Orange Pi R1 Plus LTS
User Manual

Contents

1. Basic Features of Orange Pi R1 Plus LTS

1.1. What is Orange Pi R1 Plus LTS

Orange Pi is an open source single-board card computer, a new generation of arm64 development boards, which can run operating systems such as Ubuntu, Debian and OpenWRT. The Orange Pi R1 Plus LTS uses Rockchip rk3328 SoC and has 1GB LPDDR3 memory

1.2. Purpose of Orange Pi R1 Plus LTS

Typical application:

- A router
- One switch

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

1.3. Who is Orange Pi R1 Plus LTS designed for?

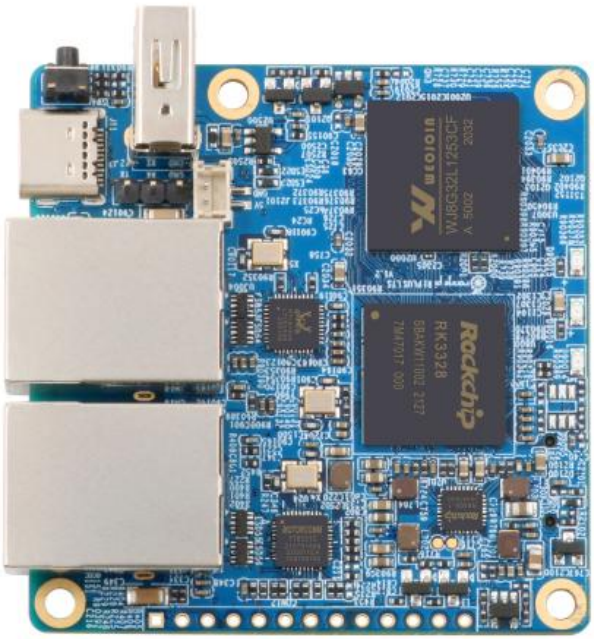
The Orange Pi development board is not just a consumer product, it is designed for anyone who wants to use technology to create and innovate. It's a very simple, fun, and useful tool that you can use to shape the world around you

1. 4. Hardware Features of Orange Pi R1 Plus LTS

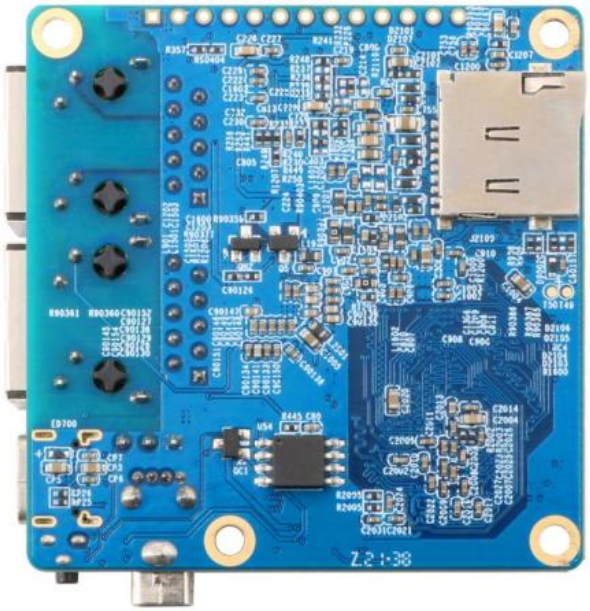
Hardware Features Introduction	
CPU	Rockchip RK3328 Quad-core ARM Cortex-A53 64-bit processor, up to 1.5GHz
GPU	Mali-450MP2 Supports OpenGL ES 1.0/2.0
Power Management Chip	RK805
Memory	1GB LPDDR3(shared with GPU)
Onboard Storage	• TF card slot • 16MB SPI Flash
Onboard Ethernet	10M/100M/1000M Ethernet（YT8531C） 10M/100M/1000M USB Ethernet（RTL8153B）
Video output	TV CVBS output(via 13pin adapter board)
Audio output	3.5mm Audio port (via 13pin adapter board)
power supply	USB Type C interface 5V/2A input
USB	1 个 USB 2.0 HOST
Low-level peripherals	13pin header with UART、I2C、IR pin、Tv-out、AUDIO(No MIC)、2xUSB2.0（not support）、and 1 GPIO port
Debug serial port	UART-TX, UART-RX and GND
button	1x reset key
Fan interface	1x cooling fan interface (5V)
LED lights	Power indicator and network port status indicator
Infrared reception	Support infrared remote control (via 13pin adapter board)
Supported Operating Systems	Ubuntu, Debian, OpenWRT and other operating systems
Appearance specification introduction	
Product Size	56mm×57mm
weight	30.5g

1. 5. Top view and bottom view of Orange Pi R1 Plus LTS

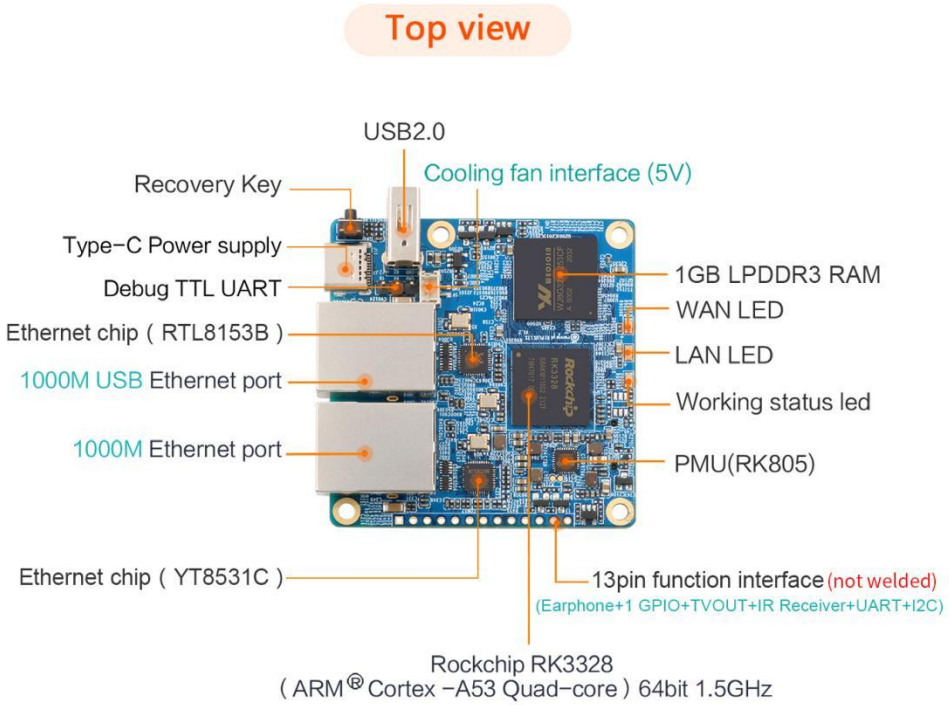
top view：

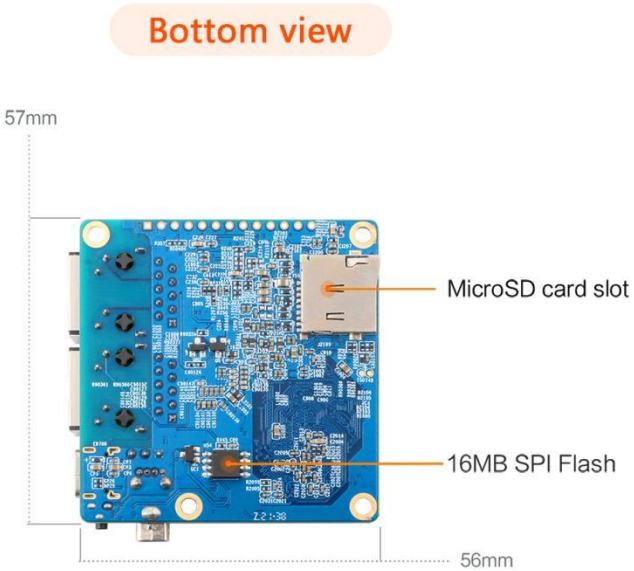


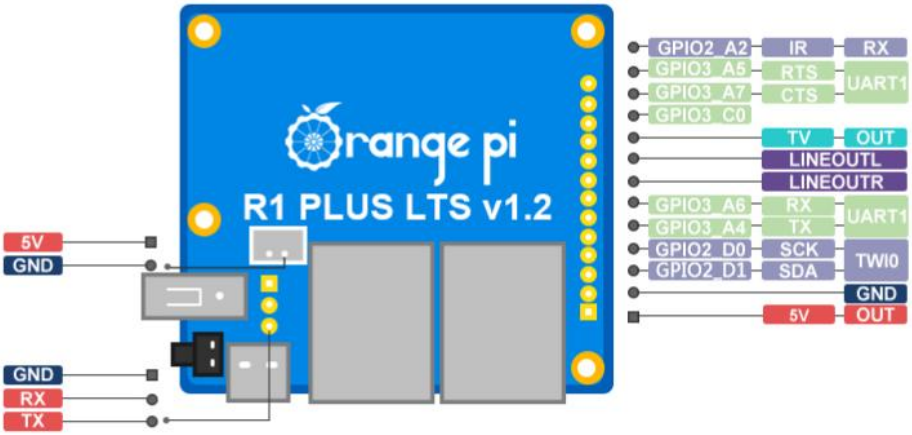
bottom view：



1. 6. Interface details of Orange Pi R1 Plus LTS






R1 PLUS LTS v1.2
5V
GND
GND
RX
TX
GPIO2_A2
IR
RX
GPIO3_A5
RTS
GPIO3_A7
CTS
UART1
GPIO3_C0
TV
OUT
LINEOUTL
LINEOUTR
GPIO3_A6
RX
GPIO3_A4
TX
UART1
GPIO2_D0
SCK
GPIO2_D1
SDA
TWI0
GND
5V
OUT

2. Introduction to the use of the development board

2. 1. Prepare the necessary accessories

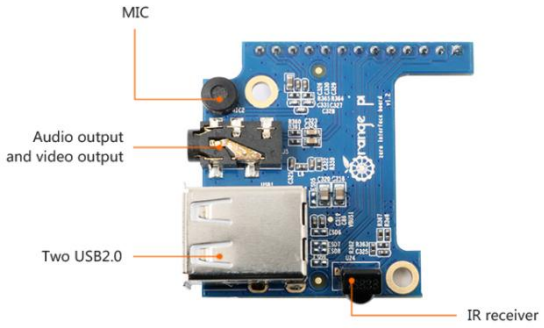
1) TF card, a high-speed card of **class 10** or above with a minimum capacity of 8GB, it is recommended to use a SanDisk TF card, the Orange Pi test is to use a SanDisk TF card, other brands of TF cards may cause the system to fail to boot.

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

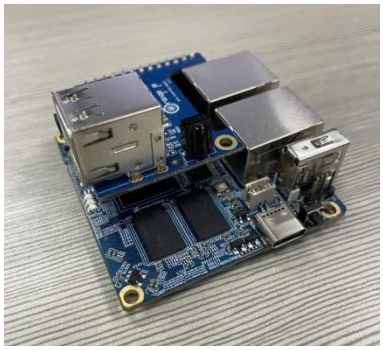
3) Power adapter, 5V/2A or 5V/3A power adapter with high-quality Type C interface

4) 13pin adapter board

a. The actual adapter board is shown below



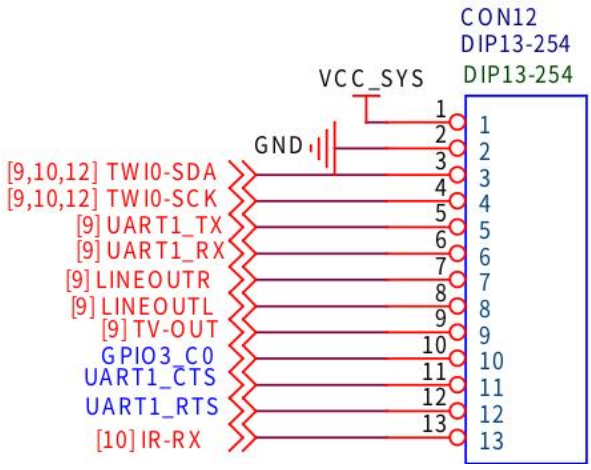
b. The way to insert the adapter board into the development board is as follows, remember not to insert it backwards



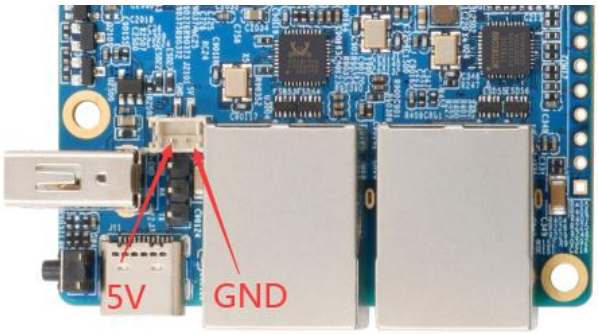
c. The 13pin pin header (unsoldered by default) on the Orange Pi R1 Plus LTS development board can be connected to an adapter board to expand the functions not on the development board. The functions included in the adapter board include:

1	microphone	**not support**
2	Analog audio and video output interface	Output analog audio and video signals via AV cable to TV (Android only)
3	USB2.0 x 2	**not support**
4	Infrared receiving function	

d. The schematic diagram of the 13pin pin header of the Orange Pi R1 Plus LTS is shown below



5) CPU fan, used for cooling the CPU**, the interface voltage is 5V, the interface specification is 2pin, and the spacing is 1.5mm**



6) Infrared remote control, mainly used to control the Android system

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

8) AV video cable, used to connect the development board to the TV through the CVBS interface to display the video (**only available for Android system**)

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

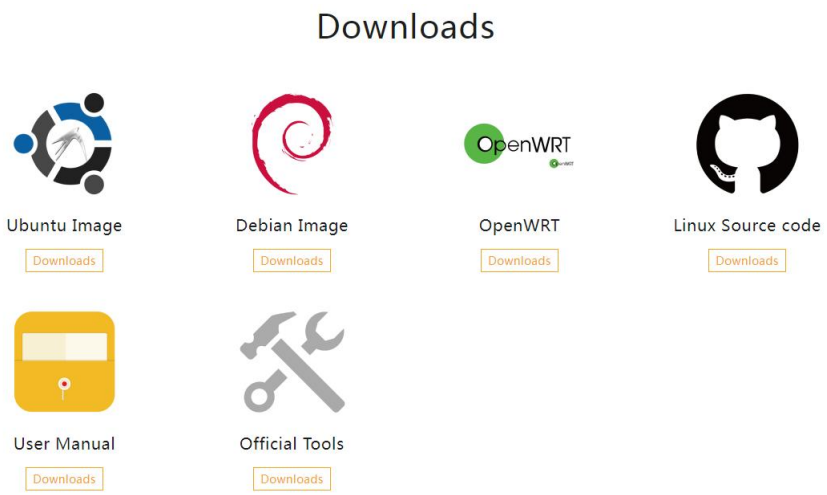
10) A PC with Ubuntu and Windows operating systems installed

1	Ubuntu18.04 PC	Optional, for compiling Linux and OpenWRT source code
2	Windows PC	For burning Linux and OpenWRT images

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

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

http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/service-and-support/Orange-Pi-R1-Plus-LTS.html



2) The data mainly includes

a. **Linux source code**: saved on github, the link address is

```
https://github.com/orangepi-xunlong/orangepi-build
```

b. **OpenWRT source code**: saved on github, the link address is

```
https://github.com/orangepi-xunlong/openwrt
```

c. **User manual and schematic diagram**: The data sheet related to the chip will also be placed here

d. **Official tools**: mainly include the software that needs to be used during the use of the development board

e. **Ubuntu image**: save on Google Drive

f. **Debian image**: saved on Google Drive

g. **OpenWRT image**: save on Google Drive

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

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

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

1) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the TF card must be above **class 10**. It is recommended to use a TF card from a brand such as SanDisk

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

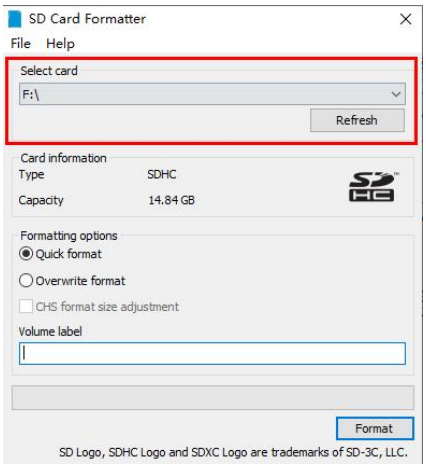
3) Then format the TF card

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

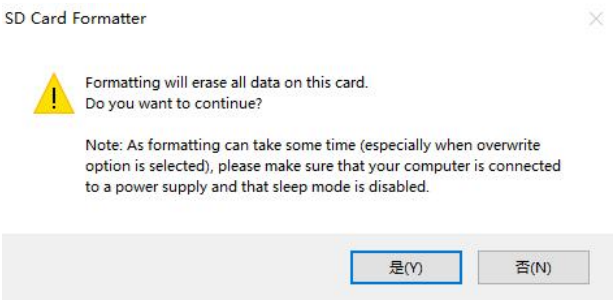
https://www.sdcard.org/downloads/formatter/eula_windows/SDCardFormatterv5_WinEN.zip

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

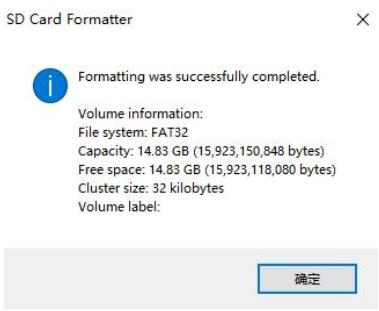
c. If only the TF card is inserted into the computer, the "**Select card**" column will display the drive letter of the TF card. If multiple USB storage devices are inserted into the computer, you can select the drive letter corresponding to the TF card through the drop-down box.



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



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



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

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

a. The download page of Win32Diskimager is

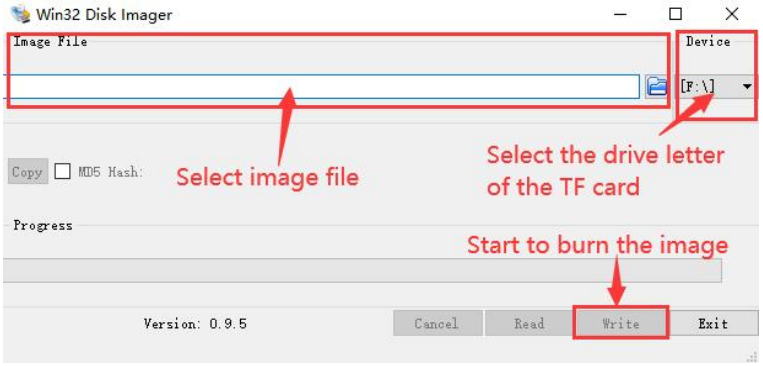
http://sourceforge.net/projects/win32diskimager/files/Archive/

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

a) First select the path of the image file

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

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



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

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

1) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the

TF card must be above **class 10**. It is recommended to use a TF card from a brand such as SanDisk

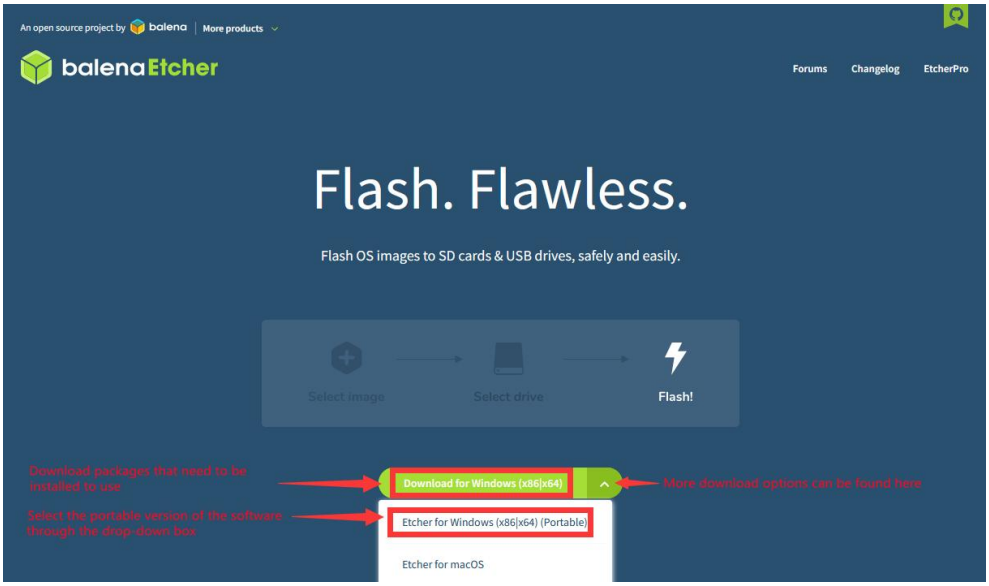
2) Then use the card reader to insert the TF card into the computer

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

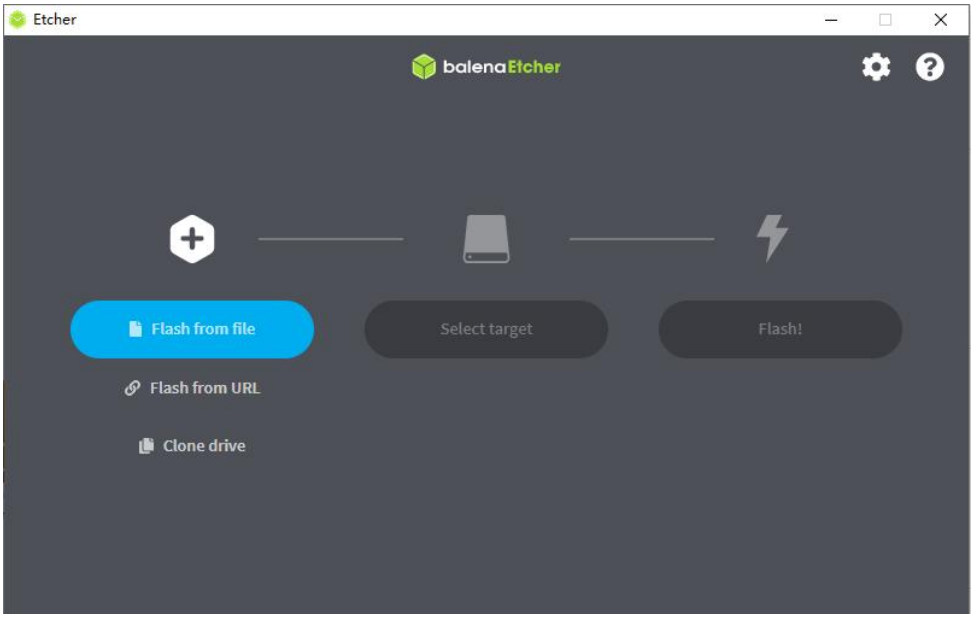
4) Then download the burning software of the Linux image - **balenaEtcher**, the download address is

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

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



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

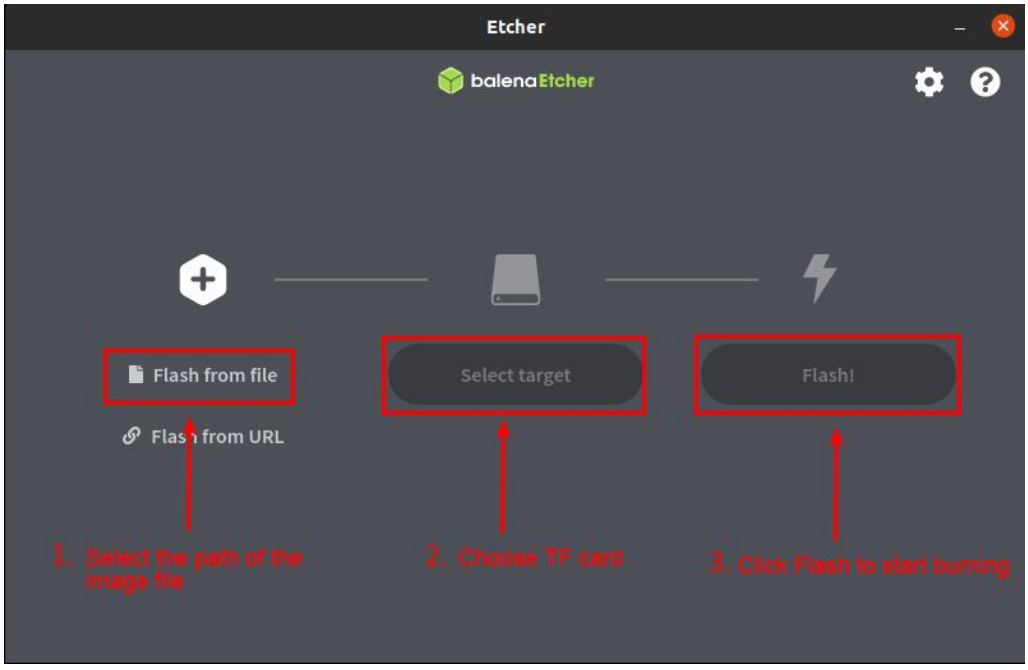


7) The specific steps to use balenaEtcher to burn a Linux image are as follows

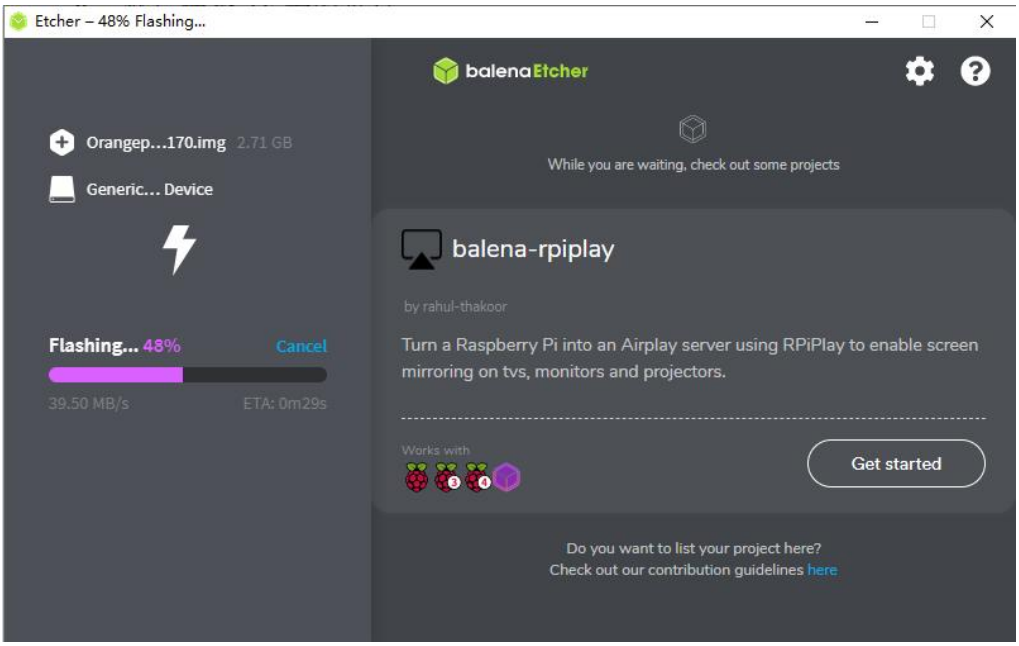
a. First select the path of the Linux image file to be burned

b. Then select the drive letter of the TF card

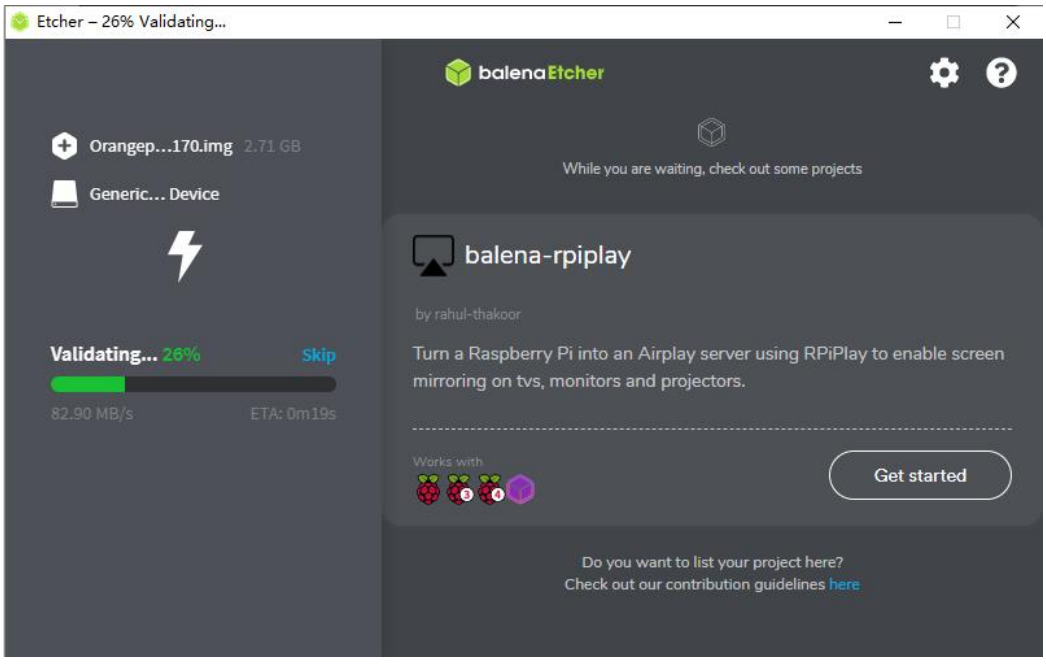
c. Finally, click Flash to start burning the Linux image to the TF card



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

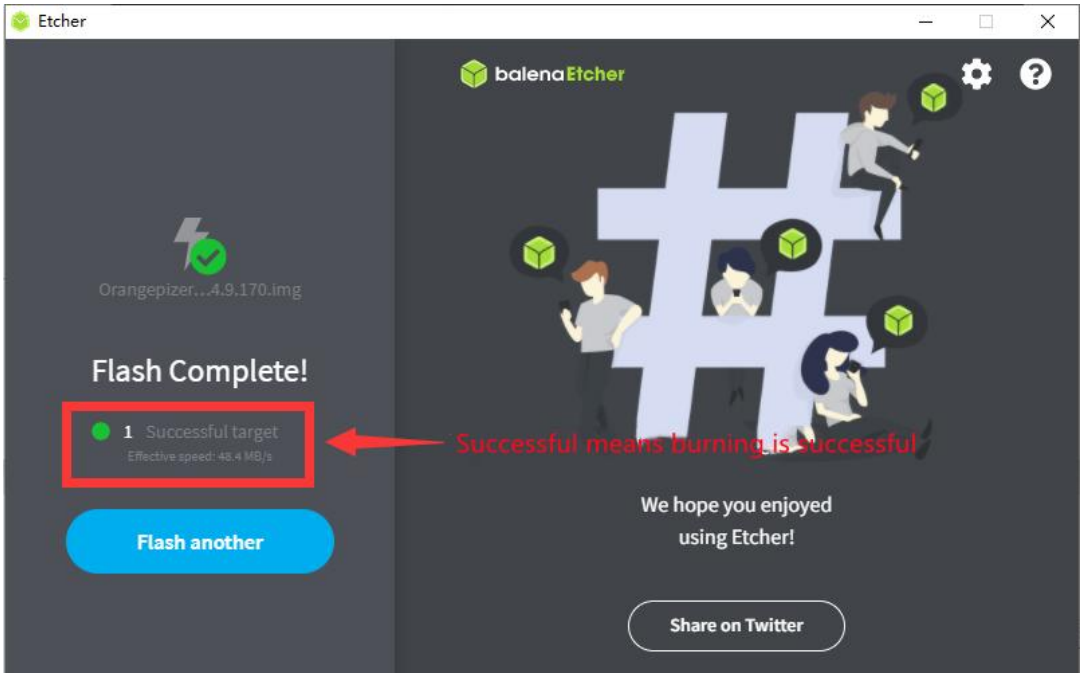


9) After the Linux image is burned, balenaEtcher will also verify the image burned to the TF card by default to ensure that there is no problem in the burning process. As shown in the figure below, a green progress bar indicates that the image has been burned, and balenaEtcher is verifying the burned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



10) After the successful burning, the display interface of balenaEtcher is shown in the figure below. If the green indicator icon is displayed, it means that the image burning is

successful. At this time, you can exit balenaEtcher, and then pull out the TF card and insert it into the TF card slot of the development board.



2. 4. The method of burning Linux image or OpenWRT image to TF card based on Ubuntu PC

11) The method of burning OpenWRT image based on Ubuntu PC is the same as the method of burning Linux image. The following is an example of burning Linux image.

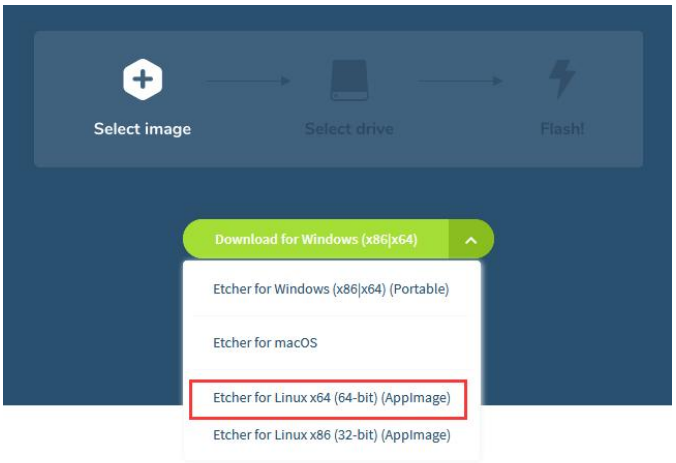
12) First prepare a TF card with a capacity of 8GB or more. The transmission speed of the TF card must be above **class 10**. It is recommended to use a TF card from a brand such as SanDisk

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

14) Download balenaEtcher software, the download address is

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

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



16) After downloading, use **unzip** to decompress it. The decompressed **balenaEtcher-1.5.109-x64.AppImage** is the software needed to burn the Linux image.

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

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

a. The decompression command for the compressed package ending with a.7z is as follows

```
test@test:~$ 7z x Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44.7z
test@test:~$ ls Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44.*
Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44.7z
Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44.img.sha  #checksum file
Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44.img      #image file
```

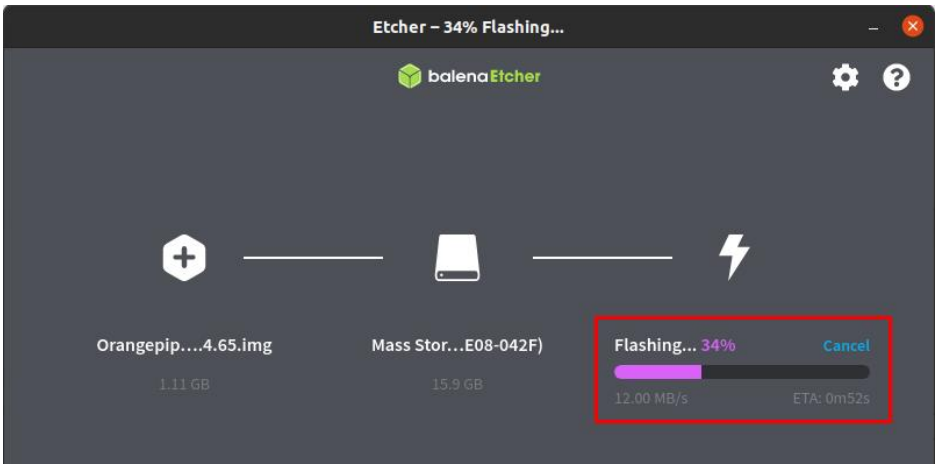
18) After decompressing the image, you can use the sha256sum -c *.sha command to calculate whether the checksum is correct. If the message is **successful**, it means that the downloaded image is correct. You can safely burn it to the TF card. If the **checksum does not match**, it means that There is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

```
test@test:~$ sha256sum -c *.sha
Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44.img:    success
```

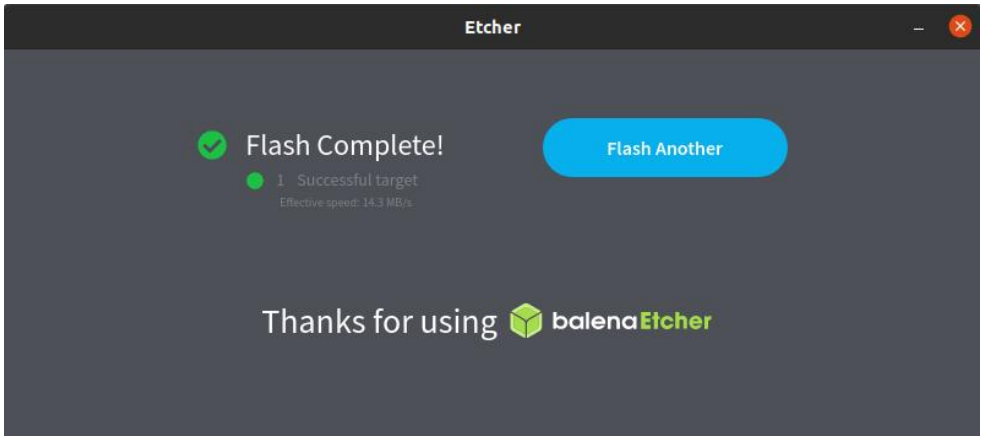
19) Then double-click balenaEtcher-1.5.109-x64.AppImage on the graphical interface of Ubuntu PC to open balenaEtcher (no installation required), and the opened interface is shown in the figure below

a. First select the path of the linux image file
b. Then select the device number of the TF card
c. Finally click Flash to start burning

20) The burning process will prompt the writing speed and remaining time



21) After burning, the following interface will be displayed. At this time, you can pull out the TF card from the computer and insert it into the development board to start.



2. 5. Start the orange pi development board

1) Insert the burned TF card into the TF card slot of the Orange Pi development board

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

3) Connect a 5V/2A (5V/3A can also) **high-quality** power adapter

   a. **Remember not to plug in the 12V power adapter, if the 12V power adapter is plugged in, it will burn out the development board**

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

4) If you want to view the output information of the system through the debug serial port, please use the USB to TTL module and DuPont cable to connect the development board to the computer. For the connection method of the serial port, please refer to the section on the use of the debug serial port.

5) Then turn on the switch of the power adapter. If everything is normal, the serial port terminal can see the output log of the system startup.

2. 6. How to use the debugging serial port

2. 6. 1. Connection instructions for debugging serial port

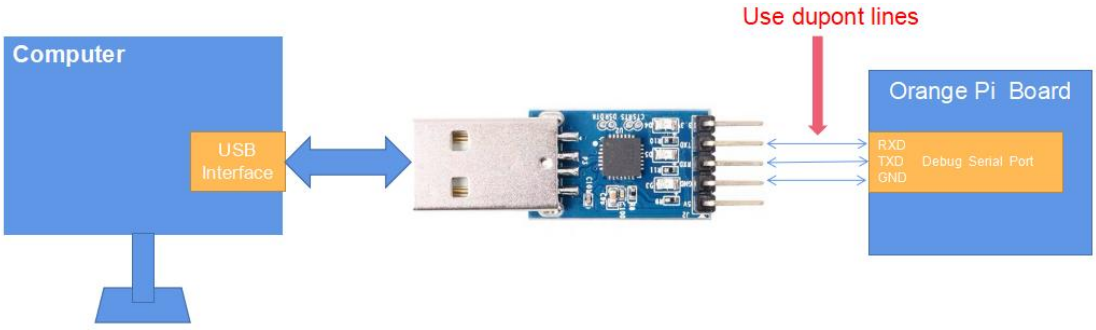
1) First, you need to prepare a USB to TTL module. For better platform compatibility, it is recommended to use the CH340 USB to TTL module. Then insert one end of the USB interface of the USB to TTL module into the USB interface of the computer

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

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

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

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



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

5) If the CP2102 USB to TTL module is used, in the case of a baud rate of 1500000,

some systems may encounter garbled or unusable problems. The specific test situation is as follows

USB to TTL module model	Host system	Support situation
CH340	win7	OK
	win10	OK
	ubuntu14.04	OK
	ubuntu18.04	OK
	ubuntu20.04	OK
CP2102	win7	OK
	win10	NO
	ubuntu14.04	OK
	ubuntu18.04	NO
	ubuntu20.04	NO

2.6.2. How to use the debugging serial port on Ubuntu platform

1) If the USB to TTL module is connected normally, you can see the corresponding device node name under /dev of the Ubuntu PC, remember this node name, it will be used when setting the serial port software later

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

2) 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

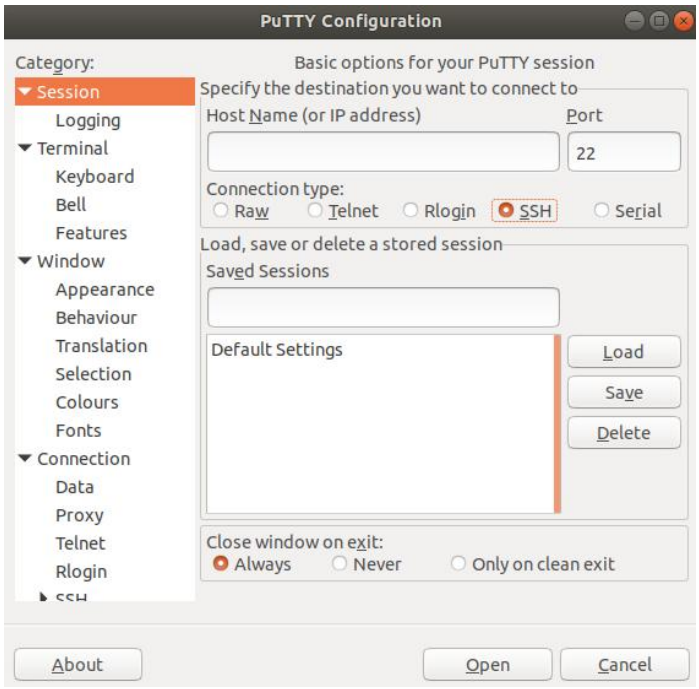
3) First install putty on Ubuntu PC

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

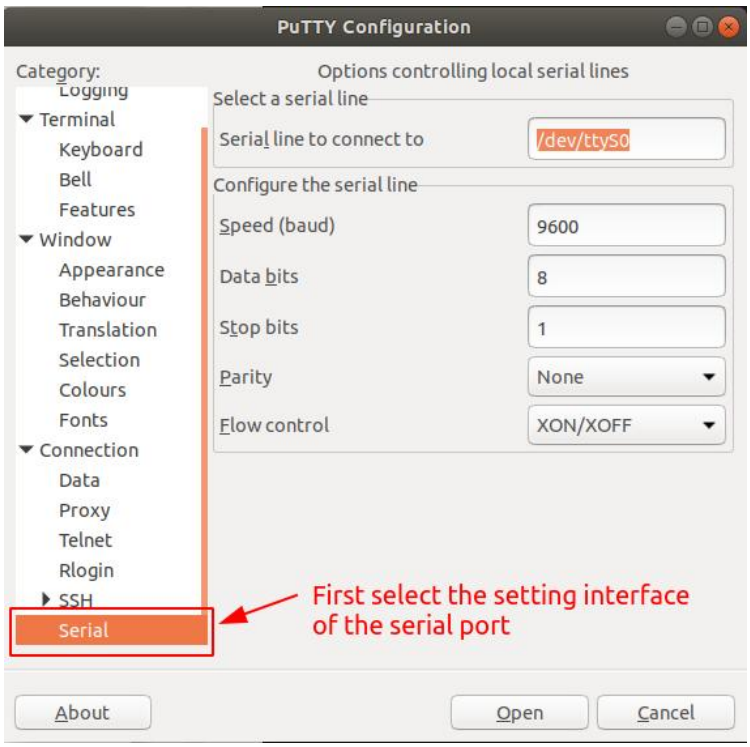
4) Then run putty, remember to add sudo permissions

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

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



6) First select the setting interface of the serial port



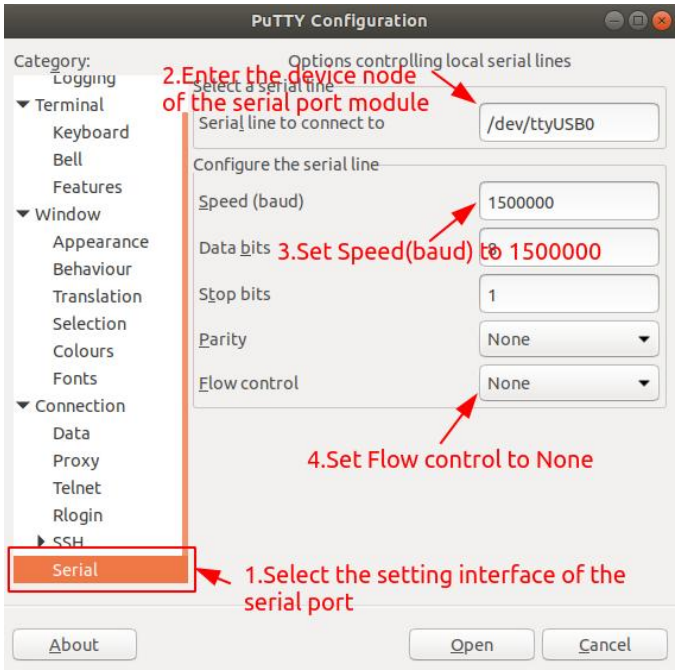
7) Then set the parameters of the serial port

a. Set Serial line to connect to to /dev/ttyUSB0 (modify to the corresponding node

name, usually /dev/ttyUSB0)

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

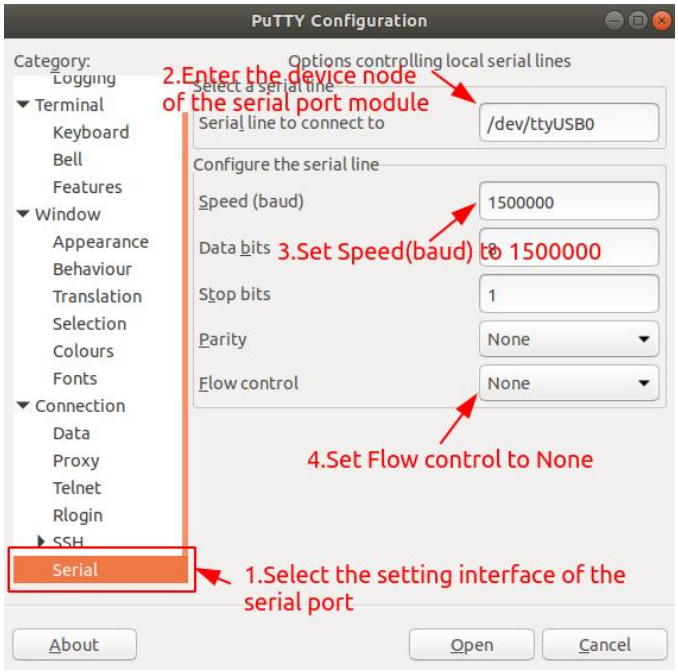
c. Set Flow control to None



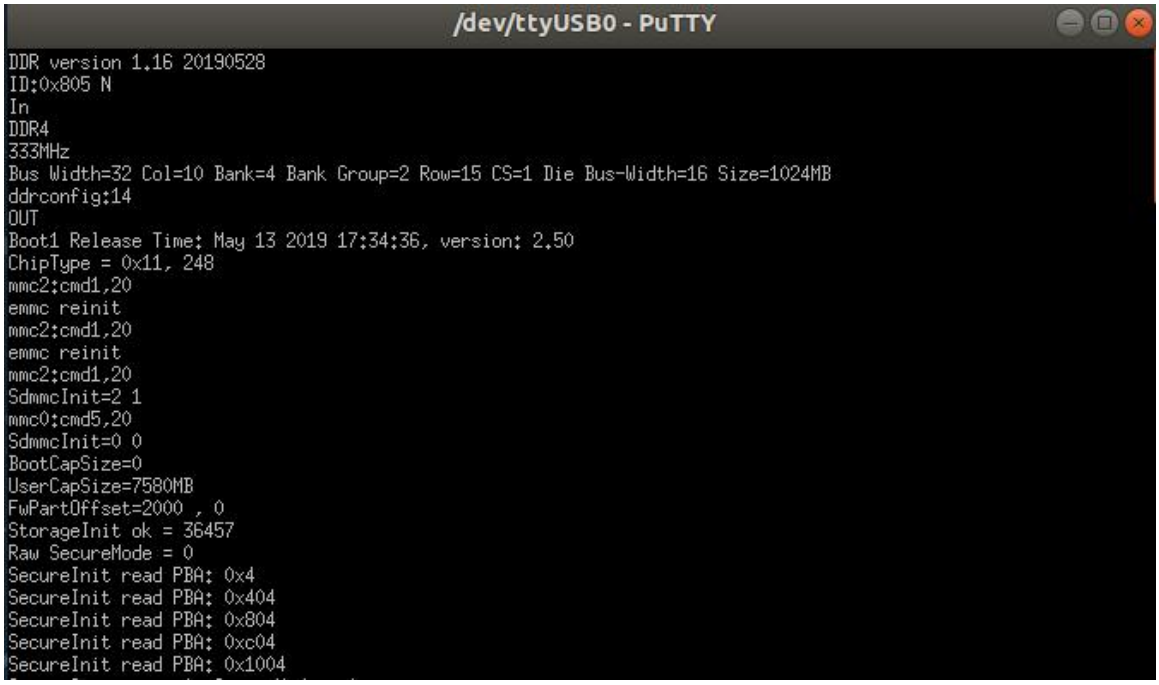
8) After setting the serial port setting interface, go back to the Session interface

a. First select the Connection type as Serial

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



9) After starting the development board, you can see the Log information output by the system from the open serial terminal



2. 6. 3. How to use the debugging serial port on Windows platform

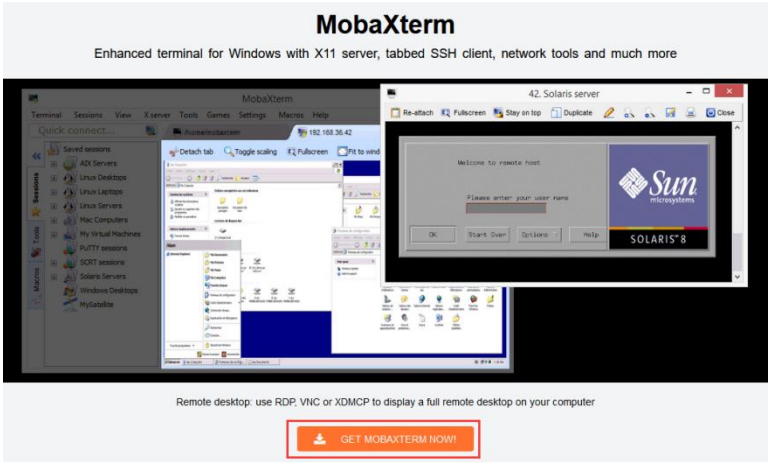
1) There are many serial debugging software that can be used under Windows, such as SecureCRT, MobaXterm, etc. The following demonstrates how to use MobaXterm. This software has a free version and can be used without purchasing a serial number.

2) Download MobaXterm

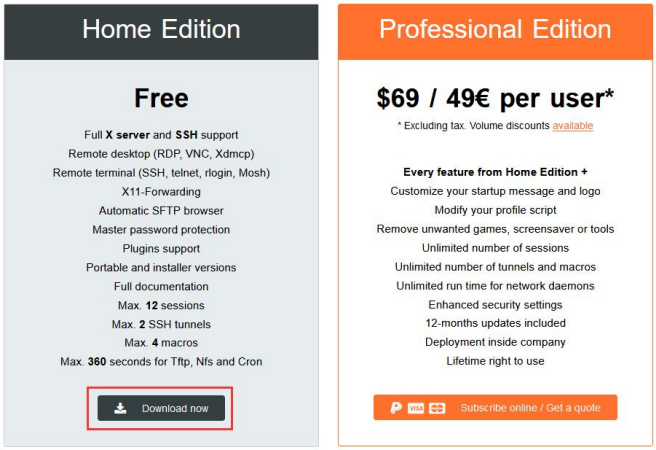
a. Download MobaXterm URL as follows

https://mobaxterm.mobatek.net/

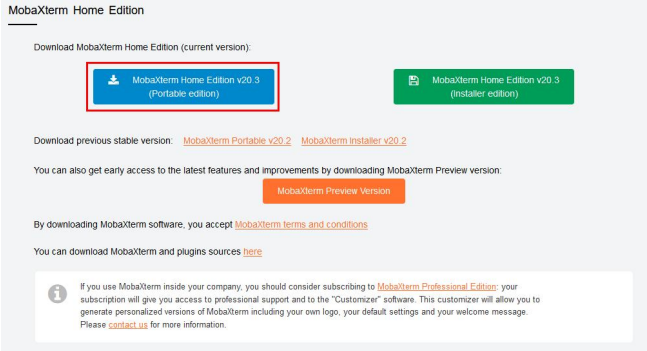
b. After entering the MobaXterm download page, click **GET XOBATERM NOW!**



c. Then choose to download the Home version



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



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

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

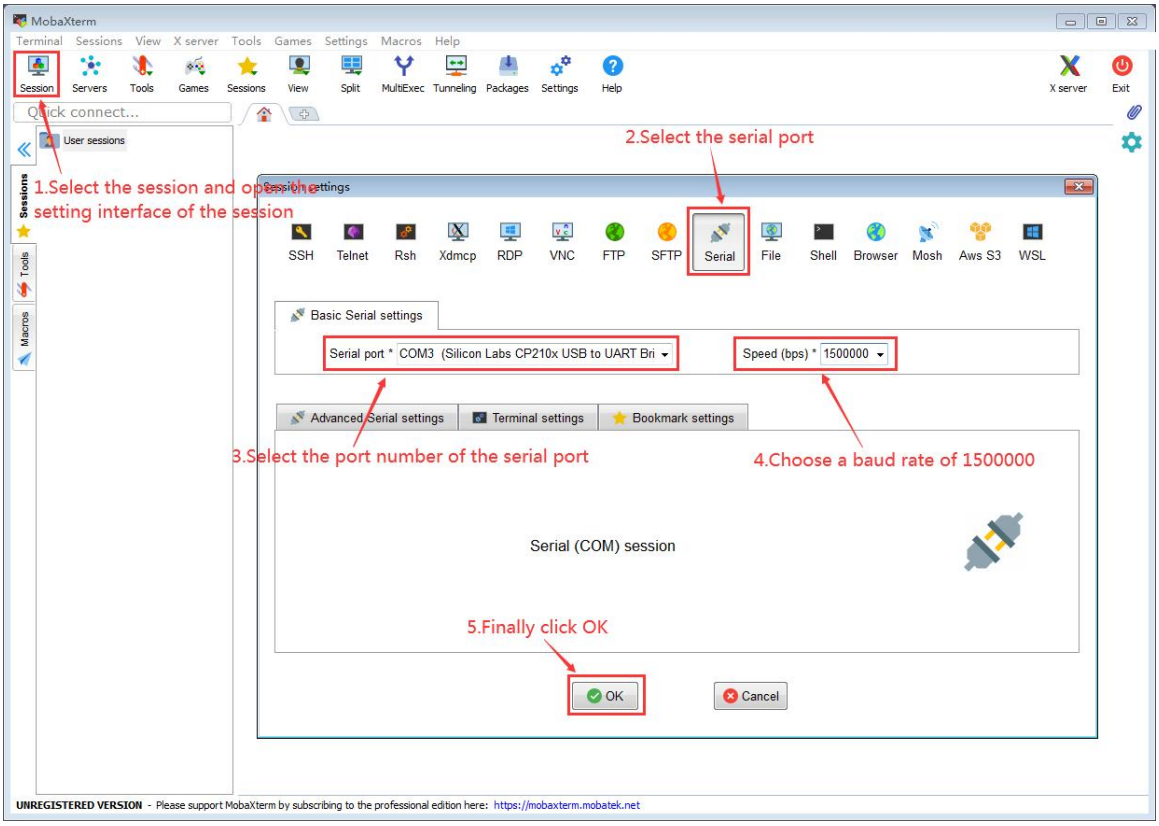
a. Open the session settings interface

b. Select the serial port type

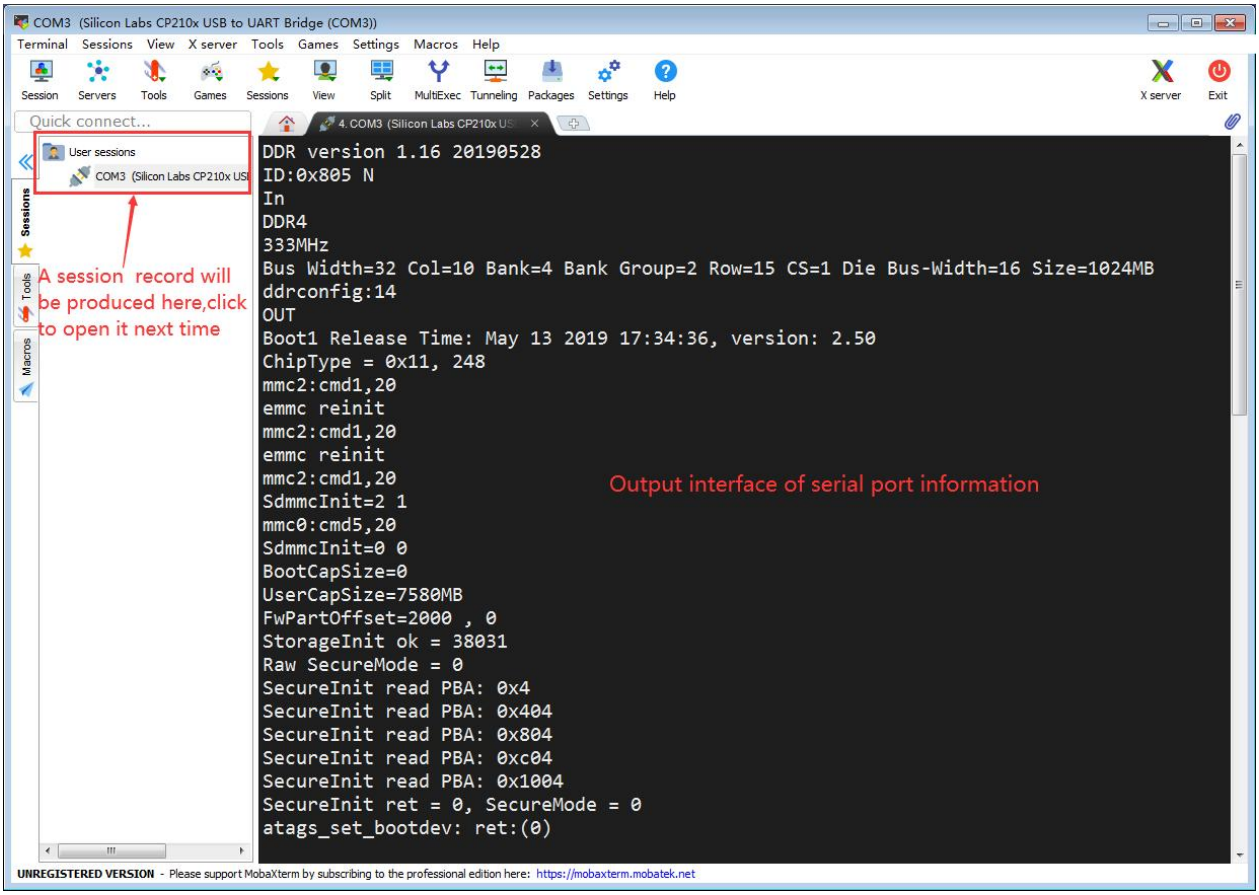
c. Select the port number of the serial port (select the corresponding port number according to the actual situation). If you cannot see the port number, please use the **360 driver master** to scan and install the driver for the USB to TTL serial port chip.

d. Select the baud rate of the serial port to be 1500000

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



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



3. Instructions for using the OpenWRT system

3. 1. OpenWRT version

1) There are two branches of openwrt code on Github

OpenWRT version	kernel version
OpenWRT openwrt-21.02 branch	Linux5.4
OpenWRT master branch	Linux5.10

2) openwrt-21.02 is developed based on OpenWrt v21.02.1 Release, and its functions tend to be stable

3) The master branch is based on the snapshot version of openwrt, which is an unstable and developing version

3. 2. OpenWRT system default login account and password

It is recommended to change a relatively secure password for web login and ssh login before use.

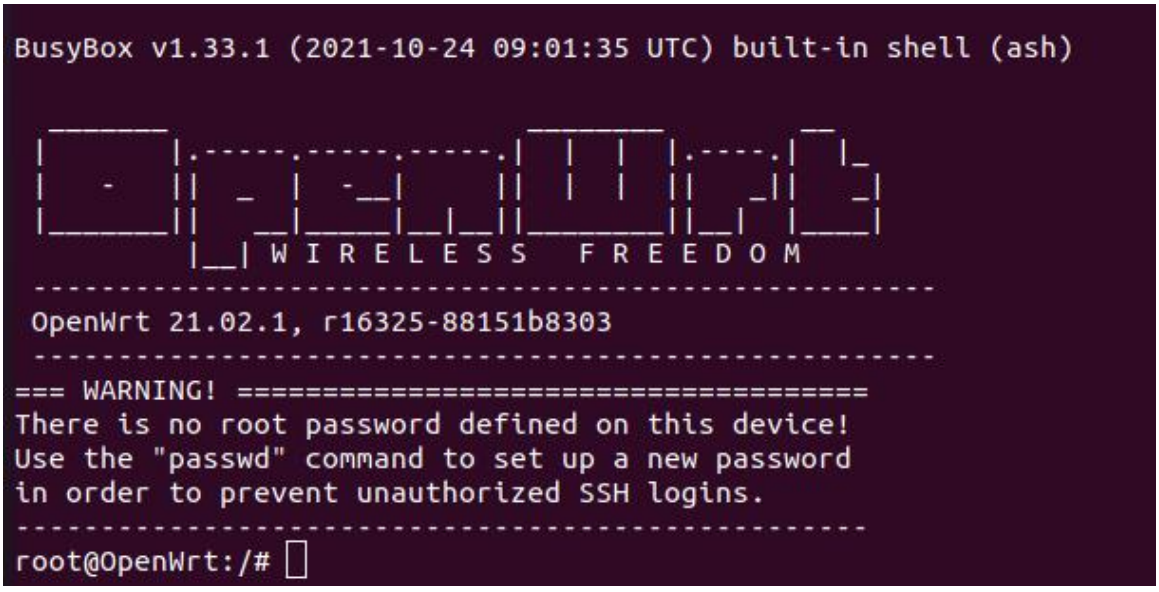
account	password
root	Default password is blank

3. 3. How to log in to the system

3. 3. 1. Serial login

1) First of all, you can refer to the chapter on **how to use the serial port to debug the use of the serial port**.

2) The OpenWrt system will automatically log in as the root user by default, and the display interface is as follows



3. 3. 2. ssh login system

The OpenWrt system enables ssh remote login by default, and allows the root user to log in to the system. Before ssh login, you need to ensure that the Ethernet is connected, and then use the sudo ifconfig command or obtain the IP address of the

development board by viewing the router

1) Log in via ssh through the WAN port
   a. First connect the WAN port of the board to the LAN port of the main router with a network cable, so that the WAN port can obtain an IP address through DHCP
   b. Then connect the WAN port of the computer to the main router with a network cable. At this time, the WAN port of the computer and the WAN port of the board are in the same network segment
   c. It is assumed here that the IP address of the WAN port of the board has been determined to be 192.168.1.130 through the debugging serial port or the management interface of the main router.
   d. If the Ubuntu system is installed on the computer, you can execute the following command to log in to the system through SSH

```
csy@ubuntu:~$ ssh root@192.168.1.130
```

   e. The display after successfully logging in to the system is as shown below

```
csy@ubuntu:~$ ssh root@192.168.1.130
The authenticity of host '192.168.1.130 (192.168.1.130)' can't be established.
ED25519 key fingerprint is SHA256:DvPAssmhqUppxwkQWPTBq+g/POdMJGldfu3hG1edIYs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.130' (ED25519) to the list of known hosts.


BusyBox v1.33.1 (2021-10-24 09:01:35 UTC) built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt 21.02.1, r16325-88151b8303
 -----------------------------------------------------
=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@OpenWrt:~#
```

   f. If the computer is installed with the Windows system, you can log in by referring to **the method of SSH remote login to the development board under Windows** in the Linux system instruction manual.

2) Log in via ssh over the LAN port

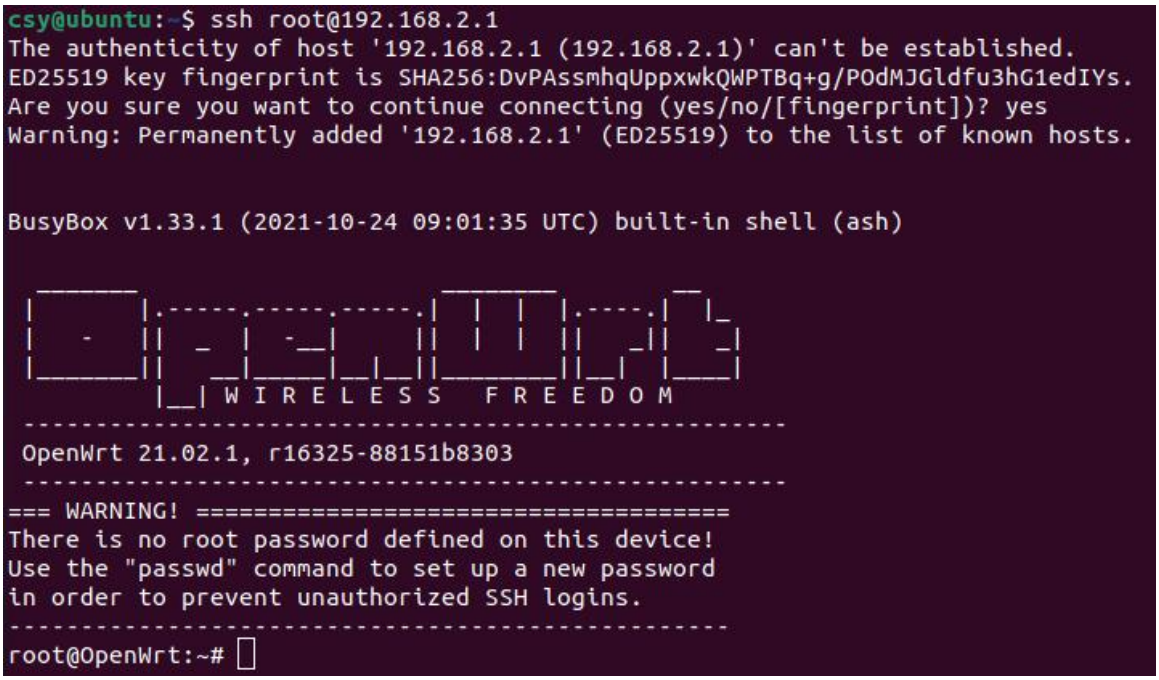
a. First connect the LAN port of the board to the WAN port of the computer with a network cable, so that the WAN port of the computer can obtain an IP address through DHCP, and the obtained IP address is **192.168.2.X**

b. The IP of the LAN port of the board is set to **192.168.2.1** by default, and you can log in through the IP address **192.168.2.1** at this time.

c. If the Ubuntu system is installed on the computer, you can execute the following command to log in to the system through SSH

```
csy@ubuntu:~$ ssh root@192.168.2.1
```

g. The display after successfully logging in to the system is as shown below

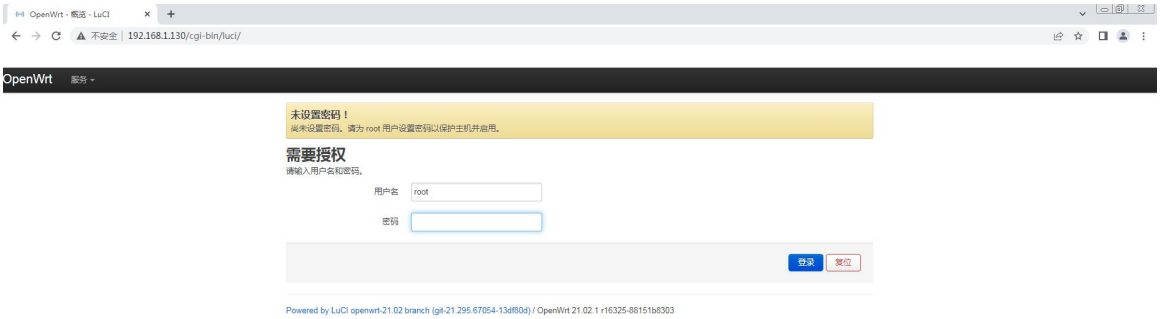


d. If the Windows system is installed on the computer, you can log in by referring to the **method of SSH remote login to the development board under Windows** in the Linux system instruction manual.

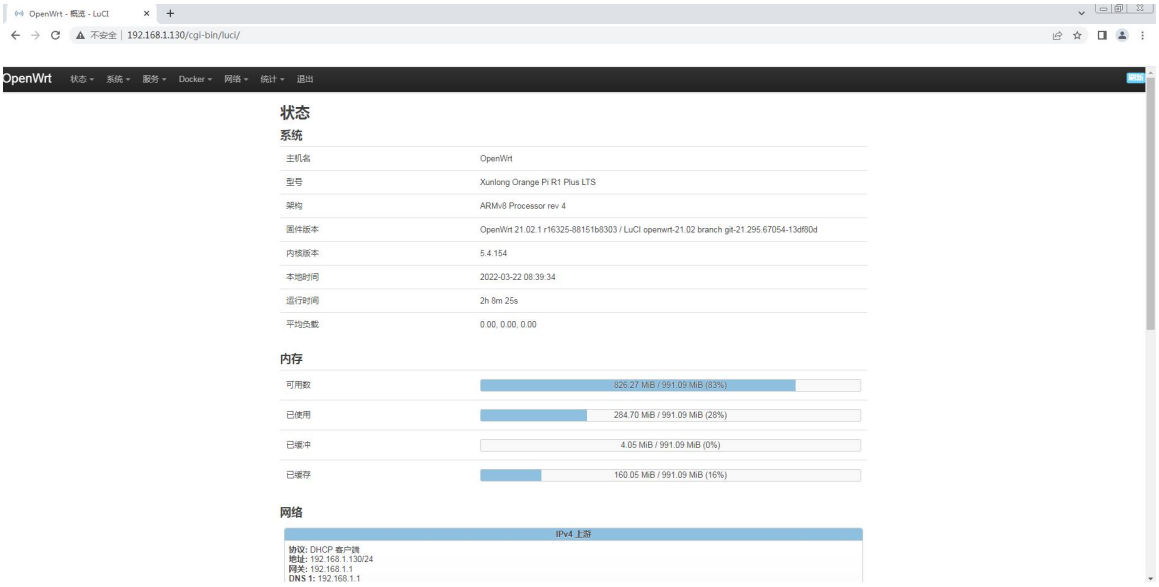
3. 3. 3. Log in to the LuCI management interface

1) Log in to the LuCI management interface through the WAN port

a. On the basis of completing the **ssh login of the WAN port**, assuming that the IP address of the WAN port of the board is 192.168.1.130, enter the IP address in the browser of the computer to log in to the LuCI interface



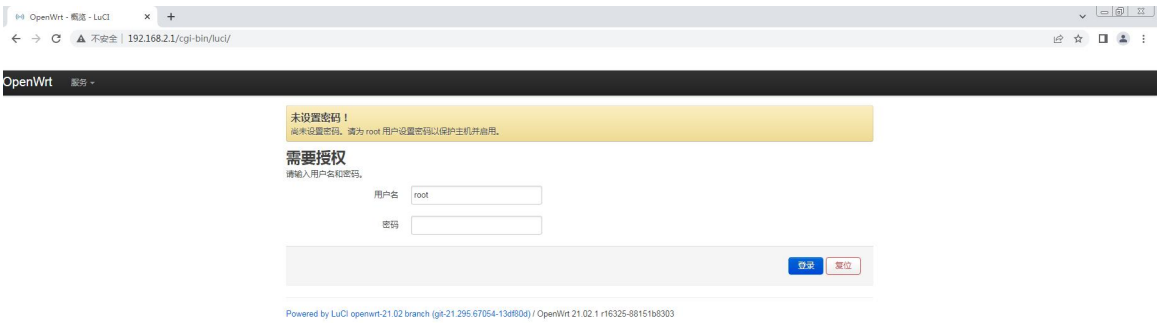
b. Because there is no password set by default, click the **login** button directly. After the login is successful, the interface will be displayed as follows



c. If a password is set, you also need to enter the password and click Login

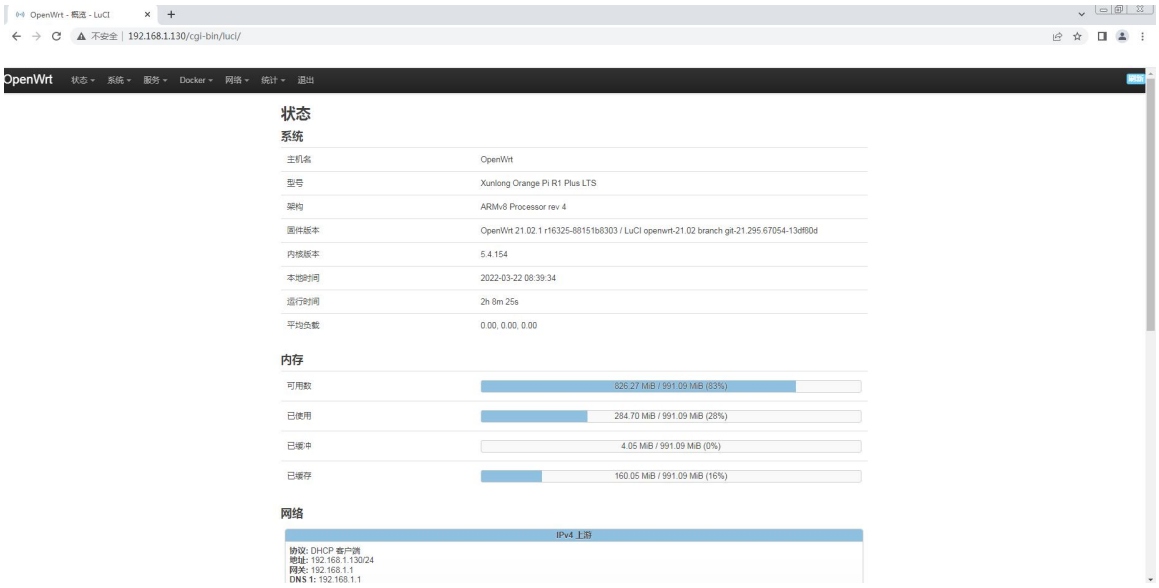
2) Log in to the LuCI management interface through the LAN port

a. After completing the **LAN port ssh login**, enter the IP address **192.168.2.1** in the browser to log in to the LuCI interface



b. Because there is no password set by default, click the **login button** directly.

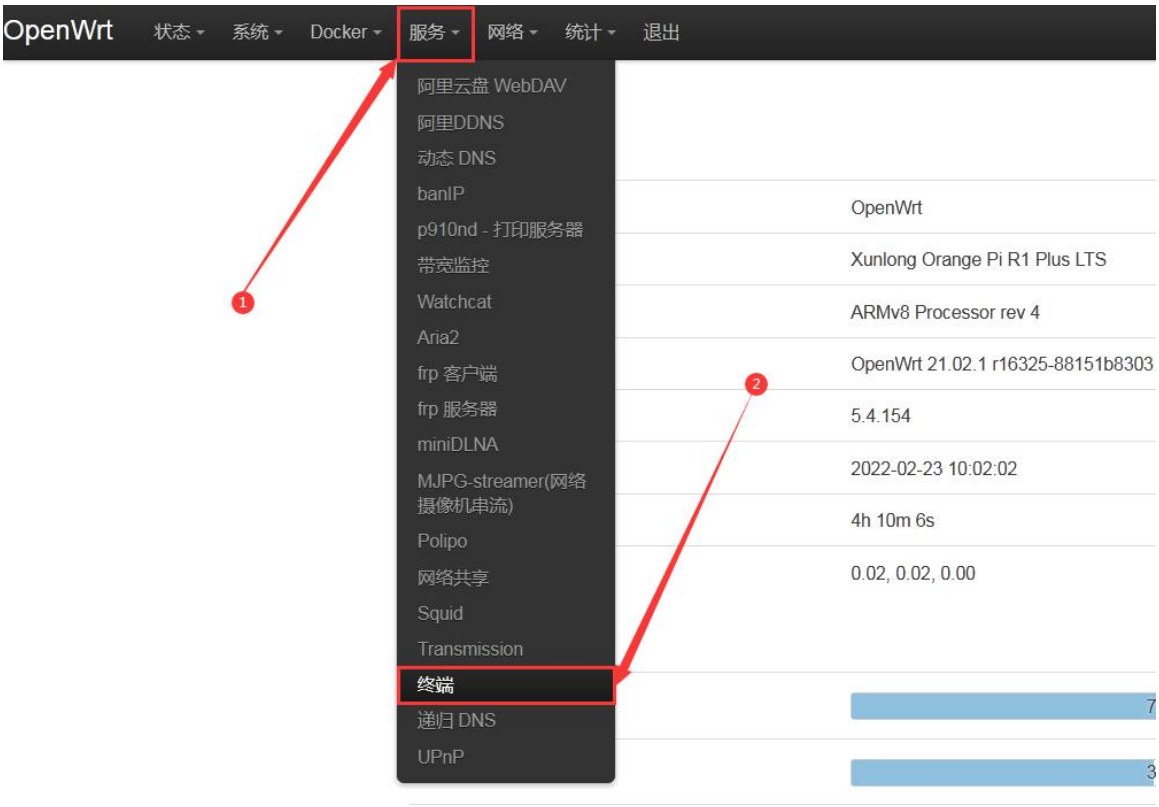
After the login is successful, the interface will be displayed as follows



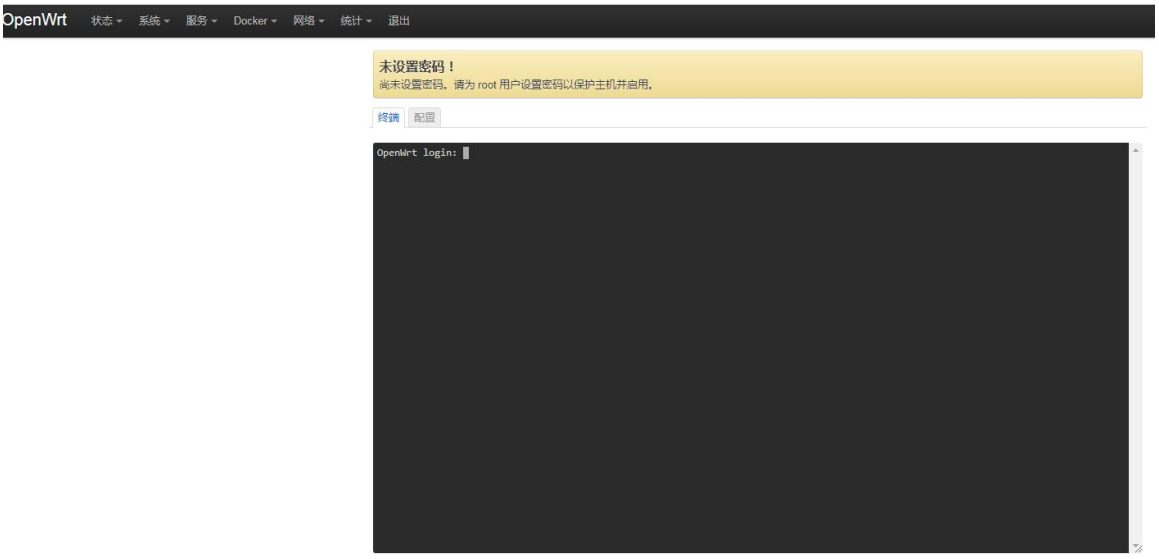
d. If a password is set, you need to enter the password and then click Login

3. 3. 4. Log in to the terminal through the LuCI management interface

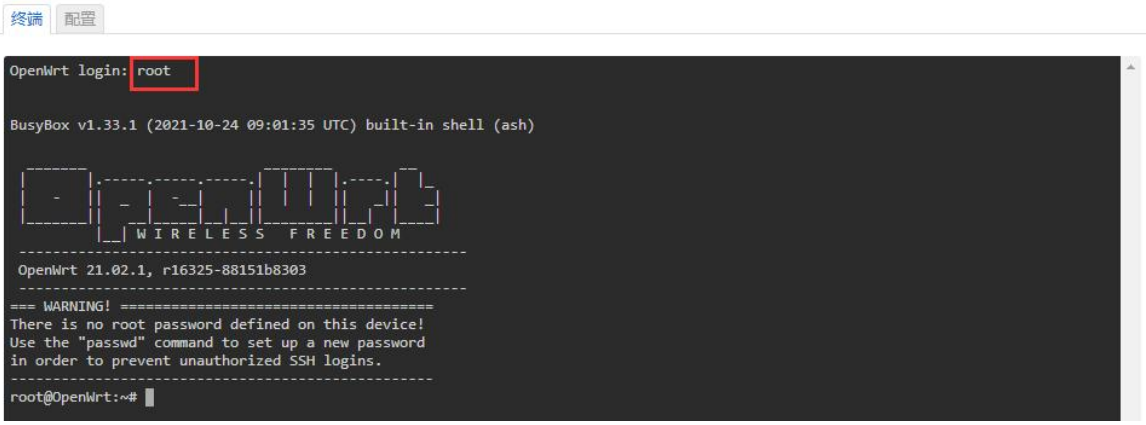
1) Select "**Terminal**" in the "**Services**" column of the navigation bar and click to enter



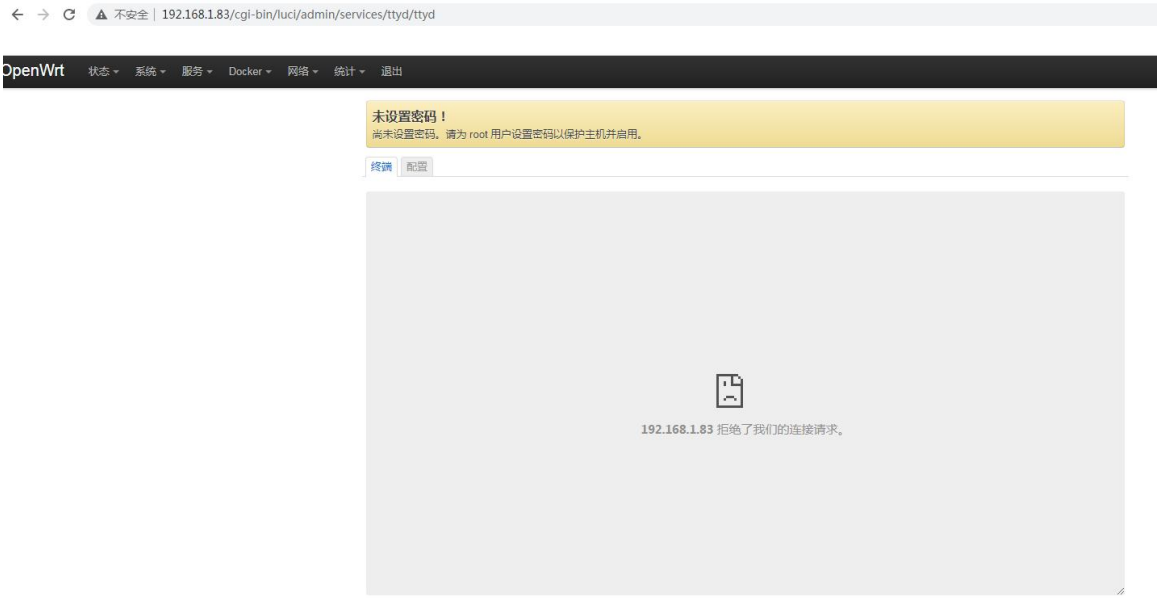
2) At this time, the terminal interface is as shown in the figure below



3) Enter the username root to log in



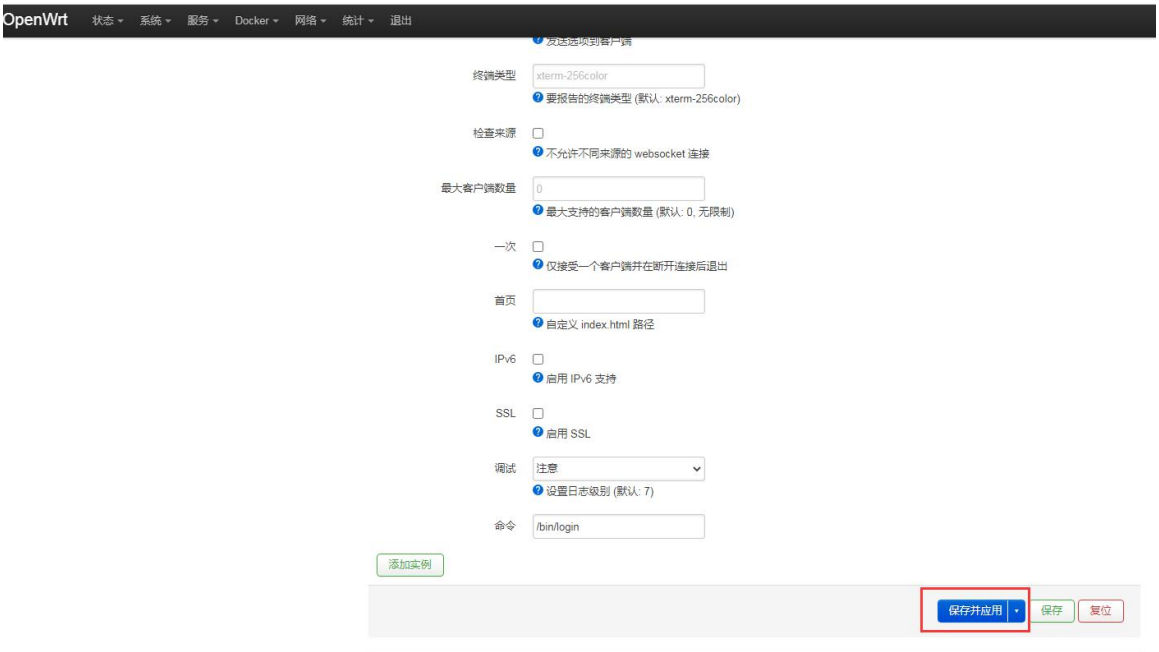
If you **log in to the LuCI management interface through the WAN port**, you will encounter the following problems, because ttyd only monitors the LAN port by default, and access from the WAN port is invalid



The solution is to modify the interface monitored by ttyd to the WAN port

a. In the configuration column, modify the interface to WAN port

b. Pull the page to the bottom, click **Save and Apply**

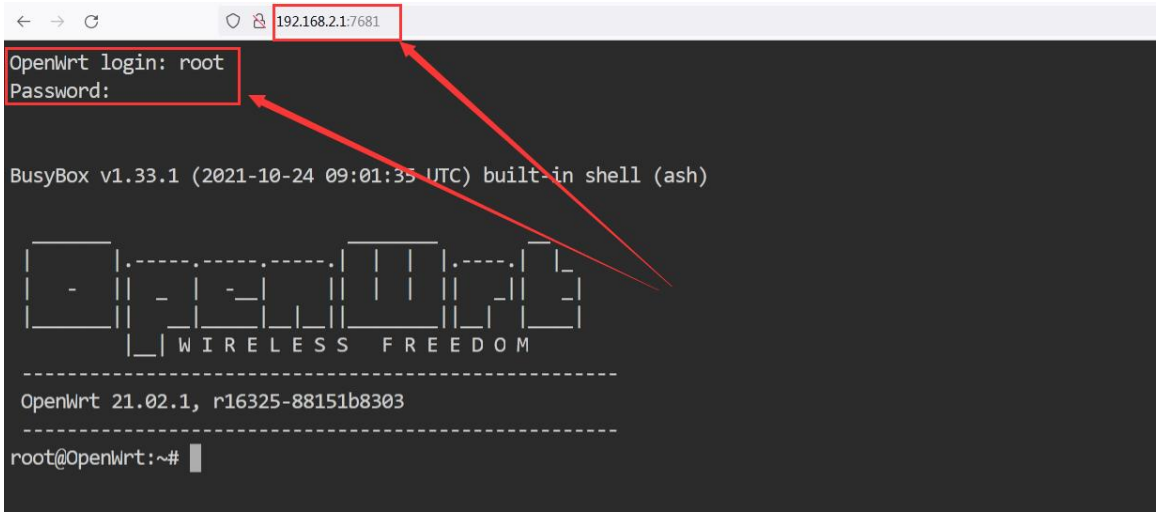


3. 3. 5. Use the IP address + port number to log in to the terminal

1) Log in through the LAN port

a. The default listening port number of OpenWRT system ttyd is 7681

b. After **logging in to the LuCI management interface through the LAN port**, enter **192.168.2.1:7681** in the browser to log in to the OpenWRT terminal



3. 4. How to modify the IP address of the LAN port

The default IP address of the LAN port of the OpenWrt system of the development board is **192.168.2.1**. If the IP address of the main router connected to the WAN port of the development board or the LAN port of the optical modem is also the same, it may be impossible to access the Internet due to IP address conflict. At this time, you need to set a non-conflicting IP address for the LAN port of the development board

1) In the OpenWrt system, a command line tool uci is provided, which can easily modify, add, delete and read the content in the configuration file. For details, please refer to the official documentation

2) First use the following command to get the network configuration, the corresponding configuration file is **/etc/config/network**, you can see that the value of **network.lan.ipaddr** is **192.168.2.1**

```
root@OpenWrt:~# uci show network
...
network.lan=interface
network.lan.device='br-lan'
network.lan.proto='static'
network.lan.ipaddr='192.168.2.1'
network.lan.netmask='255.255.255.0'
network.lan.ip6assign='60'
```

```
....
```

3) Then enter the following command to modify the item **network.lan.ipaddr**

```
root@OpenWrt:~# uci set network.lan.ipaddr='192.168.100.1'
```

4) Then enter the following command to complete the submission, that is, write to the configuration file

```
root@OpenWrt:~# uci commit
```

If the IP address in the red font is the same as the one to be set, the modification is successful.

```
root@OpenWrt:~# cat /etc/config/network
...
config interface 'lan'
        option device 'br-lan'
        option proto 'static'
        option netmask '255.255.255.0'
        option ip6assign '60'
        option ipaddr '192.168.100.1'
...
```

5) Restart the network through ubus, please refer to the **official documentation** for the usage of ubus

```
root@OpenWrt:~# ubus call network restart
```

6) At this point, enter the command to see that the IP of the LAN port is already **192.168.100.1**

```
root@OpenWrt:~# ifconfig br-lan
br-lan    Link encap:Ethernet  HWaddr FE:55:13:A3:EF:E7
          inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: fd60:c4cd:1033::1/60 Scope:Global
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

```
RX bytes:0 (0.0 B)  TX bytes:370 (370.0 B)
```

3. 5. How to modify the root password

3. 5. 1. Modification via command line

7) First, enter passwd root on the command line of the system, and the following prompt message will appear. At this time, you can enter the password you want to set, and press the Enter key to confirm

```
root@OpenWrt:/# passwd root
Enter new UNIX password:
```

8) Then you will be prompted to re-enter the password. At this time, enter the password again to confirm and press Enter.

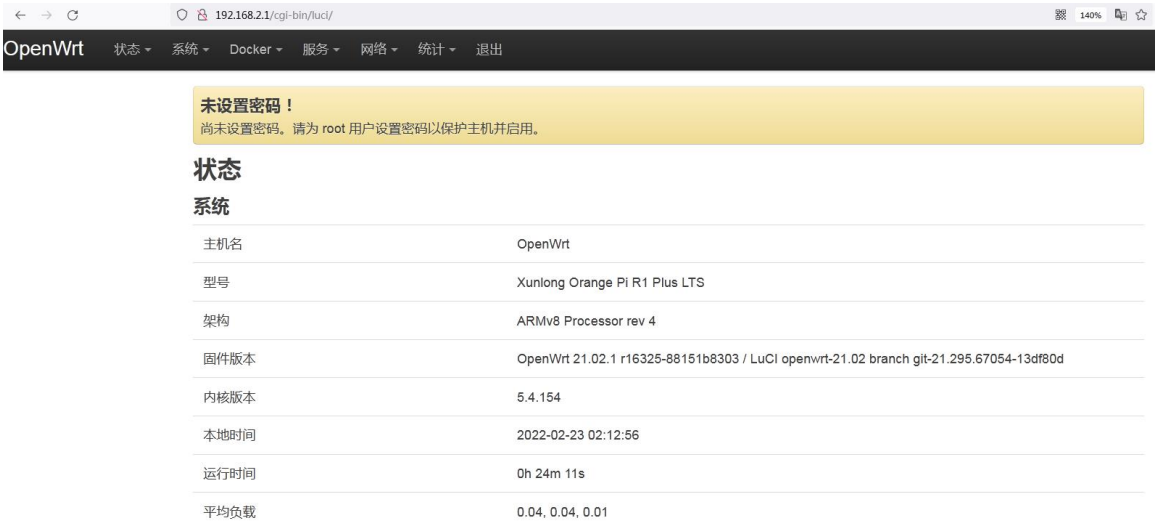
```
Retype password:
```

9) The display of successful modification is as follows

```
passwd: password for root changed by root
```

3. 5. 2. Modification through the LuCI management interface

1) First, refer to **logging in to the LuCI management interface** to enter the OpenWRT management interface



2) Enter the OpenWRT password change interface

a. Find the "**System**" option in the navigation bar and click to enter

b. In the vertical bar options below the system, select "**Administrative Rights**" and click to enter

c. Select the "**Router Password**" option on the Tab page and click to enter

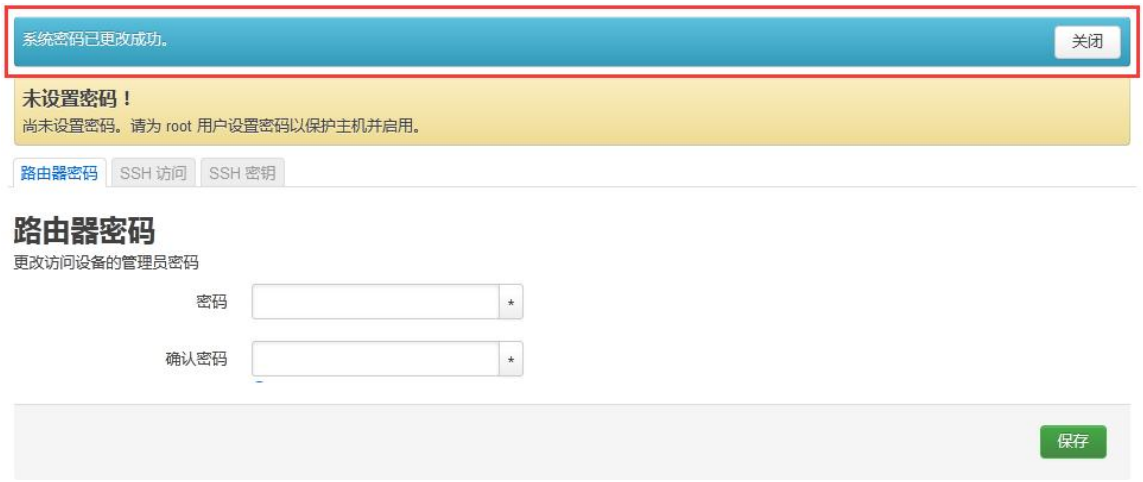
3) Modify and save the router password

a. Enter the password you set in the "**Password**" and "**Confirm Password**" dialog boxes (if you are not sure whether the password is entered correctly, you can click the "*" icon at the back of the dialog box to display the input characters)

b. Click "**Save**" to save the newly modified password

Note: In the "Password" and "Confirm Password" dialog boxes, the passwords entered twice need to be the same.

4) After the password is successfully changed, a pop-up box of "**System password has been changed successfully**" will pop up. At this time, you need a password to log in to OpenWRT.



3. 6. Expand the rootfs in the TF card before the first boot

1) When the TF card starts the OpenWRT system for the first time, it will automatically expand the rootfs by calling the resize-rootfs.sh script, so there is **no need to manually expand the capacity**

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

```
root@OpenWrt:~# df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root                14.8G     14.7G     91.6M  99% /
tmpfs                   495.5M      6.1M    489.4M   1% /tmp
tmpfs                   512.0K         0    512.0K   0% /dev
/dev/root                14.8G     14.7G     91.6M  99% /opt/docker
```

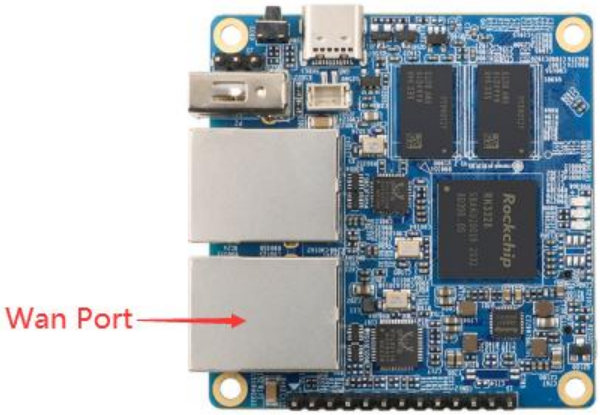
After the system is started for the first time, it will be found that the free space is

insufficient. Just restart the system, and the free space will return to normal after the system is restarted.

3. 7. Ethernet port test

3. 7. 1. Wan port test

1) First connect the WAN port of the R1 Plus LTS to your main router with a network cable, and make sure the network is unblocked



2) After the system starts, it will automatically obtain an IP address from the main router through DHCP

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

```
root@OpenWrt:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr C0:74:2B:FF:B3:41
          inet addr:192.168.1.87  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::c274:2bff:feff:b341/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1634 errors:0 dropped:84 overruns:0 frame:0
          TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:142098 (138.7 KiB)  TX bytes:13503 (13.1 KiB)
          Interrupt:2
```

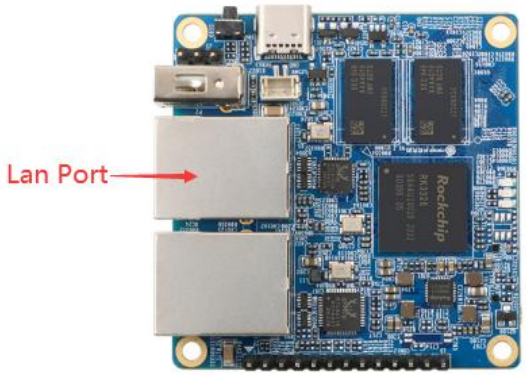
4) The command to test network connectivity is as follows

```
root@OpenWrt:/# ping www.baidu.com -I eth0
PING www.baidu.com (14.215.177.38): 56 data bytes
64 bytes from 14.215.177.38: seq=0 ttl=56 time=6.169 ms
64 bytes from 14.215.177.38: seq=1 ttl=56 time=5.473 ms
64 bytes from 14.215.177.38: seq=2 ttl=56 time=5.114 ms
64 bytes from 14.215.177.38: seq=3 ttl=56 time=5.992 ms
^C
--- www.baidu.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 5.114/5.693/6.169 ms
```

5) If there is no usb to ttl serial port, you need to know the IP address of the R1 Plus LTS, or you can view it in the background management interface of the main router

3. 7. 2. Lan port test

1) First, connect the network cable to the Lan port of the development board and the Ubuntu PC



2) After the system starts, it will automatically assign an IP address to the Ubuntu PC Ethernet card through DHCP

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

```
test@ubuntu:~# ifconfig enp5s0
eenp5s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.174  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fd50:32f6:413:0:8050:834c:16d4:d493 prefixlen 64 scopeid 0x0<global>
        inet6 fe80::686:bb73:c7ae:9e9a  prefixlen 64  scopeid 0x20<link>
        inet6 fd50:32f6:413::d00  prefixlen 128  scopeid 0x0<global>
```

```
        inet6 fd50:32f6:413:0:6bbf:bc59:16e2:1f76 prefixlen 64 scopeid 0x0<global>
        ether 40:b0:76:60:17:c3  txqueuelen 1000  (Ethernet)
        RX packets 4331701  bytes 2941416494 (2.9 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13649801  bytes 5762726379 (5.7 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

4) The command to test network connectivity is as follows

```
test@ubuntu:~# ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=64 time=0.439 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=64 time=0.455 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=64 time=0.390 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=64 time=0.473 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.390/0.439/0.473/0.034 ms
```

5) At this time, if the Wan port is connected to the network, then the Ubuntu PC can also be connected to the Internet

```
test@ubuntu:~# ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=55 time=7.32 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=55 time=7.72 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=55 time=7.80 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=55 time=7.05 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 7.055/7.478/7.808/0.305 ms
```

3.8. USB interface test

3.8.1. Mounting a USB storage device at the command line

1) First insert the U disk into the USB port of the Orange Pi development board

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

```
root@OpenWrt:~# cat /proc/partitions | grep "sd*"
major minor  #blocks  name
   8        0   15126528 sda
```

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

```
root@OpenWrt:~# mount /dev/sda /mnt/
root@OpenWrt:~# ls /mnt/
test.txt
```

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

```
root@OpenWrt:~# df -h | grep "sd"
/dev/sda                14.4G       187.2M        14.2G     1% /mnt
```

3.8.2. Mount the USB storage device on the LuCI management interface

1) First connect the U disk (or other storage device) to the development board via USB2.0

2) Then enter the mount point management interface

a. Find the "System" option in the navigation bar and click to enter

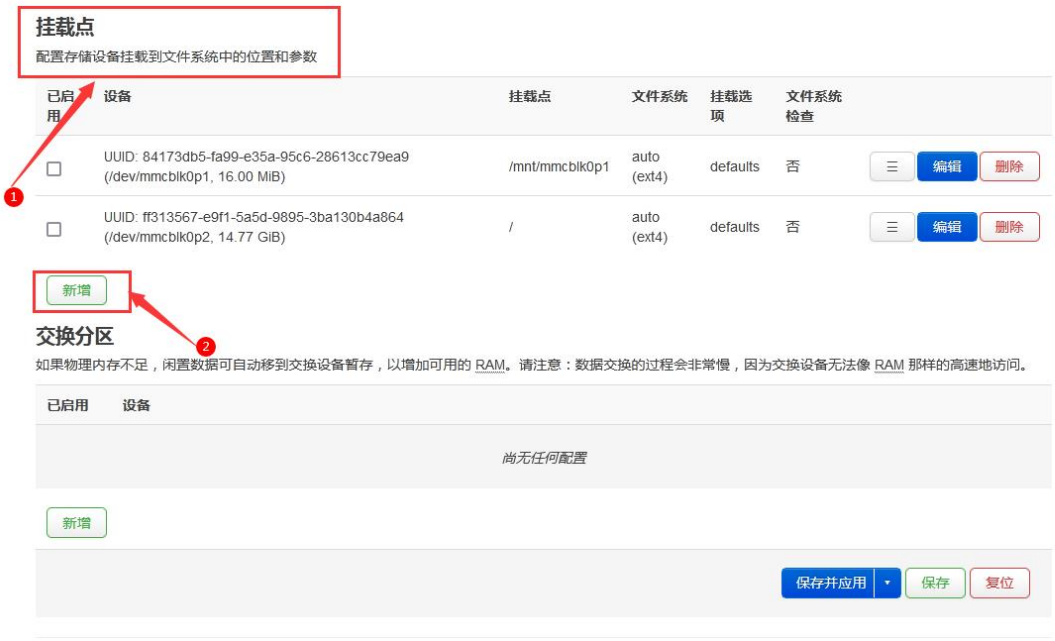
b. In the vertical bar options below the system, select "Mount Point" and click to enter

3) Then you will enter the mount point global settings page

4) Then add a mount point

a. Find "Mount Point" at the bottom of the mount point global setting interface

b. Below the mount point, select the "Add" button and click to enter



c. Then the following pop-up interface will pop up

d. Then you can start to mount the storage device

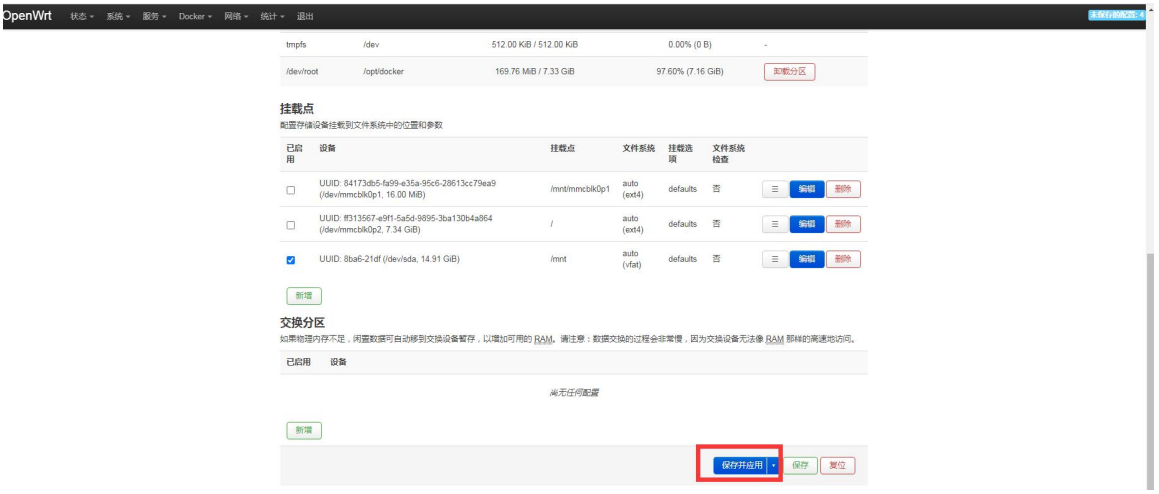
a) Check "Enabled"

b) In the general setting UUID column, select the actual connected device /dev/sda (select according to your own device)

c) Select "Custom" in the column of mount point, and fill in the target directory to be mounted to. Here, take the /mnt directory as an example, press Enter to confirm after filling in

d) Then click the "Save" button in the lower right corner

5) Then it will return to the mount point global settings page, click "Save and Apply" in the lower left corner of the page to make the mount point take effect



6) After saving, you can see the "Mounted file system", the storage device has been mounted

3.9. USB wireless network card test

The list of wireless network cards supported by the OpenWRT system is as follows. Among them, RTL8723BU and MT7601U only support Station Mode, so the test method is the same. Here only RTL8723BU is used as an example.

芯片型号	VID&PID	AP Mode	Station Mode
RTL8821CU / RTL8811CU	0bda:8153	OK	OK
RTL8723BU	0bda:b720	NO	OK
MT7601U	148f:7601	NO	OK

3.9.1. RTL8811CU test

1）Prepare the necessary accessories

a. Realtek RTL8811CU USB wireless network card module

2）Insert the USB wireless network card into the USB port of the development board, and execute the following command in the terminal to check the model of the USB wireless network card

```
root@OpenWrt:~# lsusb
Bus 002 Device 001: ID 1d6b:0001 Linux 5.4.154 ohci_hcd Generic Platform OHCI cr
Bus 004 Device 002: ID 0bda:8153 Realtek USB 10/100/1000 LAN
Bus 004 Device 001: ID 1d6b:0003 Linux 5.4.154 xhci-hcd xHCI Host Controller
Bus 001 Device 002: ID 0bda:c811 Realtek 802.11ac NIC
Bus 001 Device 001: ID 1d6b:0002 Linux 5.4.154 ehci_hcd EHCI Host Controller
Bus 003 Device 001: ID 1d6b:0002 Linux 5.4.154 xhci-hcd xHCI Host Controller
```

3）Then execute reboot to restart the system

```
root@OpenWrt:~# reboot
```

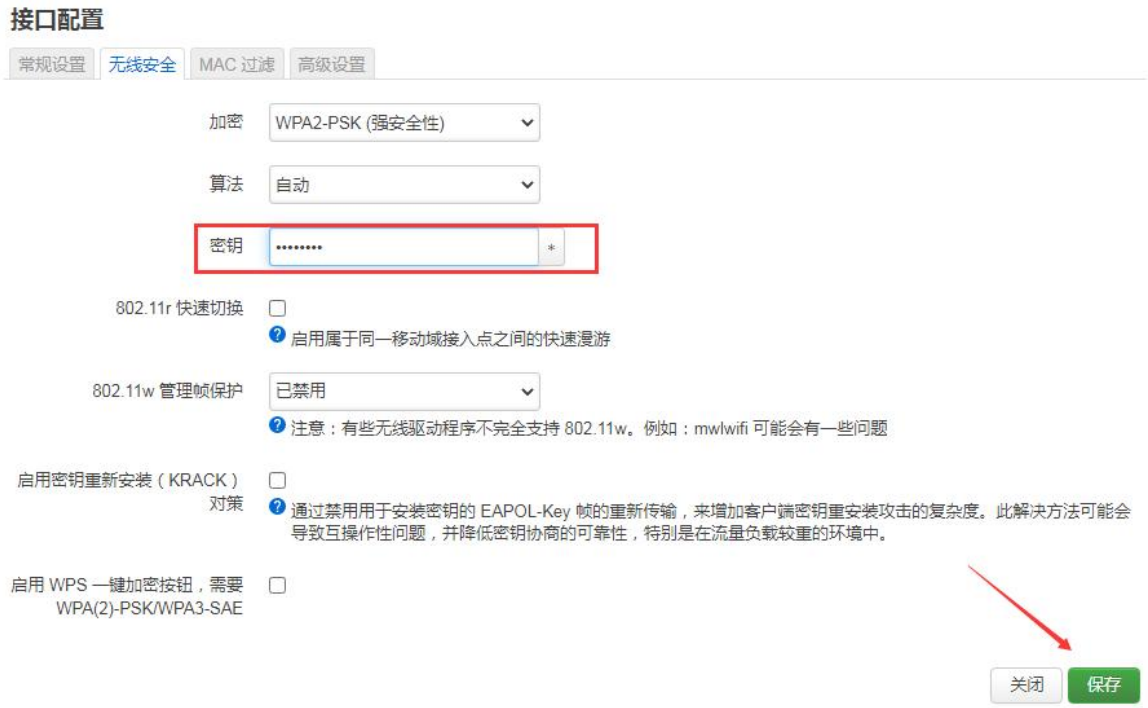
4）After the system restarts, click "Network > Wireless" to enter the wireless WiFi configuration interface

5）Click to enable, an open hotspot named "OpenWrt" will be created by default

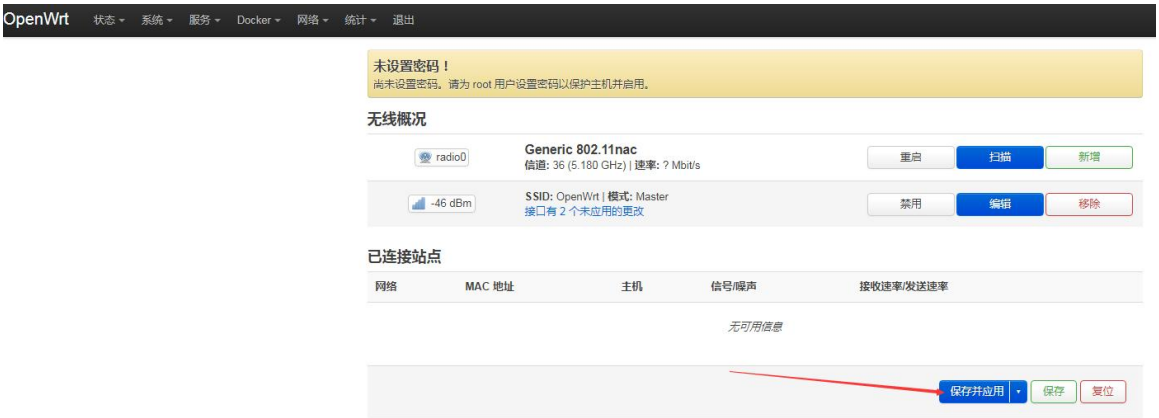
6）Use your mobile phone or computer to search for the WiFi connection corresponding to the SSID

7）If you need to set a password for the hotspot, first click the "Edit" button

8）Then enter the key in the "Wireless Security" column and click "Save"



9）Then click "Save and Apply" on the main interface of wireless settings



10）Click on the menu bar "System > Restart"

11）Finally click the "Execute Reboot" button to reboot the system

12）After the system restarts, you can see that the WiFi hotspot has been encrypted

3. 9. 2. RTL8723BU test

1）Prepare the necessary accessories

a. Realtek RTL8723BU USB wireless network card module

a. Realtek RTL8723BU USB wireless network card module

```
root@OpenWrt:~# lsusb
Bus 002 Device 001: ID 1d6b:0001 Linux 5.4.154 ohci_hcd Generic Platform OHCI cr
Bus 004 Device 002: ID 0bda:8153 Realtek USB 10/100/1000 LAN
Bus 004 Device 001: ID 1d6b:0003 Linux 5.4.154 xhci-hcd xHCI Host Controller
Bus 001 Device 002: ID 0bda:b720 Realtek 802.11n WLAN Adapter
Bus 001 Device 001: ID 1d6b:0002 Linux 5.4.154 ehci_hcd EHCI Host Controller
Bus 003 Device 001: ID 1d6b:0002 Linux 5.4.154 xhci-hcd xHCI Host Controller
```

2）Then execute reboot to restart the system

```
root@OpenWrt:~# reboot
```

3）After the system restarts, click "Network > Wireless" to enter the wireless WiFi configuration interface

4）Because RTL8723BU does not support AP, click the "Remove" button to delete the AP configuration

5）First click the "Scan" button, wait for a while, the list of scanned WIFI hotspots will appear, select the WIFI hotspot you want to connect, and click the "Join Network" button

6）Then enter the WIFI password in the pop-up window and click the "Submit" button

7）Then click "Save" in the pop-up window

8）Click "Save and Apply" in Wireless Configuration

13）Click on the menu bar "System > Restart"

14）Finally click the "Execute Reboot" button to reboot the system

15）After restarting the system, enter the wireless configuration interface again, and you can see that the hotspot is successfully connected

3. 10. On-board LED light test description

1) There are three LED lights on the development board, two yellow lights and one red light. The default display of the LED lights when the system starts is as follows

Power status light (red light)	The system starts, the red light is always on
Wan port status light (yellow light)	When the Wan port is connected to the network cable, the yellow light flashes. When the network cable is unplugged from the Wan port, the yellow light is off
Lan port status light (yellow light)	When the LAN port is connected to the network cable, the yellow light flashes, and the network cable is unplugged from the Lan port, and the yellow light is off

2) The method of setting the red/yellow light on and off and flashing is as follows (the red light is used as an example below)

a. First enter the setting directory of the red light

```
root@OpenWrt:~# cd /sys/class/leds/orangepi-r1-plus-lts:red:sys
```

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

```
root@OpenWrt:/sys/class/leds/orangepi-r1-plus-lts:red:sys# echo 0 > brightness
```

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

```
root@OpenWrt:/sys/class/leds/orangepi-r1-plus-lts:red:sys# echo 1 > brightness
```

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

```
root@OpenWrt:/sys/class/leds/orangepi-r1-plus-lts:red:sys# echo heartbeat > trigger
```

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

```
root@OpenWrt:/sys/class/leds/orangepi-r1-plus-lts:red:sys# echo none > trigger
```

3. 11. Installing packages via the command line

3. 11. 1. Install via opkg in the terminal

1) Update the list of available packages

```
root@OpenWrt:/# opkg update
```

2) Get the software list

```
root@OpenWrt:/# opkg list
```

3) Install the specified package

```
root@OpenWrt:/# opkg install <package name>
```

4) View installed software

```
root@OpenWrt:/# opkg list-installed
```

5) Uninstall the software

```
root@OpenWrt:/# opkg remove <package name>
```

3. 11. 2. Installing packages from unofficial sources

OpenWRT can easily install thousands of software packages from software sources through the above method, which can meet most of the needs of users. However, sometimes we still need to find other packages from the official sources to meet some special needs. When installing third-party software packages, you need to pay attention that some software packages have system architecture restrictions. The system architecture of Orange Pi R1 Plus LTS is aarch64_generic, which also

belongs to armv8. You need to select the software package of the corresponding architecture when downloading. Find downloads on github or other forums

3. 11. 3. Common installation errors

1) The package system architecture is incorrect, you need to download the package with the suffix **aarch64_generic**

```
root@OpenWrt:~# opkg install v2ray-core-mini_4.35.1-2_aarch64_cortex-a53.ipk
Unknown package 'v2ray-core-mini'.
Collected errors:
 * pkg_hash_fetch_best_installation_candidate: Packages for v2ray-core-mini found, but incompatible with the architectures configured
 * opkg_install_cmd: Cannot install package v2ray-core-mini.
```

2) Force uninstall the old version of the library

The related library is already provided by the old version of the software, which causes the installation to fail. You can choose to continue to use the old version of the library, or uninstall the old version of the software and then install the new version of the software, as shown below,When there are other libraries that depend on libnettle7, it will prompt that libnettle7 cannot be uninstalled. At this time, you need to add the --force-depends parameter, ignore the dependencies and uninstall libnettle7 directly.

```
root@OpenWrt:~# opkg install libnettle8
Installing libnettle8 (3.6-1) to root...
Downloading https://downloads.openwrt.org/snapshots/packages/aarch64_generic/base/libnettle8_3.6-1_aarch64_generic.ipk
Collected errors:
```

```
* check_data_file_clashes: Package libnettle8 wants to install file /usr/lib/libhogweed.so.6
        But that file is already provided by package  * libnettle7
root@OpenWrt:~# opkg remove libnettle7 --force-depends   #ninstall libnettle7
root@OpenWrt:~# opkg install libnettle8                  #Install libnettle8
```

3. 12. OpenWRT management interface installation package

If you need to add new packages, you can install them through the management interface of OpenWRT

3. 12. 1. View the list of available software packages in the system

1) First enter the package management page
   a. Find the "System" option in the navigation bar and click to enter
   b. In the vertical bar options below the system, select "Package" and click to enter

2) Then the main page of the package will appear, as shown in the image below, to get a list of available software
   a. In the "Action" option of the package, click "Update List" to get a list of available packages
   b. On the Tab page, click "Available" to view the currently available software packages
   c. View the number of currently available packages

3. 12. 2. Example of installing packages

1) Take the installation package "luci-app-acl" as an example

b. In the OpenWRT package management interface, click the filter dialog and enter "luci-app-acl"

c. You can see the version, package size and description information of the "netdata" package in the list of packages, and then click the "Install" button

d. Then the following pop-up window will appear, click "Install"

e. Then wait for the installation to complete

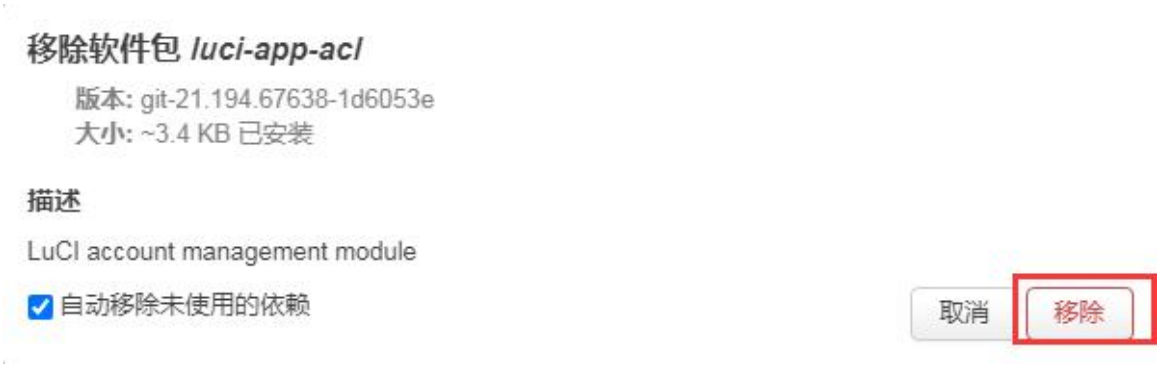
f. The installation is completed and the display is as follows

2) Check if the package is installed successfully
   a. In the OpenWRT package management interface, click the filter dialog and enter "luci-app-acl"
   b. On the Tab page, select and click "Available"On the Tab page, select and click "Available"
   c. The "luci-app-acl" package will be displayed in the package list, and the update status will be "Installed"

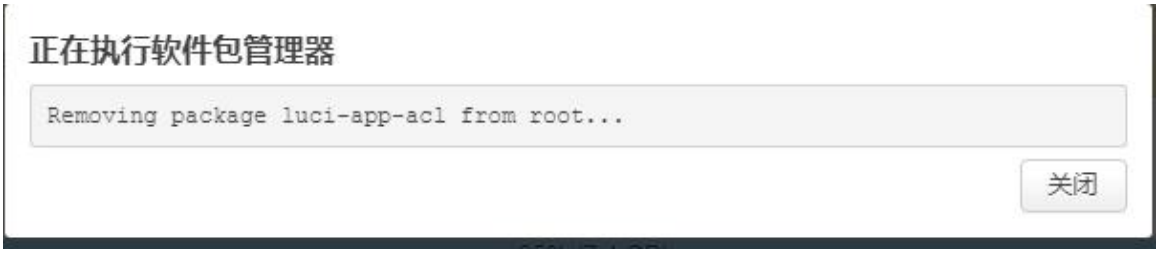
3. 12. 3. Example of removing a package

1) Take the removal of the package "luci-app-acl" as an example
   a. In the OpenWRT package management interface, click the filter dialog and enter "luci-app-acl"
   b. Select "Installed" in the Tab page to display the list of installed packages
   c. Click "Remove" on the right to remove the corresponding software package

a. Then the following pop-up window will be displayed, click "Remove"



b. After the removal is successful, the display interface is as follows



2) Check if the package is removed successfully

a. In the OpenWRT package management interface, click the filter dialog and enter "luci-app-acl"

b. On the Tab page, select and click "Installed"

c. The "luci-app-acl" package will not be displayed in the package list, and the "luci-app-acl" package has been removed successfully

3. 13. The use of Nextcloud network disk

Nextcloud is an open source and free private cloud storage network disk project that allows you to quickly and easily build a set of cloud synchronization network disks belonging to yourself or your team, so as to achieve cross-platform and cross-device file synchronization, sharing, version control, team collaboration, etc. Function. For more information about Nextcloud, please refer to the official documentation of Nextcloud

3. 13. 1. Install Nextcloud via Docker

1) First create a folder for Nextcloud

```
root@OpenWrt:~# mkdir /nextcloud -p
```

2) Then use the docker command to download the docker image of Nextcloud

```
root@OpenWrt:~# docker run -d -p 8888:80 --name nextcloud -v
/nextcloud/:/var/www/html/ --restart=always --privileged=true    arm64v8/nextcloud
                                                                  (this is a command)
```

3) The process of downloading the image is shown in the figure below

```
root@OpenWrt:~# mkdir /nextcloud -p
root@OpenWrt:~# docker run -d -p 8888:80 --name nextcloud -v /nextcloud/:/var/www/html/ --restart=always --privileged=true  arm64v8/nextcloud
Unable to find image 'arm64v8/nextcloud:latest' locally
latest: Pulling from arm64v8/nextcloud
8998bd30e6a1: Extracting [==================================>              ]  21.63MB/30.06MB
f8ef0b26fc2c: Download complete
7f8d4d2813bc: Download complete
787dfbf2c706: Download complete
095f089a21c2: Download complete
e11ff0302933: Download complete
0897c0c6da4a: Download complete
bed34cb1130a: Download complete
0ea3f9b85722: Download complete
bc1b3aaec12d: Download complete
f09bfa1c045d: Download complete
75b6580cdd47: Download complete
2d7ab8dd3008: Download complete
d13fc0a35dec: Downloading [=============>                                    ]  392.4kB/1.432MB
8814fb69ab6e: Pulling fs layer
6607c4524e98: Waiting
b63cda99e1ba: Waiting
33b9fcaf8204: Waiting
040c40da721c: Waiting
0b0b9d26f6ac: Waiting
```

4) After the download is complete, you can check whether Nextcloud is running on the LuCI management interface of OpenWrt

a. First select "**Container**" in the "**Docker**" column and click to enter

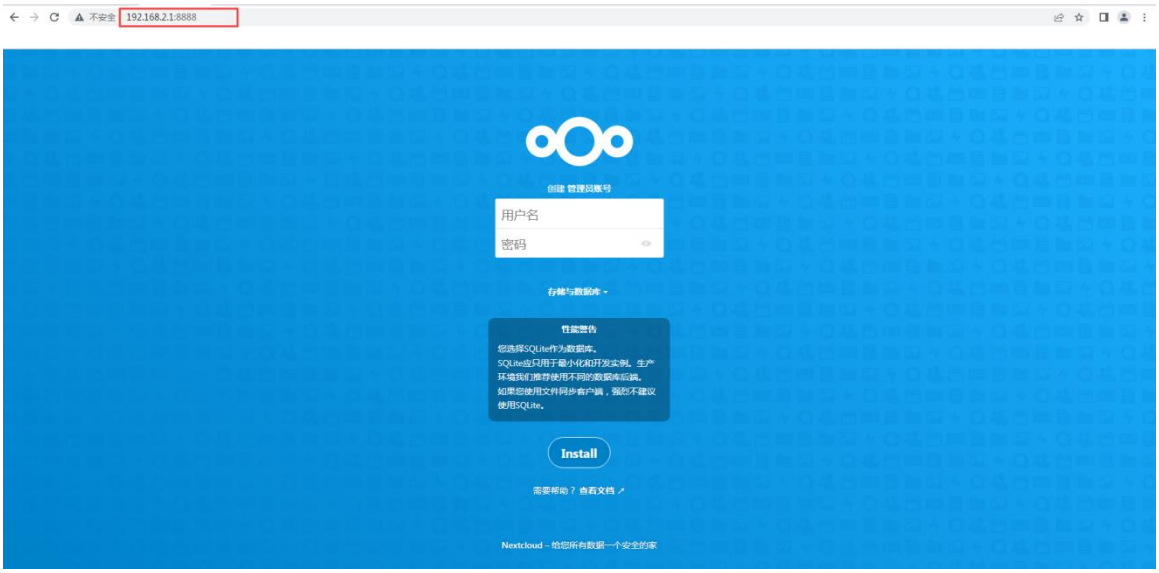
b. Then the following interface will be displayed, if the status is green, it means Nextcloud is already running

3. 13. 2. Log in to the Nextcloud network disk

Nextcloud can only be accessed through the LAN port by default, and cannot be accessed through the WAN port, so the host computer needs to be connected to the LAN port of the Orangepi R1 Plus LTS development board. At this time, you can

log in through the IP address of the LAN port + port number 8888

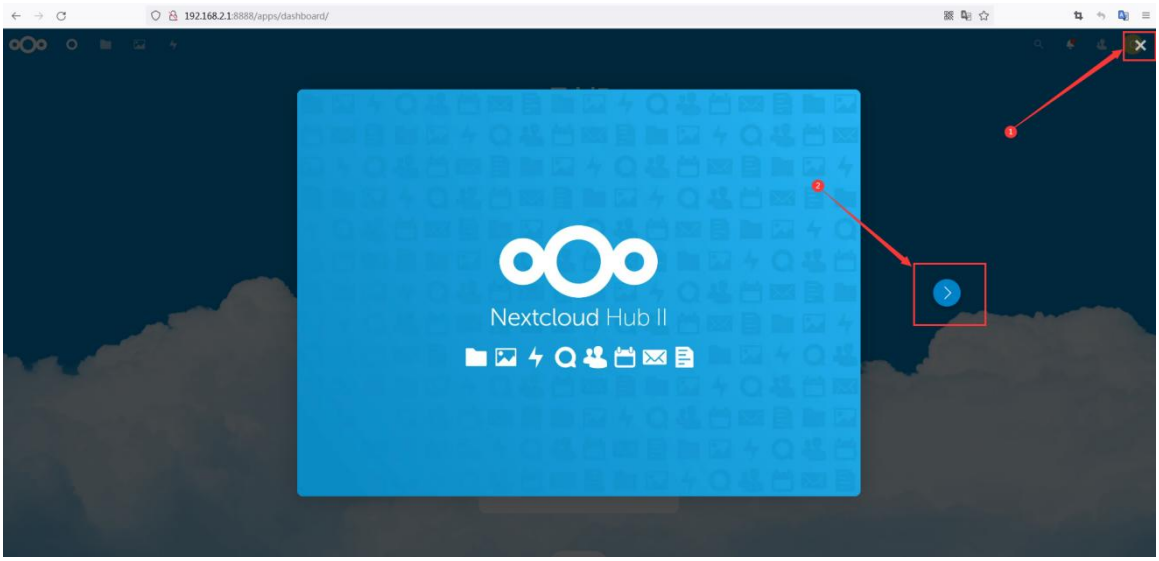
1) First, enter 192.168.2.1:8888 in the address bar of the browser to open the login interface of Nextcloud



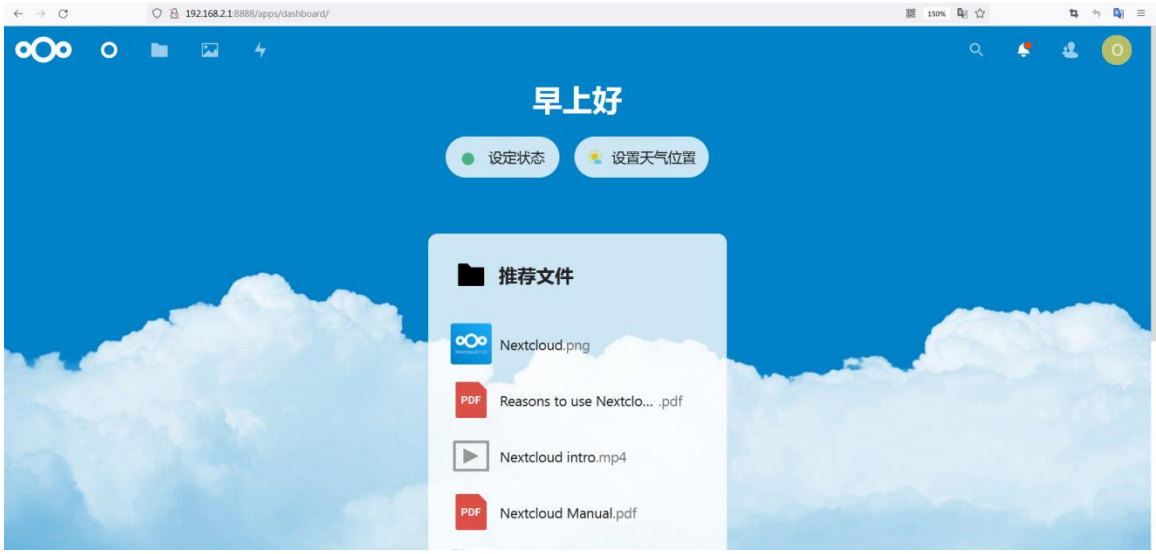
2) Then you need to create an administrator account and password
   a. First enter the username and password in the appropriate places
   b. In the storage and repository, the default data directory is /var/www/html/data, and the default database is "SQLite". If you need to modify it, please configure it yourself
   c. Finally, click "**Install**" at the bottom of the login interface to install, and you need to wait for a while.

3) After the installation is complete, the interface as shown in the figure below will pop up, click "**Cancel**" to

4) Then an interface will pop up to introduce Nextcloud, here you can click the "**X**" in the upper right corner to skip the introduction interface



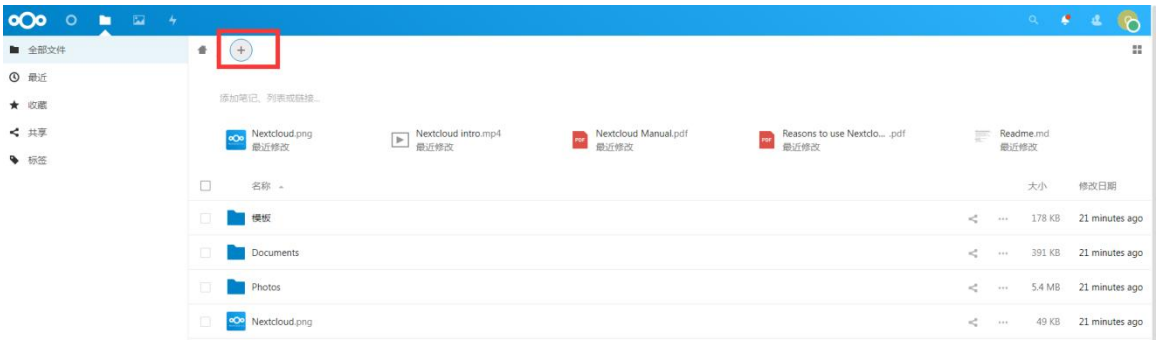
5) Then enter the main interface of Nextcloud



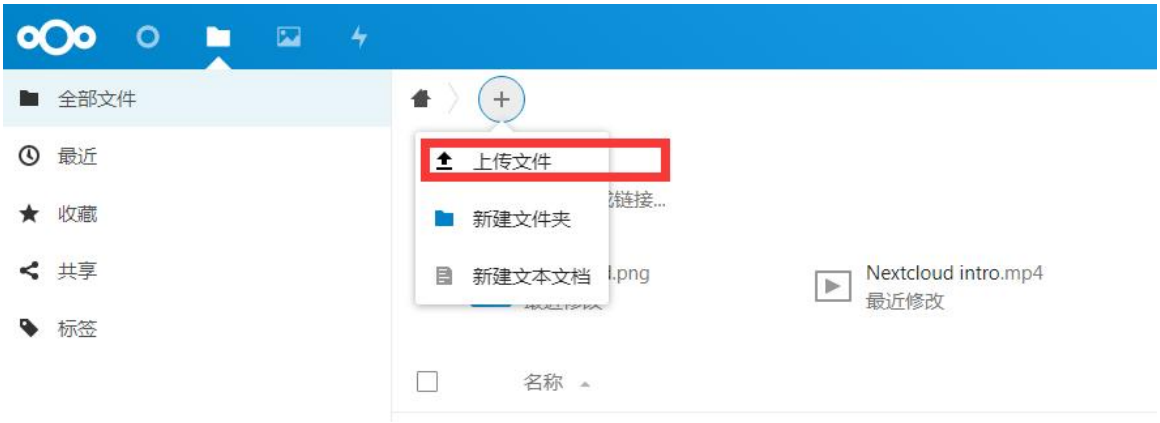
3. 13. 3. Upload a file to the Nextcloud network disk

1) First click "File" on the top left of the Nextcloud main page

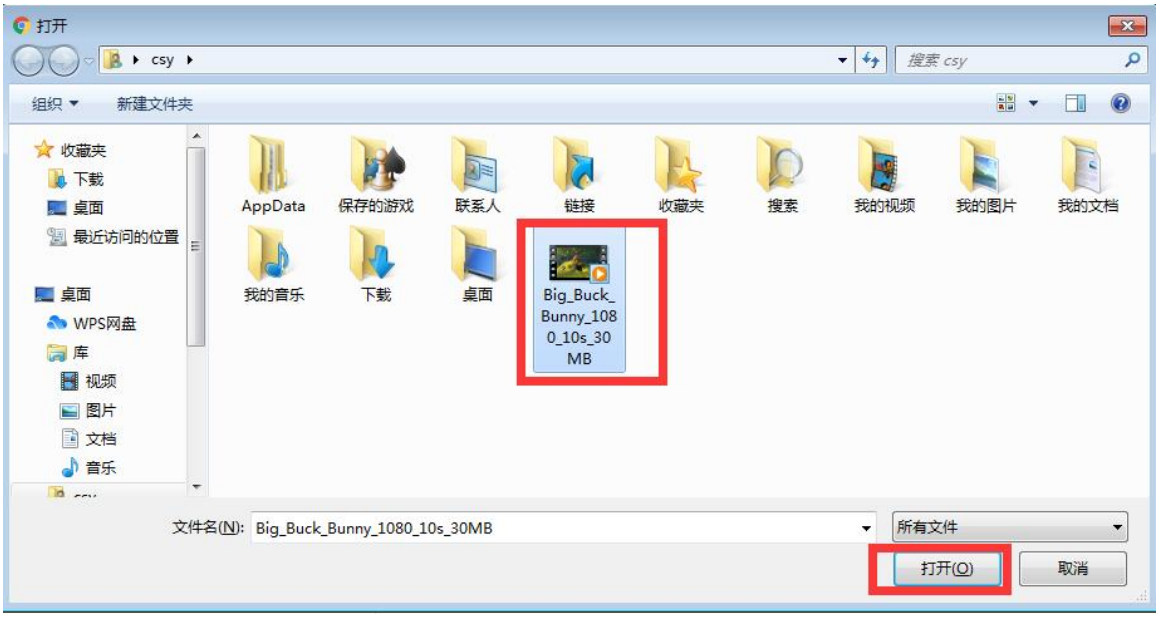
2) Then it will enter the file management interface of Nextcloud, at this time click the "+" sign in the red box below



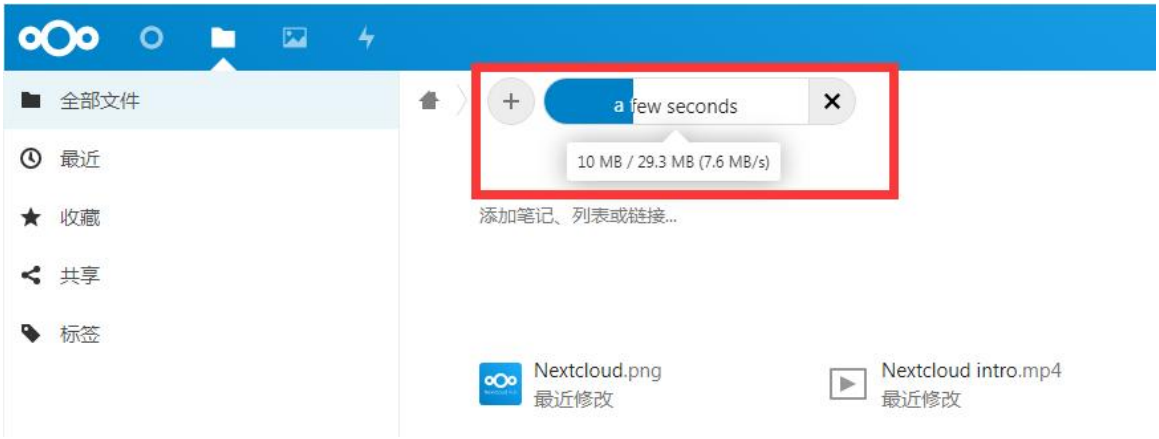
3) Then select "Upload File" in the pop-up options



4) Then select the file you want to upload in the pop-up window, and click "Open"



5) At this point, the file will start to upload, move the mouse to the position of the progress bar, and you can also see the upload speed



3. 14. Use of Portainer

Portainer is a visual Docker operation interface that provides status display panel, rapid deployment of application templates, basic operations of container image network data volumes (including uploading and downloading images, creating containers, etc.), event log display, container console operations, and Swarm clusters. For more information about Portainer, please refer to the official documentation of Portainer

3. 14. 1. Installing Portainer via Docker

1) First use the docker command to download the Portainer image

```
root@OpenWrt:~# docker run -d -p 9000:9000 --name portainer --restart always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data
portainer/portainer                                    （this is a command）
```

Parameter Description:

a. -d：the container is running in the background

b. -p 9000:9000 ： The host 9000 port maps the port 9000 in the container (the front is the host port, the latter is the container port)

c. –restart The flag will check the exit code of the container and decide whether to restart the container based on this, which will not restart by default

d. –restart=always：Automatically restart the container

e. -v /var/run/docker.sock:/var/run/docker.sock ： Mount the Unix domain socket that the host's Docker daemon listens to by default into the container

f. -v portainer_data:/data ：Mount the host portainer_data data volume to the container/data directory

g. --name portainer ： Name the container portainer

2) The process of downloading the image is shown in the figure below

```
root@OpenWrt:/# docker run -d -p 9000:9000 --name portainer --restart always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/port
ainer
Unable to find image 'portainer/portainer:latest' locally
latest: Pulling from portainer/portainer
94cfa856b2b1: Pull complete
49d59ee0881a: Pull complete
c71f4038b17b: Pull complete
Digest: sha256:fb45b43738646048a0a0cc74fcee2865b69efde857e710126084ee5de9be0f3f
Status: Downloaded newer image for portainer/portainer:latest
3495a0281cc81979323504415c5c587c6a770eb6f6ed9020428587581cec6a9e
[11213.038514] docker0: port 2(veth3aede8c) entered blocking state
[11213.039058] docker0: port 2(veth3aede8c) entered disabled state
[11213.040355] device veth3aede8c entered promiscuous mode
[11213.945214] eth0: renamed from vethbc629c2
[11213.966104] IPv6: ADDRCONF(NETDEV_CHANGE): veth3aede8c: link becomes ready
[11213.967105] docker0: port 2(veth3aede8c) entered blocking state
[11213.967695] docker0: port 2(veth3aede8c) entered forwarding state
```

3) After the download is complete, you can check whether Portainer is running on the LuCI management interface of OpenWrt

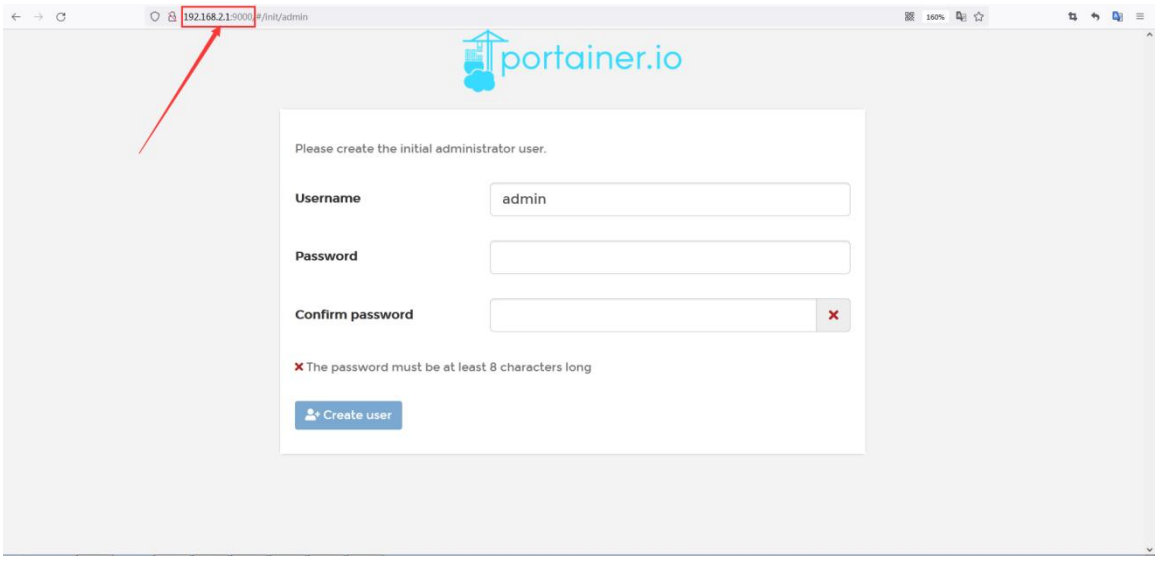
a. Select "**Container**" in the "**Docker**" column and click to enter

b. Then the following interface will be displayed, if the status is displayed in green, it means it is already running

3. 14. 2. og in to Portainer's management interface

The Portainer management interface can only be accessed through the LAN port by default, and cannot be accessed through the WAN port, so the computer host needs to be connected to the LAN port of the Orangepi R1 Plus LTS development board. At this time, you can log in through the IP address of the LAN port + port number 9000

1) First, enter **192.168.2.1:9000** in the address bar of the browser to open the login interface of Portainer



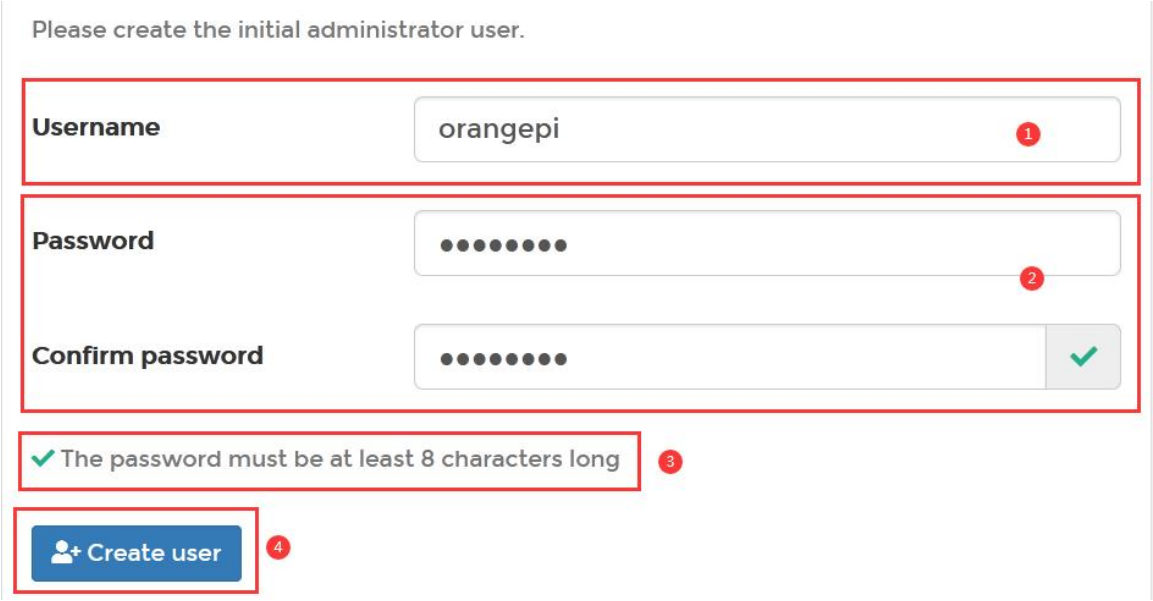
2) Then create a username and password

a. First enter your username in the "**Username**" of the registration interface

b. Then enter your password and "**Confirm password**" in the "**Password**" of the registration interface to confirm your password (**the password must be more**

than eight characters)

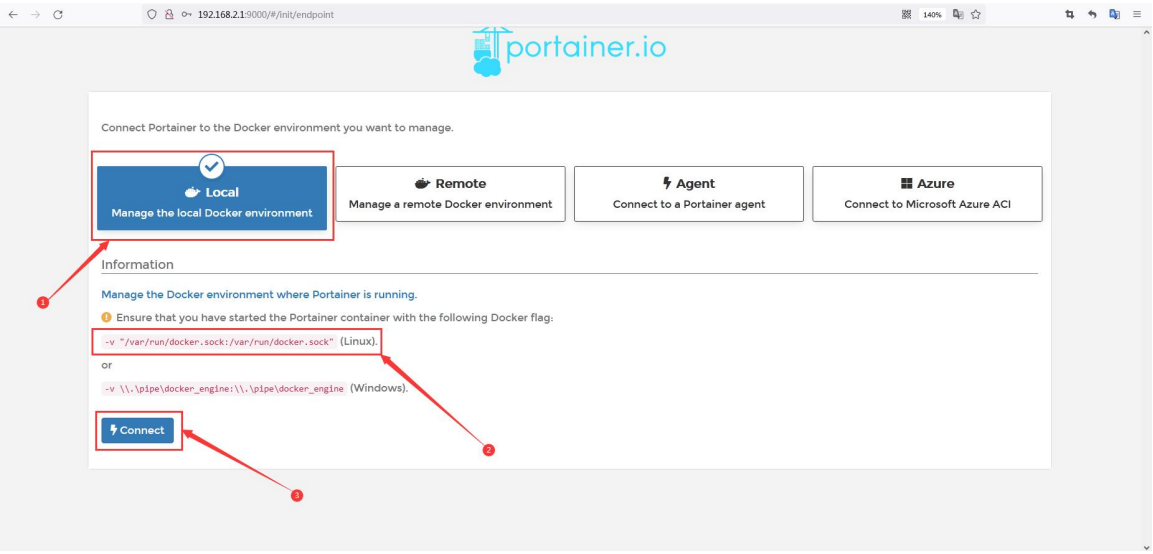
c. Finally click the "**Create user**" button to create an account and password



3) After creating a user, you need to configure Portainer to connect to the Docker environment you want to manage

a. First select "**Local**"

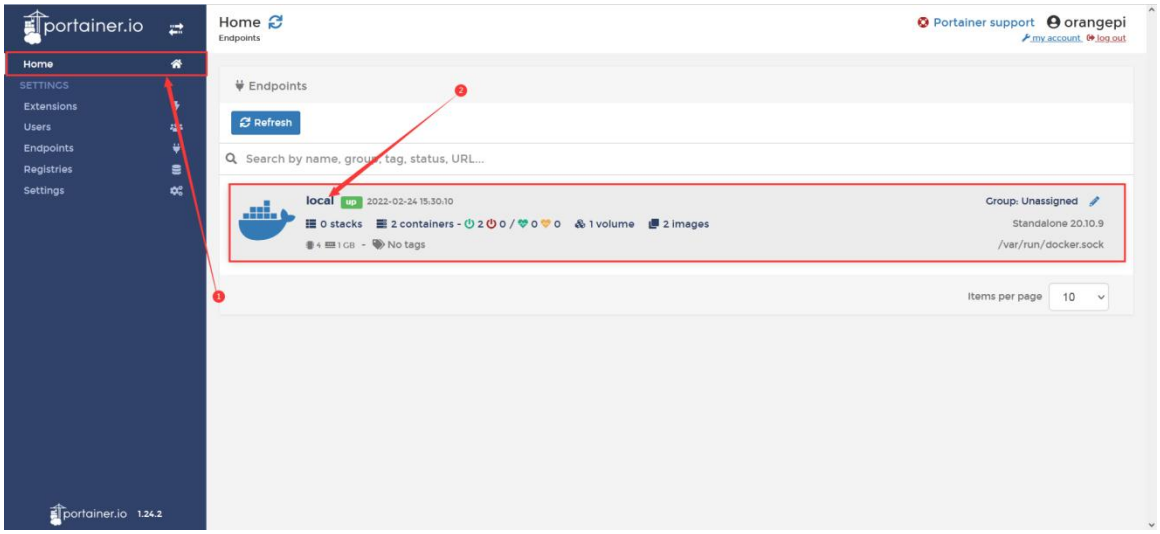
b. Then click "**Connect**" to connect to the local Docker environment



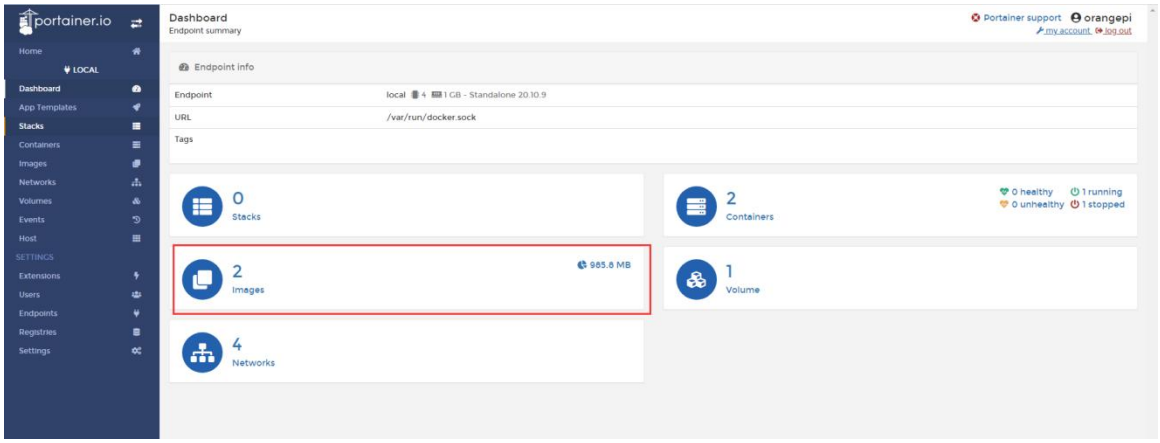
3. 14. 3. View installed docker images in Portainer

1) Find the "**Home**" option in the Portainer navigation bar and click to enter the home page

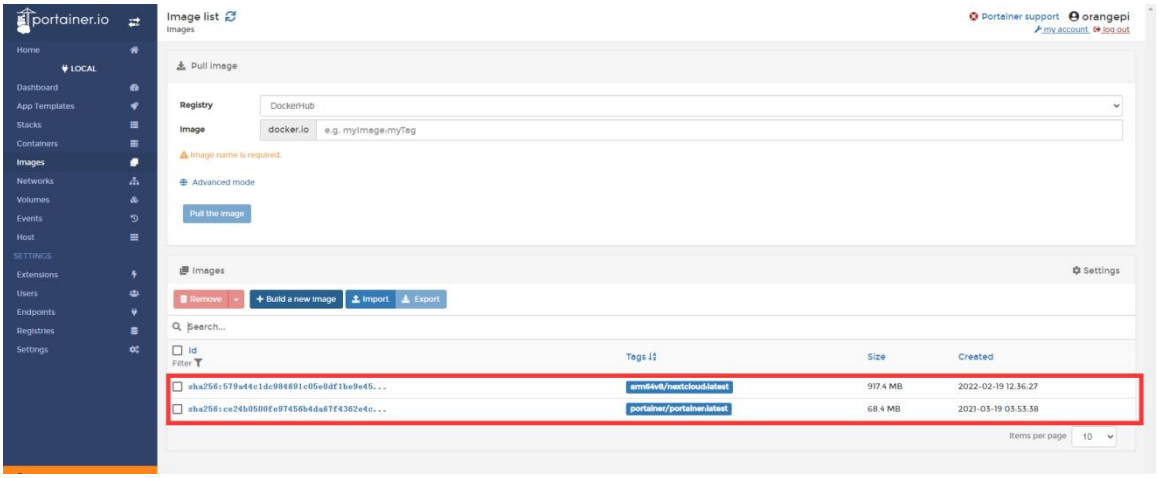
2) Then select "**local**" and click to select the environment



3) Then the displayed interface is as shown in the figure below, click the "**Images**" option at this time

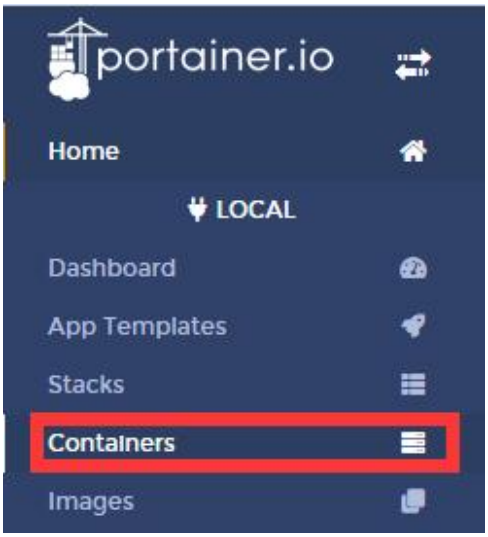


4) Then the displayed interface is shown in the figure below, you can see that the installed docker images are Nextcloud and Portainer

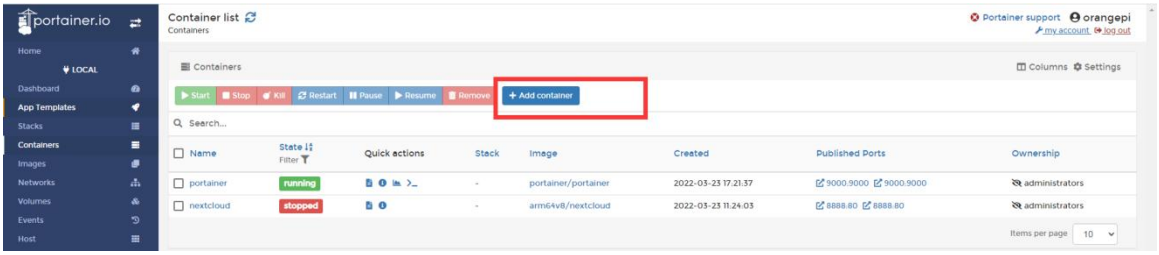


3. 14. 4. How to install jellyfin through Portainer

1) First, find the "**Containers**" option in the navigation bar in the upper left corner of the Portainer interface and click to enter



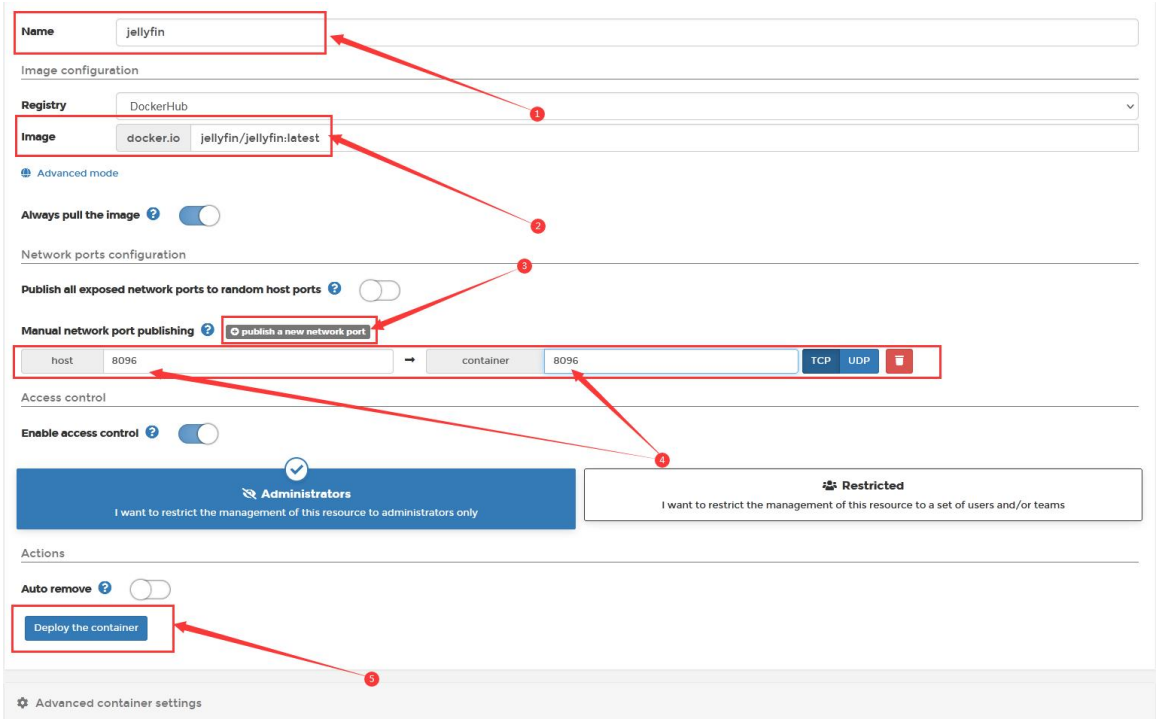
2) Then click "**Add Containers**" to add a container



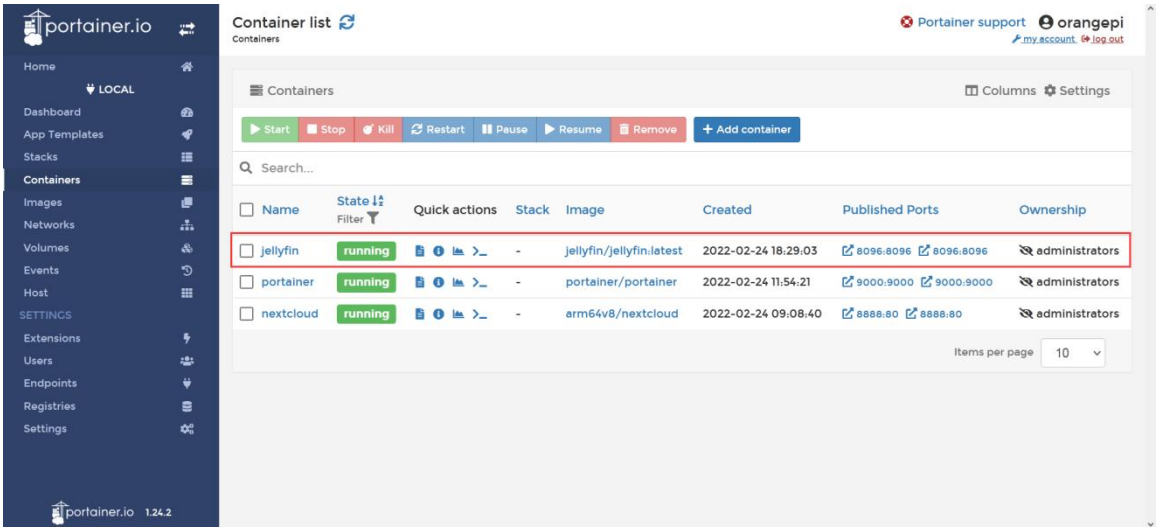
3) Then you will enter the interface of creating a container

a. Enter the name of the jellyfin image in the "**Name**" column, such as "**jellyfin**"

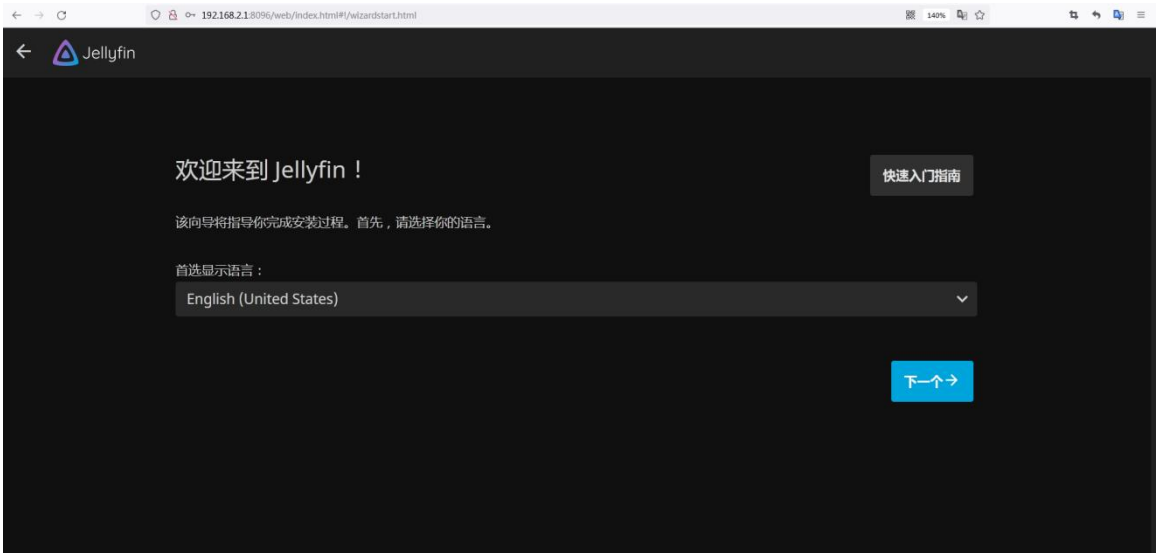
b. Enter the image name "**jellyfin/jellyfin:latest**" in the "**Image**" column
c. Click "**publish a new network port**" to add a fixed port number for jellyfin
d. Fill in the port number in both "**host**" and "**container**" as 8096
e. Click "**Deploy the container**" to deploy the jellyfin container, you will need to wait for a while



4) After the deployment of the jellyfin image is completed, it will automatically return to the main page of Containers. At this time, you can see that a jellyfin docker image has been added to the Containers



5) Enter **192.168.2.1:8096** in the address bar of the browser to access the main page of jellyfin



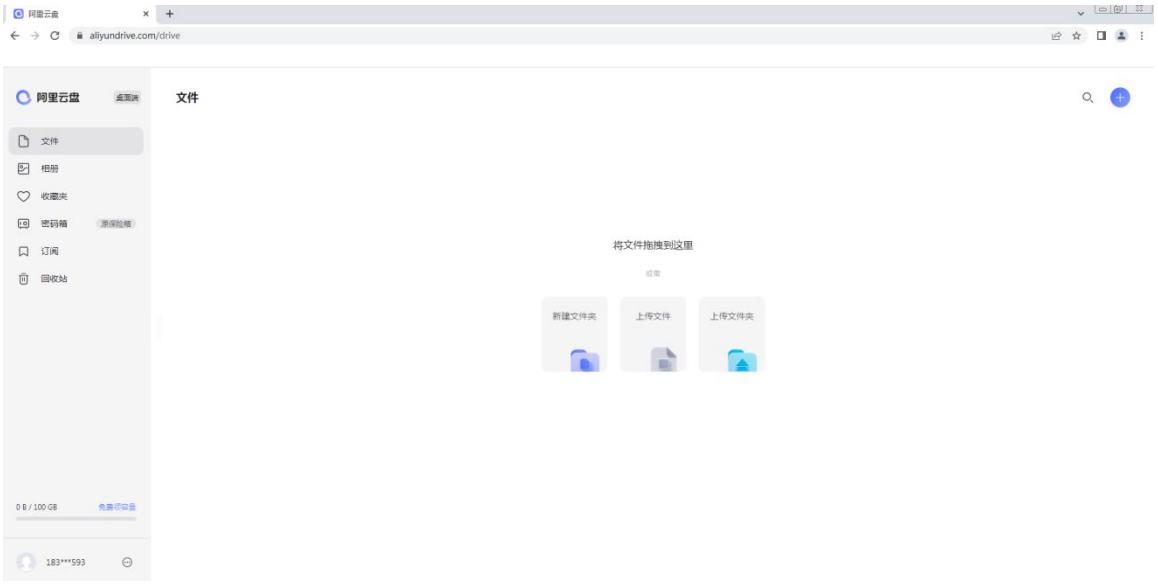
3. 15. Alibaba Cloud Disk WebDAV Service

The Alibaba Cloud WebDAV service is mainly used in conjunction with client apps that support the WebDAV protocol. For example, Infuse, nPlayer, etc. can directly watch the video content of Alibaba Cloud Disk on TV, and support downloading files, but not uploading files

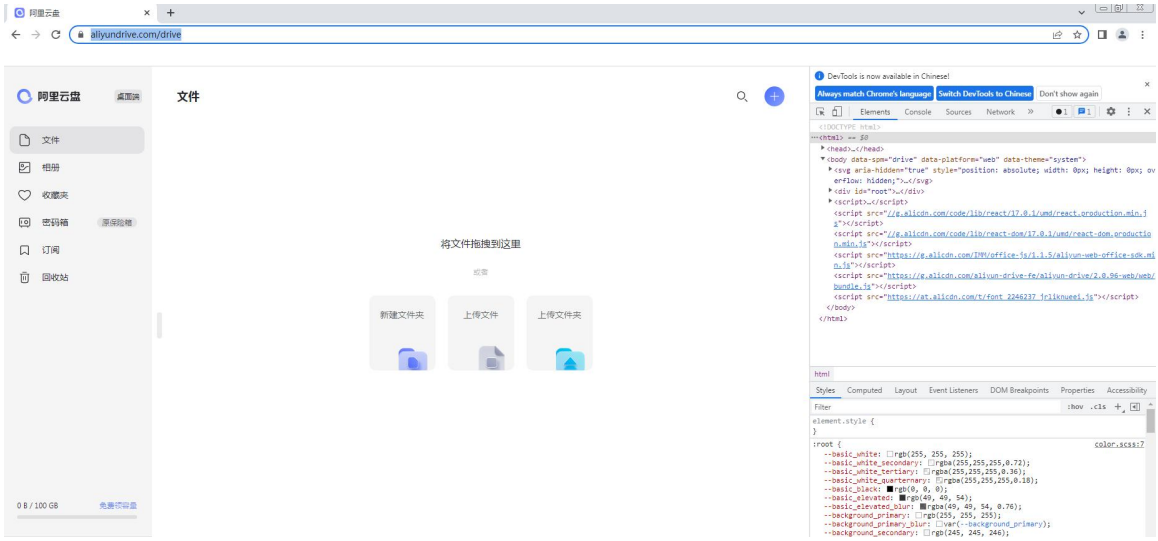
3. 15. 1. Get "refresh_token" in the Alibaba Cloud Disk web version

1) First open the official website of Alibaba Cloud Disk through a browser, and register an Alibaba Cloud Disk account with your mobile phone number

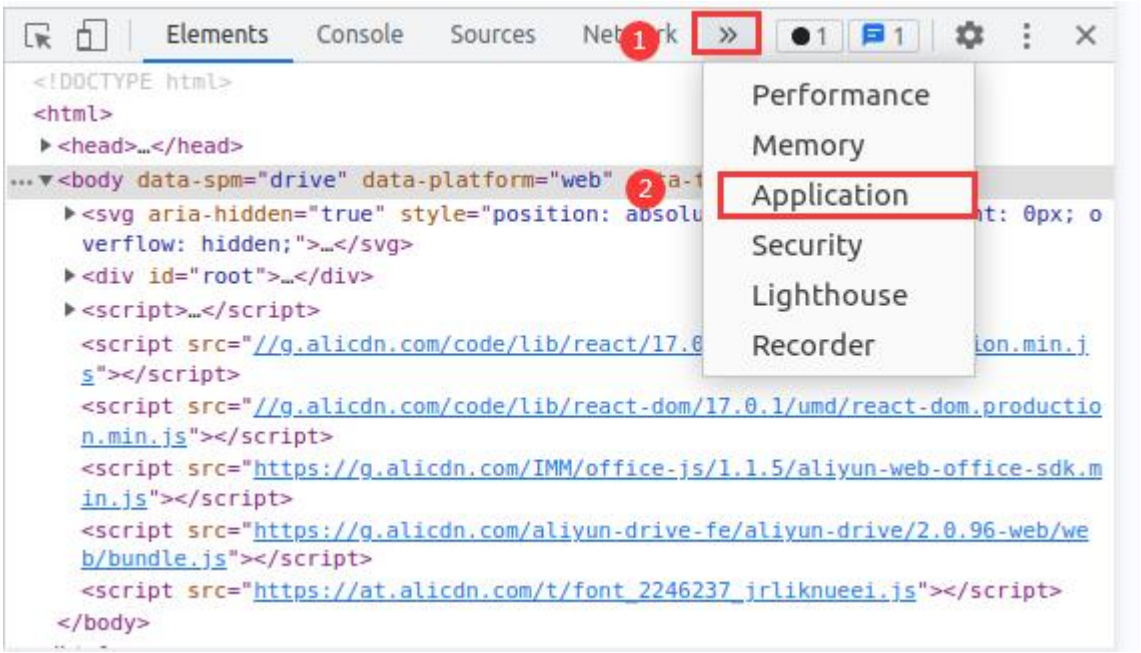
2) Then log in with the registered Alibaba Cloud Disk account. After logging in, the interface is as shown below



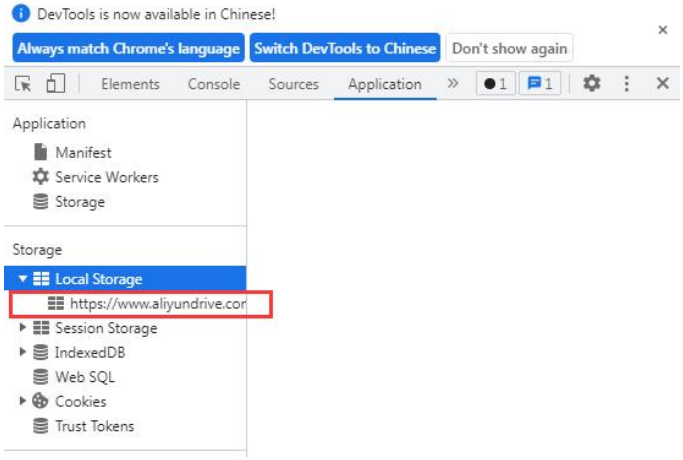
3) Then press the "**F12**" key of the keyboard to enter the browser developer mode



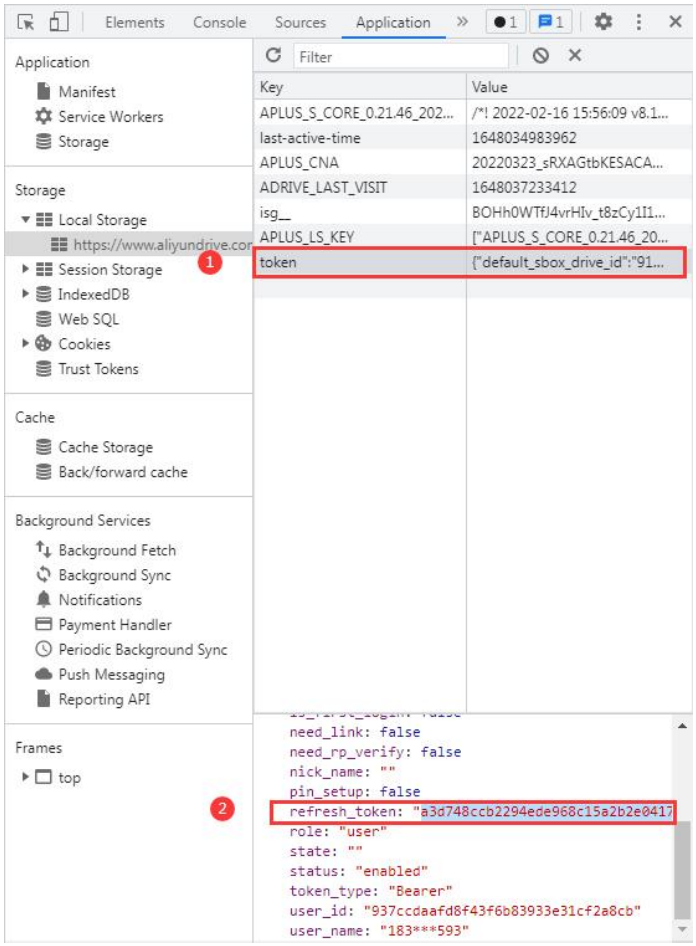
4) Then click ">>" in the developer toolbar, and select "**Application**" in the pop-up options



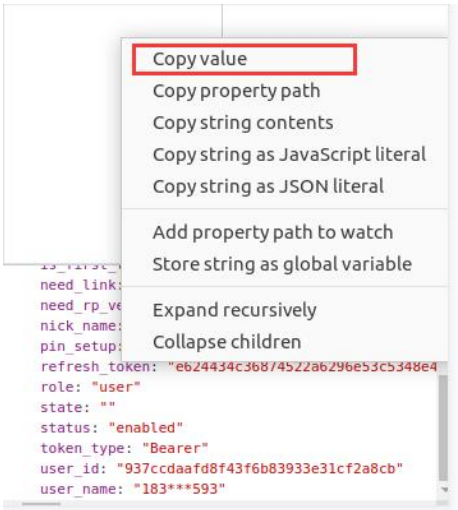
5) Then click "**Local Storage**" in the following interface, and select the link below



6) Then click "**token**" in the following interface, drag the scrolling toolbar in the lower right corner, and find "**refresh_token**"

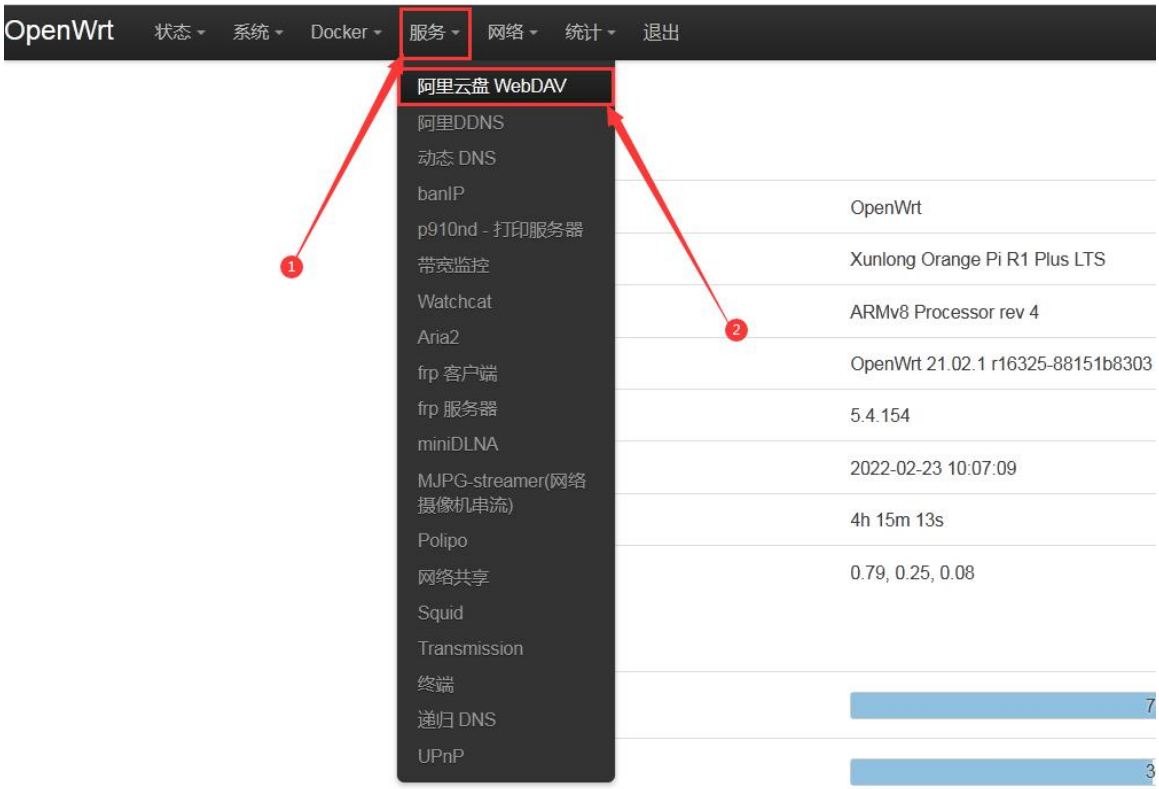


7) Then right-click "**refresh_token**", select "**Copy value**" in the pop-up options, copy the value of refresh_token, which will be used later



3. 15. 2. Configure Alibaba Cloud WebDAV service on the OpenWRT management page

1) First, select "**Alibaba Cloud WebDAV**" in the "**Services**" column of the LuCI management interface of OpenWrt and click to enter



2) Then start to configure the Alibaba Cloud **WebDAV service** of OpenWrt

a. Check "**Enable**" to start Alibaba Cloud WebDAV

b. Fill in the refresh_token value copied before in the "**Refresh Token**" column

c. Fill in the port number you want to set in the "**Port**" column. The default is **8080**. You need to use the Alibaba Cloud **WebDAV** service to log in to OpenWrt later.

d. Fill in the user name and password you want to set in the "**Username**" and "**Password**" fields. You will need to use the Alibaba Cloud **WebDAV** service to log in to OpenWrt later

e. Finally click "**Save and Apply**"

3) Then you can see that Alibaba Cloud WebDAV is already running

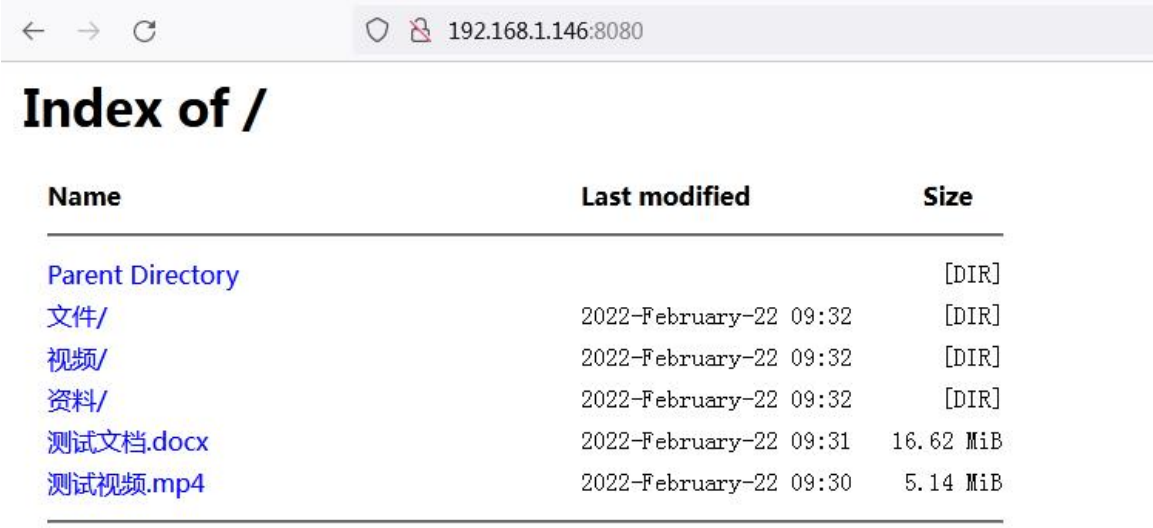
4) Then log in to the main page of the Alibaba Cloud WebDAV service of OpenWrt by means of IP address + port number

a. First, enter the host IP address of the Tab page of Alibaba Cloud Disk **WebDAV** and the port number 8080 in the address bar of the browser.

b. Enter the username and password for configuring the basic information of Alibaba Cloud Disk WebDAV

c. Click "**Login**" to enter the main page of Alibaba Cloud WebDAV

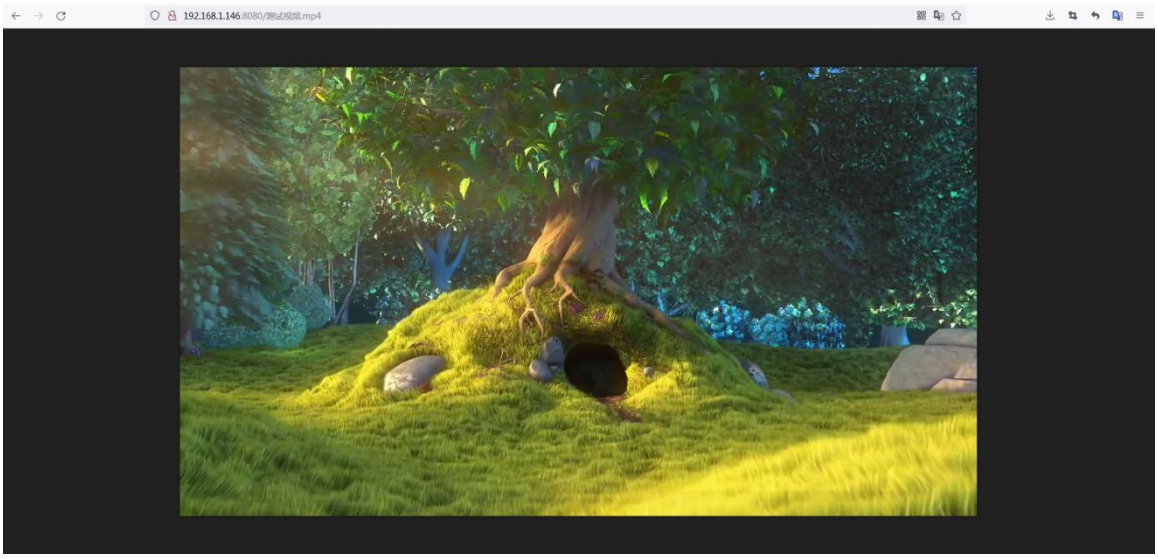
5) Enter the main page of Alibaba Cloud WebDAV



3.15.3. Application of Alibaba Cloud Disk WebDAV

1) Application of Alibaba Cloud Disk WebDAV on Computer Browser

a. Watch the Alibaba Cloud WebDAV video on the browser on the computer, and click "**Test Video.mp4**"



a) Download the Alibaba Cloud WebDAV file on the computer, and click "**Test Document**" to download it.

2) Application of Alibaba Cloud Disk WebDAV on mobile phones

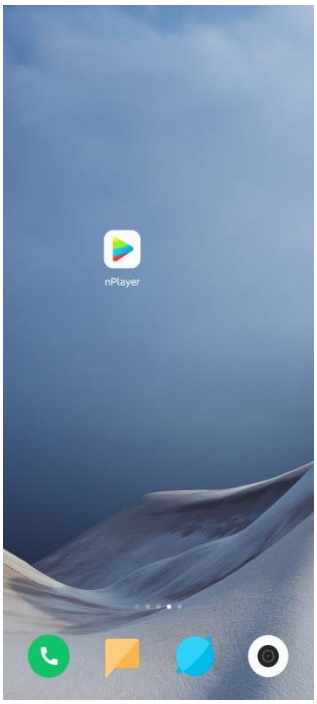
Note: The IP address of the mobile terminal needs to be in the same local area network as the host IP address of Alibaba Cloud Disk WebDAV in order to access the content of Alibaba Cloud Disk WebDAV

Since the testing process for iPhone and Android is the same, the following uses Android as an example to test

a. For iPhone, please download Nplayer from the official store

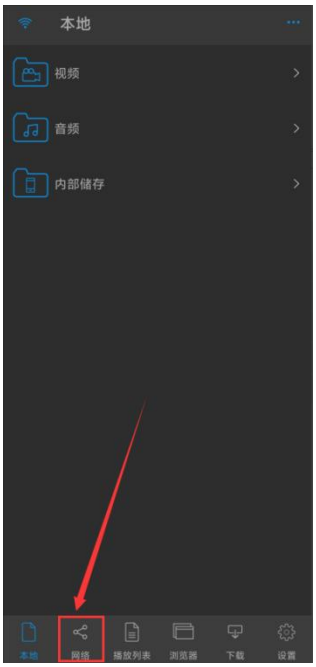
b. Nplayer Android Download Nplayer from Orange Pi's download page

a. Download Nplayer to Android phone

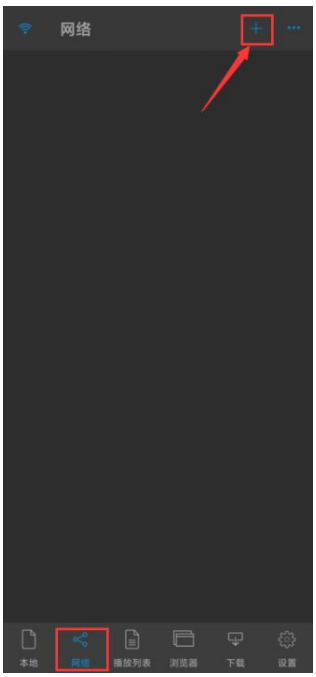


b. Click Nplayer to enter the main interface of Nplayer

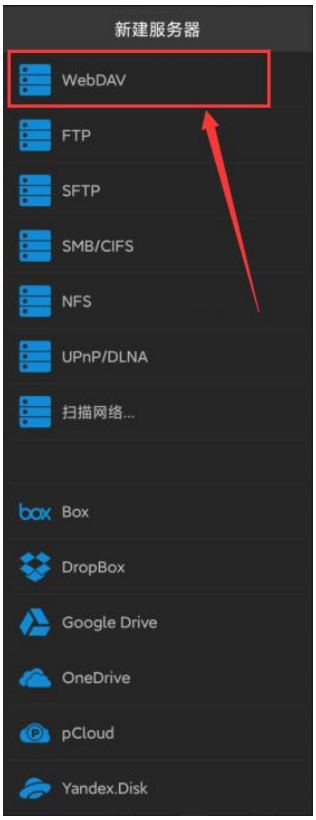
c. Click "**Network**" in the navigation bar below Nplayer to enter the main interface of the network



d. Click "+" in the upper right corner of the network to add a network address



e. In the "+" pop-up box, select "**WebDAV**"



f. Configure the content of the "WebDAV" pop-up box

a) Fill in "Alibaba Cloud" in the title bar of WebDAV

b) The host IP address should be consistent with the host IP address of Alibaba Cloud WebDAV

c) Fill in the username and password consistent with those of Alibaba Cloud WebDAV

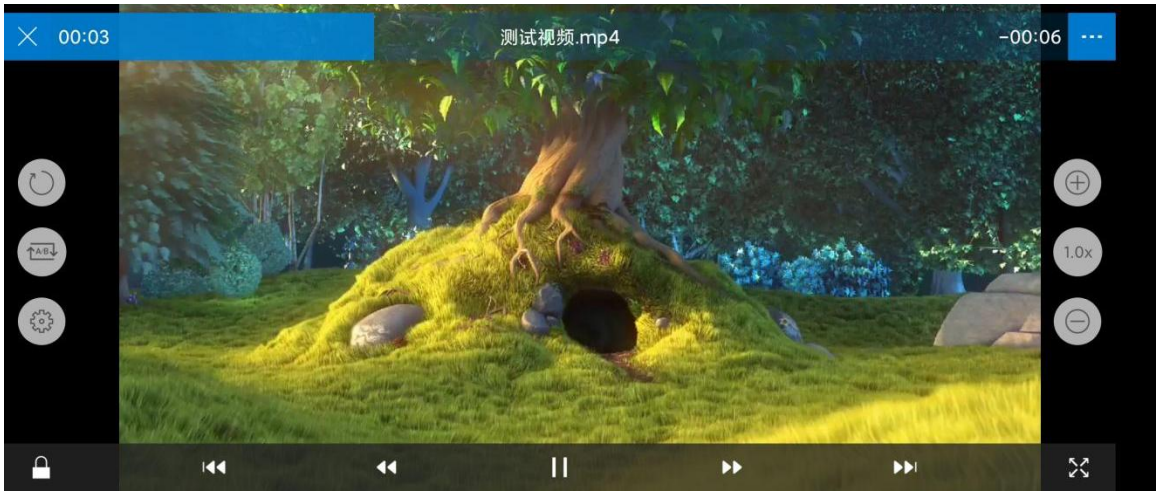
d) Fill in the port number consistent with the port number of Alibaba Cloud WebDAV

e) Click "SAVE" to save and fill in the information

g. Enter the main page of Alibaba Cloud WebDAV connected by Nplayer

h. Test the video playback and file transfer of Alibaba Cloud WebDAV connected under Nplayer

a) Open the test video of Alibaba Cloud WebDAV on the mobile phone and click "Test Video"

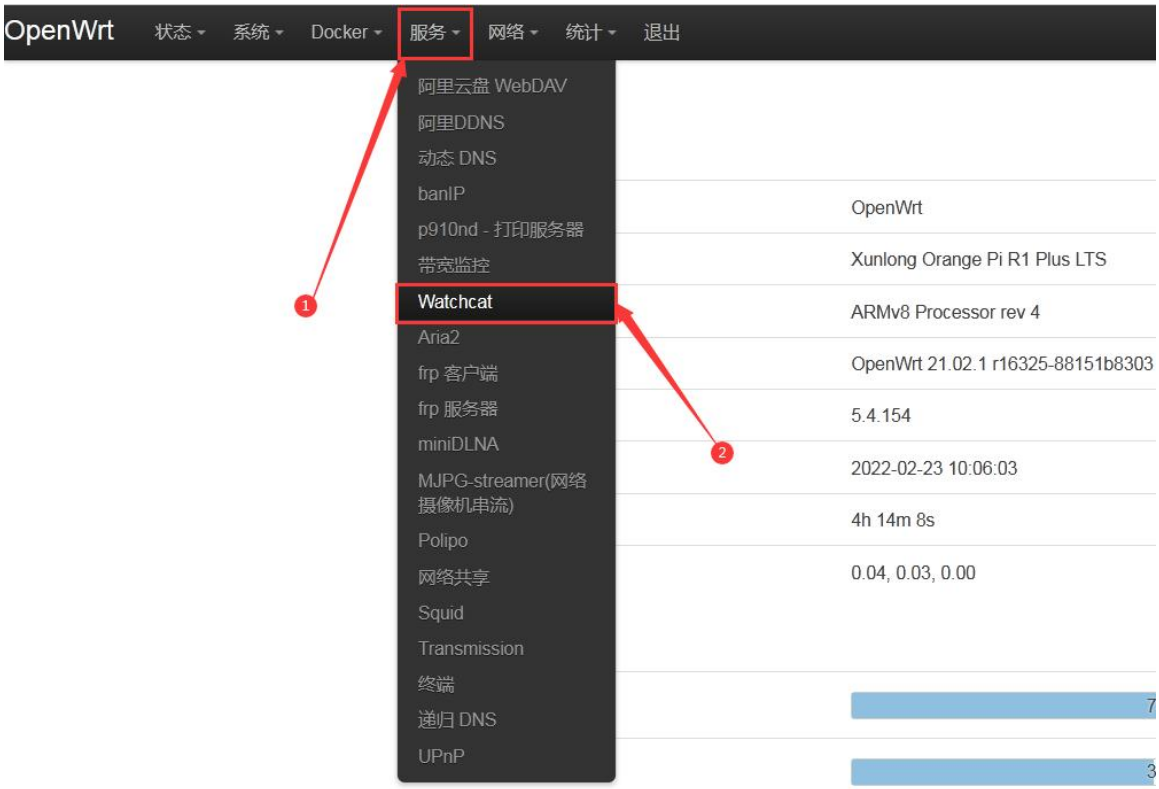


b) In the Alibaba Cloud WebDAV file, click "Test File", the following pop-up box will pop up, which is the default browser of the Android phone, click "Allow"

c) Open the Alibaba Cloud WebDAV test file on the mobile phone

3. 16. Watchcat restart settings

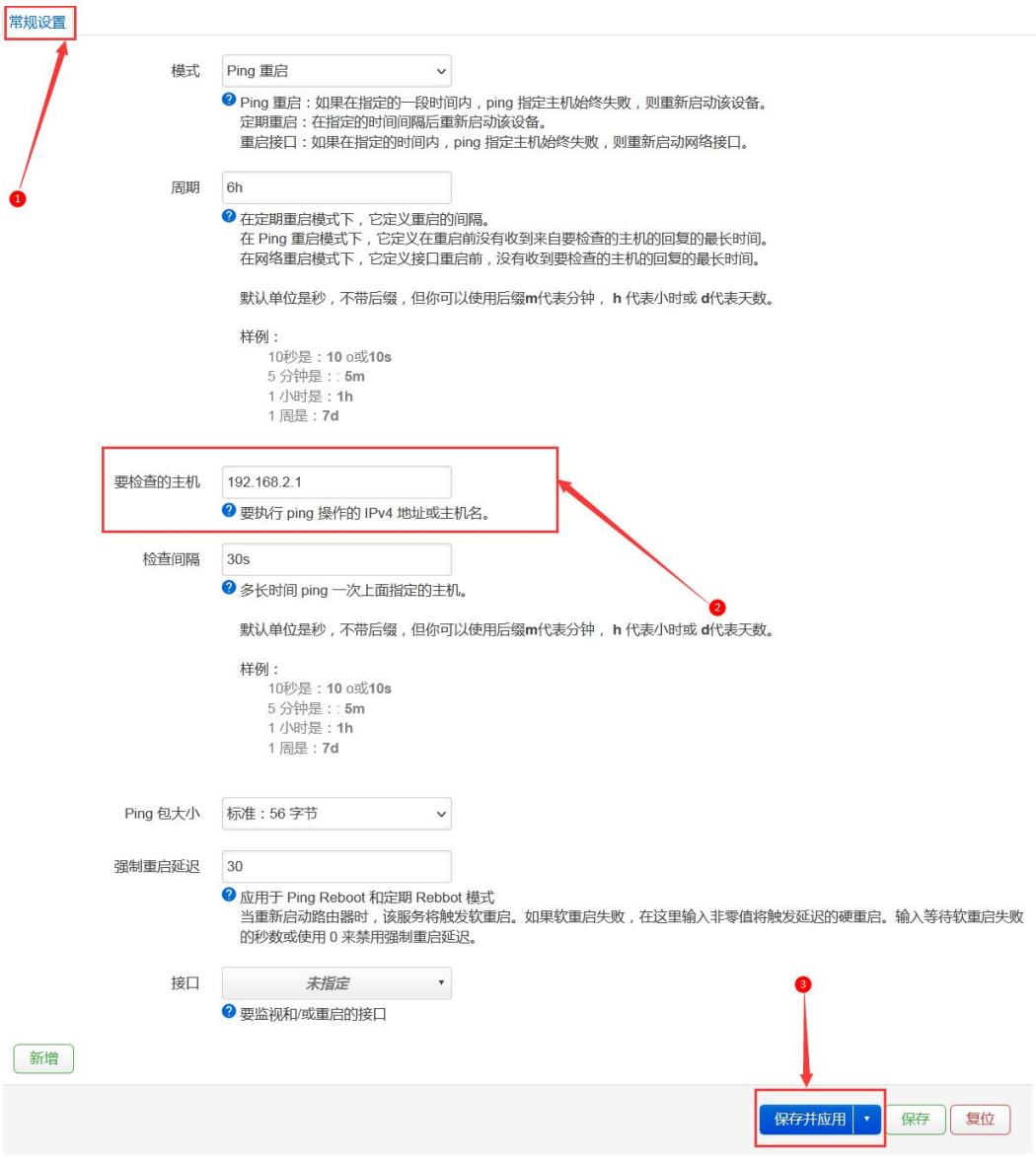
1) Enter the management page of Watchcat restart settings
   a. Find the "**Services**" option in the navigation bar and click to enter
   b. In the vertical bar options below the service, select "**Watchcat**" and click to enter the management page of Watchcat restart settings



2) Ping restart of Watchcat restart settings

The default configuration on the management page of OpenWRT's Watchcat restart settings is Ping restart

   a. On the Tab page of Watchcat, click the "**General Settings**" option and click to enter
   b. Select a different IPv4 address or host name according to the "**host to check**"
   c. Click "**Save and Apply**"



3) Periodic restart of Watchcat restart settings

a. Select "**Reboot Periodically**" under "Mode" in Watchcat's general settings

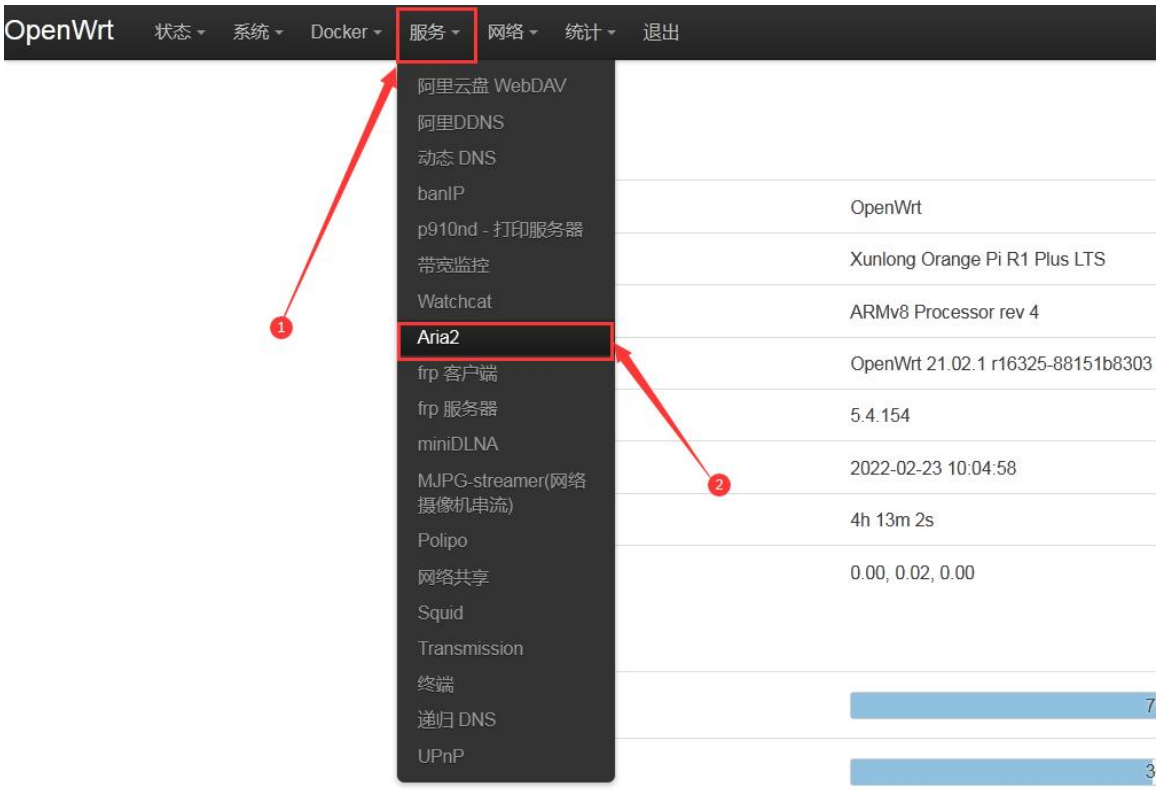
b. According to the regular restart time set by yourself, take "6h" as an example, the specific setting time is set with the following example to set the regular restart time

c. Click Save and Apply to realize OpenWRT restarts every 6 hours on a regular basis

3. 17. Using Aria2

Aria2 is a multi-platform lightweight command line download tool that supports multiple protocols and sources such as HTTP, FTP, BitTorrent, etc. For details, please check the Aria2 official website

1) Enter the management page of Aria2
   a. Find the "**Services**" option in the navigation bar and click to enter
   b. In the vertical column options below the service, select "**Aria2**" and click to enter



2) Configure Aria2-settings

a. In Aria's Tab page select "**Configure**" Aria2-Settings

b. Find "**Basic Options**" in the navigation bar of Aria2-Settings and click to enter

c. Check "**Enabled**" in the basic options to enable Aria2 files

d. Set the download directory to "**/root/aria2**"

e. Then click "**Save and Apply**" to save the settings for Aria2

3) Enter the OpenWRT terminal, create a new "**/root/aria2**" directory, and set the owner of the directory to the aria2 user

```
root@OpenWrt:/# cd /root
root@OpenWrt:~# mkdir aria2
root@OpenWrt:~# chown aria2:aria2 ./aria2
```

4) Select "AriaNg" to download the file

a. Check if Aria2 service is running

b. Select "**AriaNg**" in the installed WEB interface to download the file

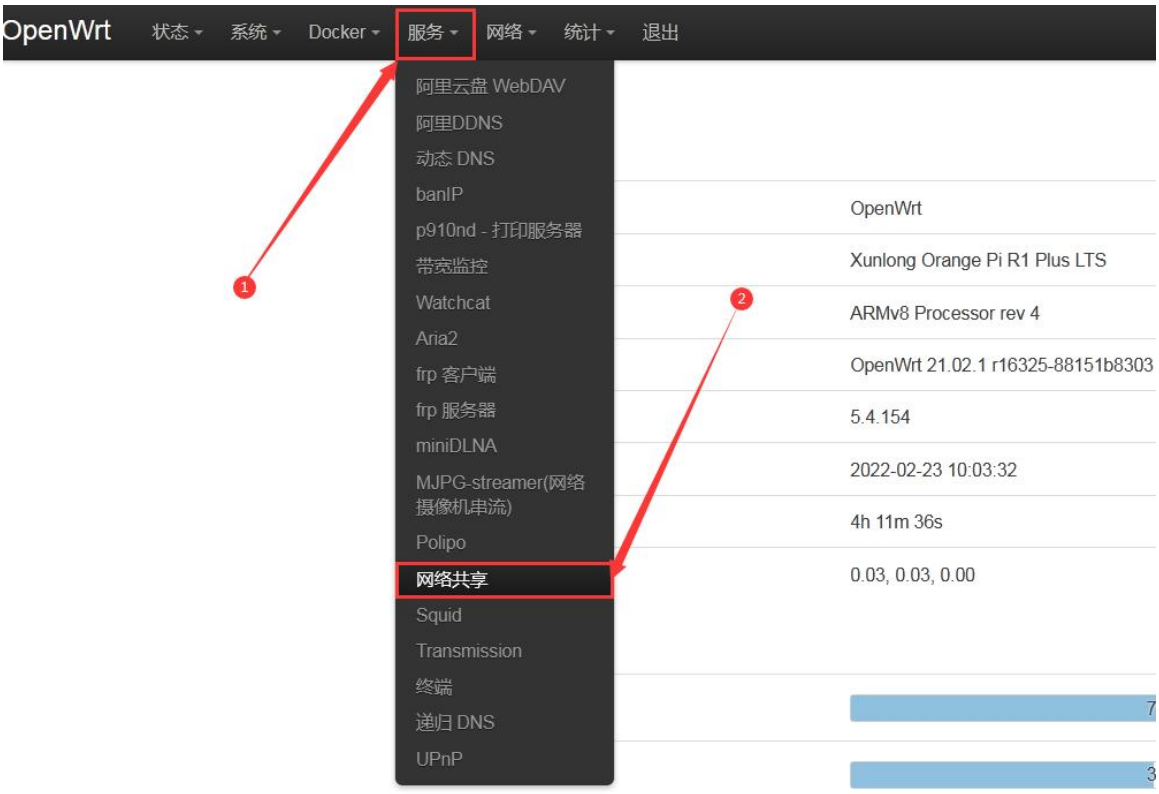
c. Click "**AriaNg**" to enter the download interface of AriaNg

5) Download files to "**/root/aria2**" via Aria2
   a. Select "**+New**" on the "**AriaNg**" interface
   b. Click "**Link**" in the newly created pop-up box, and enter the link to download the file in the link box below to download the file

3. 18. Using Samba Network Shares

There are two main software options for OpenWRT LAN file sharing implementation, Samba and NFS. Samba system has better compatibility, while NFS has better performance. For users who need to use Windows devices, it is recommended to choose Samba

1) Enter the management page of the Samba network share
   a. Find the "**Services**" option in the navigation bar and click to enter
   b. In the vertical column options below the service, select "**Network Sharing**" and click to enter



2) Select the interface that the Samba service needs to monitor

a. Select "**General Settings**" in the navigation bar of network sharing and click to enter

b. The interface is specified according to actual needs. If you want to access through the "wan port", set it to "**wan**"

3) Set the shared directory of the network share

a. In the "**General Settings**" of the network share, "**Shared Directory**" always

click "**Add**" shared directory address

b. Enter the name of the shared folder as "**mmt**" under the name

c. Under the path of the shared directory, choose to set the shared directory location "**/mnt**"

d. Check "**Browsable**" and "**Run as anonymous user**"

e. Click "**Save and Apply**" to save the configuration

4) Windows 10 starts network discovery and sharing

Note: To access Samba under Windows 10 system, you need to confirm whether Windows 10 has enabled network discovery and sharing for sharing. If not, set the following first

a. Enable access to Samba v1/v2

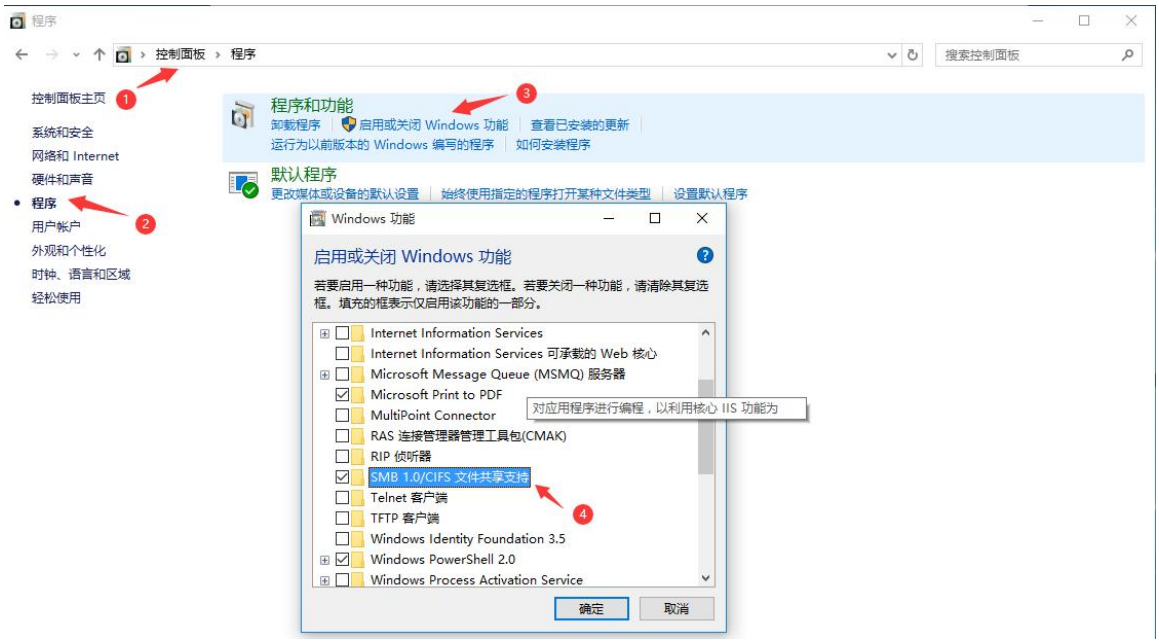
a) Go to the "Control Panel" of Windows 10

b) Click "Programs" in the left navigation bar of the control panel

c) Select "Turn Windows features on or off" in Programs and Features

d) Check "SMB 1.0/CIFS file sharing support" in the pop-up box for enabling or disabling Windows functions

e) Click "OK" to configure the application



b. Turn on network discovery of Windows 10

a) Go to the "Control Panel" of Windows 10

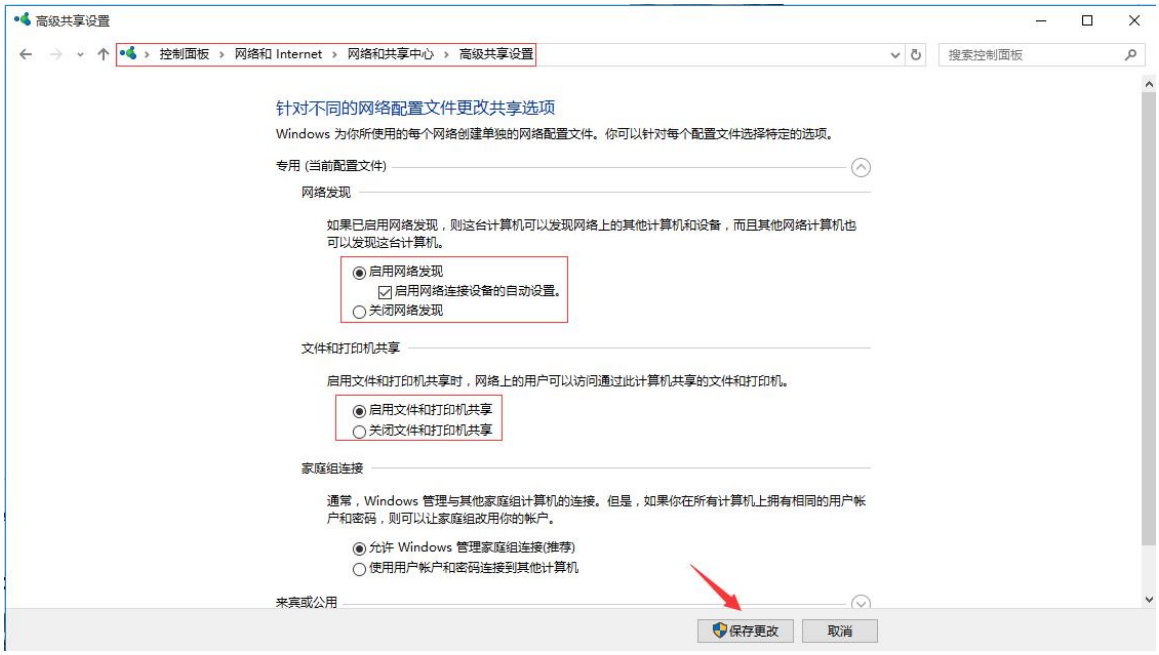
b) Select "Network and Internet" in Control Panel

c) Then open "Network and Sharing Center"

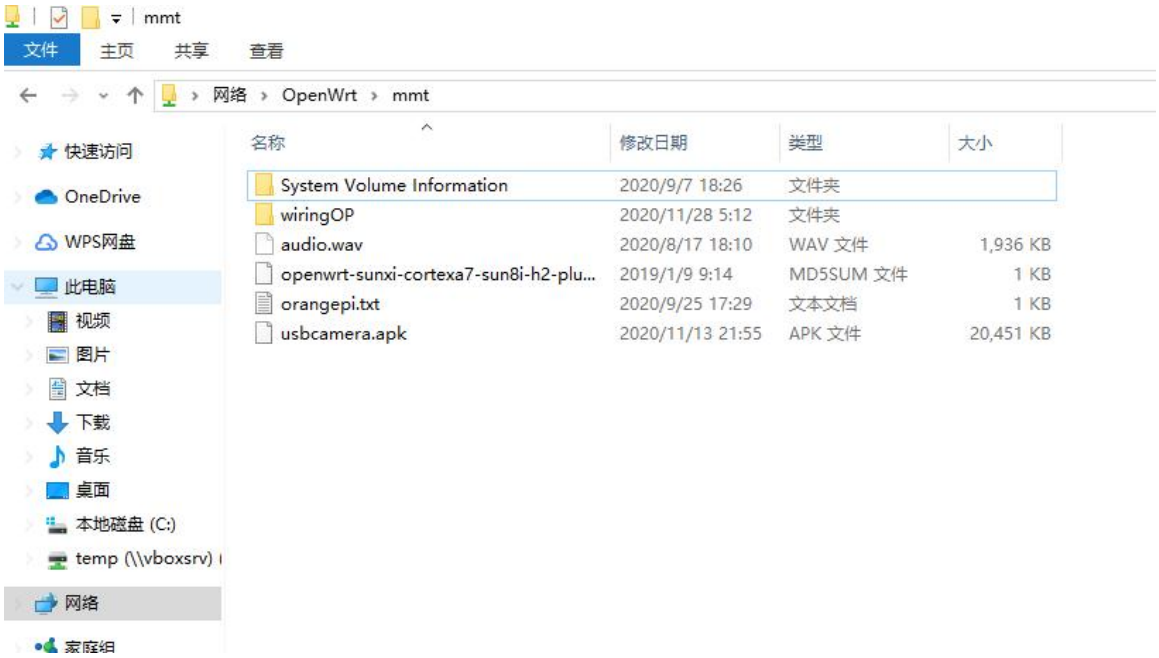
d) Click | "Advanced Sharing Settings"

e) Turn on "**Enable Network Discovery**" and "**Enable File and Printer Sharing**"

f) Click "Save Changes" to save the network discovery configuration for Windows 10



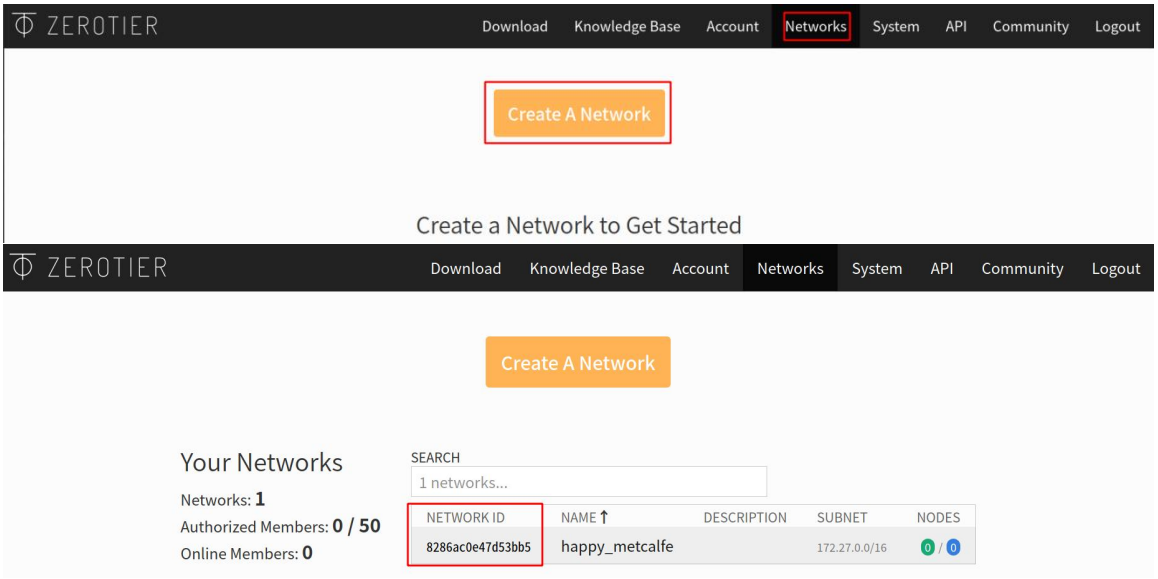
5) After the setting is completed, enter \\OpenWrt in the address bar of the resource manager to access the shared directory, the user name is root, and the password is the password set by the development board host



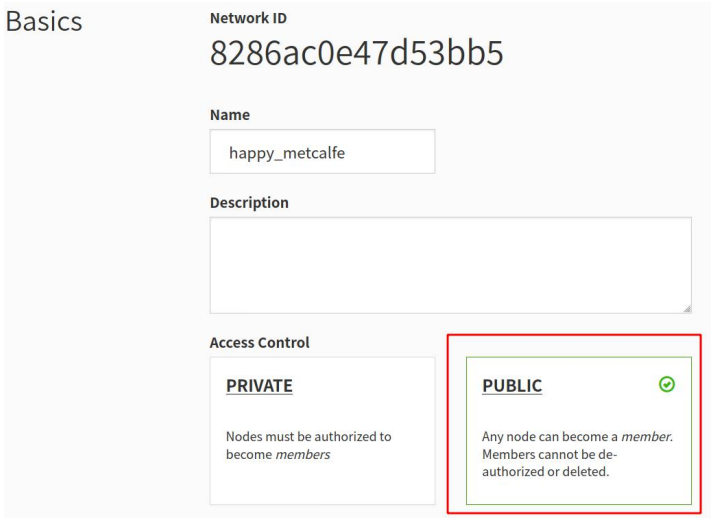
3. 19. Zerotier Instructions for Use

The OpenWRT system has pre-installed the zerotier client. After creating a virtual local area network on the zerotier official website, the client can directly join it through the Network ID. The specific operations are as follows

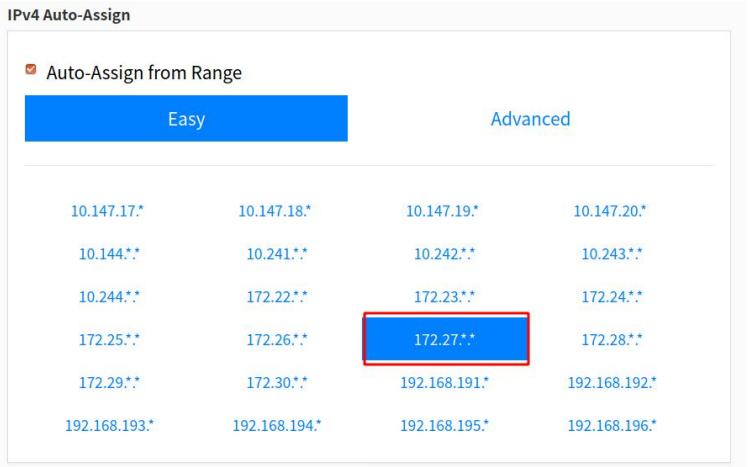
1) Log in to zerotier's official website https://my.zerotier.com/network, click Network->Create A Network after registration and login to create a virtual local area network



2) Click to enter the network console page, you can set the privacy option to public, so that the added network nodes do not need to be verified



3) The following automatically assigns the address. Here you can choose the network segment yourself, here is 172.27.*.*



4) Enter the following command in the OpenWRT terminal to join the virtual local area network created above, **where 8286ac0e47d53bb5 is the Network ID of the virtual local area network created above**

```
root@OpenWrt:/# zerotier-one -d                    #Start the zerotier client
root@OpenWrt:/# zerotier-cli join 8286ac0e47d53bb5  #Join the network
```

5) Enter ifconfig in the terminal, you can see that there is already a new **ztks54inm2** device, the IP address is **172.27.214.213**

```
root@OpenWrt:/# ifconfig
ztks54inm2 Link encap:Ethernet  HWaddr F6:4E:DE:BF:D8:52
          inet addr:172.27.214.213  Bcast:172.27.255.255  Mask:255.255.0.0
          inet6 addr: fe80::e82f:d0ff:fe5a:867e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:2800  Metric:1
          RX packets:18 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1720 (1.6 KiB)  TX byte81 (8.2 KiB)
```

6) Install the zerotier client on another device (here, Ubuntu18.04 is used as an example), and execute the following command to install it. After the installation is complete, you need to restart the computer

```
test@ubuntu:~$ curl -s https://install.zerotier.com | sudo bash
```

7) After restarting, join the virtual local area network according to the Network ID, and you can also see that the ip address assigned by zerotier has been obtained. At this time, the Ubuntu PC and OrangePi R1 Plus LTS are in the same local area network, and they can communicate freely.

```
test@ubuntu:~$ sudo zerotier-cli join 8286ac0e47d53bb5
test@ubuntu:~$ ifconfig
ztks54inm2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2800
        inet 172.27.47.214  netmask 255.255.0.0  broadcast 172.27.255.255
        inet6 fe80::5ce1:85ff:fe2b:6918  prefixlen 64  scopeid 0x20<link>
        ether f6:fd:87:68:12:cf  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 46  bytes 10006 (10.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

8) Test whether the two terminals can communicate

```
root@OpenWrt:/# ping 172.27.47.214 -I ztks54inm2
PING 172.27.47.214 (172.27.47.214): 56 data bytes
64 bytes from 172.27.47.214: seq=0 ttl=64 time=1.209 ms
64 bytes from 172.27.47.214: seq=1 ttl=64 time=1.136 ms
64 bytes from 172.27.47.214: seq=2 ttl=64 time=1.203 ms
64 bytes from 172.27.47.214: seq=3 ttl=64 time=1.235 ms
^C
--- 172.27.47.214 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.136/1.195/1.235 ms
```

9) zerotier other common commands

```
root@OpenWrt:/# zerotier-one -d                    #Start the zerotier client
root@OpenWrt:/# zerotier-cli status                #Get address and service status
root@OpenWrt:/# zerotier-cli join # Network ID     #Join the network
root@OpenWrt:/# zerotier-cli leave # Network ID    #leave the network
root@OpenWrt:/# zerotier-cli listnetworks          #list networks
```

3. 20. View OpenWRT system version information

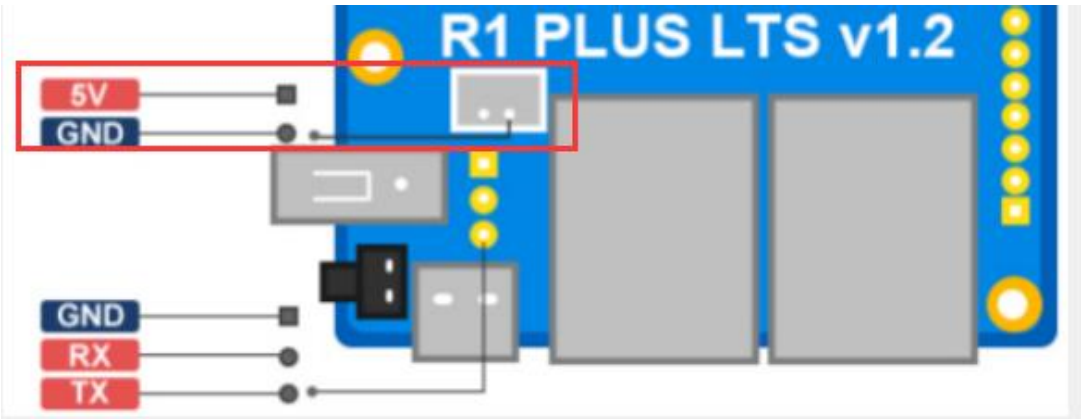
```
root@OpenWrt:~# cat /etc/os-release
NAME="OpenWrt"
VERSION="21.02.1"
ID="openwrt"
ID_LIKE="lede openwrt"
PRETTY_NAME="OpenWrt 21.02.1"
VERSION_ID="21.02.1"
HOME_URL="https://openwrt.org/"
BUG_URL="https://bugs.openwrt.org/"
SUPPORT_URL="https://forum.openwrt.org/"
BUILD_ID="r16325-88151b8303"
OPENWRT_BOARD="rockchip/armv8"
OPENWRT_ARCH="aarch64_generic"
OPENWRT_TAINTS="no-all"
OPENWRT_DEVICE_MANUFACTURER="OpenWrt"
OPENWRT_DEVICE_MANUFACTURER_URL="https://openwrt.org/"
OPENWRT_DEVICE_PRODUCT="Generic"
OPENWRT_DEVICE_REVISION="v0"
OPENWRT_RELEASE="OpenWrt 21.02.1 r16325-88151b8303"
```

3. 21. Use of Pwm fan

1）The description of the tested fan model is shown in the figure below

产品参数			
品牌	Longshengxin	尺寸	30*30*10MM
电压	DC 5V	电流	0.10A
转速	7500±5%RPM	噪音	21DBA
重量	8.1克	功率	0.5W
轴承结构	液压轴承	寿命	40000小时 40℃
温度范围	-10 +60℃	风量	2.3CFM

2）When connecting the fan, pay attention to the position of 5V and GND of the R1 Plus LTS fan interface, do not connect it in reverse



3）When the OpenWrt system starts, it will call the pwm-fan.sh script to control the fan speed. When the CPU temperature is higher than 52 degrees Celsius, the fan will start to work

4. Instructions for use of Linux system

4.1. Supported linux distribution types and kernel versions

release type	kernel version	Server Edition	desktop version
Ubuntu 18.04	Linux5.10	support	not support
Ubuntu 20.04	Linux5.10	support	not support
Debian 10	Linux5.10	support	not support

4.2. Linux5.10 kernel driver adaptation

Function	state
USB2.0	OK
TF card boot	OK
Gigabit Ethernet	OK
USB to Gigabit LAN	OK
USB camera	OK
LED	OK
13pin GPIO	OK
I2C	OK
SPI Nor Flash	OK
UART	OK
reset button	OK

4.3. Linux system default login account and password

account	password
root	orangepi
orangepi	orangepi

4. 4. Start the rootfs in the auto-expanding TF card for the first time

1) When the TF card starts the Linux system for the first time, it will call the orangepi-resize-filesystem script through the systemd service orangepi-resize-filesystem.service to automatically expand the rootfs, **so there is no need to manually expand the capacity.**

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

```
root@orangepir1plus-lts:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            430M     0  430M   0% /dev
tmpfs           100M  5.6M   95M   6% /run
/dev/mmcblk0p1   15G   915M   14G   7% /
tmpfs           500M     0  500M   0% /dev/shm
```

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

4) In addition, if you do not need to automatically expand rootfs, you can use the following method to prohibit

a. First burn the linux image to the TF card, **after burning the image to the TF, remember not to start the linux system**

b. Then insert the TF card into the Ubuntu PC (Windows does not work), the Ubuntu PC will generally automatically mount the partition of the TF card, if the automatic mounting is normal, use the ls command to see the following output, the partition name of the TF card and the following command The names shown are not necessarily the same, please modify according to the actual situation

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

c. Then switch the current user to the root user in the Ubuntu PC

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

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

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

e. Then you can uninstall the TF card, then pull out the TF card and insert it into the development board to start. When the Linux system starts, when it detects that there is a **.no_rootfs_resize** file in the /root directory, the rootfs will no longer be automatically expanded.

f. After prohibiting the automatic expansion of rootfs, you can see that the available capacity of the TF card is only about 168M

```
root@orangepir1plus-lts:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            417M     0  417M   0% /dev
tmpfs            98M  2.9M   96M   3% /run
/dev/mmcblk0p1  1.3G  1.1G  168M  87% /
tmpfs           490M     0  490M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           490M     0  490M   0% /sys/fs/cgroup
tmpfs           490M  4.0K  490M   1% /tmp
/dev/zram0       49M  1.3M   44M   3% /var/log
tmpfs            98M     0   98M   0% /run/user/0
```

4. 5. How to modify the linux log level (loglevel)

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

```
Starting kernel ...
```

```
Uncompressing Linux... done, booting the kernel.

Orange Pi 2.1.4 Focal ttyS0

orangepi login:
```

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

```
root@orangepir1plus-lts:~# sed -i "s/verbosity=1/verbosity=7/" /boot/orangepiEnv.txt
root@orangepir1plus-lts:~# sed -i "s/console=both/console=serial/"
/boot/orangepiEnv.txt
```

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

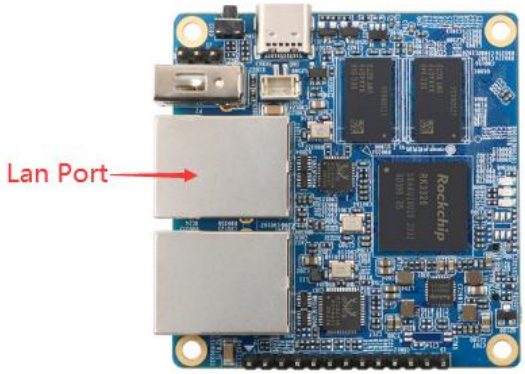
```
root@orangepir1plus-lts:~# cat /boot/orangepiEnv.txt
verbosity=7
console=serial
```

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

4. 6. Ethernet port test

4. 6. 1. Lan port test

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



2) After the system starts, it will automatically assign an IP address to the Ethernet card through DHCP

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

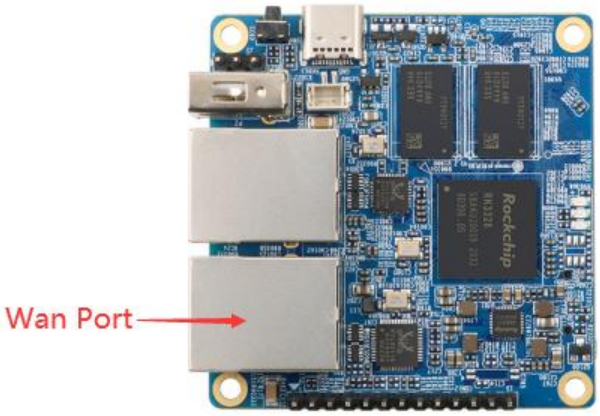
```
root@orangepir1plus-lts:~# ifconfig lan0
lan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.96  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::be68:b89f:def0:5261  prefixlen 64  scopeid 0x20<link>
        ether c0:74:2b:ff:b3:46  txqueuelen 1000  (Ethernet)
        RX packets 5737  bytes 329470 (329.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 57  bytes 5500 (5.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

4) The command to test network connectivity is as follows

```
root@orangepir1plus-lts:~# ping www.baidu.com -I lan0
PING www.a.shifen.com (180.101.49.42) from 192.168.1.96 lan0: 56(84) bytes of data.
64 bytes from 180.101.49.42 (180.101.49.42): icmp_seq=1 ttl=53 time=27.2 ms
64 bytes from 180.101.49.42 (180.101.49.42): icmp_seq=2 ttl=53 time=26.0 ms
64 bytes from 180.101.49.42 (180.101.49.42): icmp_seq=3 ttl=53 time=25.8 ms
64 bytes from 180.101.49.42 (180.101.49.42): icmp_seq=4 ttl=53 time=26.7 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 25.839/26.478/27.254/0.583 ms
```

4.6.2. Wan port test

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



2) After the system starts, it will automatically assign an IP address to the Ethernet card through DHCP

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

```
root@orangepir1plus-lts:~# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.62  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::9db4:d13e:5e66:ee6a  prefixlen 64  scopeid 0x20<link>
        ether 66:2e:7d:b9:f7:74  txqueuelen 1000  (Ethernet)
        RX packets 2055  bytes 150800 (150.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 145  bytes 13816 (13.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 29
```

4) The command to test network connectivity is as follows

```
root@orangepir1plus-lts:~# ping www.baidu.com -I eth0
PING www.a.shifen.com (14.215.177.39) from 192.168.1.62 eth0: 56(84) bytes of data.
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=56 time=6.51 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=56 time=6.45 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=56 time=6.44 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=56 time=6.59 ms
^C
--- www.a.shifen.com ping statistics ---
```

```
13 packets transmitted, 12 received, 7% packet loss, time 12032ms
rtt min/avg/max/mdev = 6.252/6.596/7.067/0.226 ms
```

4. 7. SSH remote login development board

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

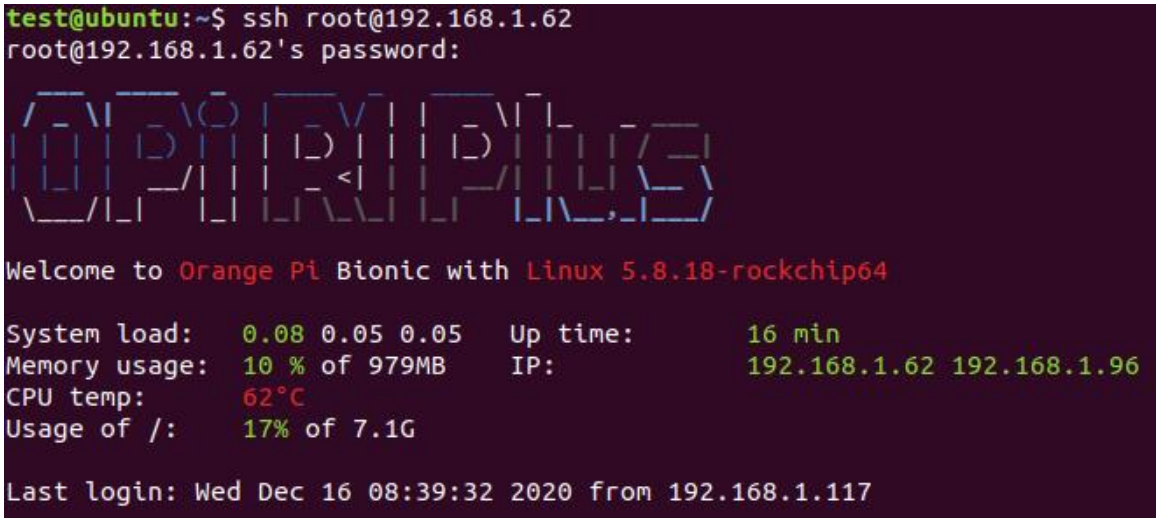
4. 7. 1. SSH remote login development board under Ubuntu

1) Obtain the IP address of the development board

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

```
test@test:~$ ssh root@192.168.1.62          (It needs to be replaced with the IP address of the development board)
root@192.168.1.62's password:          （Enter the password here, the default password is orangepi）
```

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



4) If the following error is prompted during ssh login

```
test@test:~$ ssh root@192.168.1.62
Connection reset by 192.168.1.62 port 22
```

5) You can enter the following command on the development board and try again to see if it can be connected

```
root@orangepir1plus-lts:~# rm /etc/ssh/ssh_host_*
root@orangepir1plus-lts:~# dpkg-reconfigure openssh-server
```

4. 7. 2. SSH remote login development board under Windows

1) First get the IP address of the development board

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

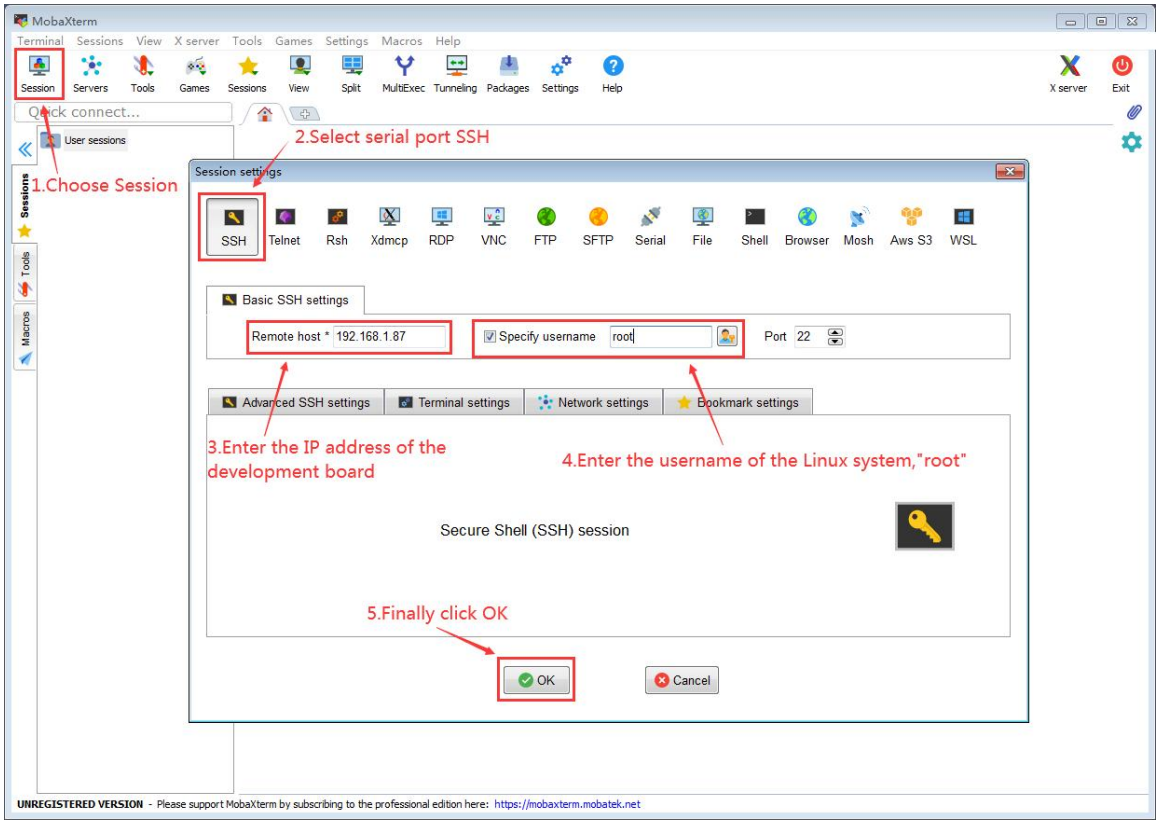
a. Open **Session**

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

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

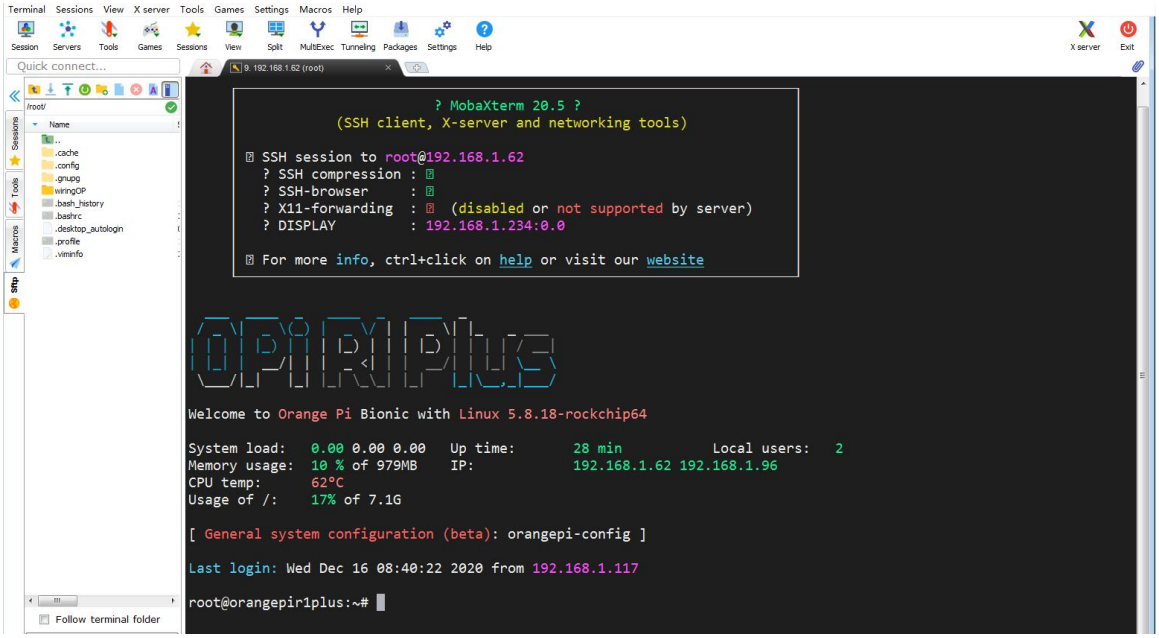
d. Then enter the username **root** or **orangepi** of the linux system in **Specify username**

e. Finally click **OK**



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

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



4. 8. On-board LED light test description

1) There are three LED lights on the development board, two yellow lights and one red light. The default display of the LED lights when the system starts is as follows

Power status light (red light)	The system starts, the red light flashes
Wan port status light (yellow light)	When the network cable is connected to the Wan port, the yellow light is always on. When the network cable is unplugged from the Wan port, the yellow light is off.
Lan port status light (yellow light)	When the LAN port is connected to the network cable, the yellow light is always on. When the network cable is unplugged from the Lan port, the yellow light is off.

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

a. First enter the setting directory of the red light

```
root@orangepir1plus-lts:~# cd /sys/class/leds/status_led
```

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

```
root@orangepir1plus-lts:/sys/class/leds/status_led# echo 0 > brightness
```

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

```
root@orangepir1plus-lts:/sys/class/leds/status_led# echo 1 > brightness
```

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

```
root@orangepir1plus-lts:/sys/class/leds/status_led# echo none > trigger
```

4. 9. USB interface test

4. 9. 1. Connect USB storage device test

1) First insert the U disk into the USB port of the Orange Pi development board

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

```
root@orangepir1plus-lts:~# cat /proc/partitions | grep "sd*"
major minor  #blocks  name
   8        0   30044160 sda
   8        1   30043119 sda1
```

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

```
root@orangepir1plus-lts:~# mount /dev/sda1 /mnt/
root@orangepir1plus-lts:~# ls /mnt/
test.txt
```

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

```
root@orangepir1plus-lts:~# df -h | grep "sd"
/dev/sda1       29G  208K   29G   1% /mnt
```

4. 10. USB wireless network card test

The available USB wireless network cards tested by the linux5.10 system are as follows. For other types of USB wireless network cards, please test them yourself. If they cannot be used, you need to transplant the corresponding USB wireless network card driver.

serial number	model
1	RTL8723BU
2	RTL8821CU

4. 10. 1. RTL8723BU test

1) First insert the RTL8723BU wireless network card module into the USB interface of the development board

2) Then the linux system will automatically load the RTL8723BU related kernel modules, you can see the following output through the lsmod command

```
root@orangepir1plus-lts:~# lsmod | grep "rtl8"
rtl8xxxu                      126976  0
mac80211                      925696  1 rtl8xxxu
```

3) You can see the loading information of the RTL8723BU module through the dmesg command

```
root@orangepir1plus-lts:~# dmesg | tail
[ 3128.895618] usb 3-1: 1d8: ff ff ff ff ff ff ff ff
[ 3128.895633] usb 3-1: 1e0: ff ff ff ff ff ff ff ff
[ 3128.895648] usb 3-1: 1e8: ff ff ff ff ff ff ff ff
[ 3128.895663] usb 3-1: 1f0: ff ff ff ff ff ff ff ff
[ 3128.895679] usb 3-1: 1f8: ff ff ff ff ff ff ff ff
[ 3128.895703] usb 3-1: RTL8723BU rev E (SMIC) 1T1R, TX queues 3, WiFi=1, BT=1, GPS=0, HI PA=0
[ 3128.895720] usb 3-1: RTL8723BU MAC: 00:13:ef:f4:58:ae
[ 3128.895737] usb 3-1: rtl8xxxu: Loading firmware rtlwifi/rtl8723bu_nic.bin
[ 3128.896110] usb 3-1: Firmware revision 35.0 (signature 0x5301)
[ 3132.191050] rtl8xxxu 3-1:1.2 wlx0013eff458ae: renamed from wlan0
```

4) Then you can see the device node of RTL8723BU WIFI through the ifconfig command. For the connection and test method of WIFI, please refer to the WIFI

connection test section

```
root@orangepir1plus-lts:~# ifconfig wlx0013eff458ae
wlx0013eff458ae: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:13:ef:f4:58:ae  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5) Then you can see a Bluetooth device through the hciconfig command. The node whose Bus type is USB is the Bluetooth node of RTL8723BU. For the Bluetooth test method, please refer to the Bluetooth test section

```
root@orangepir1plus-lts:~# hciconfig
hci0:   Type: Primary  Bus: USB
        BD Address: 00:13:EF:F4:58:AE  ACL MTU: 1021:8  SCO MTU: 255:16
        UP RUNNING
        RX bytes:1631 acl:0 sco:0 events:144 errors:0
        TX bytes:26662 acl:0 sco:0 commands:144 errors:0
```

4. 10. 2. RTL8821CU test

1) First insert the RTL8821CU wireless network card module into the USB interface of the development board

2) Then the linux system will automatically load the RTL8821CU-related kernel modules, and you can see the following output through the lsmod command

```
root@orangepir1plus-lts:~# lsmod | grep "8821"
8821cu                  2043904  0
cfg80211                 897024  3 8821cu,mac80211,rtl8xxx
```

3) You can see the loading information of the RTL8821CU module through the dmesg command

```
root@orangepir1plus-lts:~# dmesg | tail
[ 3987.552017] usb 2-1: Product: 802.11ac NIC
[ 3987.552032] usb 2-1: Manufacturer: Realtek
[ 3987.552046] usb 2-1: SerialNumber: 123456
[ 3987.560377] Bluetooth: hci0: RTL: examining hci_ver=08 hci_rev=000c lmp_ver=08
```

```
lmp_subver=8821
[ 3987.561349] Bluetooth: hci0: RTL: rom_version status=0 version=1
[ 3987.561370] Bluetooth: hci0: RTL: loading rtl_bt/rtl8821c_fw.bin
[ 3987.561818] Bluetooth: hci0: RTL: loading rtl_bt/rtl8821c_config.bin
[ 3987.562148] Bluetooth: hci0: RTL: cfg_sz 10, total sz 21678
[ 3987.974248] Bluetooth: hci0: RTL: fw version 0x826ca99e
[ 3987.998204] rtl8821cu 2-1:1.2 wlxd0c0bf8742cd: renamed from wlan0
```

4) Then you can see the device node of RTL8821CU WIFI through the ifconfig command. For the connection and test method of WIFI, please refer to the section on WIFI connection test

```
root@orangepir1plus-lts:~# ifconfig wlxd0c0bf8742cd
wlxd0c0bf8742cd: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d0:c0:bf:87:42:cd  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5) Then you can see the Bluetooth device node through the hciconfig command. The node whose Bus type is USB is the Bluetooth node of the RTL8821CU. For the Bluetooth test method, please refer to the Bluetooth test section

```
root@orangepir1plus-lts:~# hciconfig
hci0:   Type: Primary  Bus: USB
        BD Address: D0:C0:BF:87:42:CE  ACL MTU: 1021:8  SCO MTU: 255:12
        UP RUNNING
        RX bytes:1350 acl:0 sco:0 events:138 errors:0
        TX bytes:24230 acl:0 sco:0 commands:138 errors:0
```

4. 11. USB wireless network card WIFI connection test

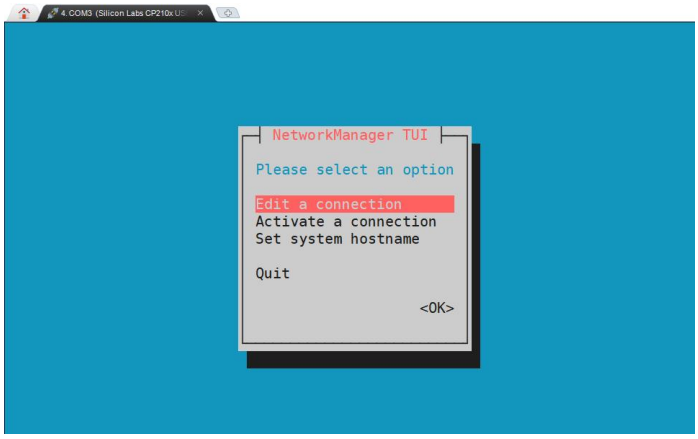
4. 11. 1. Test method of Linux system

1) First log in to the linux system using the serial port terminal (please use MobaXterm for serial port software, the graphical interface cannot be displayed using minicom)

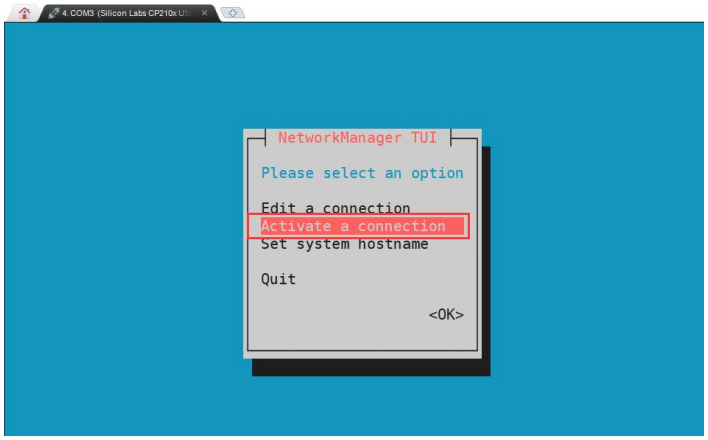
2) Then enter nmtui in the command line to open the wifi connection interface

```
root@orangepi:~# nmtui
```

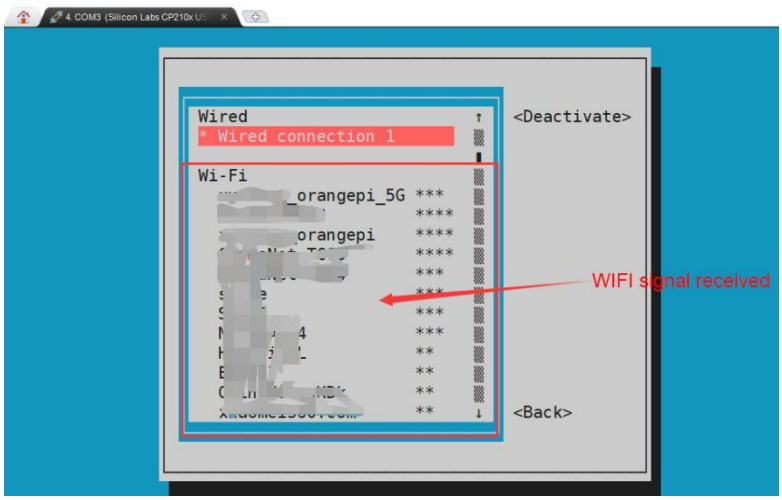
3) Enter nmtui to open the interface as shown below



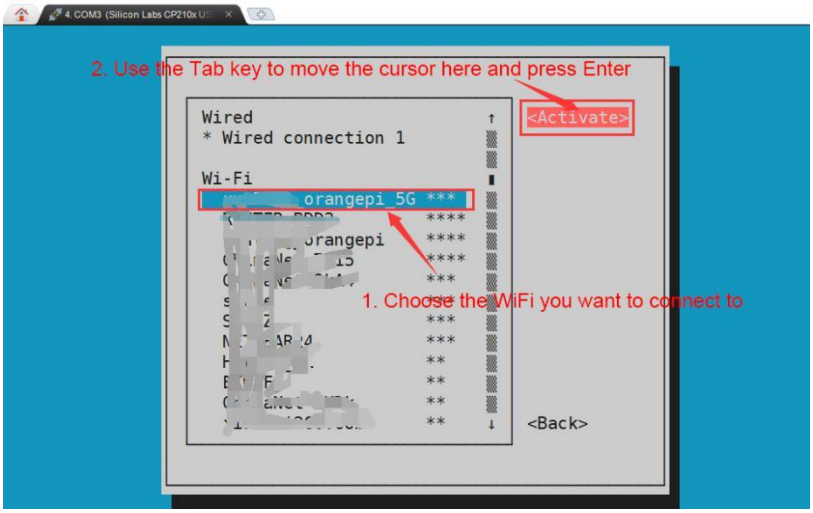
4) Select **Activate a connect** and press Enter



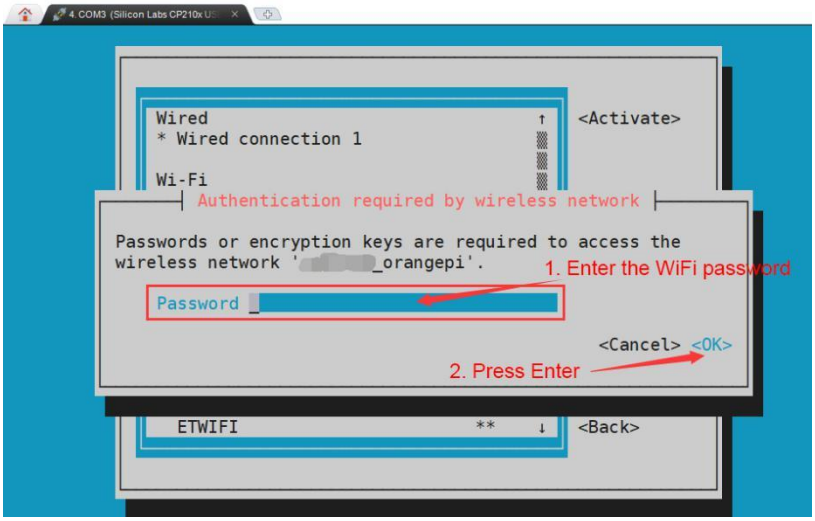
5) Then you can see all the searched WIFI hotspots



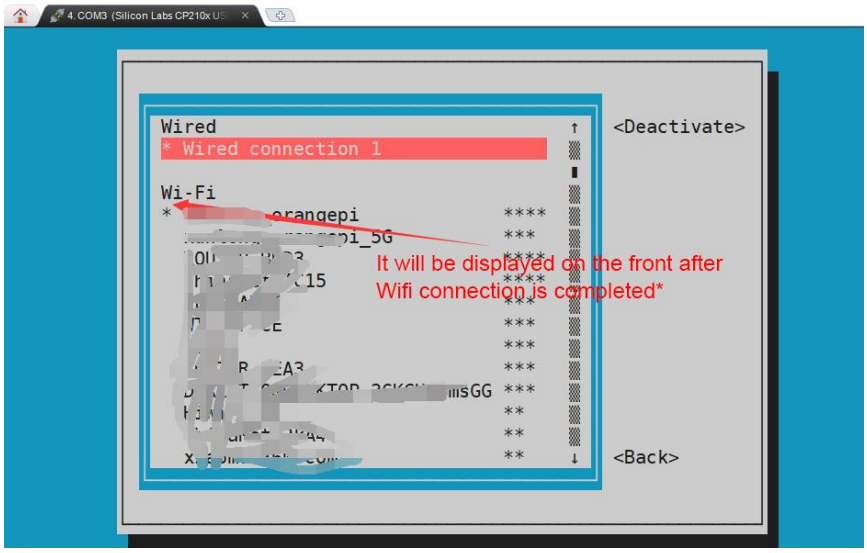
6) Select the WIFI hotspot you want to connect, then use the Tab key to position the cursor to **Activate** and press Enter



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



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



9) You can view the IP address of the wifi through the ifconfig command

```
root@orangepir1plus-lts:~# ifconfig wlxd0c0bf8742cd
wlxd0c0bf8742cd: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.198  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::db3a:3ea6:b98d:3fc9  prefixlen 64  scopeid 0x20<link>
        ether d0:c0:bf:87:42:cd  txqueuelen 1000  (Ethernet)
        RX packets 528  bytes 91350 (91.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
```

```
        TX packets 26  bytes 3802 (3.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

10) Use the ping command to test the connectivity of the wifi network

```
root@orangepir1plus-lts:~# ping www.baidu.com -I wlxd0c0bf8742cd
PING www.a.shifen.com (14.215.177.39) from 192.168.1.198 wlxd0c0bf8742cd: 56(84) bytes of data.
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=56 time=7.39 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=56 time=7.72 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=56 time=8.79 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=56 time=10.2 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=56 time=8.83 ms
^C
--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 7.391/8.607/10.296/1.021 ms
```

4.12. Bluetooth test of USB wireless network card

4.12.1. Test method of Linux system

1) The bluez tool is required to use the bluetooth connection. Use the following command to install the bluez tool

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt install bluez
```

2) First, check the Bluetooth device information through hciconfig -a. The following information appears indicating that the Bluetooth initialization is normal.

```
root@orangepir1plus-lts:~# hciconfig -a
hci0:   Type: Primary  Bus: USB
        BD Address: 00:13:EF:F4:58:AE  ACL MTU: 1021:8  SCO MTU: 255:16
        UP RUNNING PSCAN
        RX bytes:15875 acl:28 sco:0 events:367 errors:0
        TX bytes:28282 acl:28 sco:0 commands:232 errors:0
        Features: 0xff 0xff 0xff 0xfa 0xdb 0xfd 0x7b 0x87
```

```
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
        Name: 'orangepir1plus-lts'
        Class: 0x000000
        Service Classes: Unspecified
        Device Class: Miscellaneous,
        HCI Version: 4.0 (0x6)  Revision: 0x1e4c
        LMP Version: 4.0 (0x6)  Subversion: 0xc3ff
        Manufacturer: Realtek Semiconductor Corporation (93)
```

3) Use bluetoothctl to turn on bluetooth to scan for surrounding devices

```
root@orangepir1plus-lts:~# bluetoothctl
[NEW] Controller 00:13:EF:F4:58:AE orangepir1plus-lts [default]
Agent registered
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# discoverable on
Changing discoverable on succeeded
[CHG] Controller 00:13:EF:F4:58:AE Discoverable: yes
[bluetooth]# pairable on
Changing pairable on succeeded
[bluetooth]# scan on
[NEW] Device 6A:31:DF:62:08:78 6A-31-DF-62-08-78
[NEW] Device 56:73:C1:98:C6:63 56-73-C1-98-C6-63
[NEW] Device 9C:2E:A1:42:71:11 Xiaomi phone
[NEW] Device 44:F2:1B:B8:76:7B Roy
```

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

```
[bluetooth]# pair 44:F2:1B:B8:76:7B
Attempting to pair with 44:F2:1B:B8:76:7B
[CHG] Device 44:F2:1B:B8:76:7B  Connected: yes
Request confirmation
[Roy]1m[agent] Confirm passkey 996955  (yes/no): yes
```

```
[CHG] Device DC:72:9B:4C:F4:CF Paired: yes
Pairing successful
[bluetooth]# paired-devices
Device 44:F2:1B:B8:76:7B Roy
```

4. 13. USB camera test

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

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

```
root@orangepir1plus-lts:~# lsmod | grep "uvc"
uvcvideo                110592  0
videobuf2_vmalloc        20480  1 uvcvideo
videobuf2_v4l2           36864  1 uvcvideo
videobuf2_common         65536  2 videobuf2_v4l2,uvcvideo
videodev                311296  3 videobuf2_v4l2,uvcvideo,videobuf2_common
mc                      65536  4
videodev,videobuf2_v4l2,uvcvideo,videobuf2_common
```

3) Through the v4l2-ctl (**note that the l in v4l2 is a lowercase letter l, not the number 1**) command, you can see that the device node information of the USB camera is /dev/video0

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt install v4l-utils
root@orangepir1plus-lts:~# v4l2-ctl    --list-devices
USB 2.0 Camera: HD USB Camera (usb-ff5d0000.usb-1):
    /dev/video0
    /dev/video1
```

4) Use fswebcam to test the USB camera

a. Install fswebcam

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt-get install fswebcam
```

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

a) -d option is used to specify the device node of the USB camera
b) --no-banner is used to remove the watermark of the photo
c) -r option is used to specify the resolution of the photo
d) -S option is set to skip previous frames
e) ./image.jpg is used to set the name and path of the generated photo

```
root@orangepir1plus-lts:~# fswebcam -d /dev/video0 --no-banner -r 1280x720 -S 5 ./image.jpg
```

c. In the server version of the Linux system, after taking the photo, you can use the scp command to transfer the captured image to the Ubuntu PC for image viewing

```
root@orangepir1plus-lts:~# scp image.jpg test@192.168.1.55:/home/test （Modify the IP address and path according to the actual situation）
```

5) Use motion to test the USB camera

a. Install the camera test software motion

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt install motion
```

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

```
root@orangepir1plus-lts:~# sed -i \
"s/start_motion_daemon=no/start_motion_daemon=yes/" \
/etc/default/motion (this is a command)
```

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

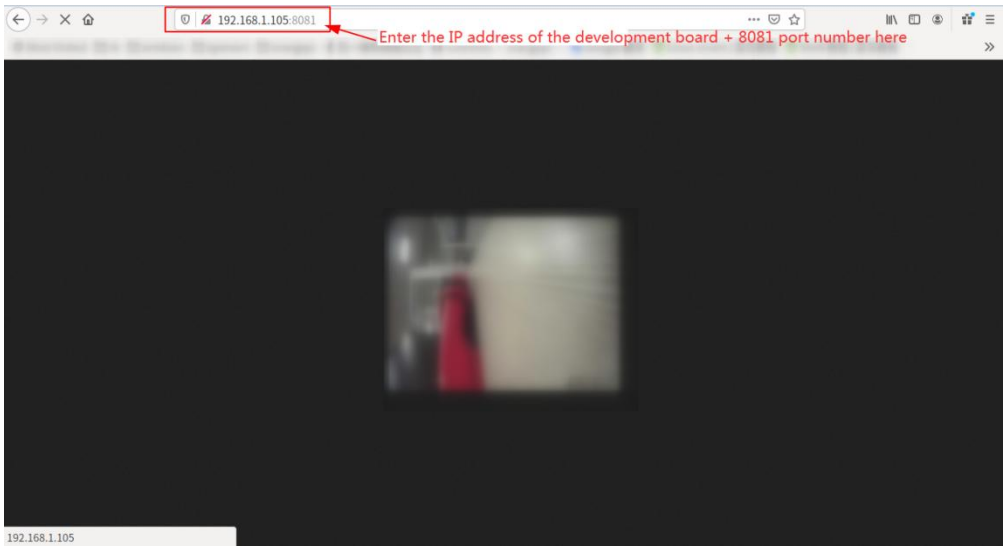
```
root@orangepir1plus-lts:~# sed -i "s/stream_localhost on/stream_localhost off/" \
/etc/motion/motion.conf (this is a command)
```

d. Then restart the motion service

```
root@orangepir1plus-lts:~# /etc/init.d/motion restart
[ ok ] Restarting motion (via systemctl): motion.service.
```

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

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



4. 14. Temperature sensor

1) RK3328 has a total of 1 temperature sensor, the command to check the temperature is as follows

```
root@orangepir1plus-lts:~# cat /sys/class/thermal/thermal_zone0/type
soc-thermal
root@orangepir1plus-lts:~# cat /sys/class/thermal/thermal_zone0/temp
61664
```

4. 15. How to install Docker

1) This method is not applicable on debian10 system

2) Uninstall the old version of docker that may exist first

```
root@orangepir1plus-lts:~# apt remove docker docker-engine docker-ce docker.io
```

3) Then install the following packages

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt install -y apt-transport-https ca-certificates curl
software-properties-common (this is a command)
```

4) Add the key of Alibaba Cloud docker

```
root@orangepir1plus-lts:~# curl -fsSL
http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg
  | sudo apt-key add  -          (this is a command)
```

5) Add the corresponding docker source to the system source of ubuntu

```
root@orangepir1plus-lts:~# add-apt-repository "deb [arch=arm64]
 https://mirrors.aliyun.com/docker-ce/linux/ubuntu $(lsb_release -cs) stable"  (this
  is a command)
```

6) Install the latest version of docker-ce

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt install docker-ce
```

7) Verify the status of docker

```
root@orangepir1plus-lts:~# systemctl status docker
● docker.service - Docker Application Container Engine
    Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
    Active: active (running) since Mon 2020-08-24 10:29:22 UTC; 26min ago
      Docs: https://docs.docker.com
 Main PID: 3145 (dockerd)
     Tasks: 15
    CGroup: /system.slice/docker.service
            └─3145 /usr/bin/dockerd -H fd://
--containerd=/run/containerd/containerd.sock
```

8) Test docker

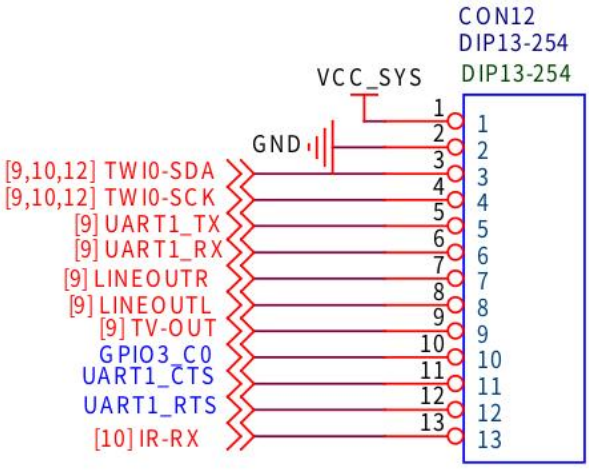
```
root@orangepir1plus-lts:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
256ab8fe8778: Pull complete
Digest:
sha256:7f0a9f93b4aa3022c3a4c147a449ef11e0941a1fd0bf4a8e6c9408b2600777c5
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
```

This message shows that your installation appears to be working correctly.

4. 16. 13 Pin Adapter Board Interface Pin Description

1) Please refer to the following figure for the order of the 13 pin interface pins of the Orange Pi R1 Plus LTS development board

2) The schematic diagram of the 13pin interface of the Orange Pi R1 Plus LTS development board is shown below



3) The function description of the interface pins of the 13 pin adapter board of the Orange Pi R1 Plus LTS development board is as follows

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

a) TV-OUT audio and video output

b) Infrared receiving function

c) After connecting the adapter board, the 3, 4, 5, 6, 10, 11 and 12 pins of the 13pin interface cannot be used.

d) **In addition, it should be noted that the MIC and 2*USB on the 13pin adapter board cannot be used on the Orange Pi R1 Plus LTS**

b. When the 13 pin interface of the Orange Pi R1 Plus LTS development board is not connected to the adapter board, pins 3, 4, 5, 6, 10, 11, 12 and 13 can be used as ordinary GPIO

pin	Function	GPIO	GPIO serial number
1	VCC_SYS		
2	GND		
3	TWI0-SDA	GPIO2_D1	89
4	TWI0-SCK	GPIO2_D0	88
5	UART1_TX	GPIO3_A4	100
6	UART1_RX	GPIO3_A6	102
7	LINEOUTR		
8	LINEOUTL		
9	TV-OUT		
10	GPIO3_C0	GPIO3_C0	112
11	UATR1_CTS	GPIO3_A7	103
12	UART1_RTS	GPIO3_A5	101
13	IR-RX	GPIO2_A2	66

4. 17. How to install wiringOP

1) Download the code of wiringOP

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt install git
root@orangepir1plus-lts:~# git clone https://github.com/orangepi-xunlong/wiringOP
```

2) Compile wiringOP

```
root@orangepir1plus-lts:~# cd wiringOP
root@orangepir1plus-lts:~/wiringOP# ./build clean
root@orangepir1plus-lts:~/wiringOP# ./build
```

3) The output of the test gpio readall command is as follows, in which the physical pins 1 to 13 are in one-to-one correspondence with the 13 Pin pins on the development board, and pins 7, 8, 9, and 14 cannot be used on WiringOP. Use please ignore

```
root@orangepir1plus-lts:~# gpio readall
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     | 5V       |      |   |  1 || 2  |   |      | GND      |     |      |
 |   89 |   0 | SDA.0    |   IN | 1 |  3 || 4  | 1 | IN   | SCK.0    | 1   | 88   |
 |  100 |   2 | TXD.1    |   IN | 1 |  5 || 6  | 1 | IN   | RXD.1    | 3   | 102  |
 |      |     |          |      |   |  7 || 8  |   |      |          |     |      |
 |      |     |          |      |   |  9 || 10 | 1 | ALT3 | GPIO3_C0 | 4   | 112  |
 |  103 |   5 | CTS.1    |   IN | 1 | 11 || 12 | 1 | IN   | RTS.1    | 6   | 101  |
 |   66 |   7 | GPIO2_A2 |   IN | 1 | 13 || 14 |   |      |          |     |      |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
root@orangepir1plus-lts:~#
```

4. 18. 13pin interface GPIO, I2C, UART test

wiringOP has been adapted to the Orange Pi R1 Plus LTS development board. Using wiringOP can test the functions of GPIO, I2C and UART

4. 18. 1. 13pin GPIO port test

1) The following is an example of how to set the high and low levels of the GPIO port by taking the No. 5 pin—the corresponding GPIO is GPIO3_A4—the corresponding wPi serial number is 2

```
root@orangepir1plus-lts:~# gpio readall
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     | 5V       |      |   |  1 || 2  |   |      | GND      |     |      |
 |   89 |   0 | SDA.0    |   IN | 1 |  3 || 4  | 1 | IN   | SCK.0    | 1   | 88   |
 |  100 |   2 | TXD.1    |   IN | 1 |  5 || 6  | 1 | IN   | RXD.1    | 3   | 102  |
 |      |     |          |      |   |  7 || 8  |   |      |          |     |      |
 |      |     |          |      |   |  9 || 10 | 1 | ALT3 | GPIO3_C0 | 4   | 112  |
 |  103 |   5 | CTS.1    |   IN | 1 | 11 || 12 | 1 | IN   | RTS.1    | 6   | 101  |
 |   66 |   7 | GPIO2_A2 |   IN | 1 | 13 || 14 |   |      |          |     |      |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
```

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

```
root@orangepir1plus-lts:~/wiringOP# gpio mode 2 out
```

Use gpio readall to see that the mode of pin 5 has changed to out

```
root@orangepir1plus-lts:~# gpio readall
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     | 5V       |      |   |  1 || 2  |   |      | GND      |     |      |
 |   89 |   0 | SDA.0    |   IN | 1 |  3 || 4  | 1 | IN   | SCK.0    | 1   | 88   |
 |  100 |   2 | TXD.1    |  OUT | 0 |  5 || 6  | 1 | IN   | RXD.1    | 3   | 102  |
 |      |     |          |      |   |  7 || 8  |   |      |          |     |      |
 |      |     |          |      |   |  9 || 10 | 1 | ALT3 | GPIO3_C0 | 4   | 112  |
 |  103 |   5 | CTS.1    |   IN | 1 | 11 || 12 | 1 | IN   | RTS.1    | 6   | 101  |
 |   66 |   7 | GPIO2_A2 |   IN | 1 | 13 || 14 |   |      |          |     |      |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
```

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

```
root@orangepir1plus-lts:~/wiringOP# gpio write 2 0
```

Use gpio readall to see that the value of pin 5 (V) has become 0

```
root@orangepir1plus-lts:~# gpio readall
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     | 5V       |      |   |  1 || 2  |   |      | GND      |     |      |
 |   89 |   0 | SDA.0    |   IN | 1 |  3 || 4  | 1 | IN   | SCK.0    | 1   | 88   |
 |  100 |   2 | TXD.1    |  OUT | 0 |  5 || 6  | 1 | IN   | RXD.1    | 3   | 102  |
 |      |     |          |      |   |  7 || 8  |   |      |          |     |      |
 |      |     |          |      |   |  9 || 10 | 1 | ALT3 | GPIO3_C0 | 4   | 112  |
 |  103 |   5 | CTS.1    |   IN | 1 | 11 || 12 | 1 | IN   | RTS.1    | 6   | 101  |
 |   66 |   7 | GPIO2_A2 |   IN | 1 | 13 || 14 |   |      |          |     |      |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
```

4) Then set the GPIO port to output a high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is 3.3v, it means that the high level is successfully set.

```
root@orangepir1plus-lts:~/wiringOP# gpio write 2 1
```

Use gpio readall to see that the value (V) of pin 5 has changed to 1

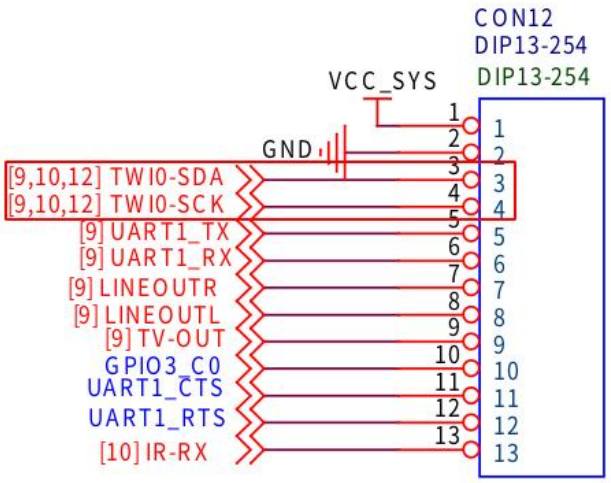
```
root@orangepir1plus-lts:~# gpio readall
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     | 5V       |      |   |  1 || 2  |   |      | GND      |     |      |
 |   89 |   0 | SDA.0    |   IN | 1 |  3 || 4  | 1 | IN   | SCK.0    | 1   | 88   |
 |  100 |   2 | TXD.1    |  OUT | 1 |  5 || 6  | 1 | IN   | RXD.1    | 3   | 102  |
 |      |     |          |      |   |  7 || 8  |   |      |          |     |      |
 |      |     |          |      |   |  9 || 10 | 1 | ALT3 | GPIO3_C0 | 4   | 112  |
 |  103 |   5 | CTS.1    |   IN | 1 | 11 || 12 | 1 | IN   | RTS.1    | 6   | 101  |
 |   66 |   7 | GPIO2_A2 |   IN | 1 | 13 || 14 |   |      |          |     |      |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | GPIO |
 +------+-----+----------+------+---+ R1 Plus  +---+---+--+----------+-----+------+
```

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

4. 18. 2. 13pin I2C test

1) The i2c controller in 13pin is disabled by default in the linux5.10 system in dts. If you need to use the i2c, you need to open the i2c configuration first. The way to open the i2c in the linux5.10 system is as follows:

a. From the schematic diagram of 13pin, the i2c available for the development board is i2c0



b. Then set overlays=i2c0 in /boot/orangepiEnv.txt to open the i2c0 configuration

```
overlays=i2c0
```

c. Then restart the system. When starting, you can see the configuration output of I2C DT overlays in the startup log of u-boot

```
Applying kernel provided DT overlay rockchip-i2c0.dtbo
2698 bytes read in 8 ms (329.1 KiB/s)
```

```
Applying kernel provided DT fixup script (rockchip-fixup.scr)
```

d. After the system is started, if there are more i2c device nodes under /dev, it means the configuration is correct

```
root@orangepir1plus-lts:~# ls /dev/i2c*
/dev/i2c-0  /dev/i2c-1
```

e. The corresponding relationship of different i2c device nodes is as follows, where

a) i2c0 in 13pin corresponds to /dev/i2c-0

```
root@orangepir1plus-lts:~# ls /sys/class/i2c-adapter/i2c-* -lh
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/i2c-adapter/i2c-0 -> ../../devices/platform/ff150000.i2c/i2c-0
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/i2c-adapter/i2c-1 -> ../../devices/platform/ff160000.i2c/i2c-1
root@orangepir1plus-lts:~#
```

2) Then start testing i2c, first install i2c-tools

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt install i2c-tools
```

3) Then connect an i2c device to the i2c0 pin of the 13pin header

	i2c0
sda pin	Corresponds to pin 3
Sck pin	Corresponds to pin 4
Vcc pin	Corresponds to pin 1
Gnd pin	Corresponds to pin 2

4) Then use the i2cdetect -y 0 command. If the address of the connected i2c device can be detected, it means that the i2c can be used normally

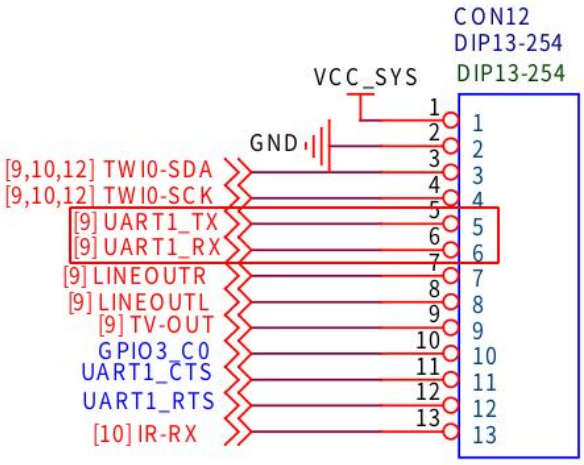
```
root@orangepir1plus:~# i2cdetect -y 0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- 38 -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- 51 -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

4. 18. 3. 13pin UART test

1) The uart controller in 13pin is disabled by default in the linux5.10 system in the dts. If

you need to use the uart, you need to open the uart configuration first. The way to open the uart in the linux5.10 system is as follows:

a. From the schematic diagram of 13pin, the uart available for the development board is uart1



b. Then set overlays=uart1 in /boot/orangepiEnv.txt to open the configuration of uart1

```
overlays=uart1
```

c. Then restart the system. During startup, you can see the configuration output of UART-related DT overlays in the startup log of u-boot

```
Applying kernel provided DT overlay rockchip-uart1.dtbo
2698 bytes read in 8 ms (329.1 KiB/s)
Applying kernel provided DT fixup script (rockchip-fixup.scr)
```

d. After the system starts, you can see the information of ttyS1 under /sys/class/tty, and uart1 in 13pin corresponds to /dev/ttyS1

```
root@orangepir1plus-lts:~# ls /sys/class/tty/ttyS* -lh
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS0 -> ../../devices/platform/serial8250/tty/ttyS0
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS1 -> ../../devices/platform/ff120000.serial/tty/ttyS1
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS2 -> ../../devices/platform/ff130000.serial/tty/ttyS2
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS3 -> ../../devices/platform/serial8250/tty/ttyS3
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS4 -> ../../devices/platform/serial8250/tty/ttyS4
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS5 -> ../../devices/platform/serial8250/tty/ttyS5
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS6 -> ../../devices/platform/serial8250/tty/ttyS6
lrwxrwxrwx 1 root root 0 Nov 25 10:28 /sys/class/tty/ttyS7 -> ../../devices/platform/serial8250/tty/ttyS7
```

2) Then start to test the uart interface, **first use the DuPont line to short-circuit the rx and tx of the uart1 interface to be tested**

	uart1
tx pin	Corresponds to pin 5
rx pin	Corresponds to pin 6

3) Then modify the serial device node name opened by the serial test program serialTest in wiringOP to **/dev/ttyS1**

```
root@orangepir1plus-lts:~/wiringOP/examples# vim serialTest.c
```

```
int main ()
{
  int fd ;
  int count ;
  unsigned int nextTime ;

  if ((fd = serialOpen ("/dev/ttyS1", 115200)) < 0)
  {
    fprintf (stderr, "Unable to open serial device: %s\n", strerror (errno)) ;
    return 1 ;
  }
```

4) Recompile the serial test program serialTest in wiringOP

```
root@orangepir1plus-lts:~/wiringOP/examples# make serialTest
[CC] serialTest.c
[link]
root@orangepir1plus-lts:~/wiringOP/examples#
```

5) Finally run serialTest, if you can see the following print, it means the serial communication is normal

```
root@orangepir1plus-lts:~/wiringOP/examples# ./serialTest

Out:    0:  ->    0
Out:    1:  ->    1
Out:    2:  ->    2
Out:    3:  ->    3
Out:    4:  ->    4
Out:    5:  ->    5
Out:    6:  ->    6
Out:    7:  ->    7
Out:    8:  ->    8^C
```

4. 19. Hardware watchdog test

1) Download the code of wiringOP

```
root@orangepir1plus-lts:~# apt update
root@orangepir1plus-lts:~# apt -y install git
root@orangepir1plus-lts:~# git clone https://github.com/orangepi-xunlong/wiringOP
```

2) Compile the watchdog test program

```
root@orangepir1plus-lts:~# cd wiringOP/examples/
root@orangepir1plus-lts:~/wiringOP/examples# gcc watchdog.c -o watchdog
```

3) Run the watchdog test program

a. The second parameter 10 represents the counting time of the watchdog. If the dog is not fed within this time, the system will restart

b. We can feed the dog by pressing any key on the keyboard (except ESC), after feeding the dog, the program will print a line of keep alive to indicate that the dog was fed successfully

```
root@orangepir1plus-lts:~/wiringOP/examples# ./watchdog 10
open success
options is 33664,identity is Synopsys DesignWare Watchdog
put_usr return,if 0,success:0
The old reset time is: 30
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
```

4. 20. The method of outputting the kernel print information to the 13pin serial port

The kernel console outputs to ttyS2 by default, which is the 3pin debug serial port on the development board. We can also set the kernel console output to be redirected to UART1 in the 13pin interface as follows:

1) The linux5.10 system needs to open the configuration of uart1 first. For the detailed

configuration method, see the UART test of 13pin

2) Then modify console=ttyS2 in /boot/boot.cmd to console=ttyS1

```
root@orangepir1plus-lts:~# vim /boot/boot.cmd
```

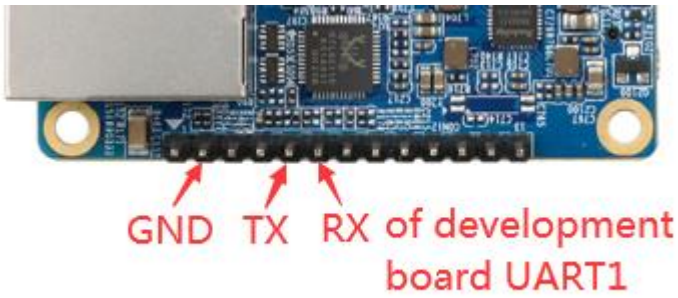
```
if test "${console}" = "display" || test "${console}" = "both"; then setenv consoleargs "console=ttyS1,1500000 console=tty1"; fi
if test "${console}" = "serial" || test "${console}" = "both"; then setenv consoleargs "console=ttyS1,1500000 ${consoleargs}"; fi
if test "${earlycon}" = "on"; then setenv consoleargs "earlycon ${consoleargs}"; fi
if test "${bootlogo}" = "true"; then setenv consoleargs "bootsplash.bootfile=bootsplash.orangepi ${consoleargs}"; fi
```

3) Then recompile /boot/boot.cmd to /boot/boot.scr (operate in the linux system of the development board)

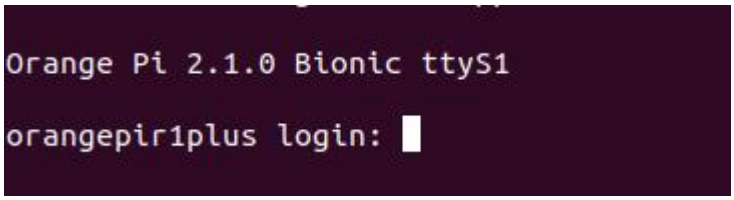
```
root@orangepir1plus-lts:~# mkimage -C none -A arm -T script -d /boot/boot.cmd /boot/boot.scr
Image Name:
Created:          Tue Dec  8 02:35:43 2020
Image Type:       ARM Linux Script (uncompressed)
Data Size:        2448 Bytes = 2.39 KiB = 0.00 MiB
Load Address: 00000000
Entry Point:      00000000
Contents:
   Image 0: 2440 Bytes = 2.38 KiB = 0.00 MiB
```

4) Then connect the USB to TTL module to the UART1 pin of the 13pin interface through the DuPont line

a. The GND of the USB to TTL module is connected to the GND of the 13pin interface of the development board
b. The RX of the USB to TTL module is connected to the TX of the UART1 of the development board
c. The TX of the USB to TTL module is connected to the RX of the UART1 of the development board



5) Then restart the development board, you can see that the kernel console is output to ttyS1 by default. Note that the output log of u-boot is still output to ttyS2 at this time, and will not be output to ttyS1



4. 21. Check the serial number of the RK3328 chip

1) The command to view the serial number of the RK3328 chip is as follows, the serial number of each chip is different, so the serial number can be used to distinguish multiple development boards

```
root@orangepir1plus-lts:~# cat /sys/devices/platform/board/info
Hardware        : ORANGEPI-R1PLUS
Revision        : 0002
Serial          : 9b25e2e5a704467c

Model           : OrangePi R1PLUS LTS
Manufacturer    : Shenzhen Xunlong Software CO.,Limited
```

4. 22. How to restart the system

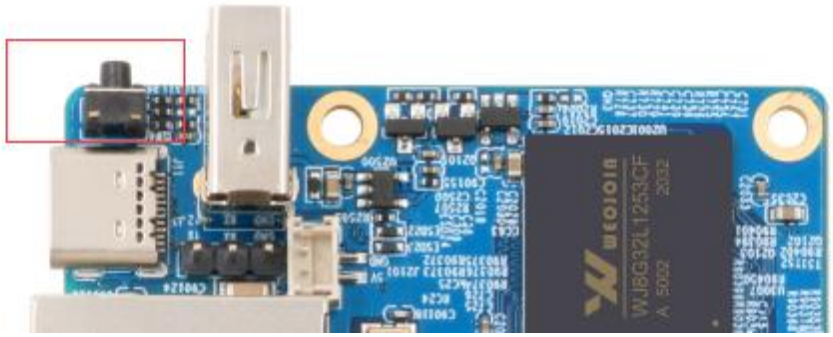
1) Restart using the reboot command

```
root@orangepir1plus-lts:~# reboot
```

2) Use the **poweroff** command to shut down, if you want to start, you need to re-plug the power

```
root@orangepir1plus-lts:~# poweroff
```

3) You can short press the reset button on the development board to restart the system



5. Instructions for using OpenWRT SDK

5.1. Download the source code of OpenWRT SDK

5.1.1. Download OpenWRT from github

1) There are currently two branches of the openwrt code on github. The openwrt-21.02 branch is adapted based on the official stable version of openwrt, and the function tends to be stable. The master branch is adapted based on the official snapshot version of openwrt and is the version under development. Please download the corresponding branch as required

2) Download the openwrt-21.02 branch code

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install git
test@test:~$
git clone https://github.com/orangepi-xunlong/openwrt.git -b openwrt-21.02
```

3) Download the master branch code

```
test@test:~$ git clone https://github.com/orangepi-xunlong/openwrt.git -b master
```

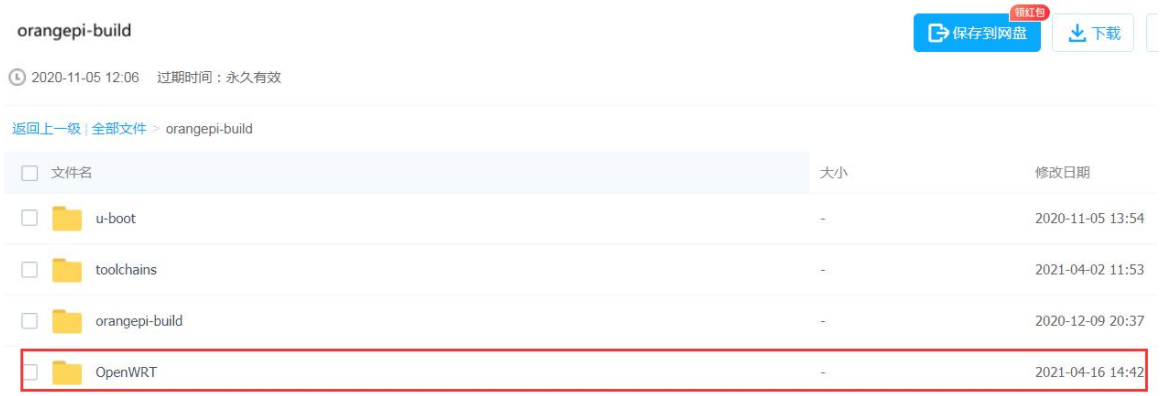
4) After the OpenWRT code is downloaded, the following files and folders will be included

```
test@test:~/openwrt$ ls
BSDmakefile  Config.in  include  Makefile  README.md  scripts  toolchain
Config  feeds.conf.default  LICENSE  package  rules.mk  target  tools
```

5.1.2. Download OpenWRT from Google Drive

1) First download the compressed package of OpenWRT from Google Drive, the download link of Google Drive is

```
Link:
https://drive.google.com/drive/folders/1YsHaZR0imZRlpuFETdOPDnOQmKat74wzExtr
```



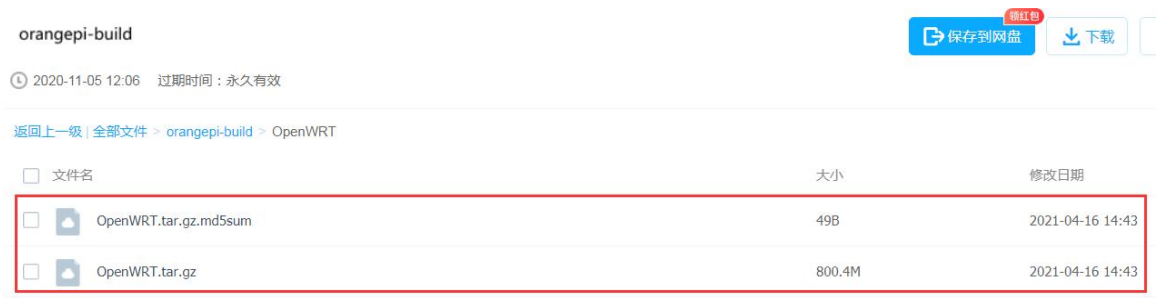
2) There are two files in the OpenWRT folder of Google Drive

a. **OpenWRT.tar.gz** is the compressed package of openwrt source code

b. **OpenWRT.tar.gz.md5sum** is the MD5 checksum file of the compressed package of the openwrt source code

c. After downloading, please first check whether the MD5 checksum of the OpenWRT.tar.gz compressed package is correct, which can prevent problems with the downloaded compressed package. If it is not correct, please download it again and check whether the checksum is correct. The command is

```
test@test:~$ md5sum -c OpenWRT.tar.gz.md5sum
OpenWRT.tar.gz: success
```



3) Then you can use the tar -zxf command to decompress OpenWRT.tar.gz

```
test@test:~$ tar -zxf OpenWRT.tar.gz
test@test:~$ cd openwrt/
test@test:~/openwrt$ ls
BSDmakefile Config.in feeds include Makefile README.md scripts target toolchain config dl feeds.conf.default LICENSE package rules.mk staging_dir tools
```

4) Before compiling the openwrt source code, please synchronize the openwrt and github servers to ensure that the code is in the latest state

```
test@test:~/openwrt$ git pull
```

5. 2. Compiling OpenWRT

5. 2. 1. Compile OpenWRT source code

1) Install the following dependencies first (the following are required to compile on Ubuntu18.04 only tested so far, if compiling on other versions of the system, please install the dependencies according to the error message)

```
test@test:~/openwrt$ sudo apt update
test@test:~/openwrt$ sudo apt install make libncurses5-dev g++ gcc gawk
```

2) Then execute ./scripts/feeds update -a and ./scripts/feeds install -a to download dependency packages

```
test@test:~/openwrt$ ./scripts/feeds update -a
test@test:~/openwrt$ ./scripts/feeds install -a
```

Note: In this step and the subsequent make compilation, many software packages from foreign sources will be downloaded. Because of network problems, it is very likely that

the download will fail and cause compilation errors. Therefore, it is recommended to use the source package of Google Drive, which already contains the packages that need to be downloaded. Software package, no need to download, you can go to step 3 directly after decompression

3) Select the profile that uses the OrangePi R1 Plus LTS

```
test@test:~/openwrt$ cp configs/OrangePi_R1_Plus_LTS_defconfig .config
test@test:~/openwrt$ make defconfig
```

4) Start compiling

Execute make V=s to start compiling

```
test@test:~/openwrt$ make V=s
```

5) When you need to select a new software package

```
test@test:~/openwrt$ make menuconfig
```

6) Save your personal package configuration for next use

```
test@test:~/openwrt$ ./scripts/diffconfig.sh > ./configs/my_config
```

7) image generation location

```
openwrt/bin/targets/rockchip/armv8/
openwrt-rockchip-armv8-xunlong_orangepi-r1-plus-lts-ext4-sysupgrade.img.gz
```

6. Instructions for using the Linux SDK

The compilation of the Linux SDK is performed on **a PC or virtual machine (VirtualBox or VMware) with Ubuntu 18.04 installed**. Please do not use other versions of Ubuntu or compile the Linux SDK on WSL.

6.1. Get the source code of linux sdk

6.1.1. Download orangepi-build from github

1) First download the code of orangepi-build. The code of orangepi-build is modified based on the armbian build compilation system. At present, the rk3328 series

development board already supports the current branch

```
test@test:~$ sudo apt update
test@test:~$ sudo apt install git
test@test:~$ git clone https://github.com/orangepi-xunlong/orangepi-build.git
```

2) The current branch generally uses u-boot and kernel close to the mainline version. The u-boot and linux kernel currently used by the rk3328 series development board are as follows

branch	u-boot version	linux kernel version
current	u-boot 2020.10	linux5.10.44

3) Orangepi-build will contain the following files and folders after downloading

a. **build.sh**: Compile startup script
b. **external**: Contains configuration files, specific scripts and source code of some programs needed to compile the image, etc.
c. **LICENSE**: GPL 2 license file
d. **README.md**: orangepi-build documentation
e. **scripts**: Generic script for compiling linux images

```
test@test:~/orangepi-build$ ls
build.sh  external  LICENSE  README.md  scripts
```

6.1.2. Download the cross-compilation toolchain

1) When orangepi-build runs for the first time, it will automatically download the cross-compilation **toolchains** and put it in the **toolchains** folder. After running the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in toolchains exists. , if it does not exist, it will restart the download, if it exists, it will be used directly, and the download will not be repeated

```
[ o.k. ] Checking for external GCC compilers
[ .... ] downloading using http(s) network [ gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz ]
 #8d7029 16MiB/24MiB(65%) CN:1 DL:7.9MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-aarch64-none-elf-4.8-2013.11_linux.tar.xz: 24.9MiB [14.4MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz ]
 #e30eec 17MiB/33MiB(50%) CN:1 DL:10MiB ETA:1s]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-none-eabi-4.8-2014.04_linux.tar.xz: 33.9MiB [9.66MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux.tar.xz ]
 #041c24 48MiB/48MiB(99%) CN:1 DL:2.7MiB]
[ o.k. ] Verified [ PGP ]
[ .... ] decompressing
[ .... ] gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux.tar.xz: 48.8MiB [13.0MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz ]
 #3dee3e 72MiB/76MiB(93%) CN:1 DL:3.7MiB ETA:1s]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi.tar.xz: 77.0MiB [14.2MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz ]
 #42e728 104MiB/104MiB(99%) CN:1 DL:2.8MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi.tar.xz:  104MiB [13.9MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz ]
 #2c065e 108MiB/111MiB(97%) CN:1 DL:3.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz:  111MiB [13.4MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf.tar.xz ]
 #d232ee 250MiB/251MiB(99%) CN:1 DL:2.0MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
[ .... ] gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf.tar.xz:  251MiB [13.7MiB/s] [==========================================================================>] 100%
[ .... ] downloading using http(s) network [ gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu.tar.xz ]
 #88b441 268MiB/269MiB(99%) CN:1 DL:0.9MiB]
[ o.k. ] Verified [ MD5 ]
[ .... ] decompressing
```

2) The image website of the cross-compilation tool chain in China is the open source software image site of Tsinghua University

```
https://mirrors.tuna.tsinghua.edu.cn/armbian-releases/_toolchain/
```

3) After the toolchains is downloaded, it will contain multiple versions of the cross-compilation toolchain

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

4) The cross-compilation toolchain used to compile the rk3328 linux5.10 kernel source code is

```
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

5) The cross-compilation toolchain used to compile the rk3328 u-boot 2020.10 source

code is

```
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

6. 1. 3. Orangepi-build complete directory structure description

1) After the orangepi-build repository is downloaded, it does not contain the linux kernel, u-boot source code and cross-compilation toolchain. The linux kernel and u-boot source code are stored in a separate git repository (**please do not download and use the kernel and u separately -boot source code to compile, unless you know how to use it**)

a. The git repository where the kernel source code of linux5.10 is stored is as follows

```
https://github.com/orangepi-xunlong/linux-orangepi/tree/orange-pi-5.10-rockchip64/
```

b. The git repository where the b.u-boot 2020.10 source code is stored is as follows

```
https://github.com/orangepi-xunlong/u-boot-orangepi/tree/v2020.10-rockchip64
```

2) When orangepi-build runs for the first time, it will download the cross-compilation toolchain, u-boot and linux kernel source code. After successfully compiling a linux image, the files and folders that can be seen in orangepi-build are:

a. **build.sh**: Compile startup script
b. **external**: Contains configuration files, scripts for specific functions and source code of some programs needed to compile the image. The rootfs compressed package cached during the process of compiling the image is also stored in external
c. **kernel**: Store the source code of the linux kernel. The folder named **orange-pi-5.10-rockchip64** stores the kernel source code of the current branch of the rk3328 development board. Please do not modify the name of the folder of the kernel source code manually. If you modify the compilation system The kernel source code will be re-downloaded at runtime
d. **LICENSE**: GPL 2 license file
e. **README.md**: orangepi-build documentation
f. **output**: Store the compiled u-boot, linux and other deb packages, compilation logs, and compiled images and other files
g. **scripts**: Generic script for compiling linux images
h. **toolchains**: Store the cross-compilation toolchain
i. **u-boot**: The folder named **v2020.10-rockchip64** in the source code of u-boot stores the u-boot source code of the current branch of the rk3328 development board. Please do not modify the name of the folder of the u-boot source code

manually. If you modify the compilation system The u-boot source code will be re-downloaded when running

j. **userpatches**: Store the configuration files needed to compile the script

```
test@test:~/orangepi-build$ ls
build.sh  external  kernel  LICENSE  output  README.md  scripts
toolchains  u-boot  userpatches
```

6.1.4. Download orangepi-build from Google Drive

1) If the speed of downloading orangepi-build from github is very slow, you can also download the orangepi-build compressed package from Google Drive. The download link of Google Drive is

```
Link:
https://drive.google.com/drive/folders/1Ai3GVaSk7Dd9IVb9PJYuQc-BRRGFzKrH
```

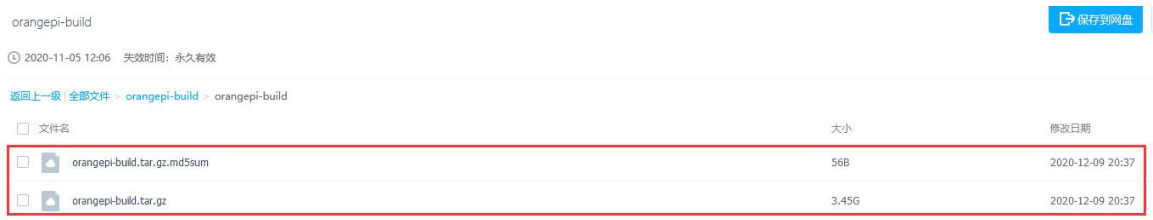
2) There are two files in the orangepi-build folder of Google Drive

a. **orangepi-build.tar.gz** is the compressed package of orangepi-build source code

b. **orangepi-build.tar.gz.md5sum** is the MD5 checksum file of the compressed package of orangepi-build source code

c. After downloading, please first check whether the MD5 checksum of the orangepi-build.tar.gz compressed package is correct, which can prevent problems with the downloaded compressed package. If it is not correct, please download it again and check whether the checksum is correct. The correct command is

```
test@test:~$ md5sum -c orangepi-build.tar.gz.md5sum
orangepi-build.tar.gz: success
```



3) Then you can use the tar -zxf command to decompress orangepi-build.tar.gz

```
test@test:~$ tar -zxf orangepi-build.tar.gz
test@test:~$ cd orangepi-build/
test@test:~/orangepi-build$ ls
build.sh external kernel LICENSE README.md scripts toolchains
u-boot userpatches
```

4) Before using orangepi-build to compile the system, please synchronize orangepi-build with the github server to ensure that the code is in the latest state

```
test@test:~/orangepi-build$ git pull
```

If there is a problem with the synchronization process between orangepi-build and the github server, you can try the following methods

```
test@test:~/orangepi-build$ sudo rm -rf external scripts
test@test:~/orangepi-build$ git checkout .
test@test:~/orangepi-build$ git pull
```

5) The **orangepi-build.tar.gz** compressed package on the Google Drive not only contains the code of the orangepi-build compilation system, but also caches the source code of the cross-compilation toolchain, u-boot and linux kernel. Therefore, in the process of compiling the image, you will not go to the github server to download the source code of u-boot and linux kernel and the cross-compilation toolchain from scratch, which can save a lot of time. When the orangepi-build compilation system starts to run, the source code of u-boot and linux kernel will be automatically synchronized from github by default to ensure that the code is in the latest state, so there is no need to manually synchronize the source code of u-boot and linux kernel

6) The decompressed kernel folder of **orangepi-build.tar.gz** on Google Drive will contain multiple versions of the kernel source code. Please do not modify the names of these kernel source code folders, otherwise the compilation system will not find the

kernel. The source code then re-downloads the kernel source code from github. Not all of the kernel source code in the kernel folder can be used by the rk3328 series development board. The kernel source code used by the rk3328 series development board is as follows:

orange-pi-5.10-rockchip64	Rk3328 系列开发板 current 分支使用

7) The decompressed u-boot folder of **orangepi-build.tar.gz** on Google Drive will contain multiple versions of u-boot source code, please do not modify the names of these u-boot source code folders, otherwise it will cause The compilation system cannot find the u-boot source code and then re-download the u-boot source code from github. Not all versions of u-boot source code in the u-boot folder can be used by Rk3328 series development boards. The u-boot source code used by Rk3328 series development boards are:

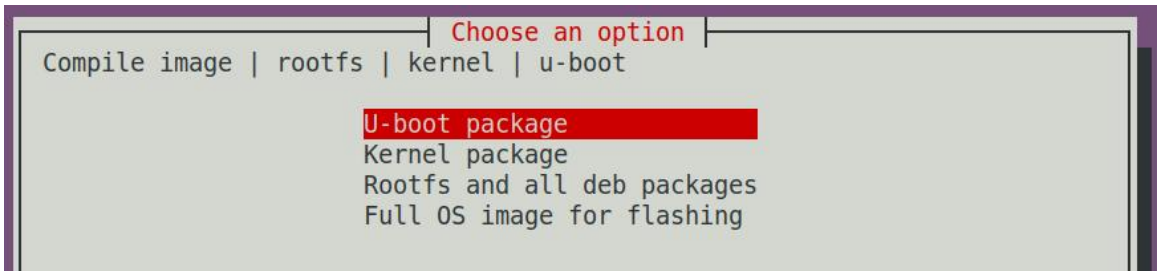
v2020.10	Rk3328 series development board current branch use

6.2. Compile u-boot

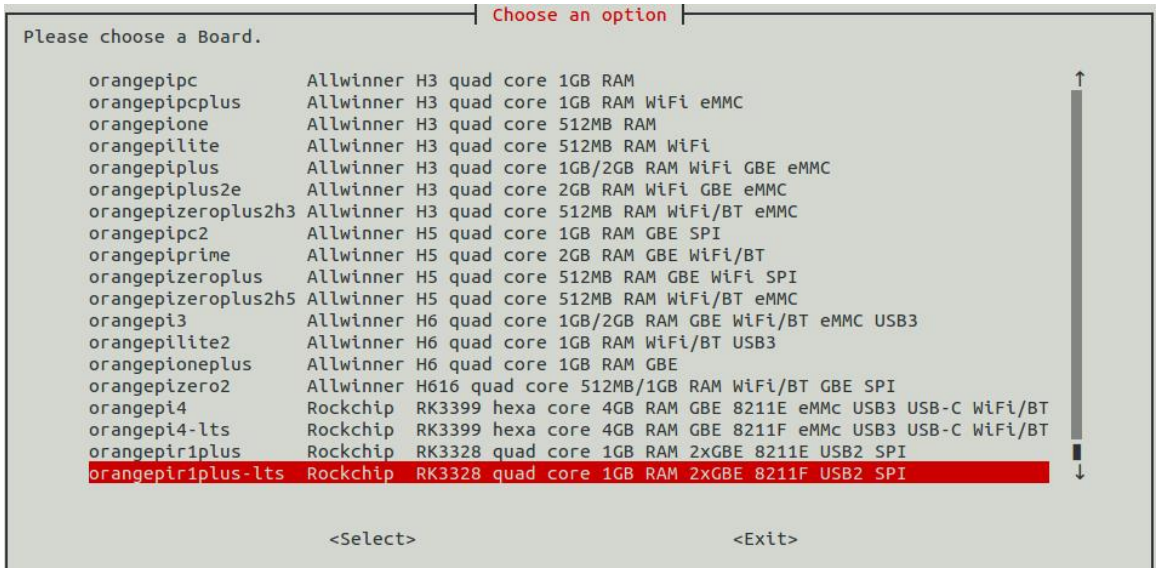
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **U-boot package** and press Enter



3) Then select the model of the development board



4) Then it will start to compile u-boot, and some of the information prompted during compilation are as follows

a. u-boot source code version

[o.k.] Compiling u-boot [**v2020.10**]

b. The version of the cross-compile toolchain

[o.k.] Compiler version [**aarch64-none-linux-gnu-gcc 9.2.1**]

c. The path to the generated u-boot deb package

[o.k.] Target directory [**output/debs/u-boot**]

d. The package name of the u-boot deb package generated by compilation

[o.k.] File name [**linux-u-boot-current-orangepir1plus-lts_2.1.4_arm64.deb**]

e. Compilation time used

[o.k.] Runtime [**1 min**]

f. Repeat the command to compile u-boot, use the following command to start compiling u-boot directly without selecting through the graphical interface

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepir1plus-lts BRANCH=current BUILD_OPT=u-boot BUILD_DESKTOP=no KERNEL_CONFIGURE=yes**]

5) View the compiled u-boot deb package

test@test:~/orangepi-build$ **ls output/debs/u-boot/**
linux-u-boot-current-orangepir1plus-lts_2.1.4_arm64.deb

6) The files contained in the generated u-boot deb package are as follows

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ $ dpkg -x \
linux-u-boot-current-orangepir1plus-lts_2.1.4_arm64.deb .     (Note that there is a
  "." at the end of the command)
test@test:~/orangepi_build/output/debs/u-boot$ ls
linux-u-boot-current-orangepir1plus-lts_2.1.4_arm64.deb    usr
```

b. The decompressed file is as follows

```
test@test:~orangepi-build/output/debs/u-boot$ tree usr
usr
└── lib
    ├── linux-u-boot-current-orangepir1plus-lts_2.1.4_arm64
    │   ├──idbloader.bin
    │   ├──uboot.img          //u-boot binaries
    │   └──trust.bin
    └── u-boot
        ├── LICENSE
        ├── orangepi_r1_plus_rk3328_defconfig
        └── platform_install.sh

3 directories, 6 files
```

7) The orangepi-bulid compilation system will first synchronize the u-boot source code with the u-boot source code of the github server when compiling the u-boot source code, so if you want to modify the u-boot source code, you first need to turn off the download and update function of the source code (**You need to compile u-boot once before closing this function, otherwise you will be prompted that the source code of u-boot cannot be found**), otherwise the modifications will be restored. The method is as follows:

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

8) When debugging the u-boot code, you can use the following method to update the

u-boot in the linux image for testing

a. Upload the compiled u-boot deb package to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs/u-boot
test@test:~/orangepi_build/output/debs/u-boot$ scp \
linux-u-boot-current-orangepir1plus-lts_2.1.4_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and uninstall the installed deb package of u-boot

```
root@orangepir1plus-lts:~# apt purge -y linux-u-boot-orangepir1plus-lts-current
```

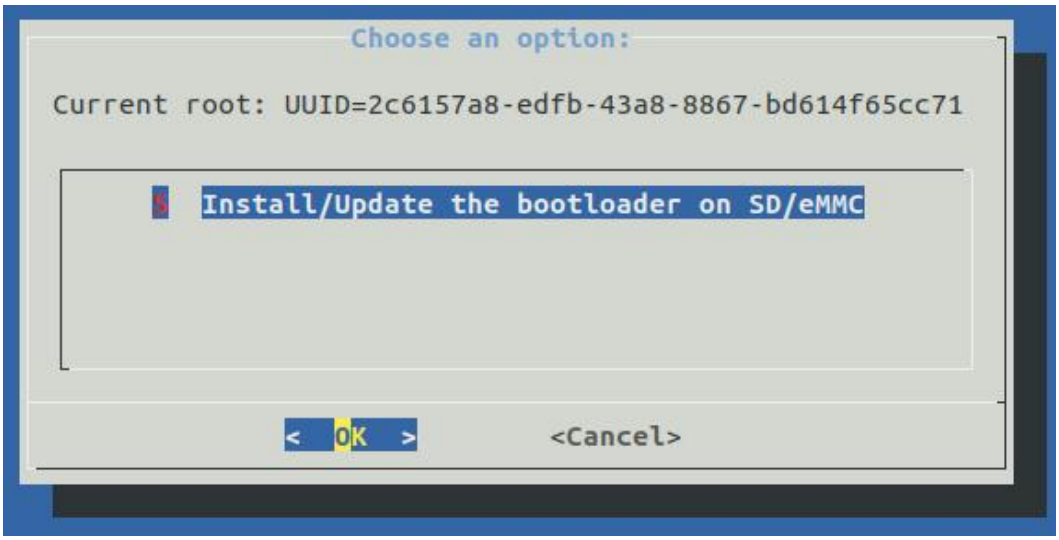
c. Install the new u-boot deb package just uploaded

```
root@orangepir1plus-lts:~#
dpkg -i linux-u-boot-current-orangepir1plus-lts_2.1.4_arm64.deb
```

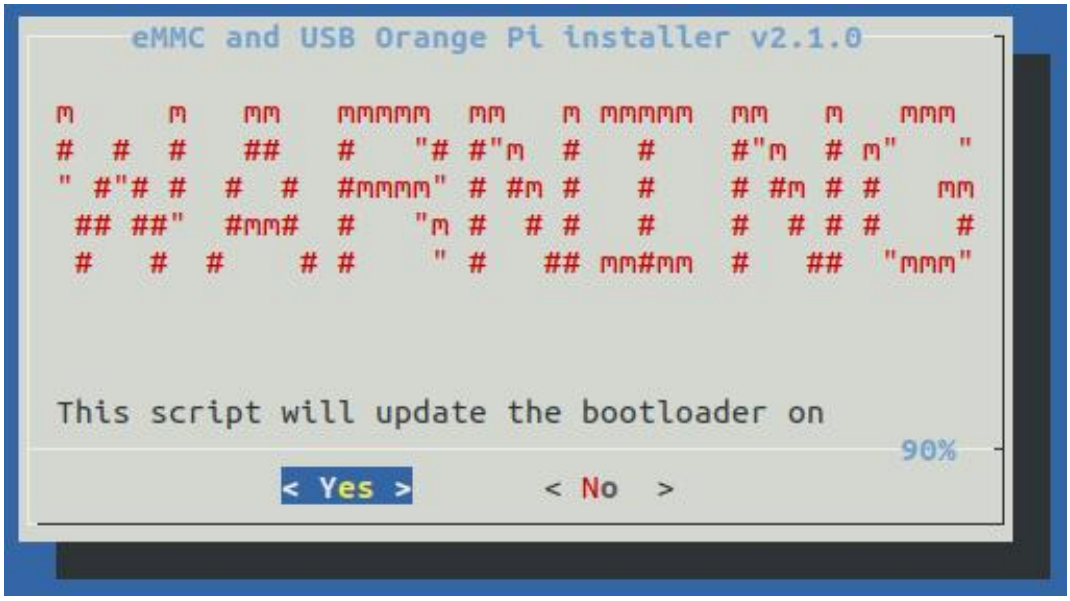
d. Then run the nand-sata-install script

```
root@orangepir1plus-lts:~# nand-sata-install
```

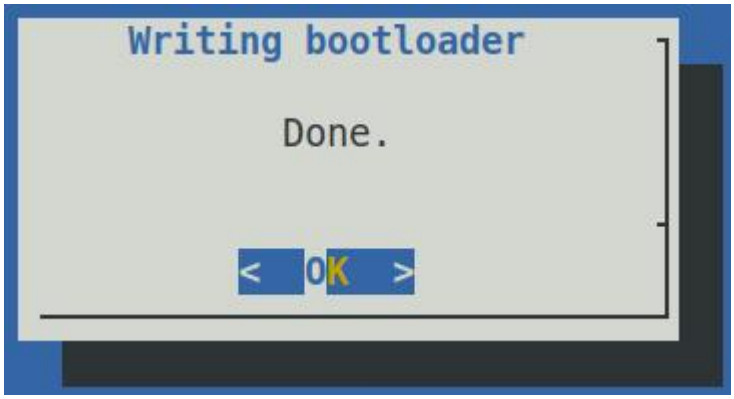
e. Then select **5 Install/Update the bootloader on SD/eMMC**



f. After pressing the Enter key, a Warring will pop up first



g. Press the Enter key again to start updating u-boot. After the update, the following information will be displayed



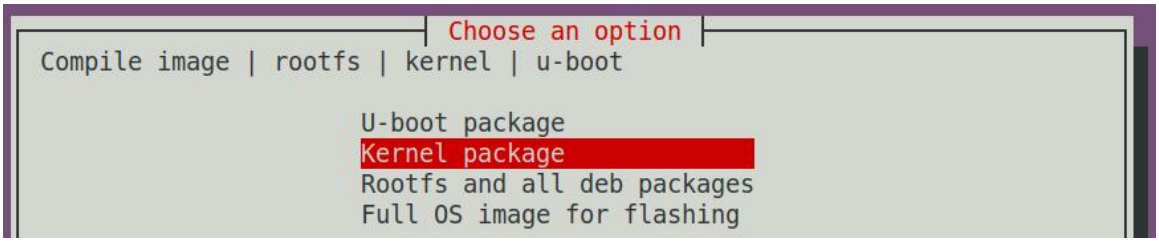
h. Then you can restart the development board to test whether the modification of u-boot takes effect

6. 3. Compile the linux kernel

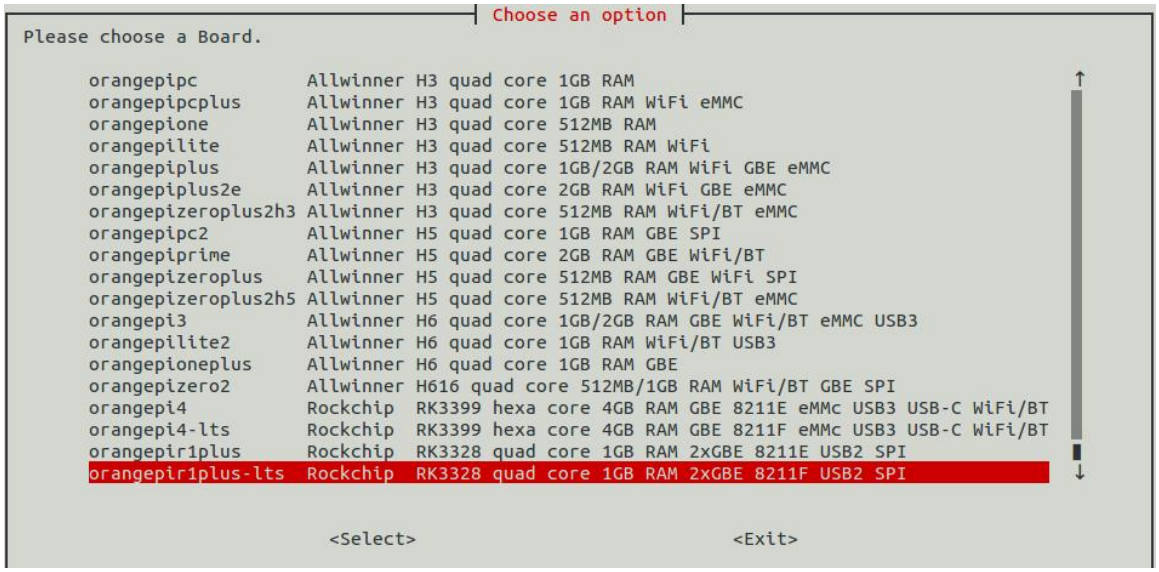
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

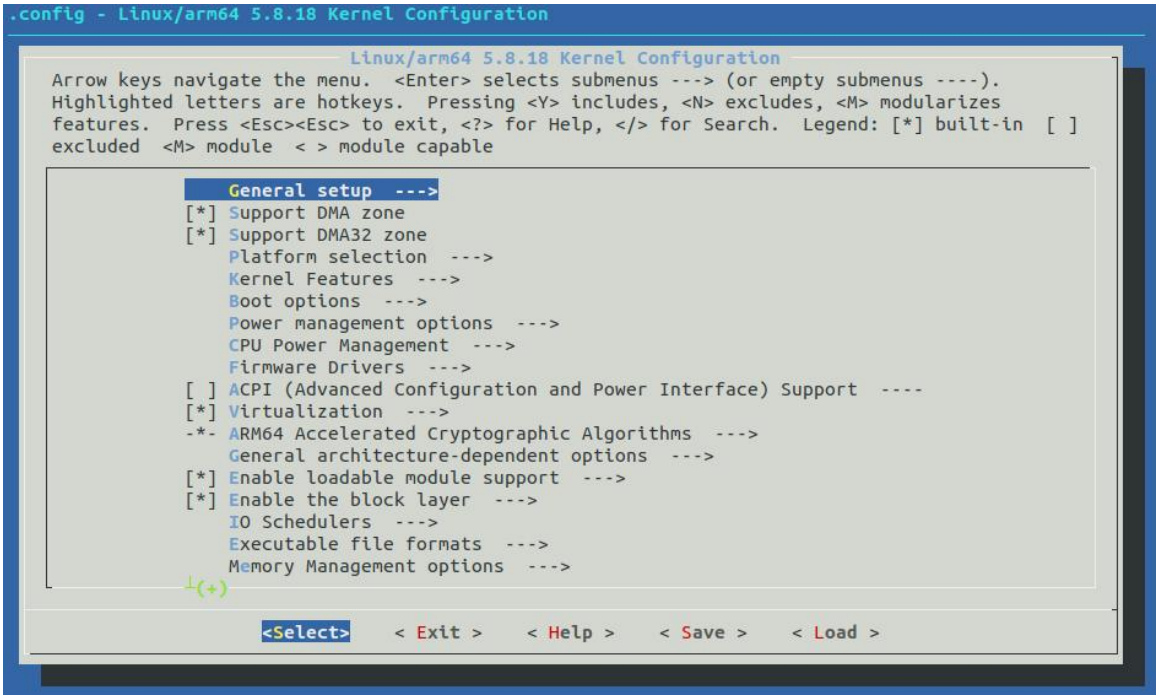
2) Select **Kernel package**, then press Enter



3) Then select the model of the development board



4) Then the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration. If you do not need to modify the kernel configuration, you can simply exit. After exiting, the kernel source code will be compiled.



a. If you do not need to modify the configuration options of the kernel, when running the build.sh script, pass in **KERNEL_CONFIGURE=no** to temporarily shield the configuration interface of the pop-up kernel

```
test@test:~/orangepi-build$ sudo ./build.sh KERNEL_CONFIGURE=no
```

b. You can also set **KERNEL_CONFIGURE=no** in the orangepi-build/userpatches/config-default.conf configuration file to permanently disable this feature

c. If the following error is displayed when compiling the kernel, this is because the terminal interface of Ubuntu PC is too small, so the interface of make menuconfig cannot be displayed. Please adjust the terminal of Ubuntu PC to the maximum, and then re-run the build.sh script

```
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
Your display is too small to run Menuconfig!
It must be at least 19 lines by 80 columns.
scripts/kconfig/Makefile:28: recipe for target 'menuconfig' failed
make[1]: *** [menuconfig] Error 1
Makefile:560: recipe for target 'menuconfig' failed
make: *** [menuconfig] Error 2
[ error ] ERROR in function compile_kernel [ compilation.sh:376 ]
[ error ] Error kernel menuconfig failed
[ o.k. ] Process terminated
```

5) Some of the information prompted when compiling the kernel source code are explained as follows

a. The version of the linux kernel source code

```
[ o.k. ] Compiling legacy kernel [ 5.10.44 ]
```

b. The version of the cross-compilation toolchain used

```
[ o.k. ] Compiler version [ aarch64-none-linux-gnu-gcc 9.2.1 ]
```

c. The configuration file used by the kernel by default and the path where it is stored

```
[ o.k. ] Using kernel config file [ config/kernel/linux-rockchip64-current.config ]
```

d. If **KERNEL_CONFIGURE=yes** is set, the final configuration file .config used by the kernel will be copied to **output/config**. If the kernel configuration is not modified, the final configuration file is the same as the default configuration file

```
[ o.k. ] Exporting new kernel config [ output/config/linux-rockchip64-current.config ]
```

e. The path to the generated kernel-related deb package

```
[ o.k. ] Target directory [ output/debs/ ]
```

f. The package name of the kernel image deb package generated by compilation

```
[ o.k. ] File name [ linux-image-current-rockchip64_2.1.4_arm64.deb ]
```

g. Compilation time used

```
[ o.k. ] Runtime [ 25 min]
```

h. Finally, the compilation command to repeat the compilation of the last selected kernel will be displayed. Use the following command to directly start compiling the kernel source code without selecting through the graphical interface.

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepir1plus-lts BRANCH=current BUILD_OPT=kernel RELEASE=bionic BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

6) View the compiled and generated kernel-related deb packages

a. **linux-dtb-current-rockchip64_2.1.4_arm64.deb** contains dtb files used by the kernel

b. **linux-headers-current-rockchip64_2.1.4_arm64.deb** contains the kernel headers

c. **linux-image-current-rockchip64_2.1.4_arm64.deb** contains the kernel image and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
```

```
output/debs/linux-dtb-current-rockchip64_2.1.4_arm64.deb
output/debs/linux-headers-current-rockchip64_2.1.4_arm64.deb
output/debs/linux-image-current-rockchip64_2.1.4_arm64.deb
```

7) The files contained in the generated linux-image deb package are as follows

a. Use the following command to decompress the deb package

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi_build/output/debs$ mkdir test
test@test:~/orangepi_build/output/debs$ cp \
linux-image-current-rockchip64_2.1.4_arm64.deb test/
test@test:~/orangepi_build/output/debs$ cd test
test@test:~/orangepi_build/output/debs/test$ dpkg -x \
linux-image-current-rockchip64_2.1.4_arm64.deb .
test@test:~/orangepi_build/output/debs/test$ ls
boot  etc  lib  linux-image-current-rockchip64_2.1.4_arm64.deb  usr
```

b. The decompressed file is as follows

```
test@test:~/orangepi_build/output/debs/test$ tree -L 2
.
├── boot
│   ├── config-5.10.44-rockchip64          //The configuration file used to compile the kernel source code
│   ├── System.map-5.10.44-rockchip64
│   └── vmlinuz-5.10.44-rockchip64         //Compile the generated kernel image file
├── etc
│   └── kernel
├── lib
│   └── modules                            //compile the generated kernel module
├── linux-image-current-rockchip64_2.1.4_arm64.deb
└── usr
    ├── lib
    └── share
8 directories, 4 files
```

8) The orangepi-bulid compilation system will first synchronize the Linux kernel source code with the linux kernel source code of the github server when compiling the linux

kernel source code, so if you want to modify the linux kernel source code, you first need to turn off the update function of the source code (**you need to compile it once This function can only be turned off after the linux kernel source code, otherwise it will prompt that the source code of the linux kernel cannot be found**), otherwise the modification will be restored, the method is as follows:

Set the IGNORE_UPDATES variable in userpatches/config-default.conf to "yes"

```
test@test:~/orangepi-build$ vim userpatches/config-default.conf
IGNORE_UPDATES="yes"
```

9) If the kernel is modified, you can use the following methods to update the kernel and kernel modules of the development board linux system

a. Upload the compiled deb package of the linux kernel to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-image-current-rockchip64_2.1.4_arm64.deb root@192.168.1.207:/root
```

b. Then log in to the development board and uninstall the deb package of the installed linux kernel

```
root@orangepir1plus-lts:~# apt purge -y linux-image-current-rockchip64
```

c. Install the deb package of the new linux kernel just uploaded

```
root@orangepir1plus-lts:~# dpkg -i linux-image-current-rockchip64_2.1.4_arm64.deb
```

d. Then restart the development board, and then check whether the kernel-related modifications have taken effect

10) The method of installing the kernel header files into the linux system is as follows

a. Upload the deb package of the compiled linux header file to the linux system of the development board

```
test@test:~/orangepi-build$ cd output/debs
test@test:~/orangepi-build/output/debs$ scp \
linux-headers-current-rockchip64_2.1.4_arm64.deb root@192.168.1.xxx:/root
```

b. Then log in to the development board and install the deb package of the linux header file just uploaded

```
root@orangepir1plus-lts:~# dpkg -i linux-headers-current-rockchip64_2.1.4_arm64.deb
```

c. After installation, you can see the contents of the kernel header files just

installed in /usr/src

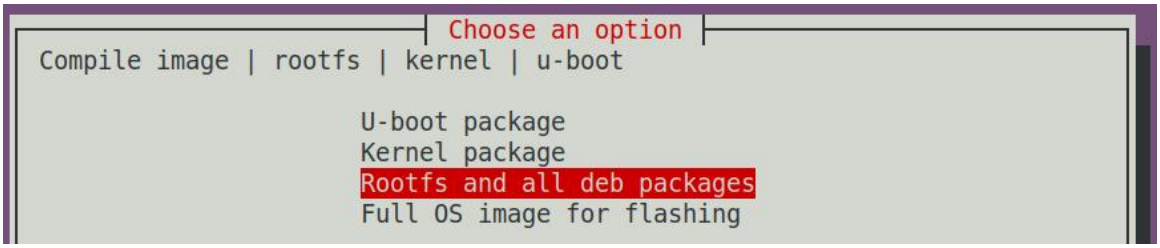
```
root@orangepir1plus-lts:~# ls /usr/src
linux-headers-5.10.44-rockchip64
root@orangepir1plus-lts:~# ls /usr/src/linux-headers-5.10.44-rockchip64
arch crypto fs ipc lib Module.symvers scripts tools block Documentation include Kconfig Makefile net security usr certs drivers init kernel mm samples sound virt
```

6. 4. Compile rootfs

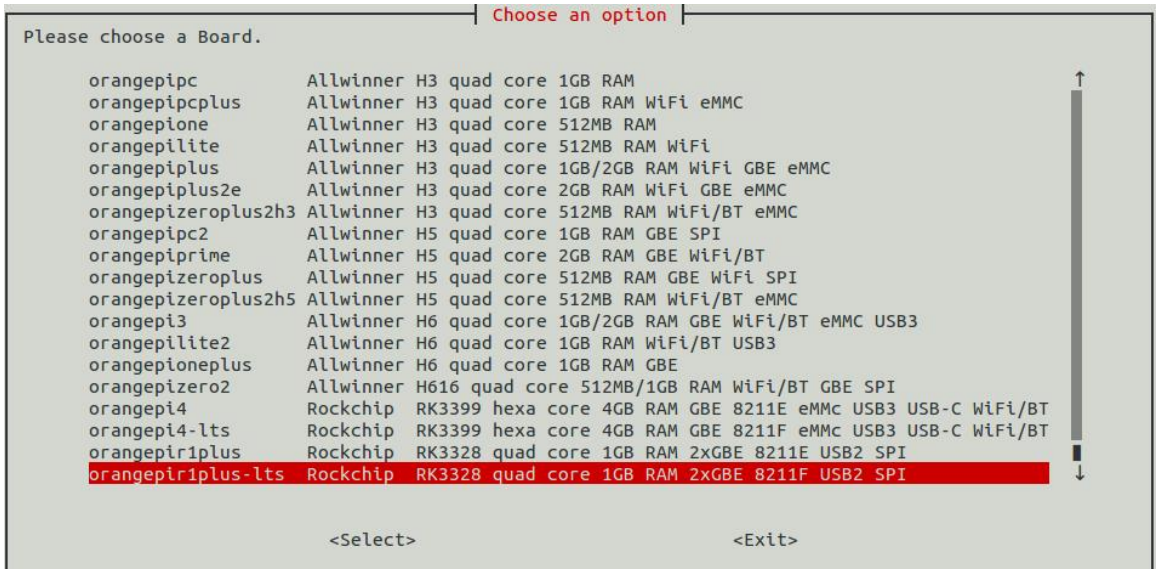
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Rootfs and all deb packages**, then press Enter



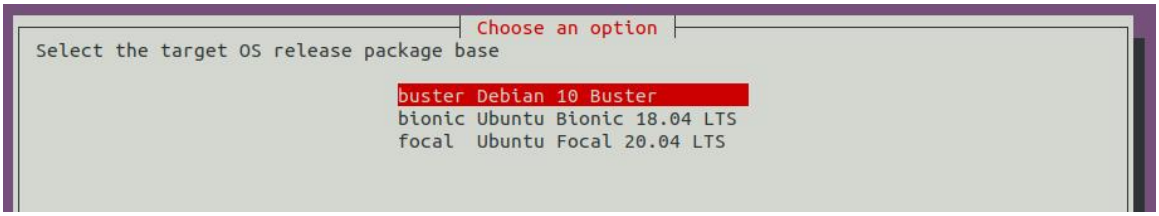
3) Then select the model of the development board



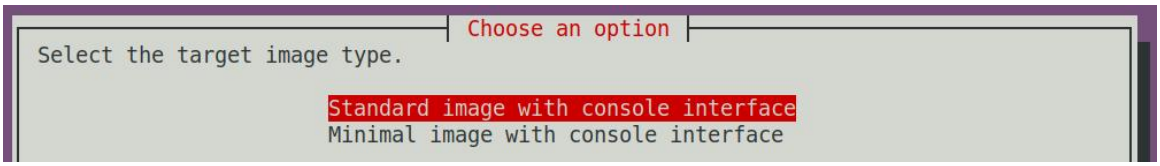
4) Then select the type of rootfs

buster	Debian 10
bionic	Ubuntu 18.04
focal	Ubuntu 20.04

The linux distributions supported by linxu5.10 are as follows



5) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The software pre-installed in the Minimal version will be much less than the Standard version.



6) After selecting the type of image, the rootfs will be compiled, and some of the information prompted during compilation are as follows

a. type of rootfs

[o.k.] local not found [Creating new rootfs cache for **bionic**]

b. The storage path of the rootfs compressed package generated by compilation

[o.k.] Target directory [**external/cache/rootfs**]

c. The name of the rootfs compressed package generated by compilation

[o.k.] File name [**bionic-cli-arm64.153618961f14c28107ca023429aa0eb9.tar.lz4**]

d. Compilation time

[o.k.] Runtime [**13 min**]

e. Repeat the command to compile rootfs, use the following command to start compiling rootfs directly without selecting through the graphical interface

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepir1plus-lts BRANCH=current BUILD_OPT=rootfs RELEASE=bionic BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes**]

7) View the rootfs compressed package generated by compilation

a. **bionic-cli-arm64.153618961f14c28107ca023429aa0eb9.tar.lz4** is the compressed package of rootfs, the meaning of each field of the name is

a) **bionic** indicates the type of linux distribution for rootfs

b) **cli** indicates that rootfs is the type of server version

c) **arm64** represents the architecture type of rootfs

d) **153618961f14c28107ca023429aa0eb9** is the MD5 hash value generated by the package names of all packages installed by rootfs. As long as the list of packages installed by rootfs is not modified, this value will not change, and the compilation script will use this MD5 hash value to Determine if you need to recompile rootfs

b. **bionic-cli-arm64.153618961f14c28107ca023429aa0eb9.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
bionic-cli-arm64.153618961f14c28107ca023429aa0eb9.tar.lz4
bionic-cli-arm64.153618961f14c28107ca023429aa0eb9.tar.lz4.list
```

8) If the required rootfs already exists under **external/cache/rootfs**, then compiling rootfs again will skip the compilation process directly and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has There is a rootfs available for cache, if there is one, use it directly, which can save a lot of download and compilation time

9) Since it takes a long time to compile rootfs, if you don't want to compile rootfs from scratch, or there is a problem with the process of compiling rootfs, you can directly download the rootfs compressed package cached by Orange Pi. The download link of the rootfs compressed package Google Drive is as follows, download A good rootfs compressed package (don't decompress it) needs to be placed in the **external/cache/rootfs** directory of orangepi-build to be used by the compiled script.

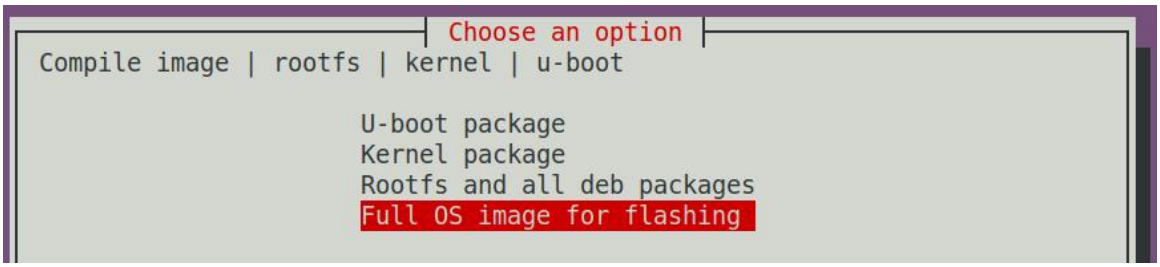
Link：https://drive.google.com/drive/folders/1Ai3GVaSk7Dd9IVb9PJYuQc-BRRGFzKrH

6. 5. Compile the linux image

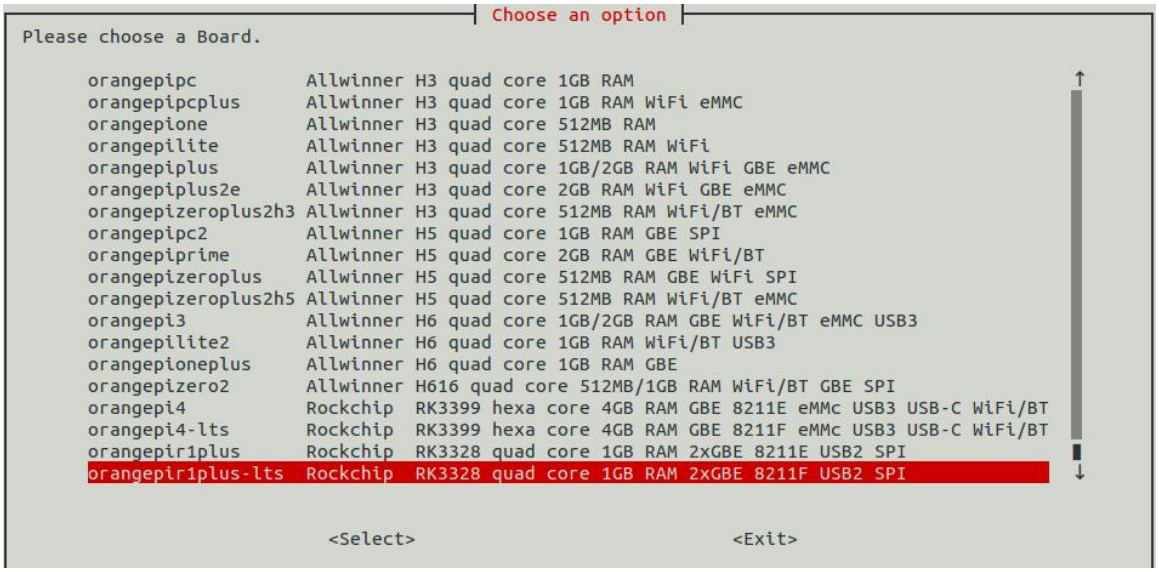
1) Run the build.sh script, remember to add sudo permissions

```
test@test:~/orangepi-build$ sudo ./build.sh
```

2) Select **Full OS image for flashing** and press Enter



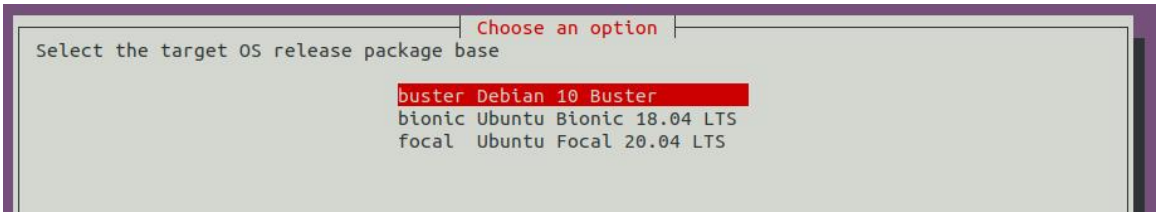
3) Then select the model of the development board



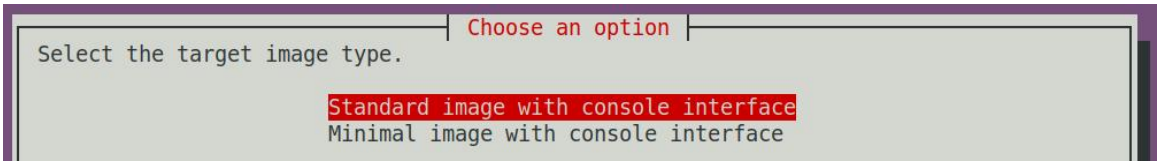
4) Then select the type of rootfs

buster	Debian 10
bionic	Ubuntu 18.04
focal	Ubuntu 20.04

The linux distributions supported by linxu5.10 are as follows



5) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The software pre-installed in the Minimal version will be much less than the Standard version.



6) After selecting the type of image, the linux image will be compiled. The general process of compilation is as follows

a. Initialize the compilation environment of the Ubuntu PC and install the software packages required for the compilation process
b. Download the source code of u-boot and linux kernel (if cached, only update the code)
c. Compile u-boot source code and generate u-boot deb package
d. Compile the linux source code to generate linux-related deb packages
e. Make a deb package of linux firmware
f. Make the deb package of the orangepi-config tool
g. Make board-level supported deb packages
h. Check whether the rootfs has been cached. If there is no cache, recreate the rootfs. If it has been cached, directly decompress it and use it
i. Install the deb package generated earlier into rootfs
j. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system

configurations, etc.

k. Then make an image file and format the partition, the default type is ext4

l. Copy the configured rootfs to the mirrored partition

m. Then update the initramfs

n. Finally, write the bin file of u-boot into the image through the dd command

7) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44/Orangepir1plus-lts_2.1.4_ubuntu_bionic_server_linux5.10.44.img ]
```

b. Compilation time used

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the command to compile the image, use the following command to start compiling the image directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepir1plus-lts BRANCH=current BUILD_OPT=image RELEASE=bionic BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```