3.19.1.2. Method of adjusting PWM duty cycle

1) The calculation formula of PWM duty cycle is as follows. We can adjust the PWM duty cycle by setting the values of CCR and ARR.

PWM duty cycle = CCR / ARR

In：

The value range of CCR is 0~65535, and the default value is 512.

The value range of ARR is 1~65536, and the default value is 1024.

It should be noted that the CCR value we set needs to be less than ARR, because the duty cycle cannot be greater than 1. When setting CCR > ARR, the following error message will be prompted:

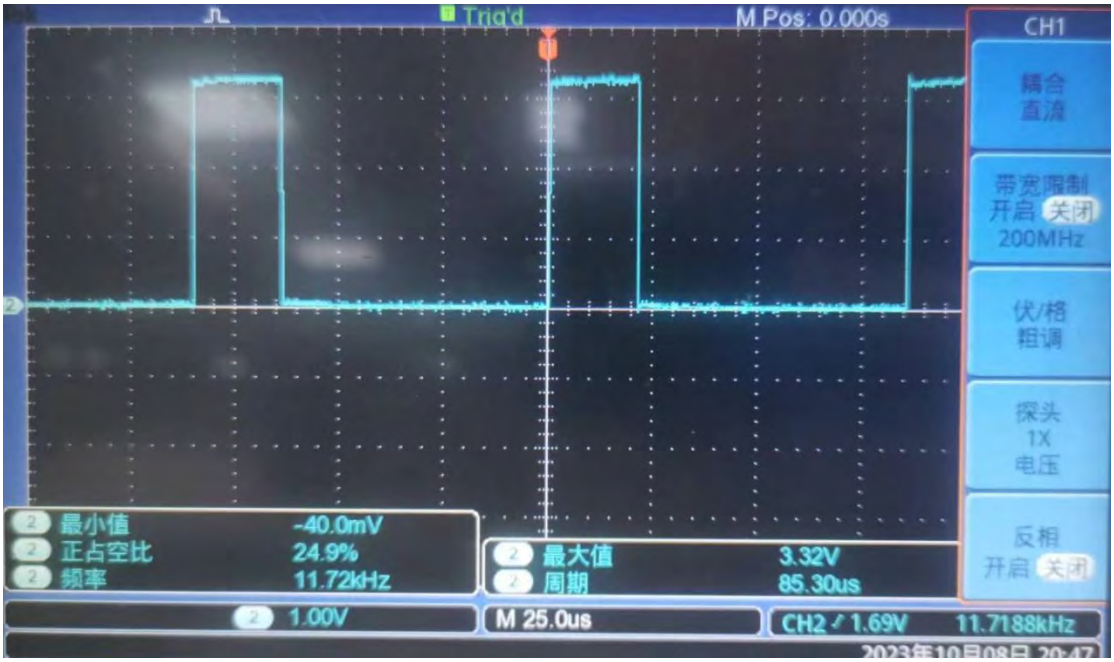
val pwmWrite 0 <= X <= 1024

Or you can set new range by yourself by pwmSetRange(range)

2) We can use the following command to set the ARR of PWM1 pin to 2048

```
orangepi@orangepi:~$ gpio pwmr 4 2048
```

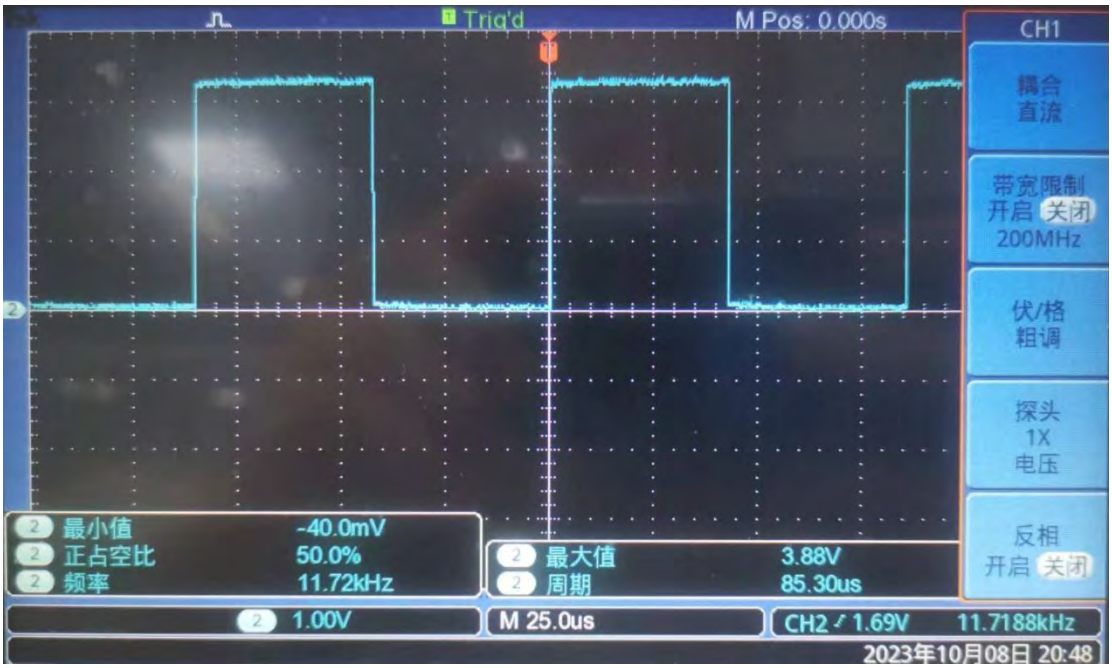
3) After running the above command, you can observe through the oscilloscope that the PWM duty cycle changes from the default 50% (512/1024) to 25% (512/2048)



4) We can use the following command to set the CCR of the PWM1 pin to 1024

```
orangepi@orangepi:~$ gpio pwm 4 1024
```

5) After running the above command, you can observe through the oscilloscope that the PWM duty cycle changes from 25% (512/2048) to 50% (1024/2048)



3. 19. 1. 3. Method of adjusting PWM frequency

3. 19. 1. 3. 1. Method to adjust PWM frequency by setting frequency division coefficient

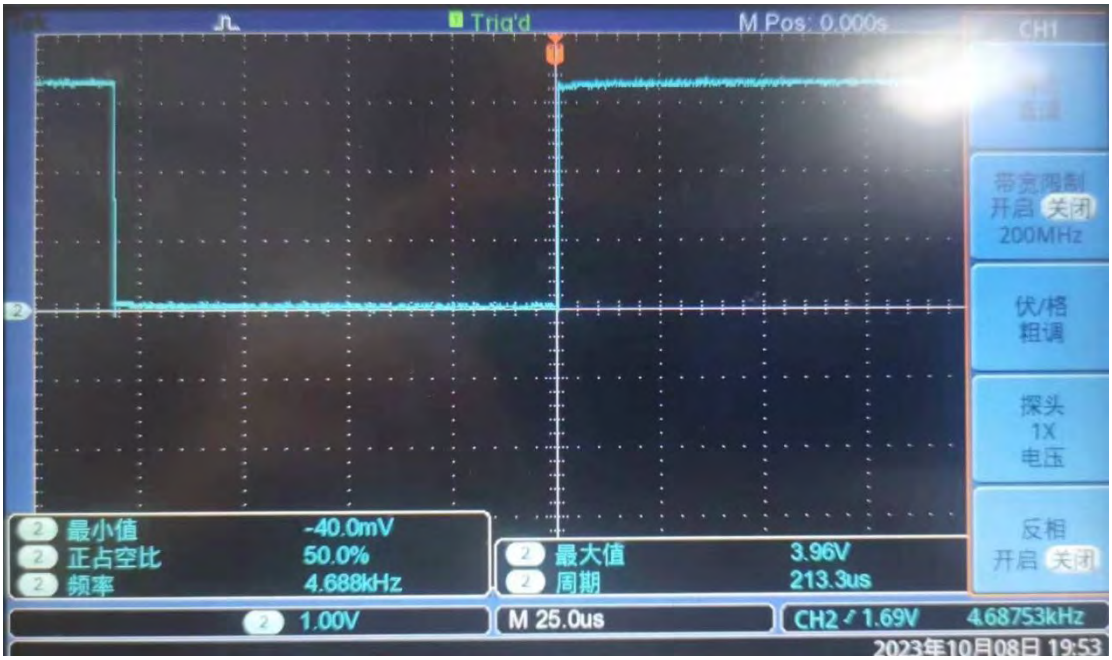
1) After setting the frequency division coefficient, the PWM frequency will become one/one of the frequency division coefficient.

分频系数的取值范围是 1~256，默认是 1。

2) For example, you can use the following command to set the frequency division coefficient of the PWM1 pin to 5

```
orangepi@orangepi:~$ gpio pwmc 4 5
```

3) The default frequency of PWM is 23475Hz. After dividing by 5, the calculated value is 4695Hz. Through the oscilloscope, it can be observed that the actual value of PWM frequency is 4688Hz, and the error can be ignored.



3. 19. 1. 3. 2. Method of directly setting the PWM frequency

1) We can use the **gpio pwmTone** command to set the frequency of the PWM pin. For

example, use the following command to set the PWM frequency of the PWM1 pin to 20000Hz.

```
orangepi@orangepi:~$ gpio pwmTone 4 20000
```

When setting the PWM frequency, you need to ensure:

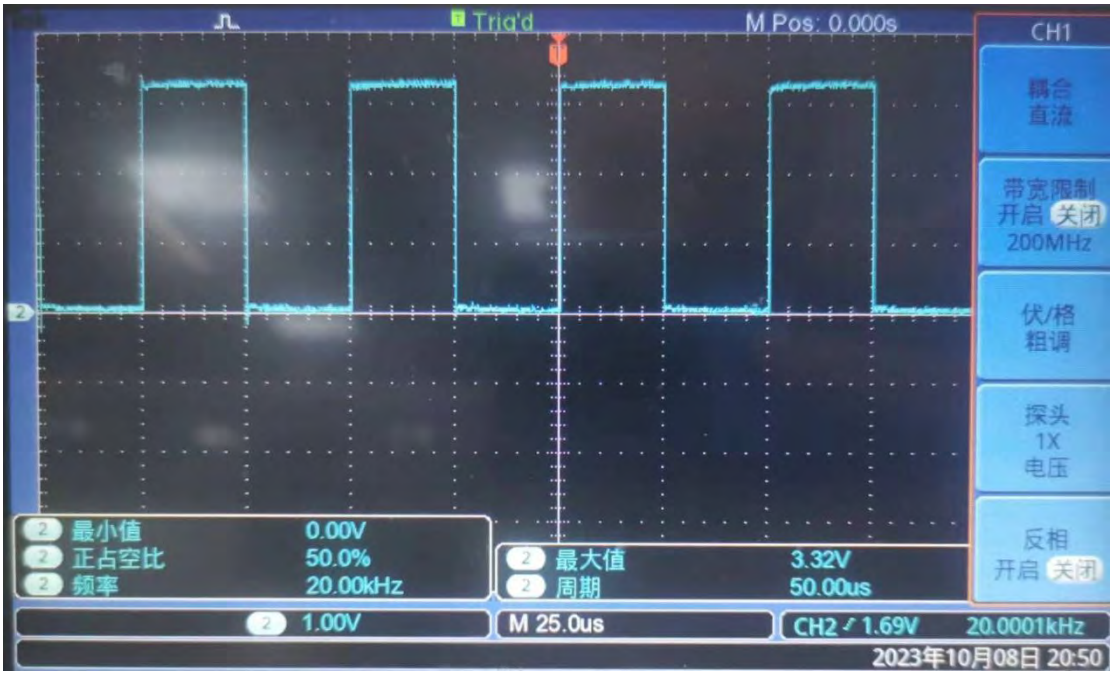
Set frequency value > 24000000 / (65536 * frequency division coefficient).

For example, the default frequency division coefficient is 1. If the frequency division coefficient is not modified, the set frequency value should be greater than 366.

If the setting value is too small, the following error will appear:

gpio: range must be between 1 and 65536

2) 然后通过示波器可以观察到 PWM 频率变为 20000Hz 了



3. 19. 2. How to use the PWM test program

1) In the example directory of wiringOP, there is a program named pwm.c. This program demonstrates how to operate PWM using the PWM-related API in wiringOP.

```
orangepi@orangepi:~$ cd wiringOP/examples/
orangepi@orangepi:~/wiringOP/examples$ ls pwm.c
pwm.c
```

2) The command to compile **pwm.c** into an executable program is as follows

```
sorangepi@orangepi:~/wiringOP/examples$ gcc -o pwm pwm.c -lwiringPi
```

3) Then you can execute the PWM test program. When executing the PWM test program, you need to specify the PWM pin. For example, you can use the following command to test the PWM1 pin:

```
sorangepi@orangepi:~/wiringOP/examples$ sudo ./pwm 4
```

4) After the pwm program is executed, the following contents will be tested in sequence:
   a. Adjust the PWM duty cycle by setting ARR
   b. Adjust the PWM duty cycle by setting CCR
   c. Adjust the PWM frequency by setting the frequency division coefficient
   d. Directly set the PWM frequency

5) After each test is completed, the PWM waveform output will be stopped for 5 seconds. After all test contents are completed, a new round of testing will be restarted.

6) The detailed execution process of the PWM test program is as follows:
   a. Adjust the PWM duty cycle by setting ARR: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 50% to 25%, maintains it for 5 seconds, and then PWM The waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 25% to 50% and remains for 5 seconds.
   b. Adjust the PWM duty cycle by setting CCR: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 50% to 100%, maintains it for 5 seconds, and then PWM The waveform changes every 0.5 seconds. After changing 8 times, the PWM duty cycle changes from 100% to 50% and remains for 5 seconds.
   c. Adjust the PWM frequency by setting the frequency division coefficient: You can observe through the oscilloscope that the PWM waveform changes every 0.5 seconds. After changing 9 times, the PWM frequency becomes 1/10 of the default PWM frequency, which is 2347Hz, and remains for 5 seconds. , and then the PWM waveform changes every 0.5 seconds. After changing 9 times, the

PWM frequency changes to the default PWM frequency, which is 23475Hz, and remains for 5 seconds.

d. Directly set the PWM frequency: It can be observed through the oscilloscope that the PWM frequency first changes to 2000Hz, and then the PWM frequency increases by 2000Hz every two seconds. After changing 9 times, the PWM frequency changes to 20000Hz and remains for 5 seconds.

3.20. How to install and use wiringOP-Python

wiringOP-Python is the Python language version of wiringOP, which is used to operate the hardware resources of the development board, such as GPIO, I2C, SPI and UART, in the Python program.

In addition, please note that all the following commands are operated under the root user.

3.20.1. How to install wiringOP-Python

1) First install the dependency package

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

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

Note that the following git clone --recursive command will automatically download the source code of wiringOP, because wiringOP-Python depends on wiringOP. Please make sure that the download process does not report errors due to network problems.

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

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

3) Then use the following command to compile wiringOP-Python and install it into the

Linux system of the development board

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

4) Then enter the following command, if there is help information output, it means that wiringOP-Python is installed successfully, press the **q** key to exit the help information interface

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

NAME
    wiringpi

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

5) The steps to test whether wiringOP-Python is successfully installed under the python command line are as follows:

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

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

b. Then import the python module of wiringpi

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

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

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

NAME
    wiringpi
```

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

CLASSES
    builtins.object
        GPIO
        I2C
        Serial
        nes

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

>>>
```

3. 20. 2. 26pin GPIO port test

wiringOP-Python is the same as wiringOP, you can also determine which GPIO pin to operate by specifying the wPi number, because there is no command to check the wPi number in wiringOP-Python, so you can only check the board wPi number and physical Correspondence between pins.

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

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

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

2) The steps to test directly with the command are as follows:

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

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

b. Then set the GPIO port to output low level. After setting, you can use a

multimeter to measure the voltage value of the pin. If it is 0v, it means that the low level is set successfully.

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

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

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

3) The steps to test in the command line of python3 are as follows:

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

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

b. Then import the python module of wiringpi

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

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

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

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

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

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

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

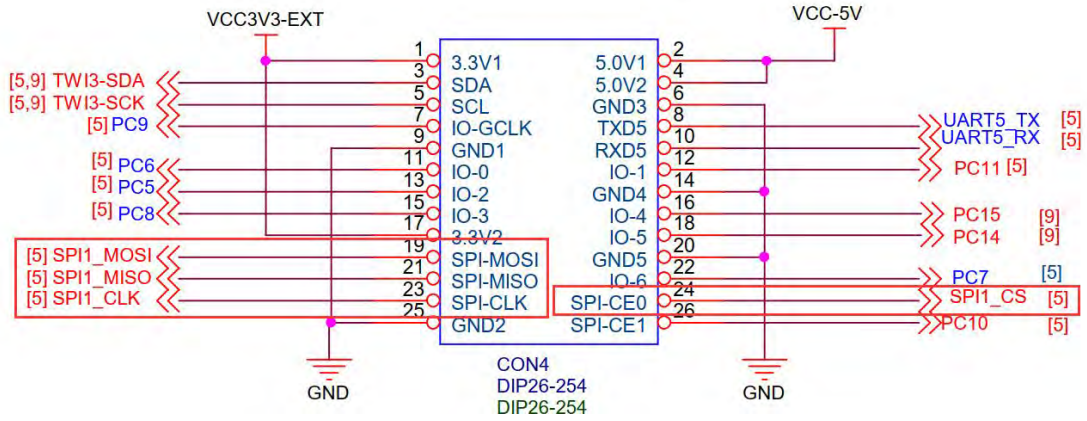
4) The method of wiringOP-Python to set GPIO high and low levels in python code can refer to the **blink.py** test program in the examples below. The **blink.py** test program will

set the voltage of all GPIO ports in the 26 pins of the development board to change continuously.

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

3. 20. 3. 26pin SPI test

1) According to the schematic diagram of the 26pin interface, the available spi is spi1

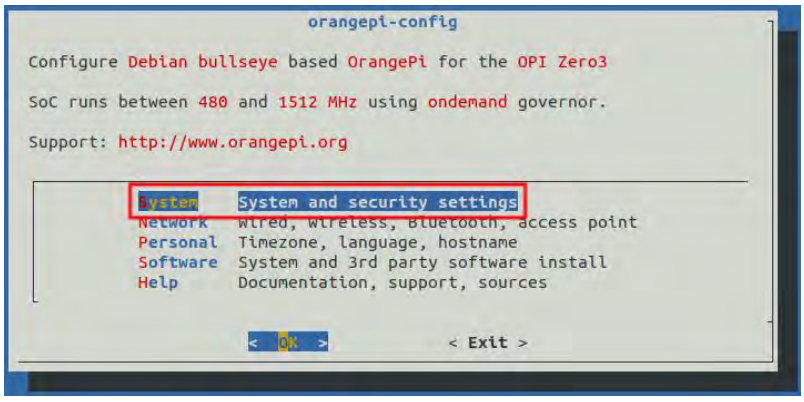


2) The spi1 is disabled by default in the Linux system and needs to be manually enabled before it can be used. The steps to open are as follows:

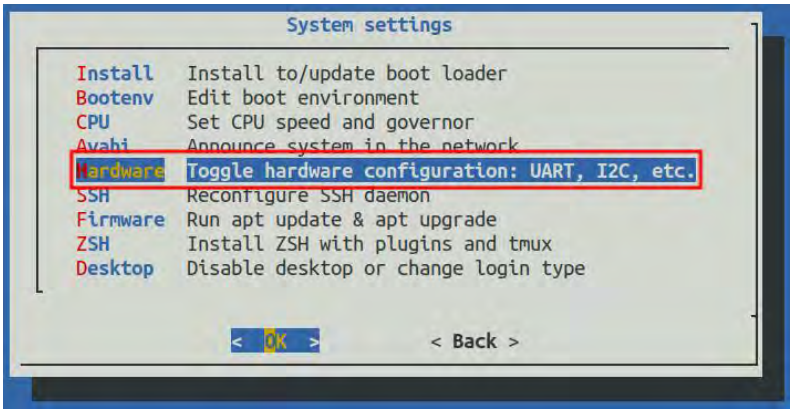
a. First run **orangepi-config**, normal users remember to add **sudo** permission

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

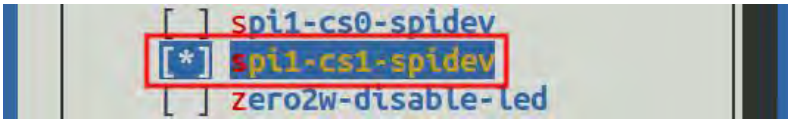
b. Then select **System**



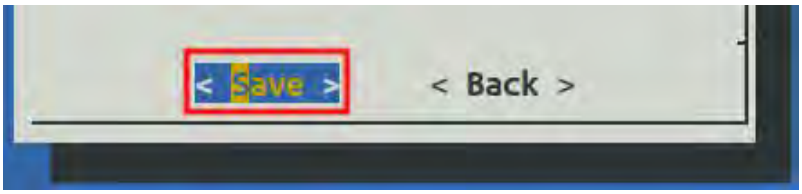
c. Then select **Hardware**



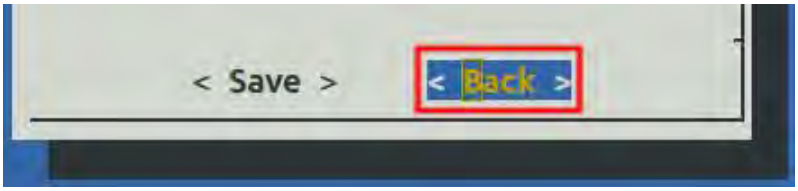
d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **spi1-cs1-spidev**



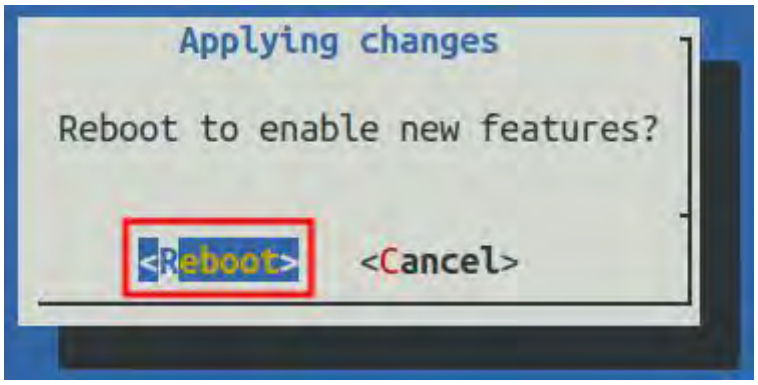
e. Then select **<Save>** to save



f. Then select **<Back>**然后选择**<Back>**



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



3) Then check whether there is a **spidev1.1** device node in the Linux system. If it exists, it means that the configuration of SPI1 has taken effect

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

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

a. **--channel**: Specify the channel number of SPI
b. **--port**: Specify the port number of SPI

5) Do not short-circuit the mosi and miso pins of SPI1, the output result of running spidev_test.py is as follows, you can see that the data of TX and RX are inconsistent

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 1 --port 1
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev1.1
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@.....|
RX | FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF    |................|
```

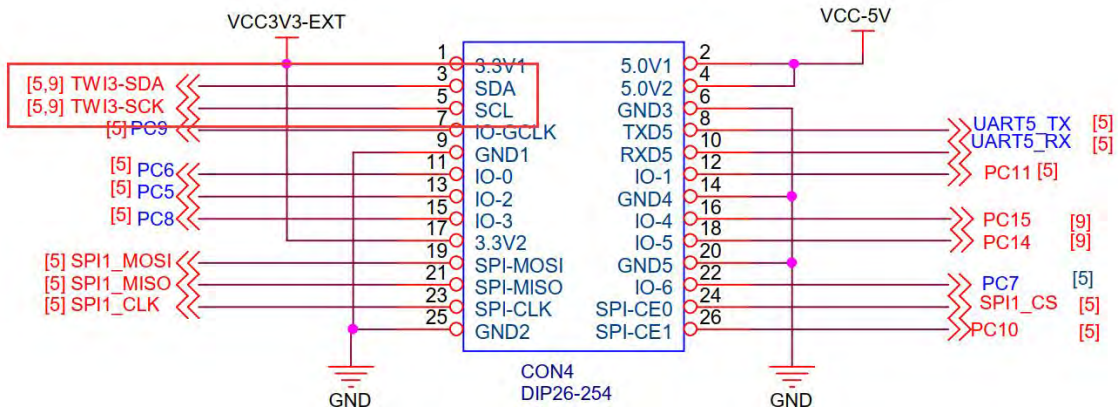
6) Then use the Dupont wire to short-circuit the two pins of txd (pin 19 in the 26pin interface) and rxd (pin 21 in the 26pin interface) of SPI1 and then run the output of spidev_test.py as follows, you can see The data sent and received are the same, indicating that the SPI1 loopback test is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 spidev_test.py \
--channel 1 --port 1
spi mode: 0x0
max speed: 500000 Hz (500 KHz)
Opening device /dev/spidev1.1
TX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF F0 0D    |......@.....|
RX | FF FF FF FF FF FF 40 00 00 00 00 95 FF FF FF FF FF FF FF FF FF FF FF FF
```

```
FF FF FF FF FF F0 0D   |.....@.....|
```

3. 20. 4. 26pin I2C test

1) According to the schematic diagram of 26pin, the available i2c is i2c3

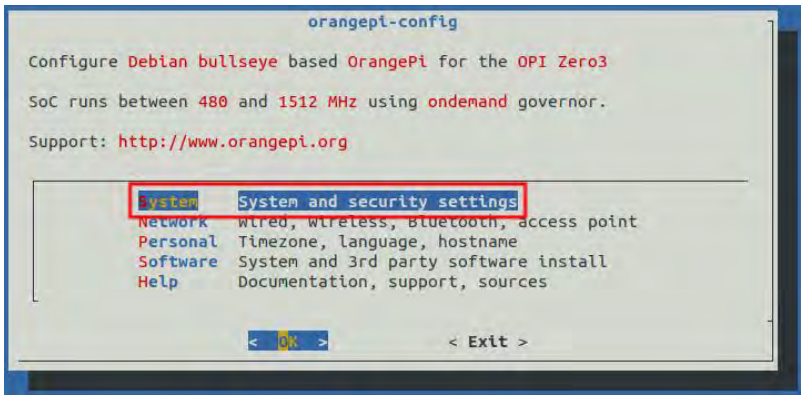


2) i2c3 is disabled by default in the Linux system, and it needs to be manually enabled before it can be used. The steps to open are as follows:

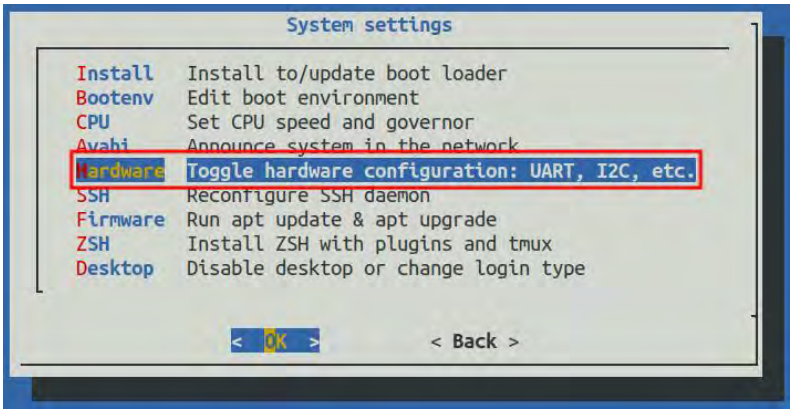
a. First run **orangepi-config**, normal users remember to add **sudo** permission

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

b. Then select **System**

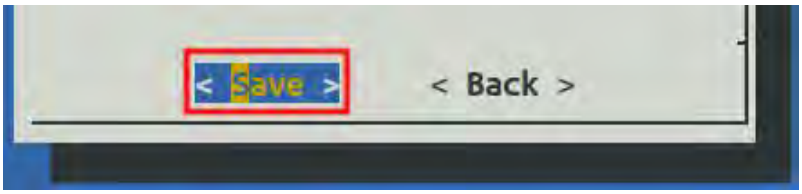


c. Then select **Hardware**

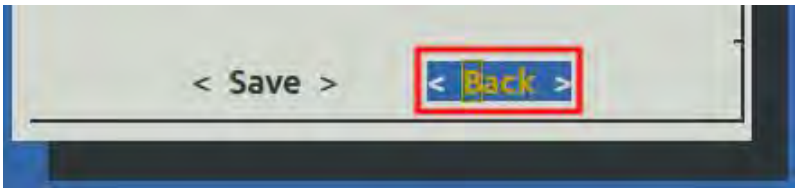


d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-i2c3**

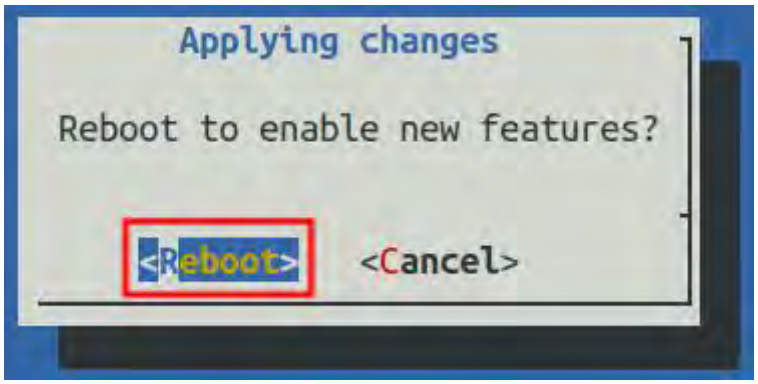
e. Then select **<Save>** to save



f. Then select **<Back>**



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



3) After starting the linux system, first confirm that there is an i2c3 device node under /dev

```
orangepi@orangepi:~$ ls /dev/i2c-3
/dev/i2c-3
```

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

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y i2c-tools
```

5) Then connect an i2c device to the i2c3 pin of the 26pin connector, here we take the DS1307 RTC module as an example

Pins of the RTC module	The pin corresponding to the 26pin of the development board
5V	**Pin 2**
GND	**Pin 6**
SDA	**Pin 3**
SCL	**Pin 5**

6) Then use the **i2cdetect -y 3** command, if the address of the connected i2c device can be detected, it means that the i2c device is connected correctly

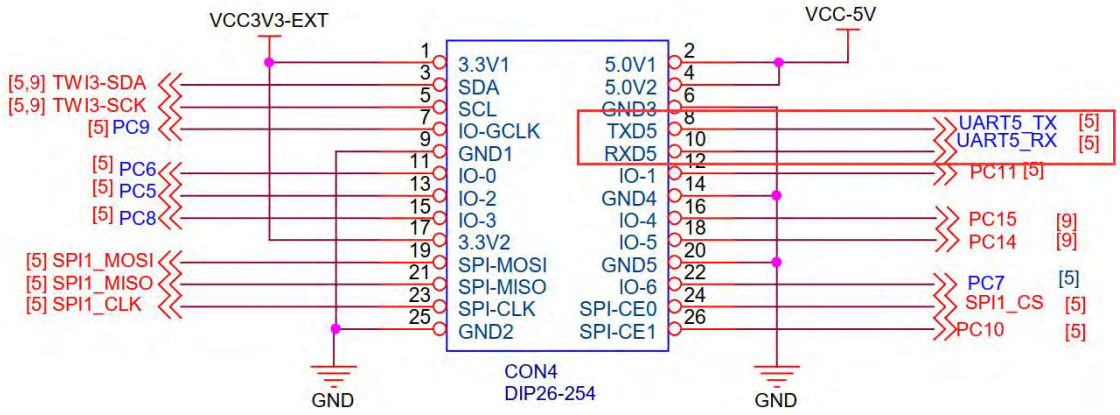
```
orangepi@orangepi:~$ sudo i2cdetect -y 3
[sudo] password for orangepi:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
orangepi@orangepi:~$
```

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

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 ds1307.py --device \
"/dev/i2c-3"
Thu 2022-06-16  04:35:46
Thu 2022-06-16  04:35:47
Thu 2022-06-16  04:35:48
^C
exit
```

3. 20. 5. 26pin UART test

1) According to the schematic diagram of the 26pin interface, the available uart is uart5

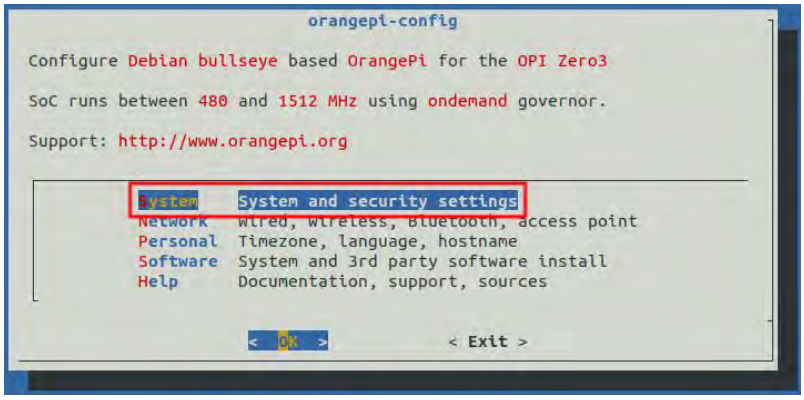


2) uart5 is disabled by default in the Linux system, and it needs to be opened manually to use it. The steps to open are as follows:

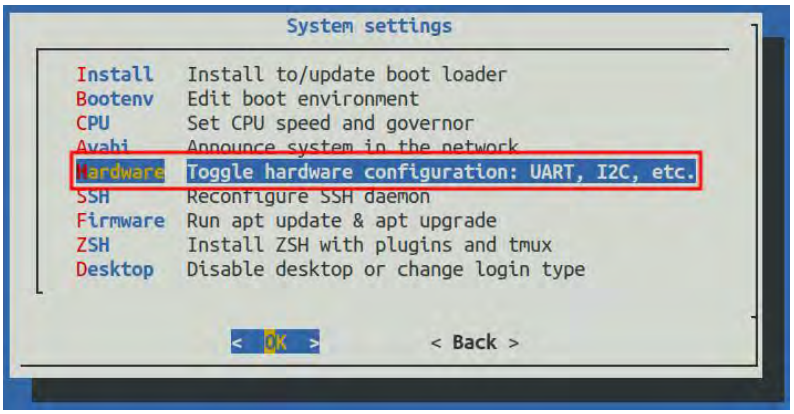
a. First run **orangepi-config**, normal users remember to add **sudo** permission

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

b. Then select **System**

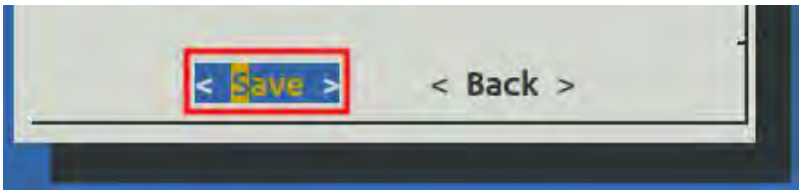


c. Then select **Hardware**

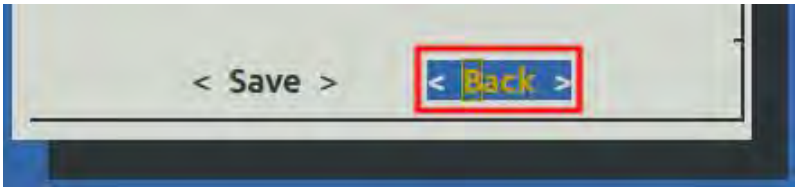


d. d. Then use the arrow keys on the keyboard to navigate to the position shown in the figure below, and then use the **space** to select **ph-uart5**

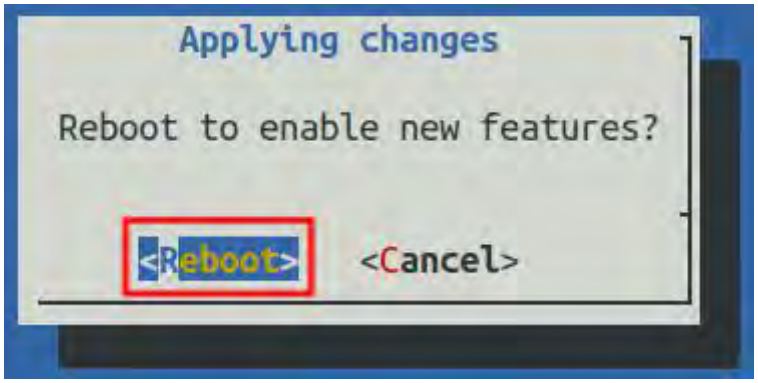
e. Then select **<Save>** to save



f. Then select **<Back>**



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



3) After entering the linux system, first confirm whether there is a uart5 device node

under **/dev**

Note that the linux5.4 system is /dev/ttyAS5.

```
orangepi@orangepi:~$ ls /dev/ttyS5
/dev/ttyS5
```

4) Then start to test the uart5 interface, first use the DuPont line to short the rx and tx of the uart5 interface to be tested

	uart5
tx pin	Corresponding to pin 8 in 26pin
rx pin	Corresponding to pin 10 in 26pin

5) Finally, you can run the **serialTest.py** program in the examples to test the loopback function of the serial port. If you can see the following print, it means that the loopback test of the serial port is normal

```
root@orangepi:~/wiringOP-Python# cd examples
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device "/dev/ttyS5"        # linux6.1 使用
root@orangepi:~/wiringOP-Python/examples# python3 serialTest.py --device "/dev/ttyAS5"       # linux5.4 使用


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

3.21. Hardware Watchdog Test

The watchdog_test program is pre-installed in the linux system released by Orange PI, which can be tested directly.。

The method of running the watchdog_test program is shown below：

a. The second parameter 10 indicates the counting of the door dog. If the dog is not fed in this time, the system will restart
b. We can feed the dog by pressing any key (except ESC) on the keyboard. After the dog is fed, the program will print a line of Keep Alive to indicate that the dog

is successful.

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

3.22. Check the chipid of the H618 chip

The command of the H618 chip Chipid is shown below. The chipid of each chip is different, so you can use Chipid to distinguish multiple development boards.

```
orangepi@orangepi:~$ cat /sys/class/sunxi_info/sys_info | grep "chipid"
sunxi_chipid            : 338020004c00480801474788246 81ed1
```

3.23. Python related instructions

3.23.1. Python source code compilation and installation method

If the Python version in the Ubuntu or Debian system software warehouse does not meet the requirements of development, if you want to use the latest version of Python, you can use the following method to download the source code package of Python to compile and install the latest version of Python.

The following demonstration is the latest version of the compilation and installation of Python3.9. If you want to compile and install other versions of Python, the method is the same (need to download the source code corresponding to the Python you want to install).

1) First install the dependency package required to compile Python

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y build-essential zlib1g-dev  \
libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libsqlite3-dev  \
libreadline-dev libffi-dev curl libbz2-dev
```

2) Then download the latest version of the Python3.9 source code and decompress it

```
orangepi@orangepi:~$ wget  \
https://www.python.org/ftp/python/3.9.10/Python-3.9.10.tgz
orangepi@orangepi:~$ tar xvf Python-3.9.10.tgz
```

3) Then run the configuration command

```
orangepi@orangepi:~$ cd Python-3.9.10
orangepi@orangepi:~$ ./configure --enable-optimizations
```

4) Then compile and install python3.9, the compilation time takes about half an hour

```
orangepi@orangepi:~$ make -j4
orangepi@orangepi:~$ sudo make altinstall
```

5) After installation, you can use the following command to view the Python version number just installed

```
orangepi@orangepi:~$ python3.9 --version
Python 3.9.10
```

6) Then update pip

```
orangepi@orangepi:~$ /usr/local/bin/python3.9 -m pip install --upgrade pip
```

3. 23. 2. Python to replace the pip source method

The Linux system PIP's default source is the official source of Python, but the source of the official Python official visits is very slow, and the Python software packaging failure is often caused by network reasons. So when installing the Python library with PIP, remember to replace the pip source

1) First install **python3-pip**

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3-pip
```

2) Method of permanent replacement of pip source under Linux

a. Create a new **~/.pip** directory first, then add the **pip.conf** configuration file, and set the source of the pip as Tsinghua source

```
orangepi@orangepi:~$ mkdir -p ~/.pip
orangepi@orangepi:~$ cat <<EOF > ~/.pip/pip.conf
[global]
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
EOF
```

b. Then install the Python library with pip3, it will be fast

3) The method of temporarily replacing the pip source under Linux, the **<packagename>** needs to be replaced with a specific package name

```
orangepi@orangepi:~$ pip3 install <packagename> -i  \
https://pypi.tuna.tsinghua.edu.cn/simple --trusted-host pypi.tuna.tsinghua.edu.cn
```

3. 24. The Method Of Installing The Docker

Orange Pi's Linux image has been pre-installed with Docker, but the Docker service has not been opened by default. Use **enable_docker.sh** script to enable the docker service, and then you can start using the docker command, and the docker service will be automatically activated at the next start of the system.

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

You can use the following command to test the docker,If you can run **hello-world** to indicate that docker can be used normally.

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

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

When using the docker command, if it prompts **permission denied**, please add the current user to the docker user group, so that the docker command can be run without sudo。

```
orangepi@orangepi:~$ sudo usermod -aG docker $USER
```

Note: You need to log out of the system and log in again to take effect, and restarting the system is also OK.

3. 25. How to install Home Assistant

Note that only the method of installing Home Assistant in Ubuntu or Debian system will be provided here. For detailed usage of Home Assistant, please refer to official documents or corresponding books。

3. 25. 1. Install via docker

1) First of all, please install docker and make sure that docker can run normally. The Method Of Installing The Docker, please refer to the instructions in the section on how to install Docker。

2) Then you can search for the docker image of Home Assistant

```
orangepi@orangepi:~$ docker search homeassistant
```

3) Then use the following command to download the docker image of Home Assistant to the local. The size of the image is about 1GB, and the download time will be relatively long. Please wait patiently for the download to complete

```
orangepi@orangepi:~$ docker pull homeassistant/home-assistant
Using default tag: latest
latest: Pulling from homeassistant/home-assistant
be307f383ecc: Downloading
5fbc4c07ac88: Download complete
..... (omit some output)
```

```
3cc6a1510c9f: Pull complete
7a4e4d5b979f: Pull complete
Digest:
sha256:81d381f5008c082a37da97d8b08dd8b358dae7ecf49e62ce3ef1eeaefc4381bb
Status: Downloaded newer image for homeassistant/home-assistant:latest
docker.io/homeassistant/home-assistant:latest
```

4) Then you can use the following command to view the docker image of Home Assistant just downloaded

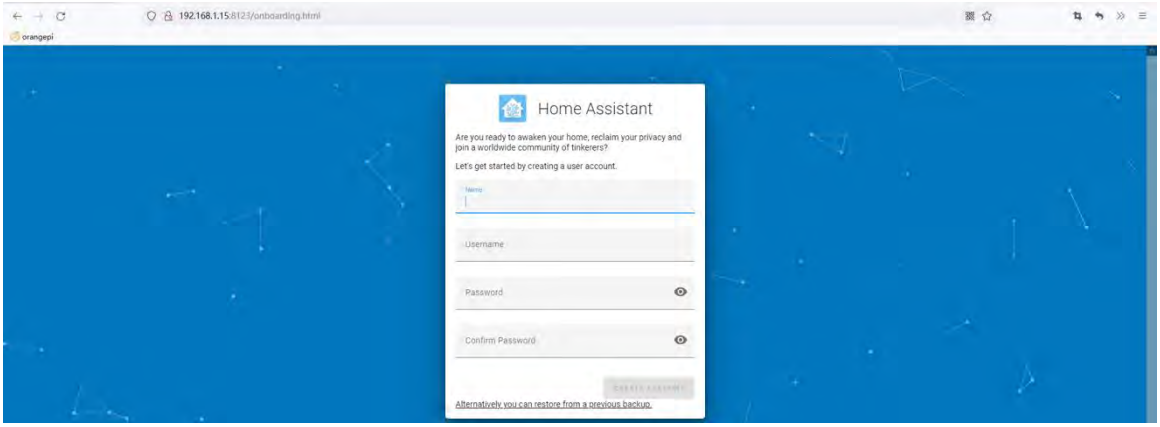
```
orangepi@orangepi:~$ docker images homeassistant/home-assistant
REPOSITORY                  TAG       IMAGE ID       CREATED         SIZE
homeassistant/home-assistant  latest    bfa0ab9e1cf5   2 months ago    1.17GB
```

5) Now you can run the docker container of Home Assistant

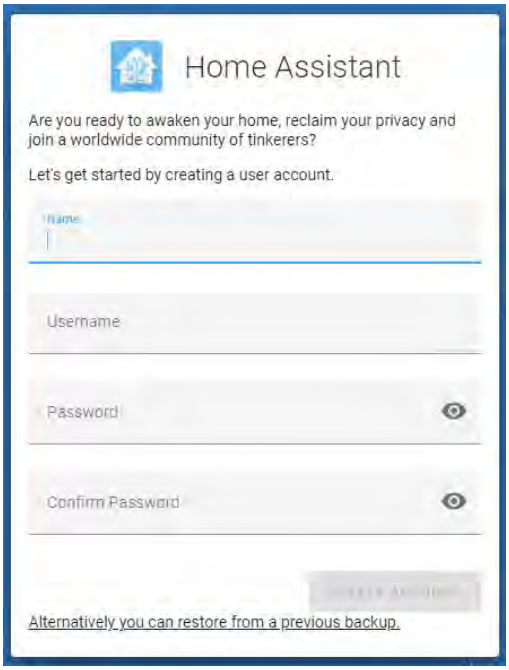
```
orangepi@orangepi:~$ docker run -d \
  --name homeassistant \
  --privileged \
  --restart=unless-stopped \
  -e TZ=Asia/Shanghai \
  -v /home/orangepi/home-assistant:/config \
  --network=host \
  homeassistant/home-assistant:latest
```

6) Then enter【IP address of the development board: 8123】in the browser to see the Home Assistant interface

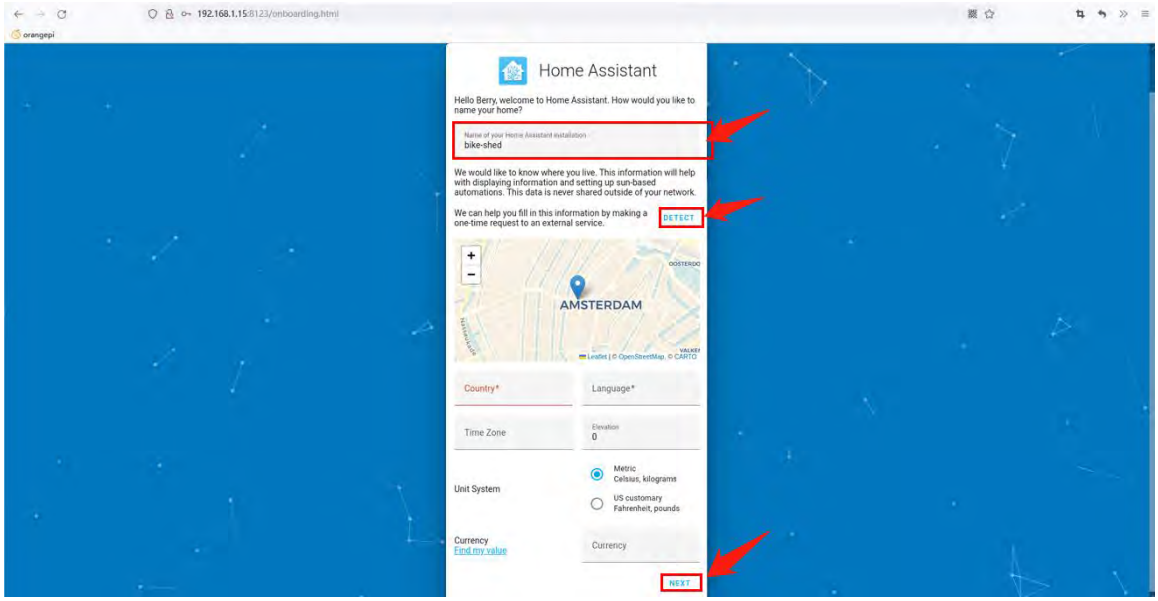
It takes a while for the Home Assistant container to start. If the following interface is not displayed normally, please wait for a few seconds and then refresh. If the following interface is not displayed normally after waiting for more than one minute, it means that there is a problem with the installation of Home Assistant. At this time, you need to check whether there is a problem with the previous installation and setting process。



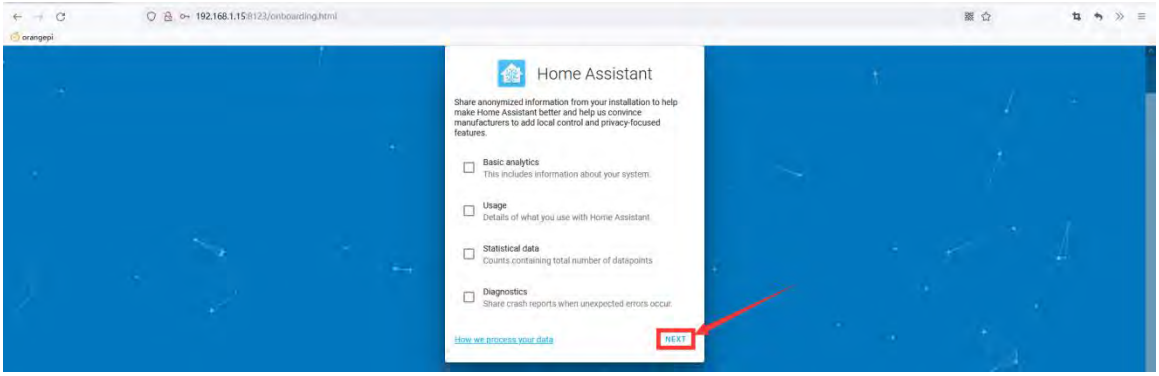
7) Then enter **your name**, **user name** and **password** and click **Create Account**



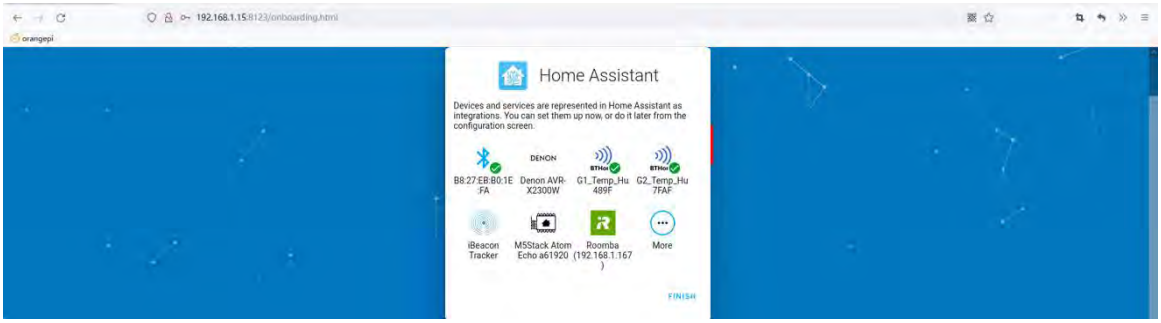
8) Then follow the interface prompts to set according to your preferences, and then click Next



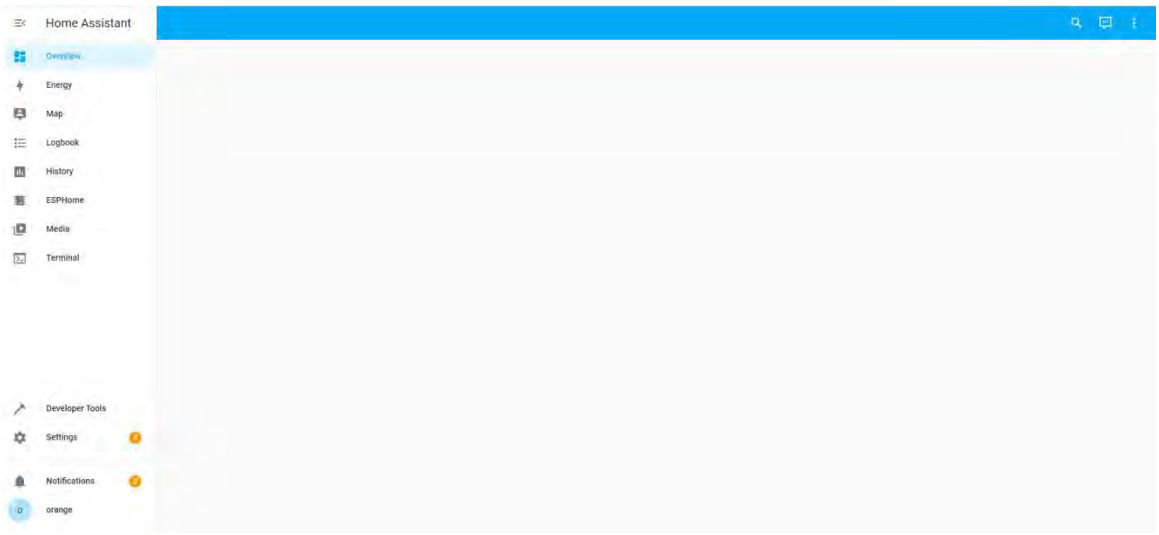
9) Then click Next



10) Then click Finish



11) The final main interface displayed by Home Assistant is shown in the figure below



12) How to stop the Home Assistant container

a. The command to view the docker container is as follows

```
orangepi@orangepi:~$ docker ps -a
```

b. The command to stop the Home Assistant container is as follows

```
orangepi@orangepi:~$ docker stop homeassistant
```

c. The command to delete the Home Assistant container is as follows

```
orangepi@orangepi:~$ docker rm homeassistant
```

3. 25. 2. Install via python

Before installation, please replace the source of pip with a domestic source to speed up the installation of the Python package. For the configuration method, see the instructions in the section on Python to replace the pip source method.

1) First install the dependency package

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3 python3-dev python3-venv    \
python3-pip libffi-dev libssl-dev libjpeg-dev zlib1g-dev autoconf build-essential    \
libopenjp2-7 libtiff5 libturbojpeg0-dev tzdata
```

If it is debian12, please use the following command：

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y python3 python3-dev python3-venv    \
python3-pip libffi-dev libssl-dev libjpeg-dev zlib1g-dev autoconf build-essential    \
```

libopenjp2-7 libturbojpeg0-dev tzdata

2) Then you need to compile and install Python3.9. For the method, please refer to the section on **how to compile and install Python source code**

The default Python version of Debian Bullseye is Python3.9, so there is no need to compile and install。

The default Python version of Ubuntu Jammy is Python3.10, so there is no need to compile and install。

The default Python version of Debian Bookworm is Python3.11, so there is no need to compile and install。

3) Then create a Python virtual environment

Debian Bookworm is python3.11, please remember to replace the corresponding command。

```
orangepi@orangepi:~$ sudo mkdir /srv/homeassistant
orangepi@orangepi:~$ sudo chown orangepi:orangepi /srv/homeassistant
orangepi@orangepi:~$ cd /srv/homeassistant
orangepi@orangepi:~$ python3.9 -m venv .
orangepi@orangepi:~$ source bin/activate
(homeassistant) orangepi@orangepi:/srv/homeassistant$
```

4) Then install the required Python package

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ python3 -m pip install wheel
```

5) Then you can install Home Assistant Core

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ pip3 install homeassistant
```

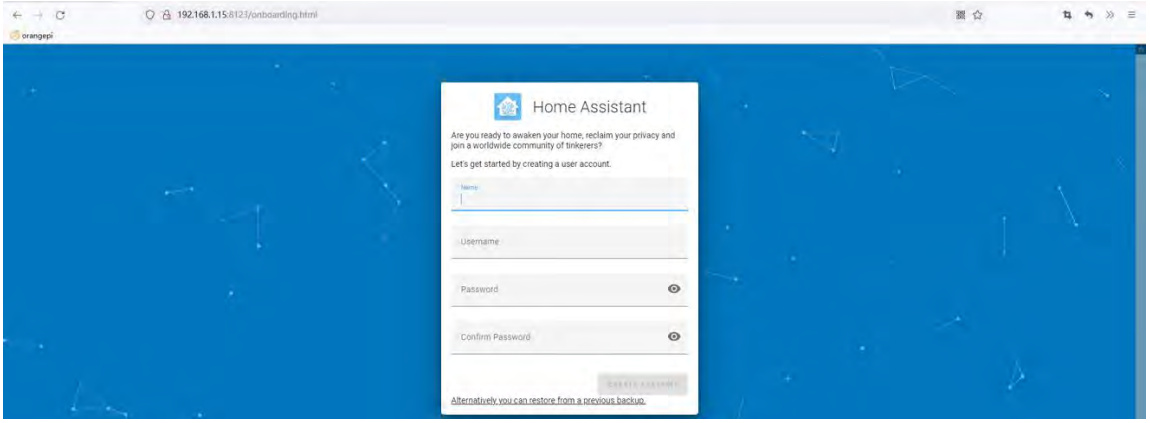
6) Then enter the following command to run Home Assistant Core

```
(homeassistant) orangepi@orangepi:/srv/homeassistant$ hass
```

7) Then enter **[IP address of the development board: 8123]** in the browser to see the interface of Home Assistant

When running the hass command for the first time, you will download and install and cache some necessary libraries and dependencies to run. This process

may take a few minutes. Note that at this time, the interface of Home Assistant cannot be seen in the browser. Please wait for a while before refreshing。



3. 26. OpenCV installation method

3. 26. 1. Use apt to install OpenCV

1) The installation command is shown below

```
orangepi@orangepi:~$ sudo apt-get update
orangepi@orangepi:~$ sudo apt-get install -y libopencv-dev python3-opencv
```

2) Then use the following command to print the output of the version number of OpenCV, indicating that the installation of OpenCV is successful

a. The version of OpenCV in ubuntu22.04 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.5.4
```

b. The version of OpenCV in Ubuntu 20.04 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.2.0
```

c. The version of OpenCV in Debian11 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.5.1
```

d. The version of OpenCV in Debian12 is shown below：

```
orangepi@orangepi:~$ python3 -c "import cv2; print(cv2.__version__)"
4.6.0
```

3. 27. The installation method of the aapanel Linux panel

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

1) First of all, you need to expand the/**tmp** memory. After setting, you need to **restart the linux system of the development board，** and the command is shown below：

```
orangepi@orangepi:~$ sudo sed -i 's/nosuid/&,size=2G/' /etc/fstab
orangepi@orangepi:~$ sudo reboot
```

2) After restarting, you can see that the size of the/**tmp** space has become 2G

```
orangepi@orangepi:~$ df -h | grep "/tmp"
tmpfs                   2.0G     12K  2.0G     1% /tmp
```

3) Then enter the following command in the Linux system to start the installation of the aapanel

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

4) Then the aapanel installation program reminds whether to install the **Bt-Panel** to the/**www** folder, and enter Y at this time

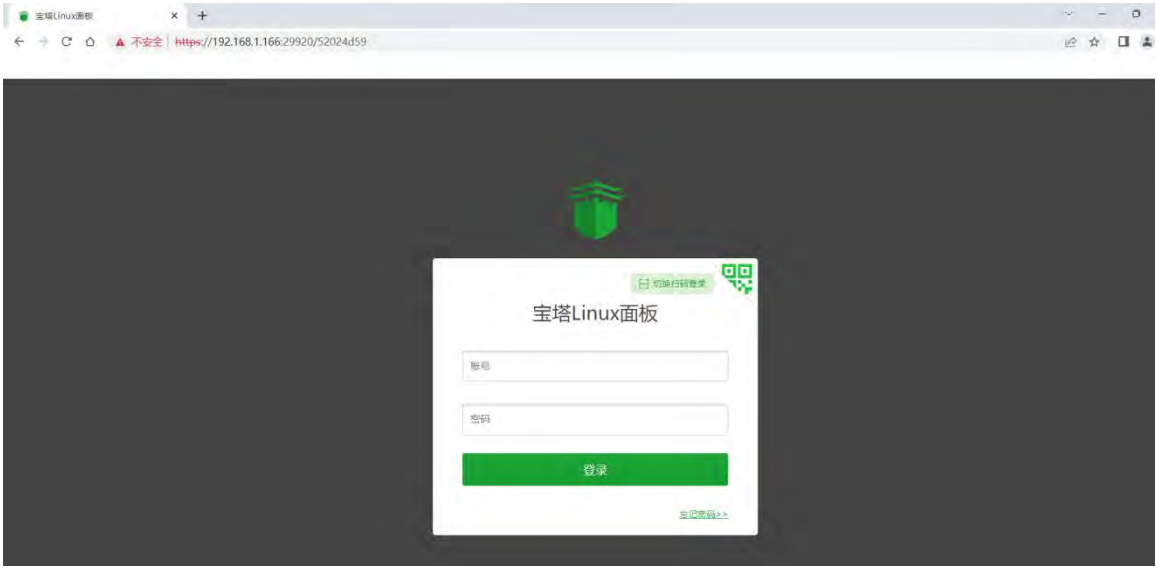
```
+----------------------------------------------------------------------
| Bt-WebPanel FOR CentOS/Ubuntu/Debian
+----------------------------------------------------------------------
| Copyright © 2015-2099 BT-SOFT(http://www.bt.cn) All rights reserved.
+----------------------------------------------------------------------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

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

5) Then you have to wait patiently. When you see the printing information below the terminal output, it means that the pagoda has been installed. The entire installation

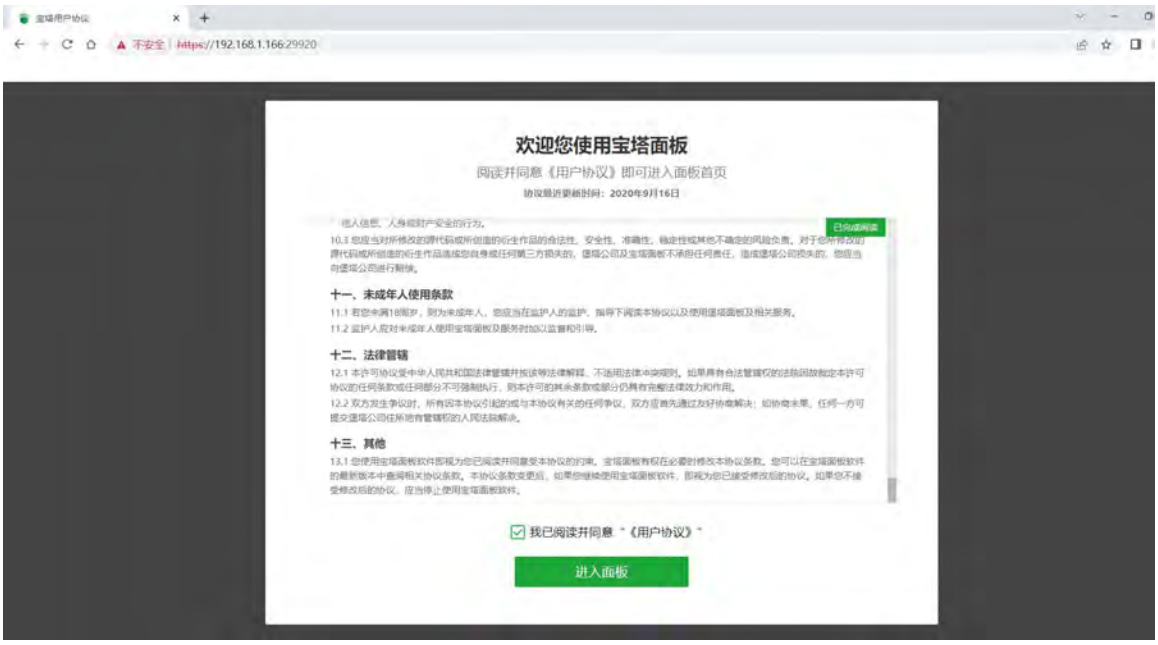
process takes about 34 minutes. There may be some differences according to the difference in network speed

```
==========================================================================
Congratulations! Installed successfully!
==========================================================================
外网面板地址: https://183.15.204.194:29920/52024d59
内网面板地址: https://192.168.1.166:29920/52024d59
username: 4qhagfrc
password: 27b2d026
If you cannot access the panel,
release the following panel port [29920] in the security group
若无法访问面板，请检查防火墙/安全组是否有放行面板[29920]端口
因已开启面板自签证书，访问面板会提示不匹配证书，请参考以下链接配置证书
https://www.bt.cn/bbs/thread-105443-1-1.html
==========================================================================
Time consumed: 34 Minute!
orangepi@orangepi:~$
```

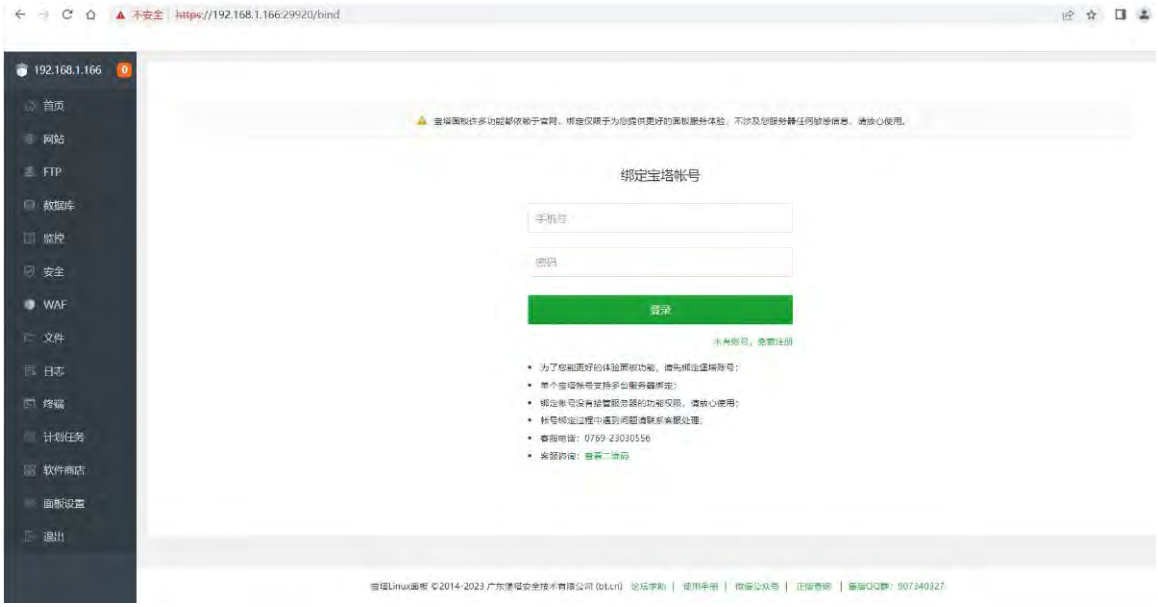
6) At this time, enter the **panel address** displayed above in the browser to open the login interface of the aapanel Linux panel, and then enter the **username** and **password** displayed in the corresponding position to log in to the aapanel



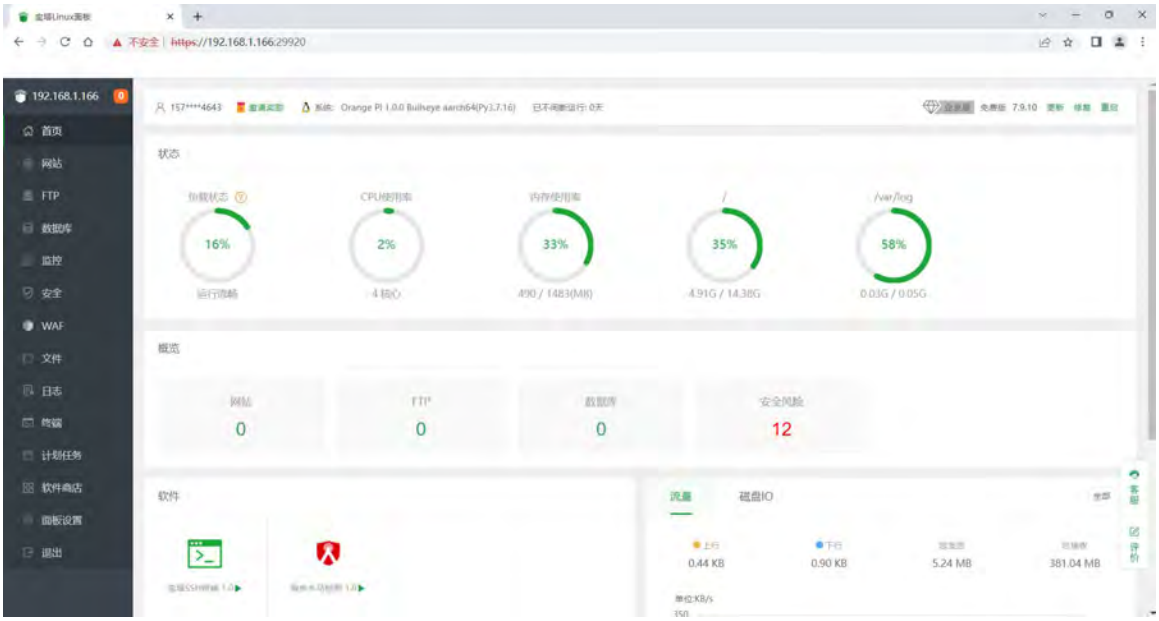
7) After successfully logging in to the aapanel, the following welcome interface will pop up. First, please take the intermediate user notice to read to the bottom, and then you can choose "I have agreed and read" User Agreement ", and then click" Enter the panel " You can enter the aapanel



8) After entering the aapanel, you will first prompt that you need to bind the account of the aapanel official website. If you do n't have an account, you can go to the aapanel 's official website (**https://www.bt.cn**) to register one



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



10) For more functions of the aapanel, please refer to the following information to explore by yourself

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

3. 28. "face_recognition" The installation and testing method of the face recognition library

Note that the contents of this section are tested in the Linux system of the desktop version, so please make sure that the system used by the development board is a desktop version system。

In addition, the installation test below is performed under OrangePi users. Please keep the environment consistent.

Debian12 is currently not adapted.

The address of the source code warehouse of Face_recognition is：
https://github.com/ageitgey/face_recognition

Face_recognition Chinese version of the explanation document is：
https://github.com/ageitgey/face_recognition/blob/master/README_Simplified_Chi

nese.md

3.28.1. The method of using script to automatically install face_recognition

1) First open a terminal on the desktop**, then download face_recognition_install.sh**

```
orangepi@orangepi:~/Desktop$ wget \
https://gitee.com/leeboby/face_recognition_install/raw/master/face_recognition_install.sh
```

2) Then execute the following command to start the installation **face_recognition**

```
orangepi@orangepi:~/Desktop$ bash face_recognition_install.sh
```

3) After face_recognition is installed, it will automatically download the source code of face_recognition, and then automatically run some examples in face_recognition. If you can finally see the following pictures popping up on the desktop, it means that the face_recognition installation test is successful.

3.28.2. Manually install face_recognition

1) First create a new **~/.pip** directory, then add the **pip.conf** configuration file, and set the mirror source of pip to Tsinghua source in it, the commands to be executed are as follows：

```
orangepi@orangepi:~$ mkdir -p ~/.pip
orangepi@orangepi:~$ cat <<EOF > ~/.pip/pip.conf
[global]
```

```
timeout = 6000
index-url = https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
EOF
```

2) Then install dependencies

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y python3-pip libopencv-dev \
python3-opencv imagemagick python3-scipy python3-setuptools python3-wheel \
python3-dev cmake python3-testresources
```

3) Then update pip3

```
orangepi@orangepi:~$ python3 -m pip install -U pip setuptools wheel
```

4) Before installing **face_recognition**, you first need to install the **dlib** library. Since the compilation and installation of the dlib library on the development board is relatively slow, I saved a compiled dlib whl file on **gitee**, and you can install it directly after downloading. The download address of the dlib whl file is as follows：

```
https://gitee.com/leeboby/python_whl
```

a. First download the python_whl warehouse to the Linux system of the development board

```
orangepi@orangepi:~$ git clone --depth=1 https://gitee.com/leeboby/python_whl
```

b. In the python_whl folder, you can see that there are multiple versions of dlib installation packages. The Linux systems corresponding to different versions of dlib are as follows：

Ubuntu20.04	**dlib-19.24.0-cp38-cp38-linux_aarch64.whl**
Ubuntu22.04	**dlib-19.24.0-cp310-cp310-linux_aarch64.whl**
Debian11	**dlib-19.24.0-cp39-cp39-linux_aarch64.whl**

c. Then you can start installing dlib, the command is as follows

a) Ubuntu20.04

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install dlib-19.24.0-cp38-cp38-linux_aarch64.whl
```

b) Ubuntu22.04

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install dlib-19.24.0-cp310-cp310-linux_aarch64.whl
```

c) Debian11

```
orangepi@orangepi:~$ cd python_whl
orangepi@orangepi:~/python_whl$ python3 -m pip install dlib-19.24.0-cp39-cp39-linux_aarch64.whl
```

d. After installation, if the version number of dlib can be printed normally by using the following command, it means that dlib is installed correctly

```
orangepi@orangepi:~/python_whl$ python3 -c "import dlib; print(dlib.__version__)"
19.24.0
```

5) Then install **face_recognition_models-0.3.0-py2.py3-none-any.whl**

```
orangepi@orangepi:~/python_whl$ python3 -m pip install face_recognition_models-0.3.0-py2.py3-none-any.whl
```

6) Then install **face_recognition**

```
orangepi@orangepi:~$ python3 -m pip install face_recognition
```

7) Then you need to **reopen a terminal** to find and run the two commands **face_detection** and **face_recognition**

a. The face_recognition command is used to recognize whose face is in a single image or a folder of images.

b. The face_detection command is used to locate the face in a single picture or a picture folder

```
orangepi@orangepi:~$ which face_detection
/usr/local/bin/face_detection
orangepi@orangepi:~$ which face_recognition
/usr/local/bin/face_recognition
```

If you can't find the above two commands after reopening the terminal, please try to manually import environment variables, and then test again

```
orangepi@orangepi:~$ export PATH=/home/orangepi/.local/bin:$PATH
```

3. 28. 3. The test method of face_recognition

Note that the following operations are demonstrated on the desktop, so please connect the HDMI monitor first, or use NoMachine/VNC to remotely log in to the Linux desktop to test。

1) There are some sample codes in the source code of **face_recognition**, which we can use directly for testing. The download address of the source code of face_recognition is

as follows：

a. GitHub official download address

```
orangepi@orangepi:~$ git clone https://github.com/ageitgey/face_recognition.git
```

b. Gitee image download address

```
orangepi@orangepi:~$ git clone https://gitee.com/leeboby/face_recognition.git
```

2) The path to the face_recognition sample code is as follows

```
face_recognition/examples
```

3) The Chinese description document link of face_recognition is as follows, please read it carefully before using face_recognition

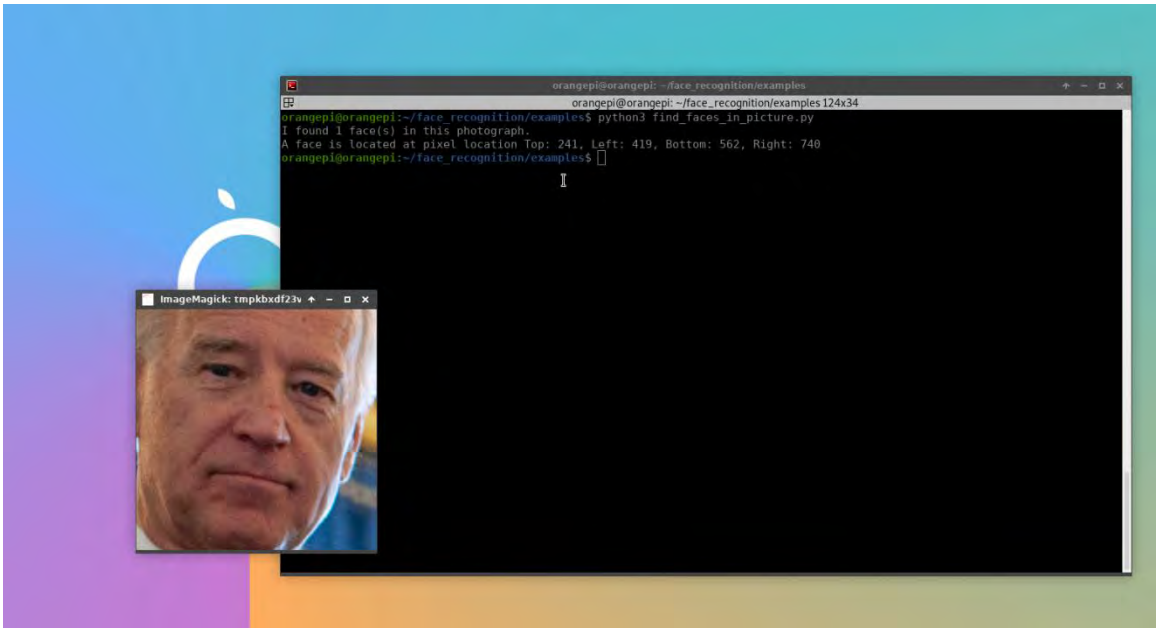
```
https://github.com/ageitgey/face_recognition/blob/master/README_Simplified_Chinese.md
```

4) **find_faces_in_picture.py** is used to locate the position of the face in the picture, the test steps are as follows

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 find_faces_in_picture.py
I found 1 face(s) in this photograph.
A face is located at pixel location Top: 241, Left: 419, Bottom: 562, Right: 740
```

b. Wait for a while and the following picture will pop up, which is the face located in the test picture

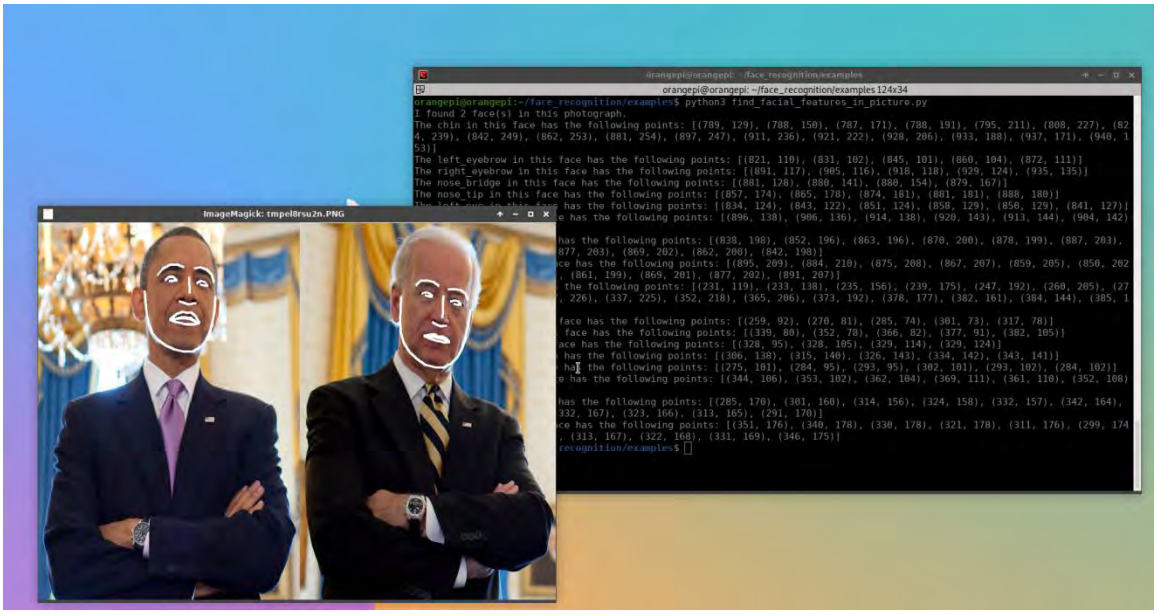


5) **find_facial_features_in_picture.py** is used to identify the key points of the face in a single picture, and the test steps are as follows

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 find_facial_features_in_picture.py
```

b. After waiting for a while, the following picture will pop up, and you can see that the outline of the face is marked

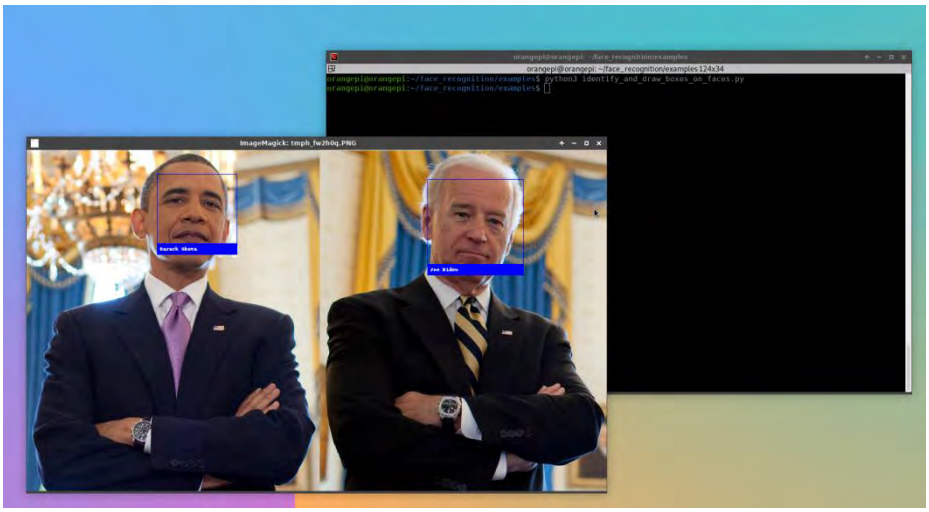


6) **identify_and_draw_boxes_on_faces.py** is used to identify faces and mark them with boxes. The test steps are as follows

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 identify_and_draw_boxes_on_faces.py
```

b. After waiting for a while, the following picture will pop up. You can see that the faces in the picture are marked with boxes, and the names of the characters are displayed correctly



7) **face_distance.py** is used to compare whether two faces belong to the same person at different precisions. First open a terminal, then enter the **face_recognition/examples** directory, and then execute the following command to see the output of the test

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 face_distance.py
The test image has a distance of 0.35 from known image #0
- With a normal cutoff of 0.6, would the test image match the known image? True
- With a very strict cutoff of 0.5, would the test image match the known image? True

The test image has a distance of 0.82 from known image #1
- With a normal cutoff of 0.6, would the test image match the known image? False
- With a very strict cutoff of 0.5, would the test image match the known image?
  False
```

8) **recognize_faces_in_pictures.py** is used to identify who the face in the unknown

picture is. First open a terminal, then enter the **face_recognition/examples** directory, and then execute the following command, and you can see the test results after waiting for a while

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ python3 recognize_faces_in_pictures.py
Is the unknown face a picture of Biden? False
Is the unknown face a picture of Obama? True
Is the unknown face a new person that we've never seen before? False
```

9) **facerec_from_webcam_faster.py** is used to recognize the face in the USB camera, the test steps are as follows：

a. First, please insert the USB camera into the USB interface of the development board, and then use the **v4l2-ctl** (**note that l in v4l2 is a lowercase letter l, not the number 1**) command to check the serial number of the device node of the USB camera

```
orangepi@orangepi:~$ sudo apt update
orangepi@orangepi:~$ sudo apt install -y v4l-utils
orangepi@orangepi:~$ v4l2-ctl --list-devices
cedrus (platform:cedrus):
        /dev/video0

USB2.0 UVC PC Camera: USB2.0 UV (usb-5311000.usb-1):
        /dev/video1
        /dev/video2
```

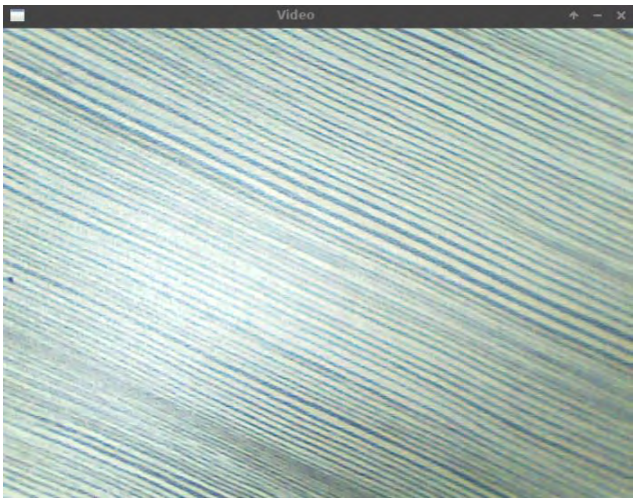
b. Then open a terminal on the desktop, enter the **face_recognition/examples** directory, and first modify the device serial number of the camera used in **facerec_from_webcam_faster.py.** For example, through the **v4l2-ctl --list-devices** command above, you can see that the USB camera is **/dev/video1**, then modify the **0** in **cv2.VideoCapture(0)** to **1**

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ vim facerec_from_webcam_faster.py
video_capture = cv2.VideoCapture(1)
```

c. Then execute the following command to run **facerec_from_webcam_faster.py**

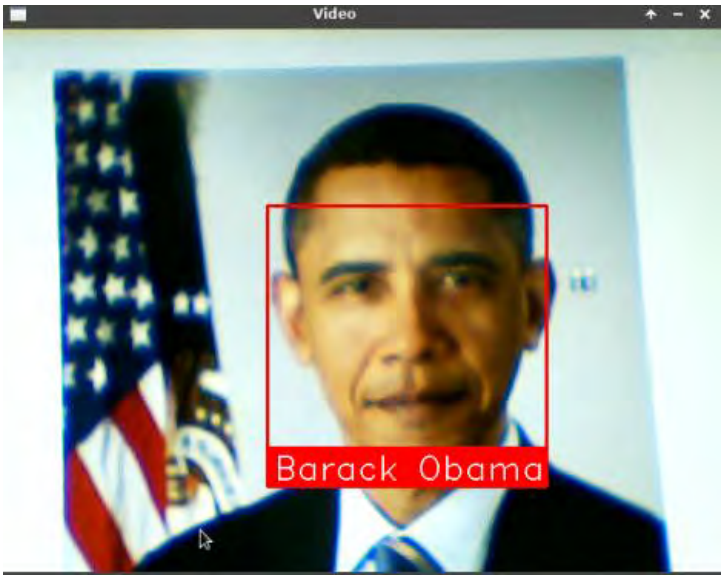
```
orangepi@orangepi:~/face_recognition/examples$ python3 facerec_from_webcam_faster.py
```

d. Wait for a while and the camera display screen will pop up



e. At this point, you can point the camera at yourself. When the camera detects a face, it will frame the detected face with a square. **Note that when detecting a face, the image displayed by the camera will be relatively slow, please do not move too fast**

f. You can also open a picture of Obama, and then use the camera to point at the opened picture. You can see that not only the face can be marked, but also the name of the detected face can be displayed correctly. **Note that when detecting a face, the image displayed by the camera will be relatively slow, please do not move too fast**



10) **web_service_example.py** is a very simple case of using a web service to upload a picture to run face recognition. The backend server will identify whether the picture is Obama, and output the recognition result as a json key-value pair. The test steps are as

follows：

a. Open a terminal on the desktop, then enter the **face_recognition/examples** directory, and then execute the following command (**if face_recognition is automatically installed using a script, then there is no need to install flask**)

```
orangepi@orangepi:~$ python3 -m pip install flask
orangepi@orangepi:~$ cd face_recognition/examples
root@orangepi:~/face_recognition/examples$ python3 web_service_example.py
 * Serving Flask app 'web_service_example' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses (0.0.0.0)
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://127.0.0.1:5001
 * Running on http://192.168.1.79:5001 (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 500-161-390
```

b. Then open another terminal and run the following command to return the result of image recognition (note that the execution path of the following command is **face_recognition/examples**)

```
orangepi@orangepi:~/face_recognition/examples$ curl -XPOST -F \
"file=@obama2.jpg" http://127.0.0.1:5001
{
  "face_found_in_image": true,
  "is_picture_of_obama": true
}
```

c. We can also copy the picture **face_recognition/examples/obama2.jpg** to other Linux computers. Of course, we can also prepare a picture named **obama2.jpg** by ourselves, and then use the following command on the Linux computer to remotely Use the service running on the development board to recognize faces (**note that the IP address in the command needs to be replaced with the IP address of the development board, and the file name after file needs to be replaced with the name of the picture you want to test**).

```
test@test:~$ curl -XPOST -F "file=@obama2.jpg" http://192.168.1.79:5001
{
  "face_found_in_image": true,
  "is_picture_of_obama": true
}
```

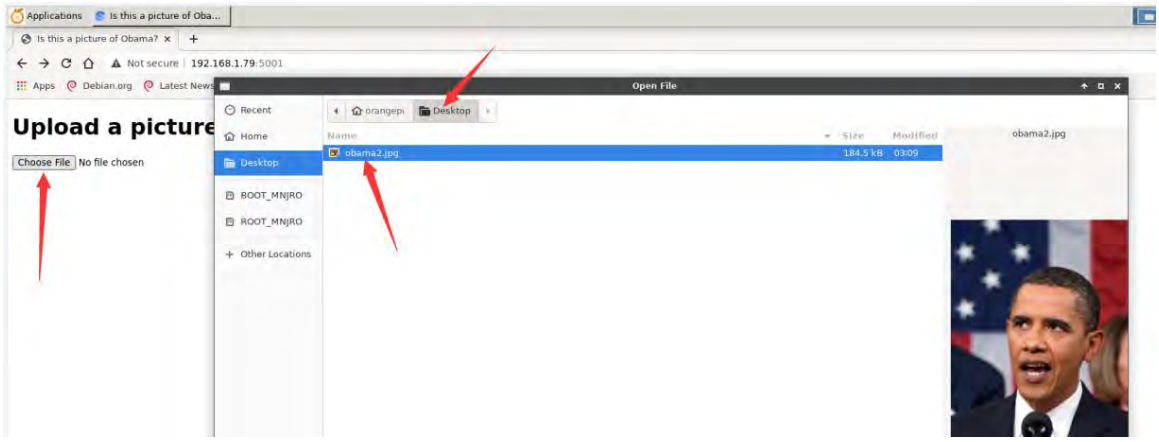
d. The method to test using the browser is as follows：

a) First open the browser, then enter the **IP address of the development board: 5001** in the address bar of the browser, and then you can see the following page

b) Then copy obama2.jpg to the desktop

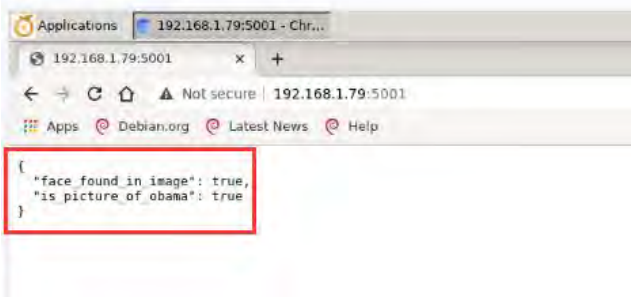
```
orangepi@orangepi:~/face_recognition/examples$ cp obama2.jpg /home/orangepi/Desktop/
```

c) Then select the picture you just copied in your browser



d) Then click **Upload** to upload the picture you just selected for face recognition

e) After waiting for a while, the detection result will be displayed



11) **face_detection** command test example

a. The face_detection command-line tool can locate the face position (output pixel coordinates) in a single picture or a picture folder. Use **face_detection --help** to view the help information of the face_detection command

```
orangepi@orangepi:~$ face_detection --help
Usage: face_detection [OPTIONS] IMAGE_TO_CHECK

Options:
  --cpus INTEGER    number of CPU cores to use in parallel. -1 means "use all in
                    system"
  --model TEXT      Which face detection model to use. Options are "hog" or
                    "cnn".
  --help            Show this message and exit.
```

b. An example of detecting a single image is shown below：

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ face_detection obama2.jpg
obama2.jpg,302,474,611,164
```

c. An example of using multiple cores to detect multiple images in parallel is shown below：

a) First enter the **face_recognition/examples** folder

b) Then create a new test folder
c) Then copy the jpg images to the test folder
d) Then use all cpus to run **face_detection** in parallel to check the pictures in the test folder, where **--cpus -1** means to use all cpus

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ mkdir test
orangepi@orangepi:~/face_recognition/examples$ cp *.jpg test
orangepi@orangepi:~/face_recognition/examples$ face_detection --cpus -1 test
test/obama-240p.jpg,29,261,101,189
test/obama_small.jpg,65,215,169,112
test/obama2.jpg,302,474,611,164
test/two_people.jpg,62,394,211,244
test/two_people.jpg,95,941,244,792
test/obama.jpg,136,624,394,366
test/obama-480p.jpg,65,507,189,383
test/obama-720p.jpg,94,751,273,572
test/obama-1080p.jpg,136,1140,394,882
test/biden.jpg,233,749,542,439
```

12) **face_recognition** command test example
a. **face_recognition** command-line tool can recognize whose face is in a single picture or a picture folder. Use **face_recognition --help** to view the help information of the face_recognition command

```
orangepi@orangepi:~$ face_recognition --help
Usage: face_recognition [OPTIONS] KNOWN_PEOPLE_FOLDER
IMAGE_TO_CHECK

Options:
  --cpus INTEGER                  number of CPU cores to use in parallel (can speed
                                  up processing lots of images). -1 means "use all in
                                  system"
  --tolerance FLOAT               Tolerance for face comparisons. Default is 0.6.
                                  Lower this if you get multiple matches for the same
                                  person.
  --show-distance BOOLEAN         Output face distance. Useful for tweaking tolerance
                                  setting.
```

--help	Show this message and exit.

b. First create a new face picture folder **known_people** with a known name, then copy two pictures to **known_people**, and then copy **obama2.jpg** to **unknown.jpg**, which is the picture we want to identify

```
orangepi@orangepi:~$ cd face_recognition/examples
orangepi@orangepi:~/face_recognition/examples$ mkdir known_people
orangepi@orangepi:~/face_recognition/examples$ cp biden.jpg obama.jpg known_people
orangepi@orangepi:~/face_recognition/examples$ cp obama2.jpg unkown.jpg
```

c. Then you can use the following command to identify the name of the person in the **unknown.jpg** picture, and you can see that the unknown.jpg picture is recognized as obama

```
orangepi@orangepi:~/face_recognition/examples$ face_recognition known_people \
unkown.jpg
unkown.jpg,obama
```

d. If we identify an irrelevant image, unknown_person will be displayed

```
root@orangepi:~/face_recognition/examples$ face_recognition known_people \
alex-lacamoire.png
alex-lacamoire.png,unknown_person
```

e. We can also create a new test folder and put multiple pictures in it, and then we can use all the CPUs to recognize all the pictures in parallel

```
orangepi@orangepi:~/face_recognition/examples$ mkdir test
orangepi@orangepi:~/face_recognition/examples$ cp *.jpg *.png test
orangepi@orangepi:~/face_recognition/examples$ face_recognition --cpus -1 \
known_people test
test/obama-240p.jpg,obama
test/alex-lacamoire.png,unknown_person
test/obama_small.jpg,obama
test/unkown.jpg,obama
test/obama2.jpg,obama
test/lin-manuel-miranda.png,unknown_person
test/two_people.jpg,biden
test/two_people.jpg,obama
test/obama-720p.jpg,obama
test/obama.jpg,obama
test/obama-480p.jpg,obama
```

```
test/biden.jpg,biden
test/obama-1080p.jpg,obama
```

3.29. Setting Chinese environment and installing Chinese input method

Note, please make sure that the Linux system used by the development board is a desktop system before installing the Chinese input method。

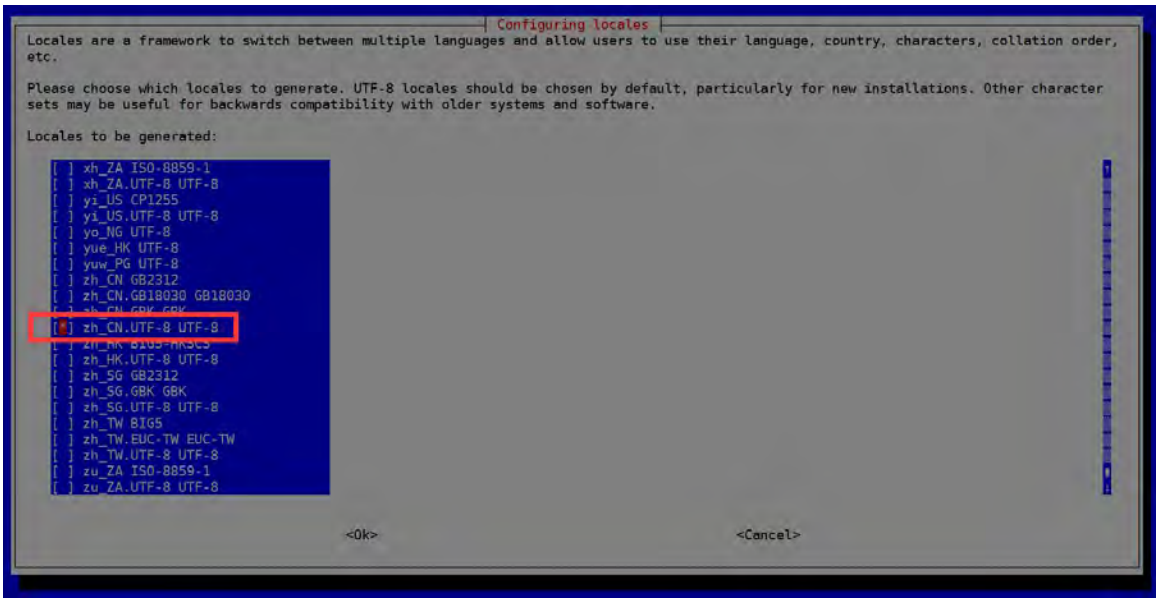
3.29.1. How to install Debian system

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

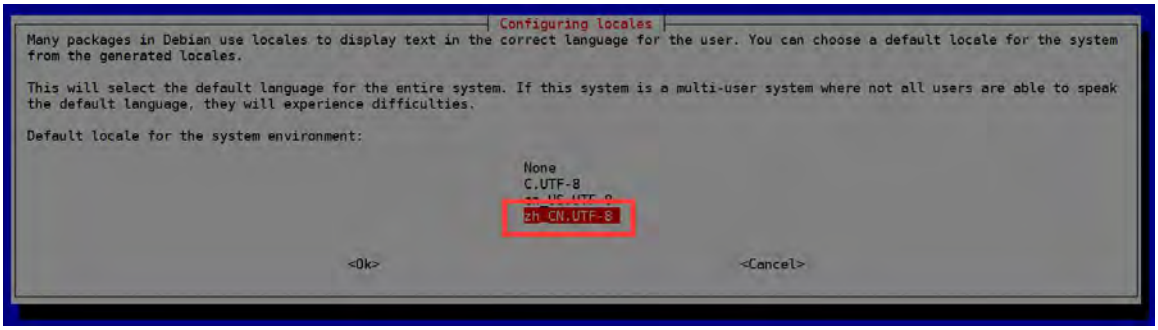
a. Enter the following command to start configuring the **locale**

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

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



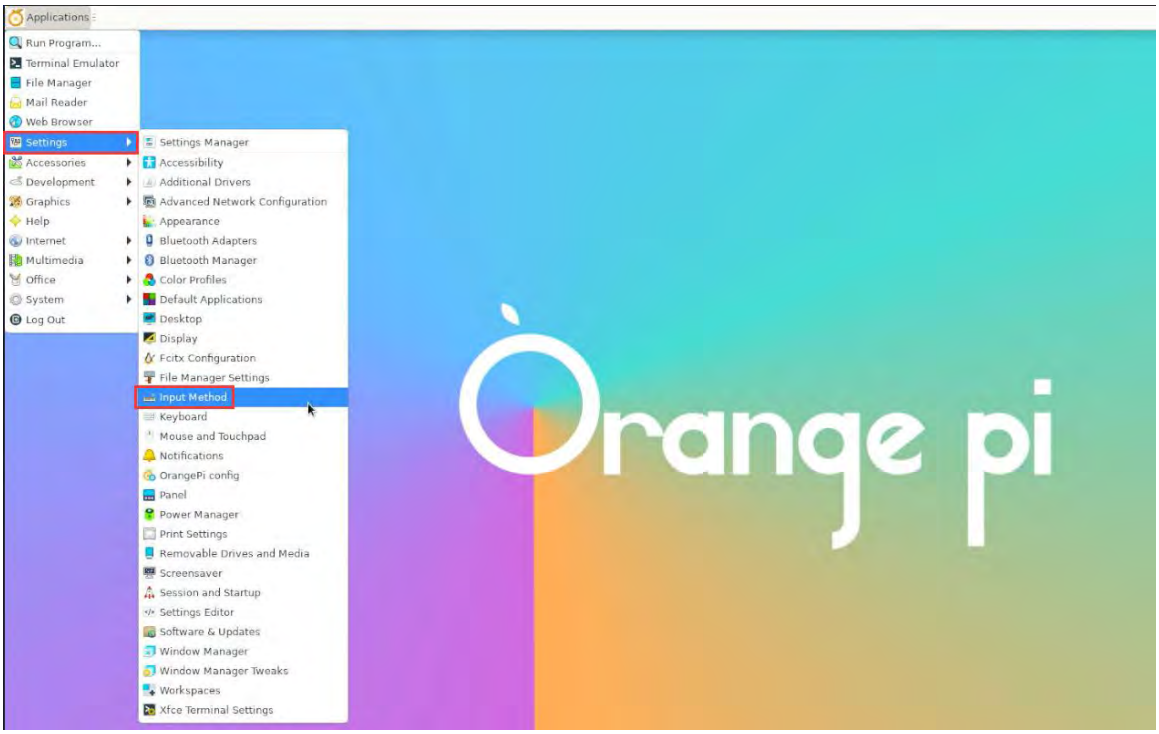
c. Then set the default **locale** to **zh_CN.UTF-8**



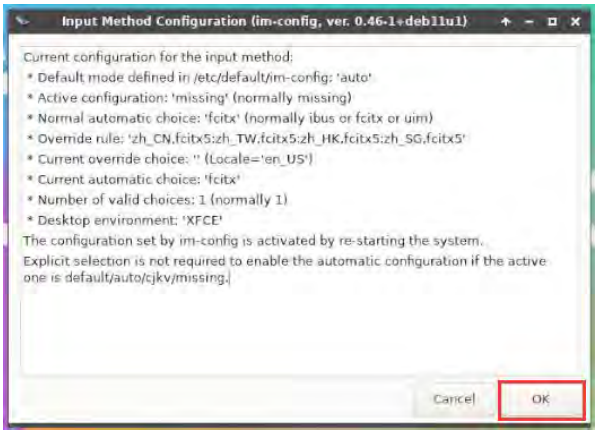
d. After exiting the interface, the **locale** setting will start, and the output displayed on the command line is as follows

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

2) Then open the **Input Method**

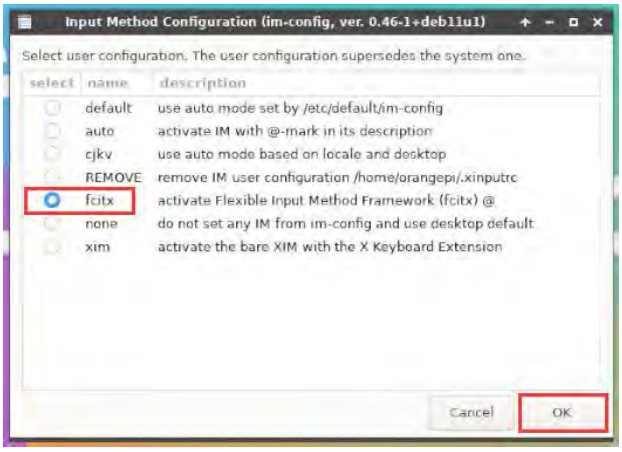


3) Then choose **OK**



4) Then choose**Yes**

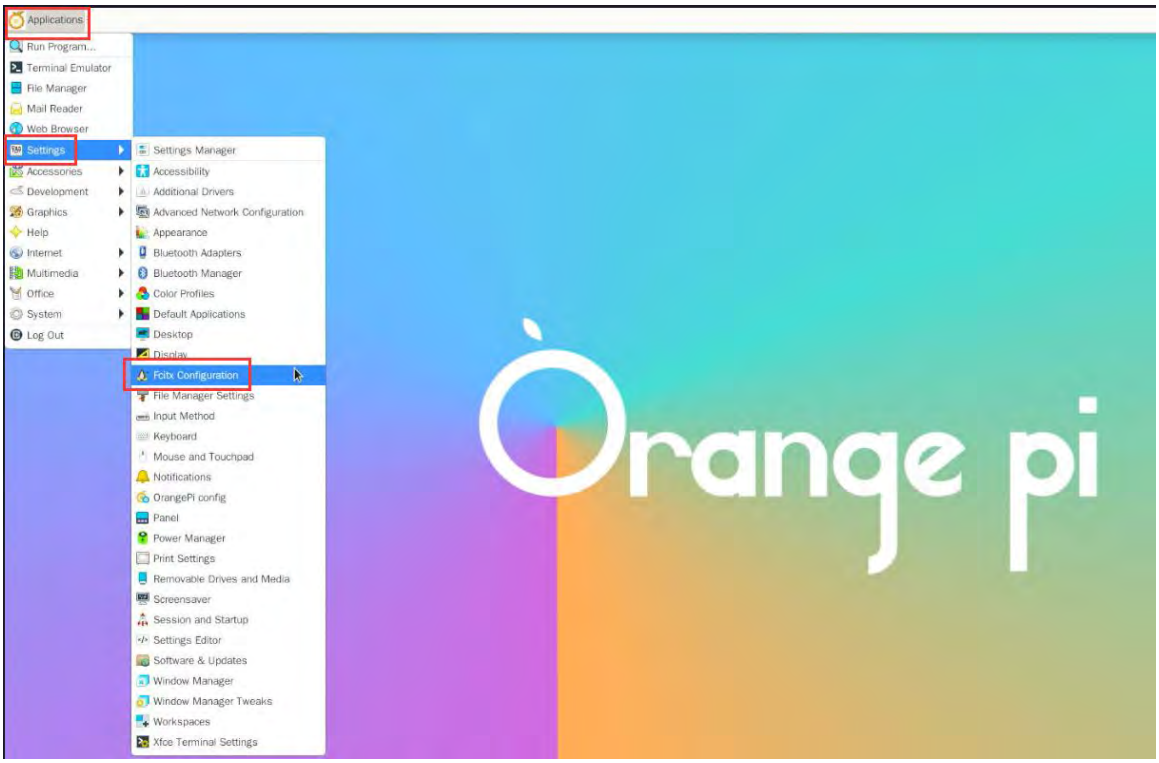
5) Then choose**fcitx**



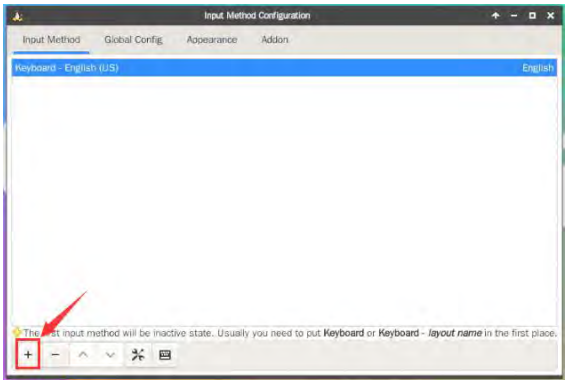
6) Then choose**OK**

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

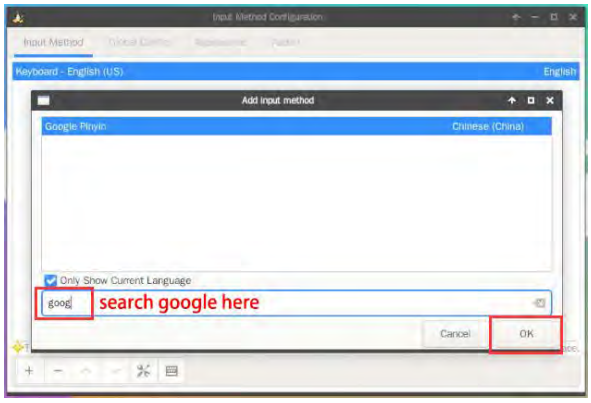
8) Then open **Fcitx configuration**



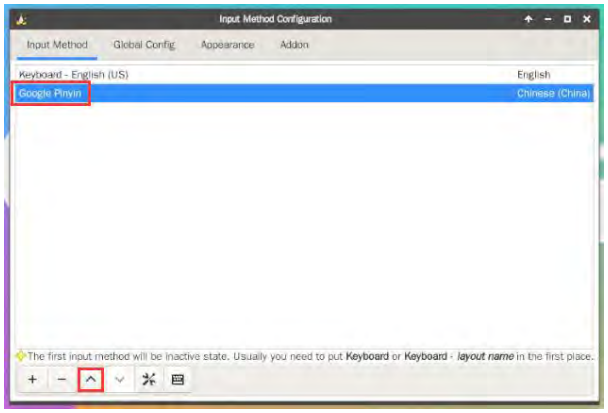
9) Then click the + sign in the position shown in the figure below

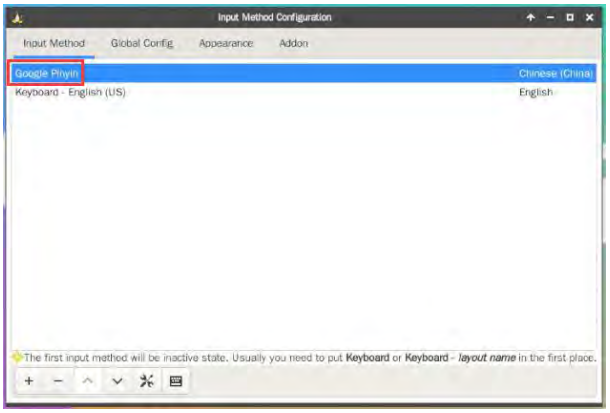


10) Then search **Google Pinyin** and click **OK**



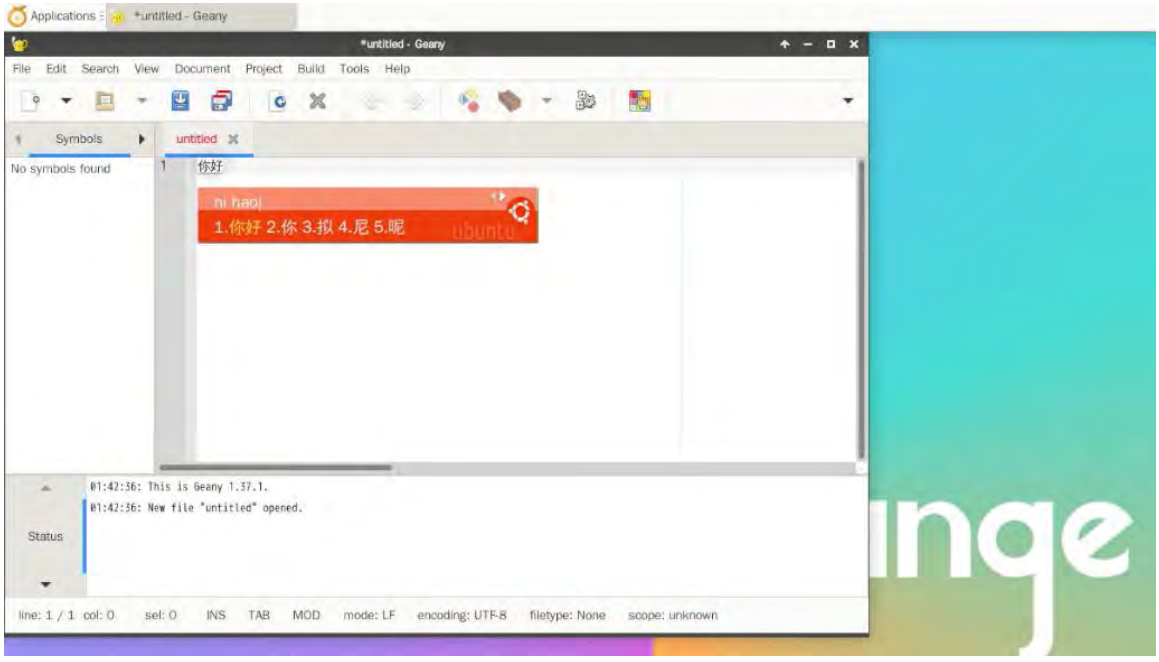
11) Then bring **Google Pinyin** to the front





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

13) The Chinese input method test is as follows

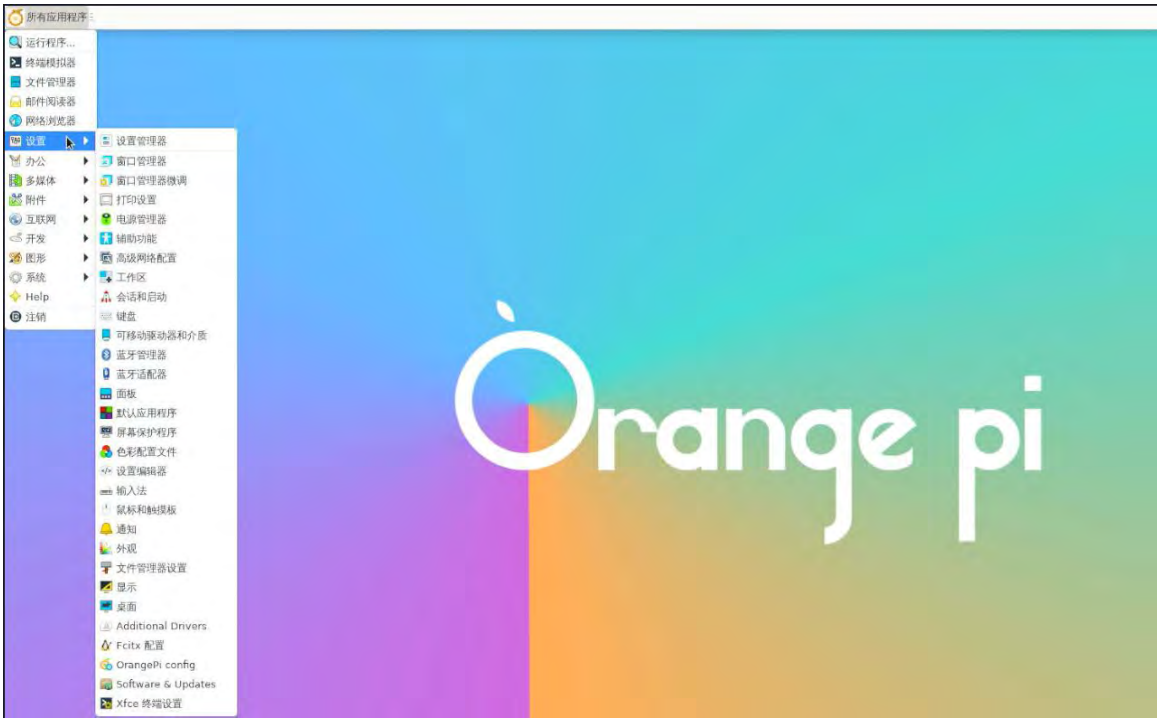


14) The Chinese and English input methods can be switched through the **Ctrl+Space** shortcut key

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

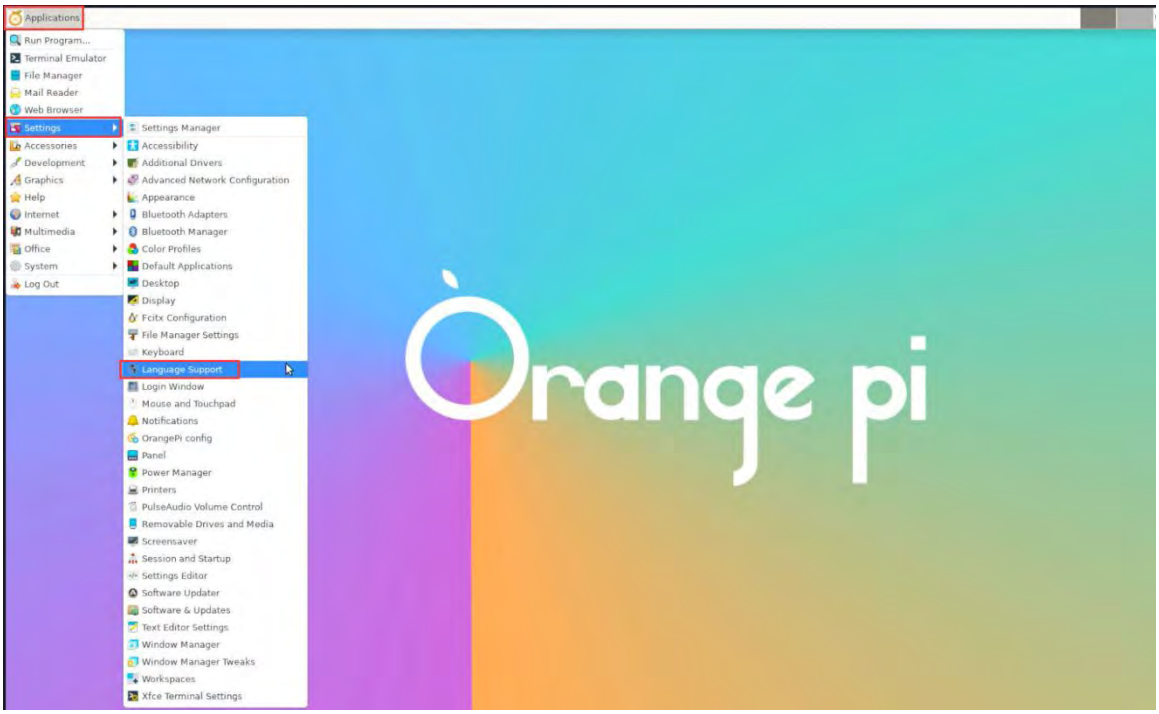
```
orangepi@orangepi:~$ sudo vim /etc/default/locale
#  File generated by update-locale
LC_MESSAGES=zh_CN.UTF-8
LANG=zh_CN.UTF-8
LANGUAGE=zh_CN.UTF-8
```

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

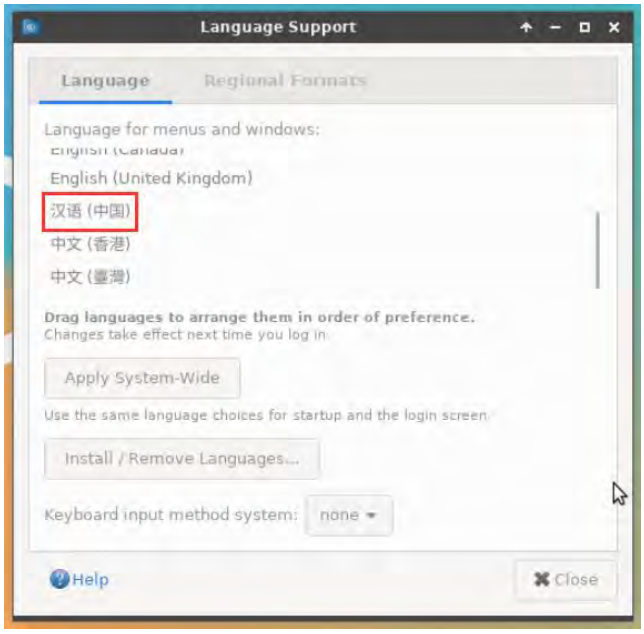


3. 29. 2. How to install Ubuntu 20.04 system

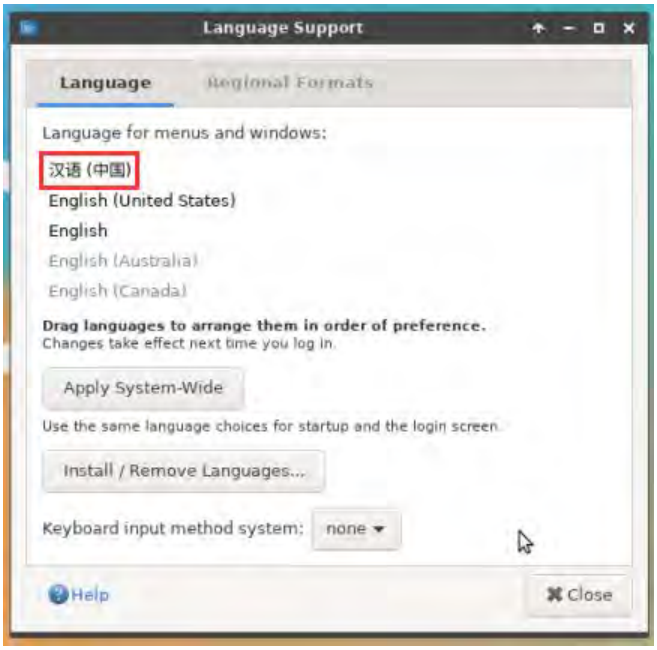
1) First open **Language Support**



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

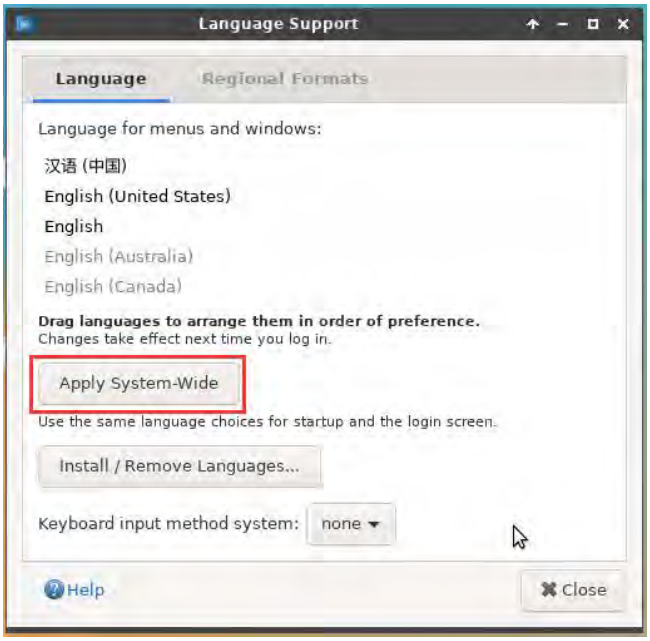


3) Then please use the left mouse button to select **Chinese (China)** and hold it down, then drag it up to the initial position, the display after dragging is as shown in the figure below：



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

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

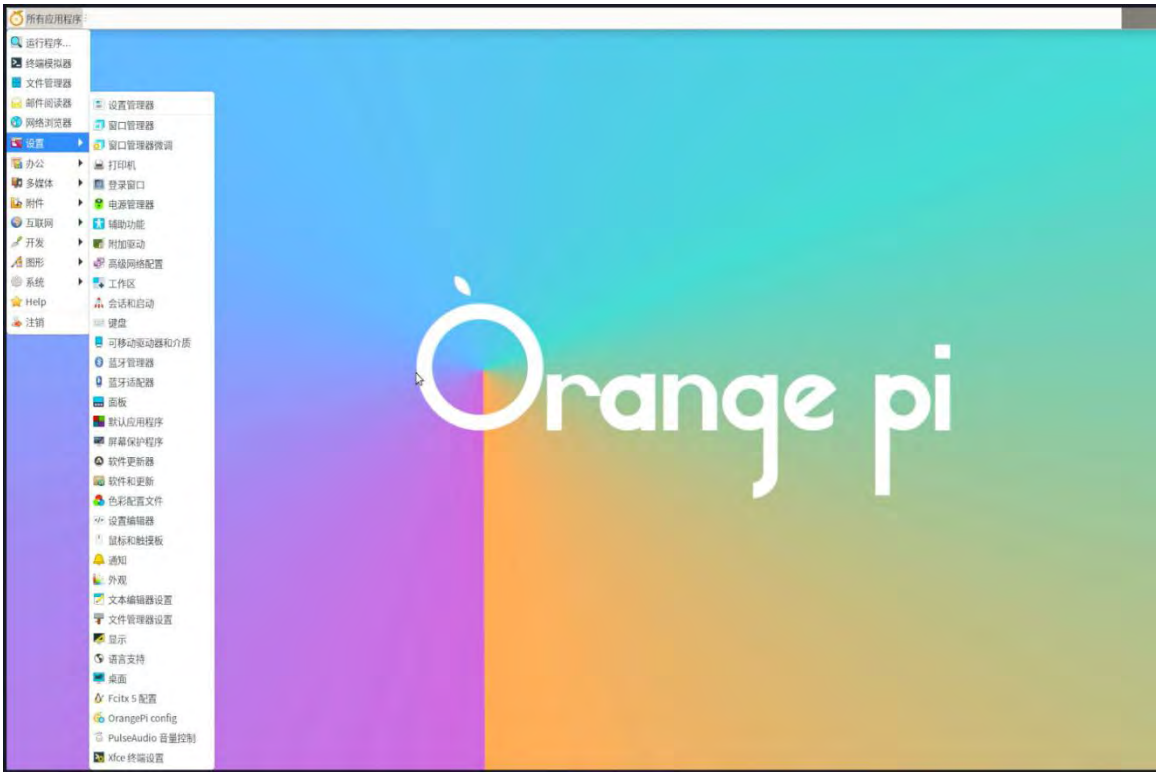


5) Then set**Keyboard input method system** to **fcitx**

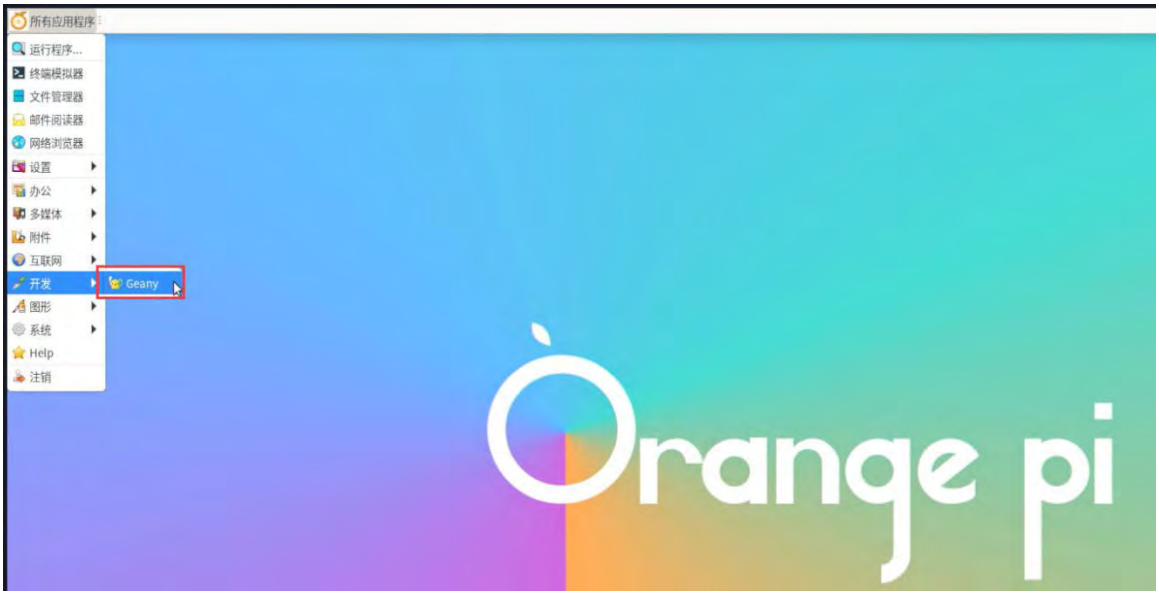
6) **Then restart the Linux system to make the configuration take effect**

7) After re-entering the system, **please choose not to ask me again** in the following interface, and then please decide whether the standard folder should also be updated to Chinese according to your preferences

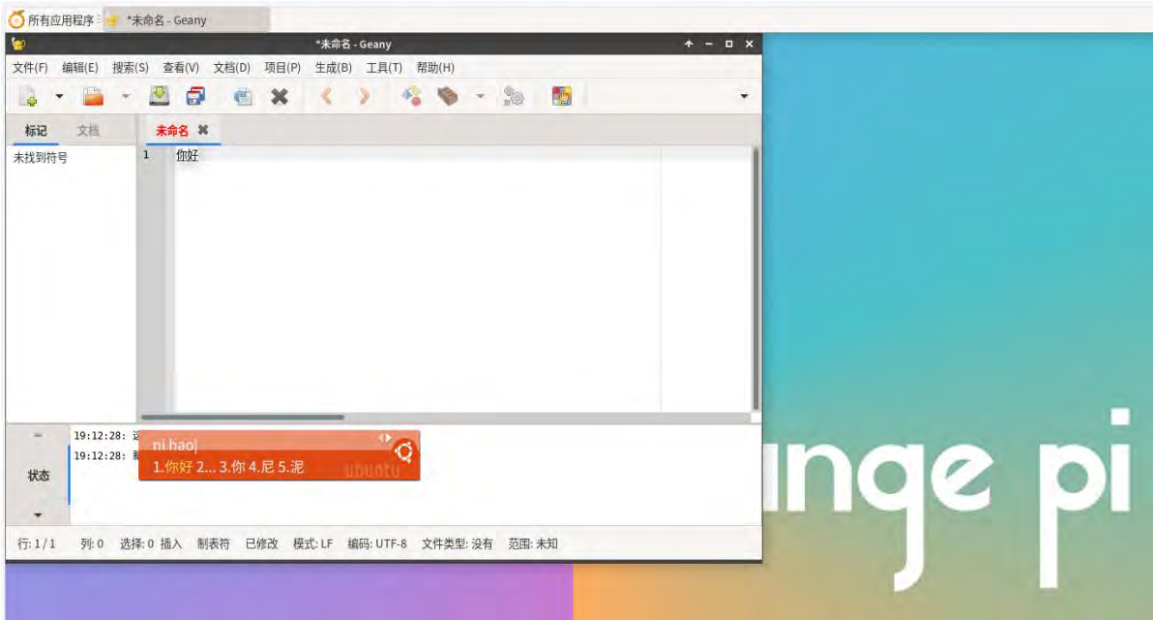
8) Then you can see that the desktop is displayed in Chinese



9) Then we can open **Geany** to test the Chinese input method, as shown in the figure below

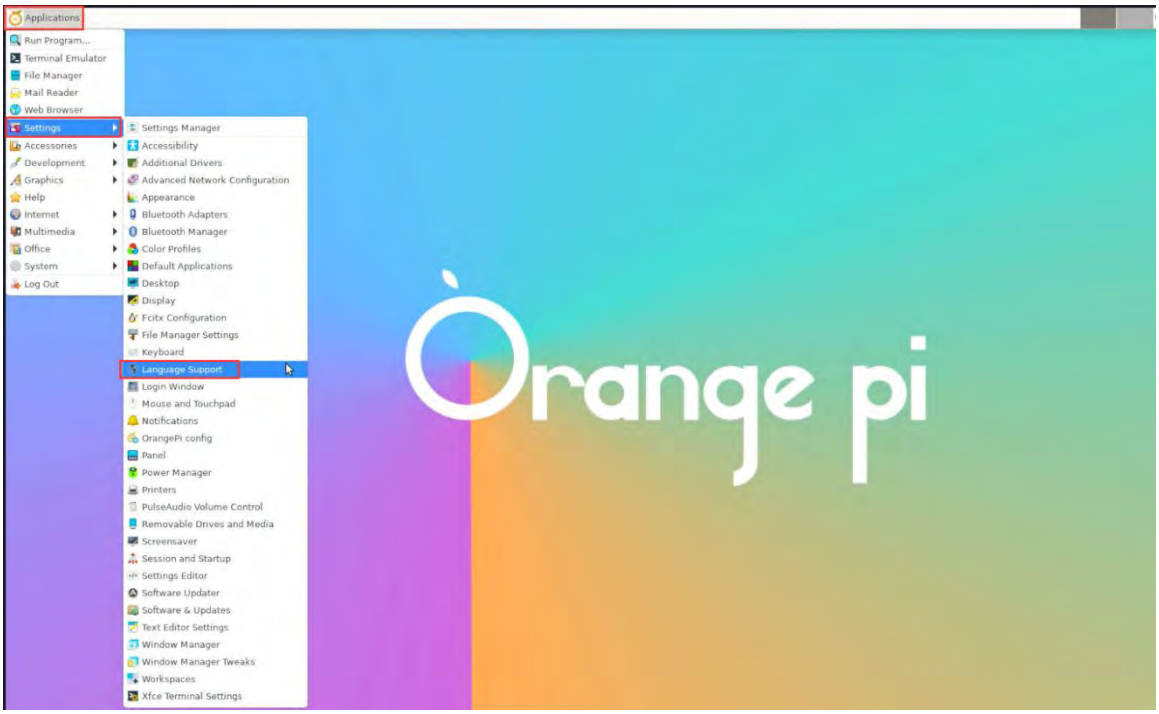


10) After opening **Geany**, the default is English input method, we can switch to Chinese input method through **Ctrl+Space** shortcut key, and then we can input Chinese



3. 29. 3. How to install Ubuntu 22.04 system

1) First open **Language Support**

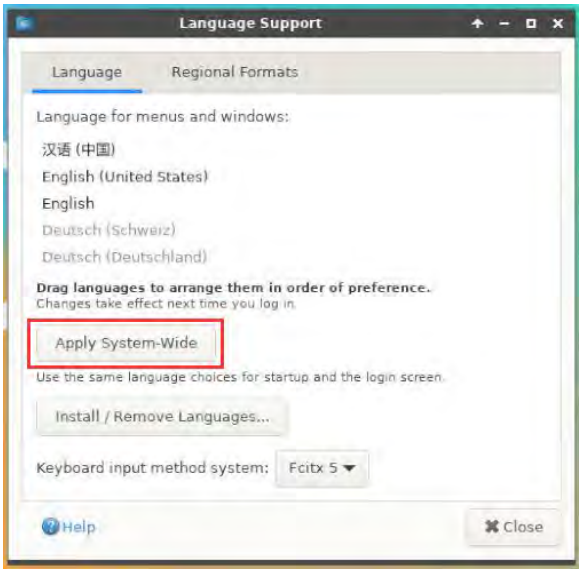


2) Then find the **Chinese (China)** option

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

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

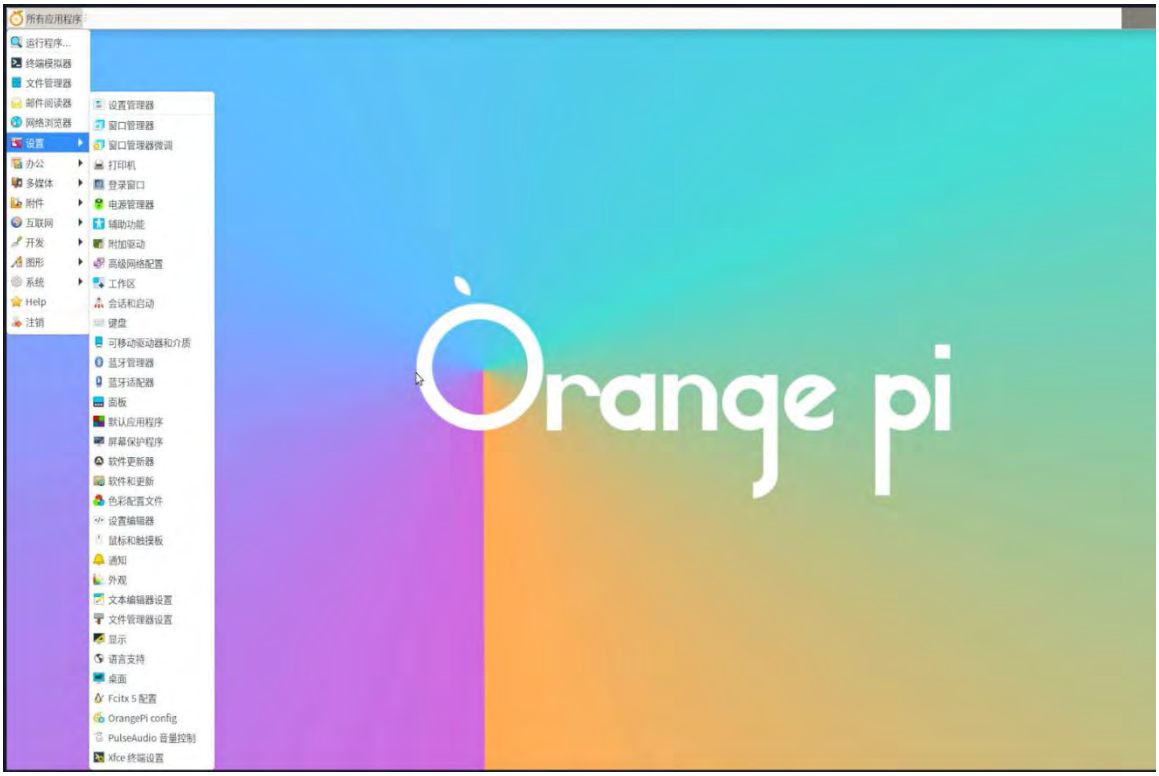
4) Then select **Apply System-Wide** to apply the Chinese settings to the entire system



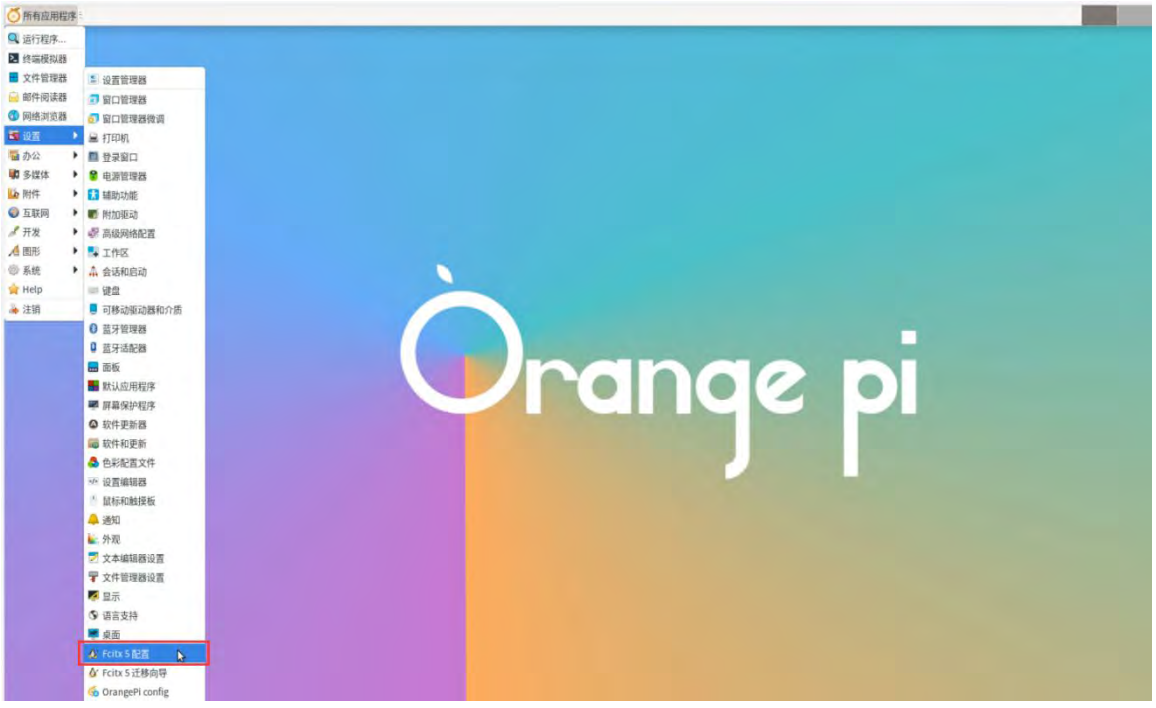
5) **Then restart the Linux system to make the configuration take effect**

6) After re-entering the system, please choose **not to ask me again** in the following interface, and then please decide whether the standard folder should also be updated to Chinese according to your preferences

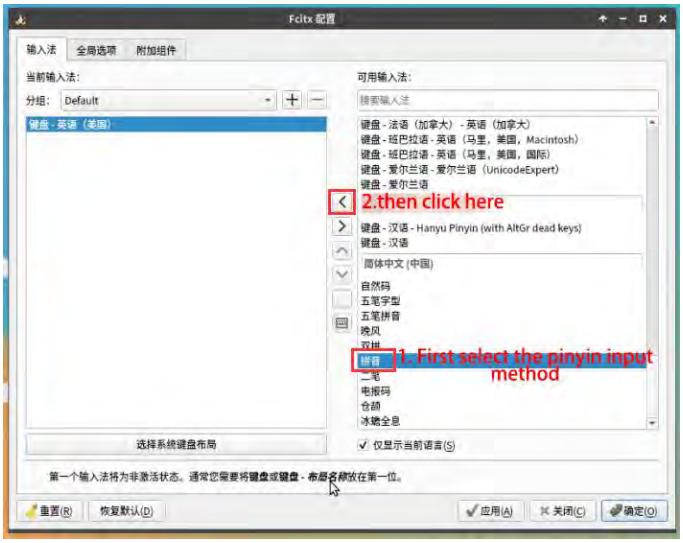
7) Then you can see that the desktop is displayed in Chinese



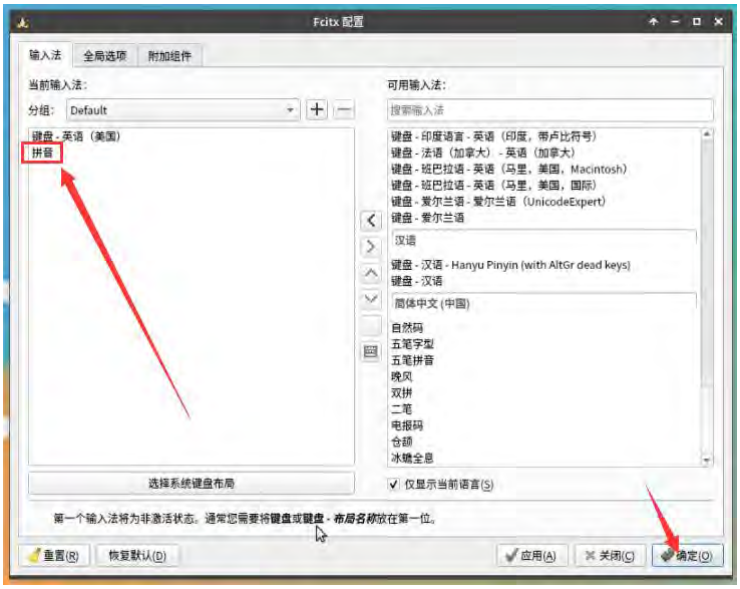
8) Then open the Fcitx5 configuration program



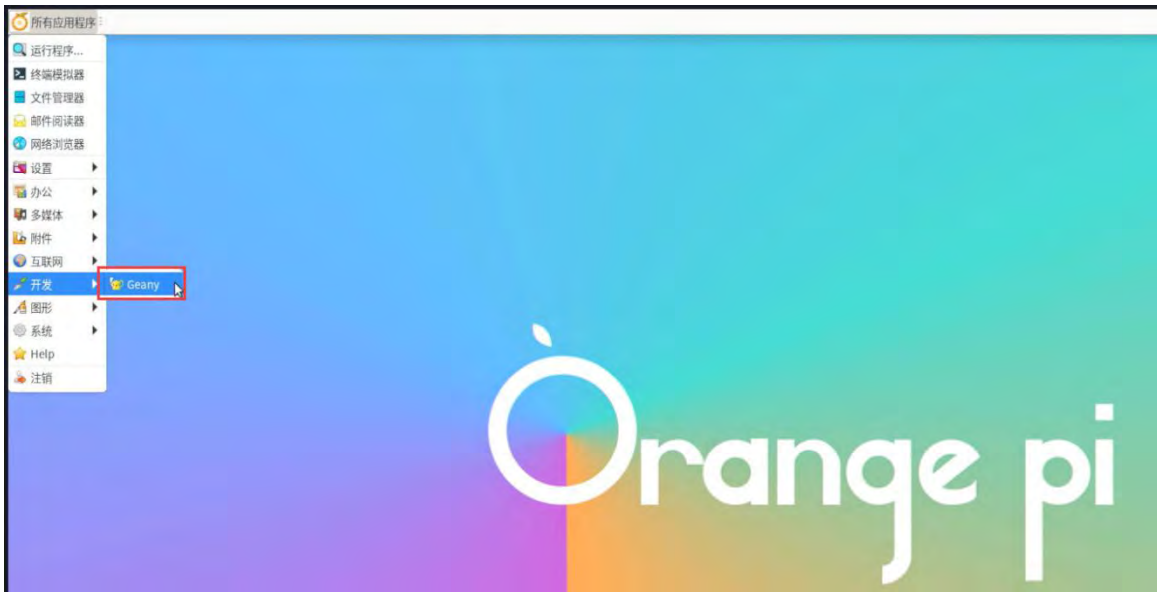
9) Then choose to use Pinyin input method



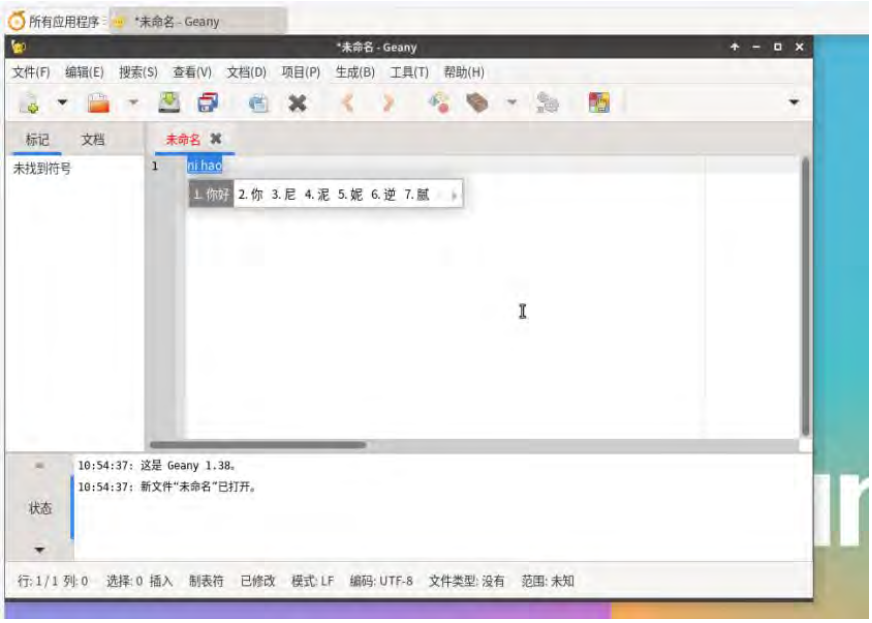
10) The interface after selection is as shown below, and then click OK



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



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



3. 30. How to Remotely Log In to the Desktop of Linux System

3. 30. 1. Remote login using NoMachine

Make sure the Ubuntu or Debian system installed on the development board is a desktop system version. In addition, NoMachine also provides detailed documents.

It is strongly recommended to read this document to be familiar with the use of NoMachine. The document link is shown below：

https://knowledgebase.nomachine.com/DT10R00166

NoMachine supports Windows, Mac, Linux, iOS and Android platforms, so we can remotely log in and control the OrangePi development board through NoMachine on a variety of devices. The following demonstrates how to remotely log in to the Linux system desktop of the OrangePi development board through NoMachine in Windows. For installation methods on other platforms, please refer to the official documentation of NoMachine

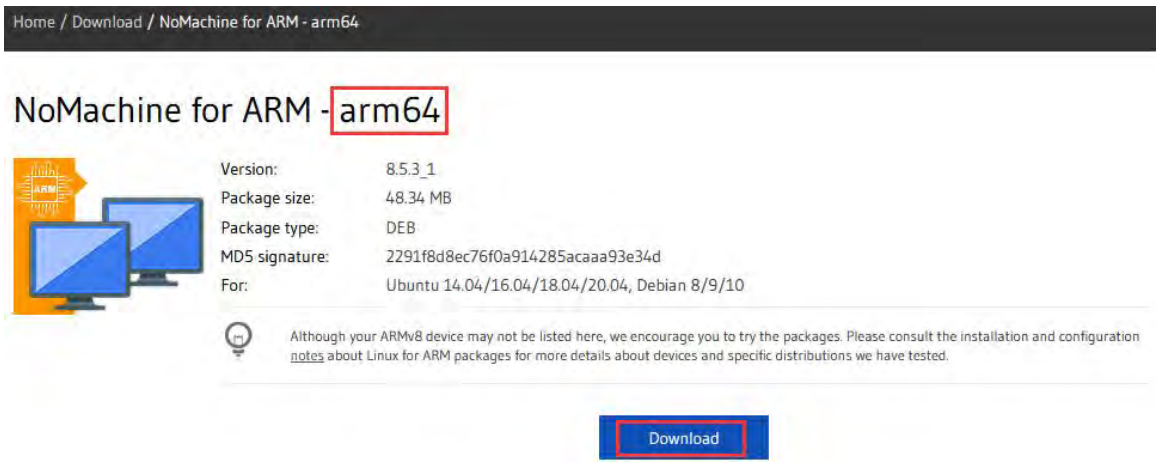
Before operation, please make sure that the Windows computer and the development board are in the same LAN, and can log in to the Ubuntu or Debian system of the development board through ssh

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

a. Since H618 is a SOC of the ARMV8 architecture, the system we use is Ubuntu or Debian, so you need to download the **NoMachine for ARM ARMv8 DEB** installation package. The download link is shown below：

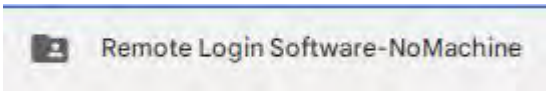
Note that this download link may change, please look for the deb package of the Armv8/Arm64 version。

https://downloads.nomachine.com/download/?id=118&distro=ARM

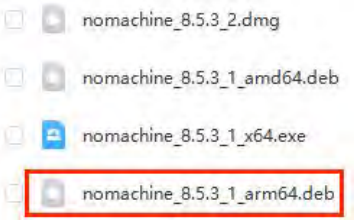


b. In addition, the installation package of **NoMachine** can also be downloaded in the **official tool**

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



Then download the arm64 version of the deb installation package



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

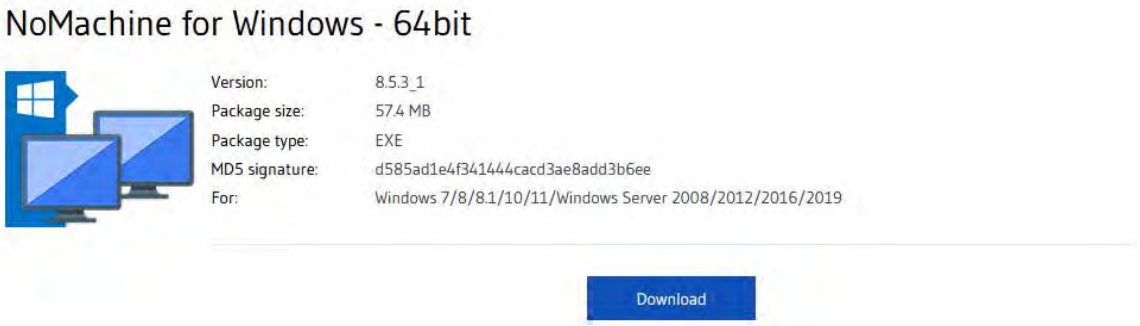
d. Then use the following command to install **NoMachine** in the Linux system of the development board

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

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

Note that this download link may change。

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

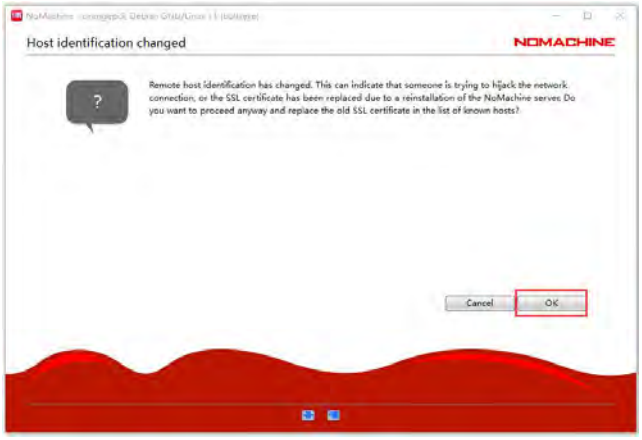


3) Then install NoMachine in Windows, **please restart the computer after installation**

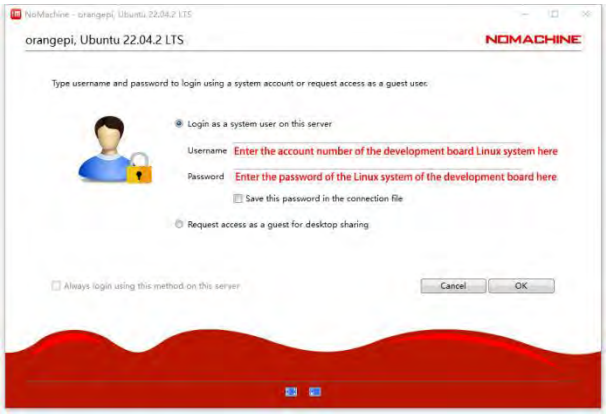
4) Then open **NoMachine** in Window

5) After Nomachine starts, it will automatically scan other devices installed in the local area network. After entering the main interface of Nomachine, you can see that the development board is already in the connected device list, and then click the location shown in the red box below. Start log in to the linux system desktop of the development board

6) Then click **OK**



7) Then enter the username and password of the linux system in the corresponding position in the figure below, and then click **OK** to start logging in



8) Then click OK in the next interface

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

3. 30. 2. Log in remotely using VNC

Before operation, please make sure that the Windows computer and the development board are in the same local area network, and can log in to the Ubuntu or Debian system of the development board normally with ssh.

Ubuntu20.04 has many problems testing VNC, please do not use this method

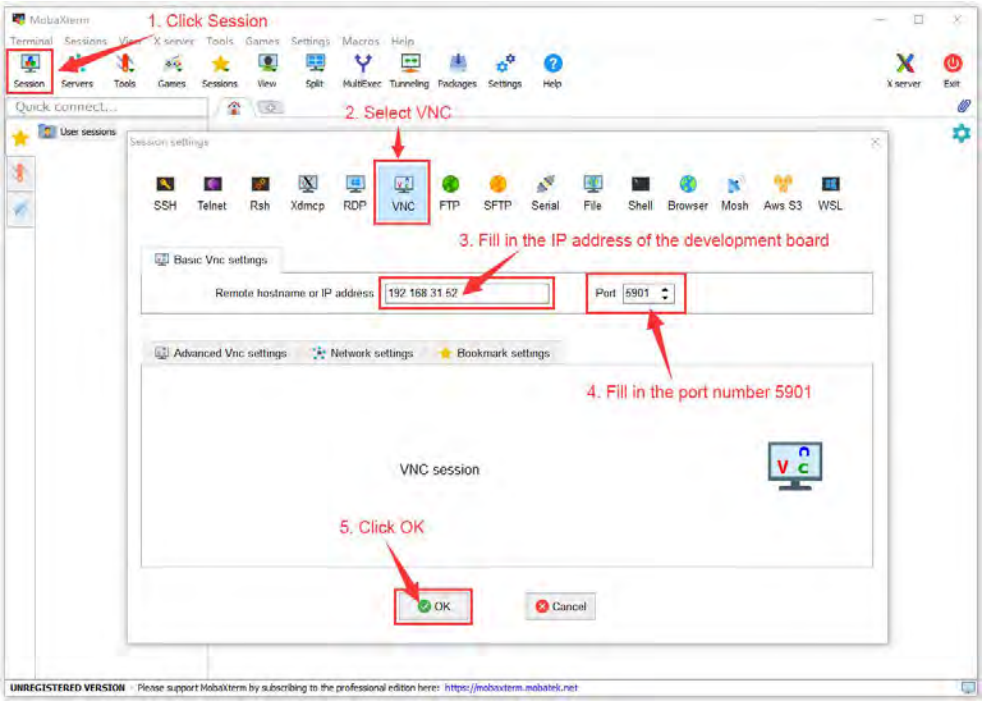
1) First run the **set_vnc.sh** script to set up vnc, remember to **add sudo permission**

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

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

New 'X' desktop is orangepi:1

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

Killing Xtightvnc process ID 3047

New 'X' desktop is orangepi:1

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

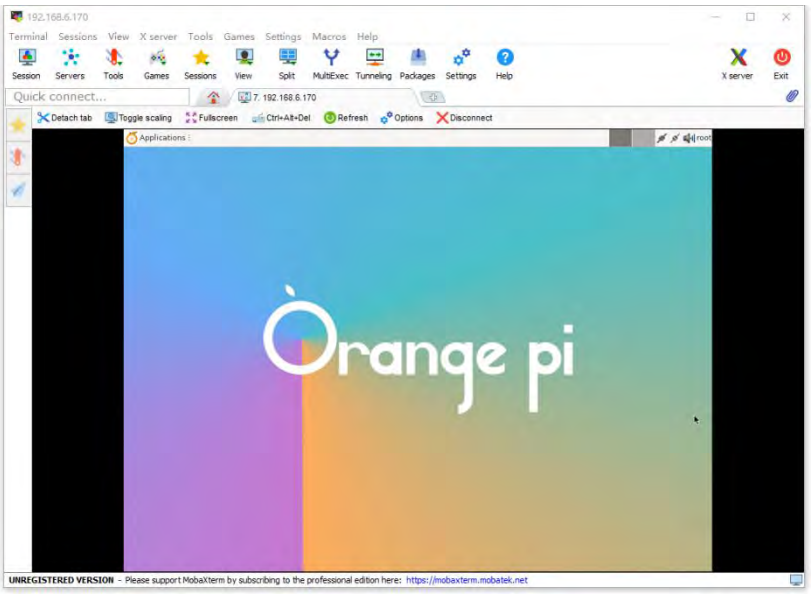
2) The steps to use the MobaXterm software to connect to the desktop of the Linux system of the development board are as follows：

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



b. Then enter the VNC password set earlier

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



3. 31. How to install QT

1) Use the script below to install QT5 and QT Creator

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

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

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

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

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

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

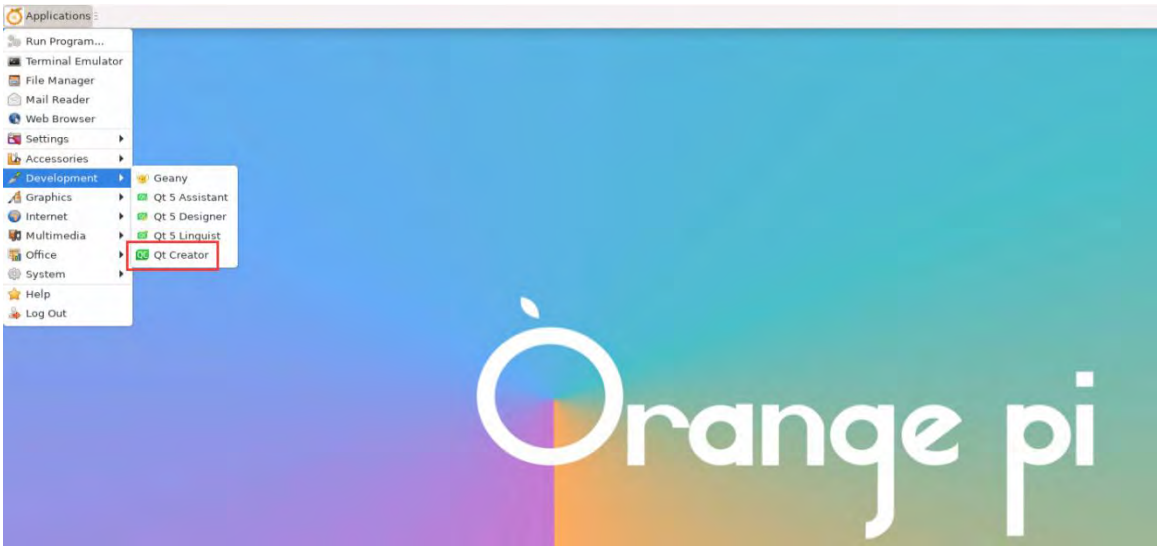
c. The QT version that comes with Debian11 is **5.15.2**

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

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

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

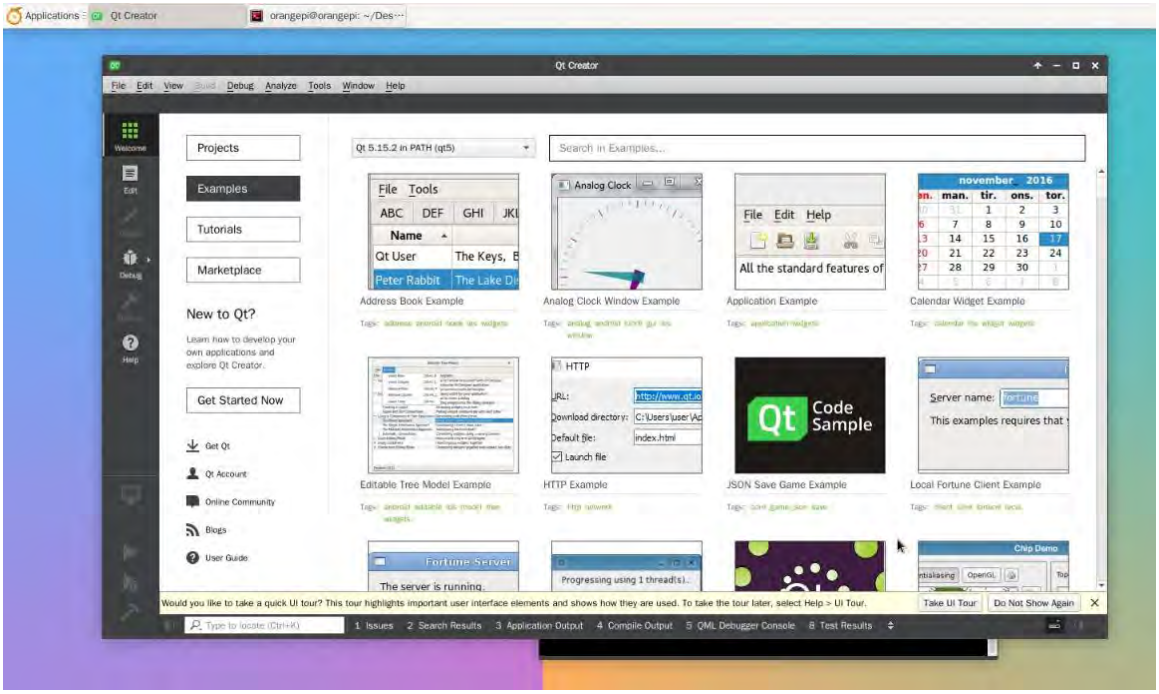
3) Then you can see the QT Creator startup icon in **Applications**



You can also use the following command to open QT Creator

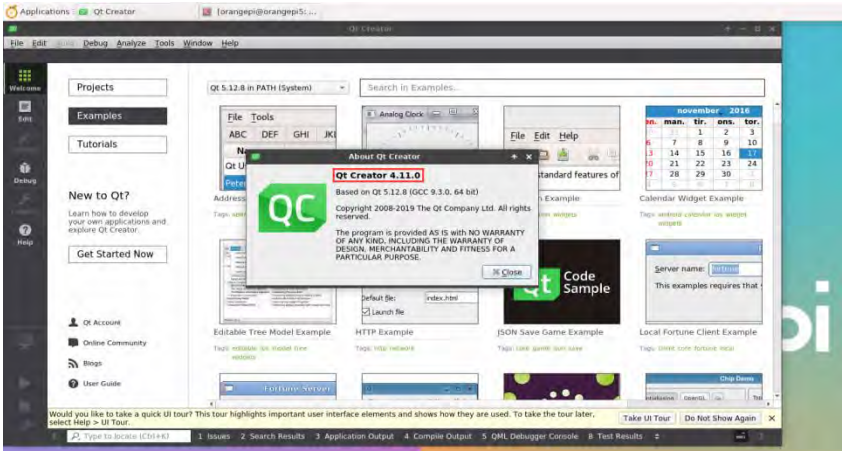
```
orangepi@orangepi:~$ qtcreator
```

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

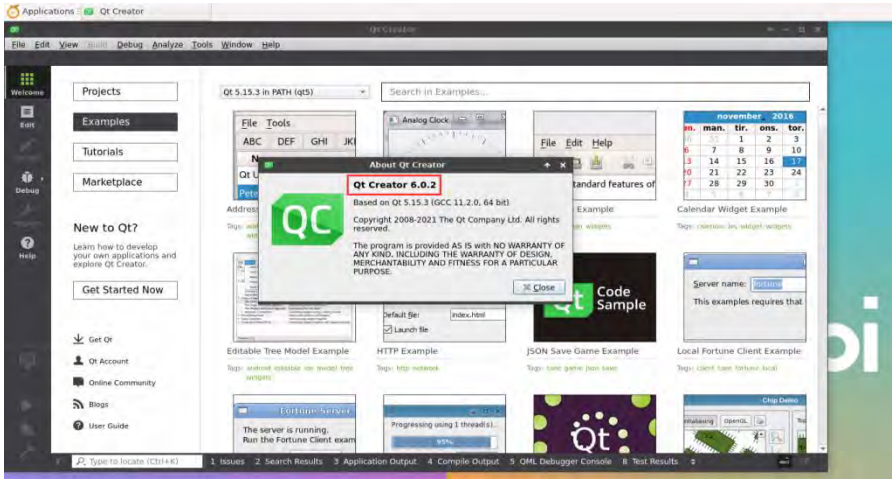


5) The version of QT Creator is as follows

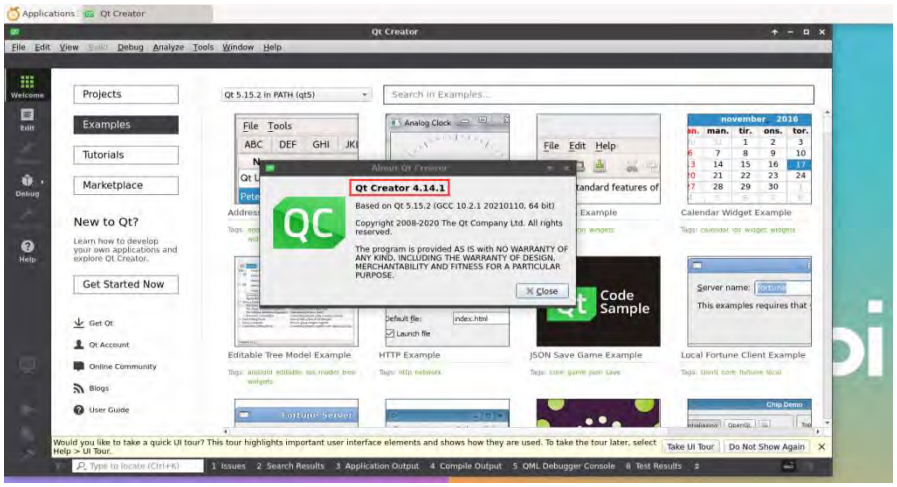
a. The default version of QT Creator in **Ubuntu20.04** is as follows



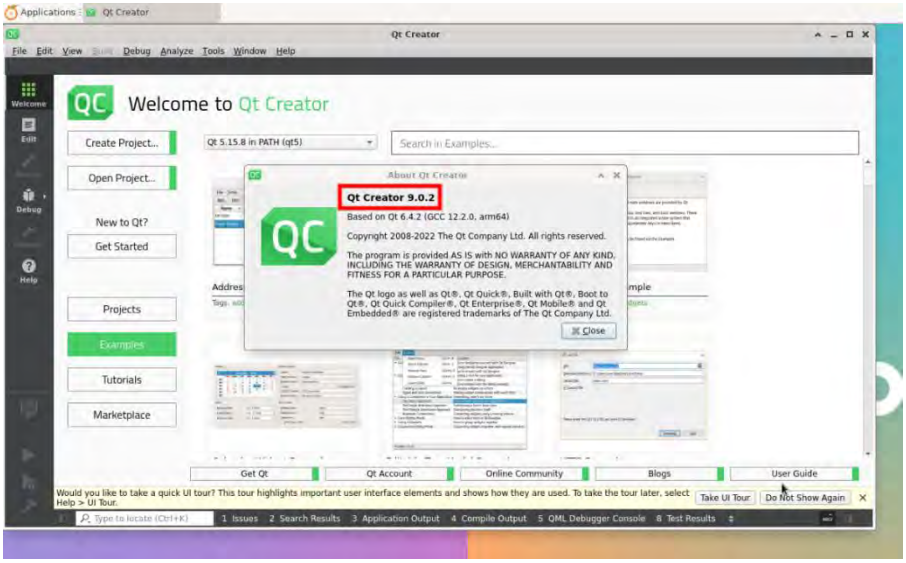
b. The default version of QT Creator in **Ubuntu22.04** is as follows



c. The default version of QT Creator in **Debian11** is as follows

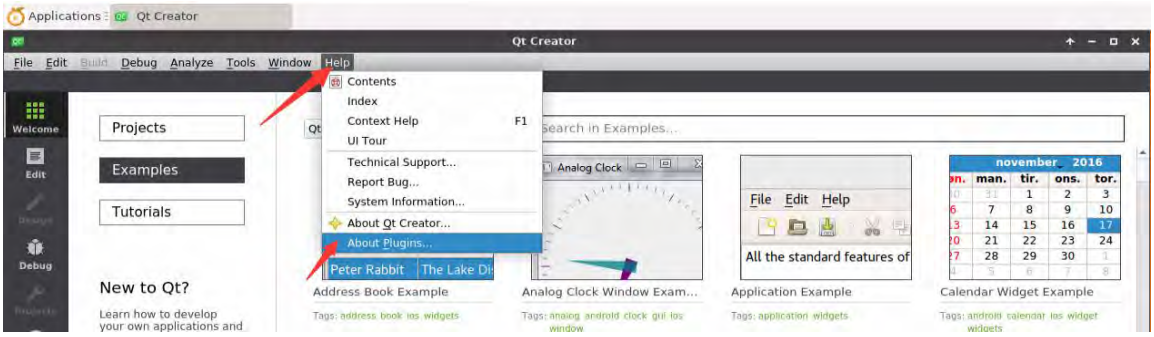


d. The default version of QT Creator in **Debian12** is as follows

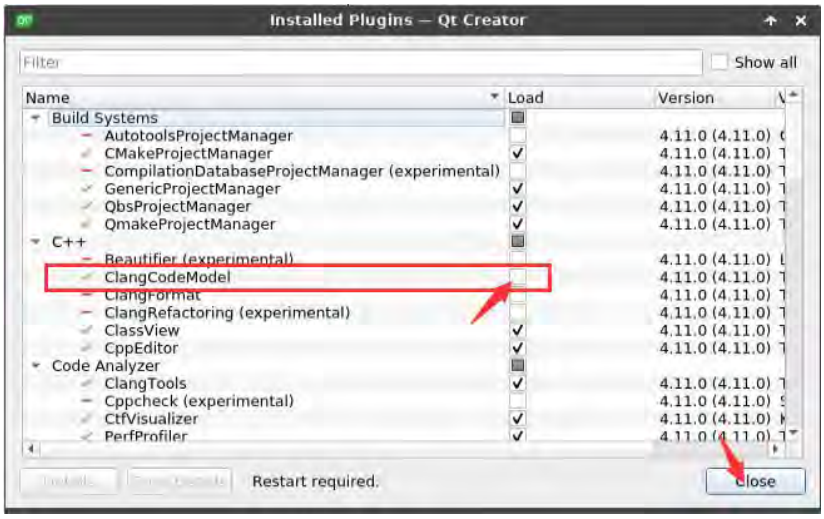


6) Then set up QT

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



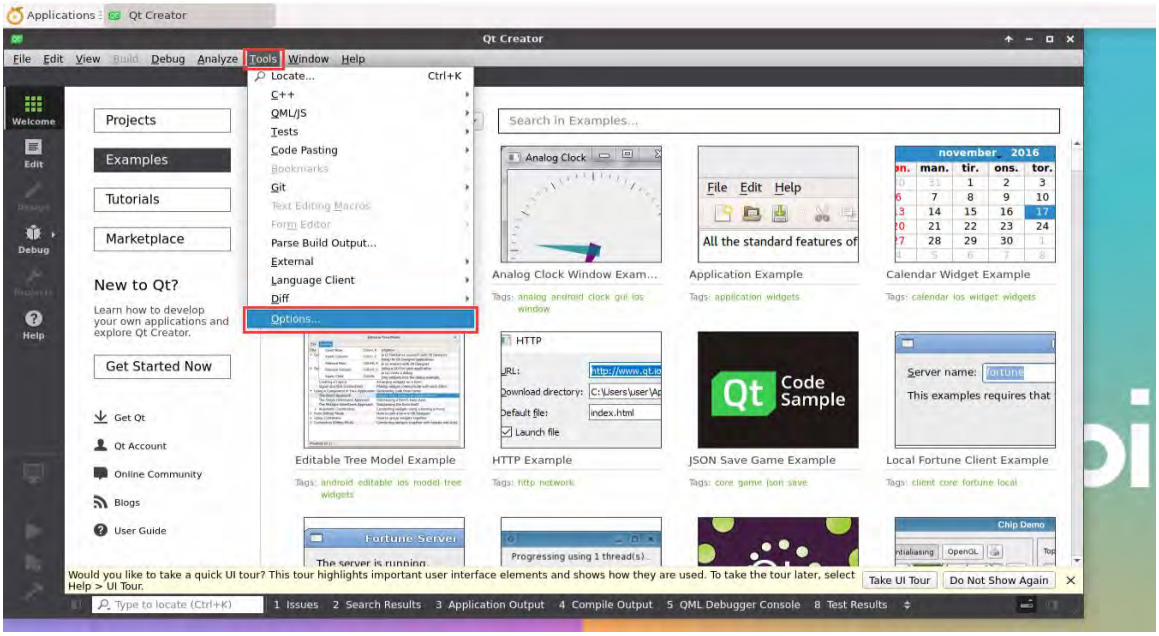
b. Then remove the tick of **ClangCodeModel**

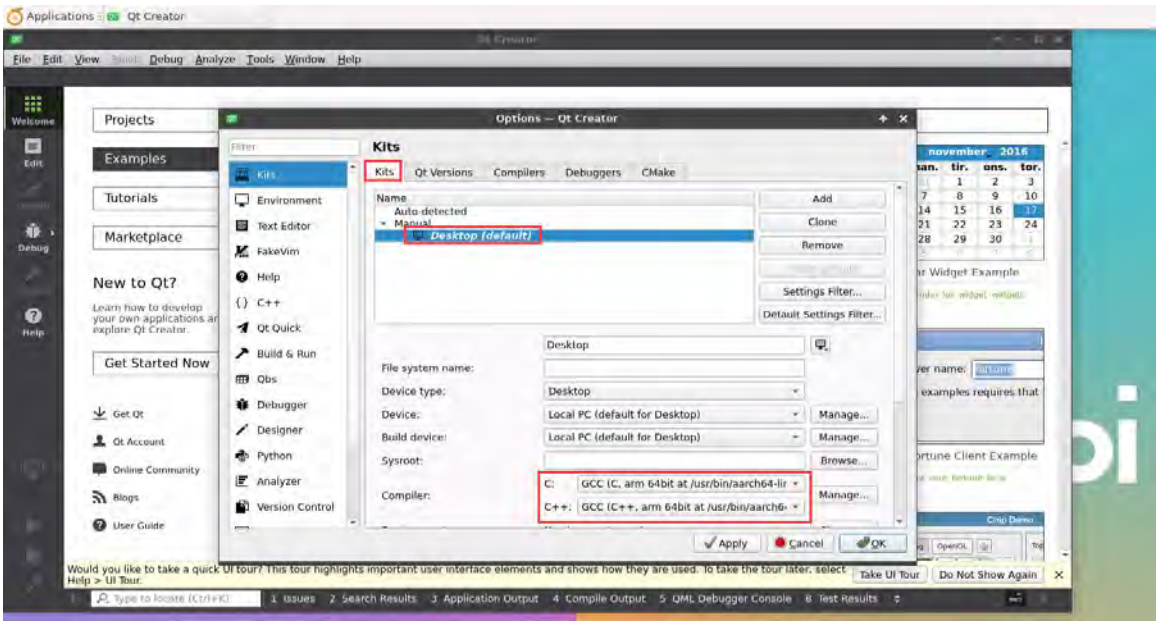


c. **After setting, you need to restart QT Creator**

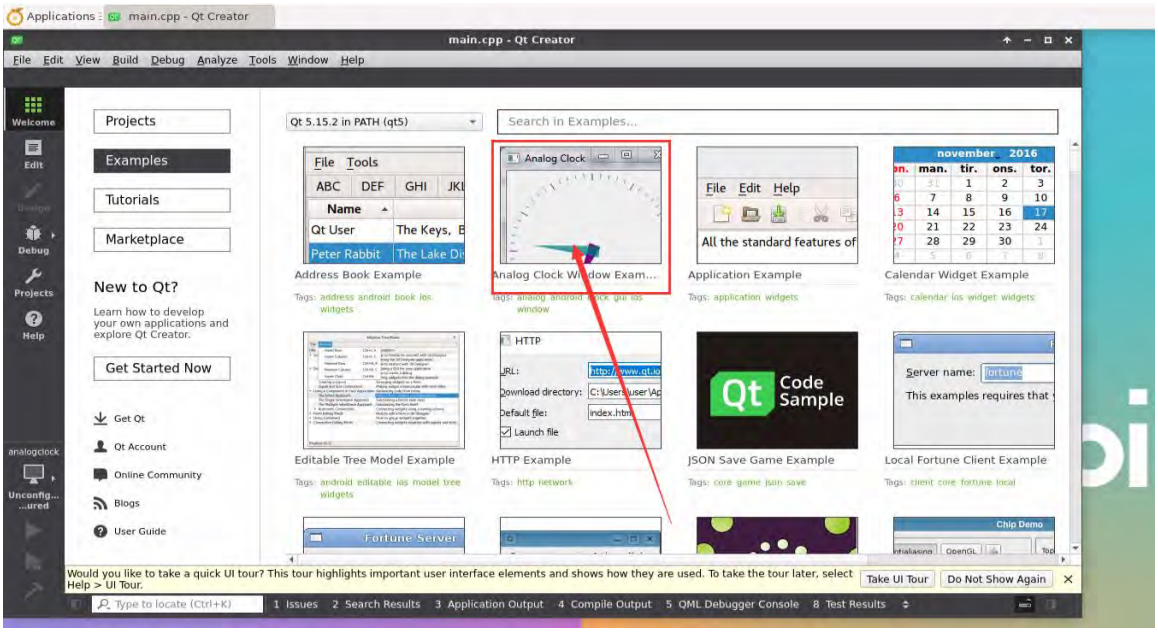
d. Then make sure the GCC compiler used by QT Creator, if the default is Clang, please modify it to GCC

Debian12 please skip this step.

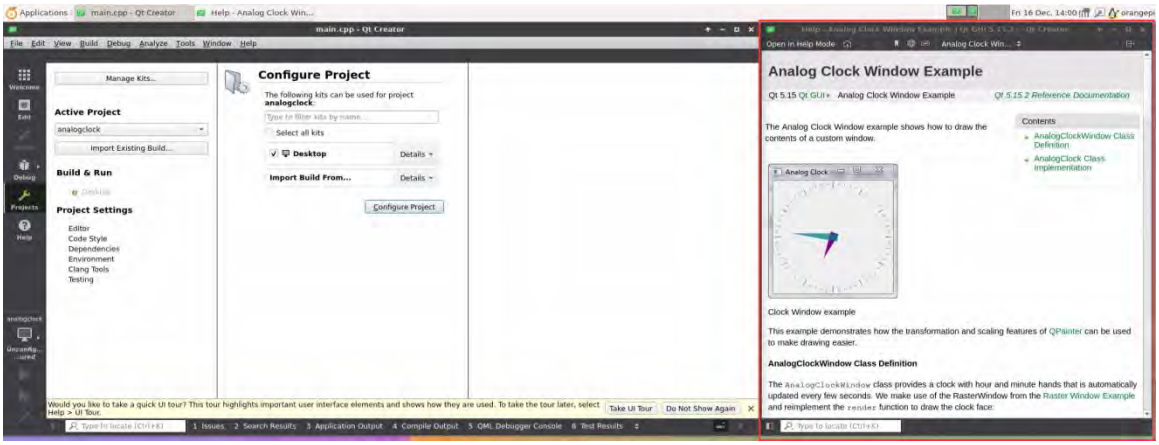




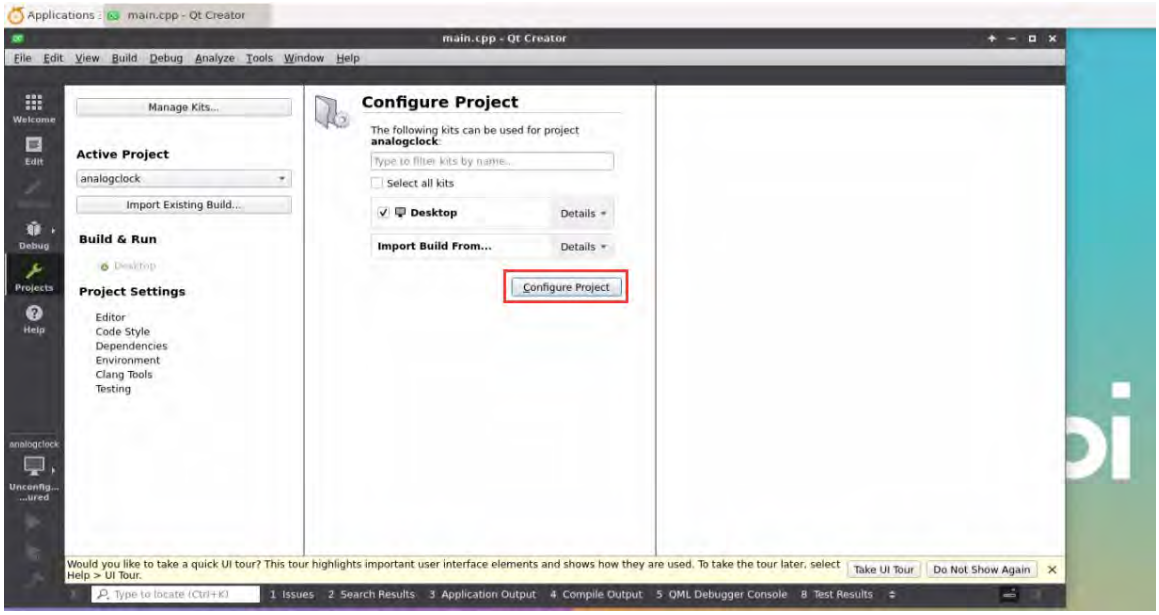
7) Then you can open a sample code



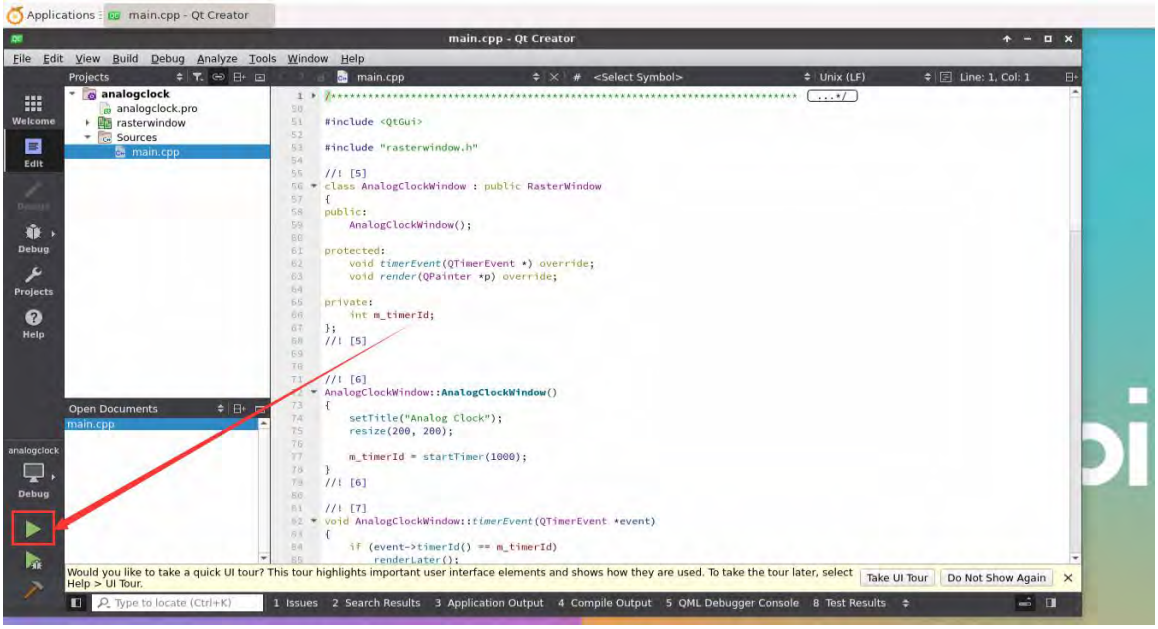
8) After clicking the sample code, the corresponding instruction document will be opened automatically, you can read the instructions carefully



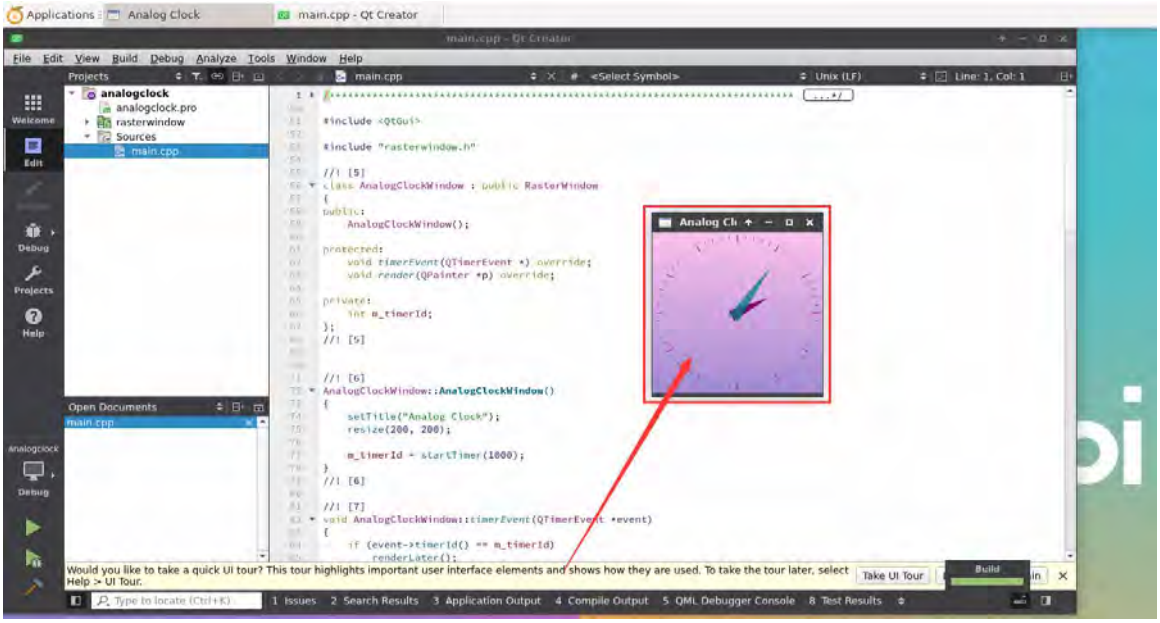
9) Then click **Configure Project**



10) Then click the green triangle in the lower left corner to compile and run the sample code



11) After waiting for a period of time, the interface shown in the figure below will pop up, which means that QT can compile and run normally



12) References

https://wiki.qt.io/Install_Qt_5_on_Ubuntu
https://download.qt.io/archive/qtcreator
https://download.qt.io/archive/qt

3. 32. How to install ROS

3. 32. 1. How to install ROS 1 Noetic on Ubuntu 20.04

1) The currently active version of ROS 1 is shown below, and the recommended version is **Noetic Ninjemys**

Active ROS 1 distributions

Recommended

Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)

http://docs.ros.org
https://wiki.ros.org/Distributions

2) ROS 1 **Noetic Ninjemys** official installation documentation link is as follows：

http://wiki.ros.org/noetic/Installation/Ubuntu

3) In the ROS **Noetic Ninjemys** official installation document, Ubuntu recommends using Ubuntu20.04, so please make sure that the system used by the development board is the **Ubuntu20.04 desktop system**

http://wiki.ros.org/noetic/Installation

4) Then use the script below to install ros1

orangepi@orangepi:~$ **install_ros.sh ros1**

5) Before using the ROS tool, you first need to initialize rosdep, and then you can quickly install some system dependencies and some core components in ROS when compiling the source code

Note that running the following command needs to ensure that the development board can access github normally, otherwise an error will be reported due to network problems

The install_ros.sh script will try to modify /etc/hosts and run the following commands automatically. However, this method cannot guarantee that github can be accessed normally every time. If install_ros.sh prompts the following error after installing ros1, please find other ways to allow the linux system of the development board to access github normally, and then manually run the following command。

https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
ERROR: error loading sources list:
The read operation timed out

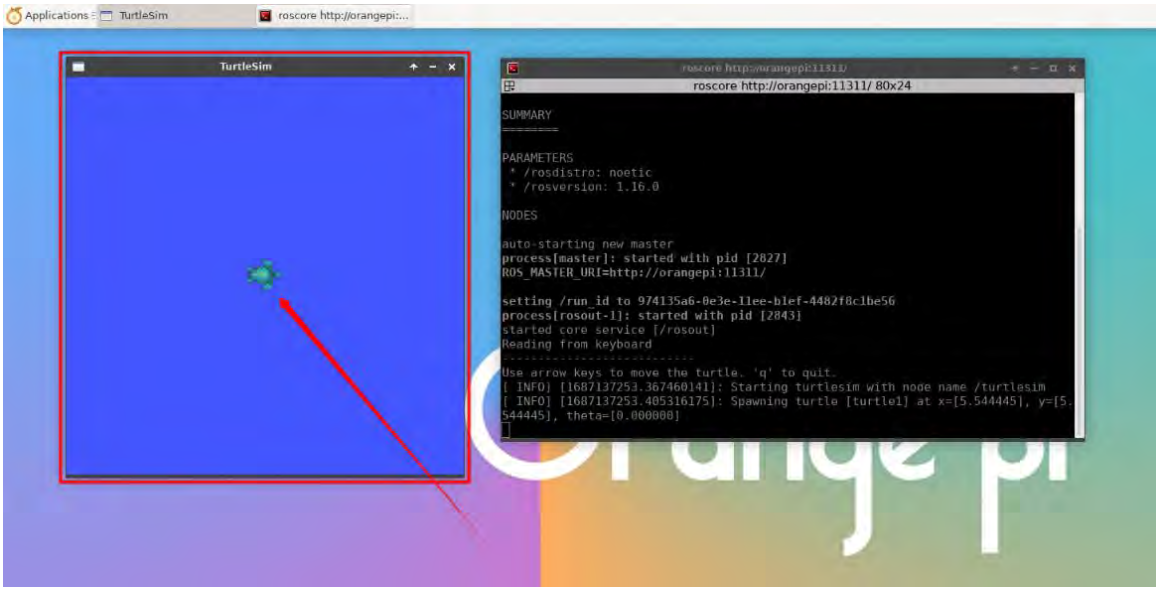
```
orangepi@orangepi:~$ source /opt/ros/noetic/setup.bash
orangepi@orangepi:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

        rosdep update
orangepi@orangepi:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index
https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
```

```
Add distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/orangepi/.ros/rosdep/sources.cache
```

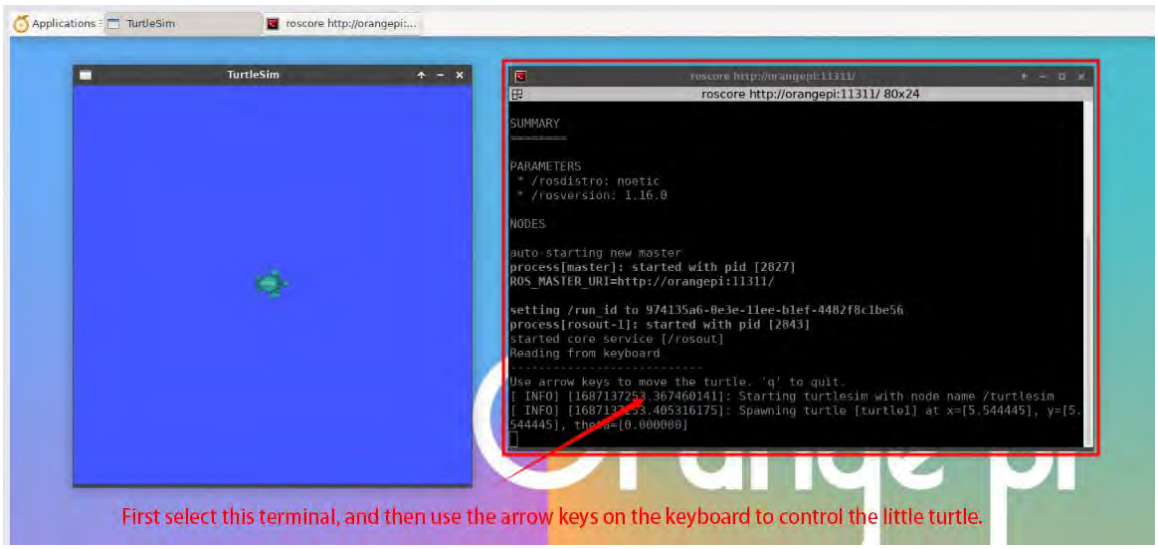
6) Then open a command line terminal window on the **desktop**, and use the **test_ros.sh** script to start a small turtle routine to test whether ROS can be used normally

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

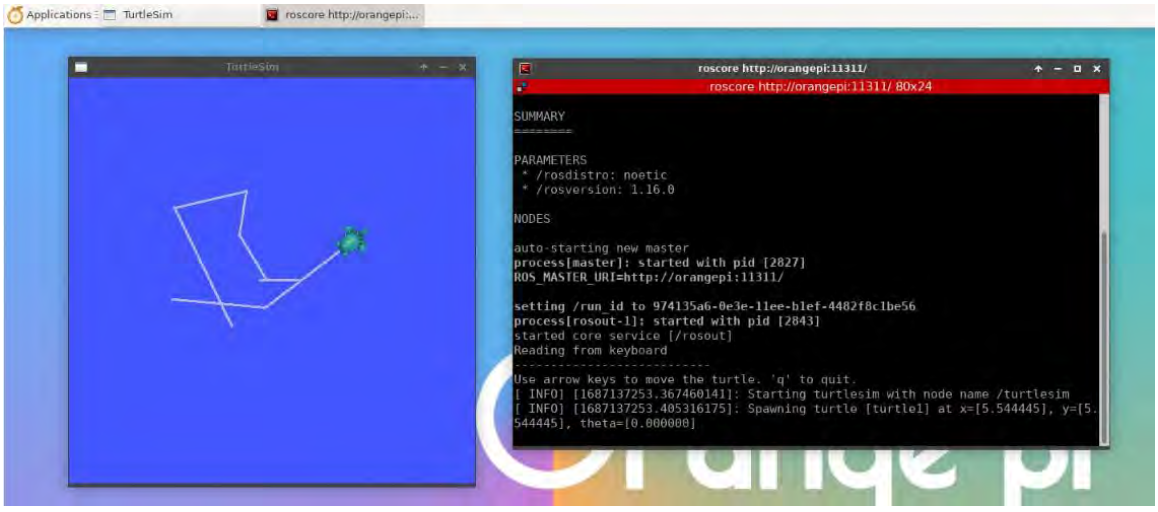
7) After running the **test_ros.sh** script, a little turtle as shown in the figure below will pop up



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



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

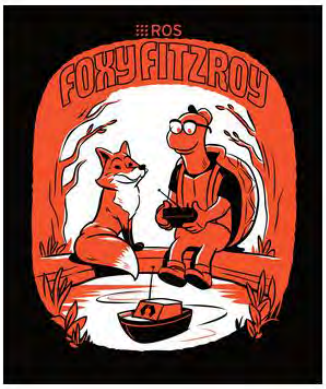


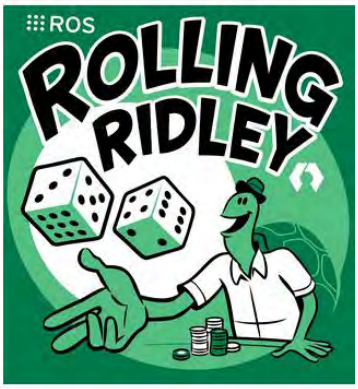
3. 32. 2. How to install ROS 2 Galactic on Ubuntu 20.04

1) The currently active version of ROS 2 is shown below, and the recommended version is **Galactic Geochelone**

Active ROS 2 distributions

Recommended **Development**





Distro	Release date	Logo	EOL date
Humble Hawksbill	May 23rd, 2022		May 2027
Galactic Geochelone	May 23rd, 2021		November 2022
Foxy Fitzroy	June 5th, 2020		May 2023

http://docs.ros.org
http://docs.ros.org/en/galactic/Releases.html

2) The link to the official ROS 2 **Galactic Geochelone** installation documentation is as follows：

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

3) In the official ROS 2 **Galactic Geochelone** installation document, Ubuntu Linux recommends using Ubuntu 20.04, so please make sure that the system used by the development board is the **Ubuntu 20.04 desktop system**. There are several ways to install ROS 2. The following demonstrates how to install ROS 2 **Galactic Geochelone**

through **Debian packages**.

4) Ros2 can be installed using the **install_ros.sh** script

```
orangepi@orangepi:~$ install_ros.sh ros2
```

5) The **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete

```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

  Call `ros2 <command> -h` for more detailed usage.
```

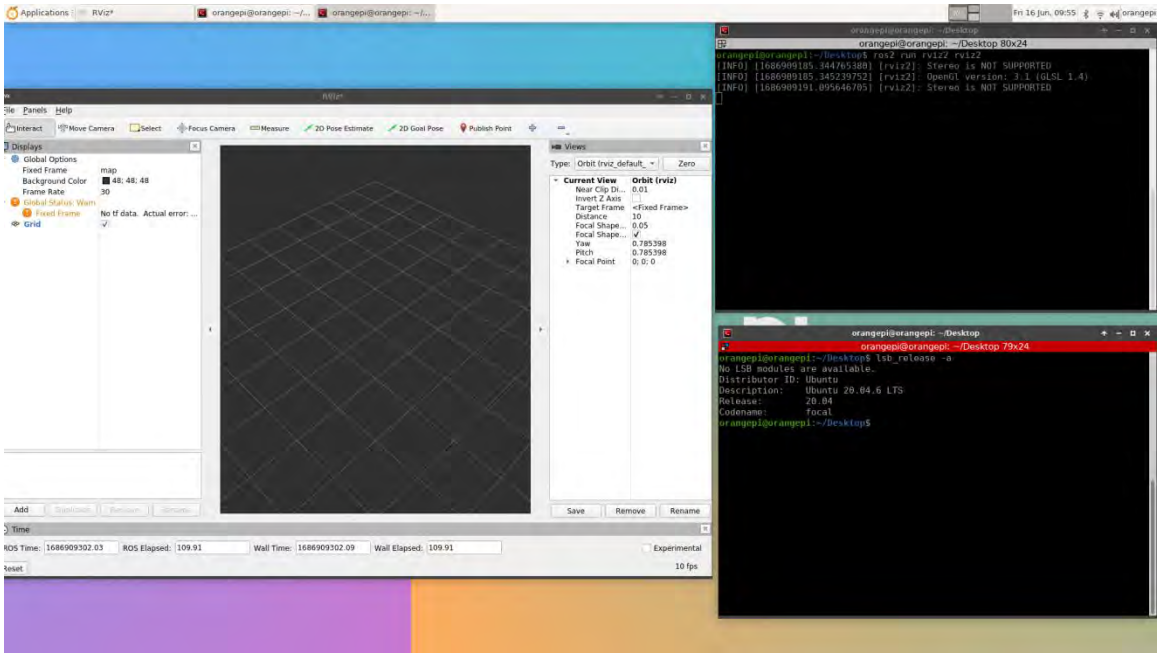
6) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully.

If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepi:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

7) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/galactic/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



8) For how to use ROS, please refer to the documentation of ROS 2

http://docs.ros.org/en/galactic/Tutorials.html

3. 32. 3. How to install ROS 2 Humble on Ubuntu 22.04

1) Ros2 can be installed using the **install_ros.sh** script

```
orangepi@orangepi:~$ install_ros.sh ros2
```

2) The **install_ros.sh** script will automatically run the **ros2 -h** command after installing ros2. If you can see the following print, it means that the ros2 installation is complete

```
usage: ros2 [-h] Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

optional arguments:
  -h, --help            show this help message and exit

Commands:
  action      Various action related sub-commands
  bag         Various rosbag related sub-commands
  component   Various component related sub-commands
  daemon      Various daemon related sub-commands
  doctor      Check ROS setup and other potential issues
  interface   Show information about ROS interfaces
  launch      Run a launch file
  lifecycle   Various lifecycle related sub-commands
  multicast   Various multicast related sub-commands
  node        Various node related sub-commands
  param       Various param related sub-commands
  pkg         Various package related sub-commands
  run         Run a package specific executable
  security    Various security related sub-commands
  service     Various service related sub-commands
  topic       Various topic related sub-commands
  wtf         Use `wtf` as alias to `doctor`

  Call `ros2 <command> -h` for more detailed usage.
```

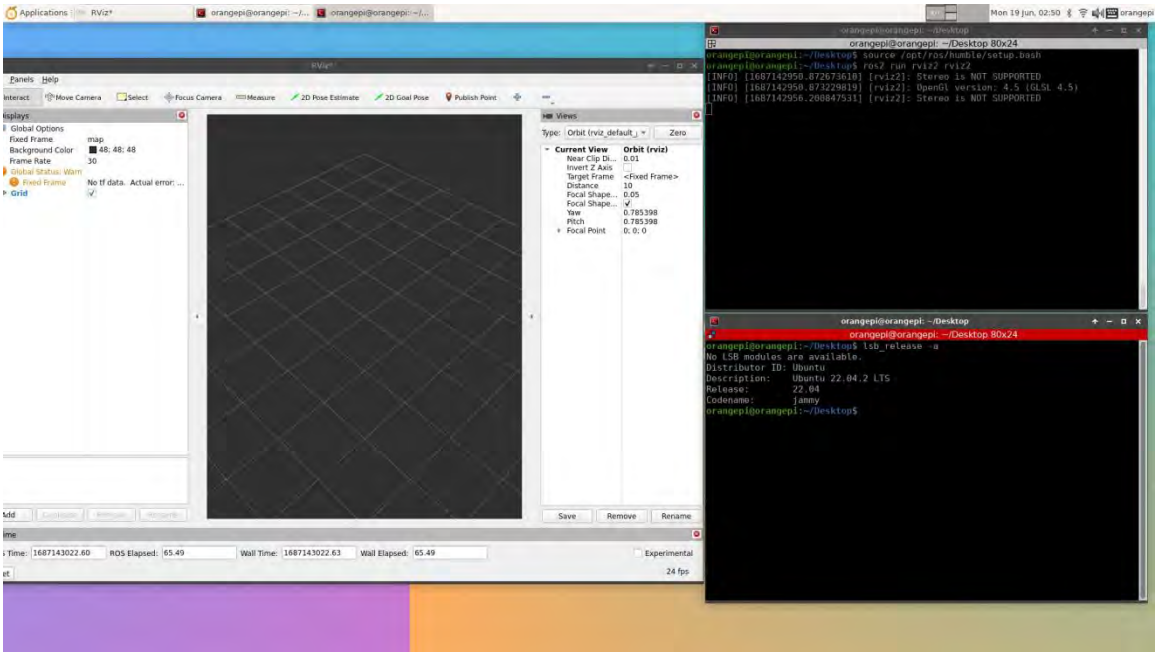
3) Then you can use the **test_ros.sh** script to test whether ROS 2 is installed successfully. If you can see the following print, it means that ROS 2 can run normally

```
orangepi@orangepi:~$ test_ros.sh
[INFO] [1671174101.200091527] [talker]: Publishing: 'Hello World: 1'
[INFO] [1671174101.235661048] [listener]: I heard: [Hello World: 1]
[INFO] [1671174102.199572327] [talker]: Publishing: 'Hello World: 2'
[INFO] [1671174102.204196299] [listener]: I heard: [Hello World: 2]
[INFO] [1671174103.199580322] [talker]: Publishing: 'Hello World: 3'
```

```
[INFO] [1671174103.204019965] [listener]: I heard: [Hello World: 3]
```

4) Run the following command to open rviz2

```
orangepi@orangepi:~$ source /opt/ros/humble/setup.bash
orangepi@orangepi:~$ ros2 run rviz2 rviz2
```



5) Reference documents

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

3. 33. How to install the kernel header file

The Debian 11 system with the Linux6.1 kernel will report a GCC error when compiling the kernel module. So if you want to compile the kernel module, please use Debian12 or Ubuntu22.04

1) The Linux image released by OPi comes with the deb package of the kernel header file by default, and the storage location is **/opt/**

```
orangepi@orangepi:~$ ls /opt/linux-headers*
/opt/linux-headers-xxx-sun50iw9_x.x.x_arm64.deb
```

2) Use the following command to install the deb package of the kernel header file

```
orangepi@orangepi:~$ sudo dpkg -i /opt/linux-headers*.deb
```

3) After installation, you can see the folder where the kernel header files are located under **/usr/src**

```
orangepi@orangepi:~$ ls /usr/src
linux-headers-x.x.x
```

4) Then you can compile the source code of the hello kernel module that comes with the Linux image. The source code of the hello module is in **/usr/src/hello**. After entering this directory, use the make command to compile。

```
orangepi@orangepi:~$ cd /usr/src/hello/
orangepi@orangepi:/usr/src/hello$ sudo make
make -C /lib/modules/5.4.125/build M=/usr/src/hello modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.125'
  CC [M]  /usr/src/hello/hello.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /usr/src/hello/hello.mod.o
  LD [M]  /usr/src/hello/hello.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.125'
```

5) After compiling, the **hello.ko** kernel module will be generated

```
orangepi@orangepi:/usr/src/hello$ ls *.ko
hello.ko
```

6) Using the **insmod** command, you can insert the **hello.ko** kernel module into the kernel

```
orangepi@orangepi:/usr/src/hello$ sudo insmod hello.ko
```

7) Then use the **demsg** command to view the output of the **hello.ko** kernel module. If you can see the output instructions below, the **hello.ko** kernel module is loaded correctly

```
orangepi@orangepi:/usr/src/hello$ dmesg | grep "Hello"
[ 2871.893988] Hello OrangePi -- init
```

8) Use the **rmmod** command to uninstall the **hello.ko** kernel module

```
orangepi@orangepi:/usr/src/hello$ sudo rmmod hello
orangepi@orangepi:/usr/src/hello$ dmesg | grep "Hello"
[ 2871.893988] Hello OrangePi -- init
[ 3173.800892] Hello OrangePi -- exit
```

3. 34. Some programming language tests supported by Linux system

3. 34. 1. Debian Bullseye System

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

a. The version of gcc is as follows

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

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

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

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

        return 0;
}
```

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

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

2) Debian Bullseye comes with Python3 installed by default

a. The specific version of Python is as follows

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

Use the Ctrl+D shortcut to exit python's interactive mode。

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

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

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

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

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

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

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

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

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

c. Edit **hello_world.java of java version**

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

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

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

3.34.2. Debian Bookworm System

1) Debian Bookworm is installed with a gcc compilation tool chain by default, which can

directly compile C language programs in the Linux system of the development board

a. The version of gcc is as follows

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

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

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

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

        return 0;
}
```

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

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

2) Debian Bookworm has Python3 installed by default

a. The specific version of Python is as follows

```
orangepi@orangepi:~$ python3
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Use the Ctrl+D shortcut to exit python's interactive mode。

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

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

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

```
orangepi@orangepi:~$ python3 hello_world.py
```

```
Hello World!
```

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

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

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

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

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

c. Edit the Javaverison's **hello_world.java**

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

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

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

3.34.3. Ubuntu Focal System

1) Ubuntu Focal has a gcc compilation tool chain installed by default, which can directly compile C language programs in the Linux system of the development board

a. The version of a.gcc is as follows

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

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

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

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

        return 0;
}
```

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

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

2) Ubuntu Focal has Python3 installed by default

a. The specific version of Python3 is as follows

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

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

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

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

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

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

3) Ubuntu Focal does not install Java compilation tools and operating environment by default

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

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

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

```
orangepi@orangepi:~$ java --version
openjdk 17.0.2 2022-01-18
OpenJDK Runtime Environment (build 17.0.2+8-Ubuntu-120.04)
```

```
OpenJDK 64-Bit Server VM (build 17.0.2+8-Ubuntu-120.04, mixed mode, sharing)
```

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

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

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

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

3. 34. 4. Ubuntu Jammy System

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

a. The version of a.gcc is as follows

```
orangepi@orangepi:~$ gcc --version
gcc (Ubuntu 11.3.0-1ubuntu1~22.04.1) 11.3.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

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

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

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

        return 0;
}
```

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

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

2) Ubuntu Jammy has Python3 installed by default

a. The specific version of Python3 is as follows

```
orangepi@orangepi:~$ python3
Python 3.10.6 (main, May 29 2023, 11:10:38) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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

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

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

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

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

3) Ubuntu Jammy does not install Java compilation tools and operating environment by default

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

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

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

```
orangepi@orangepi:~$ java --version
openjdk 18.0.2-ea 2022-07-19
OpenJDK Runtime Environment (build 18.0.2-ea+9-Ubuntu-222.04)
OpenJDK 64-Bit Server VM (build 18.0.2-ea+9-Ubuntu-222.04, mixed mode, sharing)
```

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

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

```
}
```

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

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

3. 35. The method of uploading files to the Linux system of the development board

3. 35. 1. How to upload files to the development board Linux system in Ubuntu PC

3. 35. 1. 1. How to upload files using the scp command

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

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

b. **orangepi：** It is the user name of the Linux system of the development board, and it can also be replaced with other ones, such as root

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

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

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

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

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

3) There are more usages of scp, please use the following command to view the man manual

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

3. 35. 1. 2. How to upload files using filezilla

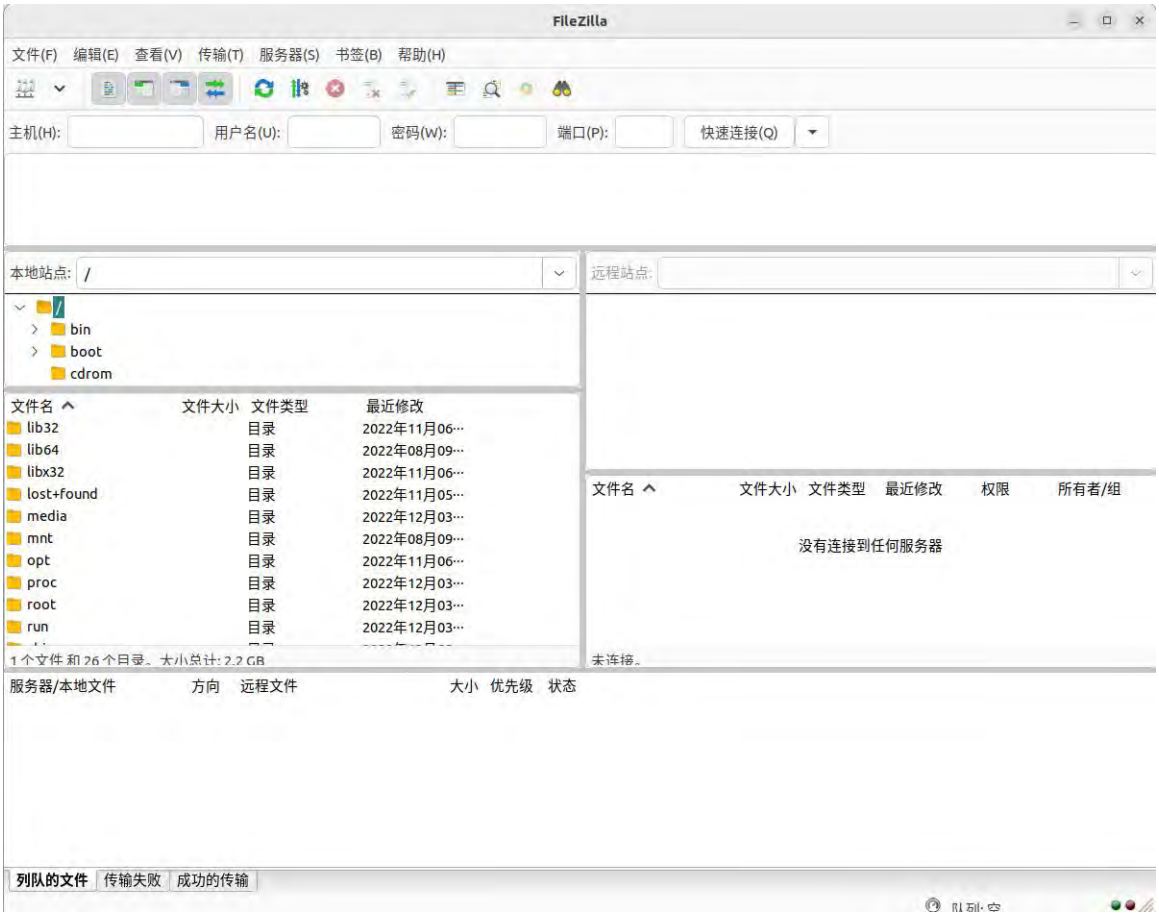
1) First install filezilla in Ubuntu PC

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

2) Then use the following command to open filezilla

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

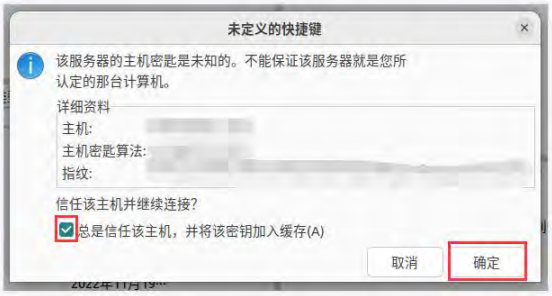
3) The interface after filezilla is opened is as follows, at this time, the display under the remote site on the right is empty



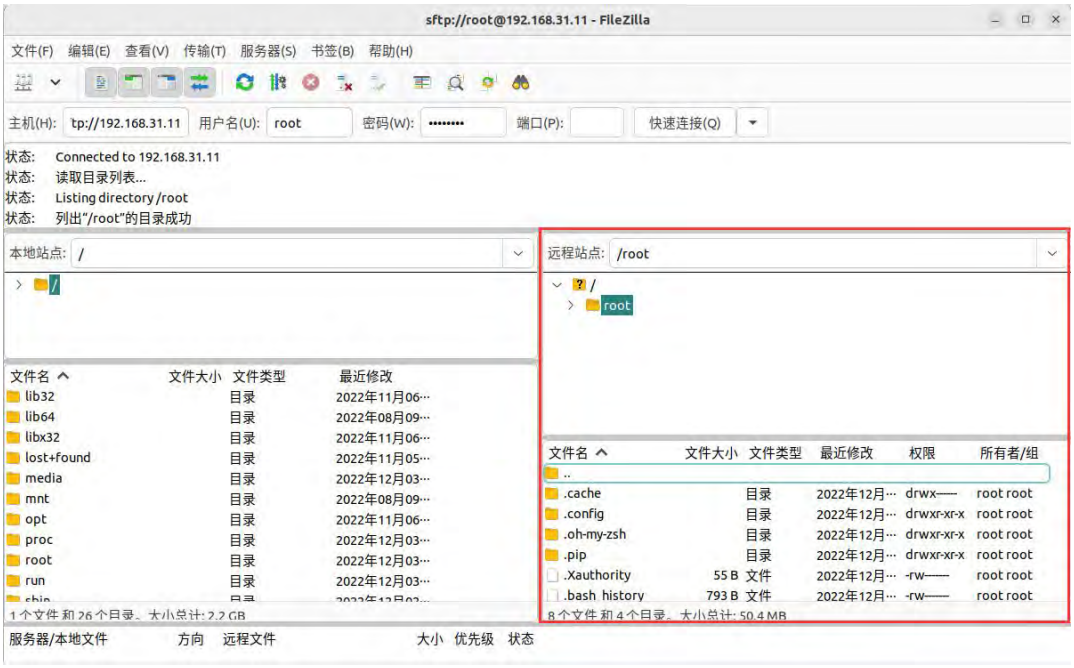
4) The method of connecting the development board is shown in the figure below

5) Then choose to **save the password**, and then click **OK**

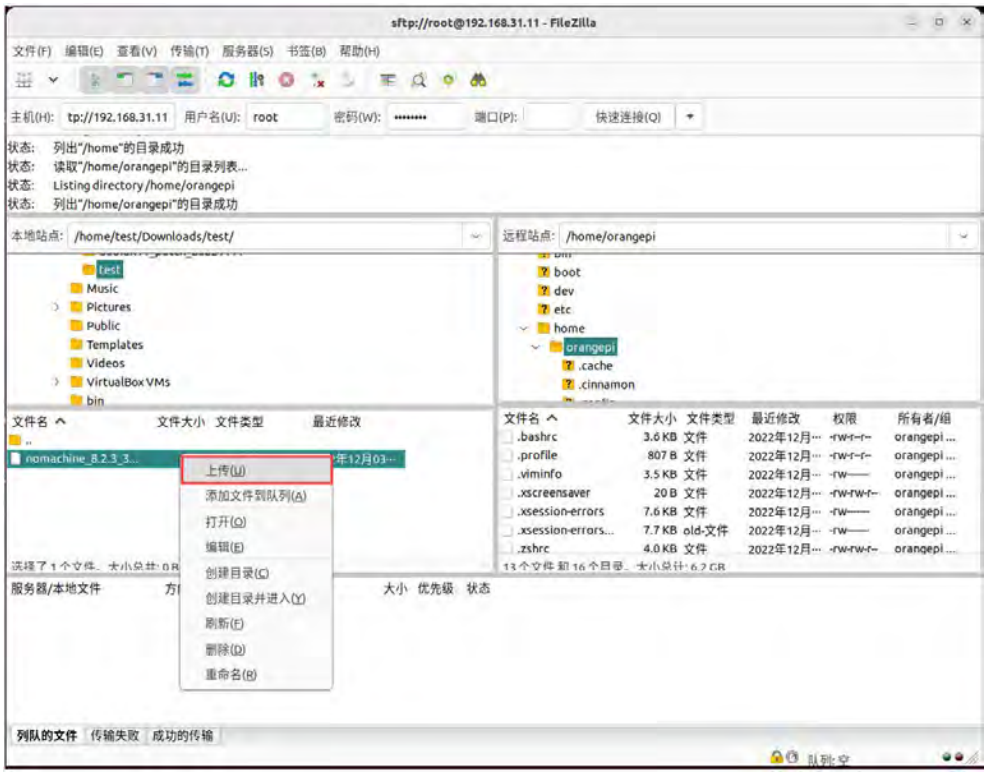
6) Then choose to **always trust this host**, and then click **OK**



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



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



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

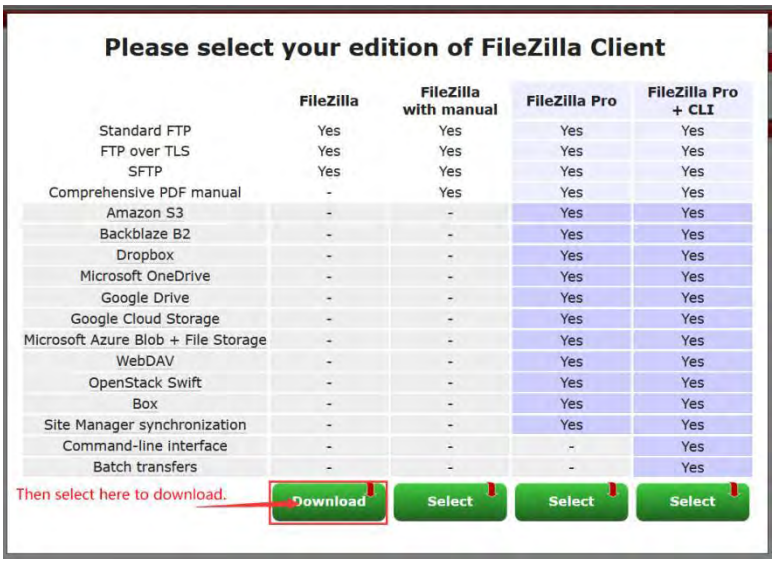
10) The method of uploading a folder is the same as that of uploading a file, so I won't go into details here

3.35.2. How to upload files to the development board Linux system in Windows PC

3.35.2.1. How to upload files using filezilla

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

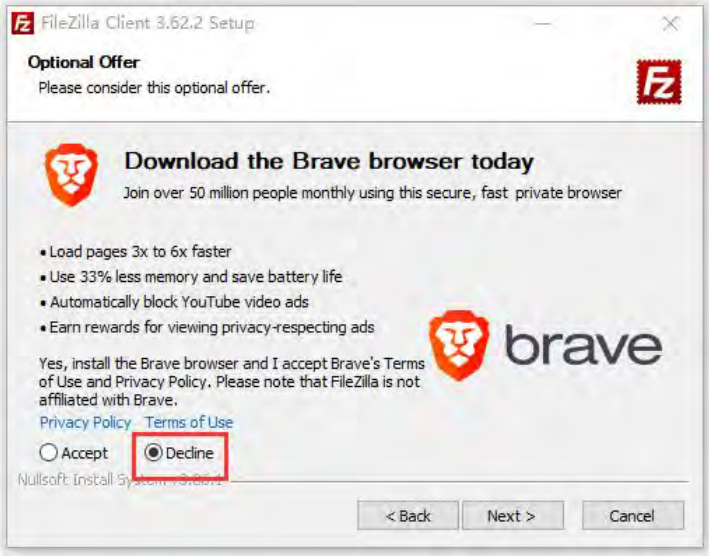
https://filezilla-project.org/download.php?type=client



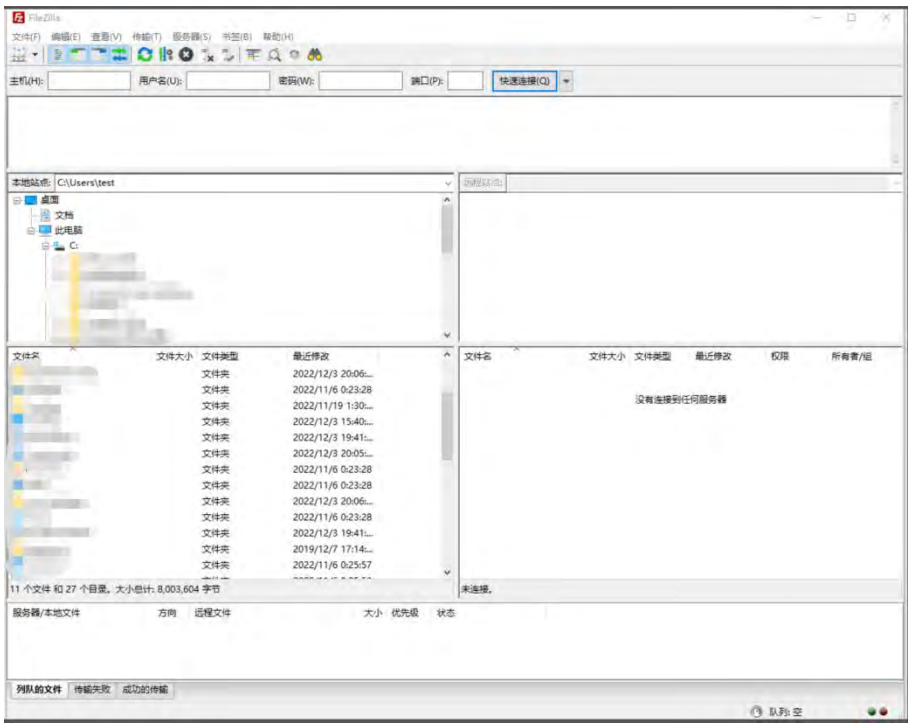
2) The downloaded installation package is as follows, and then double-click to install directly

FileZilla_Server_1.5.1_win64-setup.exe

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



3) The interface after filezilla is opened is as follows, and the display under the remote site on the right is empty

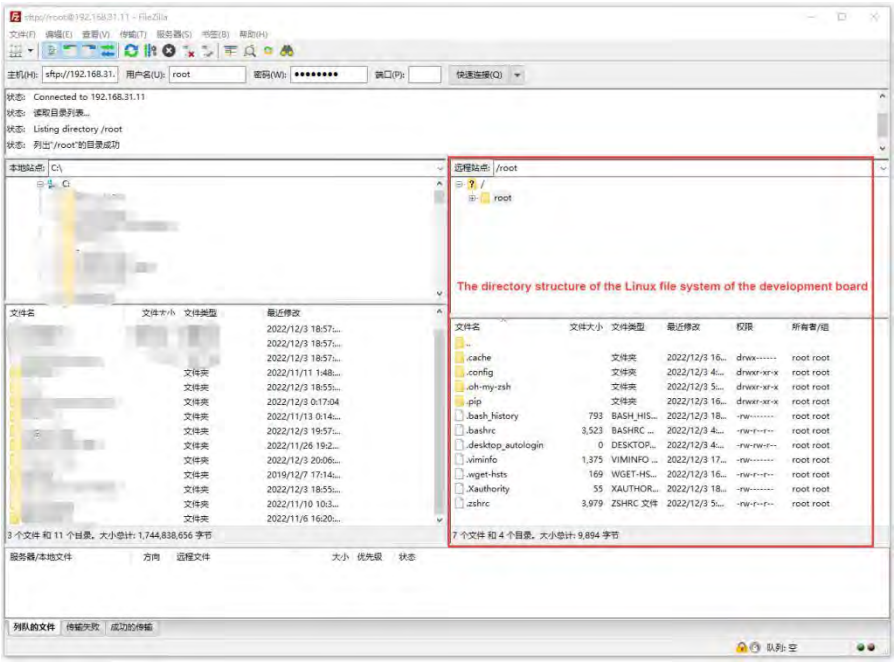


4) The method of connecting the development board is shown in the figure below:

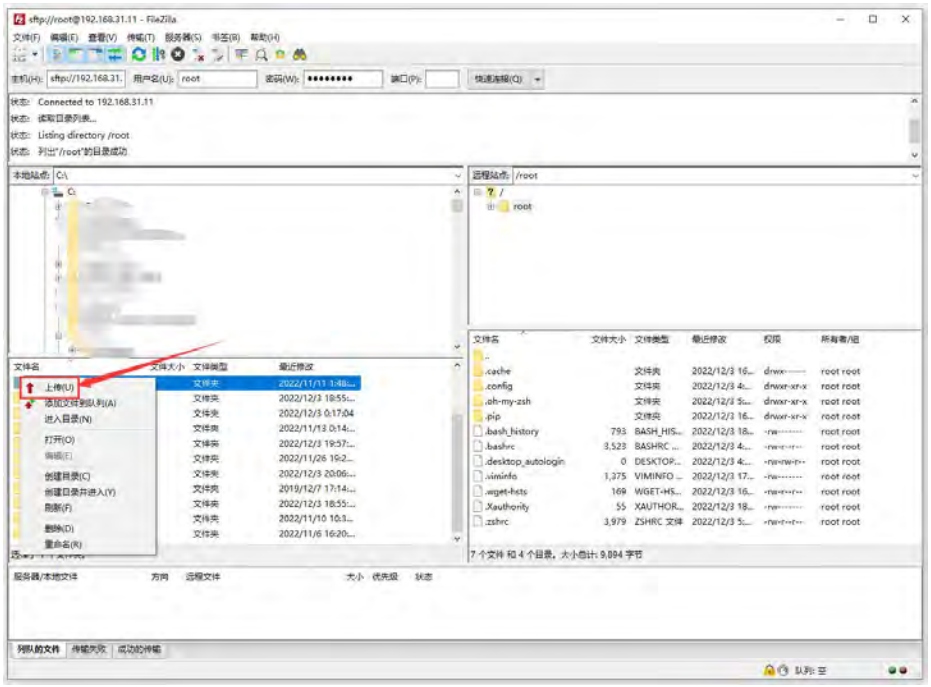
5) Then choose to **save the password**, and then click **OK**

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

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



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



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

10) The method of uploading a folder is the same as that of uploading a file, so I won't go into details here

3. 36. Instructions for using the switch logo

1) By default, the switch logo will only be displayed in the desktop version of the system

2) Set the **bootlogo** variable to **false** in **/boot/orangepiEnv.txt** to turn off the switch logo

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

3) Set the **bootlogo** variable to **true** in **/boot/orangepiEnv.txt** to enable the switch logo

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

4) The location of the boot logo image in the Linux system is

```
/usr/share/plymouth/themes/orangepi/watermark.png
```

5) After replacing the boot logo picture, you need to run the following command to take effect

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

3. 37. How to shut down and restart the development board

1) During the running of the Linux system, if the power is directly unplugged, the file system may lose some data. It is recommended to use the **poweroff** command to shut down the Linux system of the development board before power off, and then unplug the power

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

Note that after the development board is turned off, the power supply needs to be re-plugged to turn it on.

2) Use the **reboot** command to restart the Linux system on the development board

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

4. Linux SDK——orangepi-build instruction

4. 1. Compilation system requirements

The Linux SDK, **orangepi-build**, only supports running on X64 computers with **Ubuntu 22.04** installed, so before downloading orangepi-build, please make sure that the Ubuntu version installed on your computer is Ubuntu 22.04. The command to check the Ubuntu version installed on the computer is as follows. If the Release field does not display **22.04**, it means that the current Ubuntu version does not meet the requirements. Please replace the system before performing the following operations.

```
test@test:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04 LTS
Release:        22.04
Codename:       jammy
```

If the computer is installed with a Windows system and there is no computer with Ubuntu 22.04 installed, you can consider using **VirtualBox** or **VMware** to install an Ubuntu 22.04 virtual machine in the Windows system. But please note, do not compile orangepi-build on the WSL virtual machine, because orangepi-build has not been tested in the WSL virtual machine, so it cannot be guaranteed that orangepi-build can be used normally in WSL, and please do not compile it on the Linux system of the development

board Use orangepi-build in. The download address of the installation image of Ubuntu 22.04 **amd64** version is:

https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/22.04/ubuntu-22.04-desktop-amd64.iso

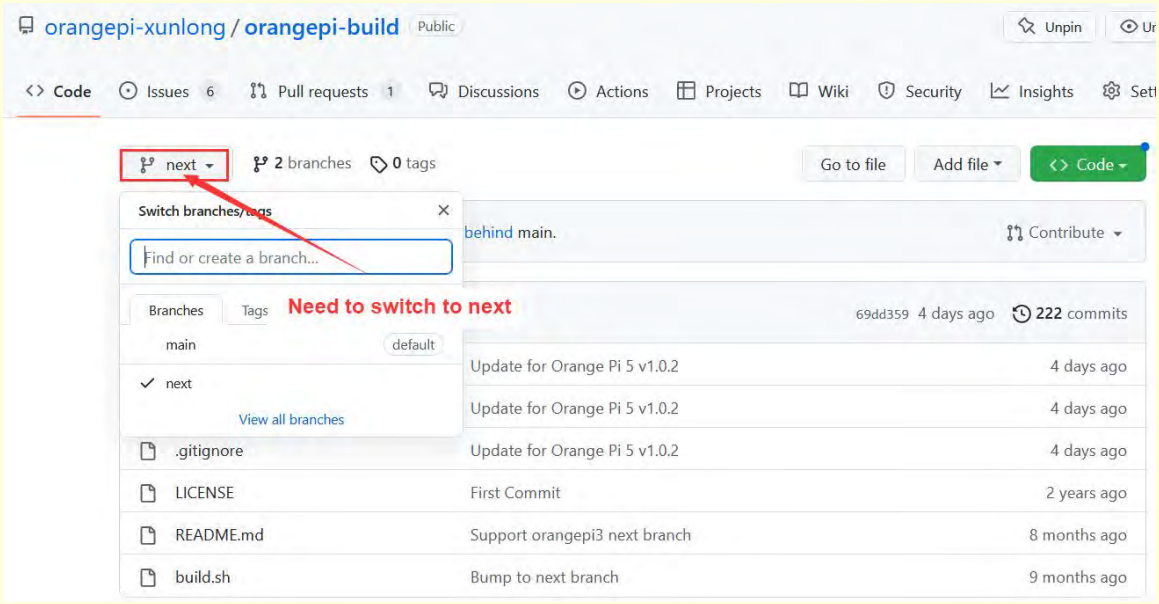
4. 2. Get the source code of linux sdk

4. 2. 1. Download orangepi-build from github

The linux sdk refers to the code of orangepi-build. orangepi-build is modified based on the armbian build system. Using orangepi-build, multiple versions of linux images can be compiled. Use the following command to download the orangepi-build code:

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

Note that the development board using the H618 Soc needs to download the source code of the next branch of orangepi-build. The above git clone command needs to specify the branch of the orangepi-build source code as next.



Downloading the orangepi-build code through the git clone command does not require entering the user name and password of the github account (the same is true for downloading other codes in this manual), if the Ubuntu PC prompts the user to

enter the github account after entering the git clone command The name and password are usually entered incorrectly in the address of the orangepi-build warehouse behind the git clone. Please check the spelling of the command carefully, instead of thinking that we forgot to provide the username and password of the github account.

The u-boot and linux kernel versions currently used by the H618 series development boards are as follows:

branch	u-boot version	Linux kernel version
current	u-boot v2018.05	linux5.4
next	u-boot v2021.07	linux6.1

The branch mentioned here is not the same thing as the branch of the orangepi-build source code, please do not confuse it. This branch is mainly used to distinguish different kernel source code versions.

We currently define the linux5.4 bsp kernel provided by Allwinner as the current branch. The latest linux6.1 LTS kernel is defined as the next branch.

After orangepi-build is downloaded, it will contain the following files and folders:

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

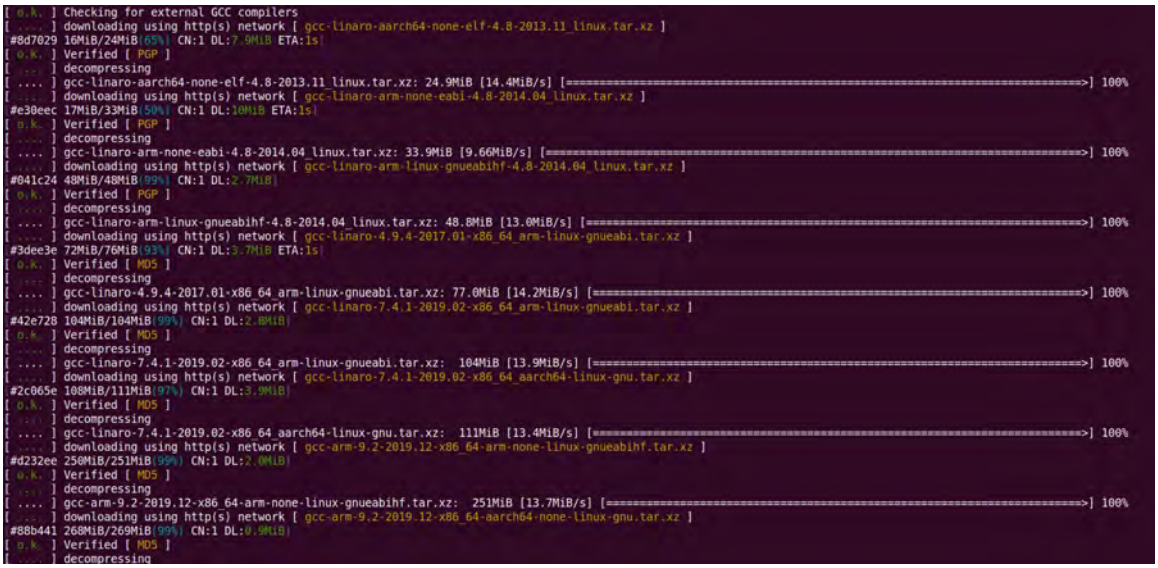
```
test@test:~/orangepi-build$ ls
build.sh  external  LICENSE  README.md  scripts
```

If you downloaded the code of orangepi-build from github, after downloading, you may find that orangepi-build does not contain the source code of u-boot and linux kernel, nor does u-boot and linux kernel need to use cross-compilation tools Chain, this is normal, because these things are stored in other separate github warehouses or some servers (the addresses will be detailed below). orangepi-build will specify the address of u-boot, linux kernel and cross-compilation toolchain in the script and configuration file. When running orangepi-build, when it finds that

there are no such things locally, it will automatically go to the corresponding place to download them.

4. 2. 2. Download the cross-compilation toolchain

When orangepi-build runs for the first time, it will automatically download the cross-compilation toolchain and put it in the **toolchains** folder. Every time you run the build.sh script of orangepi-build, it will check whether the cross-compilation toolchain in **toolchains** exists. If If it does not exist, the download will be restarted, if it exists, it will be used directly, and the download will not be repeated.



The mirror URL of the cross-compilation toolchain in China is the open source software mirror site of armbian University:

https://imola.armbian.com/dl/_toolchain/

After the **toolchains** are downloaded, it will contain multiple versions of the cross-compilation toolchain:

```
test@test:~/orangepi-build$ ls toolchains/
gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu
gcc-linaro-4.9.4-2017.01-x86_64_aarch64-linux-gnu
gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi
gcc-arm-11.2-2022.02-x86_64-arm-none-linux-gnueabihf
gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi
gcc-linaro-aarch64-none-elf-4.8-2013.11_linux
gcc-arm-9.2-2019.12-x86_64-aarch64-none-linux-gnu
```

```
gcc-linaro-5.5.0-2017.10-x86_64_arm-linux-gnueabihf
gcc-linaro-arm-linux-gnueabihf-4.8-2014.04_linux
gcc-arm-9.2-2019.12-x86_64-arm-none-linux-gnueabihf
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu
gcc-linaro-arm-none-eabi-4.8-2014.04_linux
```

The cross-compilation toolchain used to compile the H618 Linux kernel source code is:

a. linux5.4

gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu

b. linux6.1

gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu

The cross-compilation toolchain used to compile the H618 u-boot source code is:

a. v2018.05

gcc-linaro-7.4.1-2019.02-x86_64_arm-linux-gnueabi

b. v2021.07

gcc-arm-11.2-2022.02-x86_64-aarch64-none-linux-gnu

4.2.3. orangepi-build complete directory structure description

1) The orangepi-build repository does not contain the source code of the linux kernel, u-boot, and cross-compilation toolchain after downloading. The source code of the linux kernel and u-boot is stored in an independent git repository

a. The git warehouse where the linux kernel source code is stored is as follows, pay attention to switch the branch of the linux-orangepi warehouse to

a) Linux5.4

https://github.com/orangepi-xunlong/linux-orangepi/tree/**orange-pi-5.4-sun50iw9**

b) Linux6.1

https://github.com/orangepi-xunlong/linux-orangepi/tree/**orange-pi-6.1-sun50iw9**

b. The git warehouse where the u-boot source code is stored is as follows, pay attention to switch the branch of the u-boot-orangepi warehouse to

a) v2018.05

https://github.com/orangepi-xunlong/u-boot-orangepi/tree/**v2018.05-h618**

b) v2021.07

https://github.com/orangepi-xunlong/u-boot-orangepi/tree/**v2021.07-sunxi**

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 the startup script
b. **external**: Contains the configuration files needed for compiling the image, scripts for specific functions, and the source code of some programs. The rootfs compressed package cached during the compiling process is also stored in external
c. **kernel**: Store the source code of the linux kernel
d. **LICENSE**: GPL 2 license file
e. **README.md**: orangepi-build documentation
f. **output**: Store compiled u-boot, linux and other deb packages, compilation logs, and compiled images and other files
g. **scripts**: General script for compiling linux images
h. **toolchains**: Store the cross-compilation toolchain
i. **u-boot**: Store the source code of u-boot
j. **userpatches**: Store the configuration files needed to compile the script

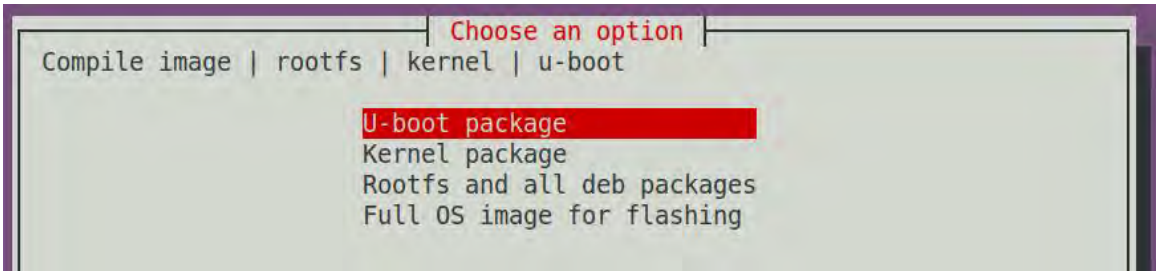
```
test@test:~/orangepi-build$ ls
build.sh  external  kernel  LICENSE  output  README.md  scripts  toolchains  u-boot  userpatches
```

4. 3. Compile u-boot

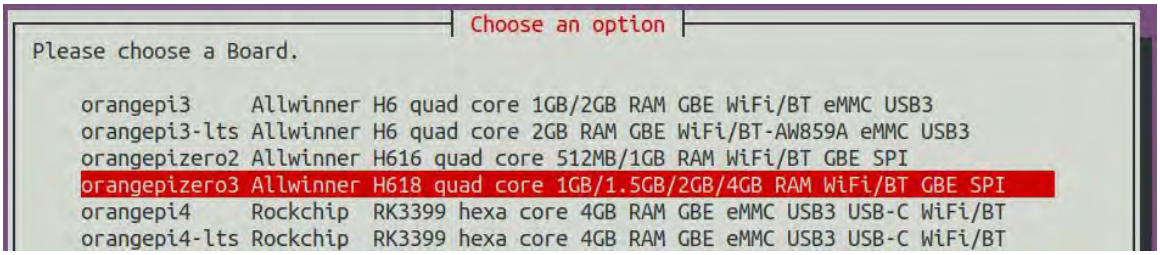
1) Run the build.sh script, remember to add sudo permission

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

2) Select **U-boot package**, then press Enter



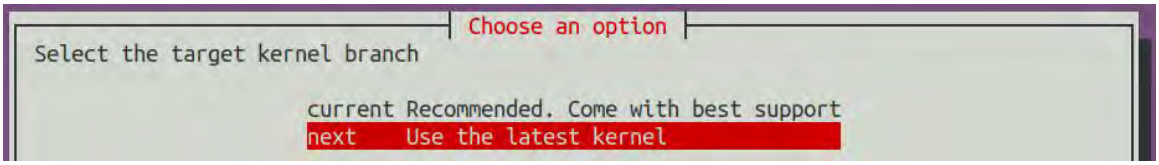
3) Then select the model of the development board



4) Then select the branch type of u-boot

a. The current branch will compile the u-boot v2018.05 code that needs to be used in the linux5.4 image

b. The next branch will compile the u-boot v2021.07 version code that needs to be used in the linux6.1 image



5) Then it will start to compile u-boot, and part of the information prompted when compiling the next branch is as follows:

a. u-boot source code version

[o.k.] Compiling u-boot [**v2021.07**]

b. The version of the cross-compilation toolchain

[o.k.] Compiler version [**aarch64-linux-gnu-gcc 11**]

c. Path to the generated u-boot deb package

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

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

[o.k.] File name [**linux-u-boot-next-orangepizero3_x.x.x_arm64.deb**]

e. Compilation time

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

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

[o.k.] Repeat Build Options [**sudo ./build.sh BOARD=orangepizero3 BRANCH=next BUILD_OPT=u-boot**]

6) View the u-boot deb package generated by compilation

```
test@test:~/orangepi-build$ ls output/debs/u-boot/
linux-u-boot-next-orangepizero3_x.x.x_arm64.deb
```

7) When the orangepi-bulid compilation system compiles the u-boot source code, it will first synchronize the u-boot source code with the u-boot source code of the github server, 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 **(This function needs to be fully compiled once u-boot, otherwise it will prompt that the source code of u-boot cannot be found)**, otherwise the changes made will be restored, the method is as follows:

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

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

8) When debugging u-boot code, you can use the following method to update u-boot in the linux image for testing

a. First 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-next-orangepizero3_x.x.x_arm64.deb root@192.168.1.xxx:/root
```

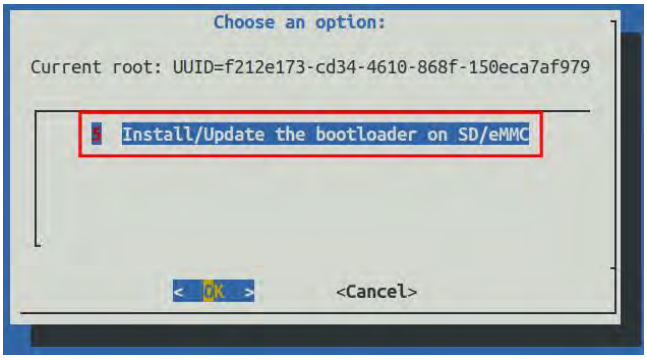
b. Install the new u-boot deb package just uploaded

```
orangepi@orangepi:~$ sudo dpkg -i linux-u-boot-next-orangepizero3_x.x.x_arm64.deb
```

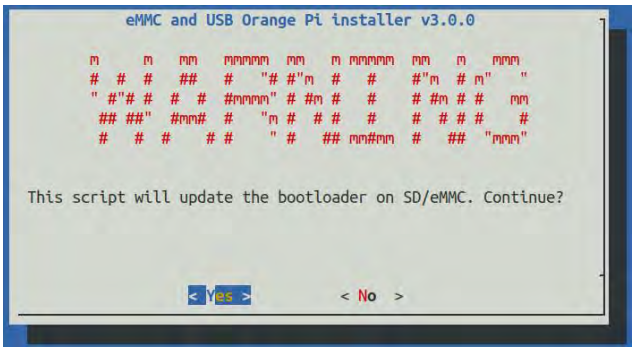
c. Then run the nand-sata-install script

```
orangepi@orangepi:~$ sudo nand-sata-install
```

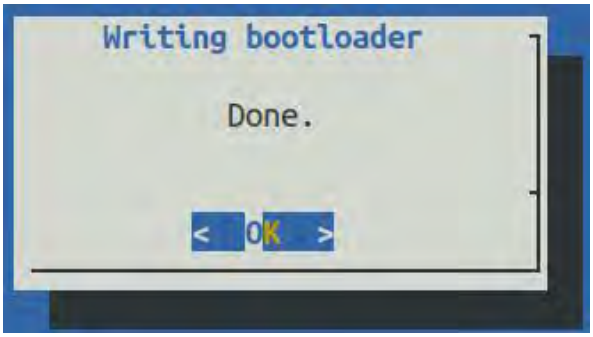
d. Then select **5 Install/Update the bootloader on SD/eMMC**



e. After pressing the Enter key, a Warning will pop up first



f. Press the Enter key again to start updating u-boot, and the following information will be displayed after the update



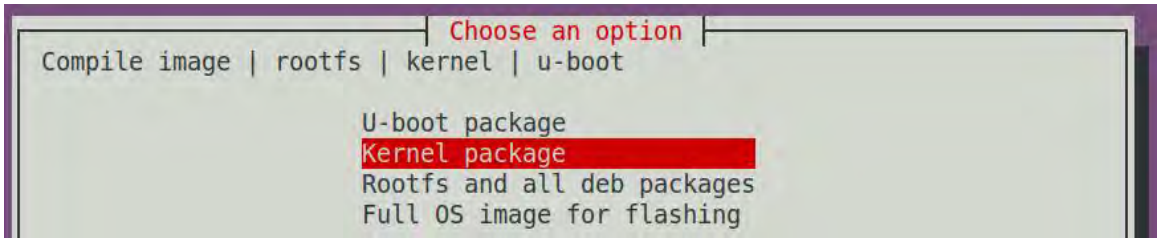
g. Then you can restart the development board to test whether the modification of u-boot takes effect

4. 4. Compile the linux kernel

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

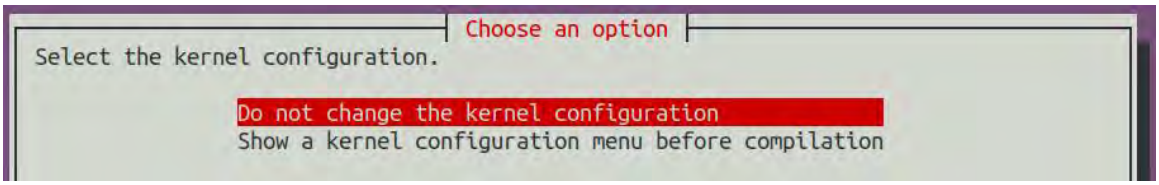
```
test@test:~/orangepi-build$ sudo ./build.sh
```

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

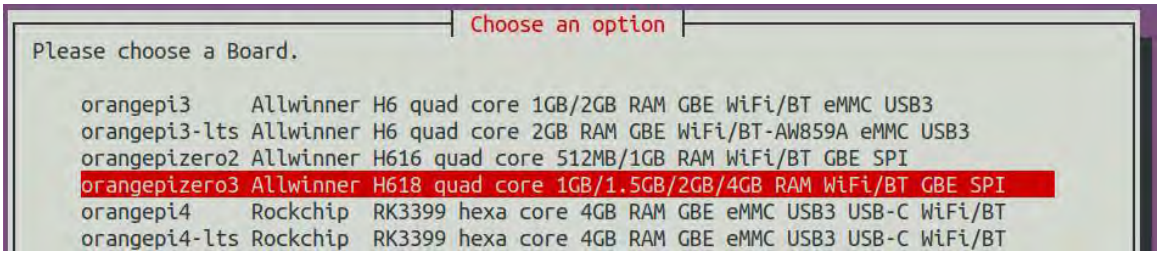


3) Then you will be prompted whether you need to display the kernel configuration interface. If you do not need to modify the kernel configuration, select the first one. If

you need to modify the kernel configuration, select the second one.



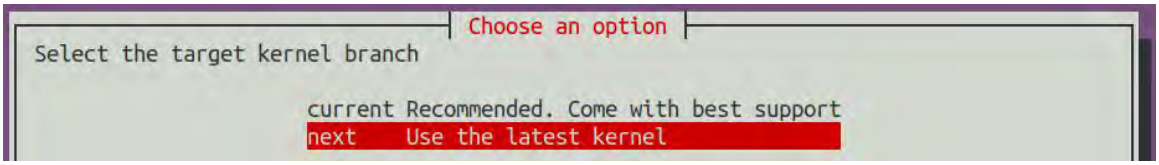
4) Then select the model of the development board



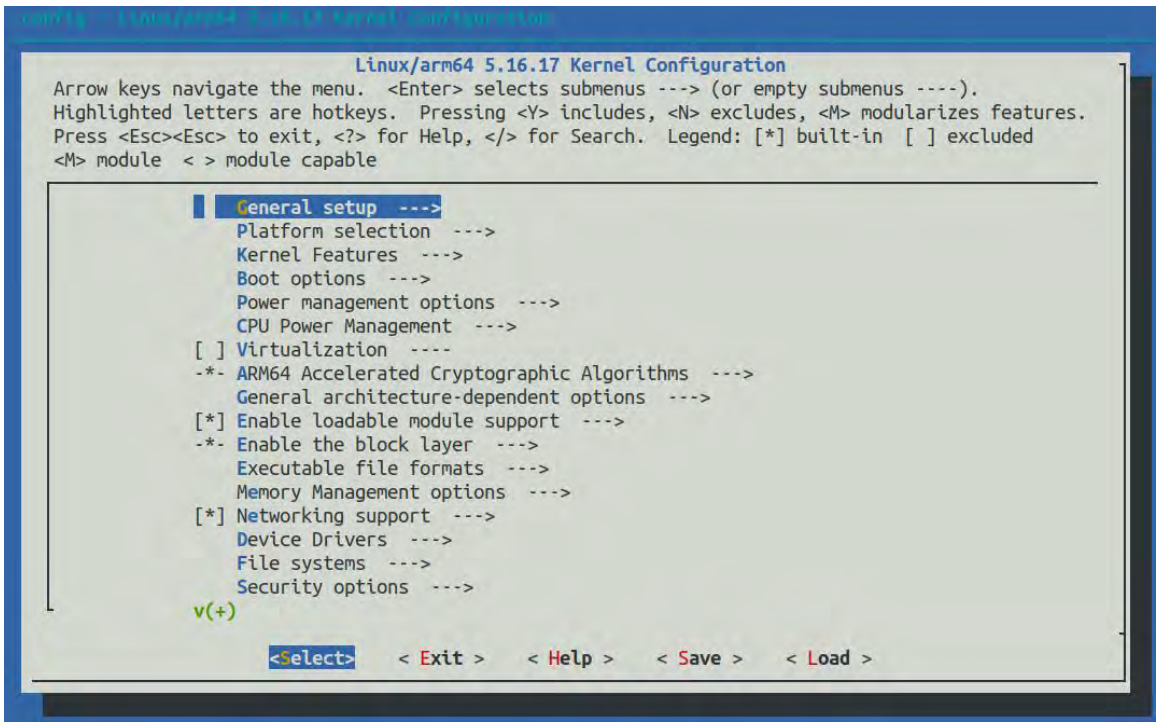
5) Then select the branch type of the kernel source code

a. current branch will compile linux5.4 kernel source code

b. The next branch will compile the linux6.1 kernel source code



6) If you choose to display the kernel configuration menu (the second option) in step 3), the kernel configuration interface opened by **make menuconfig** will pop up. At this time, you can directly modify the kernel configuration, save and exit after modification. Yes, after exiting, it will start compiling the kernel source code.



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 block the pop-up kernel configuration interface

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

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

c. If the following error is displayed when compiling the kernel, it is because the terminal interface of the Ubuntu PC is too small to display the make menuconfig interface. Please maximize the terminal of the Ubuntu PC and run the build.sh script again

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

7) Part of the information prompted when compiling the next branch kernel source code is as follows：

a. The version of the linux kernel source code

```
[ o.k. ] Compiling current kernel [ 6.1.31 ]
```

b. The version of the cross-compilation toolchain used

```
[ o.k. ] Compiler version [ aarch64-linux-gnu-gcc 11 ]
```

c. The configuration file used by the kernel by default and the path where it is stored are as follows

```
[ o.k. ] Using kernel config file
[ orangepi-build/external/config/kernel/linux-6.1-sun50iw9-next.config ]
```

d. The path of the deb package related to the kernel generated by compiling

```
[ o.k. ] Target directory [ output/debs/ ]
```

e. The package name of the compiled kernel image deb package

```
[ o.k. ] File name [ linux-image-next-sun50iw9_x.x.x_arm64.deb ]
```

f. The time used for compilation

```
[ o.k. ] Runtime [ 10 min ]
```

g. Finally, the compilation command to repeatedly compile the kernel selected last time will be displayed. Use the following command to start compiling the kernel source code directly without selecting through the graphical interface

```
[ o.k. ] Repeat Build Options [ sudo ./build.sh  BOARD=orangepizero3
BRANCH=next BUILD_OPT=kernel KERNEL_CONFIGURE=no ]
```

8) View the deb package related to the kernel generated by compilation

a. **linux-dtb-next-sun50iw9_x.x.x_arm64.deb** Contains dtb files used by the kernel

b. **linux-headers-next-sun50iw9_x.x.x_arm64.deb** Include kernel headers

c. **linux-image-next-sun50iw9_x.x.x_arm64.deb** Contains kernel images and kernel modules

```
test@test:~/orangepi-build$ ls output/debs/linux-*
output/debs/linux-dtb-next-sun50iw9_x.x.x_arm64.deb
output/debs/linux-headers-next-sun50iw9_x.x.x_arm64.deb
output/debs/linux-image-next-sun50iw9_x.x.x_arm64.deb
```

9) When the orangepi-bulid compilation system compiles the linux kernel source code, it

first synchronizes the linux kernel source code with the linux kernel source code of the github server, so if you want to modify the linux kernel source code, you first need to turn off the update function of the source code **(the linux kernel needs to be fully compiled once This function can only be turned off after the source code, otherwise it will prompt that the source code of the linux kernel cannot be found)**, otherwise the changes made will be restored, the method is as follows：

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

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

10) If the kernel has been modified, the following method can be used to update the kernel and kernel modules of the development board linux system

a. Upload the deb package of the compiled 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-next-sun50iw9_x.x.x_arm64.deb root@192.168.1.xxx:/root
```

b. Install the deb package of the new linux kernel just uploaded

```
orangepi@orangepi:~$ sudo dpkg -i linux-image-next-sun50iw9_x.x.x_arm64.deb
```

c. Then restart the development board, and then check whether the kernel-related modifications have taken effect

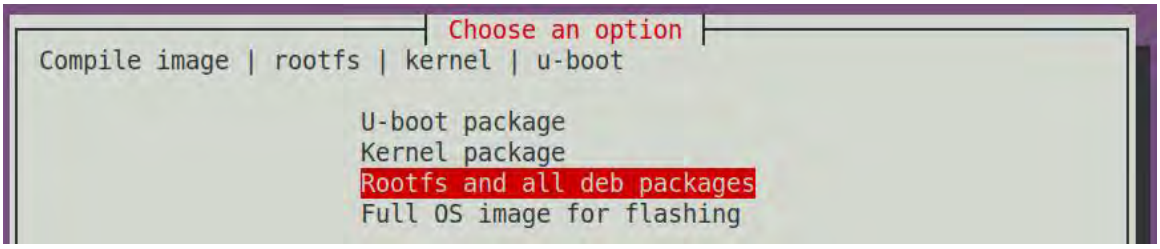
```
orangepi@orangepi:~$ sudo reboot
```

4. 5. Compile rootfs

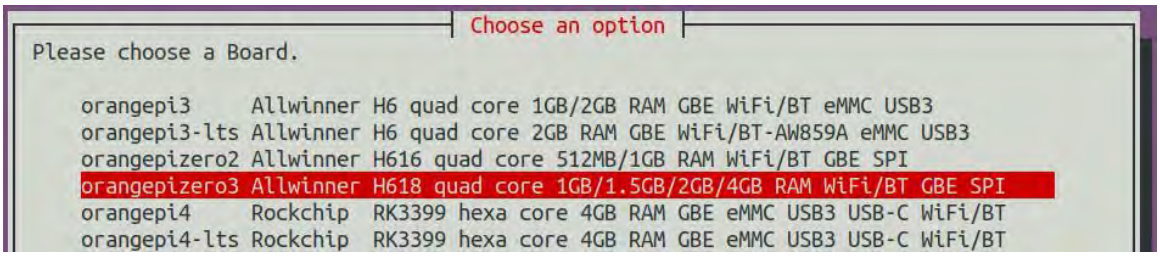
1) Run the build.sh script, remember to add sudo permission

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

2) Select **Rootfs and all deb packages**, then press Enter

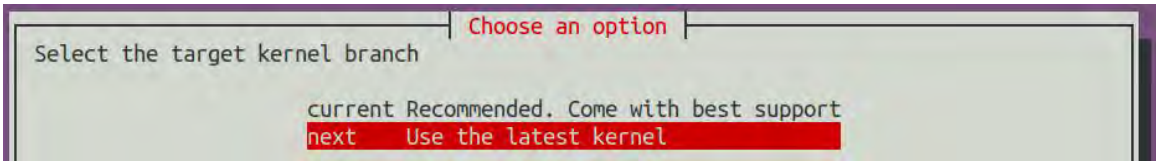


3) Then select the model of the development board

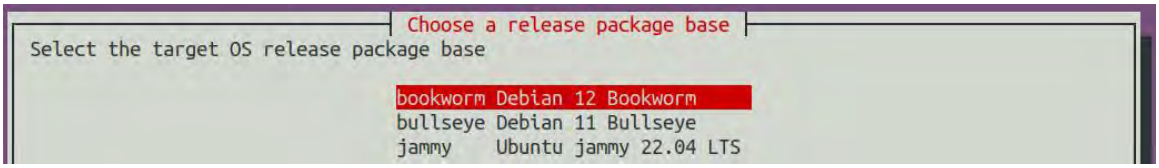


4) Then select the branch type of the kernel source code, the type of rootfs maintained by different versions of the kernel source code is different

a. The current branch can see debian11, ubuntu20.04, ubuntu22.04 three options
b. The next branch can see debian11, debian12, ubuntu22.04 three options

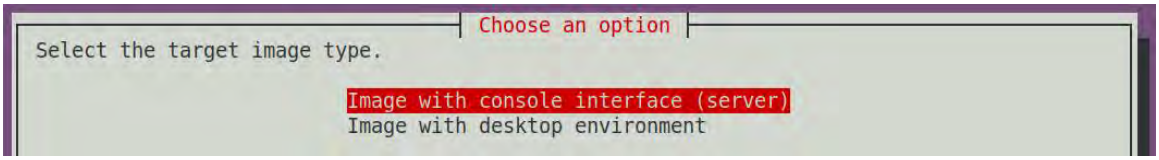


5) Then select the type of rootfs



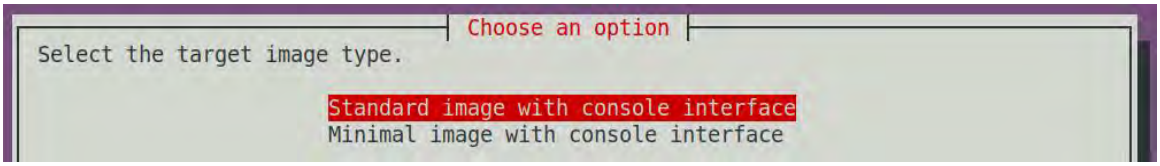
6) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small
b. **Image with desktop environment** Indicates a mirror image with a desktop, which is relatively large

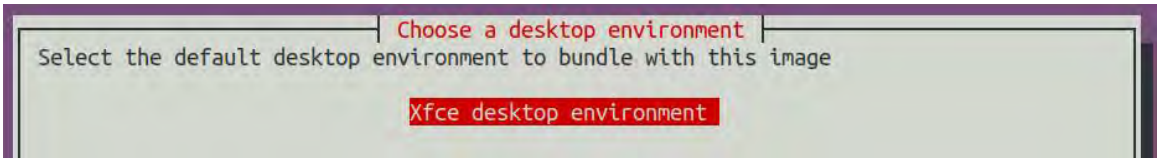


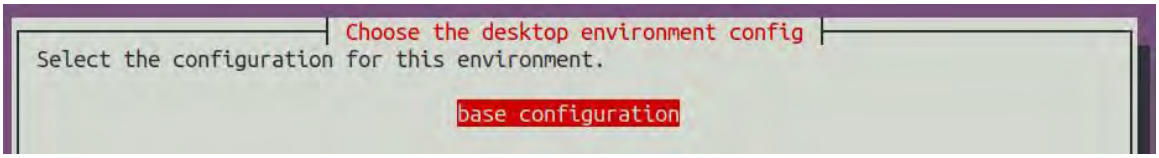
7) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the**

Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)

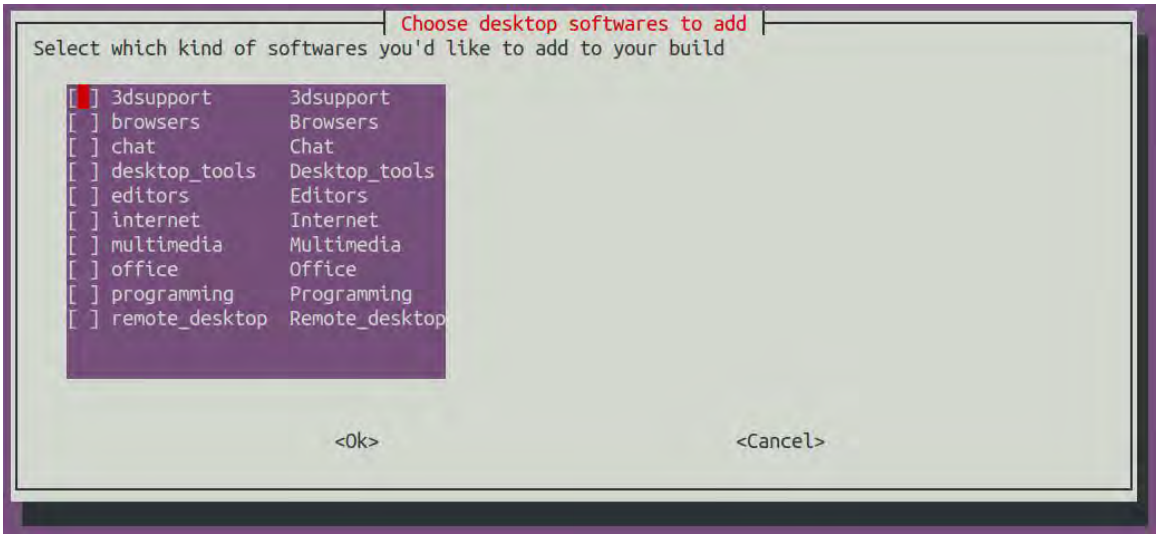


8) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently only XFCE is maintained, so please choose the XFCE type desktop





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



9) Then it will start to compile rootfs, and some information prompted during compilation is explained as follows

a. The type of rootfs

```
[ o.k. ] local not found [ Creating new rootfs cache for bullseye ]
```

b. The storage path of the compiled rootfs compressed package

```
[ o.k. ] Target directory [ orangepi-build/external/cache/rootfs ]
```

c. The name of the rootfs compressed package generated by compilation

```
[ o.k. ] File name [ bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4 ]
```

10) View the rootfs compressed package generated by compilation

a. **bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4** is the rootfs compressed package, the meaning of each field of the name is

a) **bullseye** indicates the type of linux distribution of rootfs

b) **xfce** means rootfs is the type of desktop version, if it is **cli**, it means the type of server version

c) **arm64** represents the architecture type of rootfs

d) **25250ec7002de9e81a41de169f1f89721** is the MD5 hash value generated by the package names of all software packages installed by rootfs. As long as the list of software packages installed by rootfs is not modified, this value will not change. The compilation script will use this MD5 hash value to generate Determine whether rootfs needs to be recompiled

b. **bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list** lists the package names of all packages installed by rootfs

```
test@test:~/orangepi-build$ ls external/cache/rootfs/
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.current
bullseye-xfce-arm64.5250ec7002de9e81a41de169f1f89721.tar.lz4.list
```

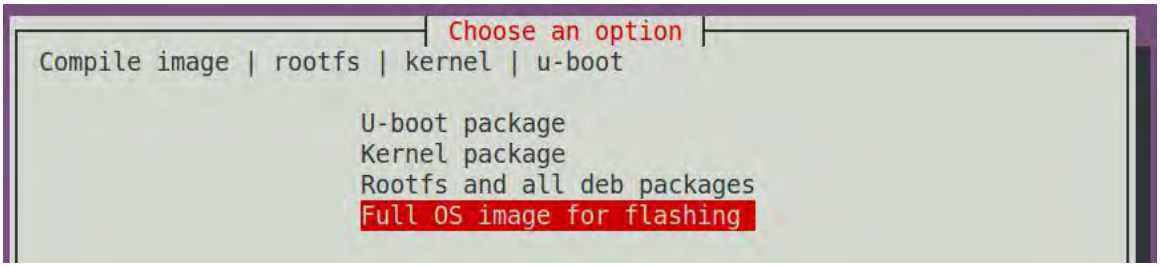
11) If the required rootfs already exists under **external/cache/rootfs**, then compiling rootfs again will directly skip the compilation process and will not restart the compilation. When compiling the image, it will also go to **external/cache/rootfs** to find out whether it has If there is rootfs available in the cache, use it directly, which can save a lot of download and compilation time

4. 6. Compile linux image

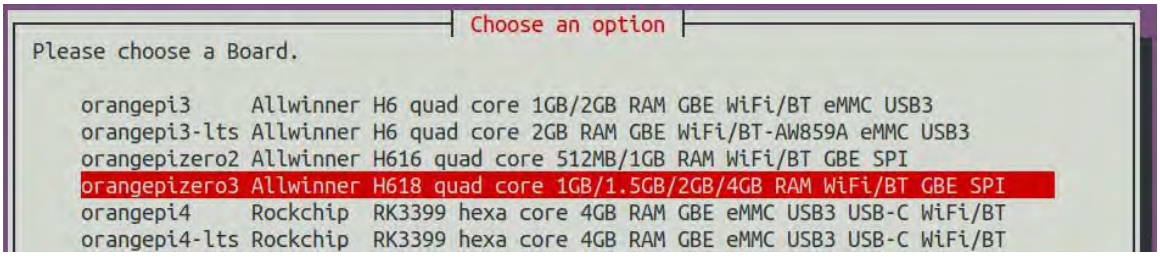
1) Run the **build.sh** script, remember to add sudo permission

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

2) Select **Full OS image for flashing**, then press Enter

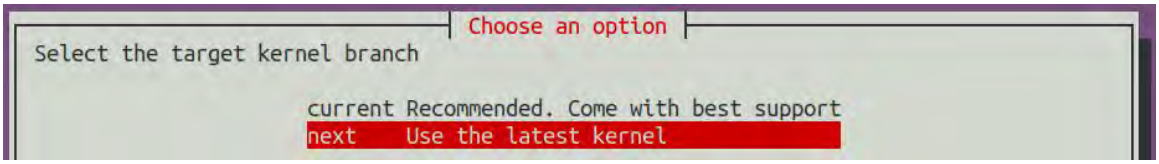


3) Then select the model of the development board



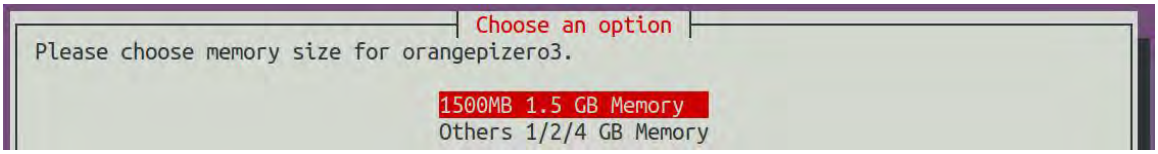
4) Then select the branch type of the kernel source code, the type of rootfs maintained by different versions of the kernel source code is different

a. The current branch can see debian11, ubuntu20.04, ubuntu22.04 three options
b. The next branch can see debian11, debian12, ubuntu22.04 three options

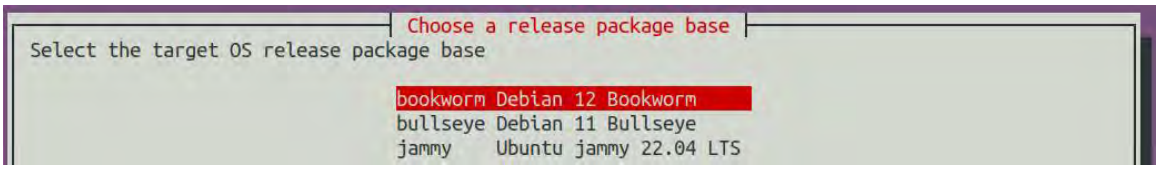


5) If you select the next branch, you will be prompted to select the size of the memory, and the current branch does not need to be selected

a. If the purchased development board has a memory size of 1.5GB, please select the first item
b. If the purchased development board is 1GB or 2GB or 4GB memory size, please choose the second item



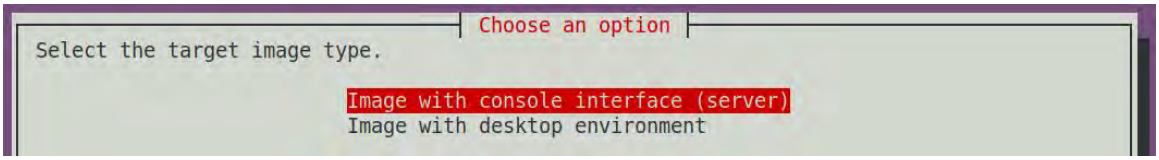
6) Then select the type of rootfs



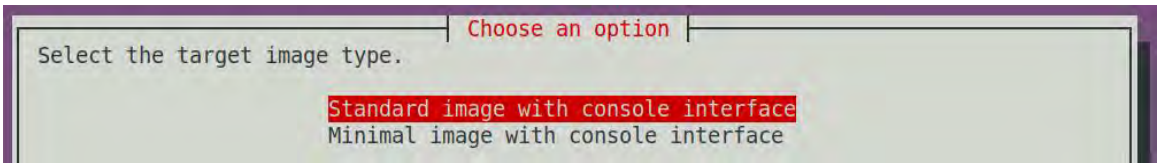
7) Then select the type of image

a. **Image with console interface (server)** Indicates the image of the server version, which is relatively small

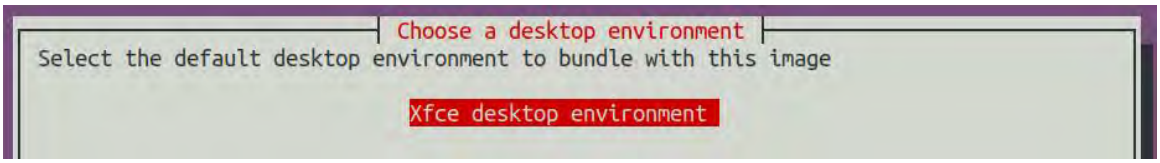
b. **Image with desktop environment** Indicates a mirror image with a desktop, which is relatively large

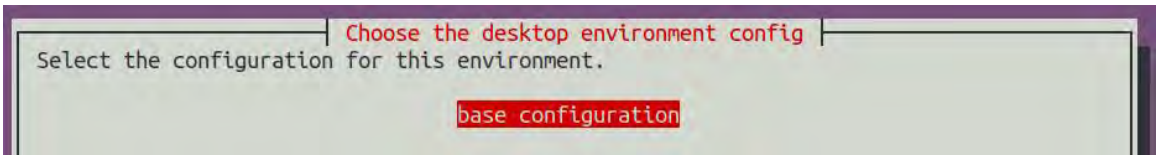


8) If you are compiling the image of the server version, you can also choose to compile the Standard version or the Minimal version. The pre-installed software of the Minimal version will be much less than that of the Standard version **(please do not choose the Minimal version if there is no special requirement, because many things are not pre-installed by default. Some functions may not be available)**

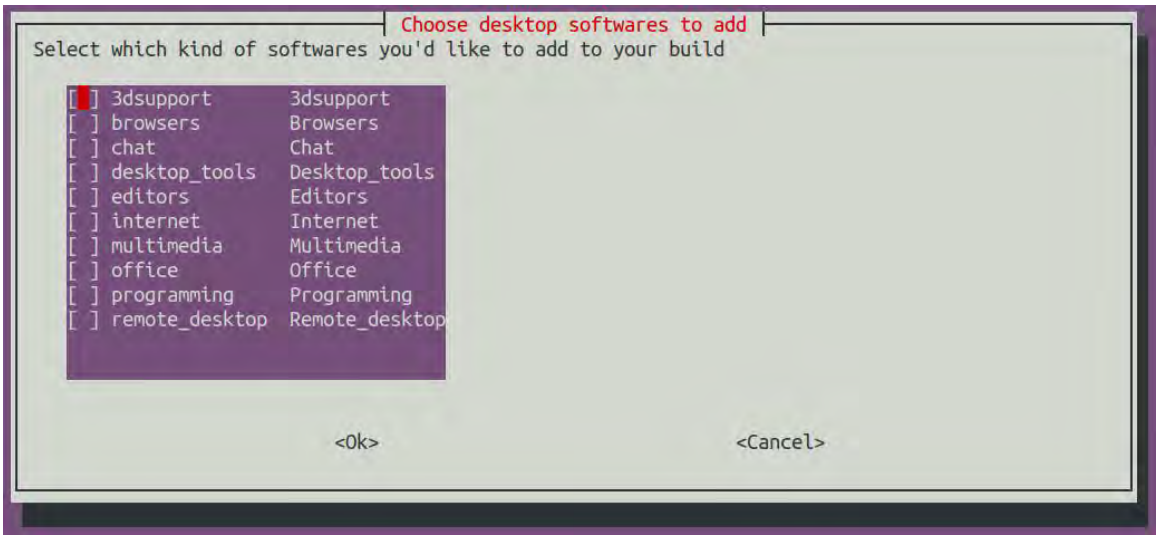


9) If you are compiling the image of the desktop version, you also need to select the type of desktop environment. Currently only XFCE is maintained, so please select the XFCE type desktop





You can then select additional packages that need to be installed. Please press the Enter key to skip directly here.



10) Then it will start to compile the linux image. The general process of compilation is as follows

a. Initialize the compilation environment of Ubuntu PC and install the software packages required for the compilation process

b. Download the source code of u-boot and linux kernel (if cached, only update the code)

c. Compile u-boot source code and generate u-boot deb package

d. Compile the linux source code and generate linux-related deb packages

e. Make the deb package of linux firmware

f. Make the deb package of the orangepi-config tool

g. Create a deb package supported by the board

h. If you are compiling the desktop image, you will also create desktop-related deb

packages

i. Check whether the rootfs has been cached, if not, recreate the rootfs, if it has been cached, directly decompress and use

j. Install the previously generated deb package into rootfs

k. Make some specific settings for different development boards and different types of images, such as pre-installing additional software packages, modifying system configuration, etc.

l. Then make an image file and format the partition, the default type is ext4

m. Then copy the configured rootfs to the mirrored partition

n. Then update initramfs

o. Finally, write the bin file of u-boot into the image through the dd command

11) After compiling the image, the following information will be prompted

a. The storage path of the compiled image

```
[ o.k. ] Done building
[ output/images/orangepizero3_x.x.x_debian_bullseye_linux6.1.xx_xfce_desktop/orangepizero3_x.x.x_debian_bullseye_linux6.1.xx_xfce_desktop.img ]
```

b. Compilation time

```
[ o.k. ] Runtime [ 19 min ]
```

c. Repeat the command to compile the image, and 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=orangepizero3 BRANCH=next BUILD_OPT=image RELEASE=bullseye BUILD_MINIMAL=no BUILD_DESKTOP=no KERNEL_CONFIGURE=yes ]
```

5. Instructions for using the OrangePi OS Arch system

5. 1. OrangePi OS Arch system function adaptation

Function	OPi OS Arch
HDMI video	OK
HDMI audio	OK
USB2.0 x 3	OK
TF card start	OK
Gigabit Ethernet	OK
Infrared receiver	OK
WiFi	OK
Bluetooth	OK
headphone audio	OK
led lights	OK
26pin GPIO	OK
26pin I2C	OK
26pin SPI1	OK
26pin UART	OK
PWM	OK
Temperature Sensor	OK
hardware watchdog	OK
Mali GPU	NO
video codec	NO
TV-OUT	NO

5. 2. OrangePi OS Arch System User Guide Instructions

First of all, please note that the OPi OS Arch system does not have a default orangepi user and password, so after burning and starting the system, it is impossible to log in remotely through the serial port and ssh directly (not even the root user). This is different from Ubuntu and Debian systems.

When the OPi OS Arch system starts for the first time, it needs to be connected to an

HDMI display, and then initialize the system settings through the user wizard (including creating a new user name and setting a password). The setup steps of the User Wizard are as follows:

a) After burning the system for the first time and enter the desktop, you will see the user wizard program shown in the figure below

b) First you need to choose the desired language

c) After selecting the language, the user guide will immediately switch to the corresponding language interface, such as the Chinese display as shown below

d) Then select the area

e) Then select the keyboard model

f) Then create a new user name and set a password

g) Then make sure that there is no problem with the selection, and then click the **install** button

h) Then wait for the installation to complete

i) After the installation is complete, you need to click the **Finish** button to restart the system

j) After restarting, the OrangePi Hello program will be started automatically. At this time, you need to remove the check status in the lower right corner **when starting up**. Otherwise, you need to manually close the OrangePi Hello program every time you start it.

At this point, you can use the newly created user name and password to log in to the OPi OS system through the serial port or ssh.

5. 3. How to set DT overlays

The multiplexing functions such as I2C/SPI/UART/PWM in the 26pin of the development board are closed by default in the dts of the kernel, and the corresponding DT overlays need to be manually opened to use.

The method of opening DT overlays in OPi OS Arch system is as follows:

1) First open the **/boot/extlinux/extlinux.conf** configuration file

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
```

2) Then open the corresponding configuration by adding **FDTOVERLAYS /dtbs/allwinner/overlay/xxx.dtbo** in **/boot/extlinux/extlinux.conf**

Note that xxx.dtbo in FDTOVERLAYS /dtbs/allwinner/overlay/xxx.dtbo needs to be replaced with the specific dtbo configuration, please do not copy it.

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
KERNEL /Image
FDT /dtbs/allwinner/sun50i-h616-orangepi-zero3.dtb
FDTOVERLAYS /dtbs/allwinner/overlay/xxx.dtbo        #Configuration that needs to be added
```

3) The storage path of xxx.dtbo in the OPi OS Arch image is as follows, please note that not all dtbos under this path can be used.

```
/boot/dtbs/allwinner/overlay/
```

4) The DT overlays configuration that can be used by the development board is as follows

Features on the development board	Corresponding DT overlays configuration
26pin - spi1	sun50i-h616-spi1-cs1-spidev.dtbo
26pin - i2c3	sun50i-h616-ph-i2c3.dtbo
26pin - uart5	sun50i-h616-ph-uart5.dtbo
26pin - pwm12	sun50i-h616-ph-pwm12.dtbo

26pin - pwm34	**sun50i-h616-ph-pwm34.dtbo**
Disable uart0 debug serial port	**sun50i-h616-disable-uart0.dtbo**
turn off the led lights	**sun50i-h616-disable-leds.dtbo**
Set the type-c usb0 interface to host mode	**sun50i-h616-usb0-host.dtbo**

5) If you need to open multiple configurations at the same time, just add the paths of multiple configurations directly behind **FDTOVERLAYS**. For example, the configurations to open i2c3 and uart5 at the same time are as follows

```
[orangepi@orangepi-pc ~]$ sudo vim /boot/extlinux/extlinux.conf
LABEL Orange Pi
KERNEL /Image
FDT /dtbs/allwinner/sun50i-h616-orangepi-zero3.dtb
FDTOVERLAYS /dtbs/allwinner/overlay/sun50i-h616-ph-i2c3.dtbo    /dtbs/allwinner/overlay/sun50i-h616-ph-uart5.dtbo
```

6) After setting, you need to restart the system to make the configuration take effect

```
[orangepi@orangepi-pc ~]$ sudo reboot
```

5. 4. How to install the software

Use the pacman package management tool to install software that is not in OPi OS. For example, the command to install the vim editor is as follows. If you want to install other software, you only need to replace vim with the package name of the software you want to install.

```
[orangepi@orangepi-pc ~]$ sudo pacman -Syy vim
```

6. Android 12 TV system instructions

6. 1. Supported Android versions

Android version	kernel version

Android 12 TV version	linux5.4

6. 2. Android 12 TV function adaptation

Function	Android12
HDMI Video	OK
HDMI Audio	OK
USB2.0 x 3	OK
TFcard boot	OK
network card	OK
infrared	OK
WIFI	OK
WIFI hotsport	OK
Bluetooth	OK
BLE Bluetooth	OK
headphone audio	OK
TV-OUT	OK
USB camera	OK
LED light	OK
Temperature Sensor	OK
Mali GPU	OK
video codec	OK

6. 3. Onboard LED light display instructions

	Green Light	Red Light
u-boot startup phase	**Off**	**Bright**
The kernel boots into the system	**Bright**	**Off**

6. 4. Android method to return to the previous interface

1) We generally use the mouse and keyboard to control the Android system of the development board. When entering certain interfaces and need to return to the previous

interface or desktop, we can only return through the **right mouse button**, and the keyboard cannot return.

2) If you have purchased the infrared remote control (other remote control does not work) and the expansion board that match the development board, after inserting the expansion board into the development board, you can return to the previous menu through the return key in the remote control. The position of the return key is shown in the figure below shown:

6. 5. How to use ADB

6. 5. 1. Use network connection adb debugging

Using the network adb does not require a USB Type C interface data cable to connect the computer and the development board, but to communicate through the network, so first make sure that the wired or wireless network of the development board has been connected, and then obtain the IP address of the development board, and then to use.

1) Make sure that the **service.adb.tcp.port** of the Android system is set to port number 5555

```
apollo-p2:/ # getprop | grep "adb.tcp"
[service.adb.tcp.port]: [5555]
```

2) If **service.adb.tcp.port** is not set, you can use the following command to set the port number of network adb in the serial port

```
apollo-p2:/ # setprop service.adb.tcp.port 5555
```

```
apollo-p2:/ # stop adbd
apollo-p2:/ # start adbd
```

3) Install adb tool on Ubuntu PC

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

4) Then connect network adb on Ubuntu PC

```
test@test:~$ adb connect 192.168.1.xxx:5555      (It needs to be modified to the IP address of the development board)
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.xxx:5555

test@test:~$ adb devices
List of devices attached
192.168.1.xxx:5555          device
```

5) Then you can log in to the android system through the adb shell on the Ubuntu PC

```
test@test:~$ adb shell
apollo-p2:/ #
```

6.5.2. Use the data cable to connect to adb debugging

1) Prepare a data cable with USB Type C interface, plug one end of the USB interface into the USB interface of the computer, and insert one end of the USB Type C interface into the power interface of the development board. In this case, the development board is powered by the USB interface of the computer, so please make sure that the USB interface of the computer can provide the most power to drive the development board

2) Install adb tool on Ubuntu PC

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

3) View the identified ADB device

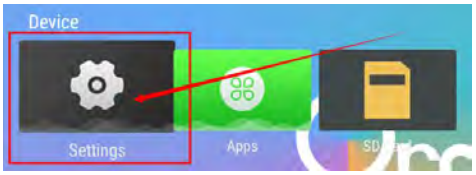
```
test@test:~$ adb devices
List of devices attached
4c00146473c28651dd0     device
```

4) Then you can log in to the android system through the adb shell on the Ubuntu PC

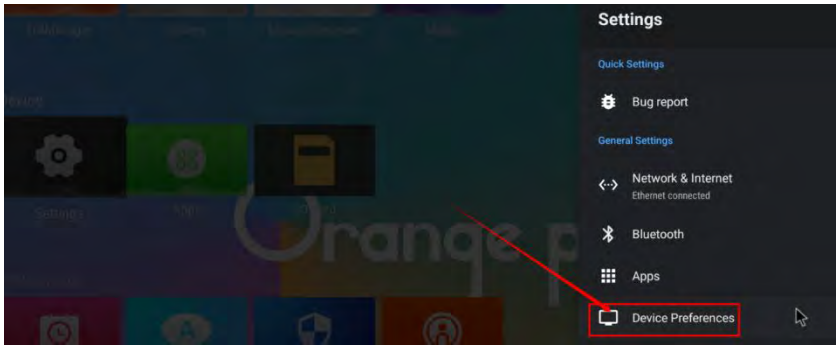
```
test@test:~$ adb shell
apollo-p2:/ $
```

6. 6. View the method of setting HDMI display resolution

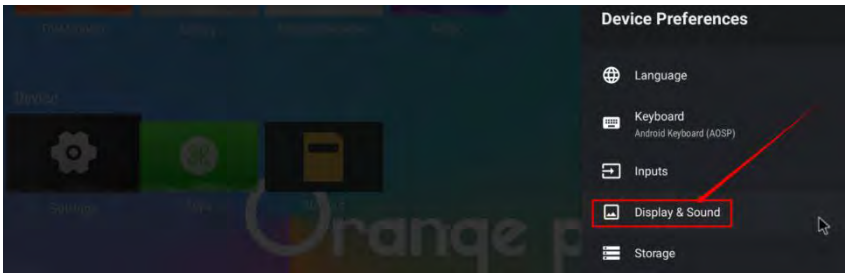
1) First enter **Settings**



2) Then select **Device Preferences**



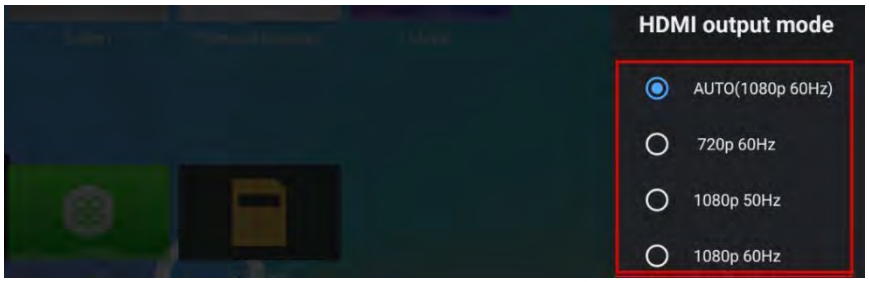
3) Then select **Display & Sound**



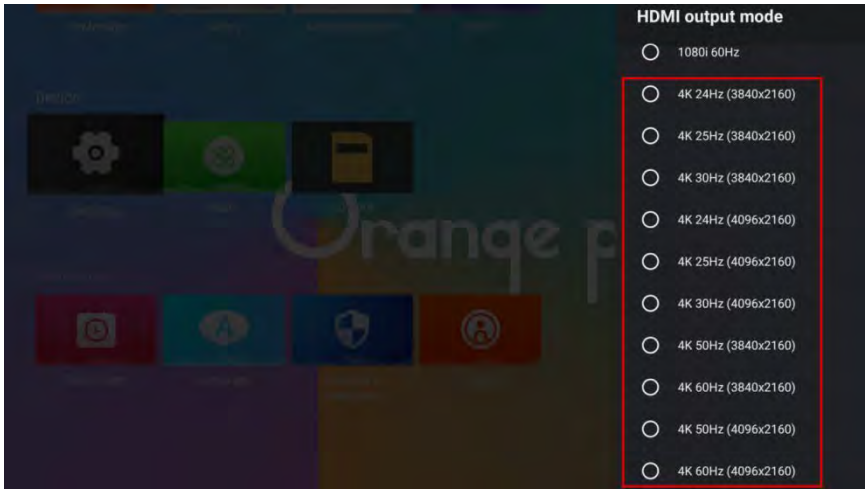
4) Then select **Advanced display settings**

5) Then select **HDMI outpu mode**

6) Then you can see the list of resolutions supported by the display. At this time, click the corresponding option to switch to the corresponding resolution. Please note that the resolutions supported by different monitors may be different. If you connect to a TV, you will generally see more resolution options than the picture below.



7) The HDMI output of the development board supports 4K display. When connected to a 4K TV, you can see the option of 4K resolution



6. 7. HDMI to VGA display test

1) First, you need to prepare the following accessories
   a. HDMI to VGA Converter

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

   c. A monitor or TV that supports VGA interface

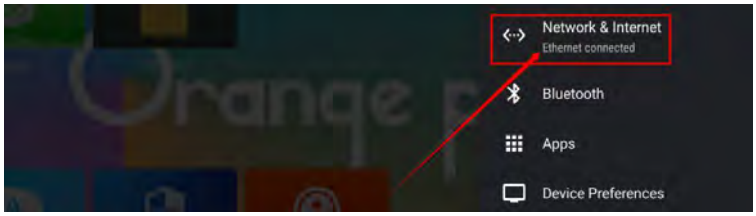
2) HDMI to VGA display test as shown below

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

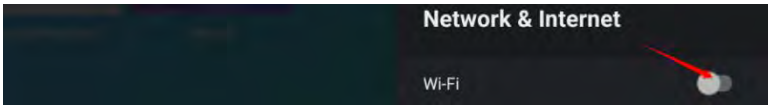
6. 8. WI-FI connection method

1) First select **Settings**

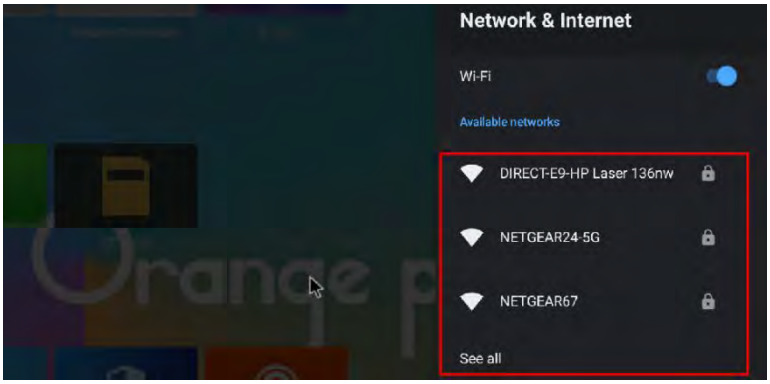
2) Then select **Network & Internet**



3) Then open WI-FI

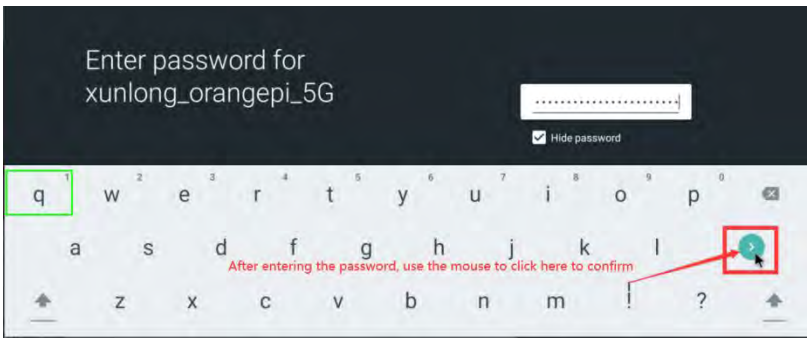


4) After turning on WI-FI, you can see the searched signal under **Available networks**

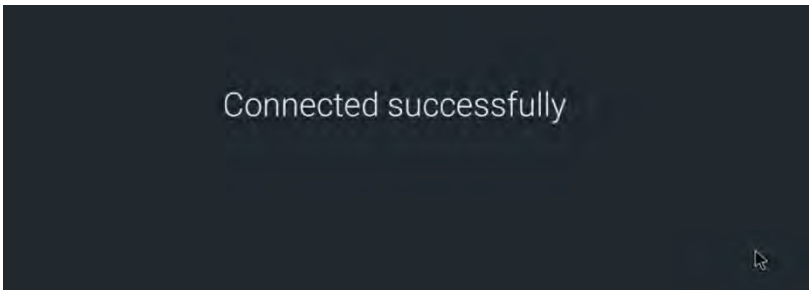


5) After selecting the WI-FI you want to connect to, the password input interface shown in the figure below will pop up

6) Then use the keyboard to enter the password corresponding to the WI-FI, and then use the **mouse** to click the Enter button in the virtual keyboard to start connecting to the WI-FI



7) After the WI-FI connection is successful, the display is as shown in the figure below

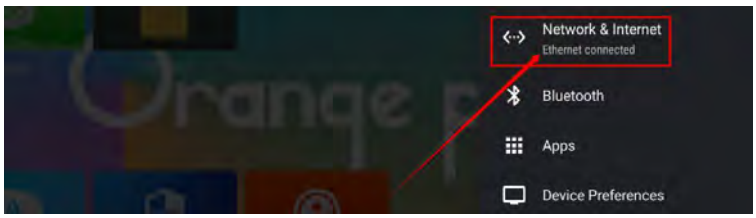


6. 9. How to use WI-FI hotspot

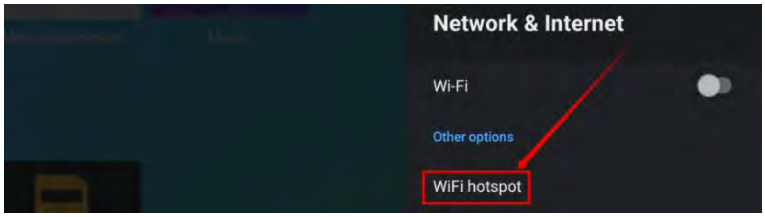
1) First, please make sure that the Ethernet port is connected to the network cable and can access the Internet normally

2) Then select **Settings**

3) Then select **Network & Internet**



4) Then select **WIFI hotspot**

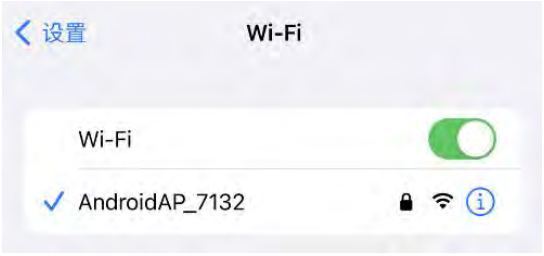


5) Then turn on **Hotspot Enable**, you can also see the name and password of the generated hotspot in the figure below, remember them and use them when connecting to the hotspot (if you need to modify the name and password of the hotspot, you need to turn off **Hotspot Enable** first, before modifying)

6) At this point, you can take out your mobile phone. If everything is normal, you can find the WIFI hotspot with the same name **(here AndroidAP_7132)** displayed under the **Hotspot name** in the above picture in the WI-FI list searched by the mobile phone. Then you can click **AndroidAP_7132** to connect to the hotspot, and the password can be seen under the **Hotspot password** in the above picture

7) After the connection is successful, it will be displayed as shown in the figure below (the interface of different mobile phones will be different, the specific interface is subject to the display of your mobile phone). At this point, you can open a webpage on your

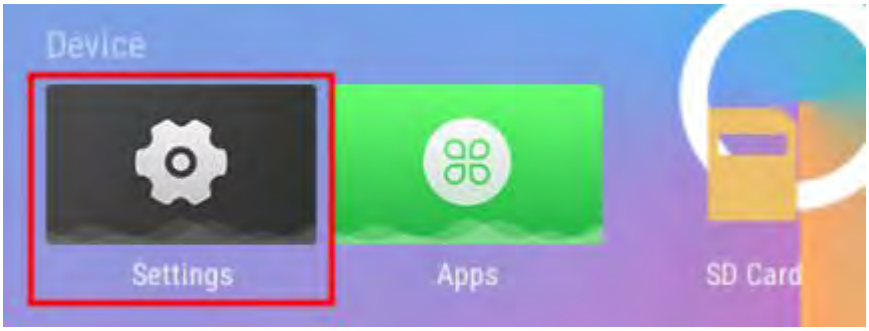
mobile phone to see if you can access the Internet. If you can open the webpage normally, it means that the **WI-FI Hotspot** of the development board can be used normally.



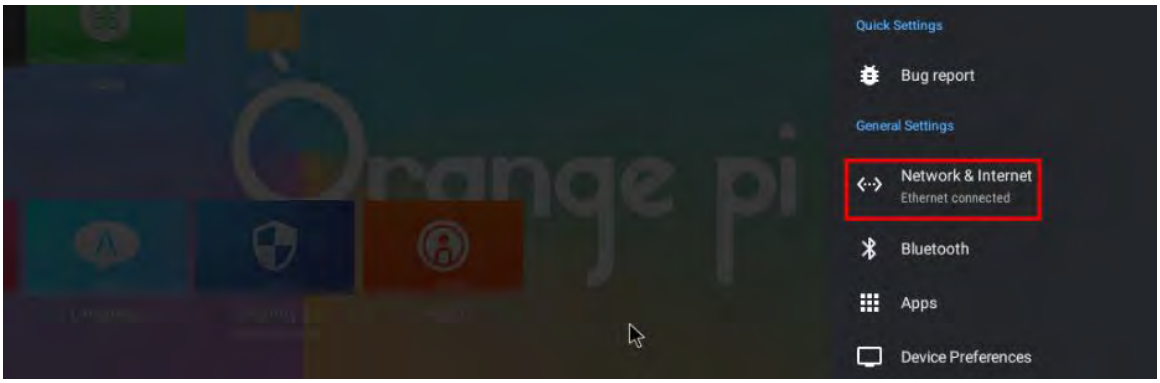
6. 10. How to check the IP address of the Ethernet port

1) First, please make sure that the Gigabit Ethernet port of the development board is connected to a router or switch

2) Then open **Settings**



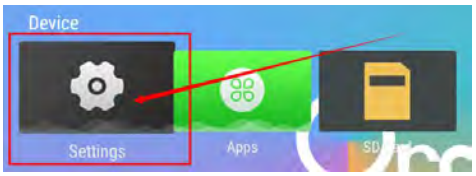
3) Then select **Network & Internet**



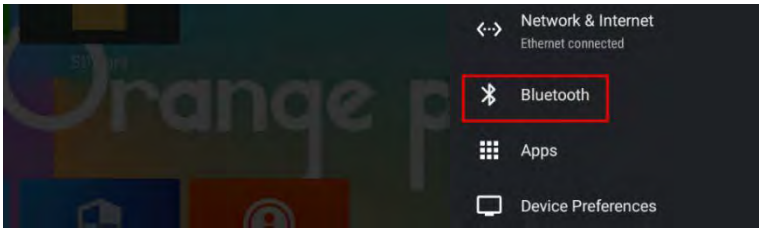
4) Then you can see the IP address of the development board's wired network port at the position shown in the figure below

6. 11. Bluetooth connection method

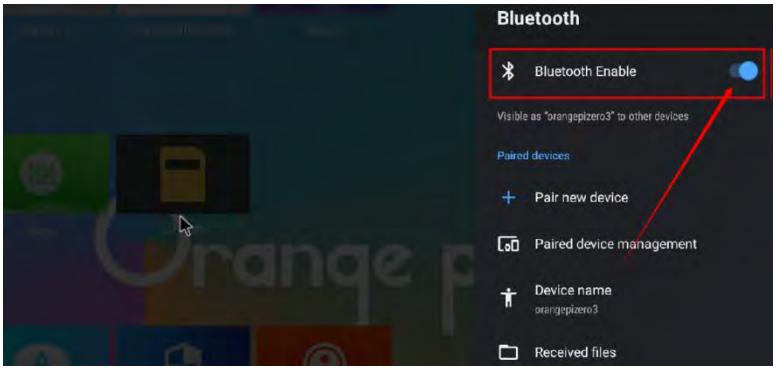
1) First select **Settings**



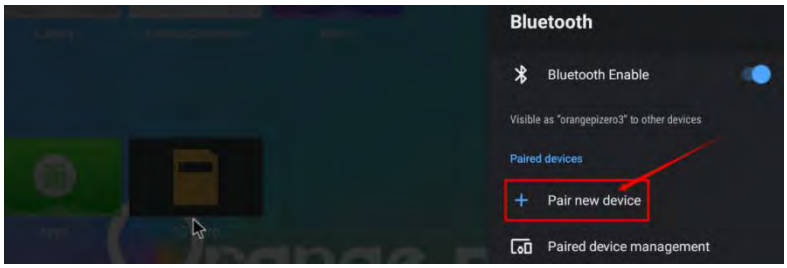
2) Then select **Bluetooth**



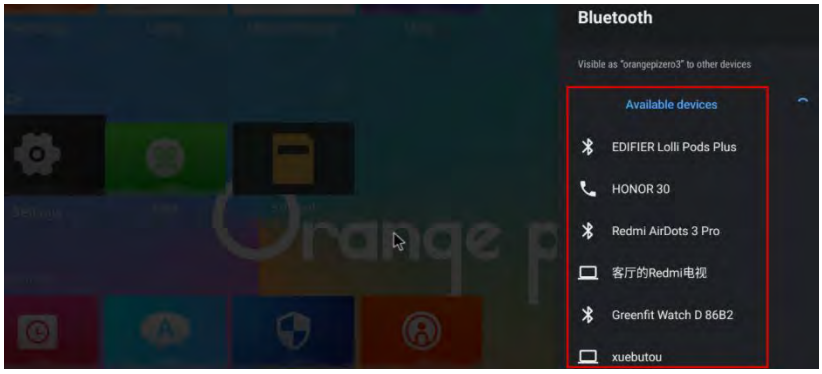
3) Then open **Bluetooth Enable**



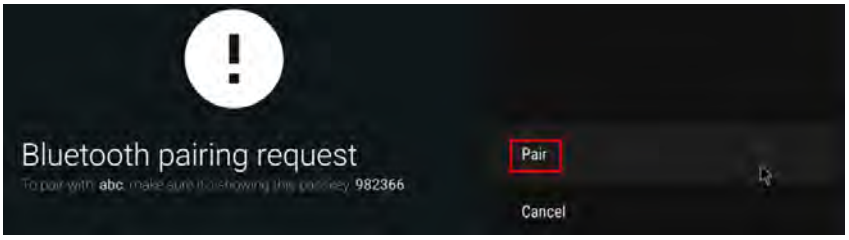
4) Then click **Pair new device** to start scanning the surrounding Bluetooth devices



5) The searched Bluetooth devices will be displayed under **Available devices**

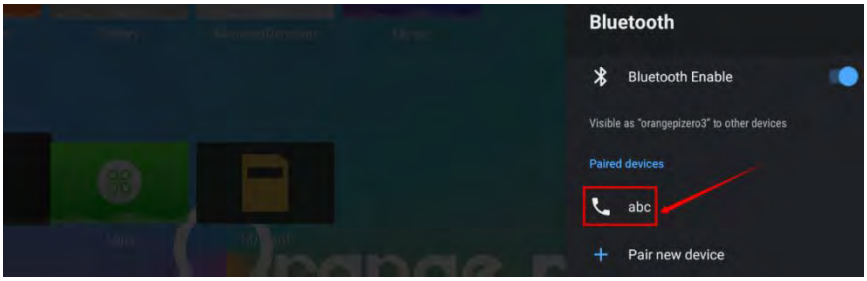


6) Then click the Bluetooth device you want to connect to start pairing. When the following interface pops up, please use the mouse to select the **Pair** option

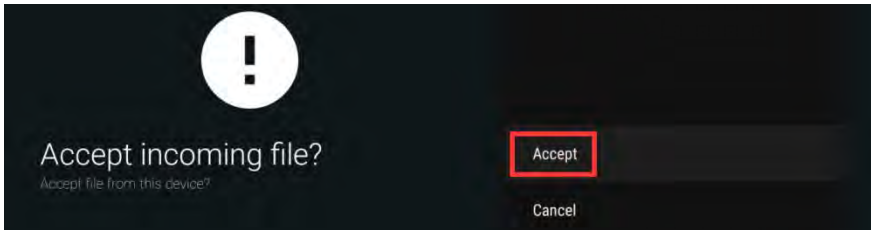


7) The test here is the configuration process of the development board and the Bluetooth of the **Android mobile phone**. At this time, the following confirmation interface will pop up on the mobile phone. After clicking the pairing button on the mobile phone, the pairing process will start

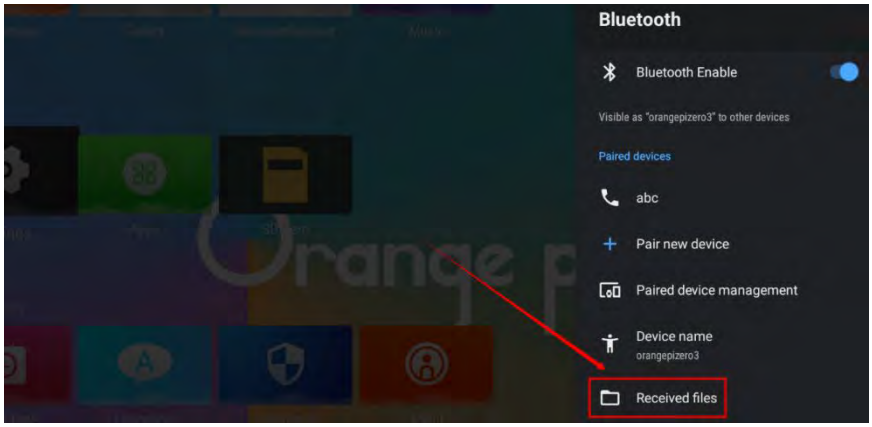
8) After the pairing is complete, open **Paired devices** and you can see the paired Bluetooth devices



9) At this time, you can use the Bluetooth of your mobile phone to send a picture to the development board. After sending, you can see the following confirmation interface in the Android system of the development board, and then click **Accept** to start receiving the picture sent by the mobile phone.



10) The pictures received by the Android system Bluetooth of the development board can be viewed in **Received files**



6. 12. How to use the USB camera

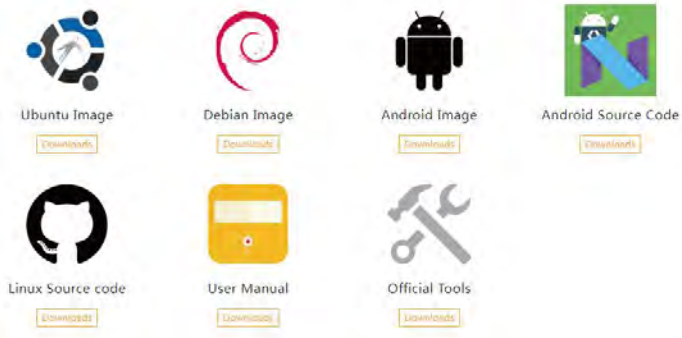
1) First insert the USB (UVC protocol) camera into the USB interface of the development board

2) If the USB camera is recognized normally, a corresponding video device node will be generated under /dev

```
console:/ # ls /dev/video0
/dev/video0
```

3) Then make sure that the adb connection between the Ubuntu PC and the development board is normal. For how to use adb, please refer to the instructions in the section on **how to use ADB**

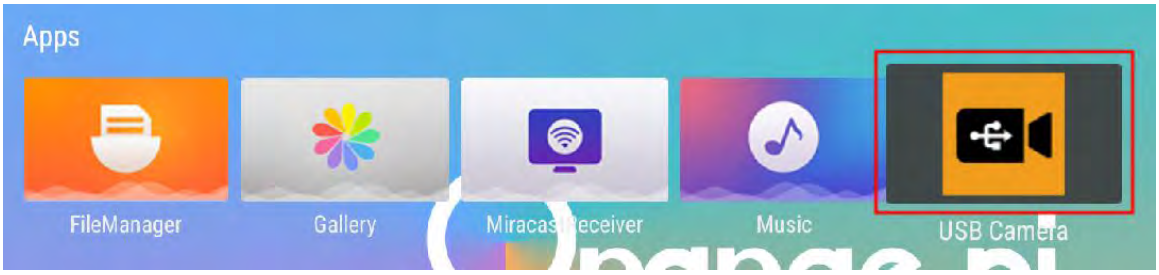
4) Download the USB camera test APP in the official tool on the development board data download page



5) Then use the adb command to install the USB camera test APP to the Android system, of course, you can also use the U disk to copy the installation

```
test@test:~$ adb install usbcamera.apk
```

6) After installation, you can see the startup icon of the USB camera on the Android desktop

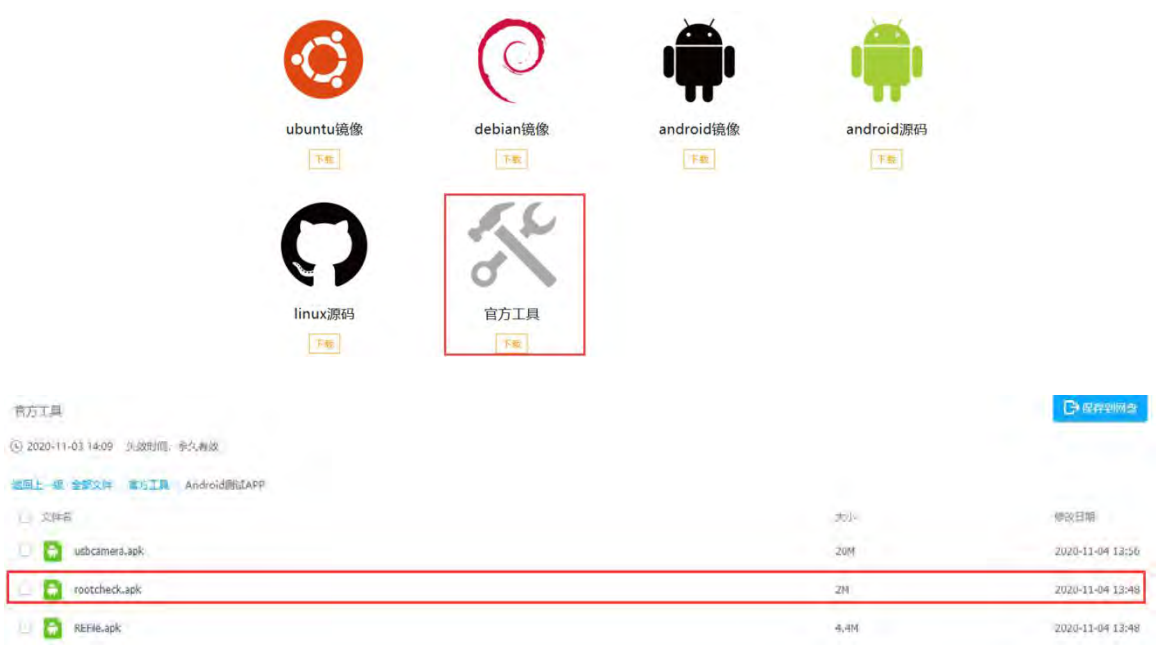


7) Then double-click to open the USB camera APP and you can see the output video of the USB camera

6. 13. Android system ROOT instructions

The Android system released by OrangePi has been ROOT, you can use the following method to test.

1) Download **rootcheck.apk** from the official tool on the development board data download page

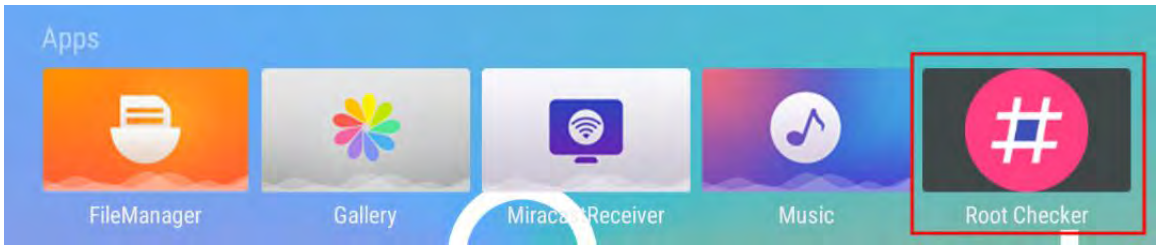


2) Then make sure that the adb connection between the Ubuntu PC and the development board is normal. For how to use adb, please refer to the instructions in the section on **how to use ADB**

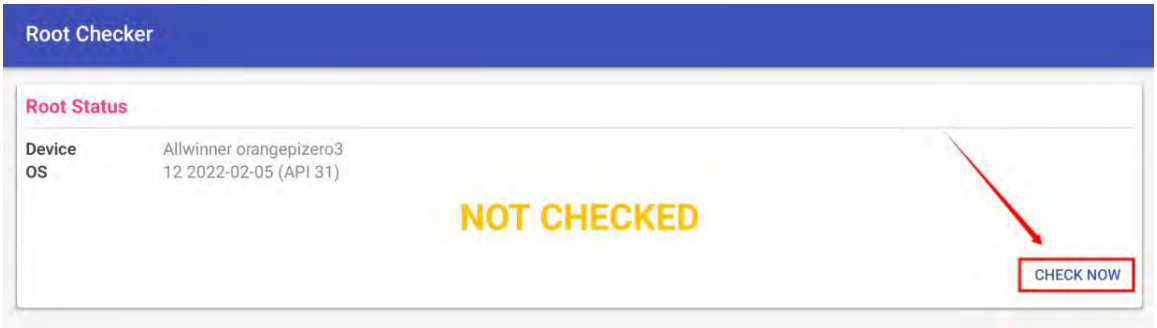
3) Then use the adb command to install rootcheck.apk to the Android system, of course, you can also use the U disk to copy the installation

```
test@test:~$ adb install rootcheck.apk
```

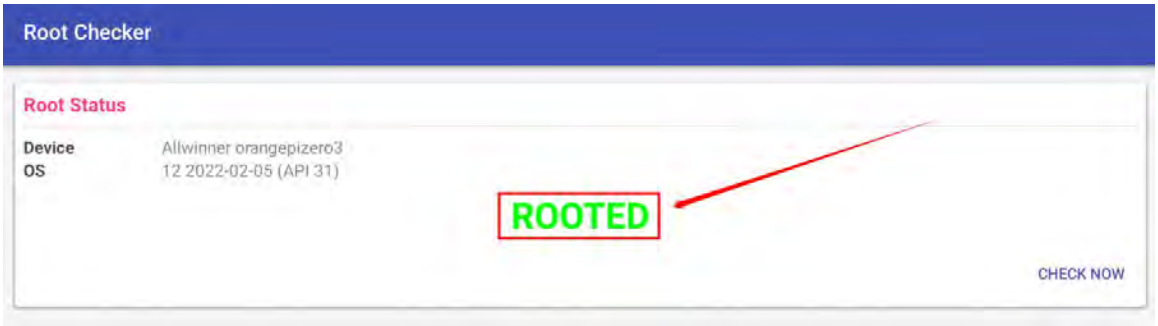
4) After installation, you can see the startup icon of the ROOT test tool on the Android desktop



5) The display interface after opening the **ROOT test tool** for the first time is shown in the figure below



6) Then you can click **CHECK NOW** to start checking the ROOT status of the Android system. After the check, the display is as follows, and you can see that the Android system has obtained the ROOT permission

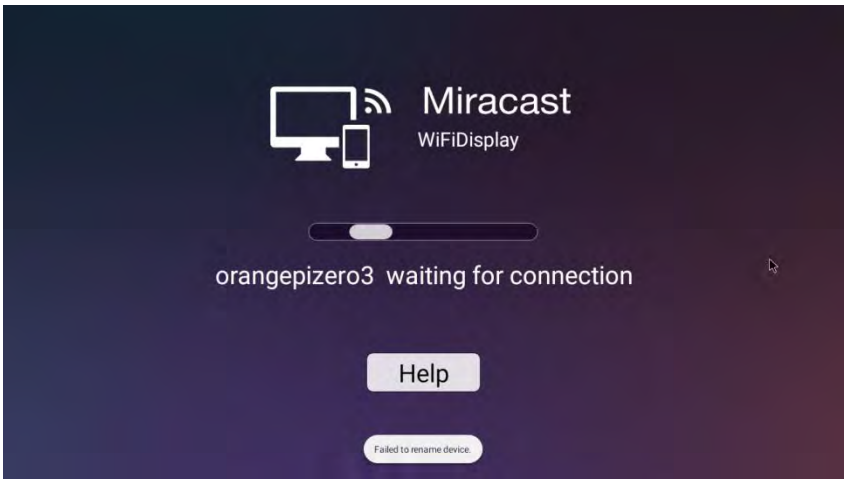


6. 14. The method of using MiracastReceiver to cast the mobile phone screen to the development board

1) First, please make sure that the development board and the mobile phone are connected to the same WIFI hotspot. For the method of connecting the development board to WIFI, please refer to the description in the section of **WI-FI connection method**

2) Then open the **MiracastReceiver** application in the Android system of the development board

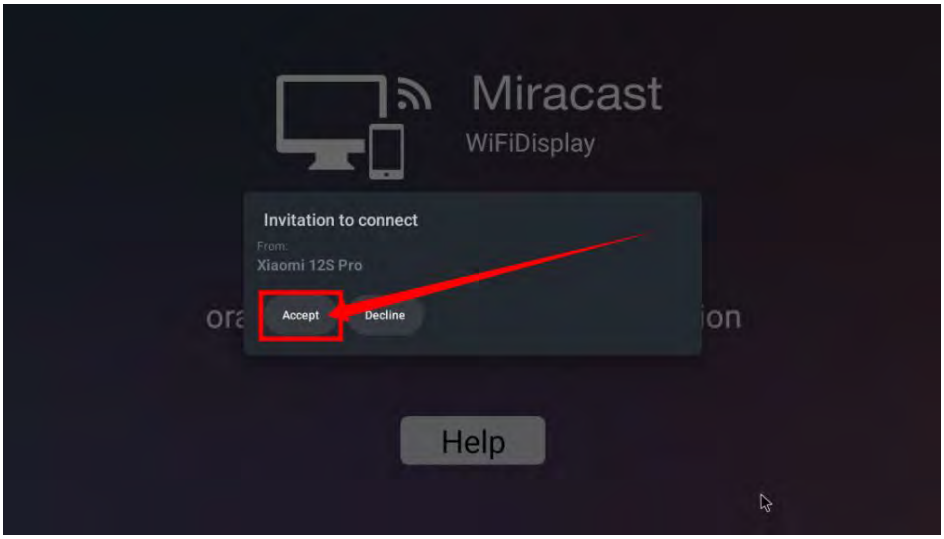
3) The interface after **MiracastReceiver** is opened is as follows



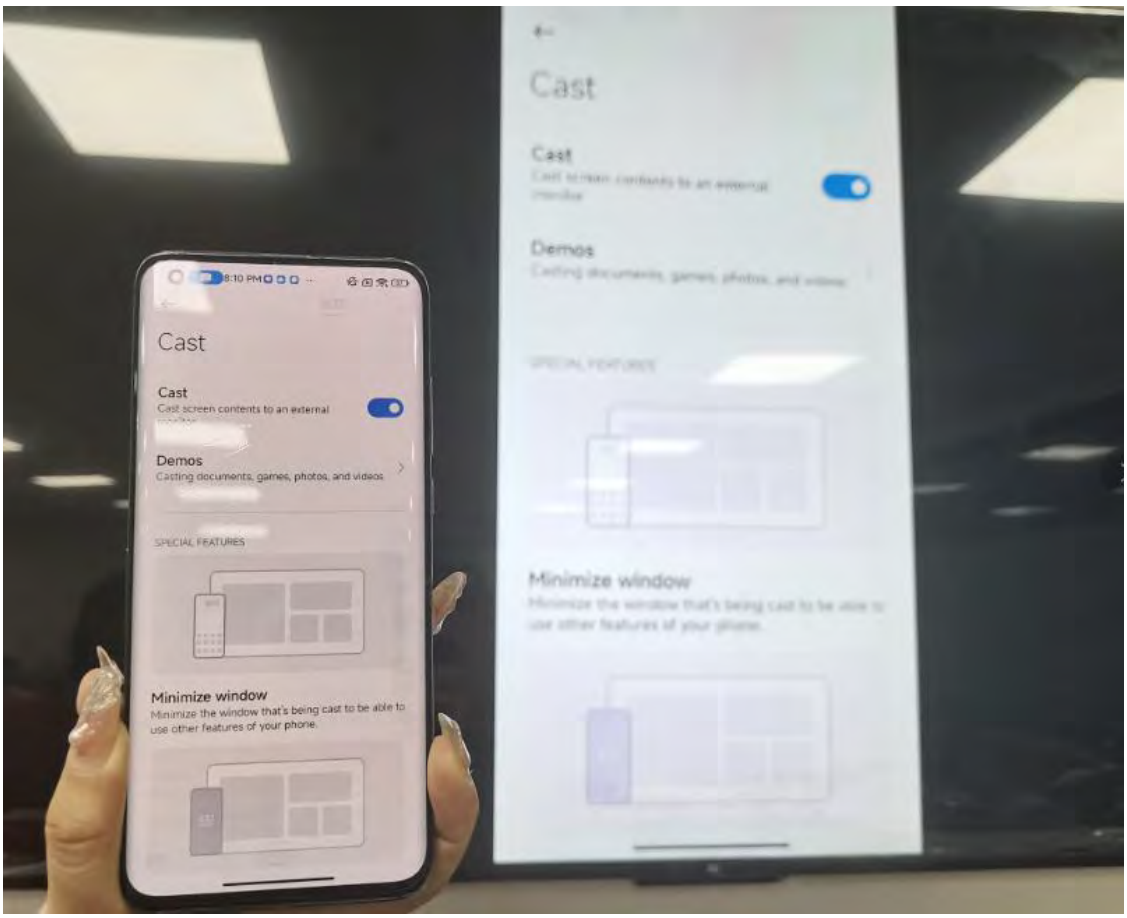
4) Then find the screen projection function in the mobile phone settings. **Here we take the Mi 12S Pro mobile phone as an example**. Please research other brands of mobile phones by yourself. As shown in the figure below, click the button in the red box to open the screen projection function of the mobile phone

5) After waiting for a while, you can see the searched and connectable devices on the mobile phone, and then we can select the device corresponding to the development board to connect

6) Then the selection box shown in the figure below will pop up on the **MiracastReceiver** application interface of the development board, here we can select **Accept**



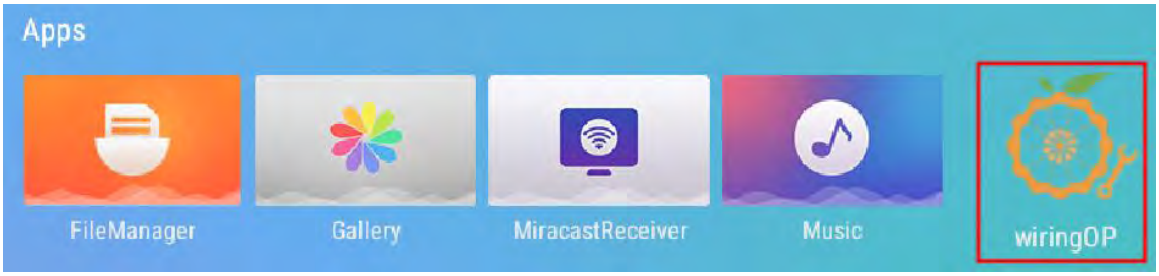
7) Then you can see the content of the mobile phone screen on the HDMI screen connected to the development board



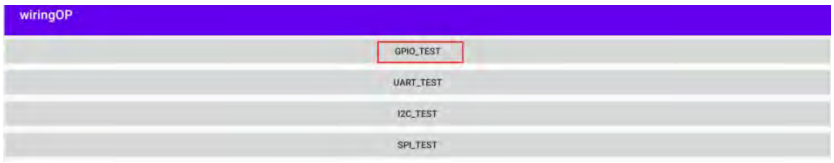
6. 15. 26pin interface GPIO, UART, SPI test

6. 15. 1. 26pin GPIO port test method

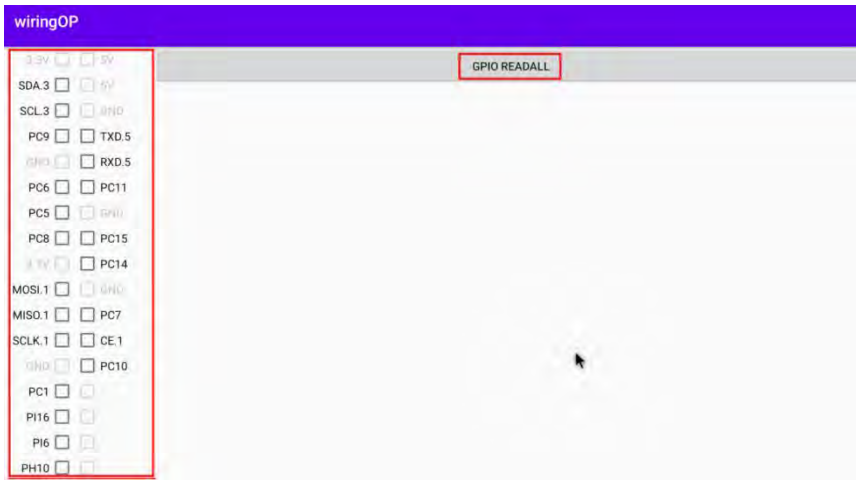
1) First open the wiringOP APP on the desktop



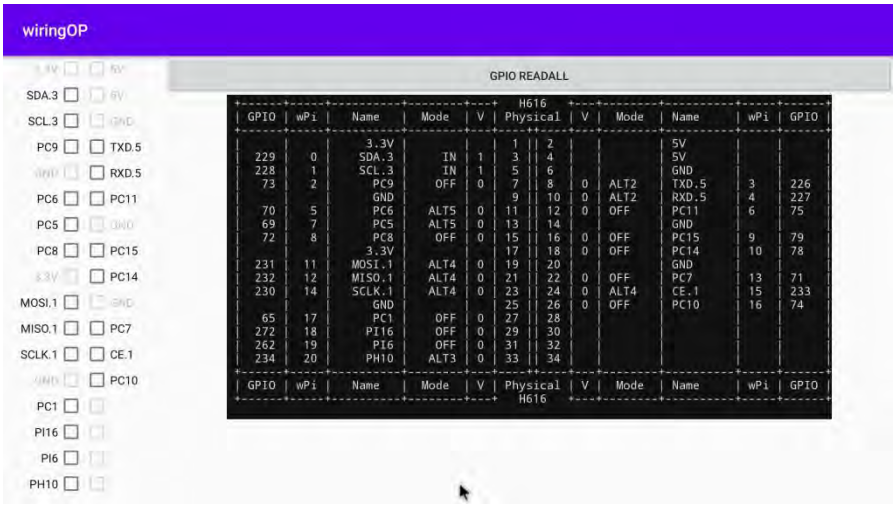
2) Then click the **GPIO_TEST** button to open the GPIO test interface



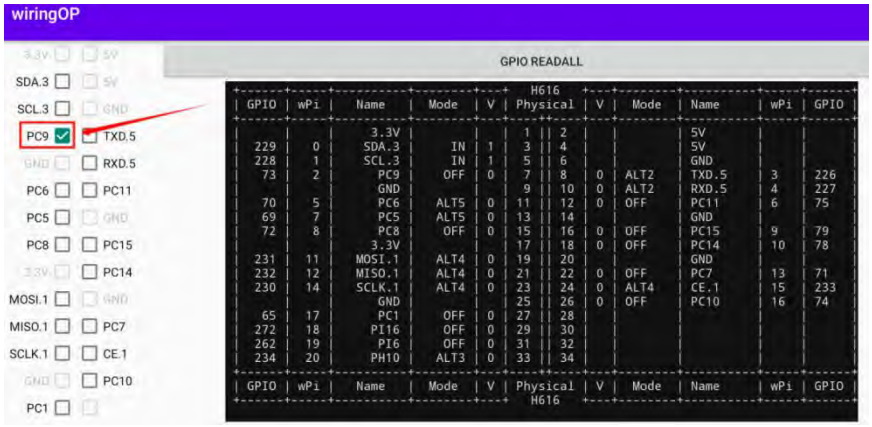
3) The GPIO test interface is shown in the figure below. There is a one-to-one correspondence between the two rows of **CheckBox** buttons on the left and the 26pin pins **(PC1/PI16/PI6/PH10 in the lower left corner are GPIO ports in the 13pin pins)**. When the **CheckBox** button is checked, the corresponding GPIO pin will be set to **OUT** mode, and the pin level will be set to high level; when the checkbox is unchecked, the GPIO pin level will be set to low level; When the **GPIO READALL** button is pressed, information such as the wPi number, GPIO mode, and pin level can be obtained.



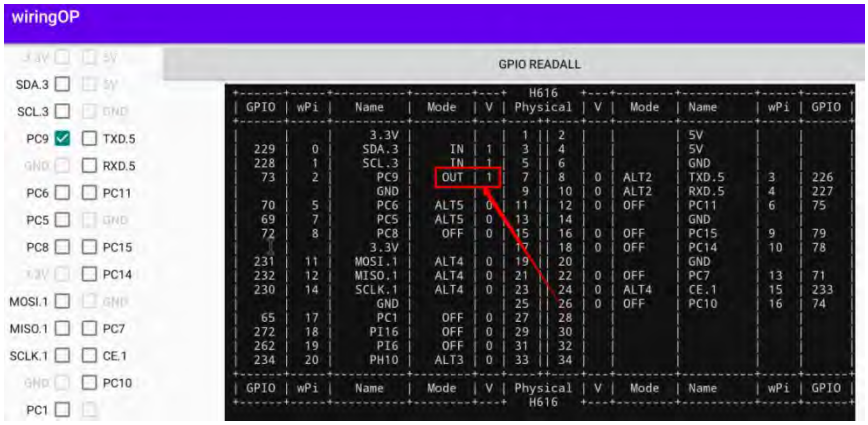
4) Then click the **GPIO READALL** button, the output information is as shown in the figure below：



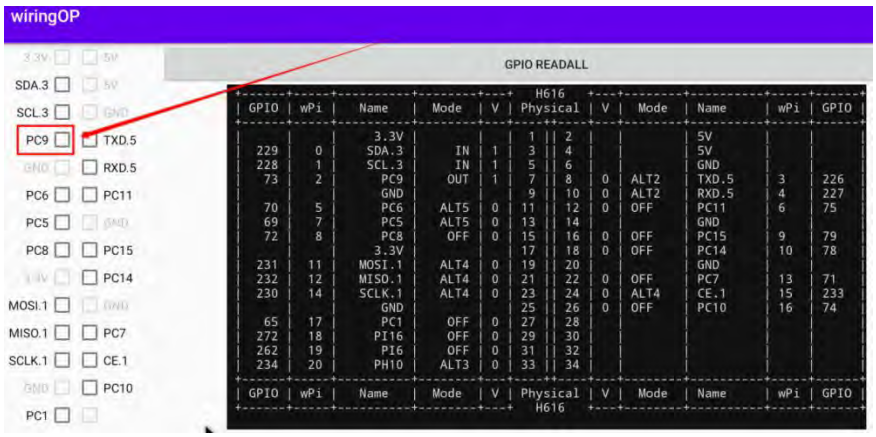
5) There are a total of 17 GPIO ports in the 26pin of the development board that can be used **(if you add 4 GPIO ports in the 13pin, then there are 21 in total)**, and the following pin No. 7 - corresponding to GPIO is PC9 - corresponding to wPi The serial number is 2——as an example to demonstrate how to set the high and low levels of the GPIO port. First click the **CheckBox** button corresponding to pin 7. When the button is selected, pin 7 will be set to high level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **3.3v**, it means setting high level success



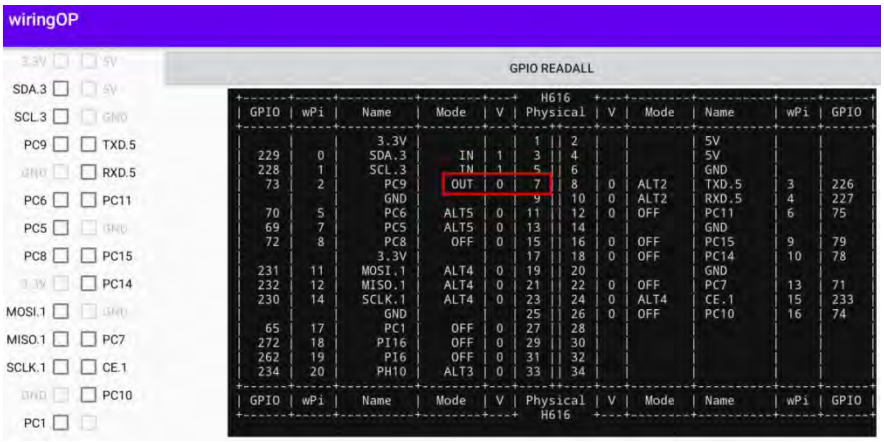
6) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is **OUT**, and the pin level is high



7) Click the **CheckBox** button in the figure below again to cancel the check status. Pin 7 will be set to low level. After setting, you can use a multimeter to measure the voltage value of the pin. If it is **0v**, it means that the low level is set successfully.



8) Then click the **GPIO READALL** button, you can see that the current pin 7 mode is OUT, and the pin level is low

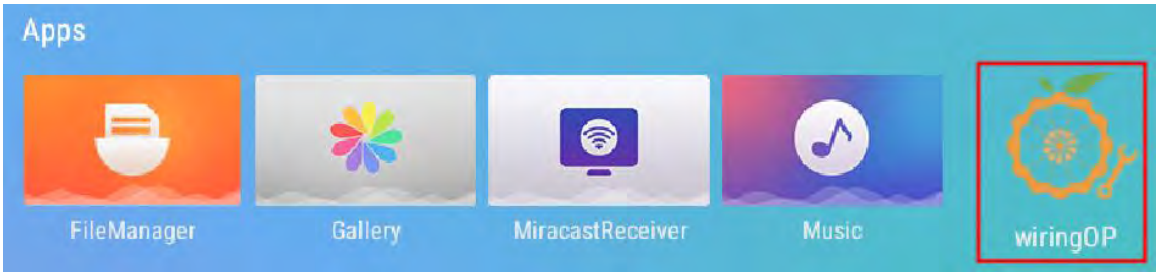


6. 15. 2. 26pin UART test method

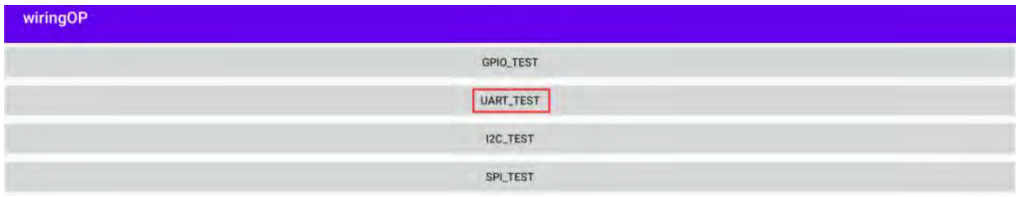
1) **UART5** is enabled by default in Android, and the corresponding device node is **/dev/ttyAS5**

```
apollo-p2:/ # ls /dev/ttyAS5
/dev/ttyAS5
```

2) First open the wiringOP APP on the desktop

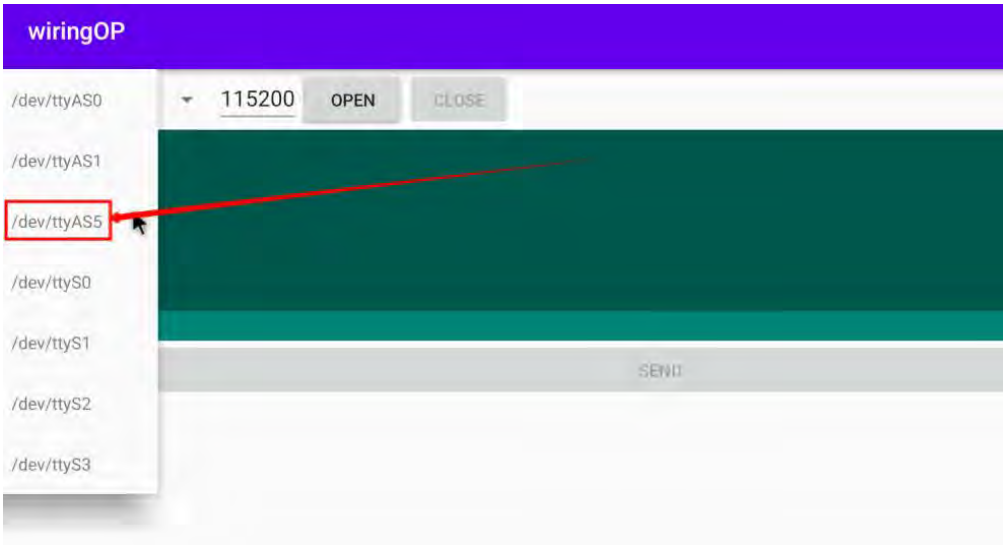


3) Then click the **UART_TEST** button to open the UART test interface

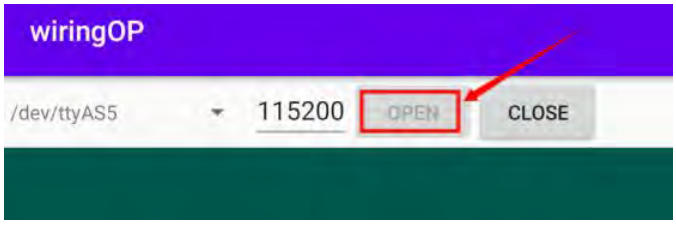


4) The serial port test interface of wiringOP is shown in the figure below

5) Then select the **/dev/ttyAS5** node in the selection box



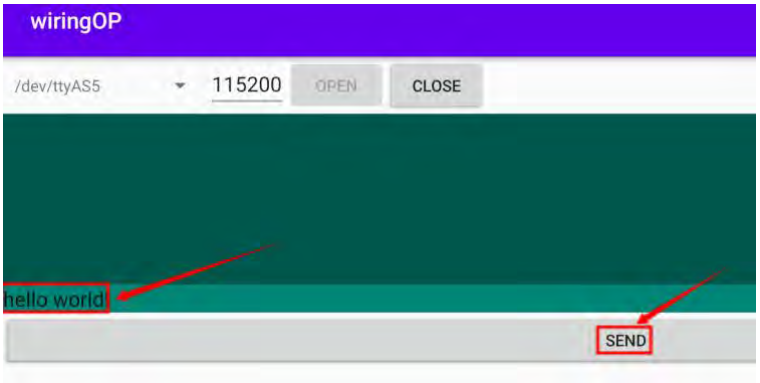
6) Enter the baud rate you want to set in the edit box, and then click the **OPEN** button to open the **/dev/ttyAS5** node. After the opening is successful, the **OPEN** button becomes unselectable, and the **CLOSE** button and **SEND** button become selectable.



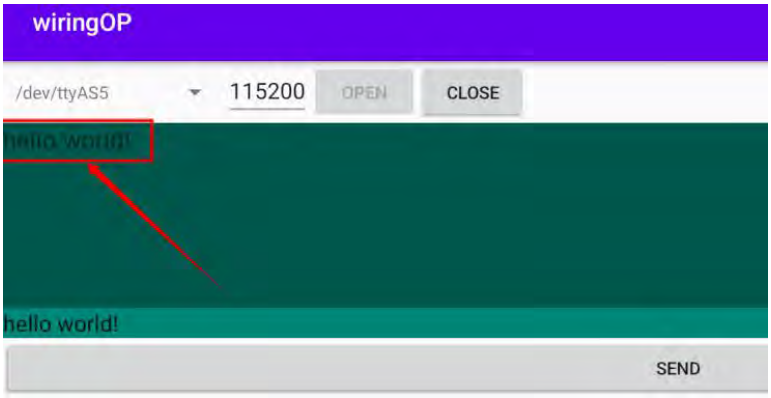
7) Then use Dupont wire to short the rx and tx pins of uart5

	uart5
Tx pin	Corresponding to pin 8 of 26pin
Rx pin	Corresponding to pin 10 of 26pin

8) Then you can enter a character in the send edit box below, click the **SEND** button to start sending

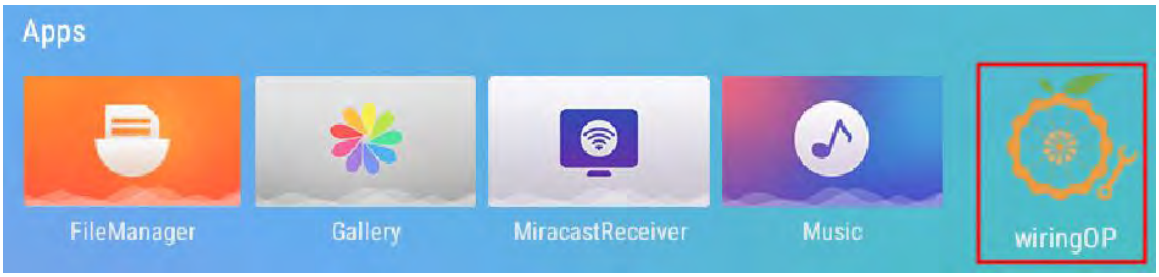


9) If everything is normal, the received string will be displayed in the receiving box



6. 15. 3. 26pin SPI test method

1) The SPI that can be used in 26pin is SPI1, and the corresponding device node is **/dev/spidev1.1**

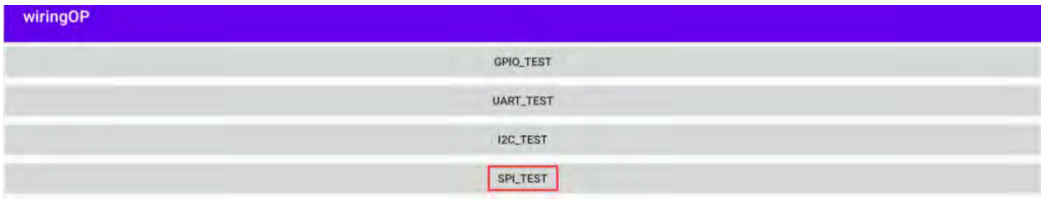


2) Here is a demonstration to test the SPI1 interface through the **w25q64** module, first connect the w25q64 module to the SPI1 interface

It doesn't matter if there is no w25q64 module, because there is a SPIFlash connected to SPI0 on the development board, and the configuration of SPI0 is also enabled by default in Android, so we can also directly use the onboard SPIFlash test.

3) Then open the wiringOP APP on the desktop

4) Then click the **SPI_TEST** button to open the SPI test interface

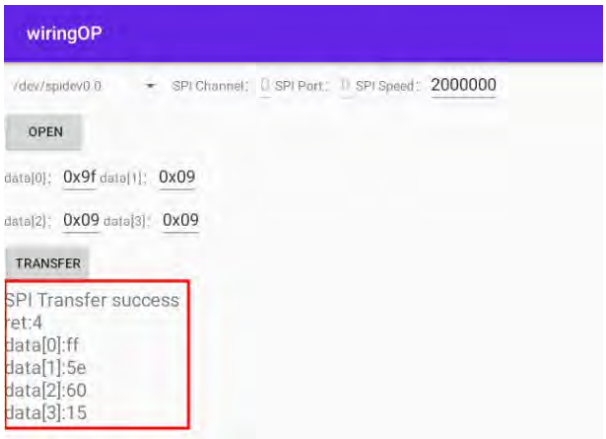


5) Then select the spi device node in the upper left corner. If you directly test the onboard SPIFlash, then keep the default **/dev/spidev0.0**. If you connect the **w25q64** module to the 26pin spi1, then please select /dev **/dev/spidev1.1**

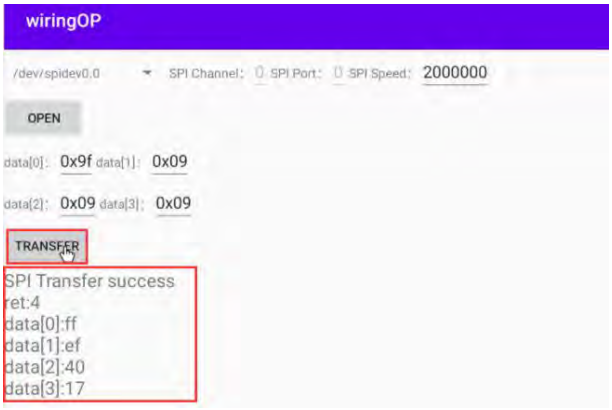
6) Then click the **OPEN** button to initialize the SPI

7) Then fill in the bytes that need to be sent, such as reading the ID information of the onboard SPIFlash, fill in the address 0x9f in data[0], and then click the **TRANSFER** button

8) Finally, the APP will display the read ID information of the onboard SPI Flash



9) If it is to read the w25q64 module connected to 26pin SPI1, then the read ID information is shown in the figure below



10) The MANUFACTURER ID of the w25q64 module is EFh, and the Device ID is 4017h, corresponding to the value read above (h stands for hexadecimal)

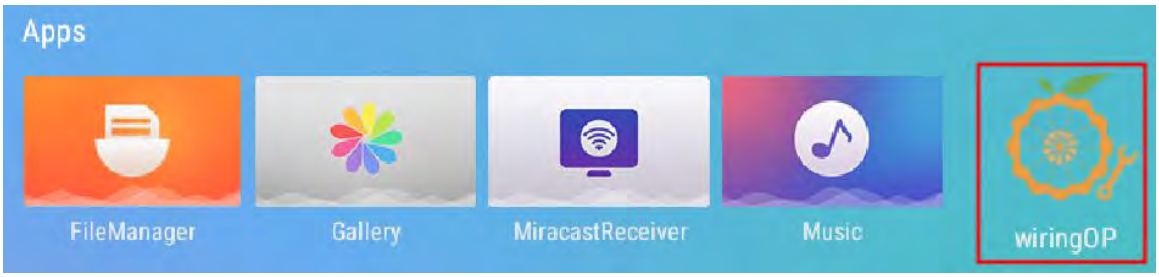
MANUFACTURER ID	**(MF7 - MF0)**	
Winbond Serial Flash	EFh	
Device ID	**(ID7 - ID0)**	**(ID15 - ID0)**
Instruction	**ABh, 90h, 92h, 94h**	**9Fh**
W25Q64FV (SPI)	16h	4017h
W25Q64FV (QPI)	16h	6017h

6. 15. 4. 26pin I2C test method

1) The i2c3 in 26pin is enabled by default in Android, and the corresponding device node is **/dev/i2c-3**

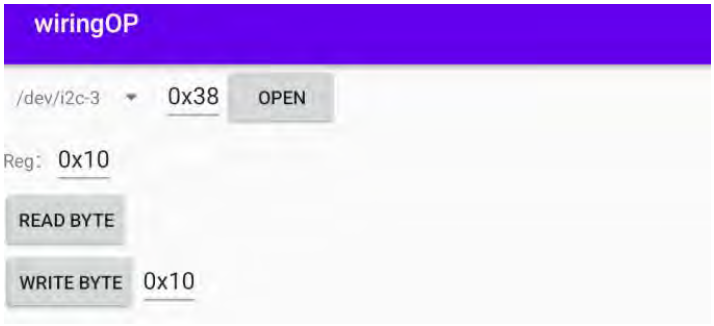
```
apollo-p2:/ # ls /dev/i2c-3
/dev/i2c-3
```

2) First open the wiringOP APP on the desktop

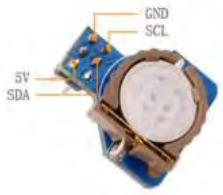


3) Then click the **I2C_TEST** button to open the i2c test interface

4) The i2c test interface of wiringOP is shown in the figure below. You can see that the default i2c is **/dev/i2c-3**, so we don’ t need to re-select



5) Then connect an i2c device to the 26pin i2c3 pin, here we take the ds1307 rtc module as an example



Pins of the RTC module	The pin corresponding to the 26pin of the development board
5V	**pin 2**
GND	**pin 6**
SDA	**pin 3**
SCL	**pin 5**

6) The i2c address of the ds1307 rtc module is 0x68. After connecting the wires, we can use the **i2cdetect -y 3** command in the serial port command line to check whether the i2c address of the ds1307 rtc module can be scanned. As shown in the figure below, if you can see the address 0x68, it means that the wiring of the ds1307 rtc module is correct.

```
apollo-p2:/ # i2cdetect -y 3
```

```
apollo-p2:/ # i2cdetect -y 3
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
apollo-p2:/ #
```

7) Then set the address of i2c to 0x68 in wiringOP, and then click the **OPEN** button to open i2c3

8) The display after clicking the **OPEN** button to open i2c3 is as follows：

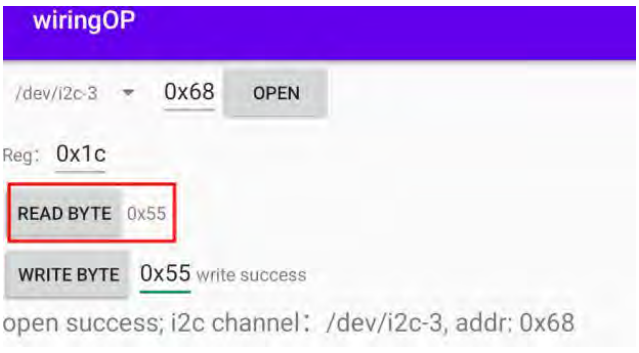
9) Then we test to write a value into the register of the rtc module, for example, write 0x55 to the 0x1c address

a. We first set the address of the register to be written to 0x1c

b. Then set the value to be written to 0x55

c. Then click the **WRITE BYTE** button to execute the write action

10) Then click the **READ BYTE** button to read the value of the 0x1c register, if it is displayed as 0x55, it means that the i2c read and write test has passed



7. How to compile Android 12 source code

7. 1. Download the source code of Android 12

1) First download the Android 12 source code sub-volume compressed package from Baidu or Google Disk

a. Baidu Netdisk

b. Google Disk

2) After downloading the sub-volume compression package of the Android 12 source code, please check whether the MD5 checksum is correct. If not, please download the source code again. The way to check the MD5 checksum is as follows:

```
test@test:~$ md5sum -c H618-Android12-Src.tar.gz.md5sum
H618-Android12-Src.tar.gzaa: OK
H618-Android12-Src.tar.gzab: OK
.....
```

3) Then you need to combine multiple compressed files into one, and then extract the Android source code. The command looks like this:

```
test@test:~$ cat H618-Android12-Src.tar.gza* > H618-Android12-Src.tar.gz
test@test:~$ tar -xvf H618-Android12-Src.tar.gz
```

7. 2. Compile the source code of Android 12

The compilation of Android12 is carried out on an x86_64 computer with Ubuntu 22.04 installed. Other versions of Ubuntu may have some differences in system package dependencies. The mirror download address of Ubuntu 22.04 amd64 version is as follows:
https://repo.huaweicloud.com/ubuntu-releases/22.04/ubuntu-22.04.2-desktop-amd64.iso

The x86_64 computer hardware configuration for compiling the Android12

source code is recommended to have 16GB or more memory, and 200GB or more hard disk space is recommended. The more CPU cores, the better.

1) First install the software packages required to compile the Android12 source code

```
test@test:~$ sudo apt-get update
test@test:~$ sudo apt-get install -y git gnupg flex bison gperf build-essential \
zip curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 \
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z1-dev ccache \
libgl1-mesa-dev libxml2-utils xsltproc unzip u-boot-tools python-is-python3 \
libssl-dev libncurses5 clang gawk
```

2) Then compile the code in the **longan** folder, which mainly contains u-boot and linux kernel

a. First run **./build.sh config** to set compilation options

```
test@test:~$ cd H618-Android12-Src/longan
test@test:~/H618-Android12-Src/longan$ ./build.sh config

Welcome to mkscript setup progress
All available platform:
    0. android
    1. linux
Choice [android]: 0
All available ic:
    0. h618
Choice [h618]: 0
All available board:
    0. ft
    1. p1
    2. p2
    3. p7
    4. p7l
    5. perf1
    6. perf2
    7. perf3
    8. qa
```

```
Choice [p2]: 2
All available flash:
    0. default
    1. nor
Choice [default]: 0
All available kern_ver:
    0. linux-5.4
Choice [linux-5.4]: 0
All available arch:
    0. arm
    1. arm64
Choice [arm64]: 1
......
*** Default configuration is based on 'sun50iw9p1smp_h618_android_defconfig'
#
# configuration written to .config
#
make[1]: Leaving directory '/home/test/H618-Android12-Src/longan/out/kernel/build'
make: Leaving directory '/home/test/H618-Android12-Src/longan/kernel/linux-5.4'
INFO: clean buildserver
INFO: prepare_buildserver
```

b. Then run the **./build.sh** script to start compiling

```
test@test:~/H618-Android12-Src/longan$ ./build.sh
```

c. After the compilation is complete, you will see the following output

```
sun50iw9p1 compile Kernel successful

INFO: Prepare toolchain ...
......
INFO: build kernel OK.
INFO: build rootfs ...
INFO: skip make rootfs for android
INFO: ----------------------------------------
INFO: build lichee OK.
INFO: ----------------------------------------
```

3) Then use the following command to compile the Android source code and generate the final Android image

```
test@test:~$ cd H618-Android12-Src
test@test:~/H618-Android12-Src$ source build/envsetup.sh
test@test:~/H618-Android12-Src$ lunch apollo_p2-userdebug
test@test:~/H618-Android12-Src$ make -j8
test@test:~/H618-Android12-Src$ pack
```

4) The storage path of the compiled and generated Android image is:

```
longan/out/h618_android12_p2_uart0.img
```

8. Appendix

8.1. User Manual Update History

Version	Date	Update Notes
v1.0	2023-07-05	initial version
v1.1	2023-07-14	Add the compilation method of Android 12 source code
v1.2	2023-09-05	Instructions for using the OrangePi OS Arch system
v1.3	2023-10-13	How to use wiringOP hardware PWM

8.2. Image update history

Date	Update Notes
2023-07-05	Orangepizero3_1.0.0_ubuntu_jammy_server_linux5.4.125.7z Orangepizero3_1.0.0_debian_bullseye_server_linux5.4.125.7z Orangepizero3_1.0.0_ubuntu_focal_desktop_xfce_linux5.4.125.7z Orangepizero3_1.0.0_ubuntu_jammy_desktop_xfce_linux5.4.125.7z Orangepizero3_1.0.0_debian_bullseye_desktop_xfce_linux5.4.125.7z Orangepizero3_1.0.0_ubuntu_jammy_server_linux6.1.31.7z Orangepizero3_1.0.0_debian_bookworm_server_linux6.1.31.7z

	Orangepizero3_1.0.0_debian_bullseye_server_linux6.1.31.7z Orangepizero3_1.0.0_ubuntu_jammy_desktop_xfce_linux6.1.31.7z Orangepizero3_1.0.0_debian_bookworm_desktop_xfce_linux6.1.31.7z Orangepizero3_1.0.0_debian_bullseye_desktop_xfce_linux6.1.31.7z OrangePi_Zero3_Android12_v1.0.tar.gz *initial version
2023-07-13	Opios-arch-aarch64-xfce-opizero3-23.07-linux6.1.31.img.xz * initial version
2023-10-07	Orangepizero3_1.0.2_ubuntu_jammy_server_linux6.1.31.7z Orangepizero3_1.0.2_debian_bookworm_server_linux6.1.31.7z Orangepizero3_1.0.2_debian_bullseye_server_linux6.1.31.7z Orangepizero3_1.0.2_ubuntu_jammy_desktop_xfce_linux6.1.31.7z Orangepizero3_1.0.2_debian_bookworm_desktop_xfce_linux6.1.31.7z Orangepizero3_1.0.2_debian_bullseye_desktop_xfce_linux6.1.31.7z * Solve the problem that HDMI still displays the logo after shutting down

FCC WARNING

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-- Reorient or relocate the receiving antenna.

-- Increase the separation between the equipment and receiver.

-- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

-- Consult the dealer or an experienced radio/TV technician for help.

To maintain compliance with FCC’s RF Exposure guidelines, This equipment should be installed and operated with minimum distance between 20cm the radiator your body: Use only the supplied antenna.

FCC ID: 2AQ9Y-OPIZERO3