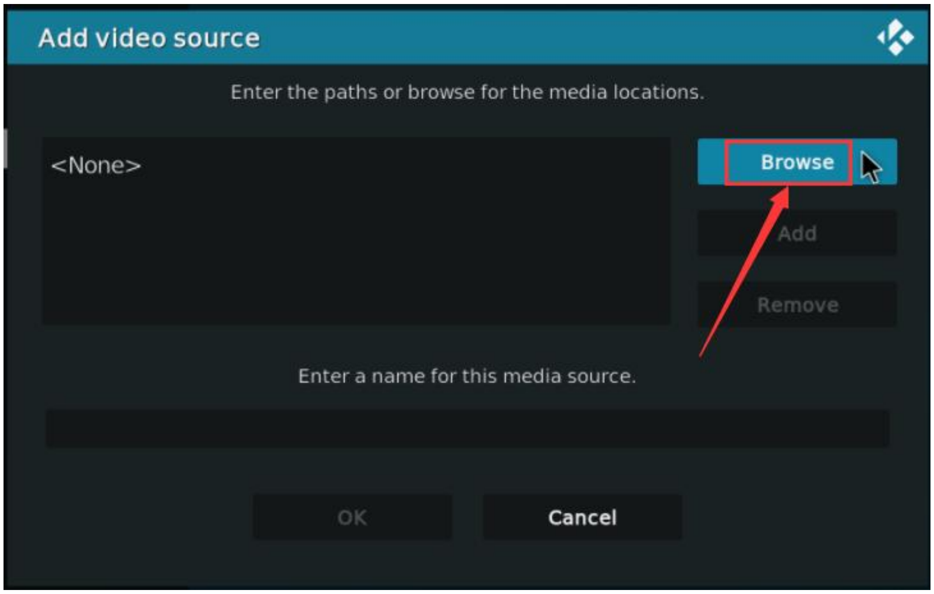
a. First enter the main interface, then select **Movies**



b. Then select **Add videos...**

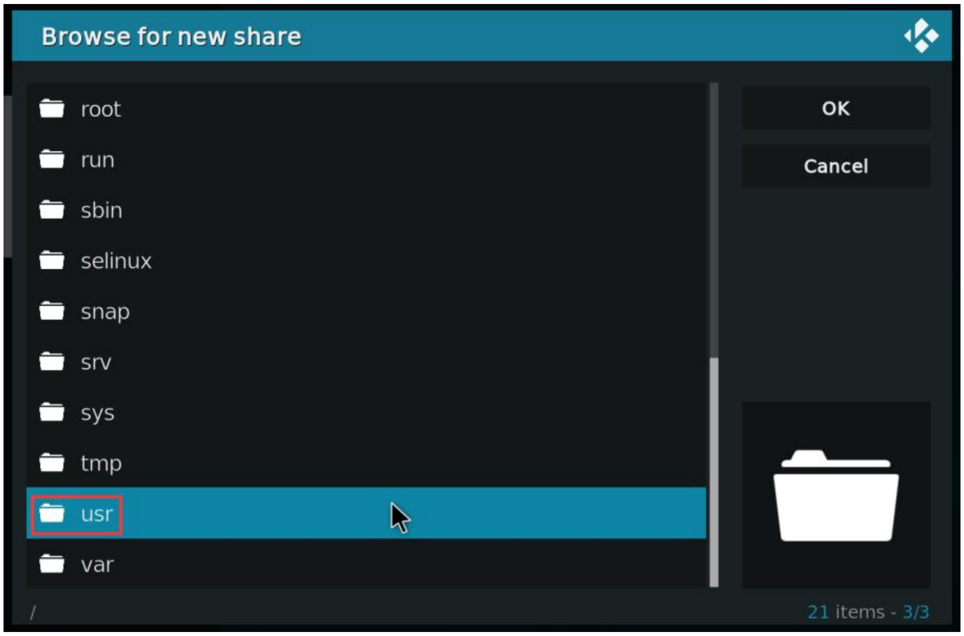


c. Then select **Browse**

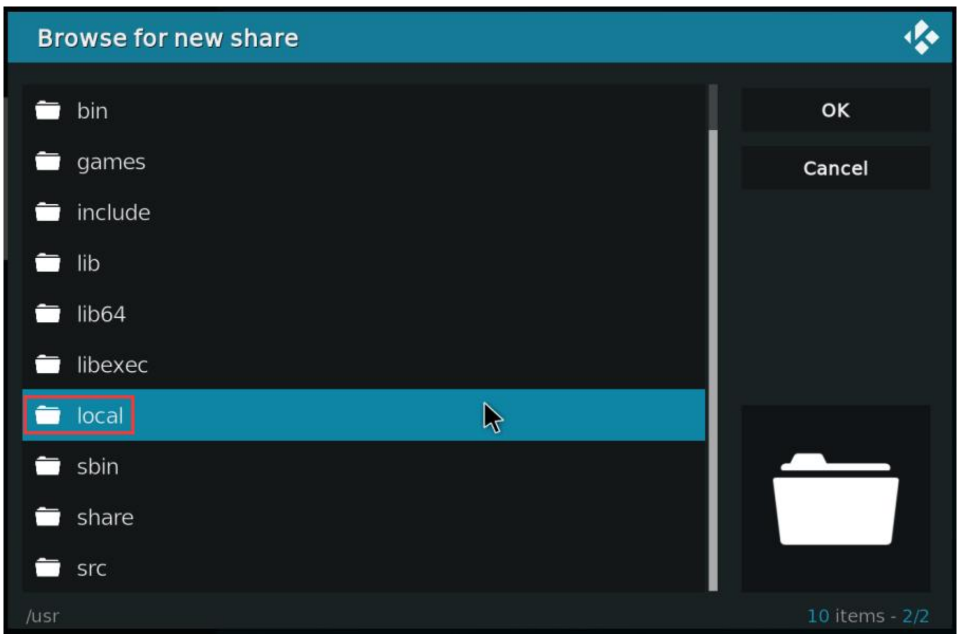


d. Then select **Root filesystem**

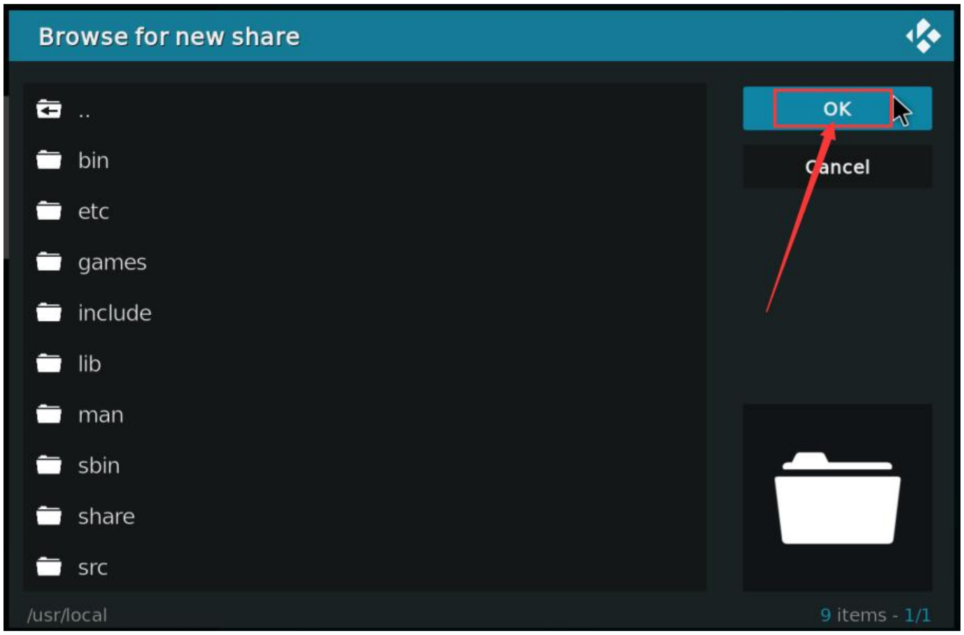
e. Then select **usr**



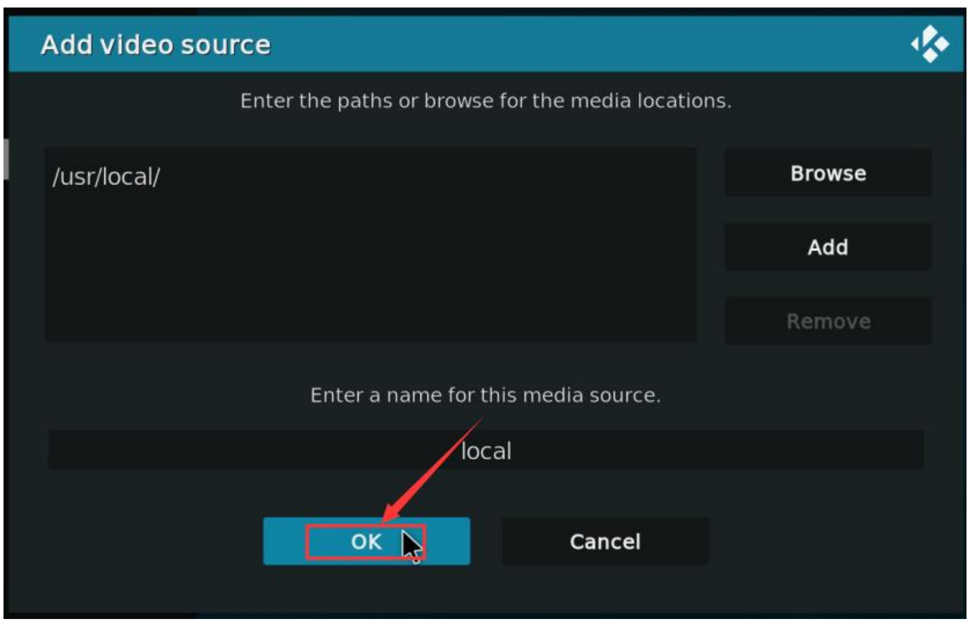
f. Then select **local**



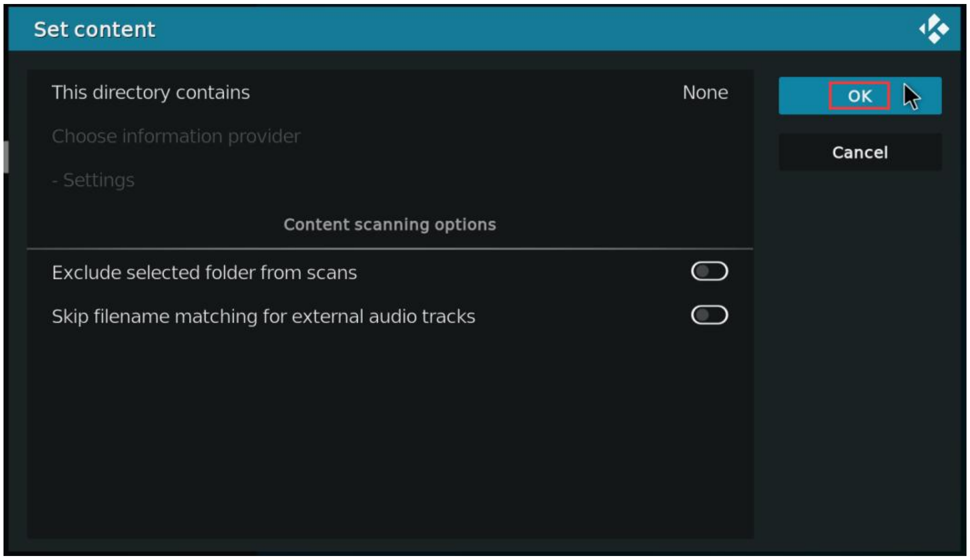
g. Then select **OK**



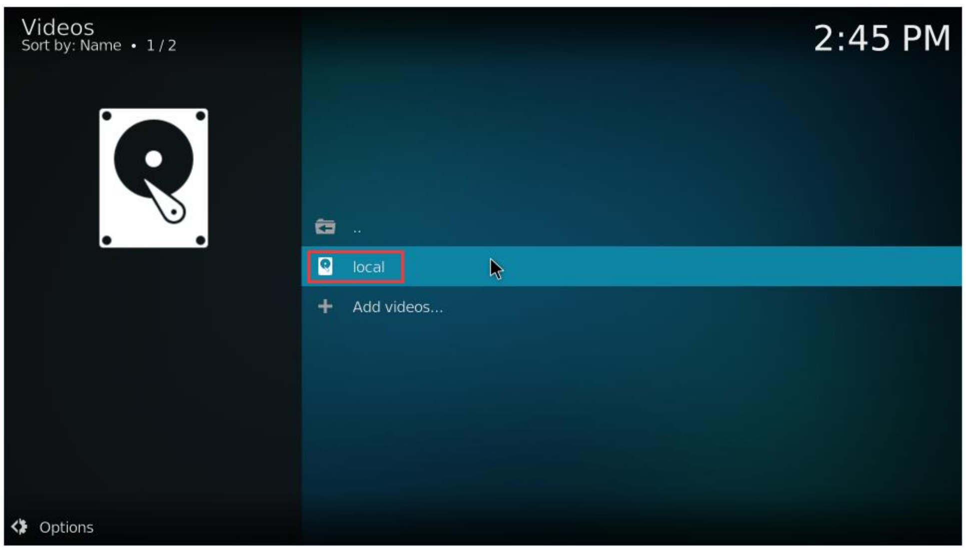
h. Then select **OK**



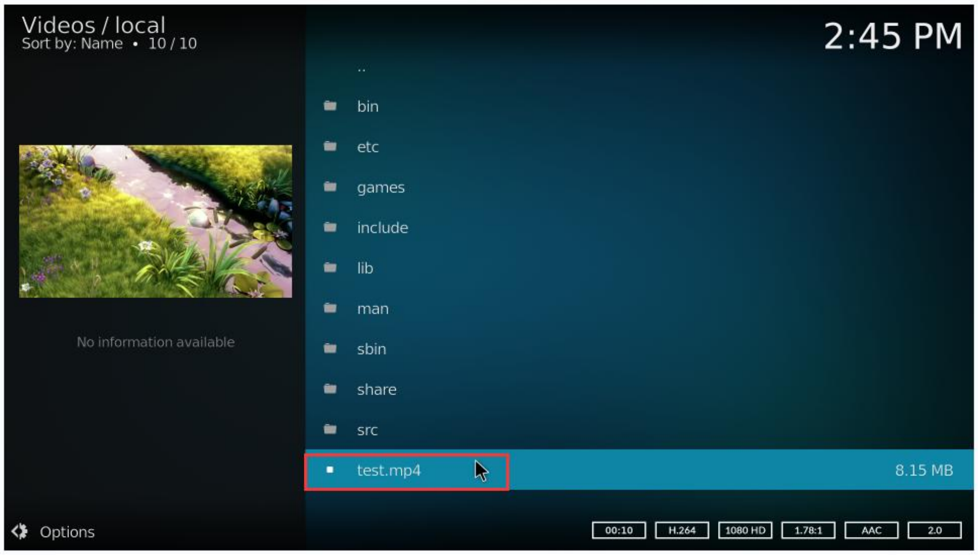
i. Then select OK



j. Then enter the local folder



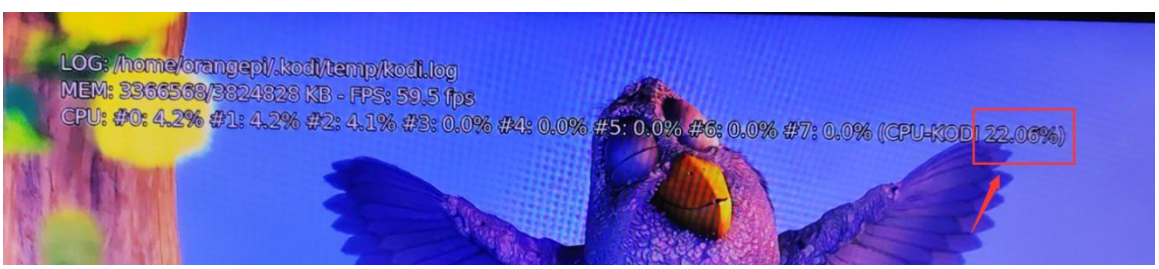
k. Then you can play the **test.mp4** test video



12) When playing the video, you can run the **vpu_debug.sh** script in the command line (via ssh or serial port). If there is the following printout, it means that the hardware is used to decode the video.

```
orangepi@orangepi:~$ vpu_debug.sh
[ 1830.938378] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2728 us
[ 1830.938461] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2617 us
[ 1830.941179] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2661 us
[ 1830.941777] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2708 us
[ 1830.944727] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 3444 us
[ 1830.945211] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 3331 us
[ 1830.970563] rk_vcodec: fdc48100.rkvdec-core:1 session 3573:2 time: 2547 us
[ 1831.199650] rk_vcodec: fdc38100.rkvdec-core:0 session 3573:2 time: 2703 us
```

13) The CPU usage when playing the **test.mp4** video file is around **20%~30%**.



4. 7. How to install ROS 2 Humble on Ubuntu 22.04 Gnome

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.

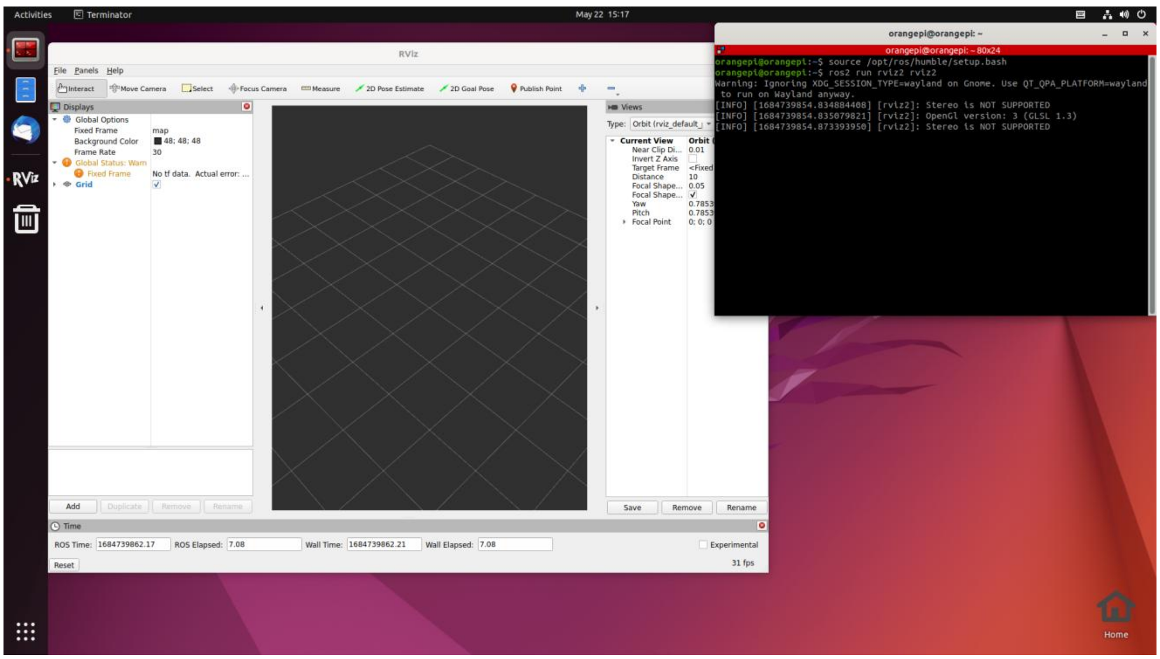
```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

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@orangepi5:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/humble/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```

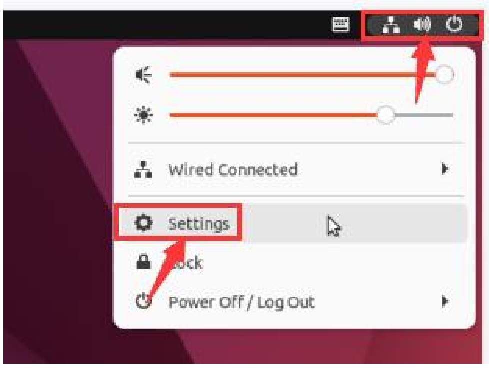


5) Reference Documents

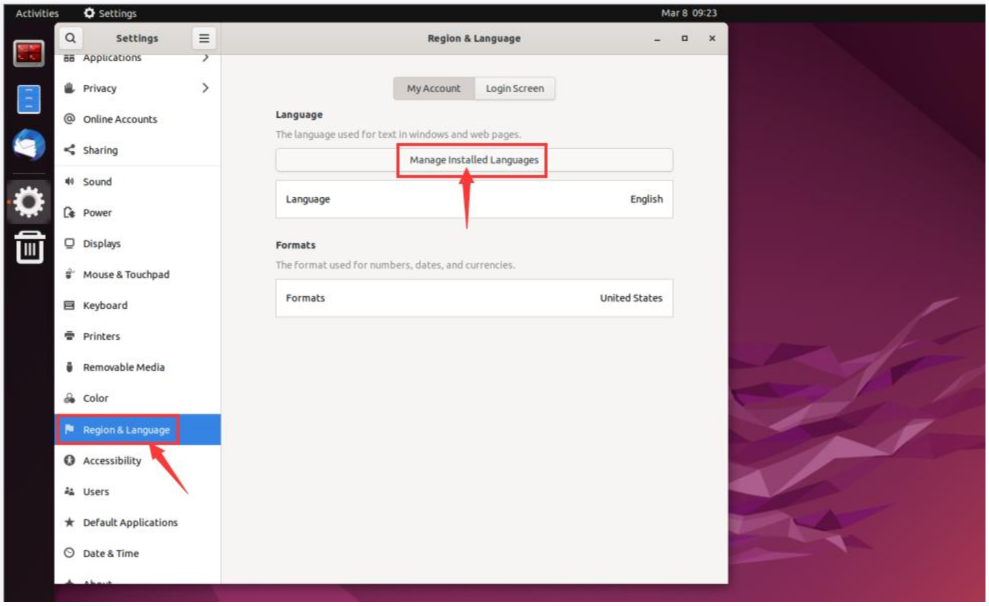
http://docs.ros.org/en/humble/index.html
http://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

4. 8. How to set up the Chinese environment and install the Chinese input method

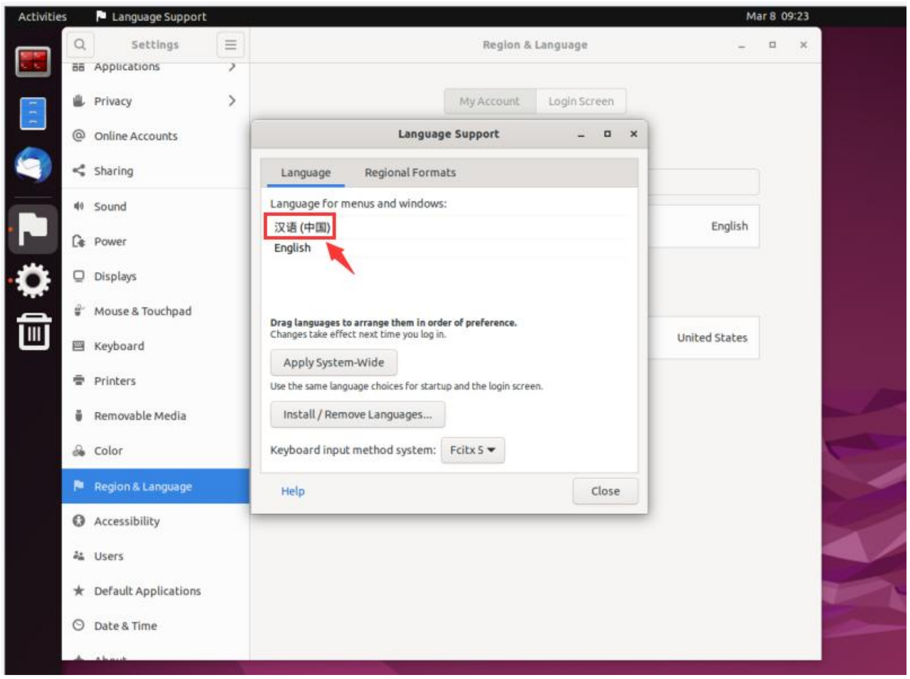
1) First click Settings



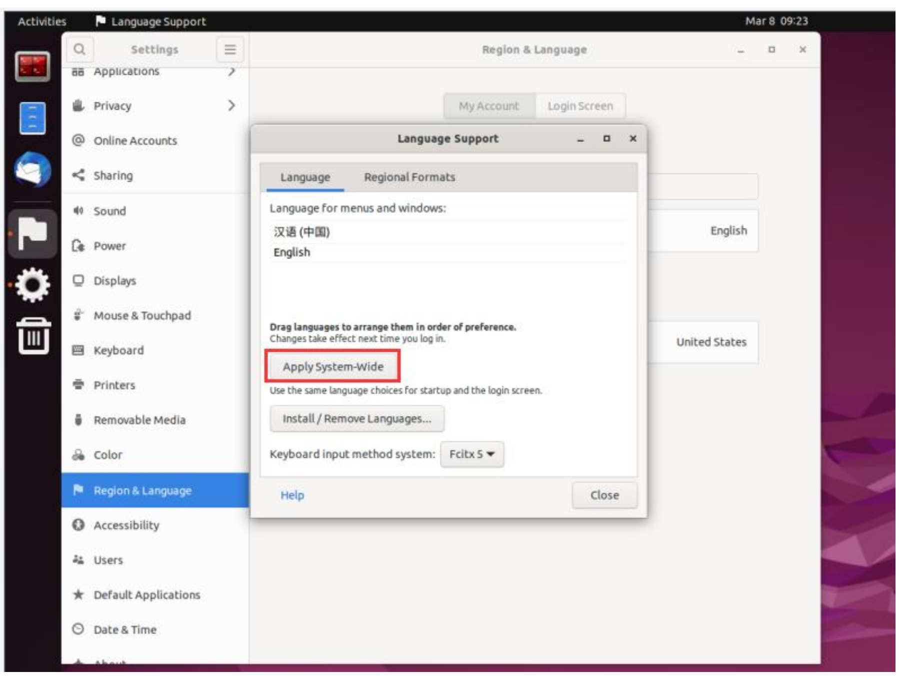
2) Then find the **Region & Language** option and click the **Manage Installed Languages** option



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



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

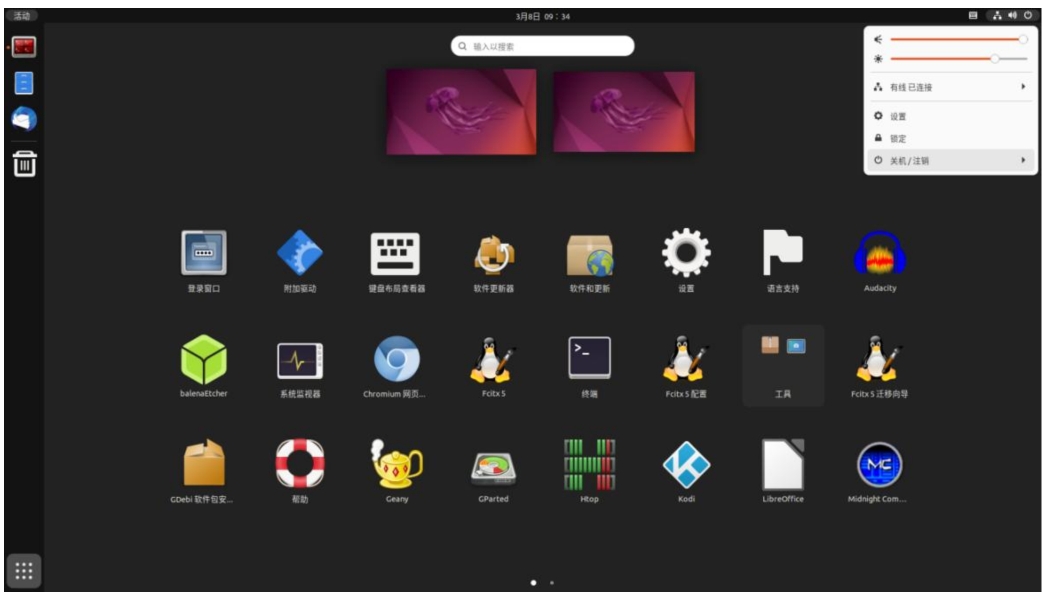


5) **Then restart the Linux system to make the configuration take effect**

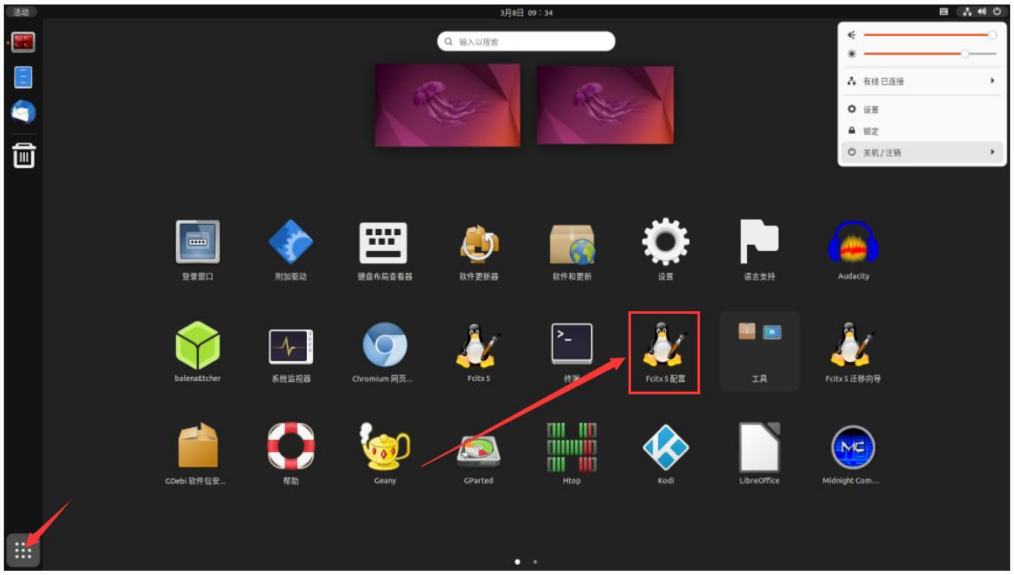
6) After re-entering the system, please select "**Do not ask me again**" in the following interface, and then decide whether to update the standard folder to Chinese according to

your own preferences.

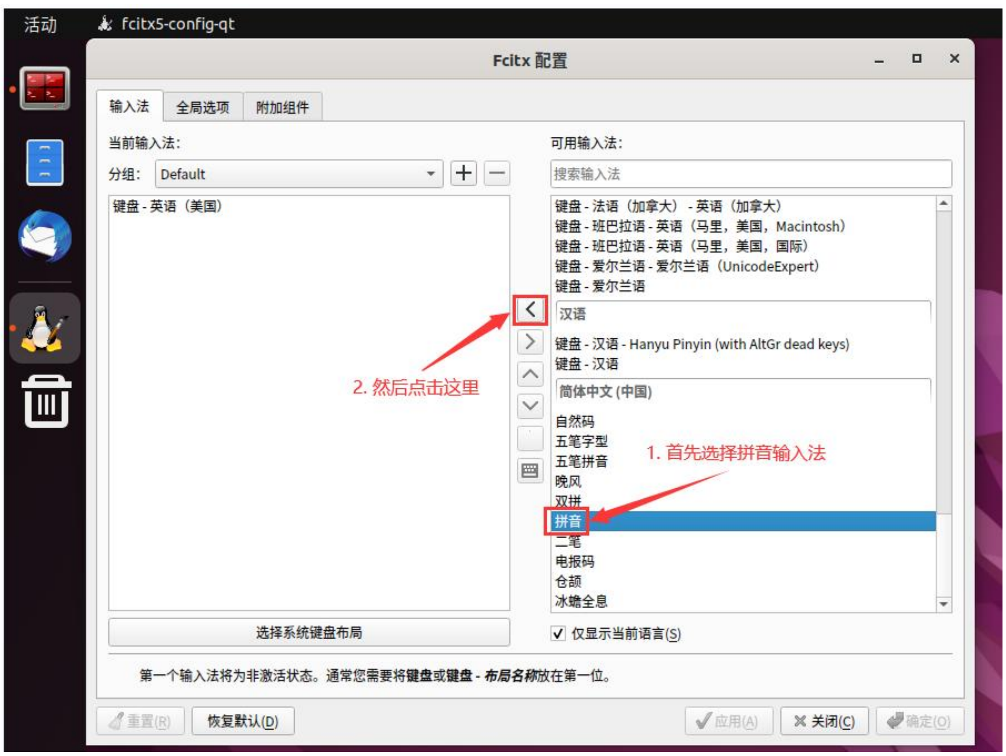
7) Then you can see that the desktop is displayed in Chinese



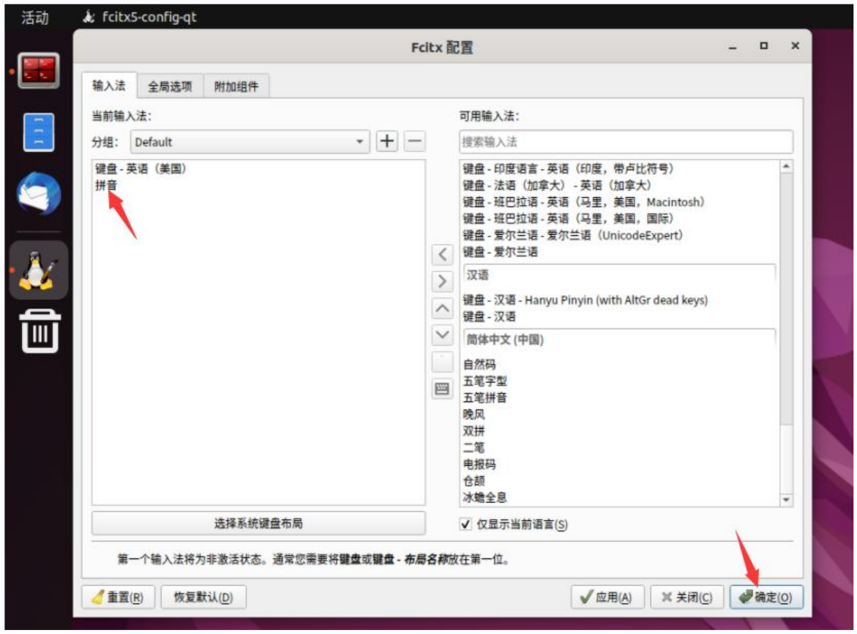
8) Then open the Fcitx5 configuration program



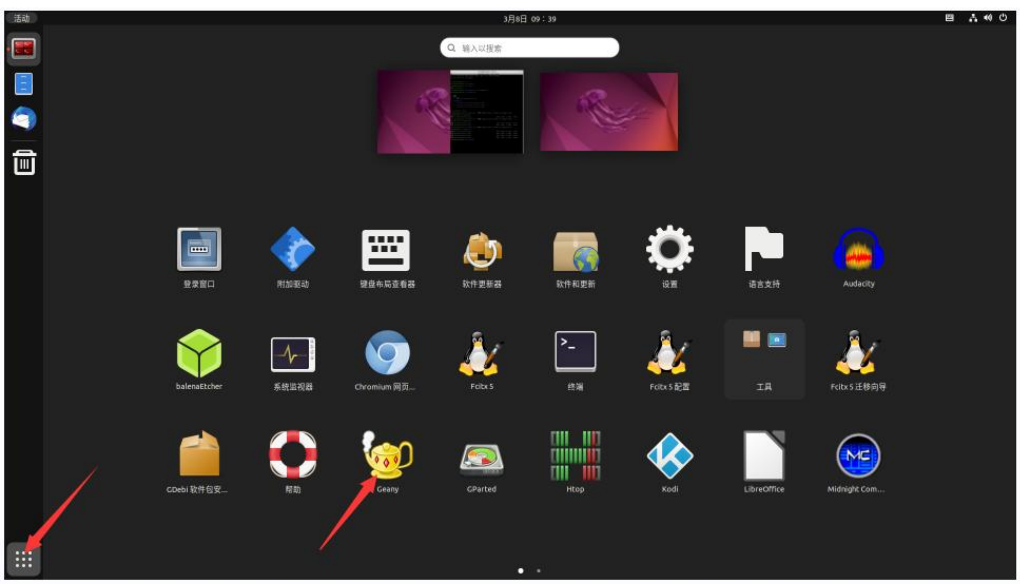
9) Then select Pinyin input method



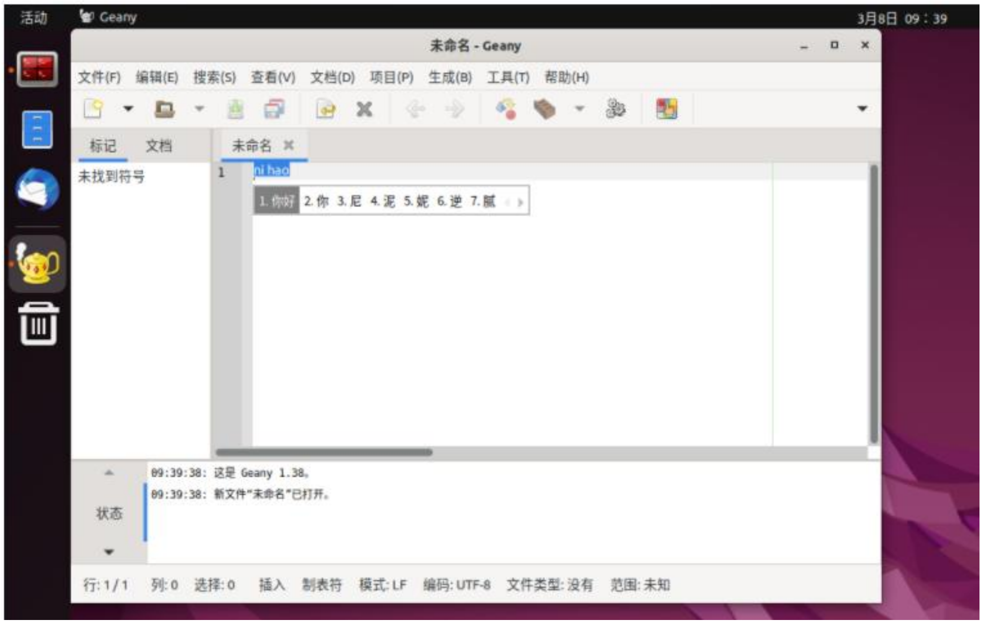
10) The interface after selection is as shown below, then click OK



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 by pressing **Ctrl+Space**, and then we can input Chinese.


活动
Geany
3月8日 09：39
未命名 - Geany
文件(F) 编辑(E) 搜索(S) 查看(V) 文档(D) 项目(P) 生成(B) 工具(T) 帮助(H)
标记 文档
未命名
未找到符号
ni hao
1. 你好 2. 你 3. 尼 4. 泥 5. 妮 6. 逆 7. 腻
09:39:38: 这是 Geany 1.38。
09:39:38: 新文件"未命名"已打开。
状态
行: 1 / 1 列: 0 选择: 0 插入 制表符 模式: LF 编码: UTF-8 文件类型: 没有 范围: 未知

5. Orange Pi OS Arch system usage instructions

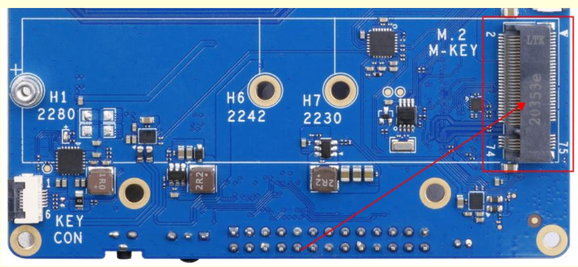
5. 1. Orange Pi OS Arch system compatibility

Function	OPi OS Arch Gnome Wayland
HDMI Display	OK
HDMI Audio	OK
USB 2.0	OK
USB 3.0	OK
WIFI	OK
Bluetooth	OK
Debug serial port	OK
FAN	OK
eMMC boot	OK
GPIO（26pin）	OK
UART（26pin）	OK
SPI（26pin）	OK
I2C（26pin）	OK
PWM（26pin）	OK
Camera1	OK
Camera2	OK
Camera3	OK
LCD Display	OK
LCD Touch	OK
Onboard MIC	OK
Headphone playback	OK
Headphone Recording	OK
speaker x 2	OK
LED Light	OK
Type-C to USB 3.0	OK
Type-C interface DP display	OK

Type-C interface DP audio	**OK**
TF card boot	**OK**
NVMe SSD Identification	**OK**
SATA SSD Identification	**OK**
Battery	**OK**
Infrared	**OK**
GPU	**OK**
NPU	**NO**
VPU	**OK**
Power button	**OK**
Watchdog test	**OK**

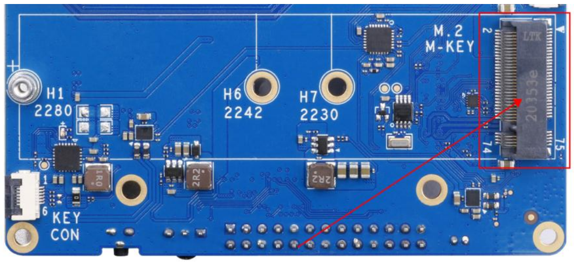
5. 2. How to use SATA SSD in OPi OS Arch system

The M.2 interface shown in the figure below can use either NVMe SSD or SATA SSD. Since the PCIe2.0 controller and SATA controller are optional, only one of them can be enabled at the same time. The OPi OS Arch image released by Orange Pi opens the PCIe configuration by default, so it can only recognize NVMe SSD by default. If you want to use SATA SSD, you need to enable the corresponding configuration.



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

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



3) SATA SSD is currently mainly used as an extended storage device.

4) Then add the following configuration in **/boot/extlinux/extlinux.conf**.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-cm5-tablet.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-ssd-sata0.dtbo #Configuration that needs to be added
```

5) **Then restart the OPi OS Arch system**

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

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

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

....

5. 3. How to use 10.1 inch MIPI LCD screen

5. 3. 1. 10.1-inch MIPI screen assembly method

1) First prepare the necessary accessories

a. 10.1 inch MIPI LCD display + touch screen.

b. Screen adapter board + 31pin to 26pin cable.

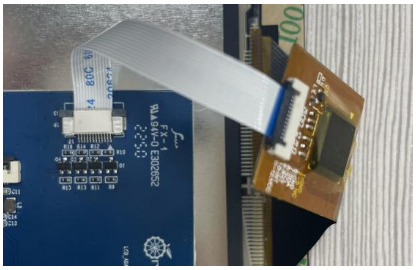
c. 30pin MIPI cable.

d. 12pin touch screen cable.

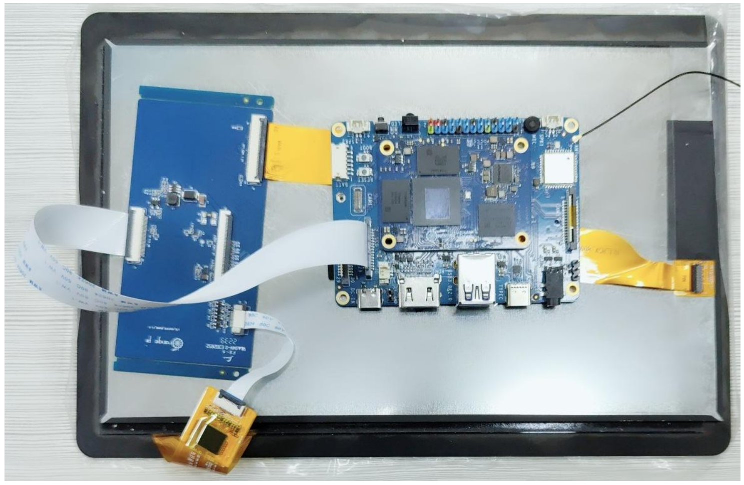
2) Connect the 12-pin touch screen cable, 31-pin to 26-pin cable, and 30-pin MIPI cable to the screen adapter board as shown below. Note that **the blue insulation side of the touch screen cable should face down**, and the insulation sides of the other two cables should face up. If they are connected incorrectly, there will be no display or touch failure.

3) Place the adapter board with the connected cables on the MIPI LCD screen as shown below, and connect the MIPI LCD screen and the adapter board via a 31-pin to 26-pin cable.

4) Then connect the touch screen and the adapter board through the 12-pin touch screen cable, paying attention to the direction of the insulating surface.



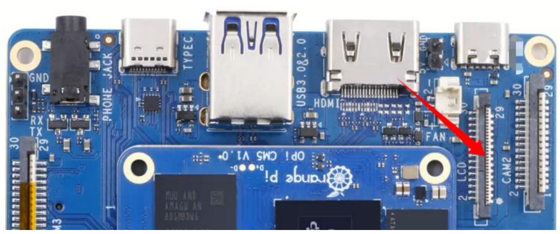
5) Finally, connect it to the LCD interface of the development board via a 30-pin MIPI cable.



5. 3. 2. How to open the 10.1-inch MIPI LCD screen configuration

1) The OPi OS Arch image does not have the mipi LCD screen configuration turned on by default. If you need to use the mipi LCD screen, you need to turn it on manually.

2) The interface of the mipi lcd screen on the development board is shown in the figure below:

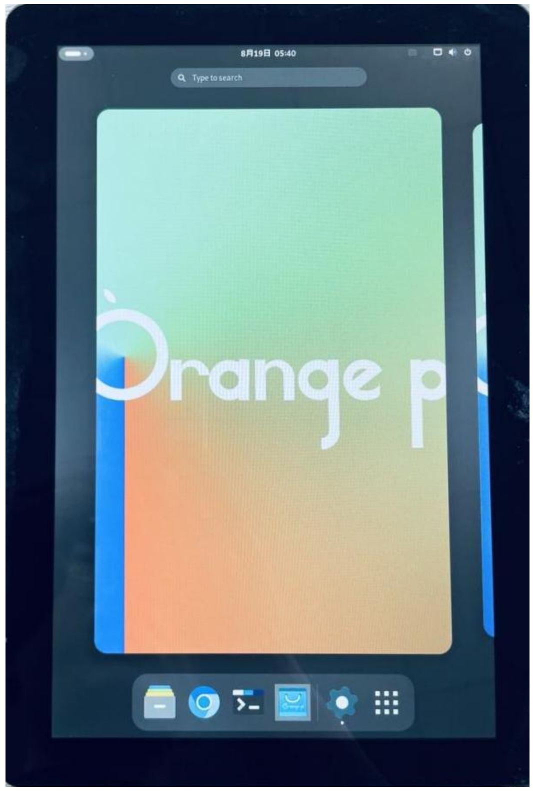


3) The method to open the mipi lcd configuration is as follows:

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-5-pro.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-opicm5-tablet-lcd.dtbo #Configuration that needs to be added
```

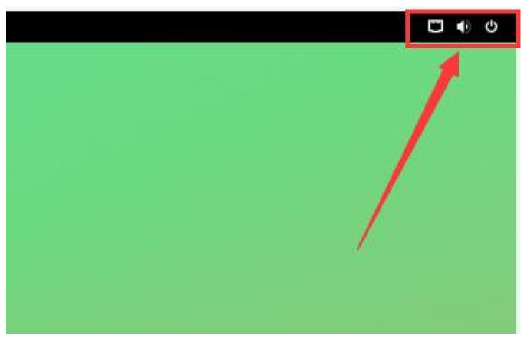
4) **Then restart the OPi OS Arch system.**

5) After restarting, you can see the LCD screen display as follows (the default is vertical):

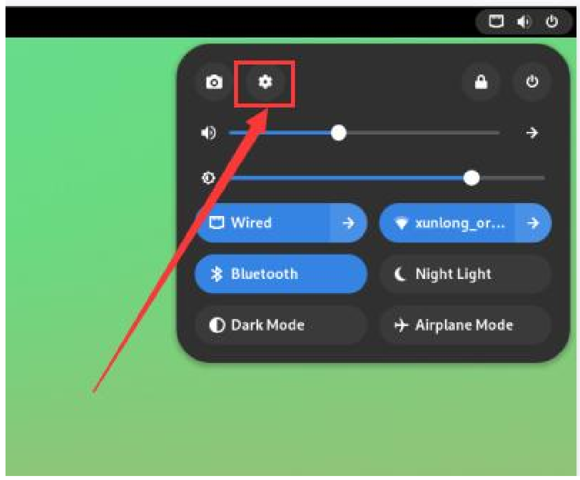


5. 3. 3. Methods for rotating display and touch direction

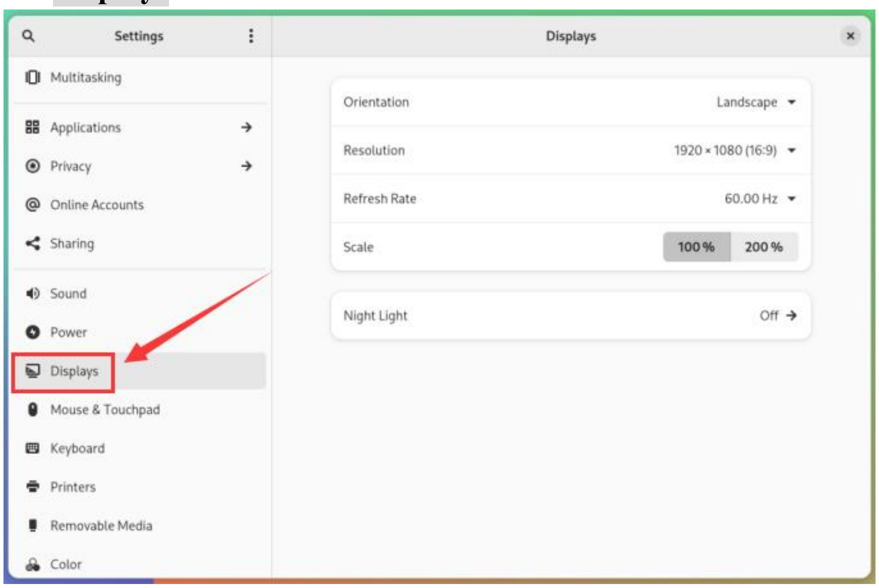
1) First click on the area in the upper right corner of the desktop.



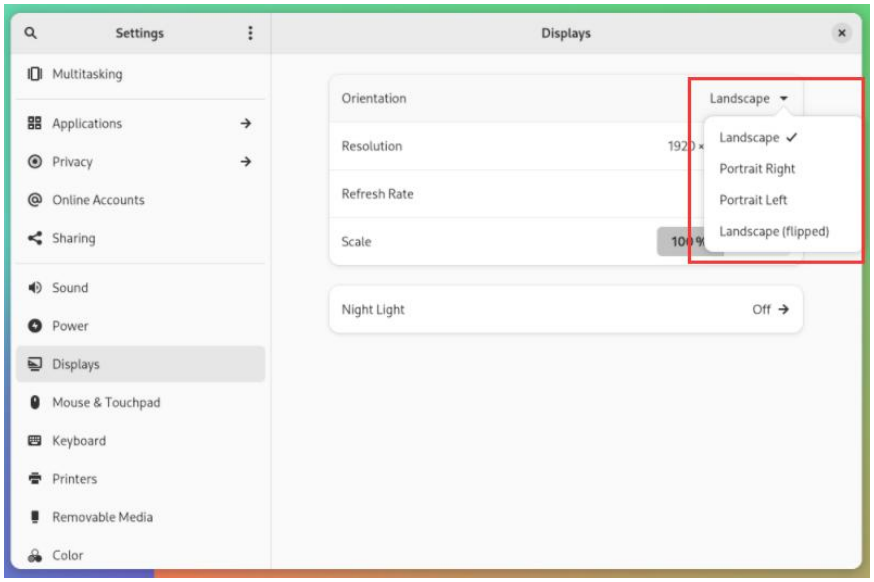
2) Then click Settings.



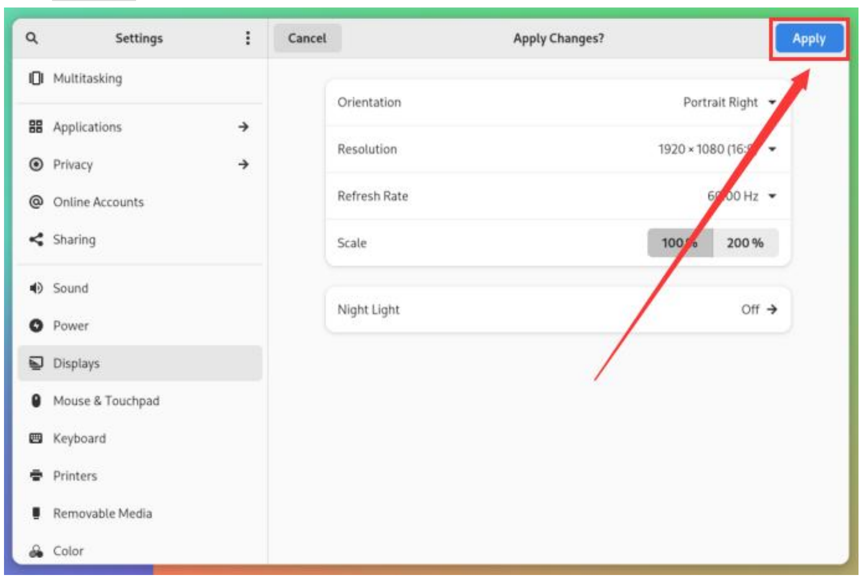
3) Then select**Displays**。



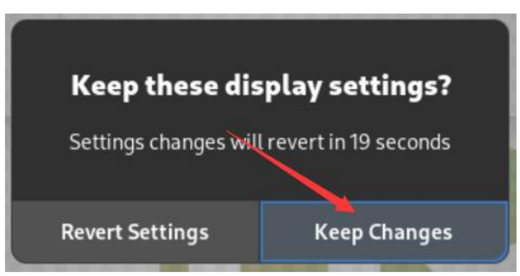
4) Then select the direction you want to rotate in **Orientation** of **Displays**.



5) Then select **Apply**。



6) Then you can see that the screen has been rotated. At this time, you need to select **Keep Changes** to finalize the rotation.



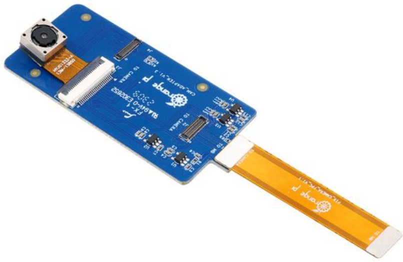
7) The LCD screen will display the following after rotating 90 degrees:

8) The touch function of the LCD screen of the OPi OS Arch system will rotate as the display direction rotates, and no other settings are required.

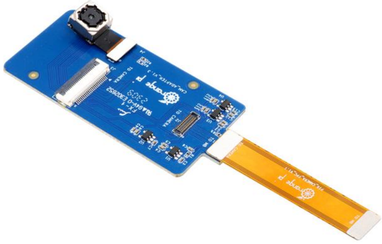
5. 4. Testing Method for OV13850 and OV13855 MIPI Cameras

At present, the development board supports two MIPI cameras, OV13850 and OV13855. The specific images are shown below:

a. OV13850 camera with 13 million MIPI interface.



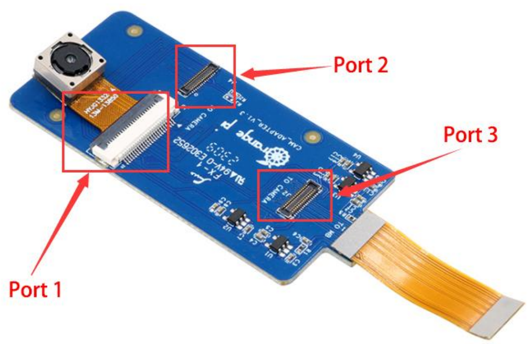
b. OV13855 camera with 13 million MIPI interface.



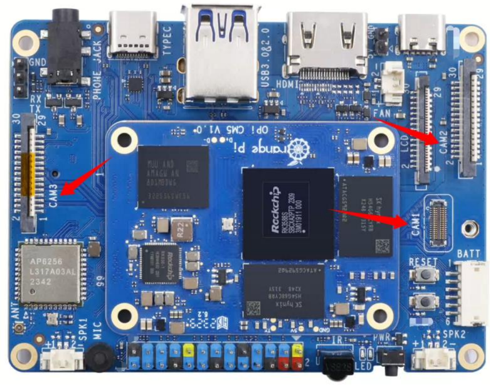
The adapter board and FPC cable used for OV13850 and OV13855 cameras are the same, except that the two cameras are connected to the adapter board in different positions. The FPC cable is shown in the following figure. Please note that the FPC cable has a direction, and the end marked as **TO MB** needs to be plugged into the camera interface of the development board, while the end marked as **TO CAMERA** needs to be plugged into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board, and only one can be connected for use at a time, as shown in the following figure. Among them:

a. **Interface 1 is connected to the OV13850 camera.**
b. **Interface 2 is connected to the OV13855 camera.**
c. Interface 3 is not in use, just ignore it.



There are a total of 3 camera interfaces on the Orange Pi CM5 Base Tablet development board, and only CAM1 can be used to connect to OV13850 or OV13855 cameras. Cam2 and Cam3 are currently not compatible with specific camera models. We define the positions of Cam1, Cam2, and Cam3 as shown in the following figure:



The method of inserting the camera into the Cam1 interface of the development board is as follows:

After connecting the camera to the development board, we can use the following method to test the camera:

a. First, add the following configuration to **/boot/extlinux/extlinux.conf**.

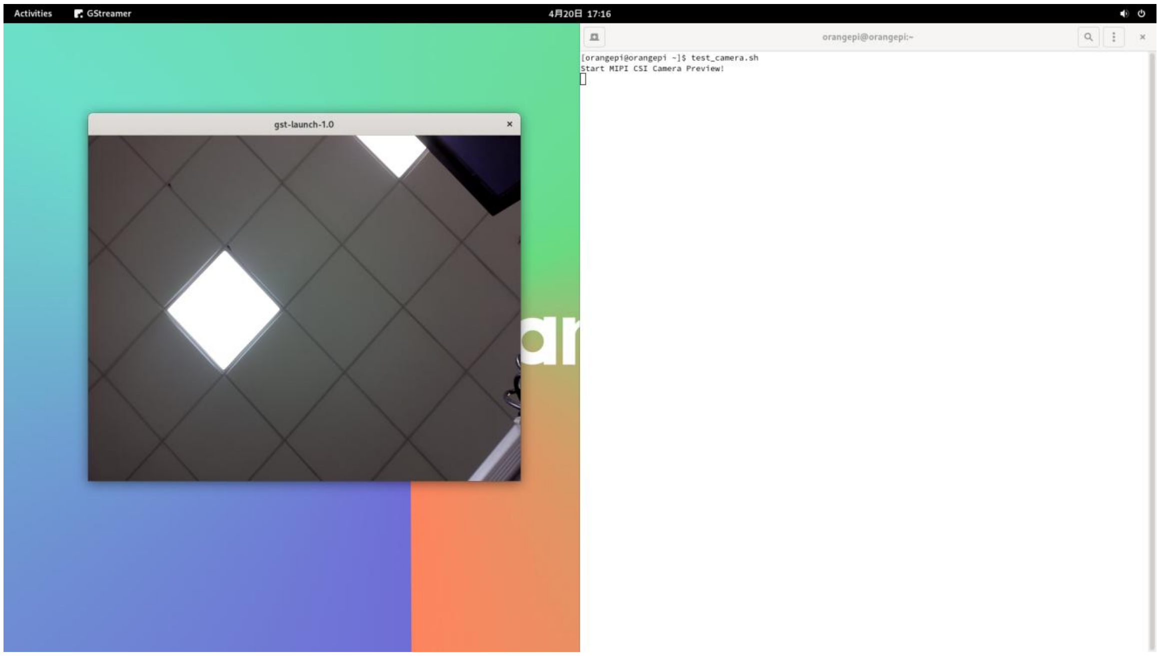
```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-cm5-tablet.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-opicm5-tablet-cam1.dtbo #Configuration that needs to be added
```

b. Then restart the OPi OS Arch system.

c. Then open a terminal in the desktop system and run the following script.

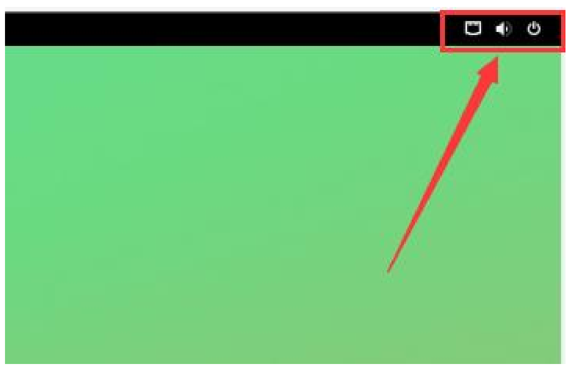
```
orangepi@orangepi:~$ test_camera.sh
```

d. Then you can see the preview screen of the camera.

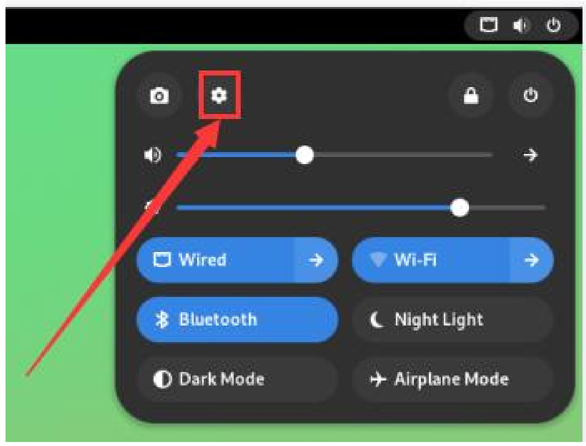


5. 5. Methods for Setting up Chinese Environment and Installing Chinese Input Method

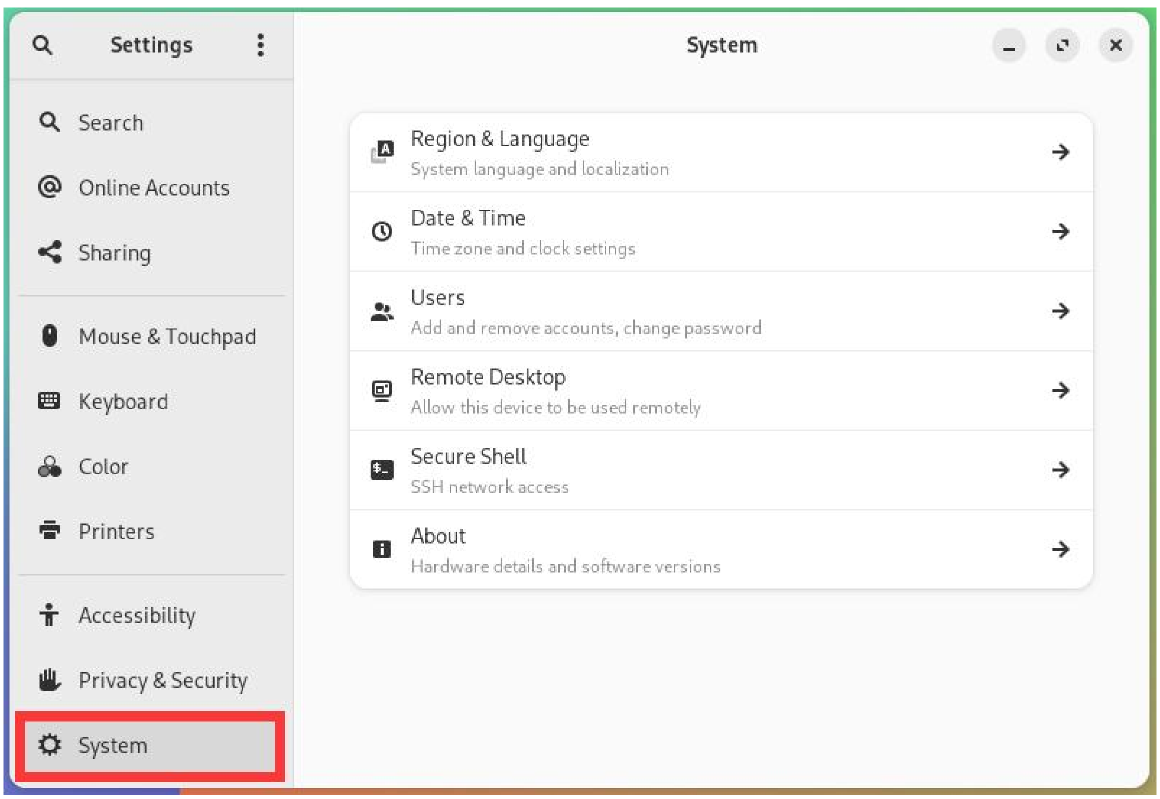
1) First, click on the area in the upper right corner of the desktop.



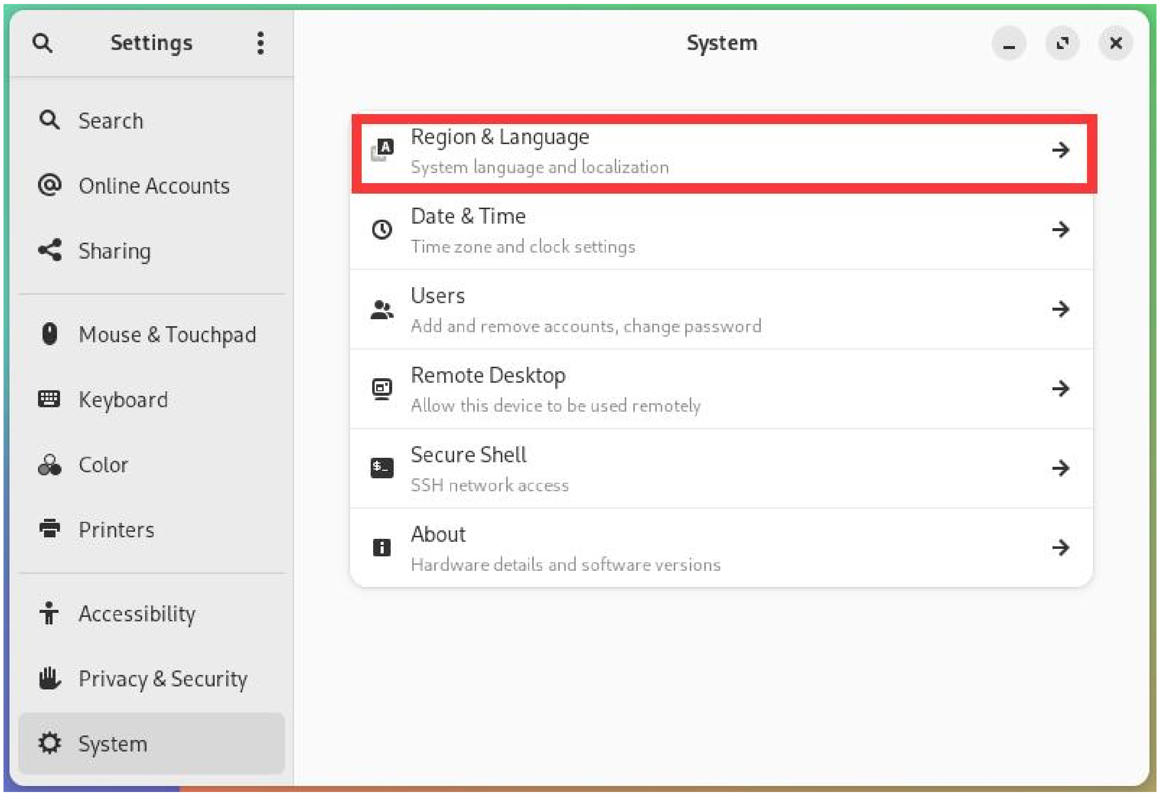
2) Then open the settings.



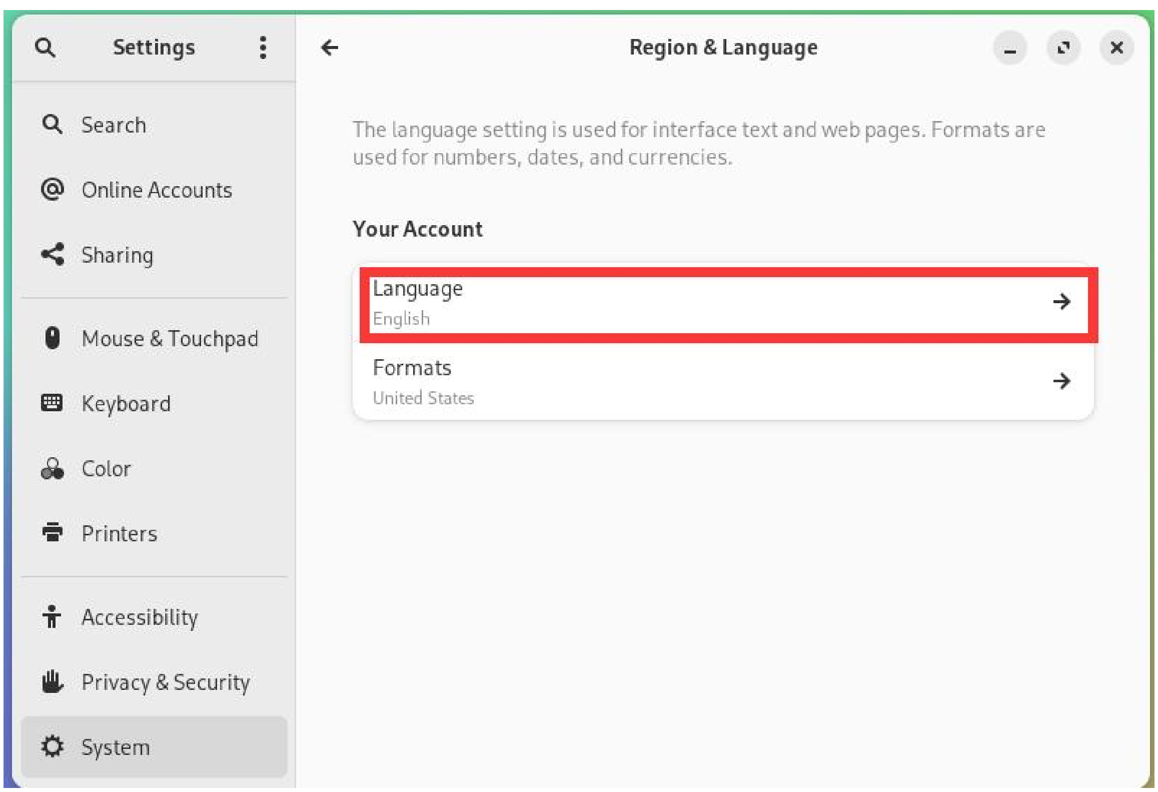
3) Then find the **System** option.



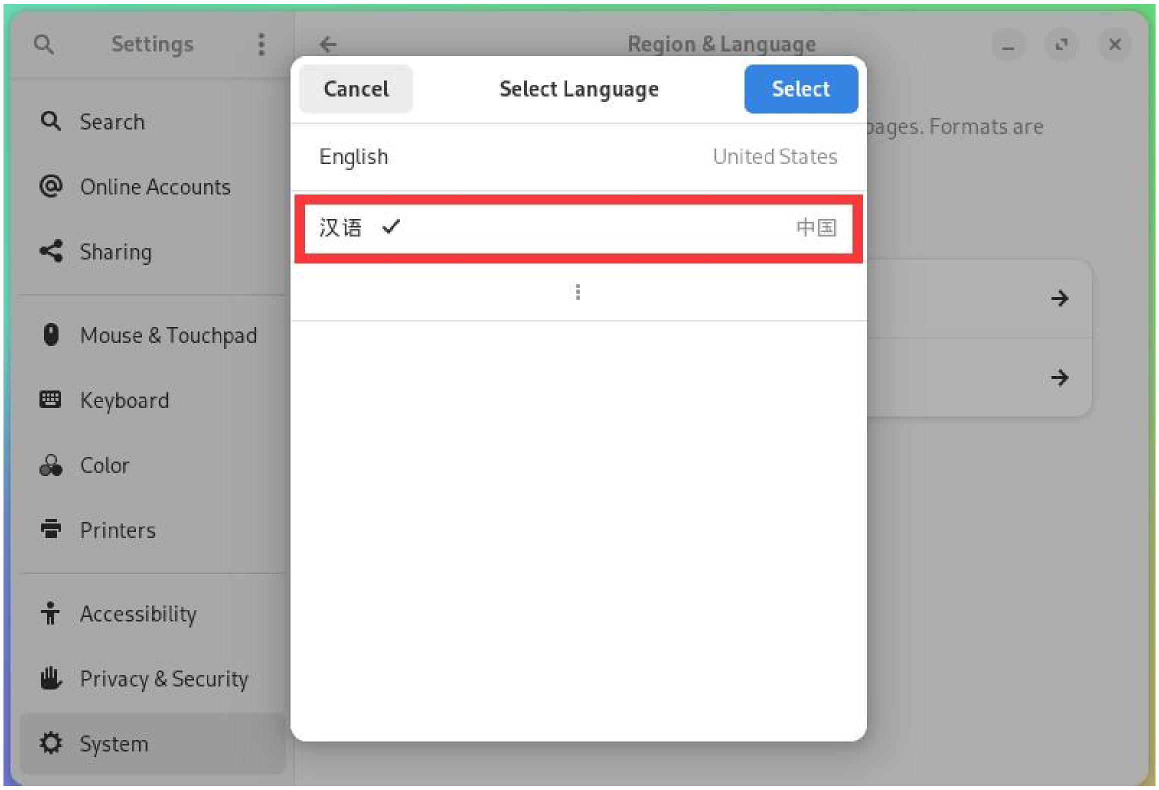
4) Then find the **Region & Language** option.



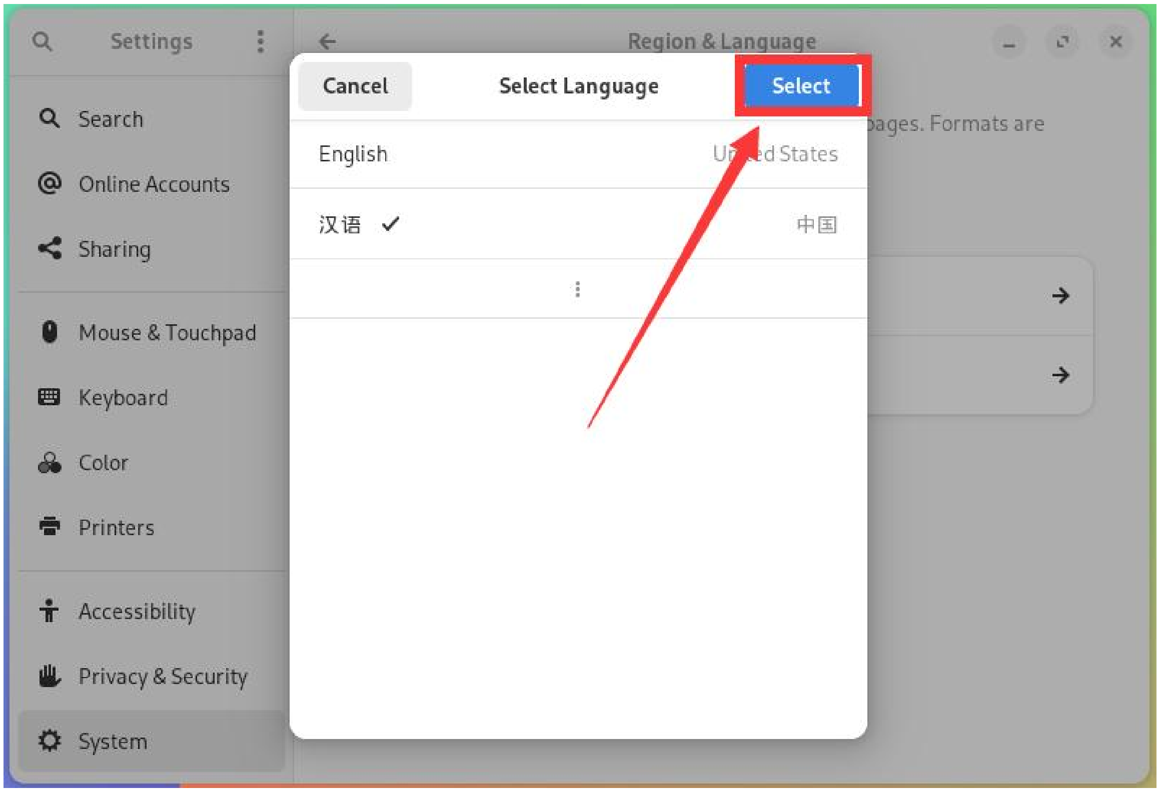
5) Then select **Language**。



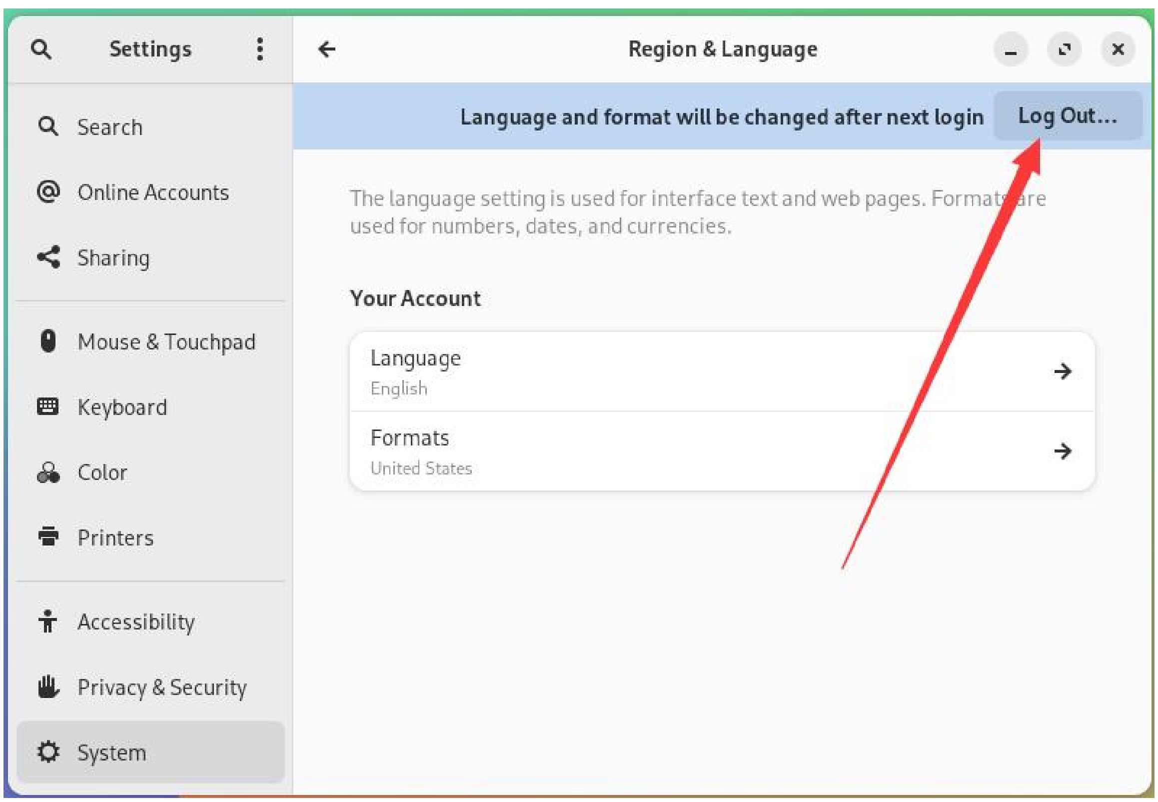
6) Then choose Chinese.



7) Then click **Select**。



8) Then click on **Log Out...** Log out of the system and log in again.

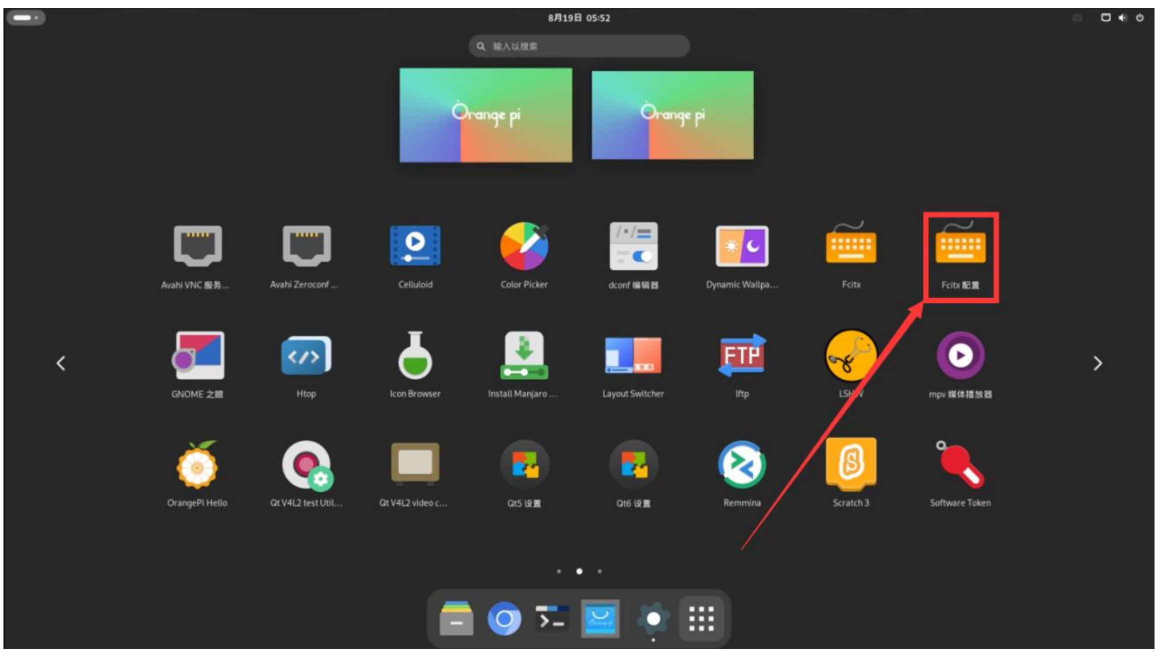


9) Then you can see that the desktop is all displayed in Chinese.

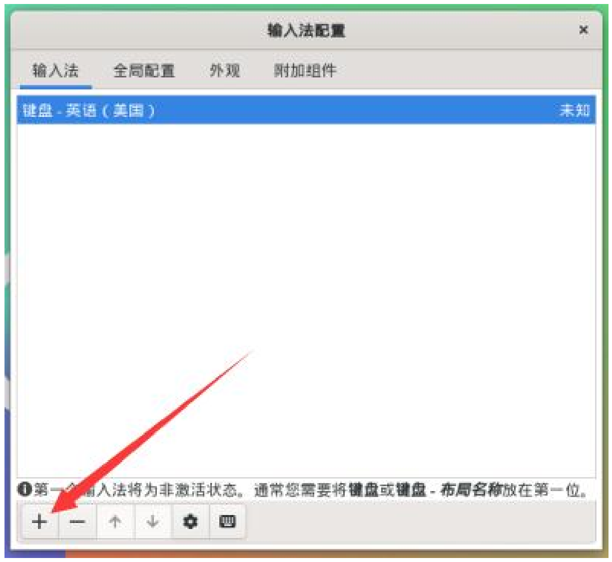
10) Then install fcitx im and fcitx configtool.

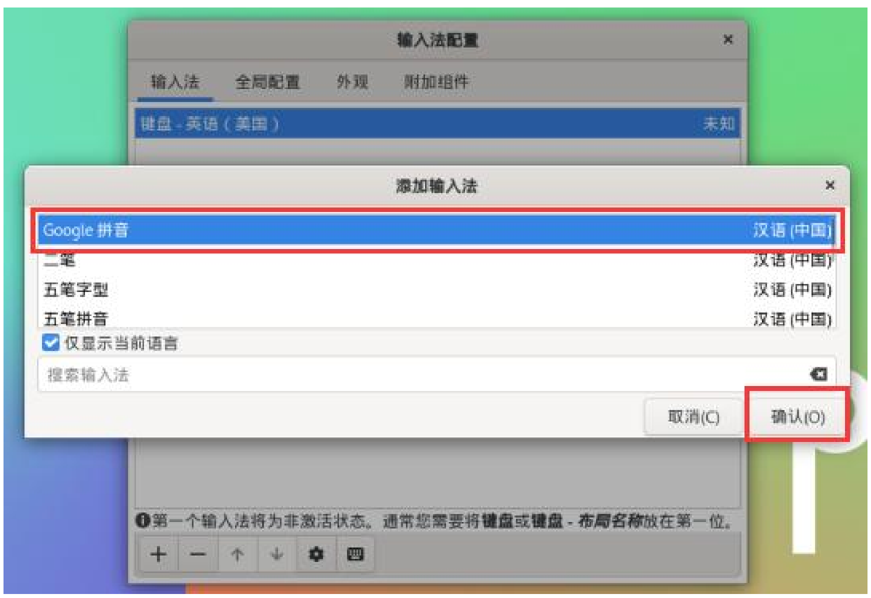
```
[orangepi@orangepi ~]$ sudo pacman -S fcitx-im fcitx-configtool
:: There are 3 members in group fcitx im:
:: Software Warehouse Community
    1) fcitx    2) fcitx-qt5    3) fcitx-qt6

Enter a selection (default=all selected): 1
```

11) Then open the Fcitx configuration program.

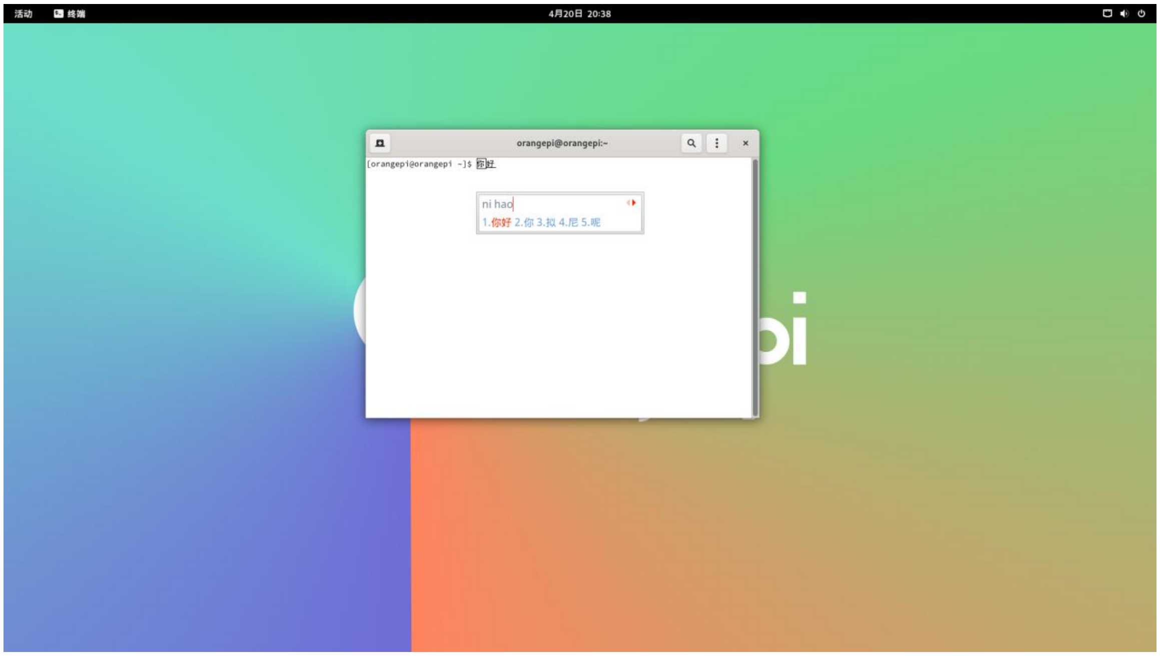


12) Then add **Google Pinyin** input method.





13) Then we can open a terminal to test the Chinese input method. After opening the terminal, if the default input method is still English, we can switch to the Chinese input method by pressing the **Ctrl+Space** shortcut key, and then we can input Chinese.



5. 6. Method of installing wiringOP

Note that WiringOP is already pre installed in the OPi OS Arch image released by Orange Pi. Unless the WiringOP code is updated, there is no need to download, compile, and install it again. You can use it directly.

After entering the system, you can run the gpio readall command. If you can see the output below, it means WiringOP is pre installed and can be used normally.

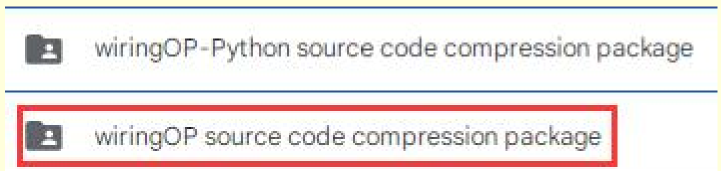
```
[root@orangepi wiringOP]# gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   63 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   62 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |     IN | 0 |  7 || 8  | 0 | IN     | TXD.6    | 3   | 33   |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | RXD.6    | 4   | 32   |
 |   47 |   5 |    RXD.4 |     IN | 1 | 11 || 12 | 0 | IN     | GPIO1_A2 | 6   | 34   |
 |   46 |   7 |    TXD.4 |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |     IN | 1 | 15 || 16 | 0 | IN     | GPIO1_A4 | 9   | 36   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A6 | 10  | 38   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | PWM3     | 13  | 39   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
```

1) Download the code for WiringOP.

```
[orangepi@orangepi ~]$ sudo pacman -Syy git
[orangepi@orangepi ~]$ git clone https://github.com/orangepi-xunlong/wiringOP.git -b next
```

Note that you need to download the code for the WiringOP next branch, please do not miss the - b next parameter.

If there are problems downloading code from GitHub, you can go to the official tool on the Orange Pi CM5 Base Tablet download page to download the source code compressed file of WiringOP.tar.gz.



2) Compile and install WiringOP.

```
[orangepi@orangepi ~]$ sudo pacman -Syy make gcc
[orangepi@orangepi ~]$ cd wiringOP
[orangepi@orangepi wiringOP]$ sudo ./build clean
[orangepi@orangepi wiringOP]$ sudo ./build
```

3) The output of the gpio readall command for testing is as follows:

```
[root@orangepi wiringOP]# gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   63 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   62 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |     IN | 0 |  7 || 8  | 0 | IN     | TXD.6    | 3   | 33   |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | RXD.6    | 4   | 32   |
 |   47 |   5 |    RXD.4 |     IN | 1 | 11 || 12 | 0 | IN     | GPIO1_A2 | 6   | 34   |
 |   46 |   7 |    TXD.4 |     IN | 1 | 13 || 14 |   |        | GND      |     |      |
 |   40 |   8 | GPIO1_B0 |     IN | 1 | 15 || 16 | 0 | IN     | GPIO1_A4 | 9   | 36   |
 |      |     |     3.3V |        |   | 17 || 18 | 0 | IN     | GPIO1_A6 | 10  | 38   |
 |   42 |  11 | SPI0_TXD |     IN | 0 | 19 || 20 |   |        | GND      |     |      |
 |   41 |  12 | SPI0_RXD |     IN | 0 | 21 || 22 | 1 | IN     | PWM3     | 13  | 39   |
 |   43 |  14 | SPI0_CLK |     IN | 0 | 23 || 24 | 1 | IN     | SPI0_CS0 | 15  | 44   |
 |      |     |      GND |        |   | 25 || 26 | 1 | IN     | SPI0_CS1 | 16  | 45   |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
```

5. 7. 26-pin interface GPIO, I2C, UART, SPI, and PWM testing

Note that if you need to set FDT overlays to open multiple configurations simultaneously, please use spaces to separate them and write them on one line like the red font configuration below.

[orangepi@orangepi ~]$ **sudo vim /boot/extlinux/extlinux.conf**

LABEL Orange Pi

LINUX /Image

FDT /dtbs/rockchip/rk3588s-orangepi-cm5-tablet.dtb

FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c5-m3.dtbo /dtbs/rockchip/overlay/rk3588-uart4-m0.dtbo

5. 7. 1. 26pin GPIO port test

1) There are a total of 17 GPIO ports available in the 26 pin development board. Taking pin 7- corresponding to GPIO as GPO1_A3- corresponding to wPi serial number 2- as an example, we will demonstrate how to set the high and low levels of GPIO ports.

```
[root@orangepi wiringOP]# gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   63 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   62 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |     IN | 0 |  7 || 8  | 0 | IN     | TXD.6    | 3   | 33   |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | RXD.6    | 4   | 32   |
```

2) Firstly, set the GPIO port to output mode, where the third parameter requires the input pin's corresponding wPi serial number.

[orangepi@orangepi ~]$ **gpio mode 2 out**

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

[orangepi@orangepi ~]$ **gpio write 2 0**

Using the gpio read command, you can see that the value (V) of pin 7 becomes 0.

[orangepi@orangepi ~]$ **gpio read 2**
0

4) Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3V, it indicates that the high level has been successfully set.

```
[orangepi@orangepi ~]$ gpio write 2 1
```

Using gpio readall, you can see that the value (V) of pin 7 has changed to 1.

```
[orangepi@orangepi ~]$ gpio read 2
1
```

5) The setting method for other pins is similar, just modify the serial number of wPi to the corresponding serial number of the pin.

5.7.2. 26-pin GPIO port pull-up and pull-down resistor setting method

1) Taking pin 7- corresponding to GPIO as GPO1_A3- corresponding to wPi serial number 2- as an example to demonstrate how to set the up and down resistors of GPIO port.

```
[root@orangepi wiringOP]# gpio readall
 +------+-----+----------+--------+---+CM5 Tablet+---+--------+----------+-----+------+
 | GPIO | wPi |   Name   |  Mode  | V | Physical | V |  Mode  | Name     | wPi | GPIO |
 +------+-----+----------+--------+---+----++----+---+--------+----------+-----+------+
 |      |     |     3.3V |        |   |  1 || 2  |   |        | 5V       |     |      |
 |   63 |   0 |    SDA.5 |     IN | 1 |  3 || 4  |   |        | 5V       |     |      |
 |   62 |   1 |    SCL.5 |     IN | 1 |  5 || 6  |   |        | GND      |     |      |
 |   35 |   2 |     PWM1 |     IN | 0 |  7 || 8  | 0 | IN     | TXD.6    | 3   | 33   |
 |      |     |      GND |        |   |  9 || 10 | 0 | IN     | RXD.6    | 4   | 32   |
```

2) Firstly, it is necessary to set the GPIO port to input mode, where the third parameter requires the serial number of the wPi corresponding to the input pin

```
[orangepi@orangepi ~]$ gpio mode 2 in
```

3) After setting to input mode, execute the following command to set the GPIO port to pull-up mode

```
[orangepi@orangepi ~]$ gpio mode 2 up
```

4) Then enter the following command to read the level of GPIO port. If the level is 1, it means that the pull-up mode is successfully set

```
[orangepi@orangepi ~]$ gpio read 2
1
```

5) Then execute the following command to set the GPIO port to pull-down mode

```
[orangepi@orangepi ~]$ gpio mode 2 down
```

6) Then enter the following command to read the level of the GPIO port. If the level is 0, it indicates that the pull-down mode has been successfully set

```
[orangepi@orangepi ~]$ gpio read 2
0
```

5. 7. 3. 26pin I2C test

1) According to the table below, the Orange Pi CM5 Base Tablet can use two sets of i2c buses, i2c4 and i2c5.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

In the OPi OS Arch system, i2c in pin 26 is turned off by default and needs to be manually turned on to use.

Add the configuration in red font below to/boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open both i2c4 and i2c5 simultaneously. If only one needs to be opened, fill in one.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-cm5-tablet.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-i2c4-m3.dtbo
/dtbs/rockchip/overlay/rk3588-i2c5-m3.dtbo
```

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

2) After starting the OPi OS Arch system, first confirm the existence of the i2c device node in/dev.

```
[orangepi@orangepi ~]$ ls /dev/i2c-*
```

3) Then connect an I2C device to the I2C pin of the 26 pin connector.

I2C bus	SDA corresponds to 26 pins	SCL corresponds to 26 pins
I2C4_M3	Pin 12	Pin 7
I2C5_M3	Pin 3	Pin 5

4) Then use the **i2cdetect -y** command. If the address of the connected i2c device can be detected, it means that i2c can be used normally.

```
[orangepi@orangepi ~]$ sudo pacman -Syy i2c-tools
[orangepi@orangepi ~]$ sudo i2cdetect -y 4          #i2c4 Command
[orangepi@orangepi ~]$ sudo i2cdetect -y 5          #i2c5 Command
```

5. 7. 4. 26 pin UART Test

1) According to the table below, the Orange Pi CM5 Base Tablet has four sets of UART buses available: UART 1, UART 4, UART 6, and UART 7.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

In the OPi OS Arch system, UART in pin 26 is turned off by default and needs to be manually turned on to use.

Add the configuration in red font below to/boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open UART 1, UART 4, UART 6, and UART 7 simultaneously. If only one needs to be opened, fill in one.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
```

```
FDT /dtbs/rockchip/rk3588s-orangepi-cm5-tablet.dtb
FDTOVERLAYS    /dtbs/rockchip/overlay/rk3588-uart0-m2.dtbo
/dtbs/rockchip/overlay/rk3588-uart1-m1.dtbo
/dtbs/rockchip/overlay/rk3588-uart4-m0.dtbo
/dtbs/rockchip/overlay/rk3588-uart6-m2.dtbo
/dtbs/rockchip/overlay/rk3588-uart7-m2.dtbo
```

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

2) After entering the Linux system, first confirm whether there is a device node corresponding to UART in/dev

```
[orangepi@orangepi ~]$ ls /dev/ttyS*
/dev/ttyS1    /dev/ttyS4    /dev/ttyS6    /dev/ttyS7 ......
```

3) Then start testing the UART interface by short circuiting the RX and TX of the UART interface to be tested using DuPont wires

UART bus	RX corresponds to 26 pins	TX corresponds to 26 pins
UART1_M1	Pin 3	Pin 5
UART4_M0	Pin 13	Pin 11
UART6_M1	Pin 10	Pin 8
UART7_M2	Pin 24	Pin 26

4) Use the **gpio serial** command to test the loopback function of the serial port as shown below. If you can see the print below, it indicates that the serial communication is normal

a. Test UART1

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS1
[sudo] password for orangepi:    #Enter password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
```

```
Out:    5:   ->    5^C
```

b. Test UART4

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS4
[sudo] password for orangepi:   #Enter password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

c. Test UART6

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS6
[sudo] password for orangepi:   #Enter password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

d. Test UART7

```
[orangepi@orangepi ~]$ sudo gpio serial /dev/ttyS7
[sudo] password for orangepi:   #Enter password here

Out:    0:   ->    0
Out:    1:   ->    1
Out:    2:   ->    2
Out:    3:   ->    3
Out:    4:   ->    4
Out:    5:   ->    5^C
```

5.7.5. 26 pin SPI test

1) According to the table below, the SPI available for Orange Pi CM5 Base Tablet are SPI0 and SPI1.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

In the OPi OS Arch system, spi0 and spi1 in pin 26 are turned off by default and need to be manually turned on to use.

Add the configuration in red font below to/boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open spi0 and spi1.

```
[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-cm5-tablet.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-spi0-m2-cs0-cs1-spidev.dtbo
rk3588-spi1-m0-cs0-spidev.dtbo
```

2) The corresponding pins of SPI0 and SPI1 in the 26 pins are shown in the table below:

	SPI1_M2 corresponds to 26 pins	SPI0-M2 corresponds to 26 pins
MOSI	**Pin 10**	**Pin 19**
MISO	**Pin 8**	**Pin 21**
CLK	**Pin 11**	**Pin 23**
CS0	**Pin 13**	**Pin 24**
CS1	**NO**	**Pin 26**

3) Then check if there is a device node for **spidevx.x** in the OPi OS Arch system. If it exists, it means that SPI has been set up and can be used directly.

```
[orangepi@orangepi ~]$ ls /dev/spidev*
/dev/spidev0.0 /dev/spidev1.0
```

4) Then, without short circuiting the mosi and miso pins of SPI0 or SPI1, run spidev_test

and the output result is shown below. It can be seen that the data of TX and RX are inconsistent.

```
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev0.0
或者
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev1.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF  | ................................
```

5) Then short circuit the two pins of SPI0 or SPI1, Mosi (pin 19 in the 26 pin interface) and Miso (pin 21 in the 26 pin interface), and run spidev_test to output the same data as sent and received.

```
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev0.0
或者
[orangepi@orangepi ~]$ sudo spidev_test -v -D /dev/spidev1.0
spi mode: 0x0
bits per word: 8
max speed: 500000 Hz (500 KHz)
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D  | ......@....▒.................▒.
```

5. 7. 6. PWM Testing Method

1) As shown in the table below, the Orange Pi CM5 Base Tablet can use PWM with PWM 0 pwm1、pwm3、 There are four PWM channels in total with PWM 13.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

In the OPi OS Arch system, the PWM in the 26 pin is turned off by default and needs to be manually turned on to use.

Add the configuration in red font below to/boot/extlinux/extlinux.conf, and then restart the OPi OS Arch system to open pwm0, pwm1, pwm3, and pwm13 simultaneously. If only one needs to be opened, fill in one.

[orangepi@orangepi ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
LINUX /Image
FDT /dtbs/rockchip/rk3588s-orangepi-cm5-tablet.dtb
FDTOVERLAYS /dtbs/rockchip/overlay/rk3588-pwm0-m1.dtbo
/dtbs/rockchip/overlay/rk3588-pwm0-m1.dtbo
/dtbs/rockchip/overlay/rk3588-pwm1-m2.dtbo
/dtbs/rockchip/overlay/rk3588-pwm3-m3.dtbo
/dtbs/rockchip/overlay/rk3588-pwm13-m2.dtbo

The red font configuration above needs to be written on one line, and different configurations need to be separated by spaces.

2) When a PWM is turned on, an additional pwmchipX (where X is a specific number) will appear in**/sys/class/pwm/**. For example, after turning on pwm13, checking the pwmchipX under**/sys/class/pwm/**/will change from two to three.

```
[orangepi@orangepi ~]$ ls /sys/class/pwm/
pwmchip0    pwmchip1    pwmchip2
```

3) Which pwmchip corresponds to pwm13 above? Let's first check the output of the **ls /sys/class/pwm/ -l** command, as shown below:

```
[orangepi@orangepi ~]$ ls /sys/class/pwm/ -lh
total 0
lrwxrwxrwx 1 root root 0  4月11日 16:53 pwmchip0 -> ../../devices/platform/fd8b0010.pwm/pwm/pwmchip0
lrwxrwxrwx 1 root root 0  4月11日 16:53 pwmchip1 -> ../../devices/platform/febf0010.pwm/pwm/pwmchip1
lrwxrwxrwx 1 root root 0  4月11日 16:53 pwmchip2 -> ../../devices/platform/febf0020.pwm/pwm/pwmchip2
[orangepi@orangepi ~]$
```

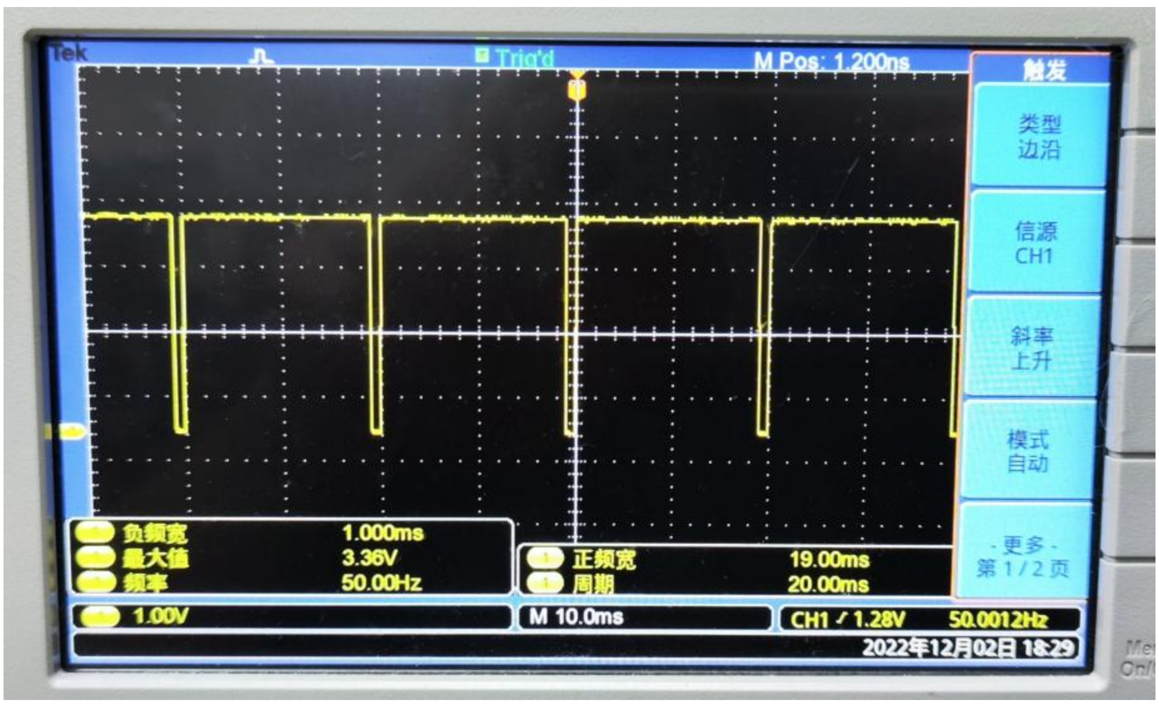
4) Then, as shown in the table below, the base address of the pwm13 register is febf0010. Looking at the output of the **ls /sys/class/pwm/ -l** command, it can be seen that

pwmchip1 is linked to febf0010.pwm, so pwm13 corresponds to pwmchip1.

复用功能	复用功能	复用功能	GPIO	GPIO序号	引脚序号	引脚序号	GPIO序号	GPIO	复用功能	复用功能
			3.3V		1	2		5V		
PWM13_M2(febf0010)	UART1_RX_M1	I2C5_SDA_M3	GPIO1_B7	47	3	4		5V		
	UART1_TX_M1	I2C5_SCL_M3	GPIO1_B6	46	5	6		GND		
	PWM1_M2(fd8b0010)	I2C4_SCL_M3	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2	SPI1_MISO_M2
			GND		9	10	57	GPIO1_D1	UART6_RX_M2	SPI1_MOSI_M2
PWM0_M1(fd8b0000)	UART4_TX_M0	SPI1_CLK_M2	GPIO1_D2	58	11	12	34	GPIO1_A2	I2C4_SDA_M3	
	UART4_RX_M0	SPI1_CS0_M2	GPIO1_D3	59	13	14		GND		
			GPIO1_B0	40	15	16	36	GPIO1_A4		
			3.3V		17	18	38	GPIO1_A6		
		SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND		
		SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3(fd8b0030)	
		SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2	UART7_RX_M2
			GND		25	26	45	GPIO1_B5	SPI0_CS1_M2	UART7_TX_M2

5) Then use the following command to make pwm13 output a 50Hz square wave (please switch to the root user first, and then execute the following command)

```
[root@orangepi orangepi]# echo 0 > /sys/class/pwm/pwmchip1/export
[root@orangepi orangepi]# echo 20000000 > /sys/class/pwm/pwmchip1/pwm0/period
[root@orangepi orangepi]# echo 1000000 > /sys/class/pwm/pwmchip1/pwm0/duty_cycle
[root@orangepi orangepi]# echo 1 > /sys/class/pwm/pwmchip1/pwm0/enable
```



6) The testing method for PWM 13 demonstrated above is similar to other PWM testing methods.

6. Linux SDK - Orangepi build usage instructions

6.1. Compilation System Requirements

We can cross compile the Linux image of the development board on an x64 computer, or compile the Linux image of the development board on the Ubuntu 22.04 system. Please choose one according to your preferences.

If using orangepi build to compile Linux images on the Ubuntu 22.04 system of the development board, please ensure proper cooling. If the heat dissipation is not done properly, it is easy to cause file system runaway errors.

6.1.1. Compiling Ubuntu 22.04 system using development board

1) The Linux SDK, also known as **orangepi-build**, supports running on **Ubuntu 22.04** on the development board (which has not been tested on other systems), so before downloading orangepi build, please ensure that the Ubuntu version installed on the development board is Ubuntu 22.04. The command to check the installed Ubuntu version on the development board is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version used does not meet the requirements. Please replace the system before performing the following operations.

```
orangepi@orangepi:~$ lsb_release -a
No LSB modules are available.
Distributor ID:	Ubuntu
Description:	Ubuntu 22.04.1 LTS
Release:	22.04
Codename:	jammy
```

2) Since the kernel and U-boot source code are stored on GitHub, it is important to ensure that the development board can download the code from GitHub properly when compiling the image.

6.1.2. Compiling on Ubuntu 22.04 computer using x64

1) The Linux SDK, also known as **orangepi-build**, supports running on computers with **Ubuntu 22.04** installed. Therefore, before downloading orangepi build, please make

sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the installed Ubuntu version on the computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version used does not meet the requirements. Please replace the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID:     Ubuntu
Description:        Ubuntu 22.04 LTS
Release:            22.04
Codename:           jammy
```

2) If the computer is installed with a Windows system and does not have Ubuntu 22.04 installed, you can consider using**VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine on the Windows system. However, please note that do not compile orangepi build on a WSI virtual machine, as orangepi build has not been tested on a WSI virtual machine, so it cannot be guaranteed that orangepi build can be used properly in WSI.

3) The installation image download address for 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 virtual machine, please first set the software source of Ubuntu 22.04 to Qinghua Source, otherwise errors may occur during software installation due to network issues.

a. The method of replacing Tsinghua Source can refer to the instructions on this webpage.

```
https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/
```

b. Note that 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 contents of the **/etc/apt/sources.list** file that needs to be replaced are:

```
test@test:~$ sudo mv /etc/apt/sources.list /etc/apt/sources.list.bak
test@test:~$ sudo vim /etc/apt/sources.list
# By default, the source code image has been annotated to improve the speed of apt updates. If necessary, you can remove the annotation yourself
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ jammy-security main restricted universe multiverse

# Pre release software 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 replacement, it is necessary to update the package information and ensure that there are no errors.

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

e. In addition, since the kernel and U-boot source code are stored on GitHub, it is important to ensure that the computer can download the code from GitHub properly when compiling the image.

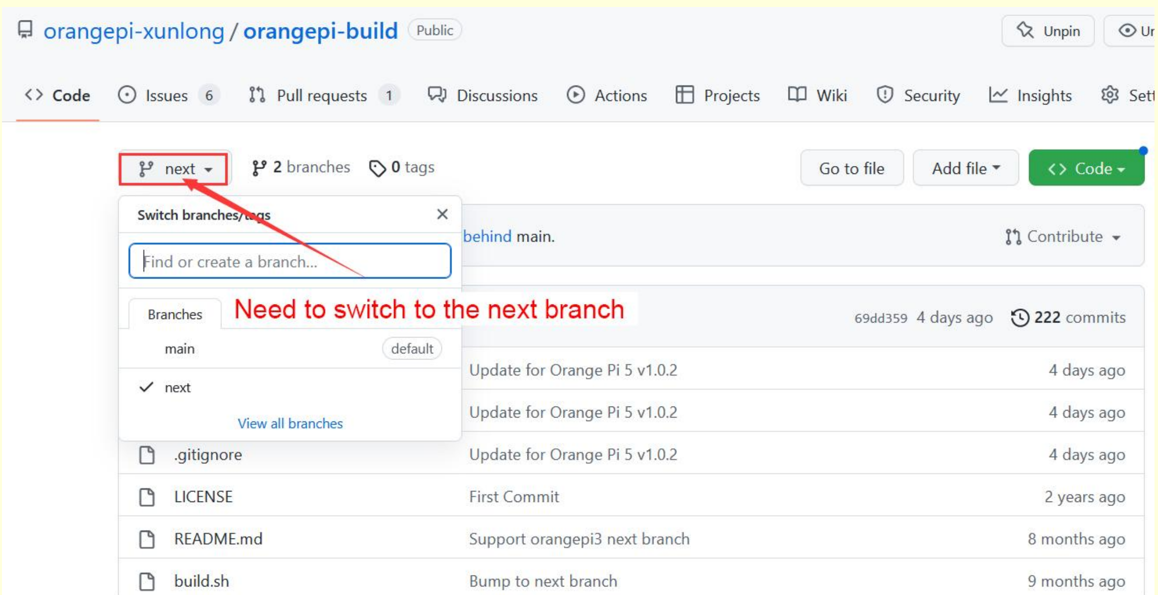
6. 2. Obtain the source code of Linux SDK

6. 2. 1. Download Orangepi build from GitHub

1) The Linux SDK actually refers to the Orangepi build code, which is modified based on the armbian build compilation system. Using Orangepi build, multiple versions of Linux images can be compiled. First, download the code for orangepi build. 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 Tablet development board requires downloading the next branch source code of orangepi build. The git clone command above needs to specify the branch of orangepi build source code as next.



Downloading the code for orangepi build through the git clone command does not require entering the username and password of the GitHub account (the same applies to downloading other code in this manual). If Ubuntu PC prompts for the username and password of the GitHub account after entering the git clone command, it is usually due to an incorrect input of the address of the orangepi build

repository after git clone. Please carefully check the spelling of the command for errors, rather than 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 on the development board are as follows:

Branch	U-boot version	Linux kernel version
legacy	u-boot 2017.09	linux5.10
current	u-boot 2017.09	linux6.1

The branch mentioned here and the branch of orangepi build source code are not the same thing, please don't confuse them. This branch is mainly used to distinguish between different versions of kernel source code.

At present, 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 include the following files and folders.

a. **build.sh**: Compile startup script.
b. **external**: Contains configuration files, specific scripts, and source code for some programs required for compiling images.
c. **LICENSE**: GPL 2 license file.
d. README.md: Orangepi build documentation.
e. **scripts**: Compile a universal script for Linux images.

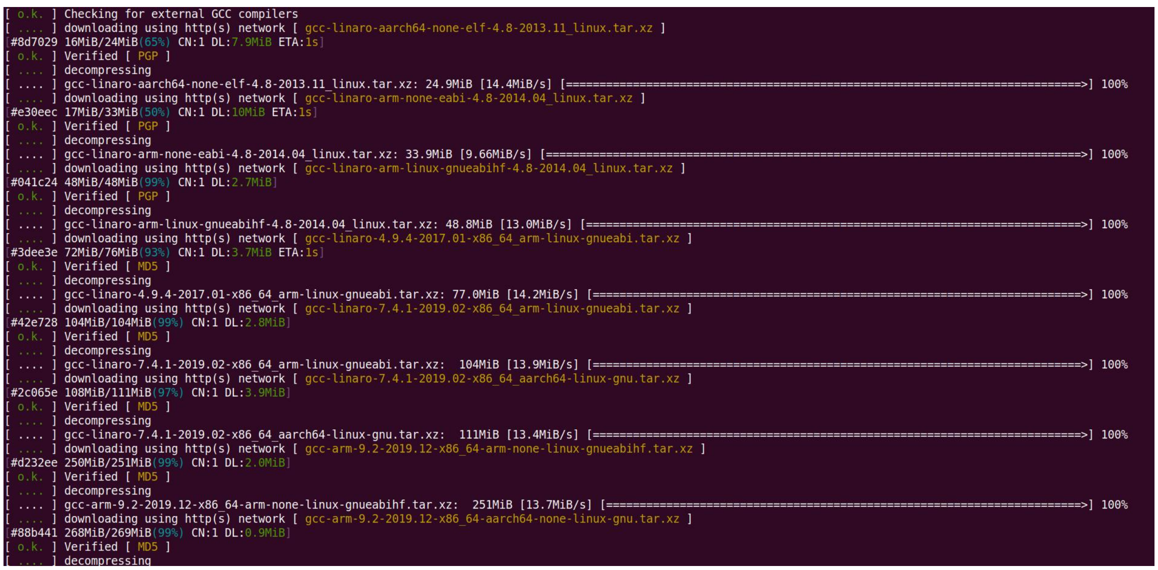
```
test@test:~/orangepi-build$ ls
build.sh    external    LICENSE    README.md    scripts
```

If you download the code for Orangepi build from GitHub, you may find that the Orangepi build does not include the source code for u-boot and Linux kernel, nor does it require a cross compilation toolchain to compile u-boot and Linux kernel. This is normal because these things are stored in other separate GitHub repositories or on certain servers (the addresses will be detailed below). Orangepi build specifies the addresses of u-boot, Linux kernel, and cross compilation toolchain in the script and configuration files. When running Orangepi build, if it finds that these things are not available locally, it will automatically download them from the corresponding places.

6. 2. 2. Download the cross compilation toolchain

The cross compilation toolchain will only be downloaded when using orangepi build to compile the image on an x64 computer. Compiling the Linux image of the development board in Ubuntu 22.04 will not download cross compilation toolchains, and orangepi build/toolchains will be an empty folder.

1) When Orangepi build runs for the first time, it automatically downloads the cross compilation toolchain and places it in the **toolchains** folder. After running the build.sh script of Orangepi build, it checks whether all the cross compilation **toolchains** in toolchains exist. If they do not exist, it will restart the download. If they do exist, it will be used directly without repeated downloads.



2) The mirror website of the cross compilation toolchain 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**, multiple versions of cross compilation toolchains will be included, 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 toolchain used to compile 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 toolchain used to compile the u-boot source code is:

a. v2017.09：

```
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
```

6. 2. 3. Explanation of the complete directory structure of orangepi build

1) After downloading the orangepi build repository, it does not include the Linux kernel, U-boot source code, or cross compilation toolchain. The Linux kernel and U-boot source code are stored in separate 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 first run, it will download the cross compilation toolchain, u-boot, and Linux kernel source code. After successfully compiling the Linux image once, the files and folders that can be seen in Orangepi build are:

a. **build.sh**: Compile the startup script.

b. **external**: Contains configuration files required for compiling images, scripts for specific functions, and source code for some programs. The rootfs compressed files cached during the image compilation process are also stored in the external folder.

c. **kernel**: The source code of the Linux kernel is stored in the folder named **orange-pi-5.10-rk35xx**, which contains the legacy branch kernel code of the RK3588/RK3588S series development board. The folder named **orange-pi-6.1-rk35xx** contains the current branch kernel code of the RK3588/RK3588S series development board. Please do not manually modify the folder name of the kernel source code. If it is modified, the compilation system will download the kernel source code again when running.
d. **LICENSE**: GPL 2 license file.
e. **README.md**: Orangepi build documentation.
f. **output**: Store compiled deb packages such as u-boot and Linux, compilation logs, and compiled images.
g. **scripts**: Compile a universal script for Linux images.
h. **toolchains**: Store cross compilation toolchains.
i. **u-boot**: The folder named v2017.09-rk3588 contains the u-boot source code of the legacy branch of the RK3588/RK3588S series development board. Please do not manually modify the folder name of the u-boot source code. If it is modified, the compilation system will re download the u-boot source code when running.
j. **userpatches**: Store the configuration files required for compiling scripts.

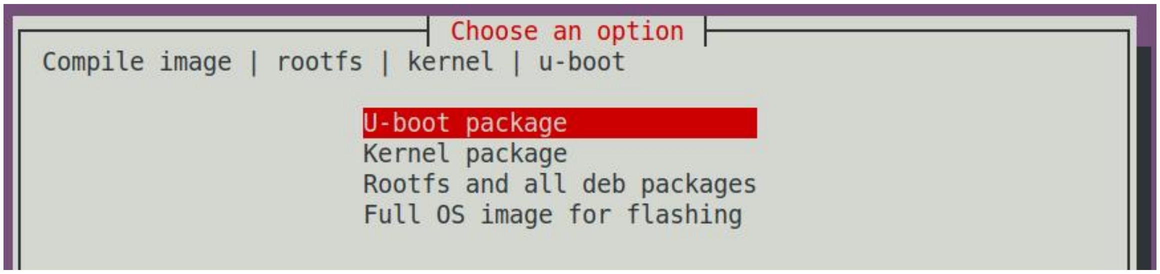
```
test@test:~/orangepi-build$ ls
build.sh    external    kernel    LICENSE    output    README.md    scripts    toolchains
u-boot    userpatches
```

6. 3. Compiling u-boot

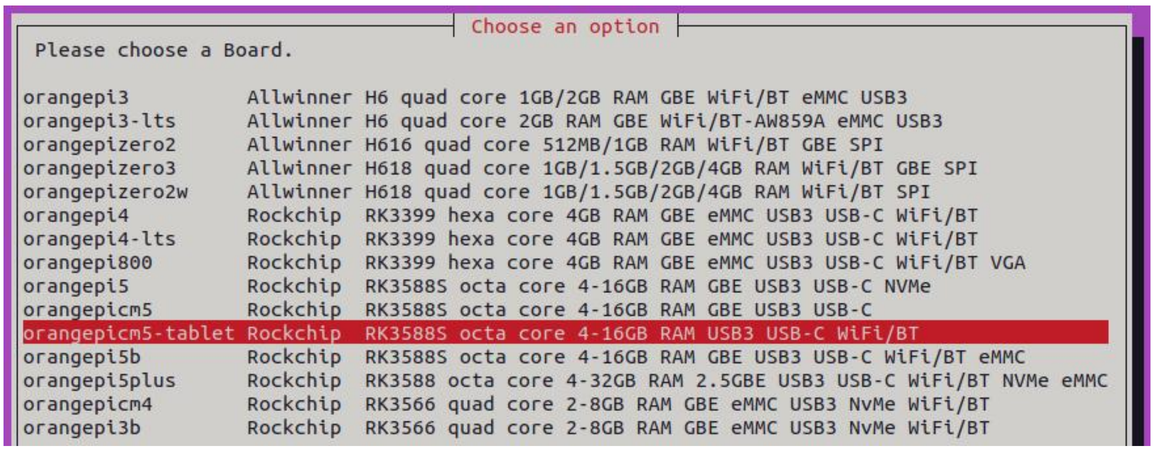
1) Run the build.sh script and remember to grant sudo privileges.

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package** and press enter.



3) Next, select the model of the development board.



4) Then it will start compiling u-boot, and some of the information prompted during compilation is as follows:

a. The version of u-boot source code.

[o.k.] Compiling u-boot [**v2017.09**]

b. The 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 compiled u-boot deb package.

[o.k.] File name [**linux-u-boot-legacy-orangepicm5-tablet_1.0.0_arm64.deb**]

e. The time taken for compilation.

[o.k.] Runtime [**1 min**]

f. Repeat the command to compile u-boot, and 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-tablet BRANCH=legacy BUILD_OPT=u-boot KERNEL_CONFIGURE=no**]

5) View the compiled u-boot deb package.

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-legacy-orangepicm5-tablet_1.0.0_arm64.deb
```

6) The generated deb package of u-boot contains the following files:

a. The following command can be used 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-tablet_1.0.0_arm64.deb . (Please 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-tablet_1.0.0_arm64.deb usr
```

b. The decompressed file is shown below:

```
test@test:~/orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-legacy-orangepicm5-tablet_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 build compilation system compiles the u-boot source code, it first synchronizes the u-boot source code with the u-boot source code on the GitHub server. Therefore, if you want to modify the u-boot source code, you first need to turn off the download and update function of the source code **(you need to compile the u-boot completely before turning off this function, otherwise it will prompt that the u-boot source code cannot be found. If it is a compressed source code downloaded from Baidu Cloud Drive, there is no problem because the u-boot source code is already**

cached). Otherwise, the modifications made will be restored. The method is as follows:

Set the IGNOREUPDATES variable to "yes" in userpatches/config-default.conf.

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

8) When debugging u-boot code, you can use the following method to update u-boot in the Linux image for testing.

a. Upload the compiled deb package of u-boot 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-tablet_1.0.0_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the deb package of the installed u-boot.

```
root@orangepi:~# apt purge -y linux-u-boot-orangepicm5-tablet-legacy
```

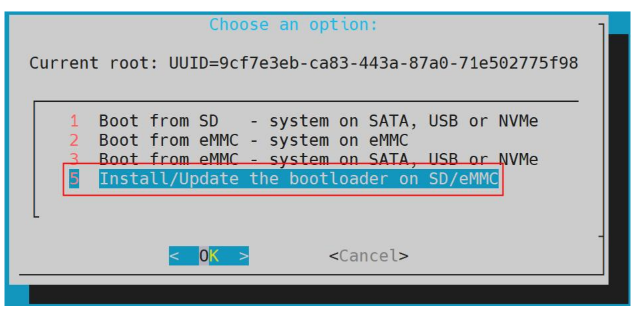
c. Reinstall the newly uploaded deb package for u-boot.

```
root@orangepi:~# dpkg -i linux-u-boot-legacy-orangepicm5-tablet_1.0.0_arm64.deb
```

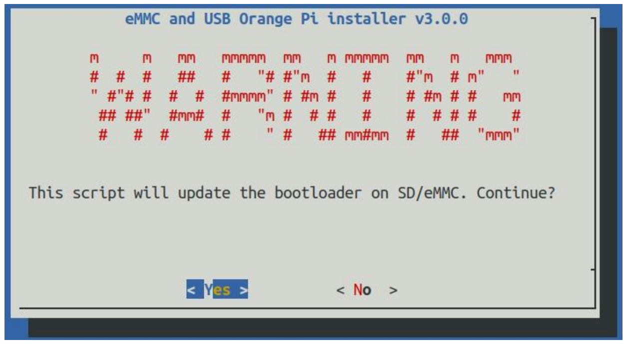
d. Then run the nand sata install script.

```
root@orangepi:~# nand-sata-install
```

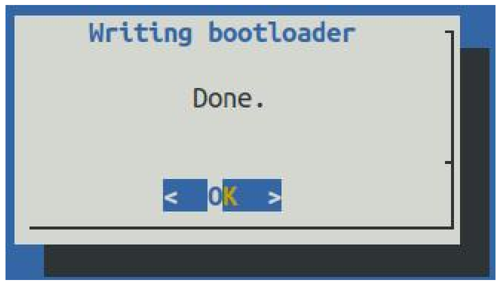
e. Then select **5 Install/Update the bootloader on SD/eMM** to update u-boot in TF card or eMMC.



f. After pressing the enter key, a warning will first pop up.



g. Pressing the enter key again will start updating u-boot, and after the update is complete, the following information will be displayed.



h. Then you can restart the development board to test whether the u-boot modifications have taken effect.

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_tablet_defconfig
```

b. In the U-boot 2017.09 source code, the development board uses dts files.

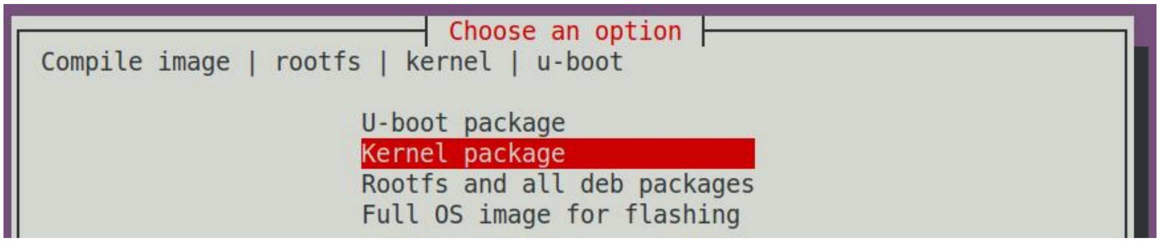
```
orangepi-build/u-boot/v2017.09-rk3588/arch/arm/dts/rk3588s-orangepi-cm5-tablet.dts
```

6. 4. Compiling Linux Kernel

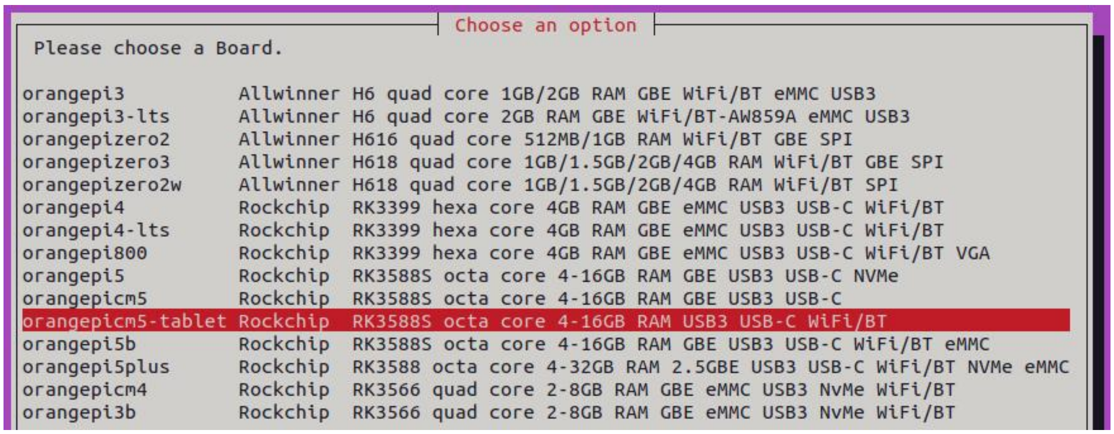
1) Run the build.sh script and remember to grant sudo privileges.

```
test@test:~/orangepi-build$ sudo ./build.sh
```

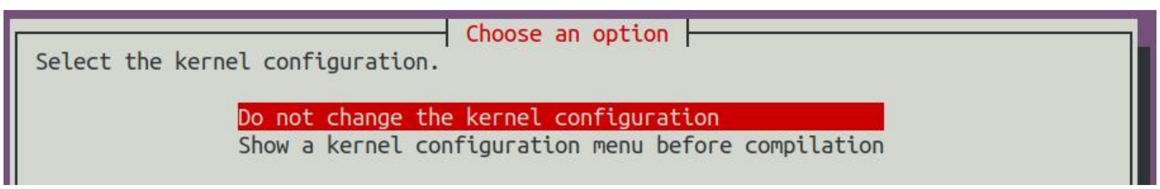
2) Select **Kernel package** and press enter.



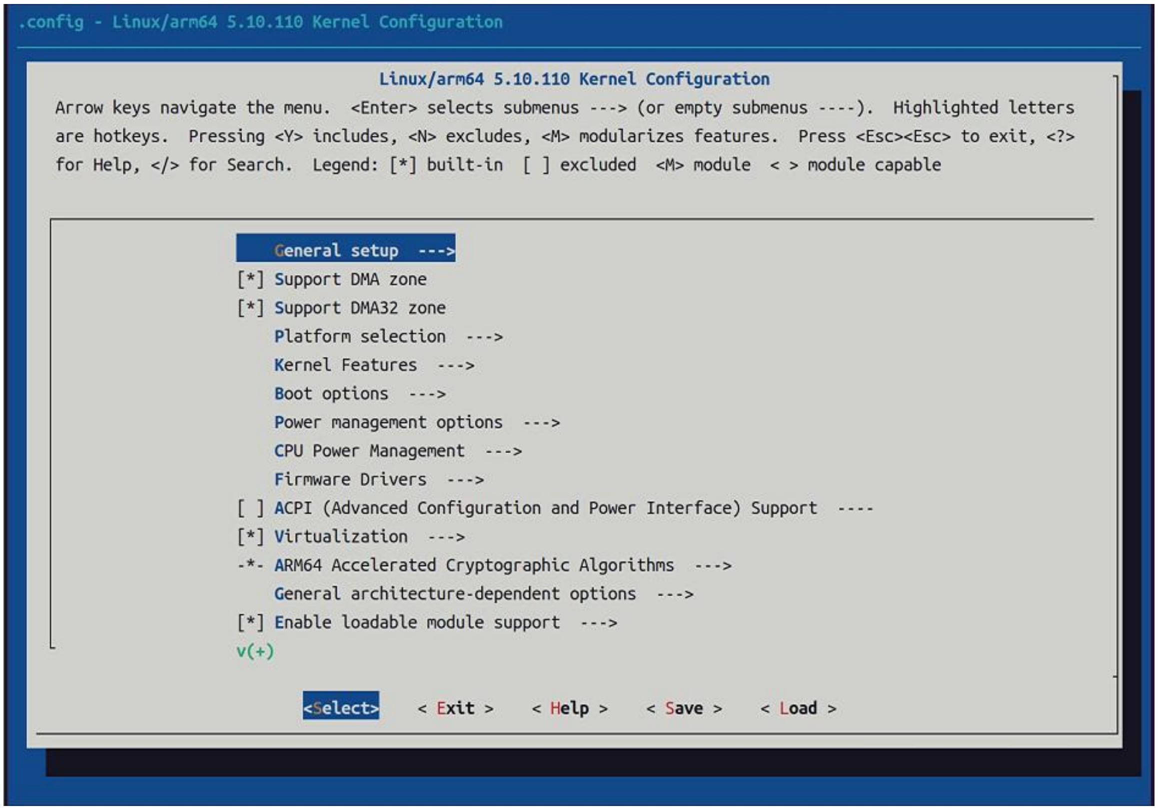
3) Next, select the model of the development board.



4) Then it will prompt whether the kernel configuration interface needs to be displayed. If the kernel configuration does not need to be modified, select the first one. If the kernel configuration needs to be modified, select the second one.



5) If step 4) selects the option to display the kernel configuration menu (second option), a kernel configuration interface opened through **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration, save and exit after modification, and the kernel source code will be compiled after exit.



a. If there is no need to modify the configuration options of the kernel, passing **KERNEL_CONFIGURE=no** when running the build.sh script can temporarily block the pop-up kernel configuration interface.

```
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** default.exe configuration file to permanently disable this feature.

c. If the following error appears when compiling the kernel, it is due to the small terminal interface of Ubuntu PC, which causes the **make menuconfig** interface to not display. Please set the Ubuntu PC terminal to its maximum size and then run the build.sh script again.

```
  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) The following is a partial explanation of the information prompted when compiling kernel source code:

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 compiled kernel related deb package.

[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 taken for compilation.

[o.k.] Runtime [**5 min**]

g. Finally, the compilation command for the previously selected kernel will be displayed. The following command can be used to start compiling the kernel source code without selecting it through the graphical interface.

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepicm5-tablet BRANCH=legacy BUILD_OPT=kernel KERNEL_CONFIGURE=no**]

7) View 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 deb package of the generated Linux image contains the following files:

a. The following command can be used 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 file is shown below:

```
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 build compilation system compiles the Linux kernel source code,

it first synchronizes the Linux kernel source code with the GitHub server's Linux kernel source code. Therefore, if you want to modify the Linux kernel source code, you first need to turn off the source code update function **(you need to compile the Linux kernel source code completely before turning off this function, otherwise it will prompt that the Linux kernel source code cannot be found. If it is a source code compressed package downloaded from Baidu Cloud Drive, there is no problem because the Linux source code is already cached)**. Otherwise, the modifications made will be restored. The method is as follows:

Set the IGNOREUPDATES variable to "yes" in **userpatches/config-default.conf**.

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

10) If modifications have been made to the kernel, the following method can be used to update the kernel and kernel modules of the Linux system on the development board.

a. Upload the compiled deb package of the Linux kernel to the Linux system on 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 deb package of the installed Linux kernel.

```
root@orangepi:~# apt purge -y linux-image-legacy-rockchip-rk3588
```

c. Reinstall the deb package of the newly uploaded Linux kernel.

```
root@orangepi:~# dpkg -i linux-image-legacy-rockchip-rk3588_1.0.0_arm64.deb
```

d. Then restart the development board and check if the kernel related modifications have taken effect.

```
root@orangepi:~# reboot
```

10) Other useful information.

a. The storage location of the kernel configuration file is as follows. Please do not search for the kernel configuration file used by the development board in the kernel source code.

```
orangepi-build/external/config/kernel/linux-rockchip-rk3588-legacy-opicm5-tablet.config
```

b. The location of the dts file used by the development board is:

```
orangepi-build/kernel/orange-pi-5.10-rk35xx/arch/arm64/boot/dts/rockchip/rk3588s-orangepi-cm5-tablet.dts
```

6. 5. Compile rootfs

1) Run the build.sh script and remember to grant sudo privileges.

test@test:~/orangepi-build$ **sudo ./build.sh**

2) Select **Rootfs and all deb packages**, then press enter.

```
Choose an option
Compile image | rootfs | kernel | u-boot

U-boot package
Kernel package
Rootfs and all deb packages
Full OS image for flashing
```

3) Next, select the model of the development board.

```
Choose an option
Please choose a Board.

orangepi3            Allwinner H6 quad core 1GB/2GB RAM GBE WiFi/BT eMMC USB3
orangepi3-lts        Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
orangepizero2        Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
orangepizero3        Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
orangepizero2w       Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
orangepi4            Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi4-lts        Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi800          Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
orangepi5            Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
orangepicm5          Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C
orangepicm5-tablet Rockchip  RK3588S octa core 4-16GB RAM USB3 USB-C WiFi/BT
orangepi5b           Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT eMMC
orangepi5plus        Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C WiFi/BT NVMe eMMC
orangepicm4          Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
orangepi3b           Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
```

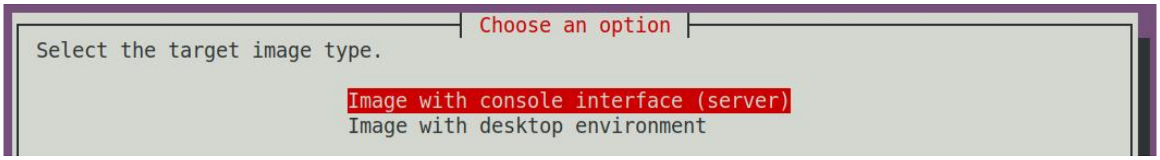
4) Then select the type of rootfs.

```
Choose a release package base
Select the target OS release package base

bookworm Debian 12 Bookworm
bullseye Debian 11 Bullseye
focal    Ubuntu Focal 20.04 LTS
jammy    Ubuntu jammy 22.04 LTS
```

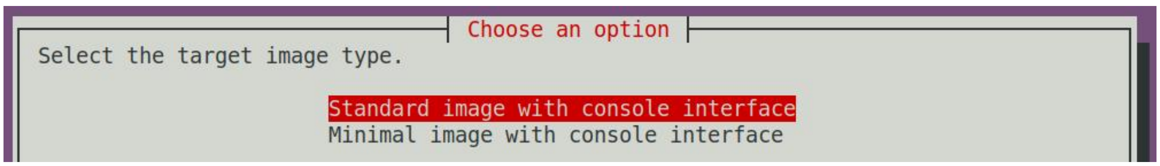
5) Then select the type of image.

a. **Image with console interface (server)** represents a server version image with a relatively small size.

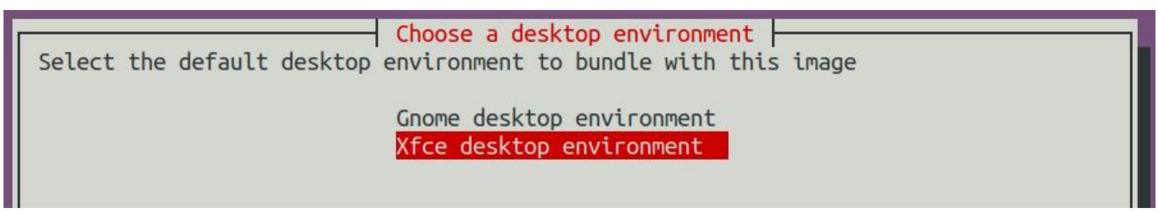
b. **Image with desktop environment** represents a desktop image with a relatively large volume.

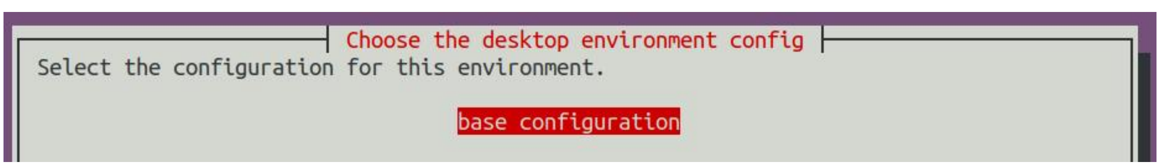


6) If compiling the server version of the image, you can also choose to compile the Standard version or the Minimal version. The Minimal version comes with much less pre installed software than the Standard version **(please do not choose the Minimal version unless there are special requirements, as many things are not pre installed by default and some features may not be available).**

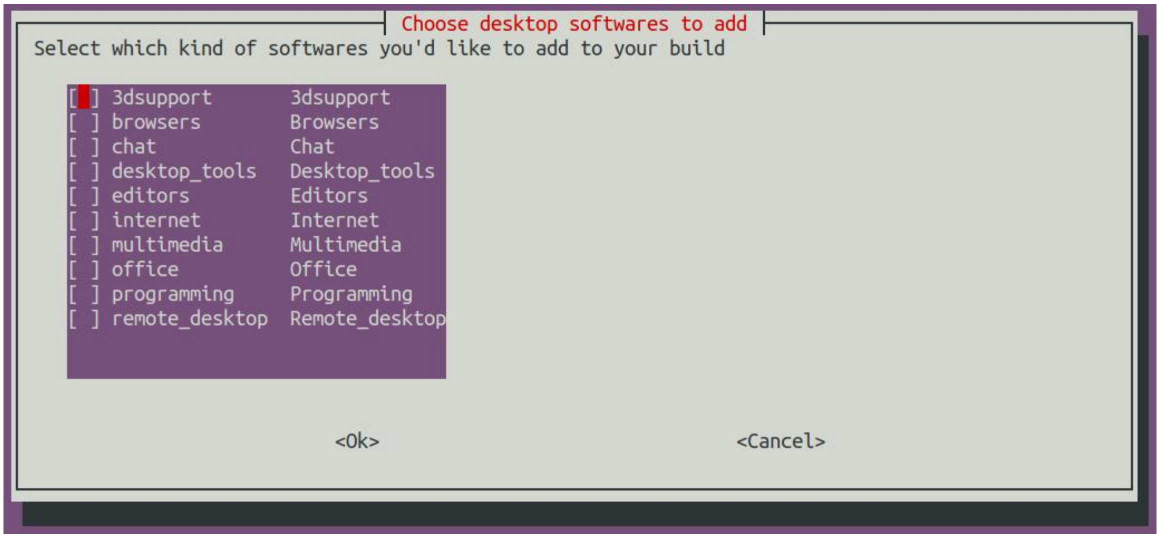


7) If compiling a desktop version of the image, you also need to choose the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktops.





Then you can choose additional software packages that need to be installed. Please press the enter key here to skip directly.



8) Then rootfs will start compiling, and some of the information prompted during compilation is as follows:

a. The type of rootfs.

[o.k.] local not found [Creating new rootfs cache for **jammy**]

b. The storage path of the rootfs compressed file generated by compilation.

[o.k.] Target directory [**external/cache/rootfs**]

c. The name of the rootfs compressed file generated by compilation.

[o.k.] File name [**jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4**]

d. The time taken for compilation.

[o.k.] Runtime [**13 min**]

9) View the compiled rootfs compressed file.

a. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4** is a compressed file of rootfs, and the meaning of each field in the name is:

a) **jammy** represents the type of Linux distribution of rootfs.

b) **Xfce** represents rootfs as the desktop version type, and if it is **cli**, it represents the server version type.

c) **arm64** represents the architecture type of rootfs.

d) **f930ff6ebbac1a72108a2e100762b18f** is an MD5 hash value generated from the package names of all software packages installed by rootfs. As long as the list of software packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to determine whether rootfs needs to be recompiled.

b. **jammy-xfce-arm64.f930ff6ebbac1a72108a2e100762b18f.tar.lz4.list** list the package names of all the 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
```

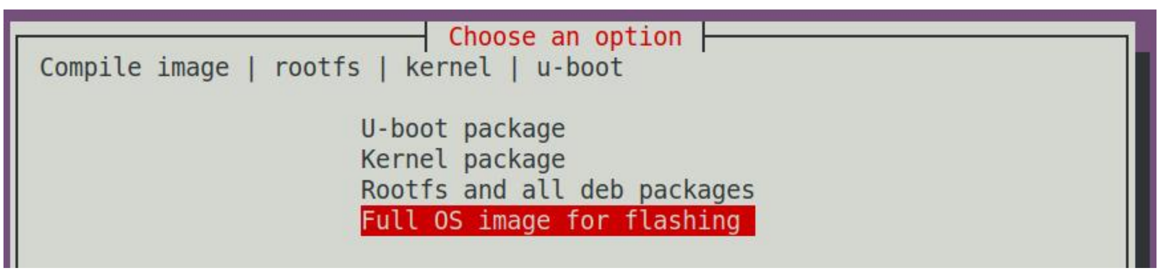
10) If the required rootfs already exist in **external/cache/rootfs**, compiling rootfs again will skip the compilation process and will not restart. When compiling the image, it will also search for available rootfs in **external/cache/rootfs**, and if so, use them directly, which can save a lot of download and compilation time.

6. 6. Compiling Linux Images

1) Run the build.sh script and remember to grant sudo privileges.

```
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.

```
Choose an option
Please choose a Board.

orangepi3          Allwinner H6 quad core 1GB/2GB RAM GBE WiFi/BT eMMC USB3
orangepi3-lts      Allwinner H6 quad core 2GB RAM GBE WiFi/BT-AW859A eMMC USB3
orangepizero2      Allwinner H616 quad core 512MB/1GB RAM WiFi/BT GBE SPI
orangepizero3      Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT GBE SPI
orangepizero2w     Allwinner H618 quad core 1GB/1.5GB/2GB/4GB RAM WiFi/BT SPI
orangepi4          Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi4-lts      Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT
orangepi800        Rockchip  RK3399 hexa core 4GB RAM GBE eMMC USB3 USB-C WiFi/BT VGA
orangepi5          Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C NVMe
orangepicm5        Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C
orangepicm5-tablet Rockchip  RK3588S octa core 4-16GB RAM USB3 USB-C WiFi/BT
orangepi5b         Rockchip  RK3588S octa core 4-16GB RAM GBE USB3 USB-C WiFi/BT eMMC
orangepi5plus      Rockchip  RK3588 octa core 4-32GB RAM 2.5GBE USB3 USB-C WiFi/BT NVMe eMMC
orangepicm4        Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
orangepi3b         Rockchip  RK3566 quad core 2-8GB RAM GBE eMMC USB3 NvMe WiFi/BT
```

4) Then select the type of rootfs.

```
Choose a release package base
Select the target OS release package base

bookworm Debian 12 Bookworm
bullseye Debian 11 Bullseye
focal    Ubuntu Focal 20.04 LTS
jammy    Ubuntu jammy 22.04 LTS
```

5) Then select the type of image.

a. **Image with console interface (server)** represents a server version image with a relatively small size.

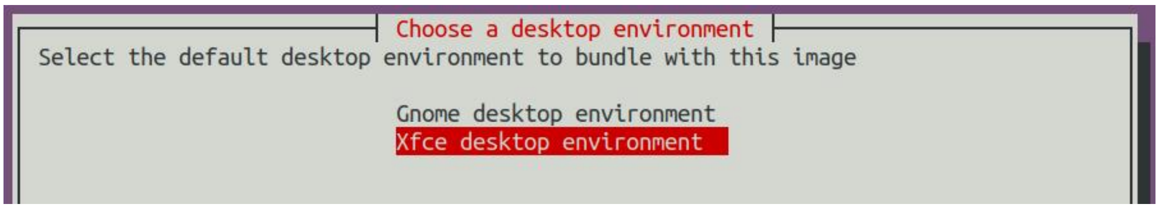
b. **Image with desktop environment** represents a desktop image with a relatively large volume.

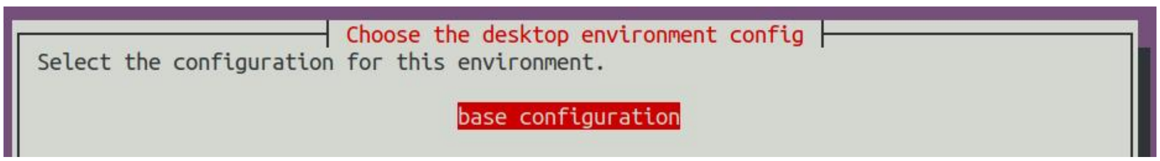
```
Choose an option
Select the target image type.

Image with console interface (server)
Image with desktop environment
```

6) If compiling the server version of the image, you can also choose to compile the Standard version or the Minimal version. The Minimal version comes with much less pre installed software than the Standard version **(please do not choose the Minimal version unless there are special requirements, as many things are not pre installed by default and some features may not be available).**

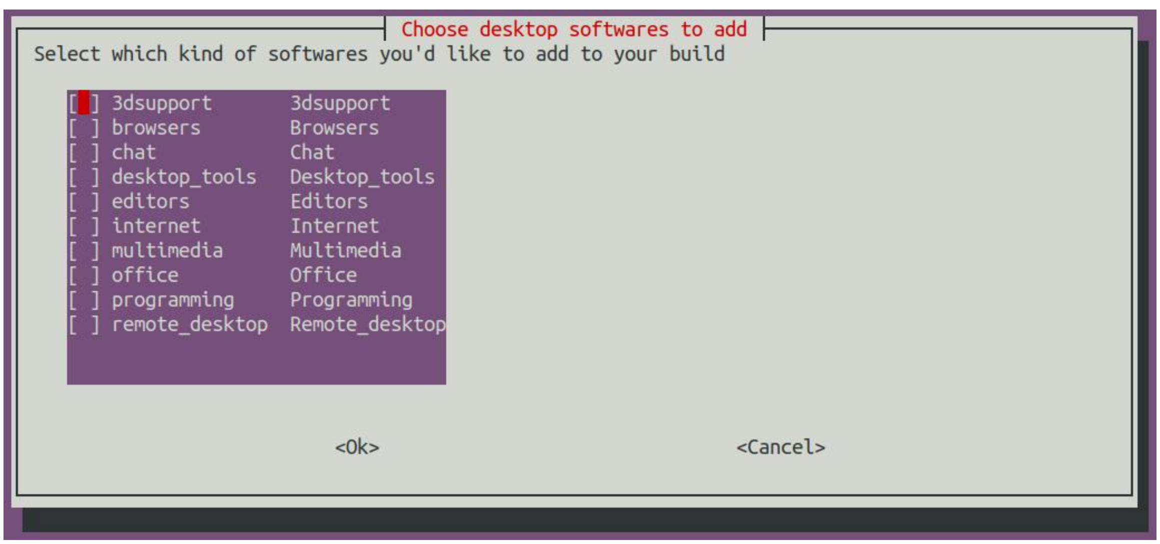
```
Choose an option
Select the target image type.

Standard image with console interface
Minimal image with console interface
```

7) If compiling a desktop version of the image, you also need to choose the type of desktop environment. Currently, Ubuntu Jammy mainly maintains XFCE and Gnome desktops, Ubuntu Focal only maintains XFCE desktops, Debian Bullseye mainly maintains XFCE and KDE desktops, and Debian Bookwork mainly maintains XFCE desktops.





Then you can choose additional software packages that need to be installed. Please press the enter key here to skip directly.



8) Then it will start compiling the Linux image, and the general process of compilation is as follows:

a. Initialize the compilation environment of Ubuntu PC and install the necessary software packages for the compilation process.
b. Download the source code for u-boot and Linux kernel (if cached, only update

the code).

c. Compile the u-boot source code and generate the deb package for u-boot.
d. Compile Linux source code and generate deb packages related to Linux.
e. Create a deb package for Linux firmware.
f. Create a deb package for the orangepi config tool.
g. Create deb packages that support board level support.
h. If compiling the desktop version image, desktop related deb packages will also be created.
i. Check if the rootfs are already cached. If not, create a new rootfs. If they are already cached, decompress and use them directly.
j. Install the deb package generated earlier into rootfs.
k. Make specific settings for different development boards and types of images, such as pre installing additional software packages, modifying system configurations, etc.
l. Then create an image file and format the partition, with the default type being ext4.
m. Copy the configured rootfs to the partition of the image.
n. Then update initramfs.
o. Finally, write the bin file of u-boot to the image using the dd command.

9) After compiling the image, the following message will be prompted:

a. The storage path of the compiled image.

```
[ o.k. ] Done building
[ output/images/orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160/orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.img ]
```

b. The time taken for compilation.

```
[ o.k. ] Runtime [ 19 min ]
```

c. The command to repeatedly compile the image can be used to start compiling the image without selecting through the graphical interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh    BOARD=orangepicm5-tablet BRANCH=legacy BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

7. Linux Development Manual

7.1. Method for separately compiling kernel source code in the Linux system of the development board

1) First, download the Linux kernel source code of the development board.

```
orangepi@orangepi:~$ git clone --depth=1 -b orange-pi-5.10-rk35xx https://github.com/orangepi-xunlong/linux-orangepi
```

If there are problems downloading code from GitHub, you can go to the official tool of the development board to download the kernel source code compressed file, then upload it to the Linux system of the development board, and finally decompress it.

The command to decompress the compressed kernel source code is:

orangepi@orangepi:~$ **tar zxf orange-pi-5.10-rk35xx.tar.gz**

orangepi@orangepi:~$ **mv orange-pi-5.10-rk35xx linux-orangepi**

After decompression, please execute the following command and synchronize the source code with GitHub to ensure that the source code is in the latest state:

orangepi@orangepi:~$ **cd linux-orangepi**

orangepi@orangepi:~/linux-orangepi$ **git pull**

2) Then configure the default kernel configuration.

```
orangepi@orangepi:~$ cd linux-orangepi
orangepi@orangepi:~/linux-orangepi$ make rockchip_linux_defconfig
```

The path of rockchip_linux_defconfig in the kernel source code is arch/arm64/configs/

3) Then compile the kernel source code.

```
orangepi@orangepi:~/linux-orangepi$ make -j10
```

4) Then install the kernel module.

```
orangepi@orangepi:~/linux-orangepi$ sudo make modules_install
```

The installation path of the kernel module is:/ lib/modules

After executing the sudo make modulars_install command, you can see an additional folder for kernel modules under/lib/modules/:

```
orangepi@orangepicm5-tablet:~$ ls /lib/modules
5.10.160+    5.10.160-rockchip-rk3588
```

5) Then install the kernel image and uInitrd.

```
orangepi@orangepi:~/linux-orangepi$ sudo make install
```

The installation path for kernel image and uInitrd is:/ boot/

After executing the sudo make install command, you can see an additional kernel file under/boot/:

```
orangepi@orangepicm5-tablet:~/orange-pi-5.10-rk3588$ ls /boot/vmlinuz*
/boot/vmlinuz-5.10.160+    /boot/vmlinuz-5.10.160-rockchip-rk3588
```

The file/boot/Image is actually loaded during system startup, and Image is a copy of the vmlinuz file.

6) Then install the dtb file into**/boot/dtb**.

```
orangepi@orangepi:~/linux-orangepi$ sudo make dtbs_install INSTALL_DTBS_PATH=/boot/dtb/
```

7) Then restarting the Linux system will load the newly compiled kernel.

```
orangepi@orangepi:~$ uname -r
```

5.10.160+

8. Instructions for using Android 13 system

8. 1. Supported Android versions

Android version	Kernel Version
Android 13	Linux5.10

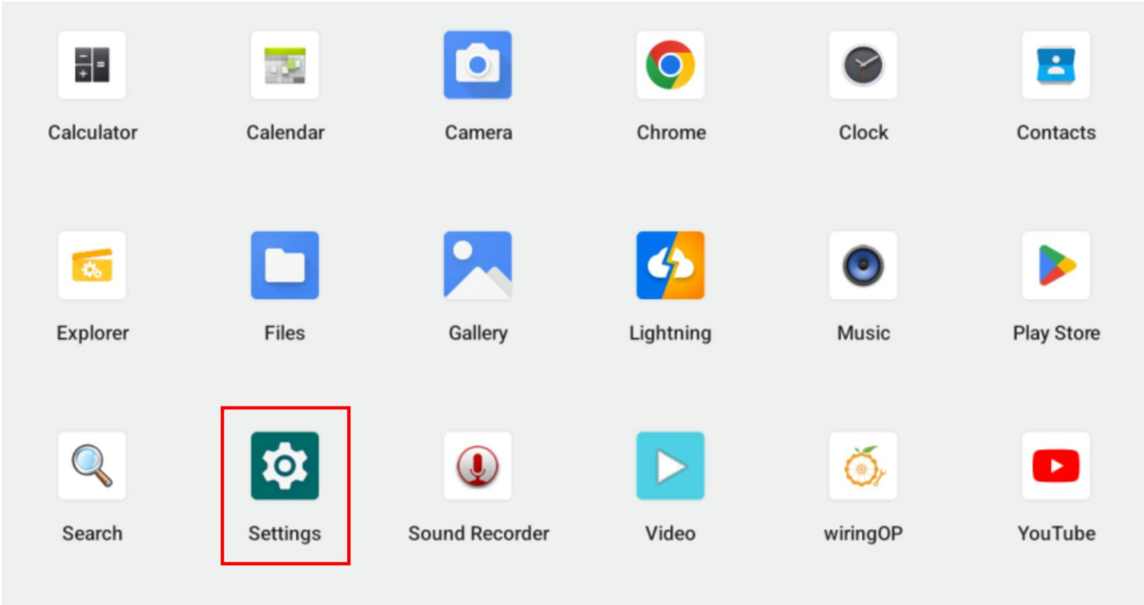
8. 2. Adaptation of Android Features

function	Android 13
HDMI display	OK
HDMI Audio	OK
USB 2.0	OK
USB 3.0	OK
WIFI	OK
Bluetooth	OK
Debug UART	OK
FAN	OK
EMMC startup	OK
GPIO（26pin）	OK
UART（26pin）	OK
SPI（26pin）	OK
I2C（26pin）	OK
PWM（26pin）	OK
Camera1	OK
Camera2	NO
Camera3	NO

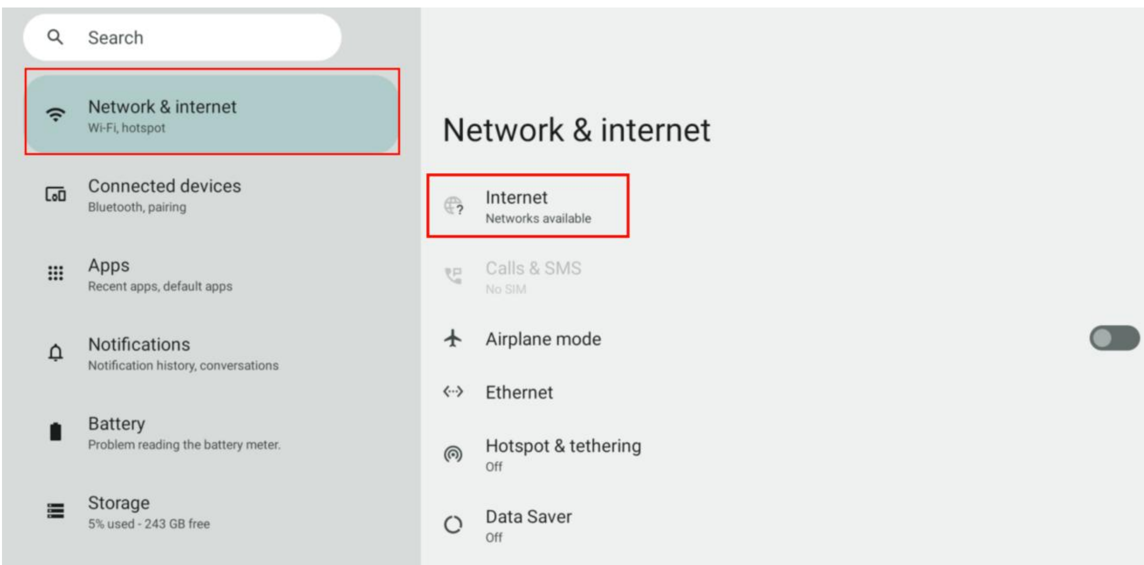
LCD	**OK**
Onboard MIC	**OK**
Headphone playback	**OK**
Earphone recording	**OK**
Speaker x 2	**OK**
LED lights	**OK**
Type-C to USB 3.0	**OK**
Type-C interface DP display	**OK**
Type-C interface DP audio	**OK**
TF card startup	**OK**
Identify NVMe SSD	**OK**
infrared	**OK**
GPU	**OK**
NPU	**OK**
VPU	**OK**
Power on/off button	**OK**
HDMI CEC function	**NO**

8. 3. WIFI Connection Testing Method

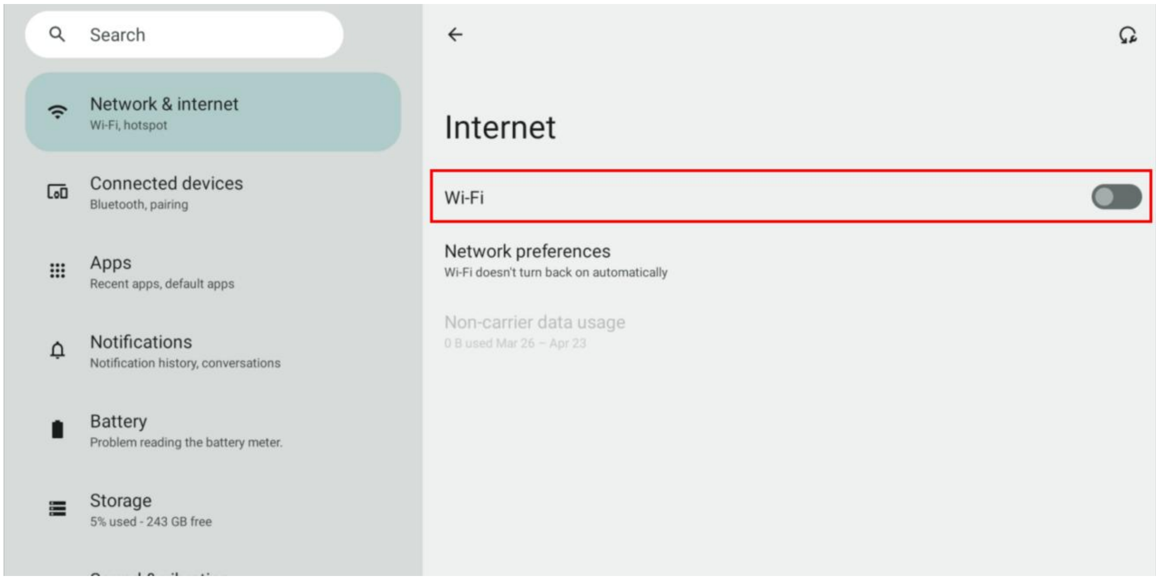
1) First, click to enter **Setting**



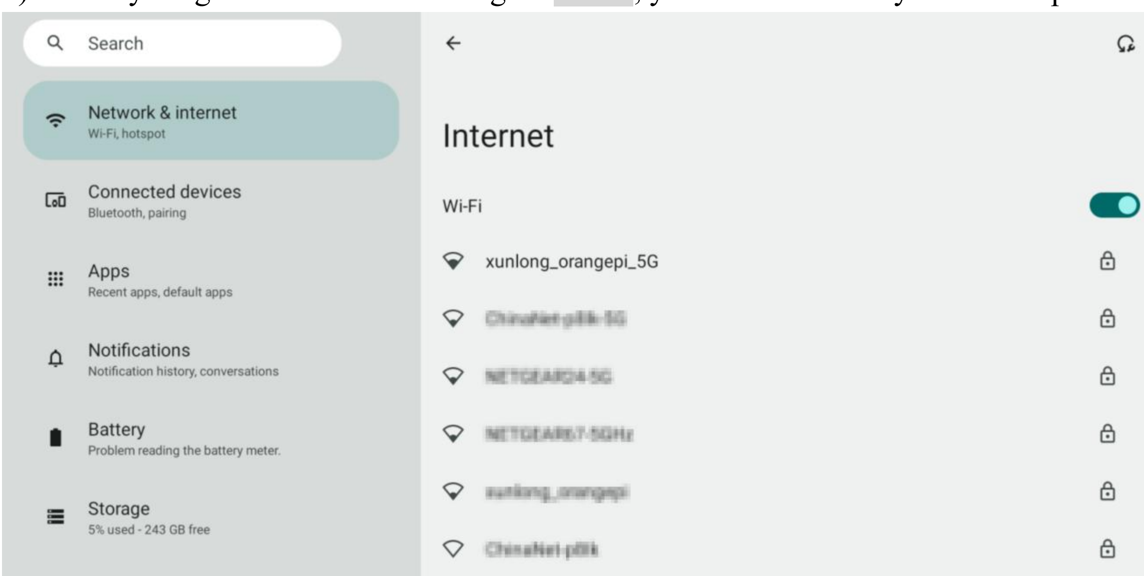
2) Then select Internet under **Network & internet**



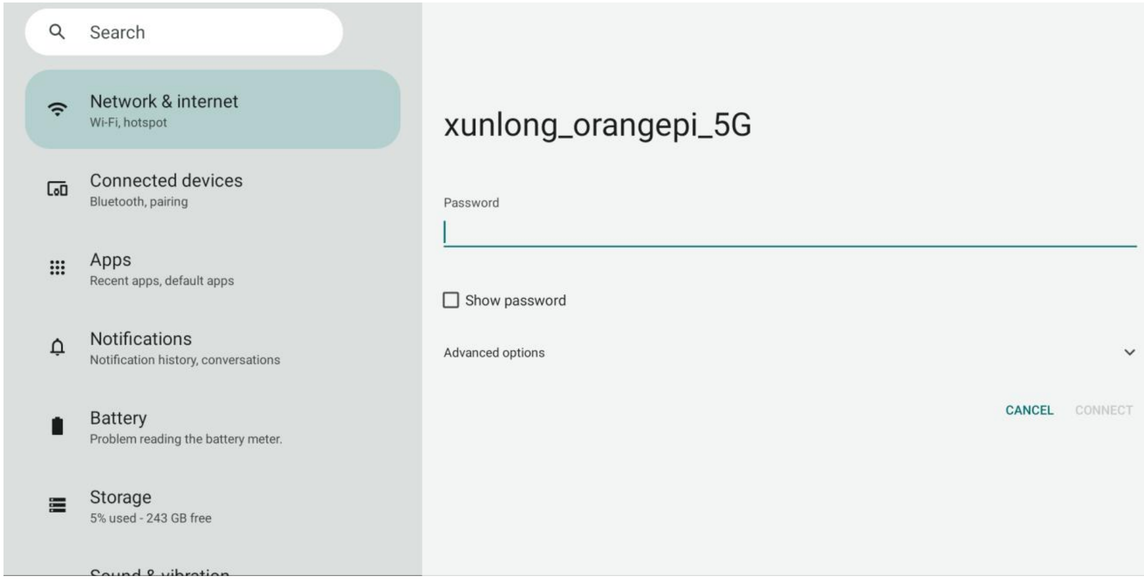
3) Then turn on the **Wi-Fi** switch



4) If everything is normal after turning on **Wi-Fi**, you can scan nearby Wi Fi hotspots

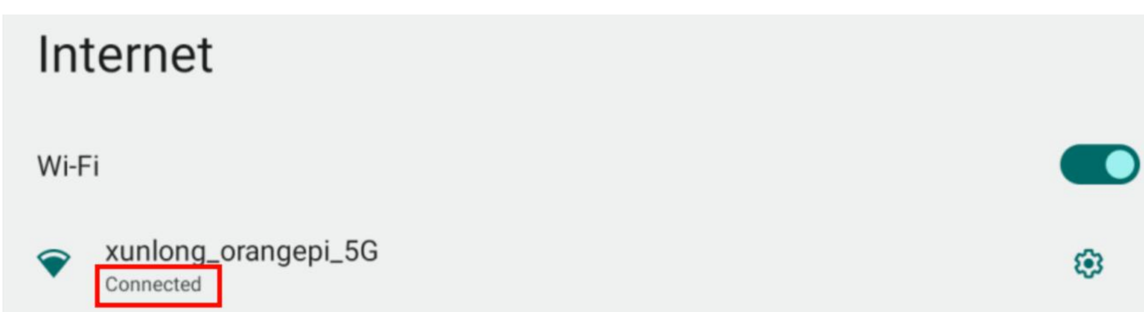


5) After selecting the Wi Fi you want to connect to, a password input interface as shown in the following figure will pop up



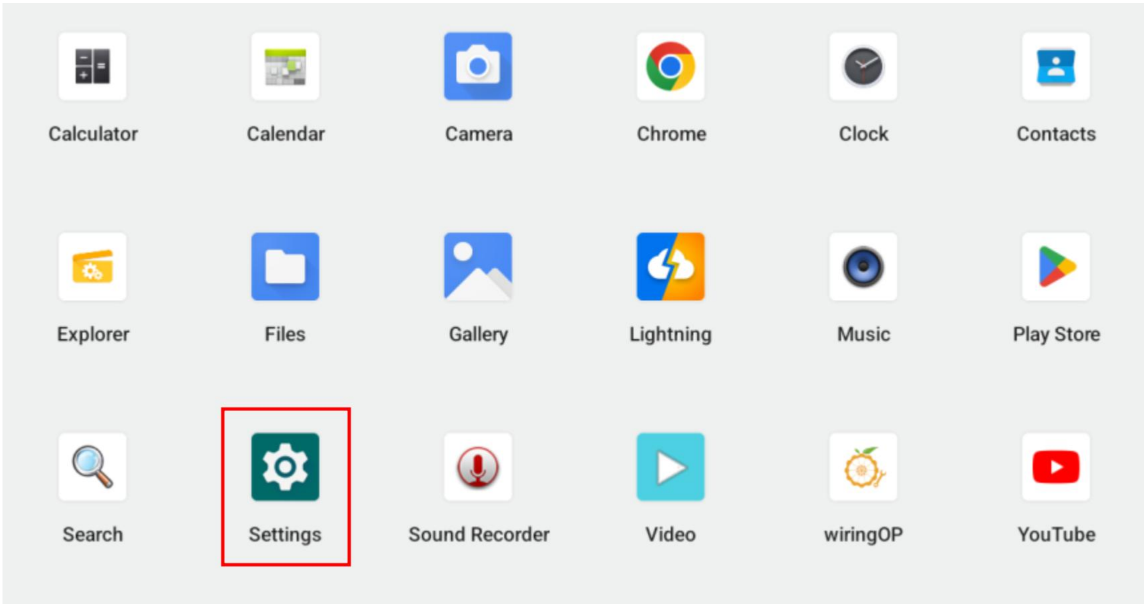
6) Then use the keyboard to enter the password corresponding to Wi Fi, and click **CONNECT** with the mouse to start connecting to Wi Fi

7) The display after successful Wi Fi connection is shown in the following figure:

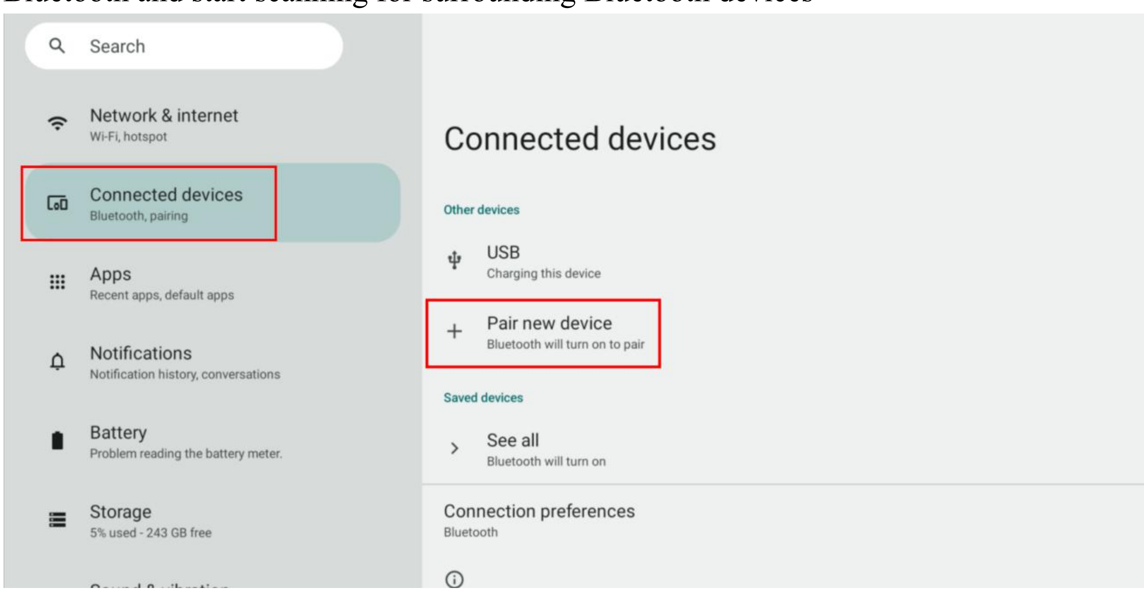


8. 4. Bluetooth testing method

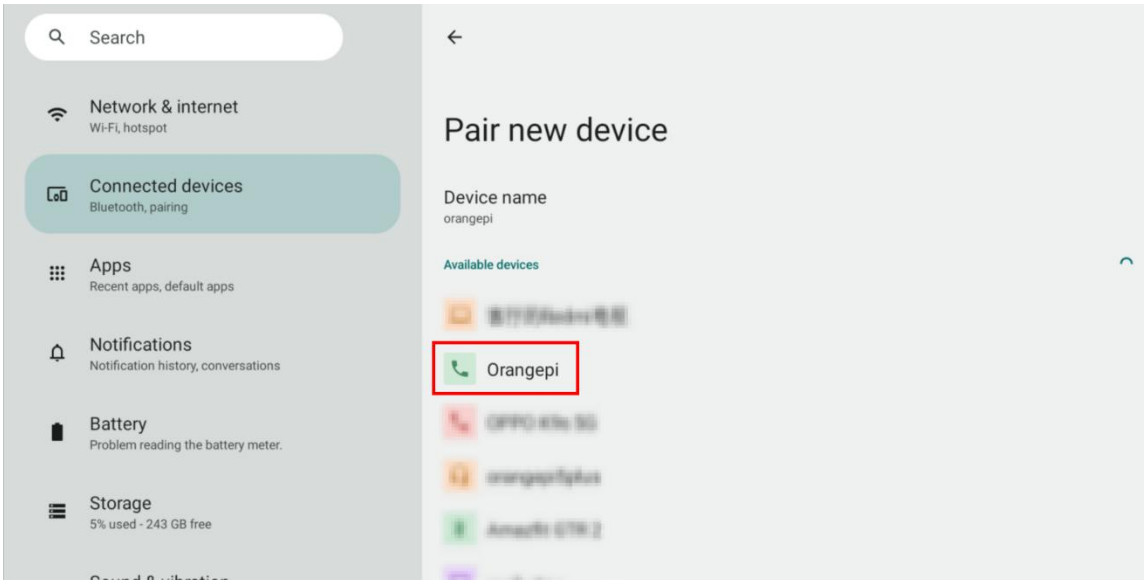
1) First, click to enter **Setting**



2) Then select the **Pair new device** option under the **Connected devices** menu to turn on Bluetooth and start scanning for surrounding Bluetooth devices



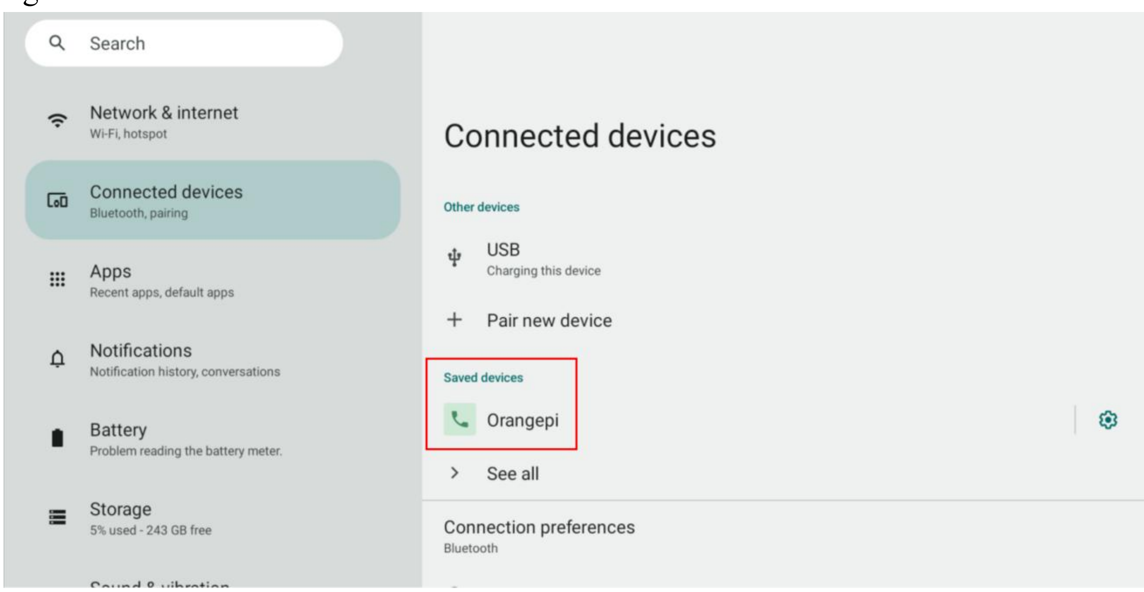
3) The Bluetooth devices found in the search will be displayed under **Available devices**



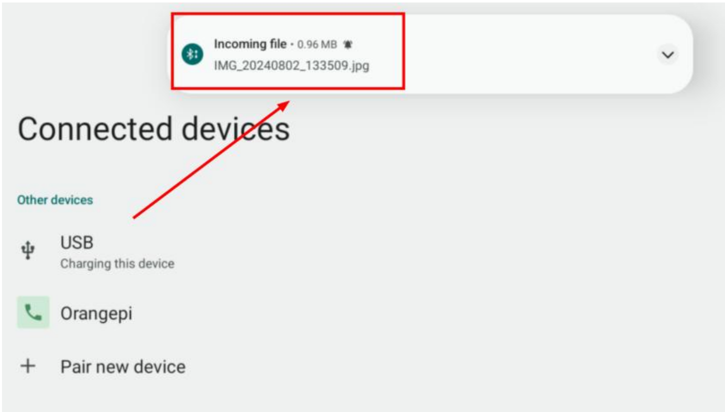
4） Then click on the Bluetooth device you want to connect to start pairing. When the interface below pops up, please use the mouse to select the **Pair** option

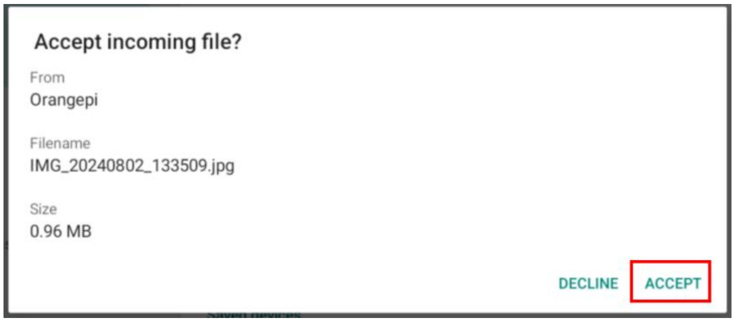
5) The test here is the configuration process of the development board and Android phone Bluetooth. At this time, the confirmation interface below will pop up on the phone. After clicking the pairing button on the phone, the pairing process will begin

6) After pairing, you can see the paired Bluetooth devices as shown in the following figure

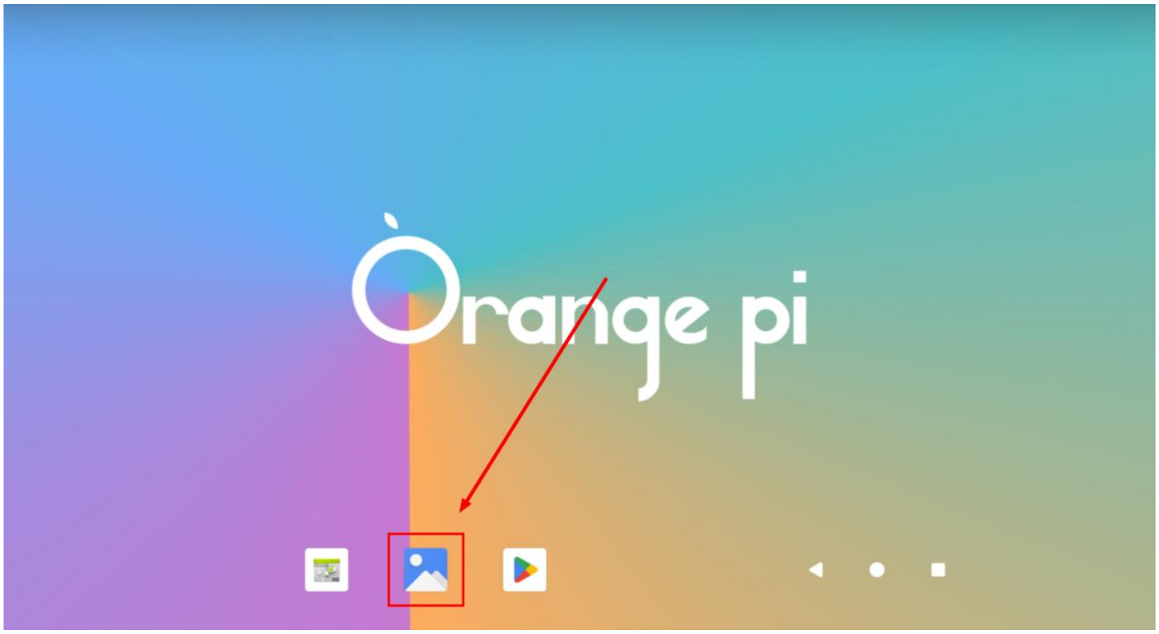


7) At this point, you can use your phone's Bluetooth to send an image to the development board. After sending, you can see a notification message in the Android system of the development board. Clicking on this message will enter the confirmation interface below, and then click **Accept** to start receiving the image sent by your phone





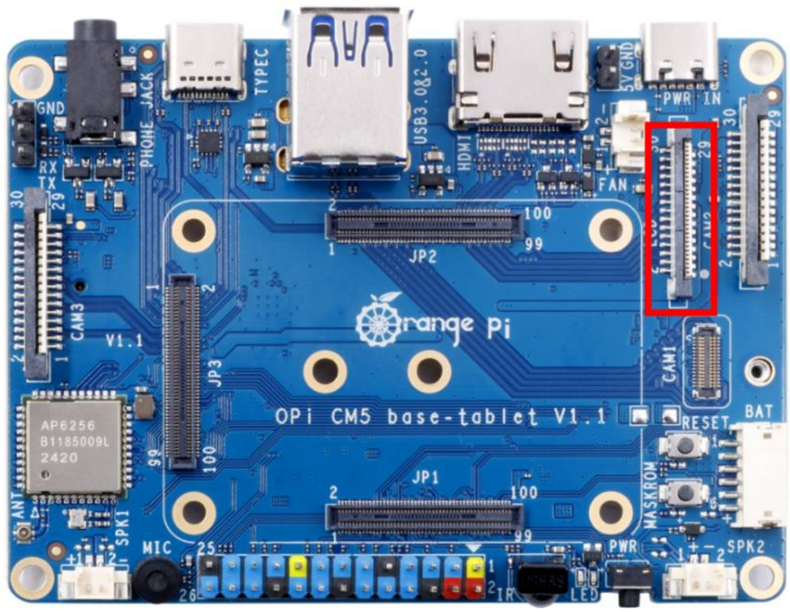
8) The images received by the Android system Bluetooth on the development board can be viewed by opening the **Download** folder in the system desktop's gallery



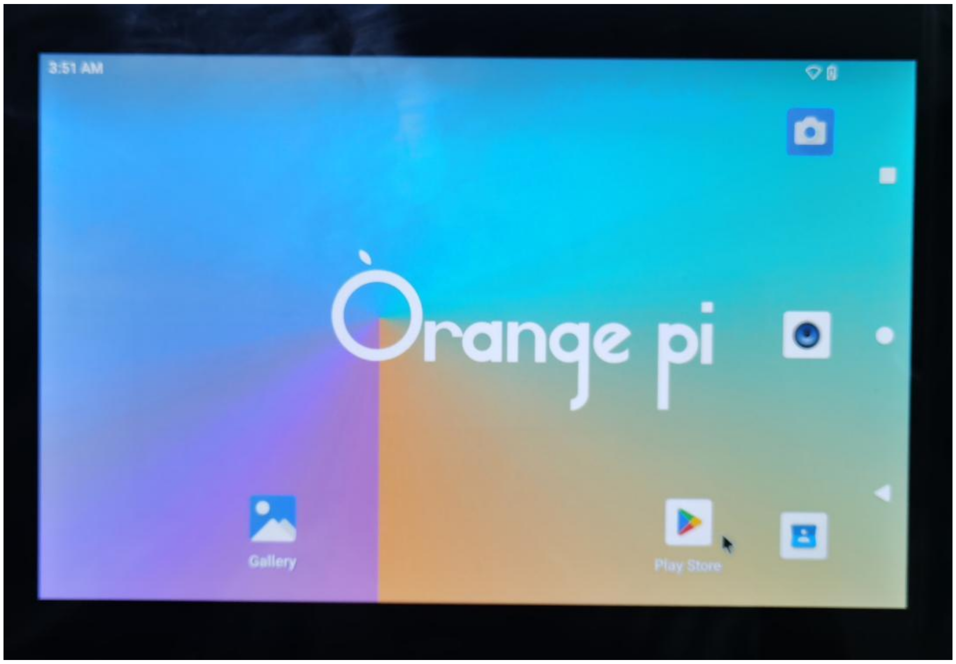
8. 5. Instructions for using 8.5.10.1 inch MIPI screen

1) Firstly, the screen needs to be assembled. Please refer to the assembly method for a 10.1-inch MIPI screen.

2) The interface position of the mipi LCD screen on the development board is shown in the following figure:



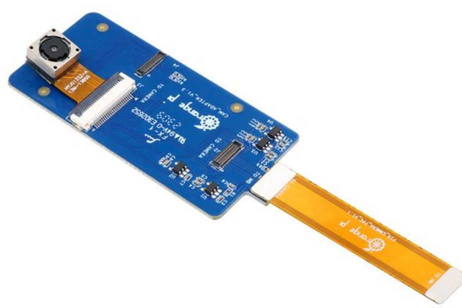
3) Connect the assembled screen to the LCD interface of the development board, connect the Type-C power supply to the board, and power it on. After the system starts up, you can see the screen display as shown in the following figure



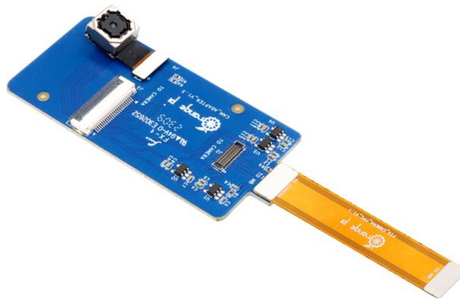
8. 6. Testing Methods for OV13850 and OV13855 MIPI Cameras

At present, the development board supports two MIPI cameras, OV13850 and OV13855. The specific images are shown below:

a. OV13850 camera with 13 million MIPI interface.



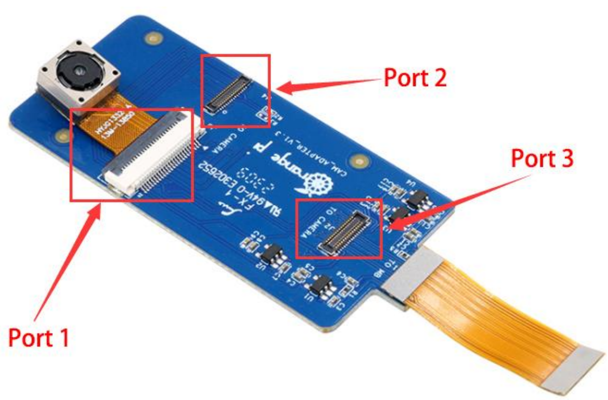
b. OV13855 camera with 13 million MIPI interface.



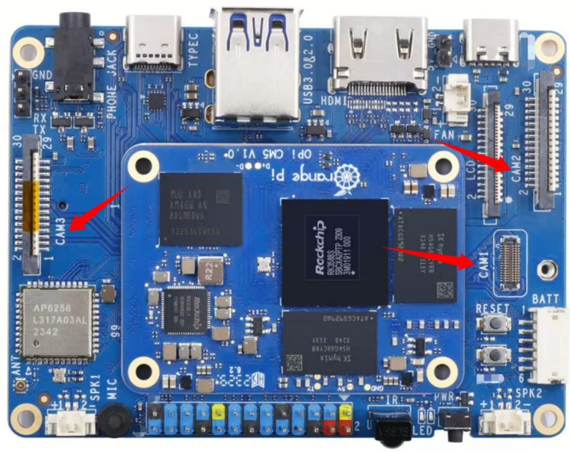
The adapter board and FPC cable used for OV13850 and OV13855 cameras are the same, except that the two cameras are connected to the adapter board at different positions. The FPC cable is shown in the following figure. Please note that the FPC cable has a direction. The end marked as **TO MB** needs to be plugged into the camera interface of the development board, and the end marked as **TO CAMERA** needs to be plugged into the camera adapter board.

There are a total of 3 camera interfaces on the camera adapter board, and only one can be connected for use at a time, as shown in the following figure. Among them:

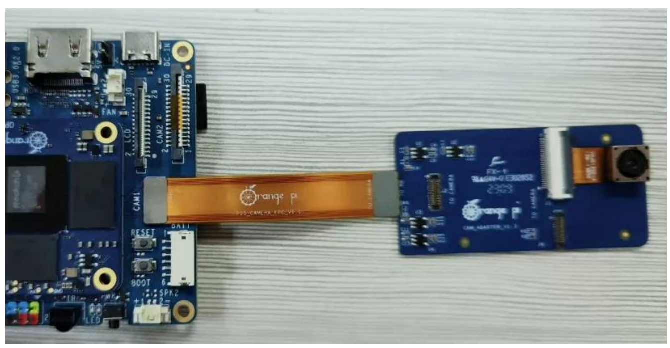
a. **Interface 1 is connected to the OV13850 camera.**
b. **Interface 2 is connected to the OV13855 camera.**
c. Interface 3 is not in use, just ignore it.



There are a total of 3 camera interfaces on the Orange Pi CM5 Base Tablet development board, and only CAM1 can be used to connect OV13850 or OV13855 cameras. We define the positions of Cam1, Cam2, and Cam3 as shown in the following figure:

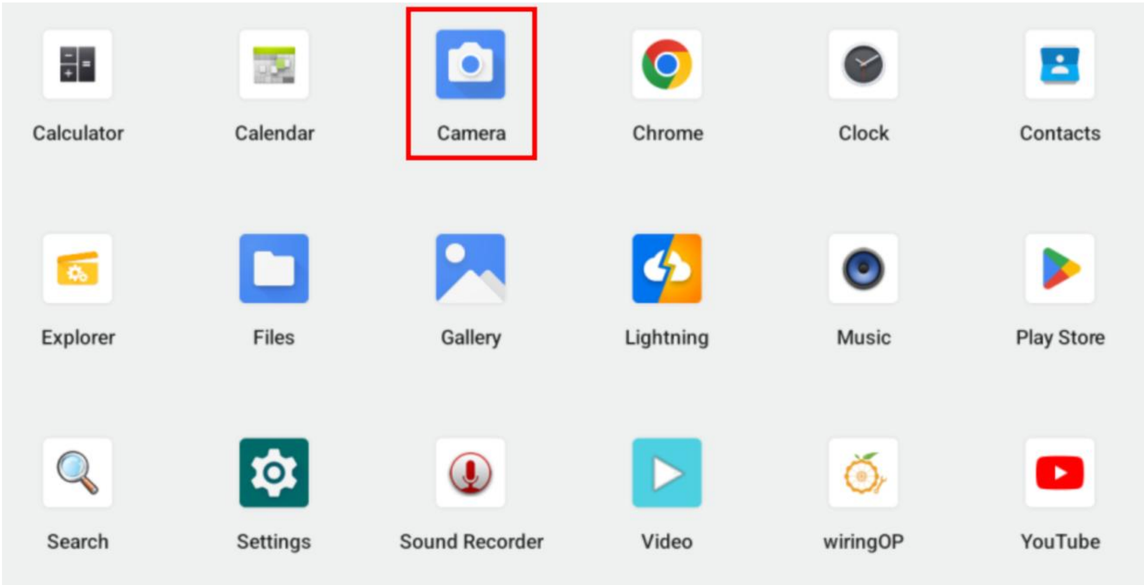


The method of inserting the camera into the Cam1 interface of the development board is as follows:

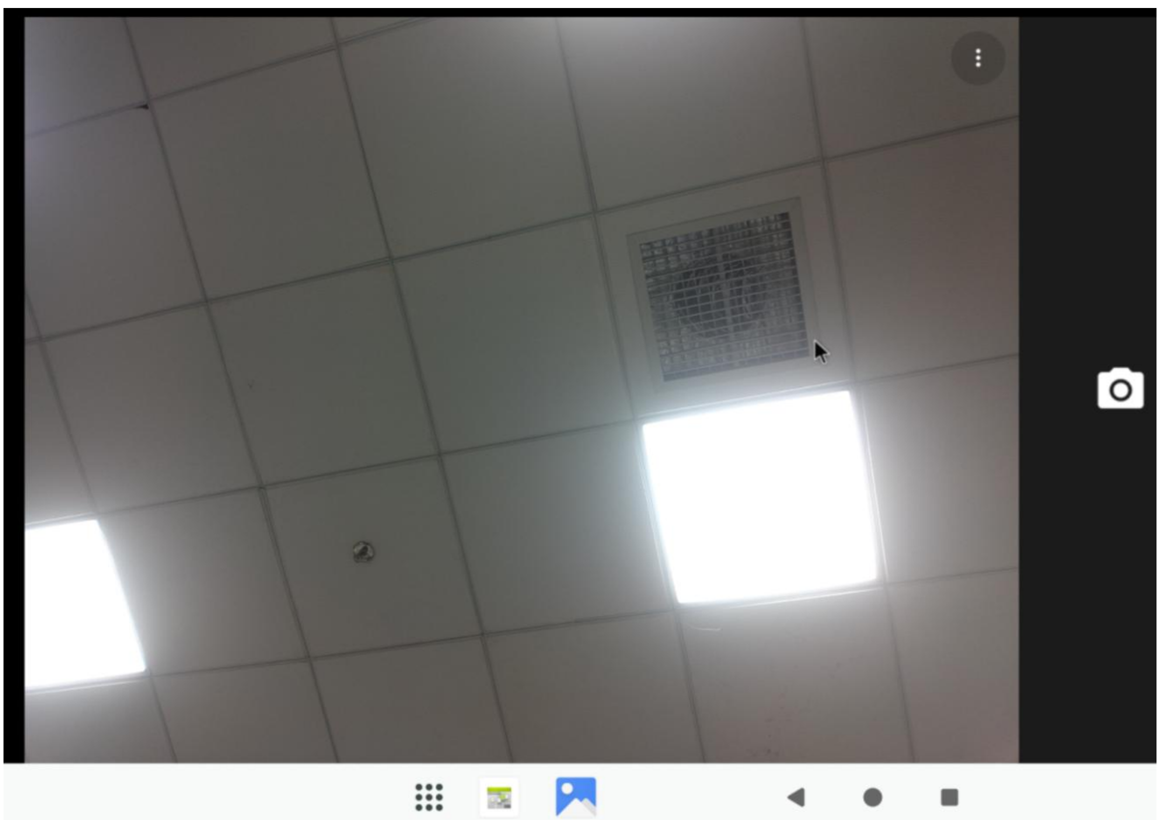


After connecting the camera to the development board, we can use the following method to test the camera:

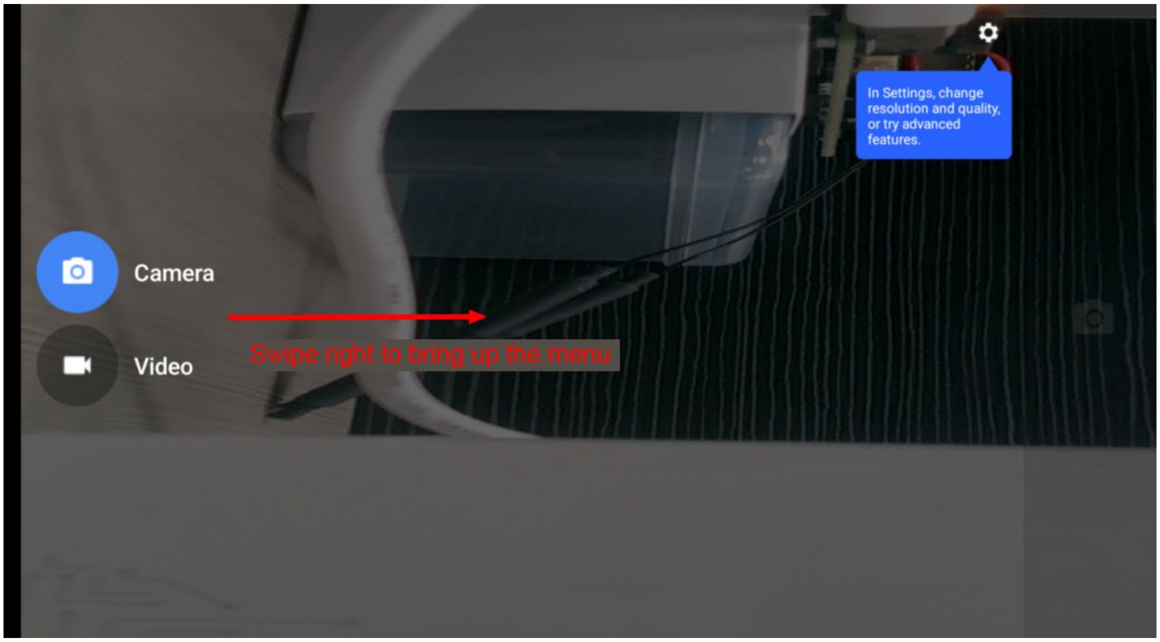
a. First, open the camera app



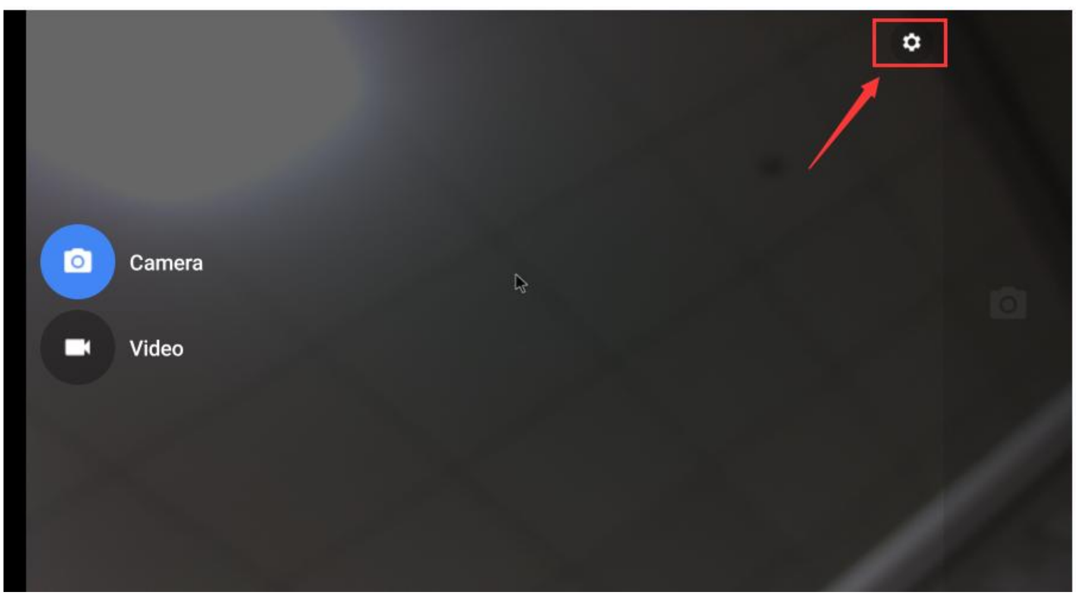
b. Then you can see the preview image of the camera



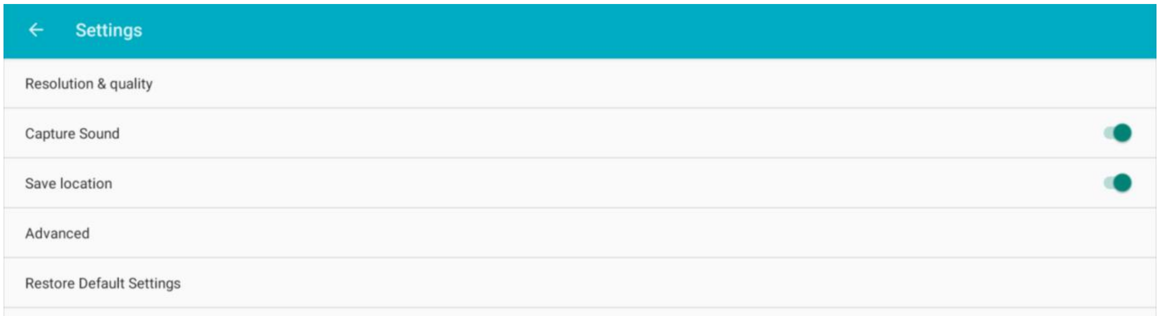
Press and hold the mouse in the area shown in the red box in the camera app, and then drag it to the right to bring up the switch interface between taking photos and filming. Click on **Video** to switch to recording mode



Click on the position shown in the following image to enter the camera settings interface

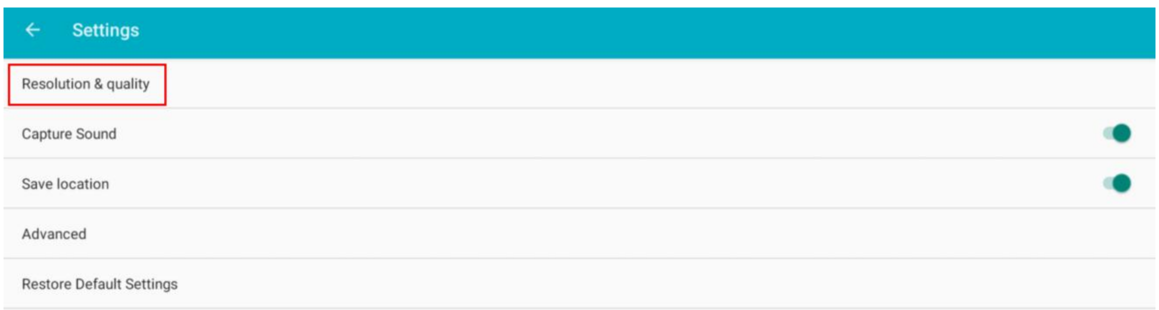


The camera settings interface is shown below:

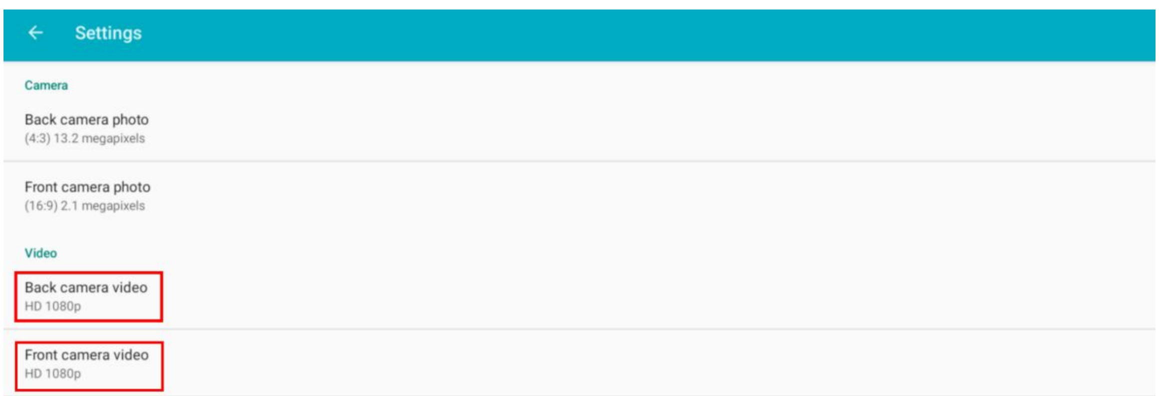


At present, OV13850 does not support 4K video recording (OV13855 supports it), and only supports 1080p at most. When recording videos, please switch the video format to 1080p in the settings. The steps are as follows:

a. Firstly, enter the settings interface of the camera app, and then click on **Resolution & quality**



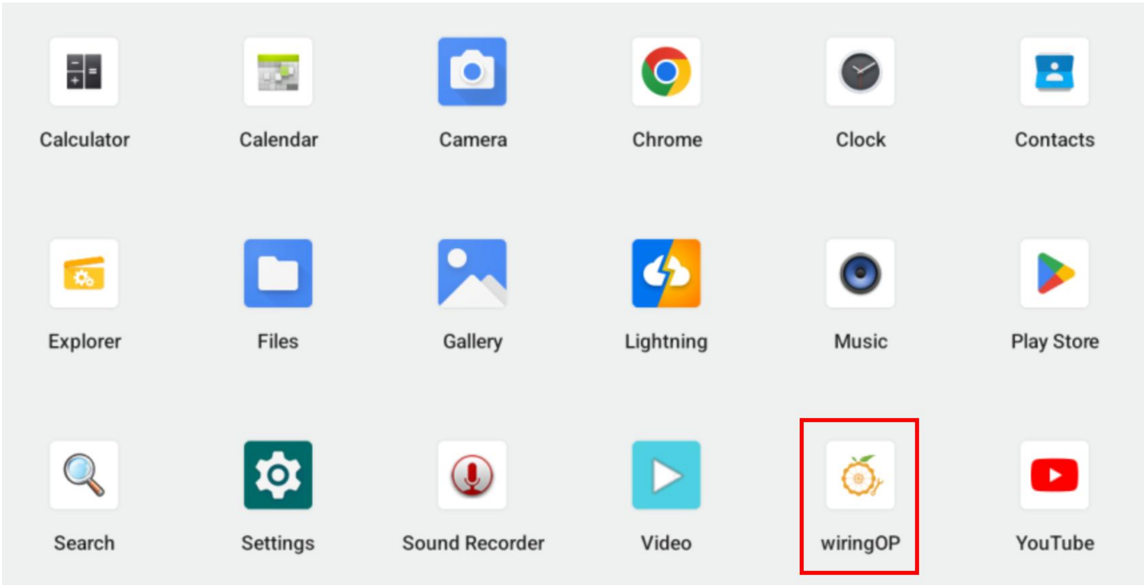
b. Then set the video format to 1080p in **Video**



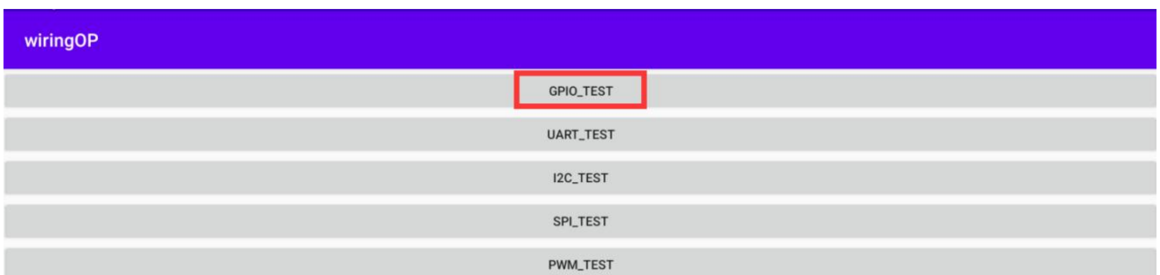
8. 7. 26 Pin interface GPIO, UART, SPI, and PWM testing

8. 7. 1. 26 Pin GPIO Port Test

1) First, click on the WiringOP icon to open the WiringOP app.

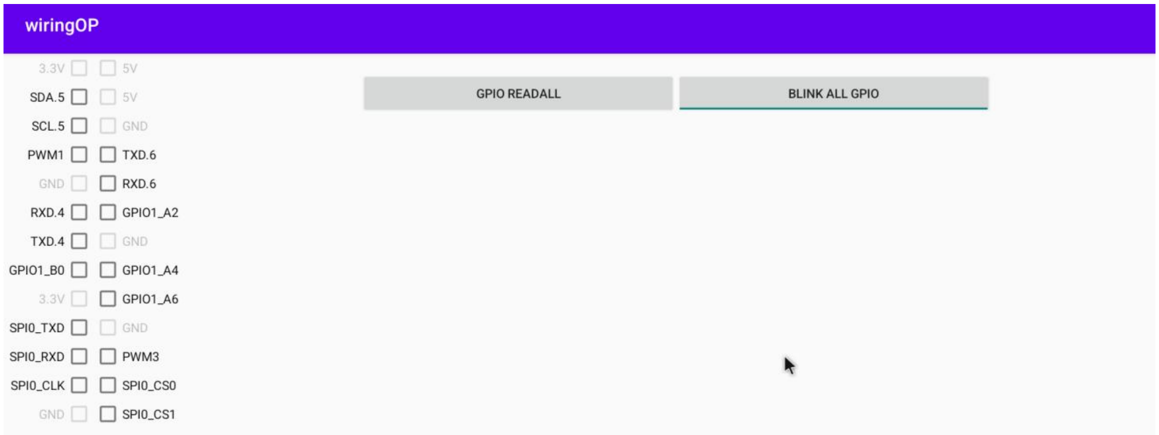


2) The main interface of WiringOP APP is displayed as shown in the following figure, and then click the **GPIO_TEST** button to open the GPIO testing interface.

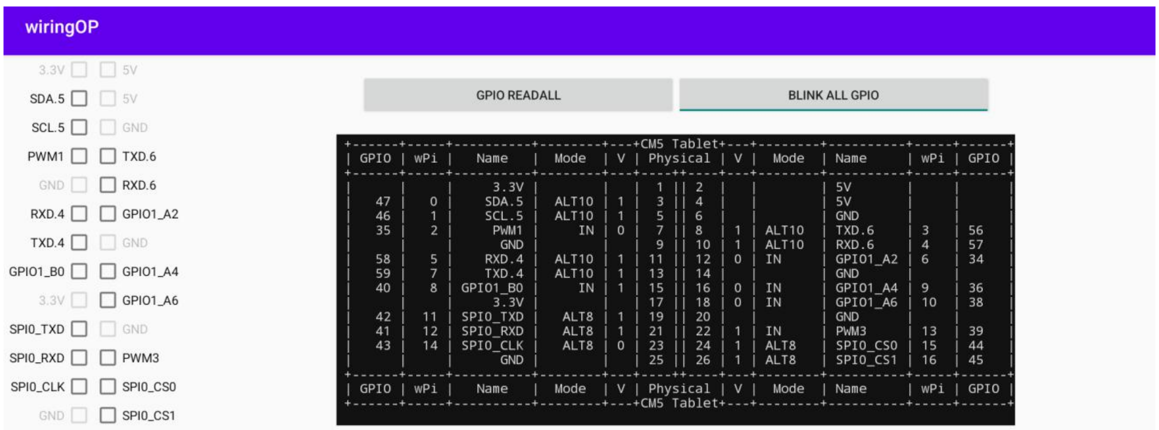


3) The GPIO testing interface is shown in the following figure, where the two rows of **CheckBox** buttons on the left correspond one-to-one with the 26 pin pin. When the **CheckBox** button is selected, the corresponding GPIO pin will be set to **OUT** mode and the pin level will be set to high level; When unchecked, the GPIO pin level is set to low level; When clicking the **GPIO READALL** button on the right, information such as wPi number, GPIO mode, and pin level can be obtained;

When clicking the **BLINK ALL GPIO** button, the program will control 17 GPIO ports to continuously switch high and low levels.

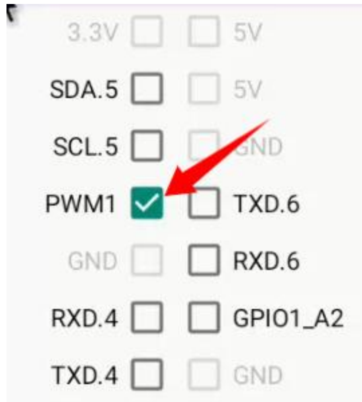


4) Then click the **GPIO READALL** button, and the output information is shown in the following figure:

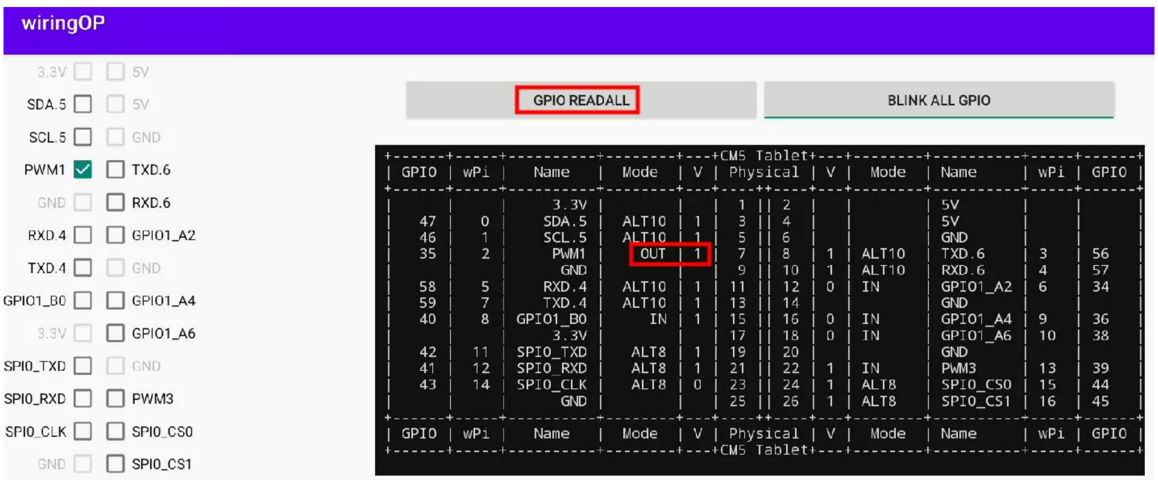


GPIO	wPi	Name	Mode	V	Physical		V	Mode	Name	wPi	GPIO
		3.3V			1	2			5V		
47	0	SDA.5	ALT10	1	3	4			5V		
46	1	SCL.5	ALT10	1	5	6			GND		
35	2	PWM1	IN	0	7	8	1	ALT10	TXD.6	3	56
		GND			9	10	1	ALT10	RXD.6	4	57
58	5	RXD.4	ALT10	1	11	12	0	IN	GPIO1_A2	6	34
59	7	TXD.4	ALT10	1	13	14			GND		
40	8	GPIO1_B0	IN	1	15	16	0	IN	GPIO1_A4	9	36
		3.3V			17	18	0	IN	GPIO1_A6	10	38
42	11	SPI0_TXD	ALT8	1	19	20			GND		
41	12	SPI0_RXD	ALT8	1	21	22	1	IN	PWM3	13	39
43	14	SPI0_CLK	ALT8	0	23	24	1	ALT8	SPI0_CS0	15	44
		GND			25	26	1	ALT8	SPI0_CS1	16	45
GPIO	wPi	Name	Mode	V	Physical		V	Mode	Name	wPi	GPIO

5) There are a total of 17 GPIO ports available in the 26 pins of the development board. Taking pin 7- corresponding to GPIO as GPO1_A3- corresponding to wPi serial number 2- as an example, we will demonstrate how to set the high and low levels of GPIO ports. Firstly, click on the **CheckBox** button corresponding to pin 7. When the button is selected, pin 7 will be set to high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **3.3v**, it means that the high level has been successfully set.

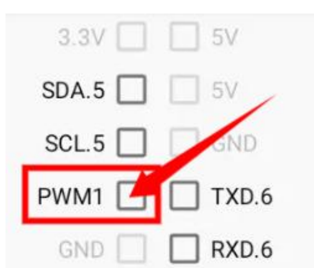


6) Then click the **GPIO READALL** button to see that the current mode of pin 7 is **OUT** and the pin level is high.

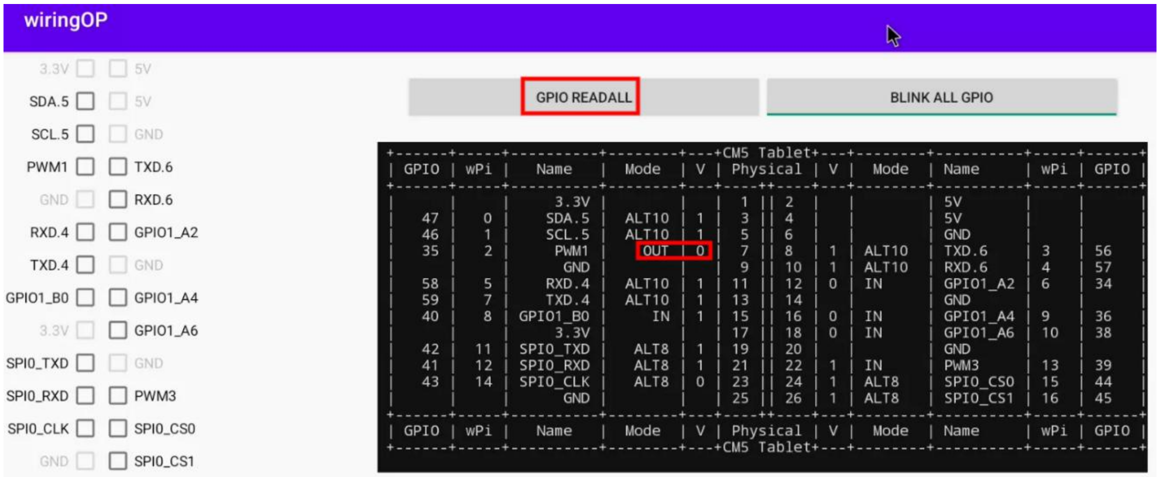


GPIO	wPi	Name	Mode	V	Physical		V	Mode	Name	wPi	GPIO
		3.3V			1	2			5V		
47	0	SDA.5	ALT10	1	3	4			5V		
46	1	SCL.5	ALT10	1	5	6			GND		
35	2	PWM1	OUT	1	7	8	1	ALT10	TXD.6	3	56
		GND			9	10	1	ALT10	RXD.6	4	57
58	5	RXD.4	ALT10	1	11	12	0	IN	GPIO1_A2	6	34
59	7	TXD.4	ALT10	1	13	14			GND		
40	8	GPIO1_B0	IN	1	15	16	0	IN	GPIO1_A4	9	36
		3.3V			17	18	0	IN	GPIO1_A6	10	38
42	11	SPI0_TXD	ALT8	1	19	20			GND		
41	12	SPI0_RXD	ALT8	1	21	22	1	IN	PWM3	13	39
43	14	SPI0_CLK	ALT8	0	23	24	1	ALT8	SPI0_CS0	15	44
		GND			25	26	1	ALT8	SPI0_CS1	16	45
GPIO	wPi	Name	Mode	V	Physical		V	Mode	Name	wPi	GPIO

7) Click the **CheckBox** button again to uncheck the status, and pin 7 will be set to low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level has been successfully set.



8) Then click the **GPIO READALL** button to see that the current mode of pin 7 is OUT and the pin level is low.

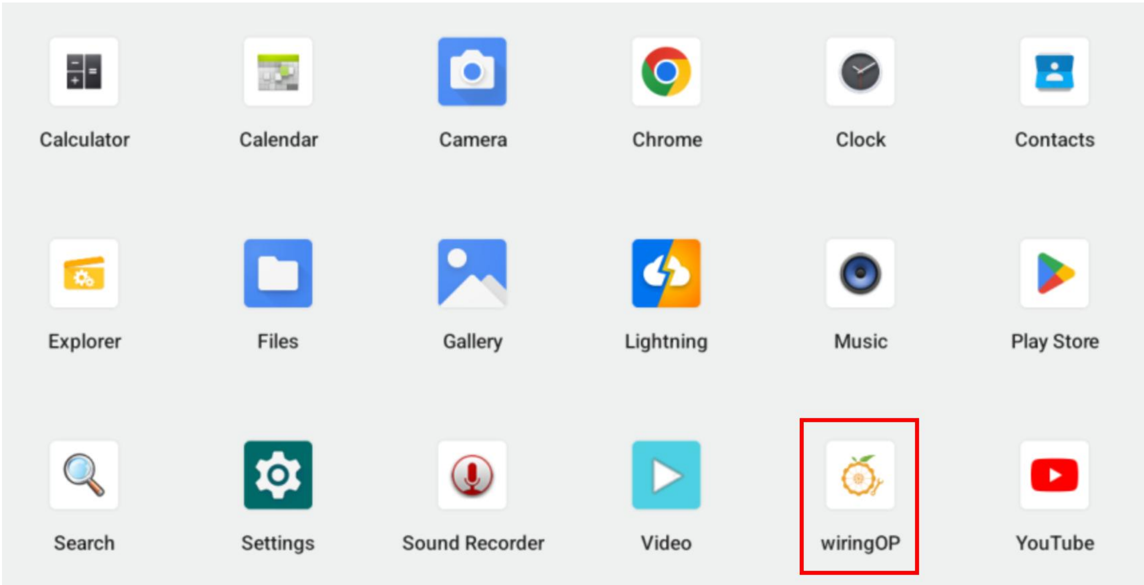


8. 7. 2. 26 Pin UART Test

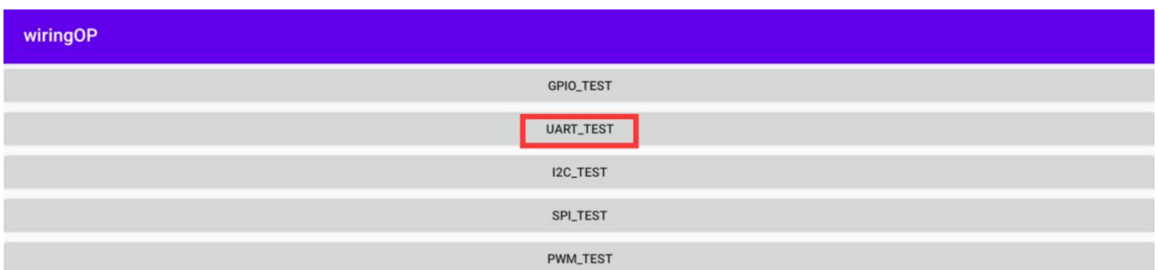
1) By default, two serial ports, UART4 and UART6, are enabled in Android. The corresponding device nodes at pin 26 are **/dev/ttyS4** and **/dev/ttyS6**.

Reuse function	GPIO	GPIO serial number	Pin number	Pin number	GPIO serial number	GPIO	Reuse function
	3.3V		1	2		5V	
I2C5_SDA_M3	GPIO1_B7	47	3	4		5V	
I2C5_SCL_M3	GPIO1_B6	46	5	6		GND	
PWM1_M2	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2
	GND		9	10	57	GPIO1_D1	UART6_RX_M2
UART4_TX_M0	GPIO1_D2	58	11	12	34	GPIO1_A2	
UART4_RX_M0	GPIO1_D3	59	13	14		GND	
	GPIO1_B0	40	15	16	36	GPIO1_A4	
	3.3V		17	18	38	GPIO1_A6	
SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND	
SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3
SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2
	GND		25	26	45	GPIO1_B5	SPI0_CS1_M2

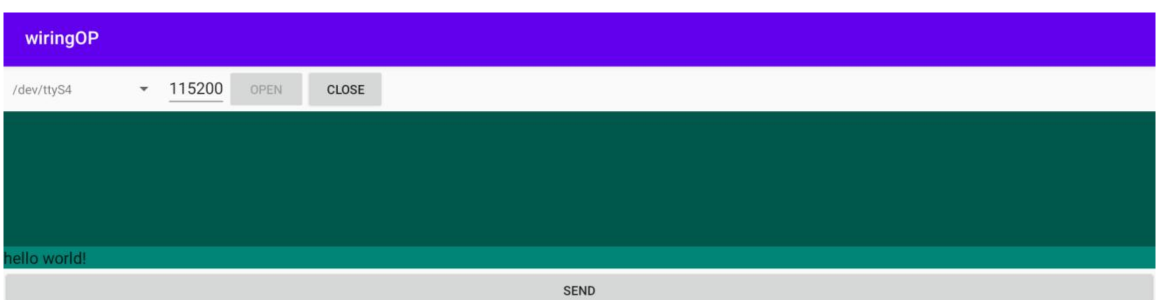
2) First, click on the WiringOP icon to open the WiringOP app.



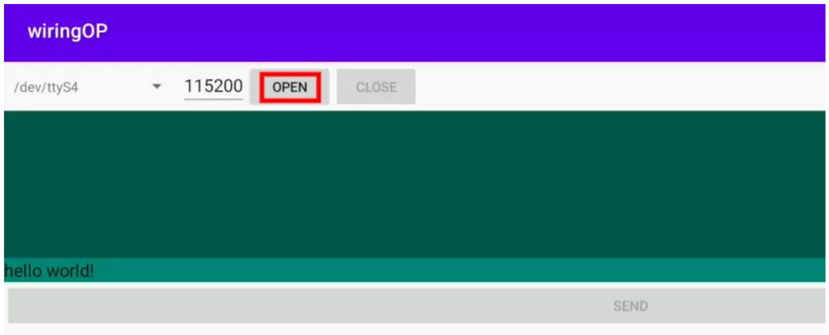
3) The main interface of WiringOP APP is displayed as shown in the following figure, and then click the **UART_TEST** button to open the UART test interface.



4) The serial port testing interface of the APP is shown in the following figure:

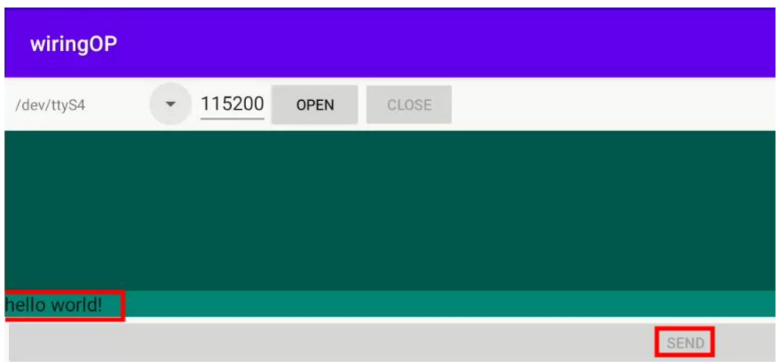


5) Next, select the serial port you want to test, such as **/dev/ttyS4**, then enter the desired baud rate in the edit box, and click the **OPEN** button to open the **/dev/ttyS4** node. After successful opening, the **OPEN** button becomes unselectable, and the **CLOSE** and **SEND** buttons become selectable.

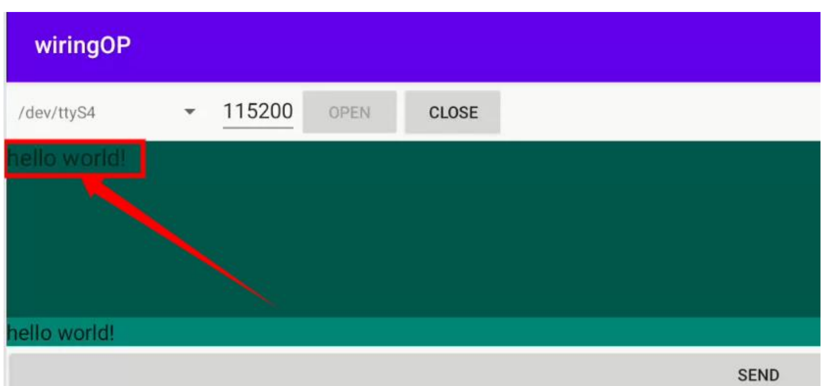


6) Then use DuPont wire to short-circuit the RXD and TXD pins of UART 4.

7) Then you can enter a character in the send edit box below and click the **SEND** button to start sending.



8) If everything is normal, the received string will be displayed in the receiving box.



8. 7. 3. 26 Pin SPI Test

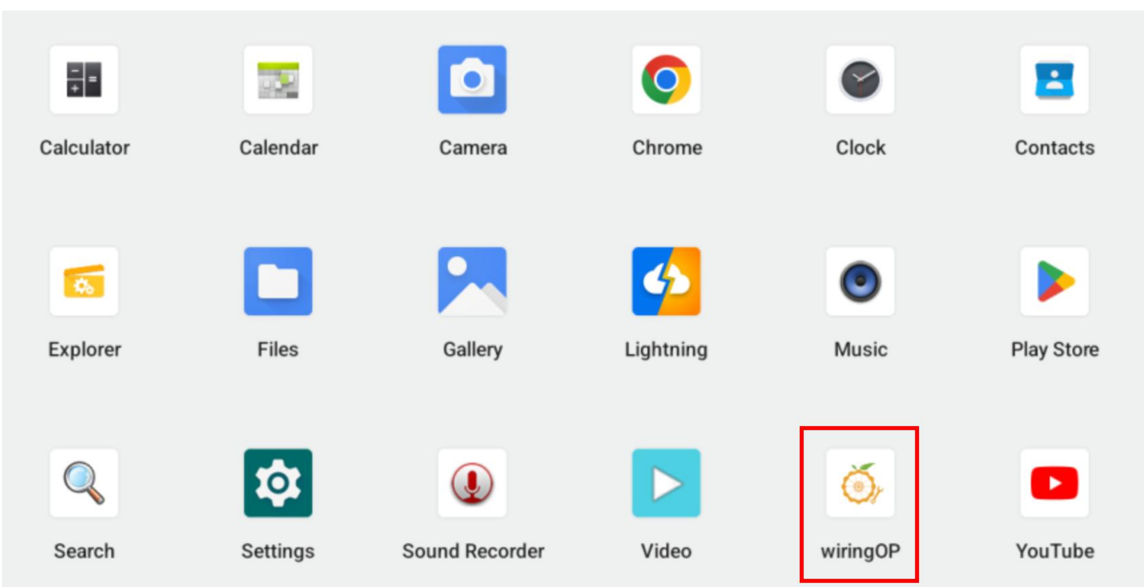
1) As shown in the table below, the SPI available for the development board is SPI0.

Reuse function	GPIO	GPIO serial	Pin number	Pin number	GPIO serial	GPIO	Reuse function

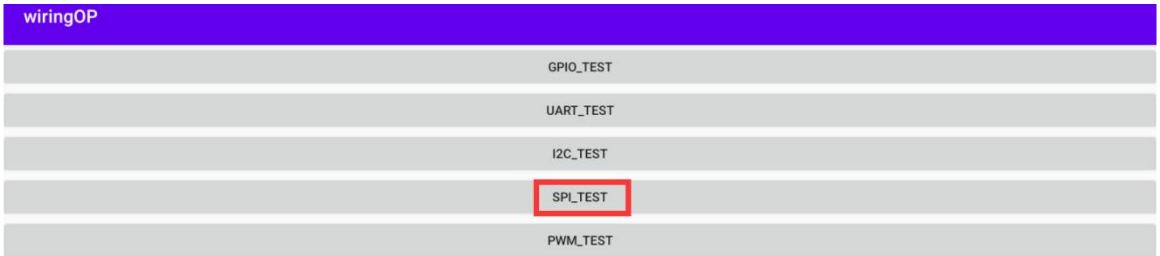
		number			number		
	3.3V		1	2		5V	
I2C5_SDA_M3	GPIO1_B7	47	3	4		5V	
I2C5_SCL_M3	GPIO1_B6	46	5	6		GND	
PWM1_M2	GPIO1_A3	35	7	8	56	GPIO1_D0	UART6_TX_M2
	GND		9	10	57	GPIO1_D1	UART6_RX_M2
UART4_TX_M0	GPIO1_D2	58	11	12	34	GPIO1_A2	
UART4_RX_M0	GPIO1_D3	59	13	14		GND	
	GPIO1_B0	40	15	16	36	GPIO1_A4	
	3.3V		17	18	38	GPIO1_A6	
SPI0_MOSI_M2	GPIO1_B2	42	19	20		GND	
SPI0_MISO_M2	GPIO1_B1	41	21	22	39	GPIO1_A7	PWM3_M3
SPI0_CLK_M2	GPIO1_B3	43	23	24	44	GPIO1_B4	SPI0_CS0_M2
	GND		25	26	45	GPIO1_B5	SPI0_CS1_M2

2) Here, the w25q64 module is used to test the SPI interface. First, connect the w25q64 device to the SPI0 interface.

3) Then click on the WiringOP icon to open the WiringOP app.

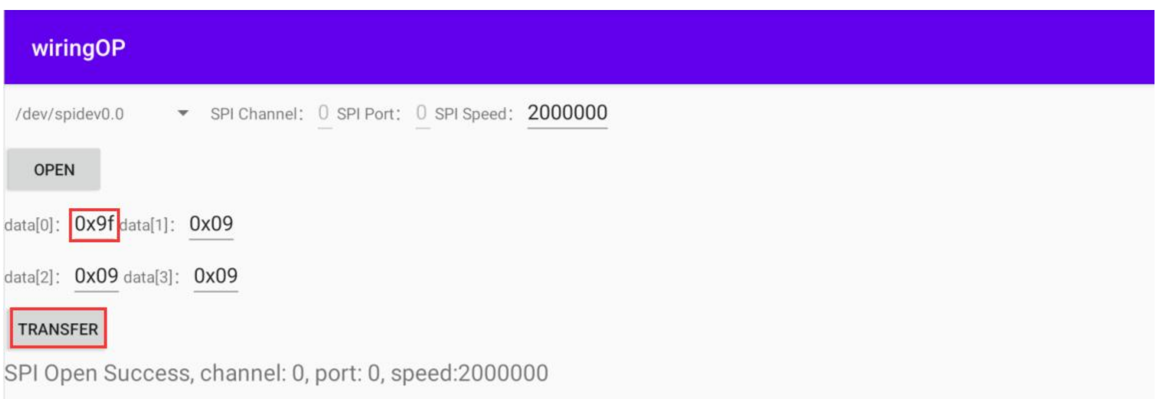


4) The main interface of WiringOP APP is displayed as shown in the following figure. Click the SPI_TEST button to open the SPI testing interface.

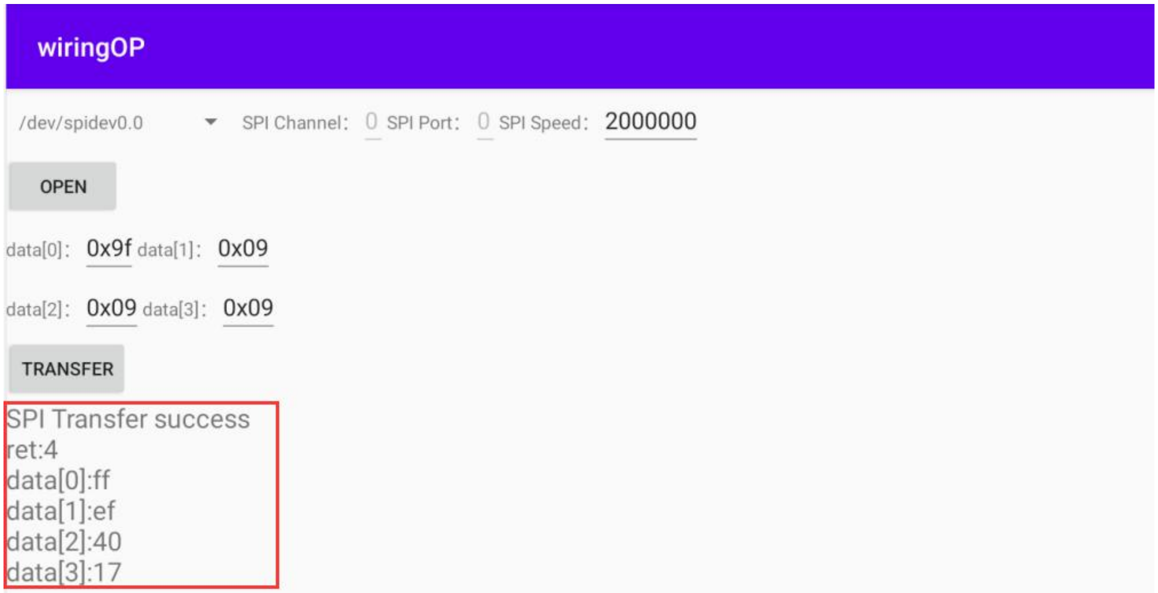


5) Then click the **OPEN** button to initialize SPI.

6) Then fill in the bytes that need to be sent, such as reading the ID information of w25q64, filling in the address 0x9f in data [0], and then clicking the **TRANSFER** button.



7) Finally, the app will display the read ID information.



8) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, which corresponds to the values read above (h represents hexadecimal)

MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

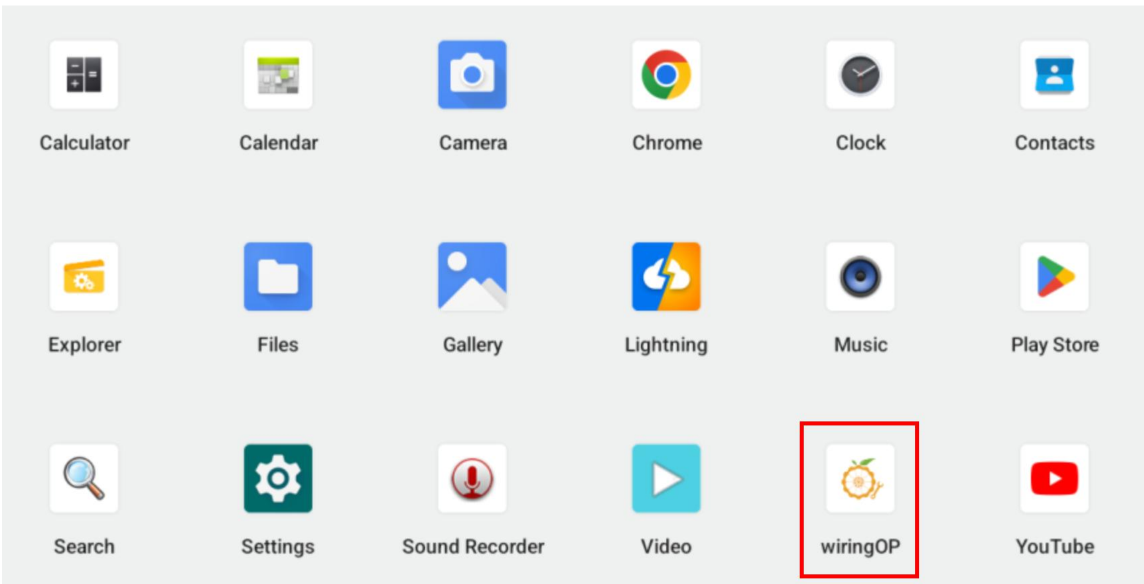
8. 7. 4. 26 Pin PWM Test

1) Android has enabled **PWM1** and **PWM3** by default, and the corresponding pins are located at pin 26 as shown in the table below:

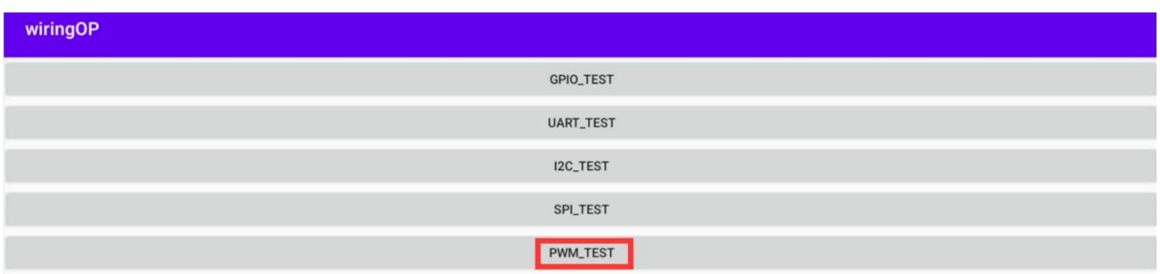
Reuse function	GPIO serial number	Pin number	Pin number	GPIO serial number	GPIO	Reuse function
		1	2		5V	
I2C5_SDA_M3	47	3	4		5V	
I2C5_SCL_M3	46	5	6		GND	
PWM1_M2	35	7	8	56	GPIO1_D0	UART6_TX_M2
		9	10	57	GPIO1_D1	UART6_RX_M2
UART4_TX_M0	58	11	12	34	GPIO1_A2	
UART4_RX_M0	59	13	14		GND	

	40	15	16	36	GPIO1_A4	
		17	18	38	GPIO1_A6	
SPI0_MOSI_M2	42	19	20		GND	
SPI0_MISO_M2	41	21	22	39	GPIO1_A7	PWM3_M3
SPI0_CLK_M2	43	23	24	44	GPIO1_B4	SPI0_CS0_M2
		25	26	45	GPIO1_B5	SPI0_CS1_M2

2) First, click on the WiringOP icon to open the WiringOP app.



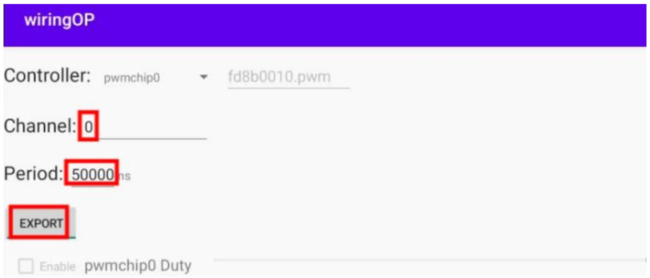
3) Then click the **PWM_TEST** button on the main interface of WirgOP to enter the PWM testing interface.



4) The base address of PWM1 is **fd8b0010**, and the base address of PWM3 is **fd8b0030**. Here, **fd8b0010.pwm** is displayed on the right side of pwmchip0, indicating that PWM1 is selected by default. If you want to select PWM3, you need to click on the drop-down option to choose another pwmchip. When **fd8b0030.pwm** is displayed on the right, it

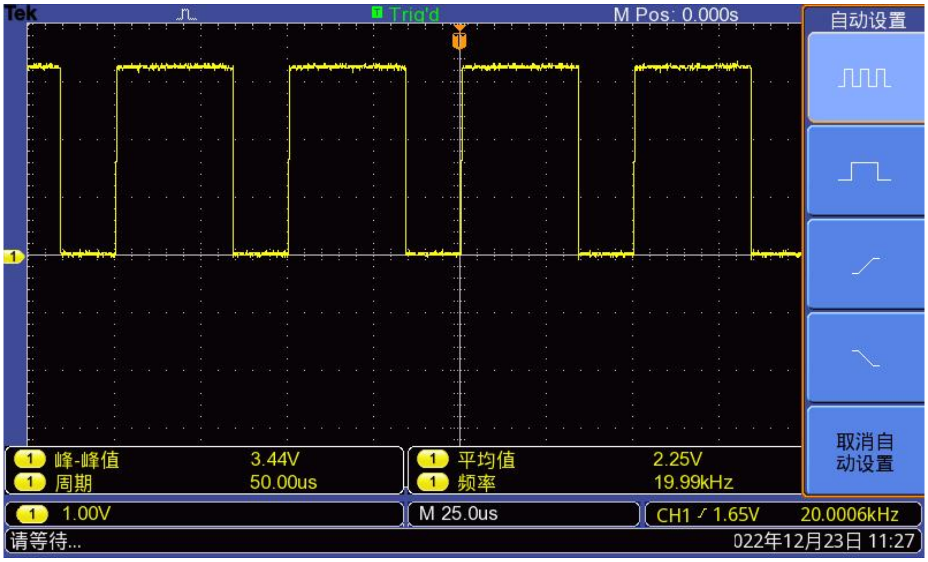
means that **PWM3** has been selected.

5) Then confirm the PWM channel, which defaults to channel 0, and confirm the PWM cycle. The default configuration is **50000ns**, and the converted PWM frequency is **20KHz**, which can be modified by clicking the **EXPORT** button to export **PWM1**.



6) Then drag the drag bar below to change the PWM duty cycle, and select Enable to output the PWM waveform.

7) Then use an oscilloscope to measure pin 7 of the 26 pins on the development board to see the waveform below.

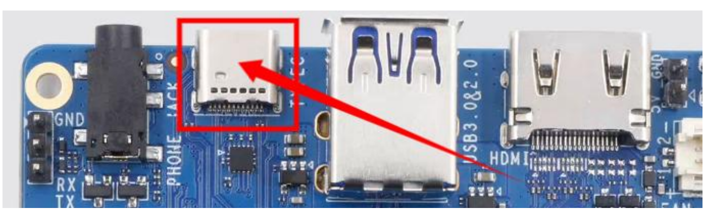


8. 8. Instructions for Using ADB

8. 8. 1. Use a data cable to connect adb for debugging

1) Firstly, prepare a high-quality Type-C data cable.

2) Then connect the development board to Ubuntu PC through a Type-C data cable. The location of the Type-C interface on the development board is shown in the following figure:



3) Then install adb tools on Ubuntu PC.

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

4) The recognized ADB devices can be viewed through the following command.

```
test@test:~$ adb devices
List of devices attached
S63QCF54CJ      device
test@test:~$ lsusb
Bus 003 Device 006: ID 2207:0006
```

5) Then you can log in to the Android system through adb shell on Ubuntu PC.

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

6) Execute the following command to remount the Android system.

```
test@test:~$ adb root
test@test:~$ adb remount
```

7) Then you can transfer the file to the Android system.

```
test@test:~$ adb push example.txt /system/
```

8. 8. 2. Debugging with network connection adb

Using network adb does not require a data cable with Typc-C interface to connect the computer and development board, but communicates through the network. Therefore, first make sure that the wireless network of the development board is connected, and then obtain the IP address of the development board, which will be used later.

1) Ensure that the **service.adb.tcp.port** of the Android system is set to port number 5555.

```
console:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the port number of the network adb.

```
console:/ # setprop service.adb.tcp.port 5555
console:/ # stop adbd
console:/ # start adbd
```

3) Install adb tools on Ubuntu PC.

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

4) Then connect the network adb on Ubuntu PC.

```
test@test:~$ adb connect 192.168.1.xxx      (The IP address needs to be changed to the IP address of the development board)
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xxx:5555

test@test:~$ adb devices
List of devices attached
192.168.1.xxx:5555          device
```

5) Then you can log in to the Android system through the adb shell on an Ubuntu PC.

```
test@test:~$ adb shell
console:/ #
```

9. Appendix

9. 1. User Manual Update History

version	Date	Update Explanation
v1.0	2024-08-20	Initial version

9. 2. Image Update History

Date	Update Explanation
2024-08-20	Orangepicm5-tablet_1.0.0_ubuntu_jammy_server_linux6.1.43.7z

	Orangepicm5-tablet_1.0.0_ubuntu_focal_server_linux5.10.160.7z Orangepicm5-tablet_1.0.0_ubuntu_jammy_server_linux5.10.160.7z Orangepicm5-tablet_1.0.0_debian_bullseye_server_linux5.10.160.7z Orangepicm5-tablet_1.0.0_debian_bookworm_server_linux6.1.43.7z Orangepicm5-tablet_1.0.0_debian_bookworm_server_linux5.10.160.7z Orangepicm5-tablet_1.0.0_ubuntu_jammy_desktop_xfce_linux6.1.43.7z Orangepicm5-tablet_1.0.0_ubuntu_focal_desktop_xfce_linux5.10.160.7z Orangepicm5-tablet_1.0.0_ubuntu_jammy_desktop_xfce_linux5.10.160.7z Orangepicm5-tablet_1.0.0_debian_bullseye_desktop_xfce_linux5.10.160.7z Orangepicm5-tablet_1.0.0_debian_bookworm_desktop_xfce_linux6.1.43.7z Orangepicm5-tablet_1.0.0_debian_bookworm_desktop_xfce_linux5.10.160.7z Opios-droid-aarch64-opicm5-tablet-24.11-linux5.10.160-en.tar.gz Opios-droid-aarch64-opicm5-tablet-24.11-linux5.10.160.tar.gz OrangePiCM5-TABLET_RK3588S_Android13_v1.0.0.tar.gz Opios-arch-aarch64-gnome-opicm5_tablet-24.09-linux5.10.160.img.xz * Update Explanation